

CAPÍTULO 5

GRÁFICOS. REPRESENTACIONES EN 2D Y EN 3D.

1.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA EXPLÍCITA

2.- REPRESENTACIÓN SIMULTÁNEA DE FUNCIONES EN 2D

3.- REPRESENTACIÓN DE FUNCIONES DEFINIDAS A TROZOS

4.- DIRECTIVAS GRÁFICAS: DE TAMAÑO Y DE COLOR

5.- ELEMENTOS GRÁFICOS EN 2D

6.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA PARAMÉTRICA

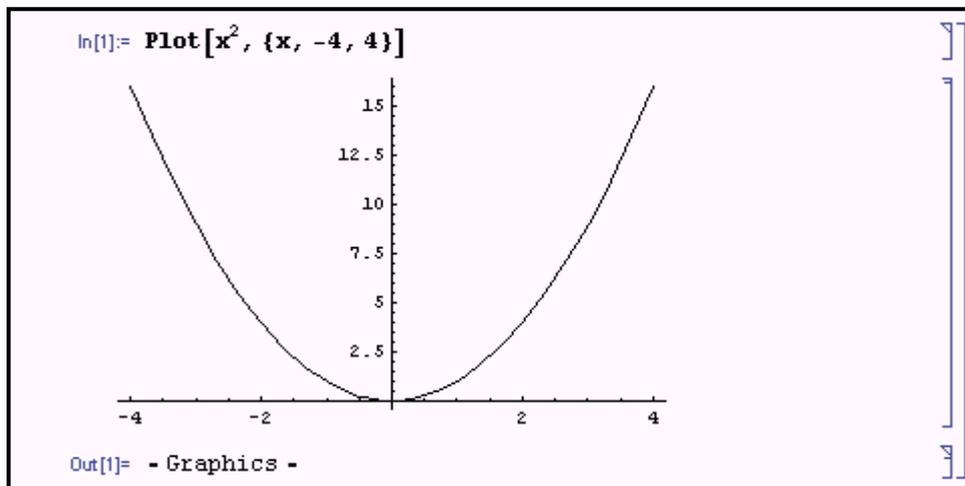
7.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA POLAR

8.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA IMPLÍCITA

1.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA EXPLÍCITA

La gráfica de una función se obtiene a partir del comando **Plot**, cuya sintaxis es:

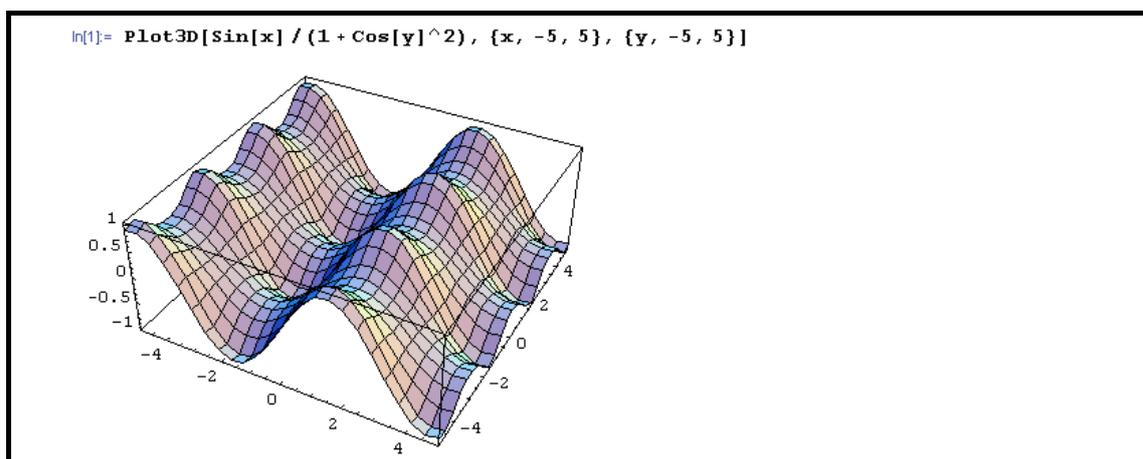
Plot[f,{x,a,b}]



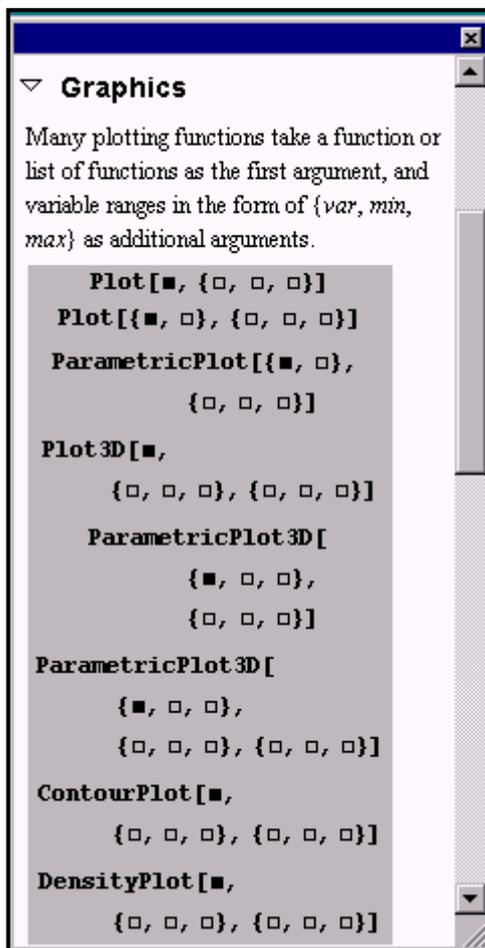
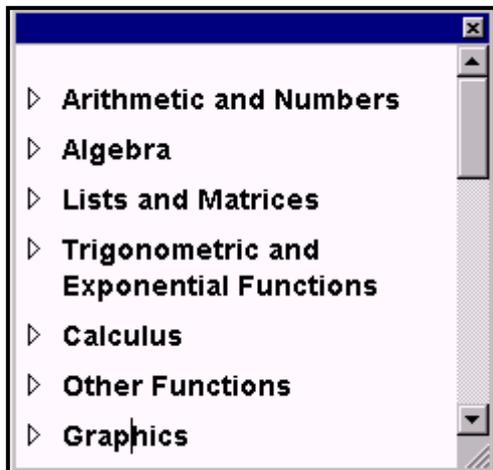
El mensaje de salida (-Graphics-), se puede omitir añadiendo como carácter final “;”.

La función **Plot3D** nos permite representar gráficamente funciones reales de dos variables definidas en forma explícita, su sintaxis es:

Plot3Df,{x,a,b},{y,c,d}]



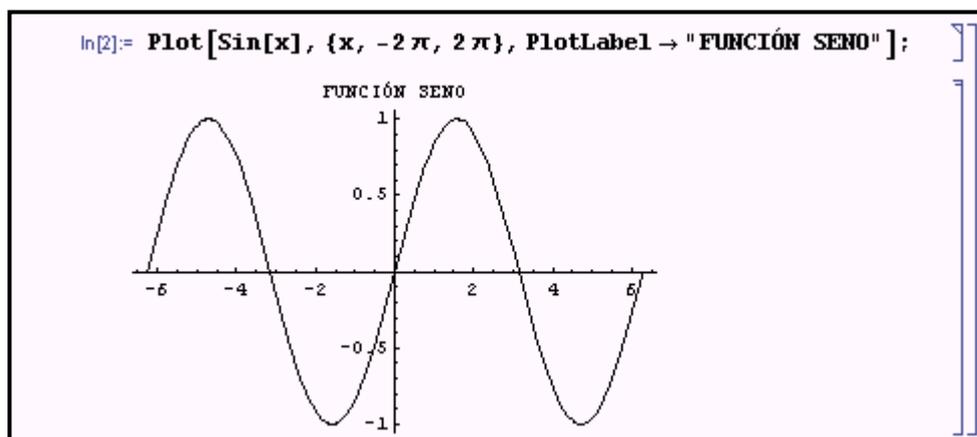
Las plantillas correspondientes a los distintos comandos gráficos aparecen en la paleta **BasicCalculations**. Al pulsar sobre la opción **Graphics** se mostrarán las diferentes plantillas.



Al representar una función existen unos parámetros con unos valores asignados por defecto, los cuales determinan el aspecto con el que aparece la función en pantalla. A continuación analizaremos las distintas opciones del comando **Plot** que permiten modificar el aspecto del gráfico.

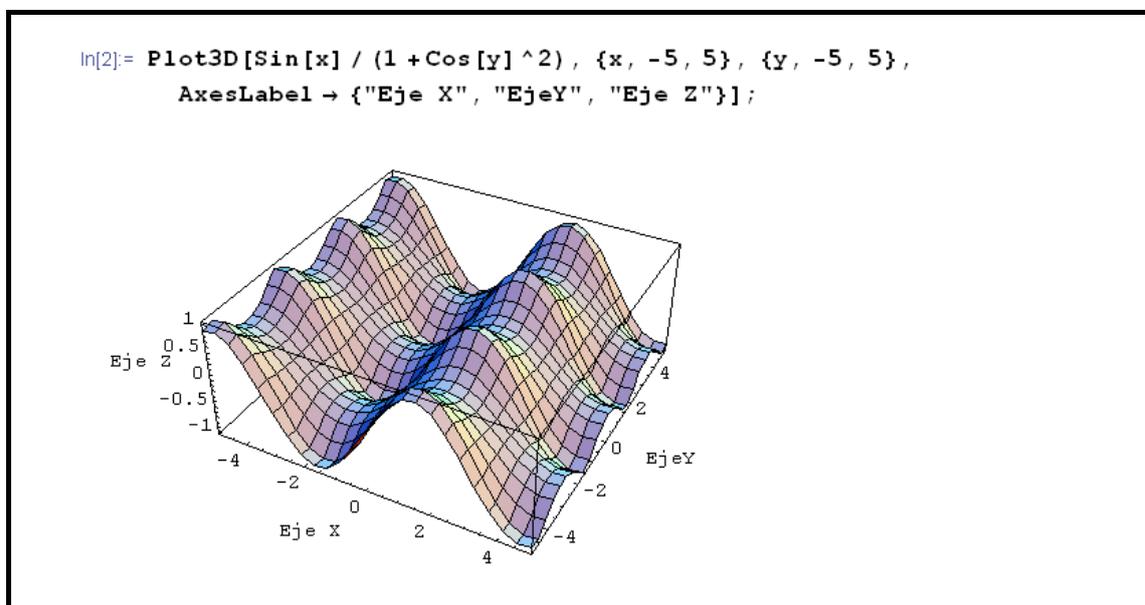
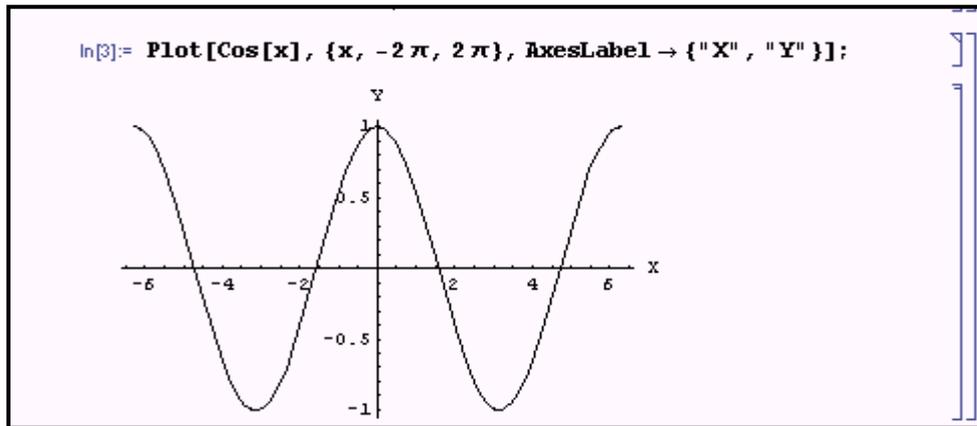
Título del gráfico

