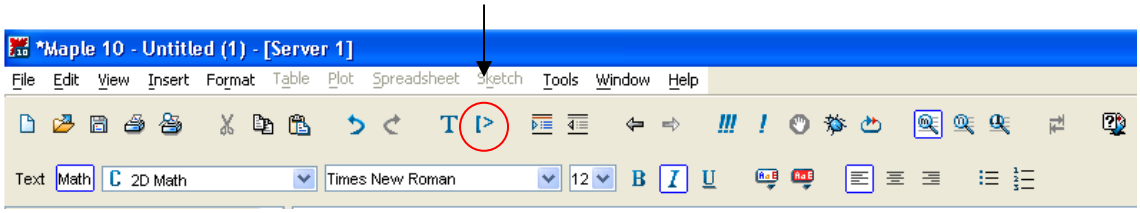
Sistemas lineales con utilización de software

Para comenzar a utilizar el programa y escribir cualquier sentencia, primero hacer clic sobre la hoja de trabajo y luego clic en el ícono [>] que se encuentra en la barra de herramientas en la parte superior de la pantalla.



Resolución de sistemas lineales de n ecuaciones en m variables

El comando que permite resolver sistemas de m ecuaciones con n variables es ***solve*** :

```
> solve({ec1,ec2,ec3,...,ecm},{x1,x2,x3,...,xn});
```

donde $ec_1, ec_2, ec_3, ..., ec_m$ son las ecuaciones del sistema y $x_1, x_2, x_3, ..., x_n$ son las incógnitas.

Para terminar la entrada de una orden hay que escribir el signo ":" (dos puntos) o ";" (punto y coma) antes de pulsar enter. En el primer caso se ejecuta la orden, pero no aparece respuesta en pantalla, en el segundo caso la respuesta se visualiza en la pantalla. A partir de la versión 10, no es necesario el ";" final.

Ejemplo 1:

$$\begin{cases} x + y = 2 \\ -x + y = 0 \end{cases}$$

```
> solve({x+y=2,-x+y=0},{x,y});
```

$$\{y = 1, x = 1\}$$

Representa en el plano a dos rectas que se cortan en el punto (1,1).

Ejemplo 2:

```
> solve({x+y=2,2*x+2*y=4},{x,y});
```

$$\{y = y, x = -y + 2\}$$

Dos rectas coincidentes en el plano.

Ejemplo 3:

```
> solve({x+y=2,x+y=1},{x,y});
```

el software no muestra ninguna respuesta ya que se trata de dos rectas paralelas en el plano.

El comando que permite graficar es ***with(plots)***; sólo se necesita cargarlo una vez.

Grafiquemos las situaciones anteriores:

```
> with(plots):
> r1:=implicitplot(x+y=2,x=-4..4,y=-4..4,color=blue):
> r2:=implicitplot(-x+y=0,x=-4..4,y=-4..4,color=red,
title="Rectas en el plano"):
> display(r1,r2);
```

Se requiere del comando ***implicitplot.***

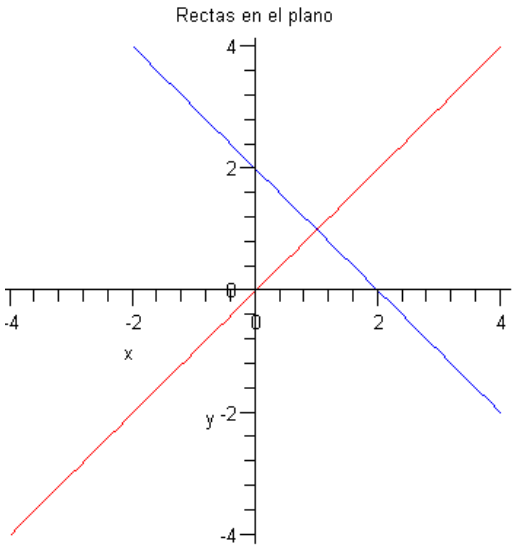
Con r1:= y r2:= se le asigna un nombre a cada recta.

Con x =…, y =… se le asigna un rango a las variables.

Color = para que se represente a cada recta con un color diferente.

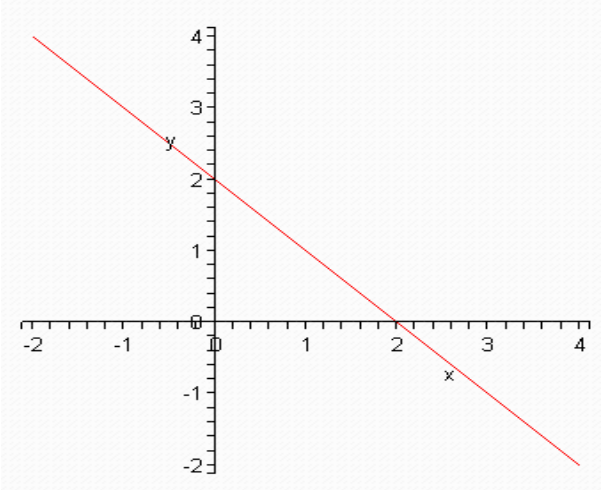
Title = "…" para que la representación se muestre con un título.