Incluyendo la opción **PlotLabel** junto a la representación de la función aparece una etiqueta a modo de título.



Representación de los ejes de coordenadas

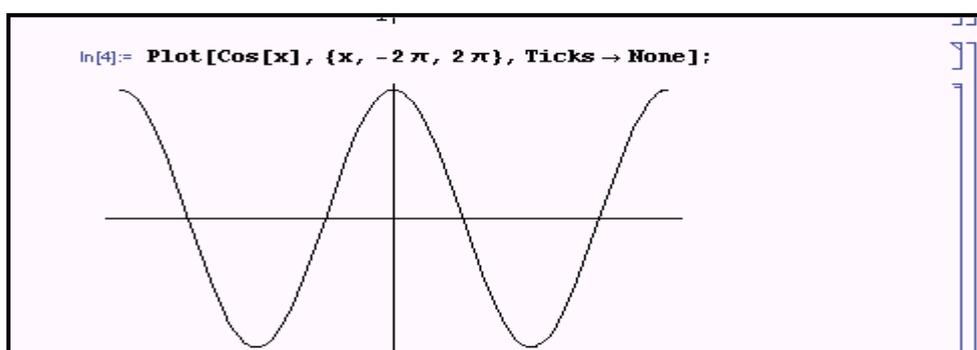
A través de la opción **AxesLabel** se podrá colocar una etiqueta en cada uno de los ejes.

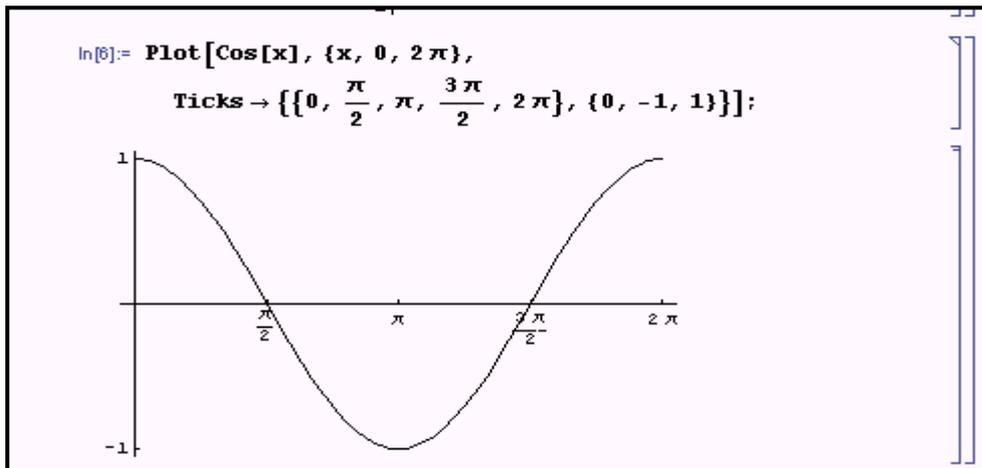


Por defecto los ejes de coordenadas aparecen con las marcas correspondientes para los distintos valores representados, los cuales se podrán suprimir utilizando la opción **Ticks** a la que se le asigna el valor **None**.

Esto permite que se especifiquen los valores que se desea representar en cada uno de los ejes,

Ticks->{valores eje de abscisas, valores eje de ordenadas}





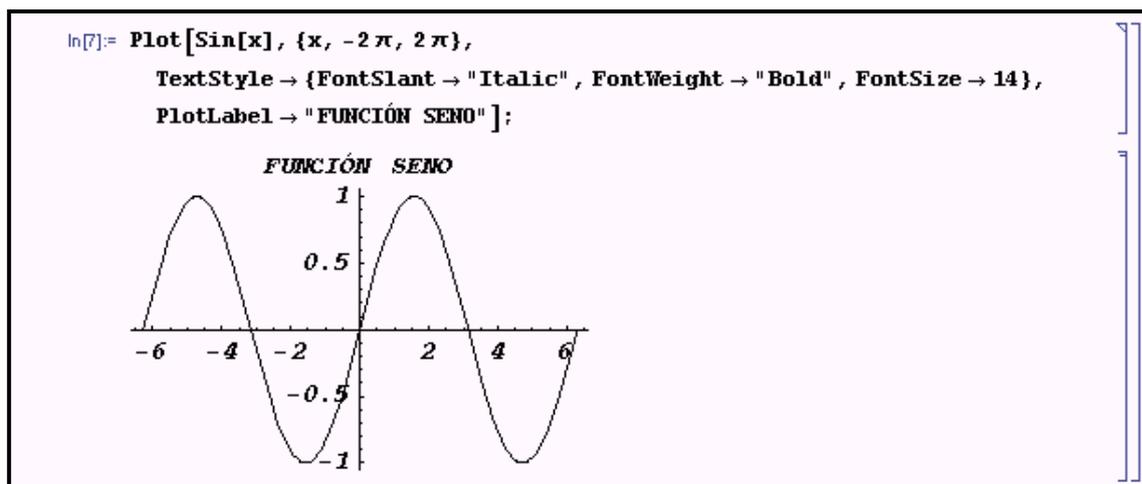
Fuente y Tamaño de caracteres

La opción **TextStyle** determina la fuente, características y tamaño de los caracteres que se incluirán en la representación del gráfico.

TextStyle->valores

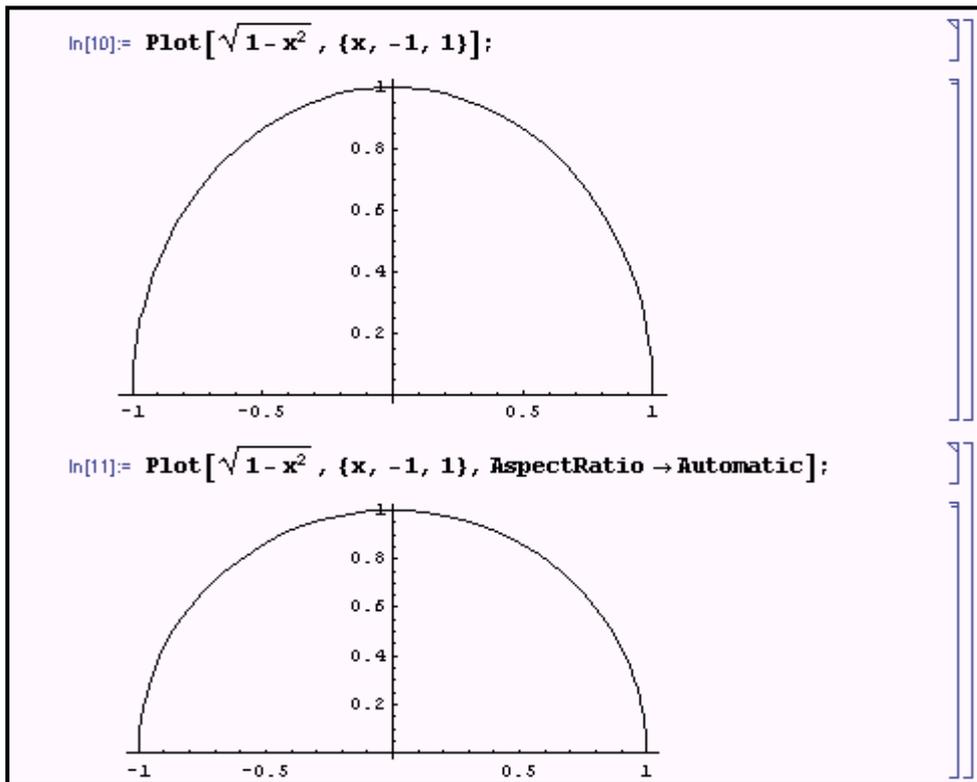
Como valores se podrán incluir:

- FontSize->n** tamaño de puntos
- FontSlant->"Italic"** caracteres en cursiva
- FontWeight->"Bold"** caracteres en negrita



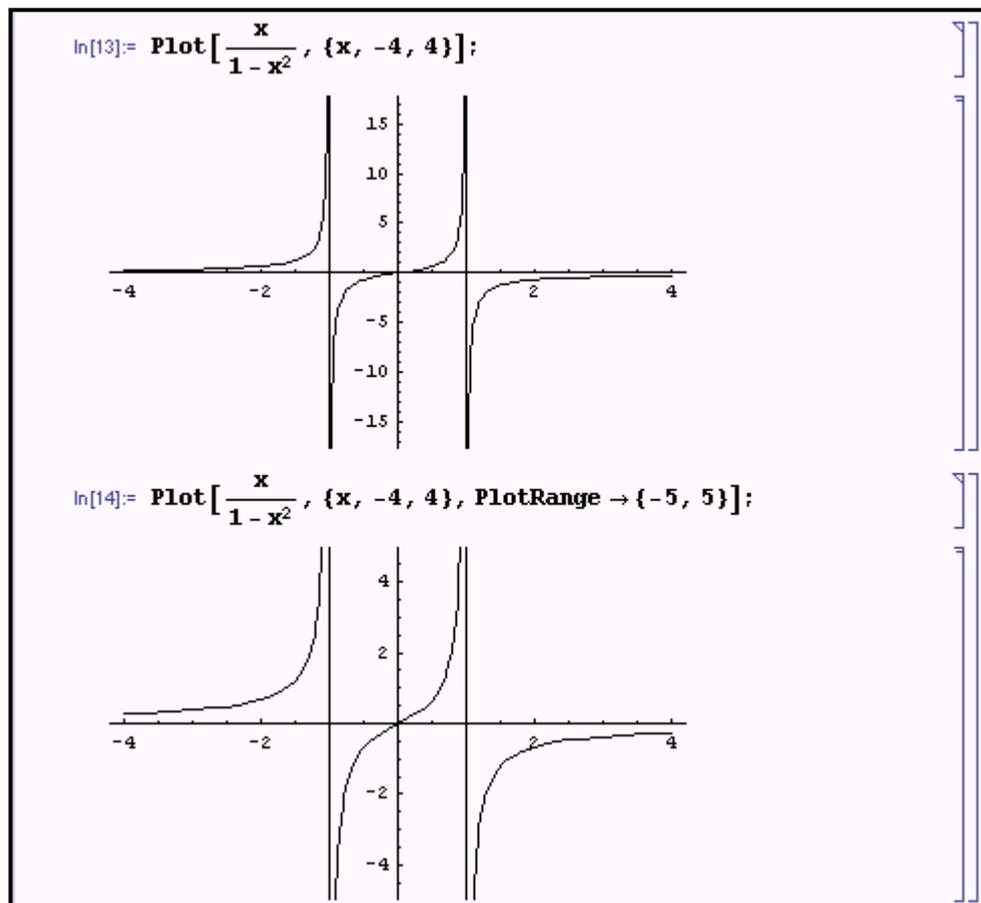
Proporciones del gráfico

La opción **AspectRatio** establece la proporción entre la altura y la anchura para la representación del gráfico.



Rango de valores

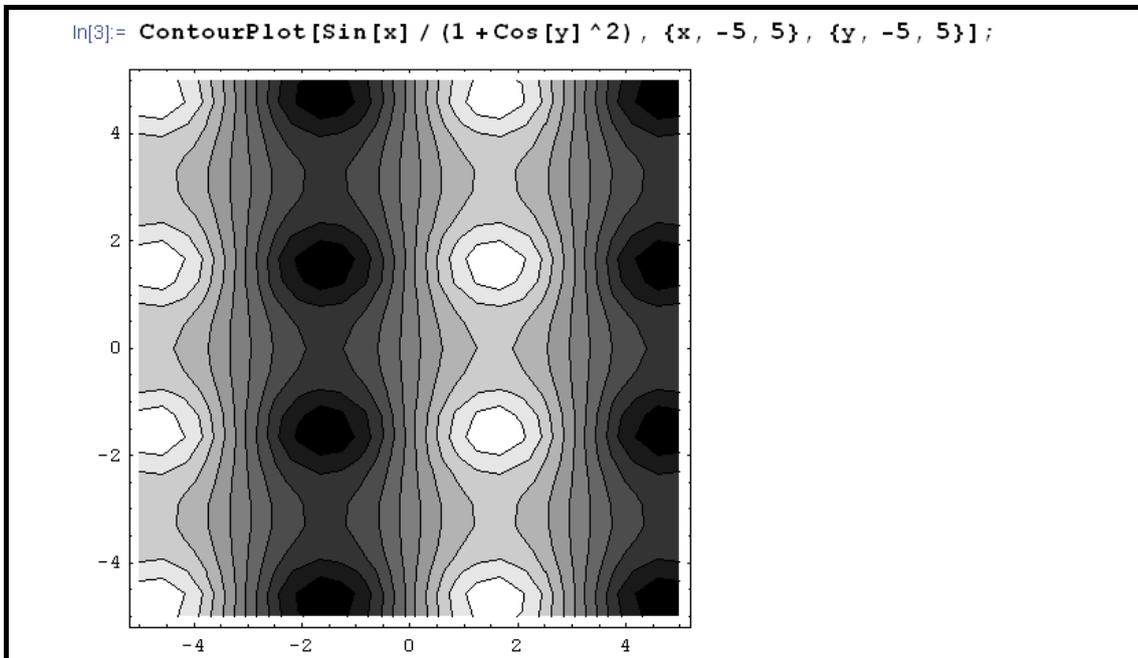
La opción **PlotRange** establece el rango de valores para las imágenes de los puntos que aparecen en la representación de la función. La sintaxis de esta opción es **PlotRange->**{ y_1, y_2 }.



Curvas de nivel de la función $z=f(x,y)$

Si queremos dibujar las curvas de nivel de una función real de dos variables, utilizaremos la función **ContourPlot**, de sintaxis:

ContourPlot[f,{x,a,b},{y,c,d}]

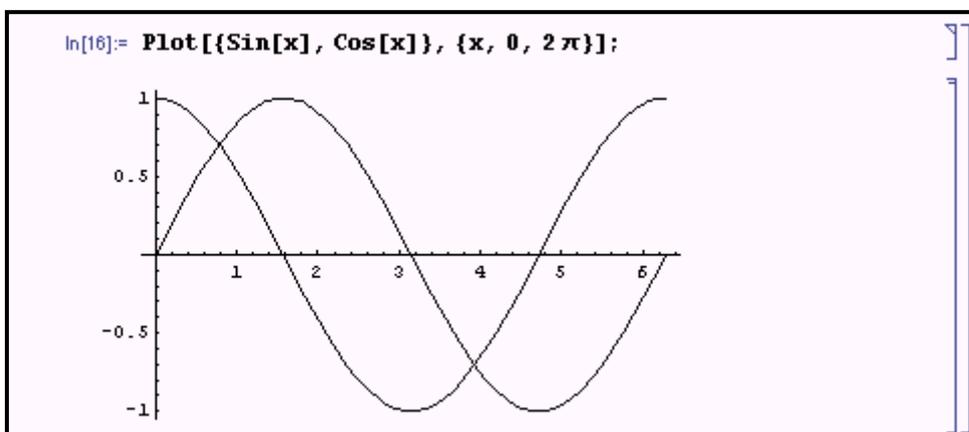


2.- REPRESENTACIÓN SIMULTÁNEA DE FUNCIONES EN 2D

La representación simultánea de varias funciones se realiza a través del comando **Plot**, utilizando la sintaxis:

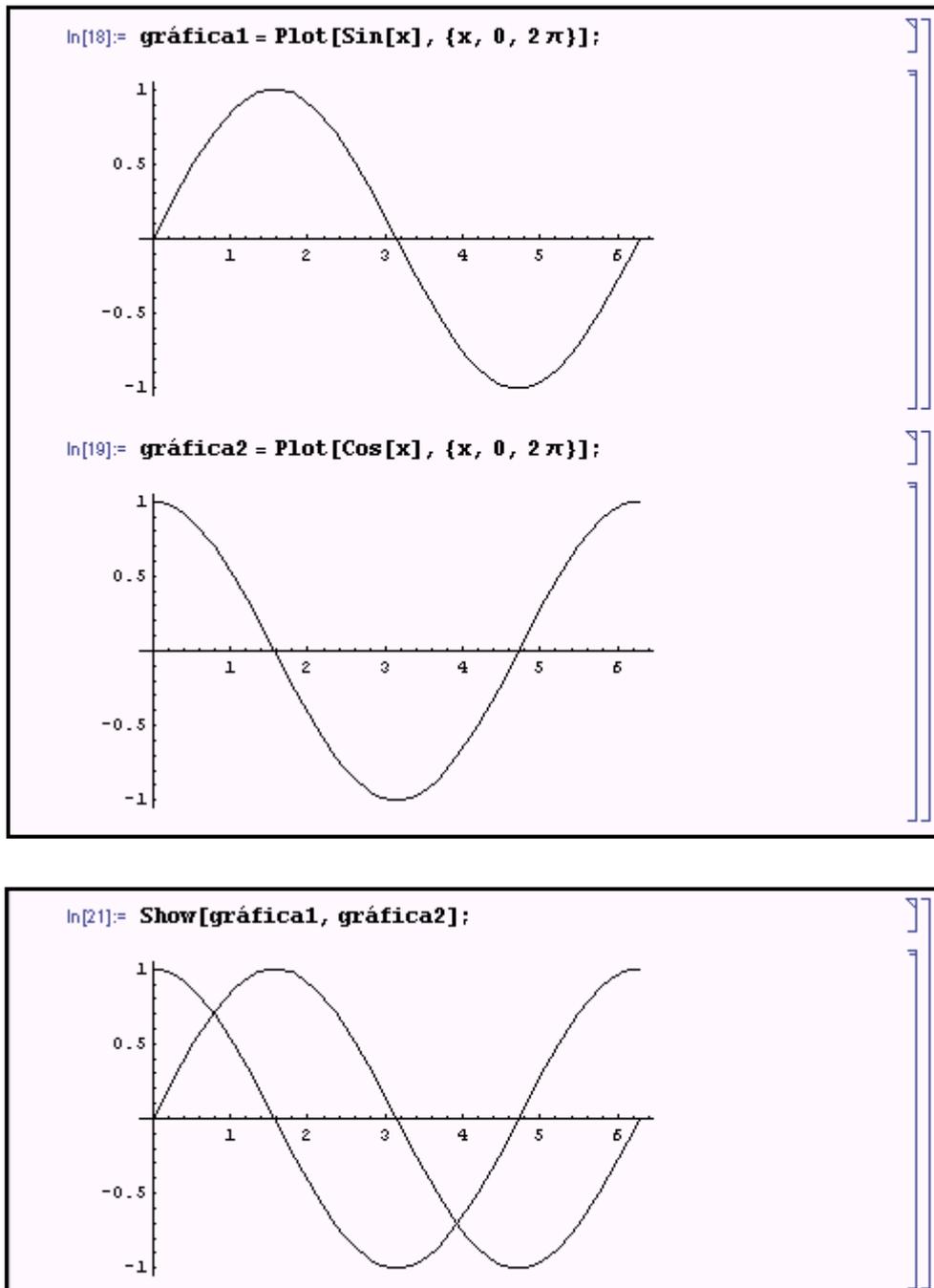
Plot[{f,g},{x,a,b}]

siendo f, g las funciones que se representan en el intervalo [a,b].



Un gráfico, obtenido mediante el comando **Plot**, se volverá a dibujar utilizando el comando **Show**. Mediante este comando conseguiremos representar de

manera simultánea varias funciones cuyas gráficas se han dibujado previamente.

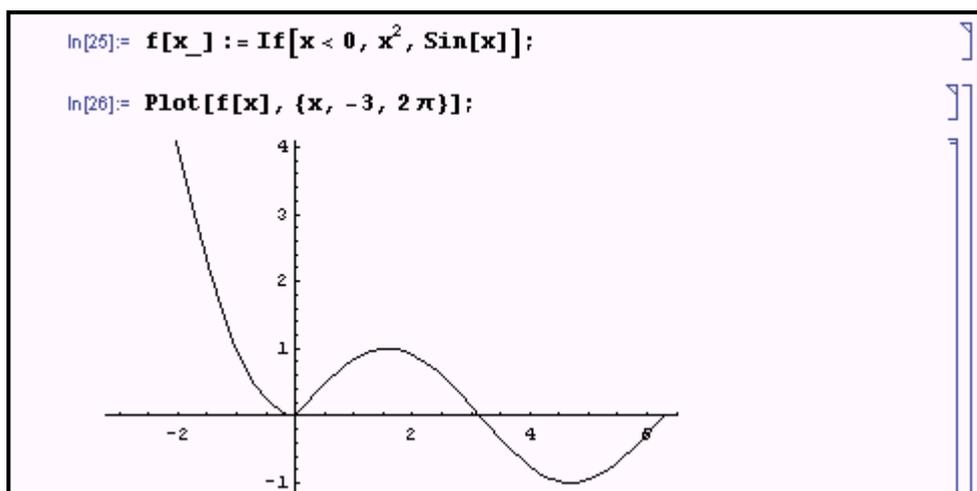


3.- REPRESENTACIÓN DE FUNCIONES DEFINIDAS A TROZOS

La representación de una función a trozos se realiza a través de los comandos **Plot** e **If**.

EJERCICIO 1

Representar la función $f(x) = \begin{cases} x^2 & \text{si } x < 0 \\ \text{sen } x & \text{si } x \geq 0 \end{cases}$



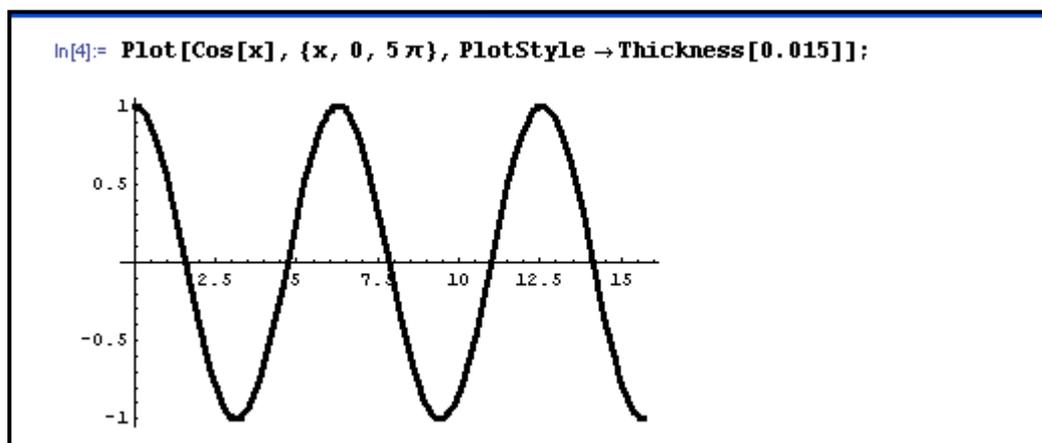
4.- DIRECTIVAS GRÁFICAS 2D: DE TAMAÑO Y DE COLOR

Las directivas gráficas se utilizan para establecer el estilo de un gráfico y se asignan como valores en la opción **PlotStyle**.

DIRECTIVAS DE TAMAÑO

Grosor en el trazo

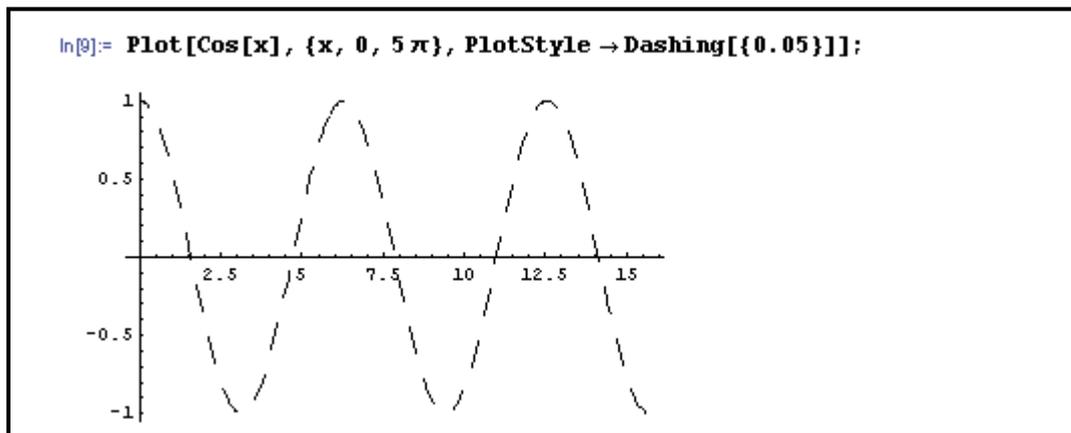
La directiva denominada **Thickness** determina el grosor de la línea utilizada para representar el elemento gráfico.



El valor asignado a esta directiva indica que el grosor de la línea será del 1,5% con respecto al tamaño total de la gráfica realizada.

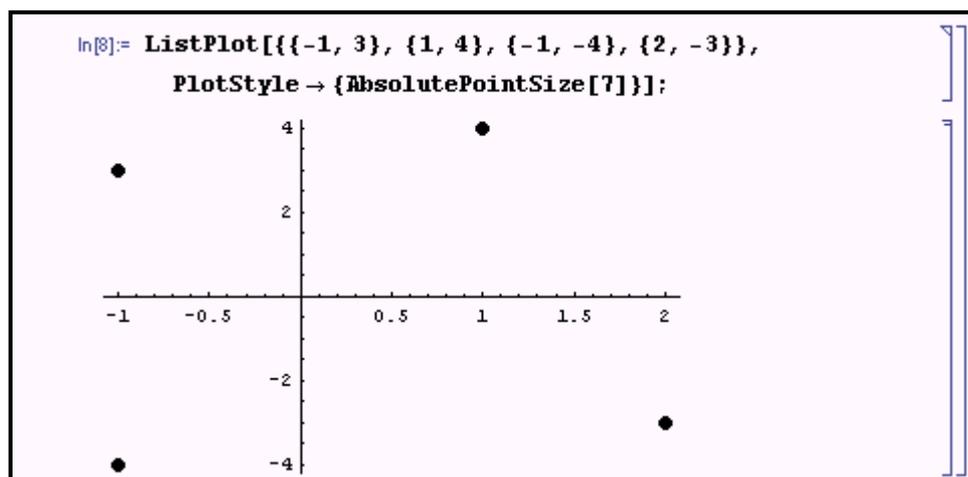
Patrón de trazado

Mediante la directiva **Dashing**, cuya sintaxis es **Dashing[{t}]**, se determina el trazado del gráfico. El valor **t** representa una línea discontinua en la que cada trazo es de una longitud **t%** de la anchura total de la gráfica, dejando entre cada dos trazos un espacio en blanco de igual longitud.



Tamaño de los puntos

La directiva **PointSize** determina el tamaño de los puntos. La sintaxis es **PointSize[r]** o bien **AbsolutePointSize[r]** donde r es el radio del círculo con el que se dibujaran cada uno de los puntos.



DIRECTIVAS DE TAMAÑO

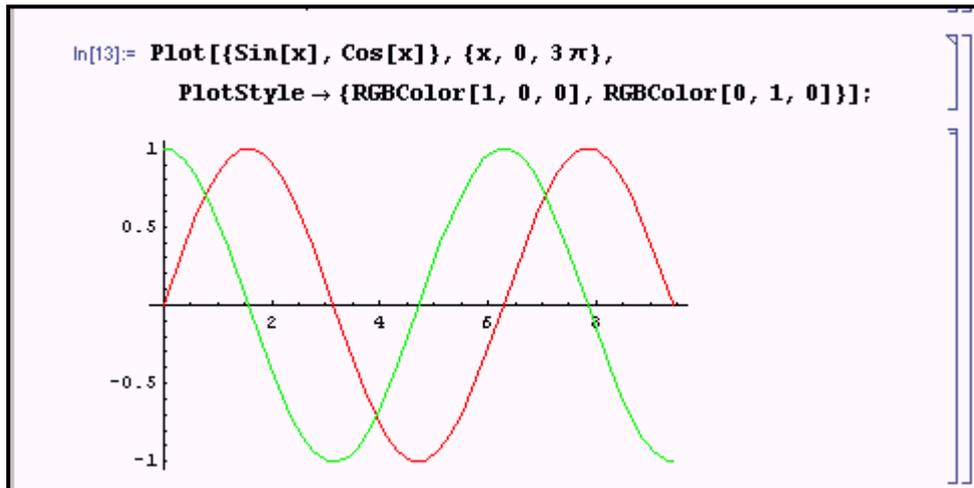
Incluyen atributos de color en los distintos elementos gráficos.

GrayLevel

Esta directiva de sintaxis **GrayLevel[g]** indica el nivel de gris que se utiliza para representar la gráfica, en la que g es un valor entre 0 y 1 que representan los valores negro y blanco respectivamente.

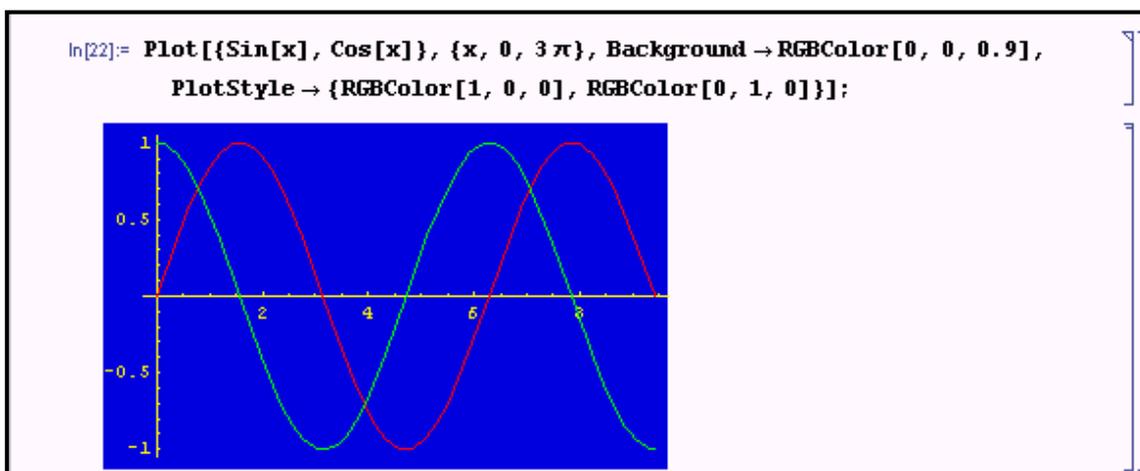
RGBColor

Con la notación **RGBColor[r,v,a]** indicamos el color que se utiliza en la representación, siendo r , v y a valores entre 0 y 1 que corresponden a los componentes rojo, verde y azul respectivamente.



Color del fondo

Especificado mediante la opción **Background**, utiliza como argumento cualquiera de las directivas de color anteriores.



5.- ELEMENTOS GRÁFICOS EN 2-D

A través del comando **Graphics**, de sintaxis **Graphics[elemento]**, vamos a presentar una serie de elementos gráficos. En este comando se pueden incluir las diferentes opciones de estilo, tamaño y color, expuestas anteriormente. En este caso la sintaxis es: **Graphics[elemento, opción->valor, ...]**

Punto

Mediante la instrucción **Pont[{x,y}]** se define el punto de coordenadas (x,y).



El tamaño del punto que aparece puede cambiarse utilizando la opción **PontSize**.

```
In[5]:= Show[Graphics[{PointSize[0.02], Point[{2, 2}]}];
```

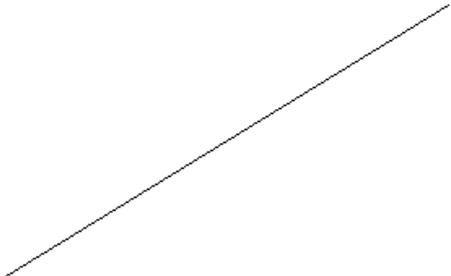


Segmento

El segmento de extremos los puntos (a,b) y (c,d) se define mediante la instrucción:

Line[{{a,b},{c,d}}]

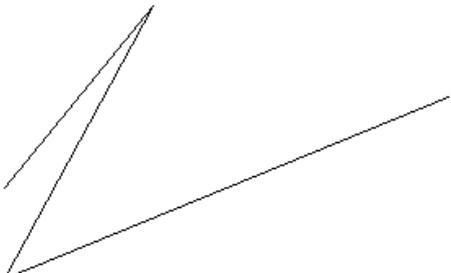
```
In[6]:= Line[{{1, 1}, {2, 5}}];
In[7]:= Show[Graphics[%]];
```



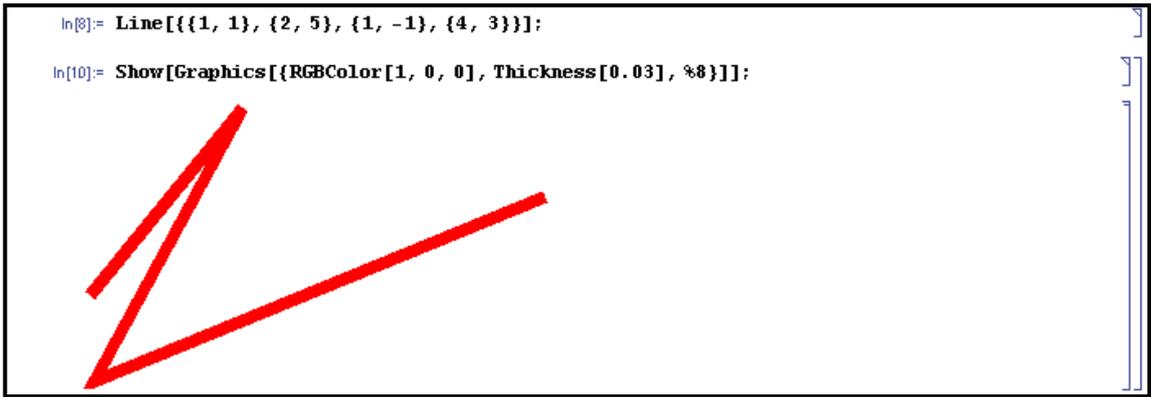
Se construirá una línea quebrada formada por varios segmentos al unir el extremo de uno con el punto inicial del siguiente. La sintaxis es:

Line[{{a,b},{c,d},{e,f},...}]

```
In[8]:= Line[{{1, 1}, {2, 5}, {1, -1}, {4, 3}}];
In[9]:= Show[Graphics[%]];
```

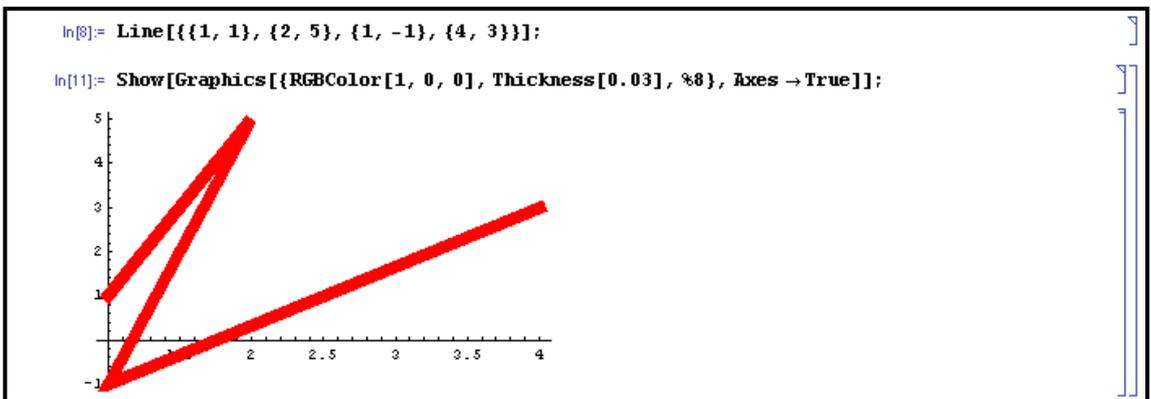


Es posible incluir cualquier opción para modificar el aspecto del gráfico.



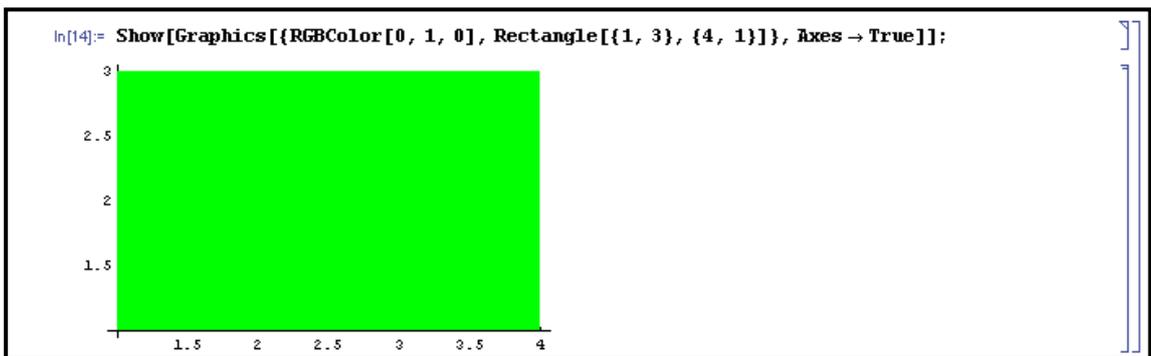
Los ejes de coordenadas se representan al incluir la opción **Axes** mediante la expresión:

Show[Graphics[segmento,Axes->True]]



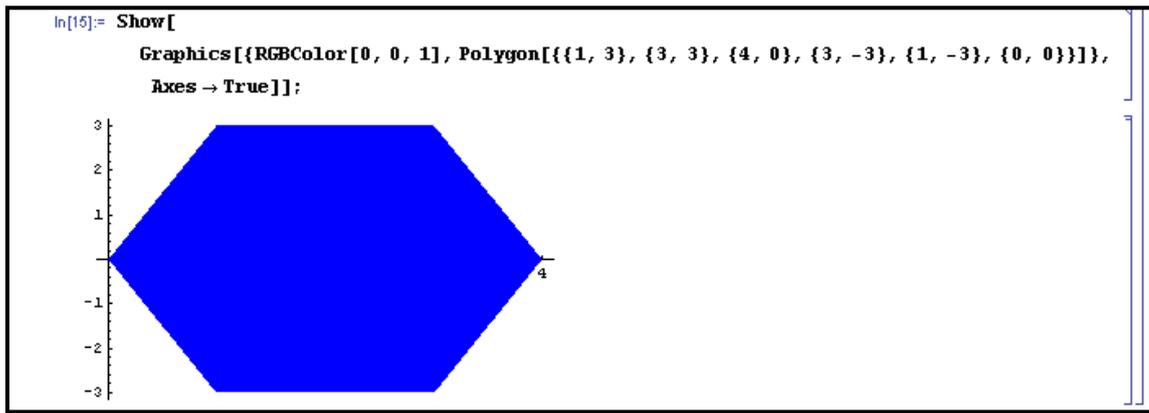
Rectángulo

Mediante la instrucción **Rectangle[*{a,b}*,*{c,d}*]** se define un rectángulo cuyos vértices opuestos corresponden a los puntos de coordenadas (a,b) y (c,d).



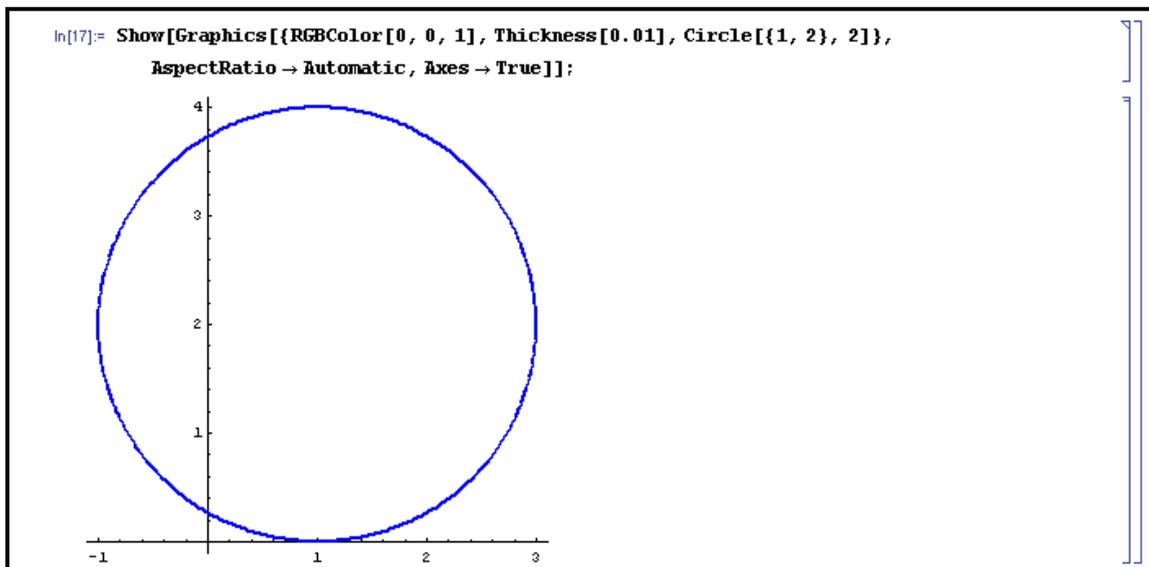
Polígono

Para definir un polígono cuyos vértices conocemos se emplea la instrucción **Polygon**, de sintaxis **Polygon[*{{a,b}*,*{c,d}*,*{e,f}*,...}**



Circunferencia y Elipse

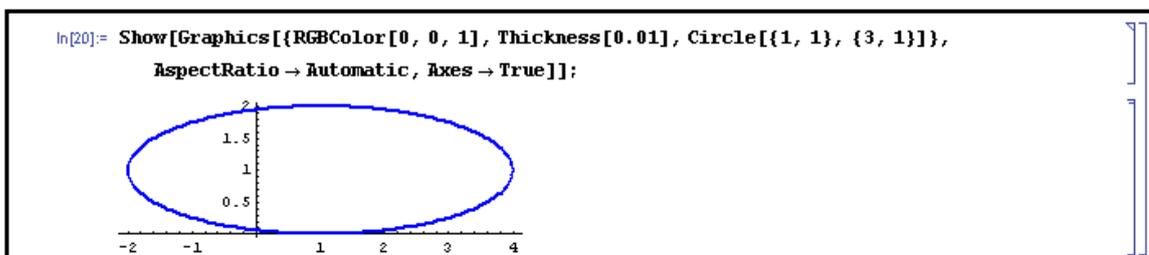
Para definir una circunferencia de centro (a,b) y radio r , se utiliza la instrucción **Circle** de sintaxis **Circle[{a,b},r]**. Para obtener visualmente la imagen de circunferencia debemos añadir la sentencia **AspectRatio -> Automatic**.



Para definir una elipse utilizaremos también la instrucción **Circle**, con la sintaxis:

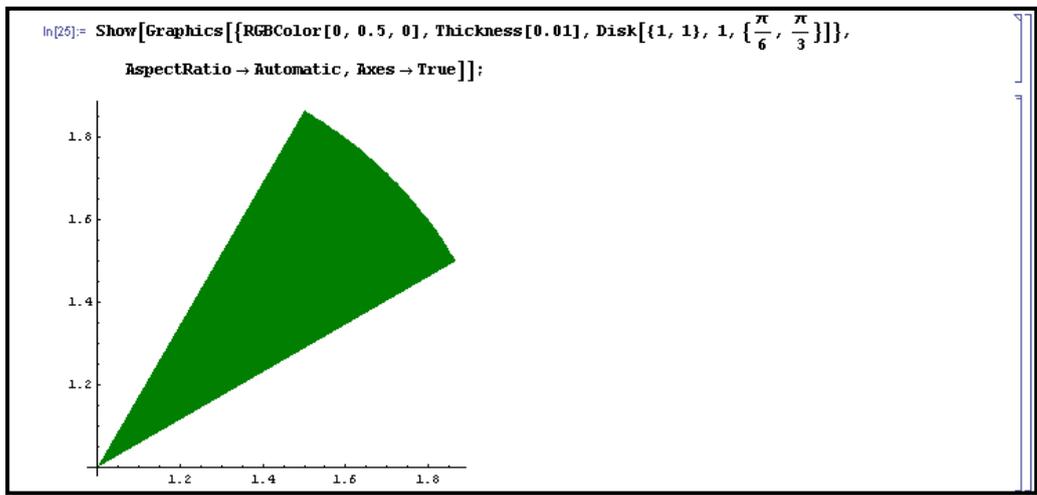
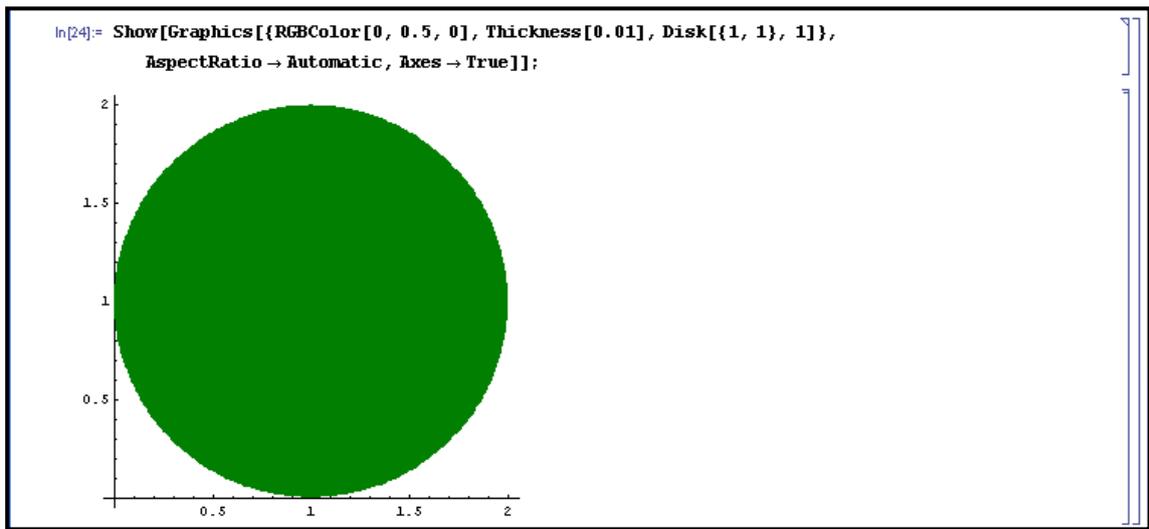
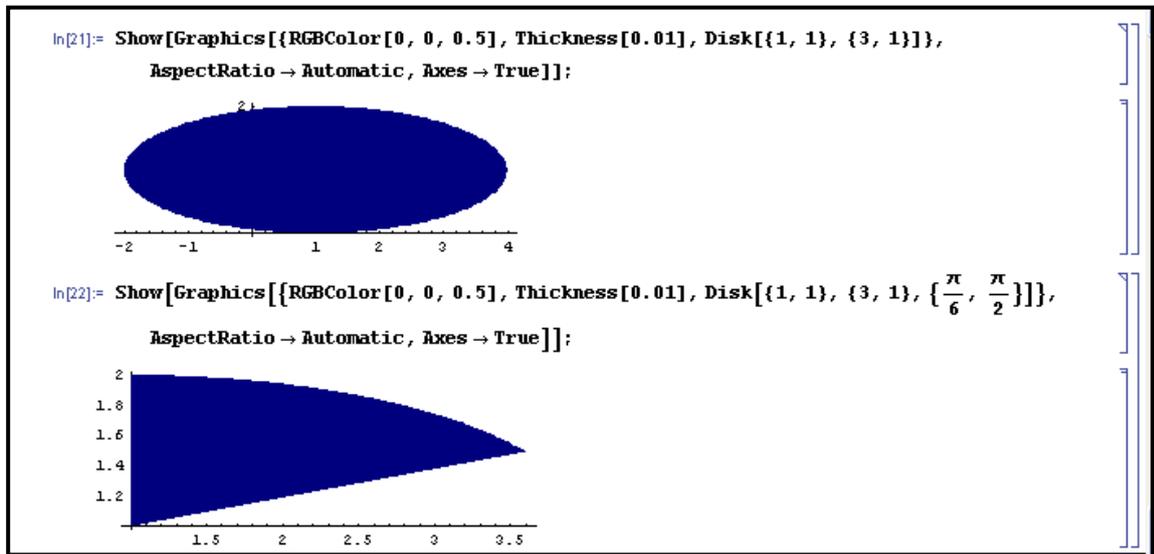
Circle[{a,b},{r₁,r₂}]

siendo (a,b) el centro de la elipse y r_1 y r_2 las longitudes de los semiejes.



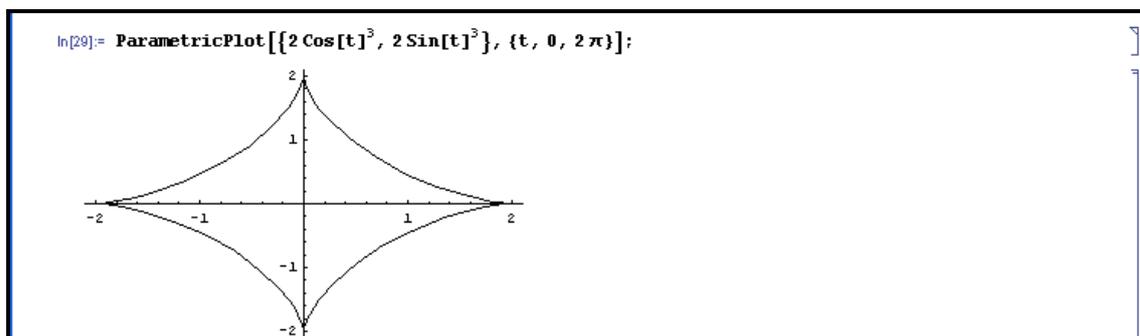
La representación de círculos, sectores circulares y elipses con el fondo de color, se realiza a través de la función **Disk**. Las distintas opciones son:

- Disk[{a,b},r]** Círculo de centro (a,b) y radio r
- Disk[{a,b},{r₁,r₂}** Elipse de centro (a,b) y semiejes r₁ y r₂
- Disk[{a,b},r,{x,y}]** Sector circular comprendido entre los ángulos x,y
- Disk[{a,b},{r₁,r₂},{x,y}]** Sector elíptico



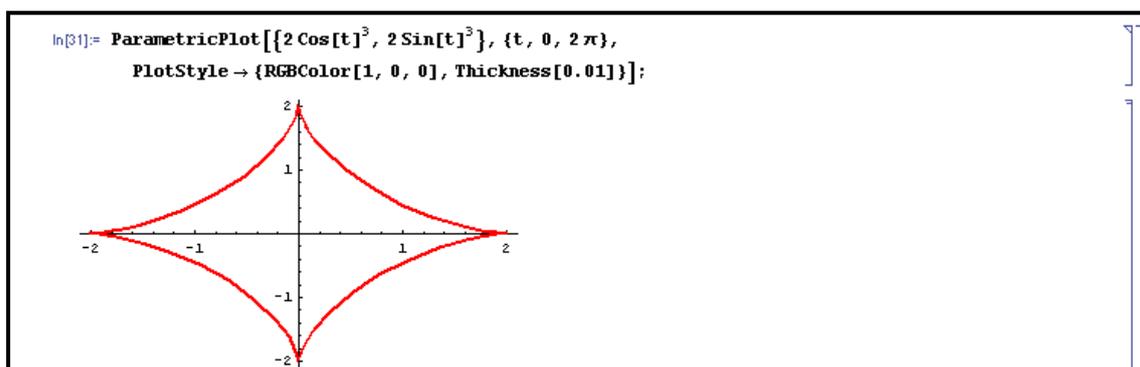
6.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA PARAMÉTRICA

Para representar una función expresada en forma paramétrica utilizaremos el comando **ParametricPlot**, cuya sintaxis es **ParametricPlot[{x,y},{t,a,b}]**.



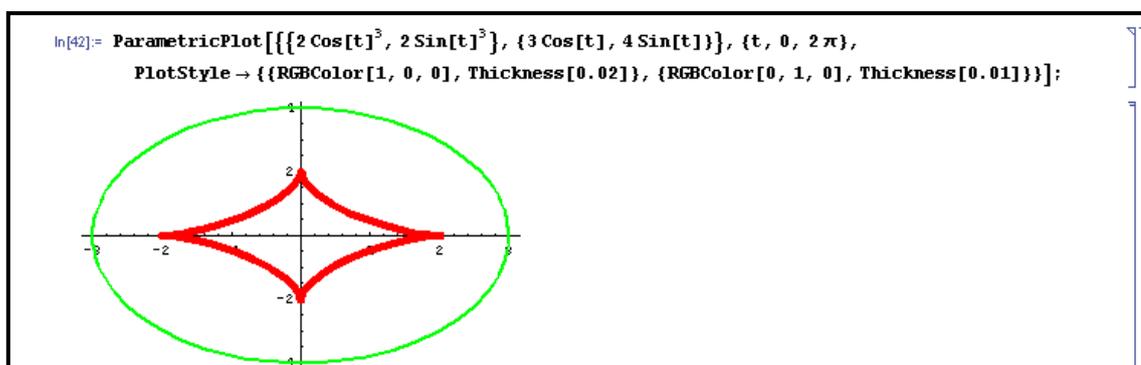
Se pueden incluir las distintas opciones para cambiar el aspecto del gráfico final, en este caso escribiremos:

ParametricPlot[{x,y},{t,a,b},opcion->valor]



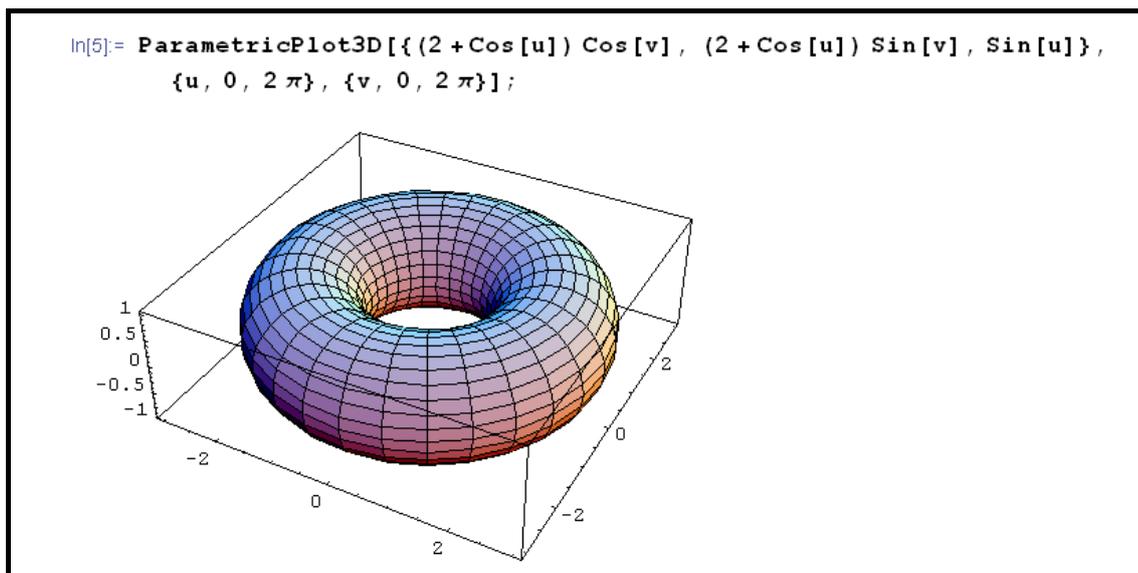
Para representar varias funciones en paramétricas seguiremos una notación similar.

ParametricPlot[{{x,y},{z,h}, ...}, {t,a,b},opcion->valor]

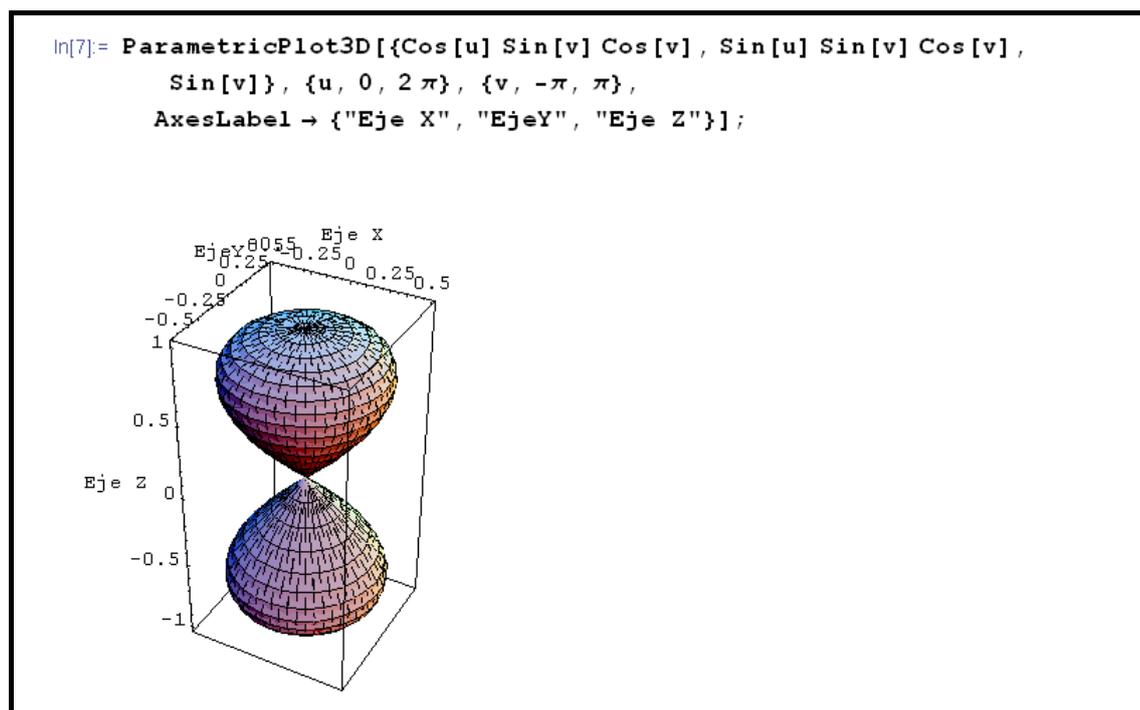


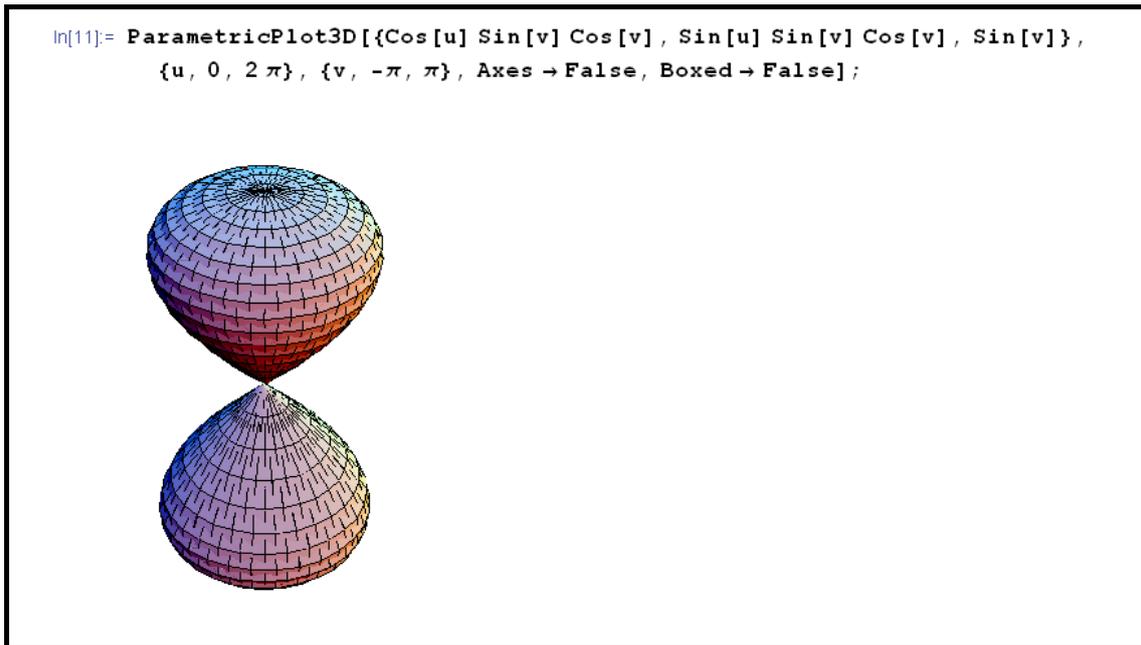
La función **ParametricPlot3D** nos permite representar gráficamente funciones de dos variables reales definidas en forma paramétrica. Su sintaxis es:

ParametricPlot3D[{f[u,v],g[u,v],h[u,v]},{u,a,b},{v,c,d}]



Se pueden incluir las distintas opciones para cambiar el aspecto del gráfico final.





7.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA POLAR

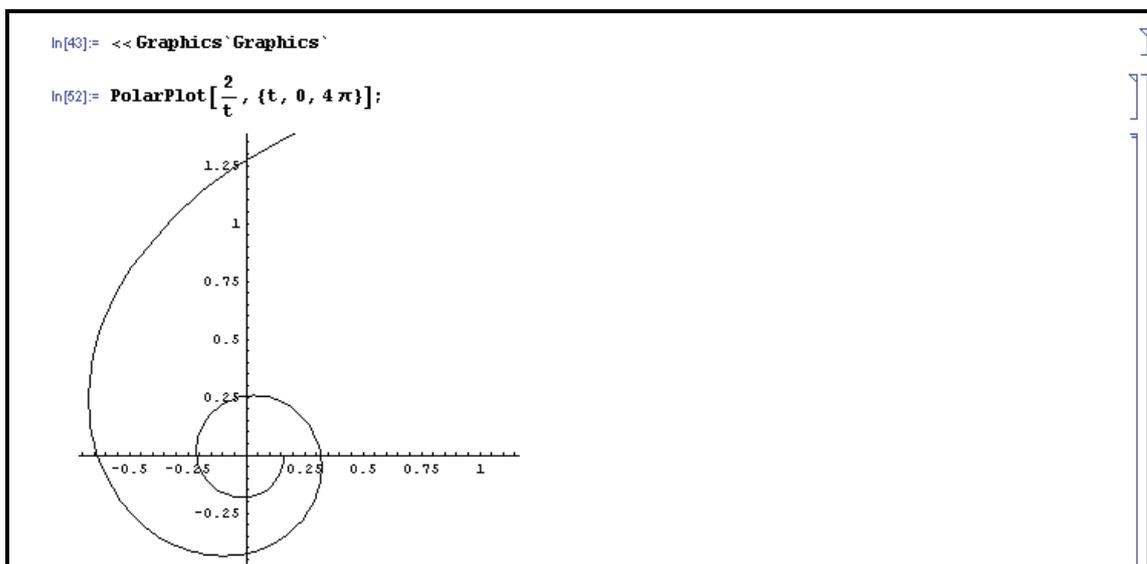
La librería **Graphics** contiene las funciones necesarias para representar funciones expresadas en polares. Para cargar esta librería debemos escribir:

```
<<Graphics`Graphics`
```

Esta librería contiene, entre otras, la función **PolarPlot**, cuya sintaxis es:

```
PolarPlot[f,{t,a,b}]
```

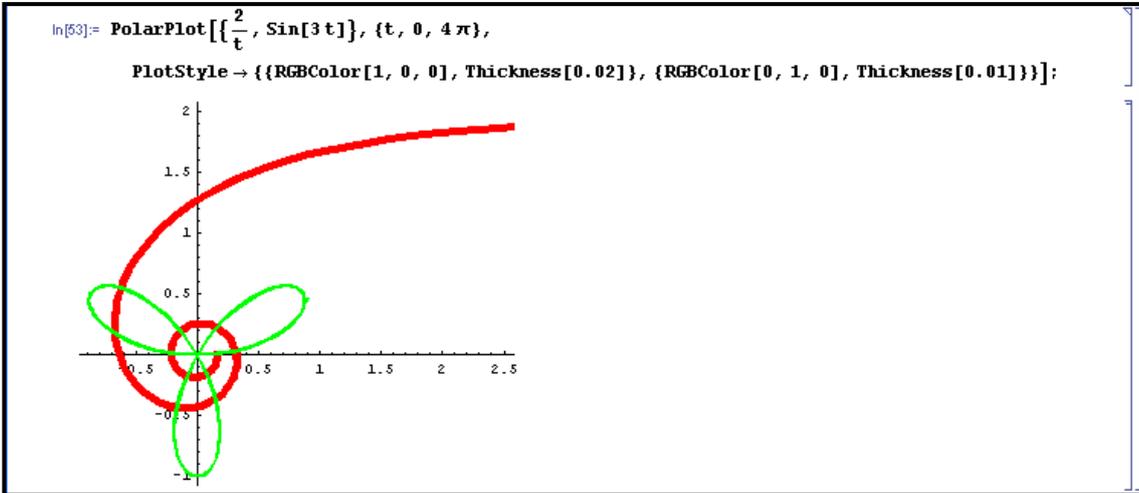
que nos representa la función f para los valores del argumento t comprendidos entre los ángulos a y b expresados en radianes.



Utilizando la notación:

PolarPlot[{f,g,...},{t,a,b}, opcion->valor]

se obtiene la representación simultánea de varias funciones expresadas en forma polar e incluye las distintas opciones para cambiar el aspecto del gráfico final,



8.- REPRESENTACIÓN DE FUNCIONES EXPRESADAS EN FORMA IMPLÍCITA

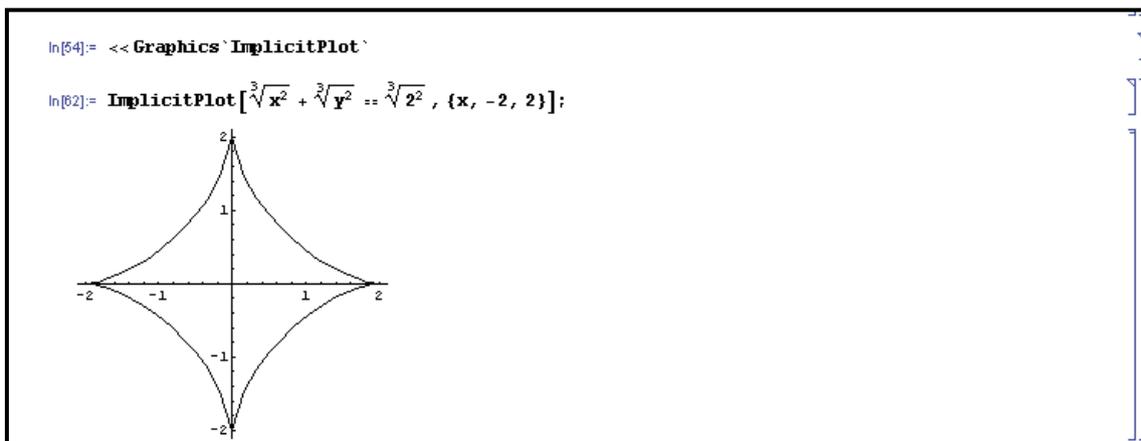
Las instrucciones que permiten representar una función dada en forma implícita se encuentran en la librería **ImplicitPlot**.

Esta librería se carga escribiendo:

<<Graphics`ImplicitPlot`

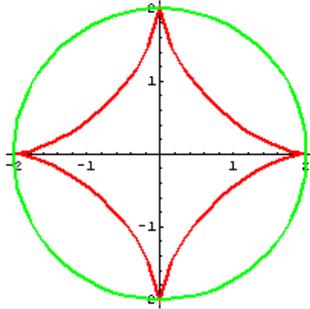
Esta librería contiene la función **ImplicitPlot**, cuya sintaxis es:

ImplicitPlot[expresión,{x,a,b}]



La representación simultánea de varias curvas se realiza de forma idéntica al resto de comandos estudiados anteriormente. Además se pueden incluir distintas opciones para modificar el aspecto de la presentación.

```
In[84]:= ImplicitPlot[{ $\sqrt[3]{x^2} + \sqrt[3]{y^2} == \sqrt[3]{2^2}$ ,  $x^2 + y^2 == 4$ }, {x, -2, 2},
  PlotStyle -> {{RGBColor[1, 0, 0], Thickness[0.01]}, {RGBColor[0, 1, 0], Thickness[0.01]}}];
```



```
In[87]:= ImplicitPlot[Table[ $2x^2 + 3y^2 == r$ , {r, 1, 20, 5}], {x, -5, 5},
  PlotStyle -> {RGBColor[1, 0, 0], Thickness[0.01]};
```