Display(…) para representar en un mismo sistema de coordenadas. Notar que al final de cada sentencia se escribe ":" en vez de ;

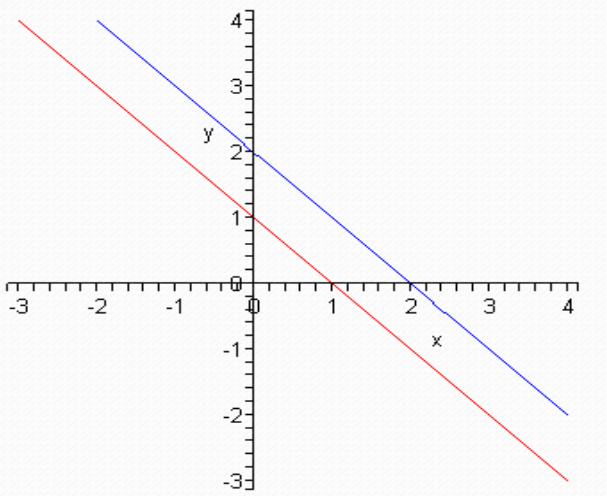


Precaución: si se copian sentencias de Word a Maple, hay que reescribir la comillas simples y dobles porque el software no las lee como tales.

```
> r1:=implicitplot(x+y=2,x=-4..4,y=-4..4,color=blue):
> r2:=implicitplot(2*x+2*y=4,x=-4..4,y=-4..4,color=red):
> display(r1,r2);
```




```
> r1:=implicitplot(x+y=2,x=-4..4,y=-4..4,color=blue):
> r2:=implicitplot(x+y=1,x=-4..4,y=-4..4,color=red):
> display(r1,r2);
```



Gráficos en en el plano

Podemos ver que si hacemos click con el cursor sobre la gráfica, aparecen inmediatamente en la última línea de la barra de menú, algunos comandos representados por los íconos que se muestran a continuación, que provocan que *Maple* ejecute alguna acción directa sobre la gráfica seleccionada:

Cambia de **Constrained** (forzado, contraído) a **Unconstrained**.

Gráfica en línea continua; **style line** (línea).

Gráfica en línea de puntos; **style point** (punto).

Gráfica con una malla; **style=patch**.

Gráfica sin malla; **style=patchnogrid.**

Cambia los ejes al estilo cuadrado.

Cambia los ejes al estilo Frame (Marco).

Cambia los ejes al estilo Normal.

Permite ver las coordenadas de un punto

Agrandar o achicar el gráfico

Mover el gráfico de forma estática

Gráfica sobre una cuadrícula

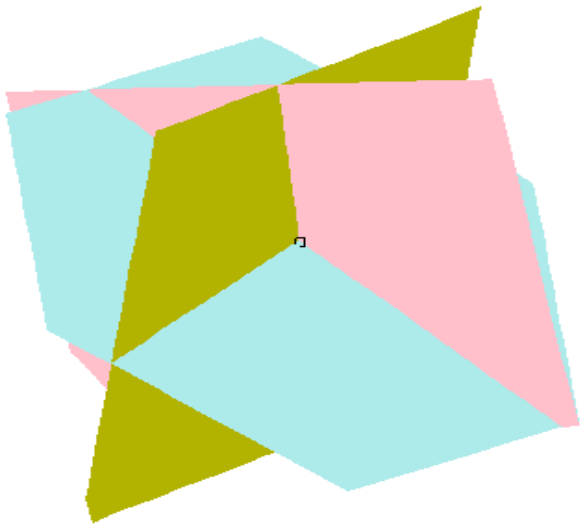
Ejemplo 4:

```
> solve({-x+y-z=1,-2*x+y+3*z=10,3*x+y+2*z=3},{x,y,z});
```

$$\{x = -1, z = 2, y = 2\}$$

Representa en el espacio tres planos que se cortan en el punto (-1,2,2).

```
> r1:=implicitplot3d(-x+y-z=1,x=-20..20,y=-20..20,z=-
20..20,color=turquoise):
> r2:=implicitplot3d(-2*x+y+3*z=10,x=-20..20,y=-20..20,z=-
20..20,color=pink):
> r3:=implicitplot3d(3*x+y+2*z=3,x=-20..20,y=-20..20,z=-
20..20,color=yellow):
> r4:=pointplot3d([-1,2,2],symbol=box,color=black,
symbolsize=10):
> display(r1,r2,r3,r4);
```



Se requiere del comando ***pointplot3d*** si se desea representar un punto.

Symbol= para definir el estilo de representación del punto.

Symbolsize= indica el tamaño del punto.

Ejemplo 5:

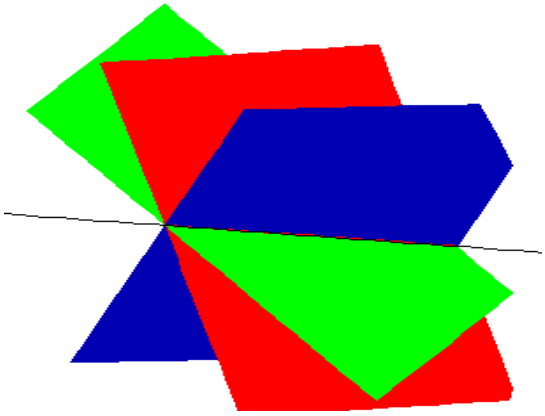
```
> solve({x+2*y-z=4,2*x+5*y+2*z=9,x+4*y+7*z=6},{x,y,z});
```

$$\{y = 1 - 4\,z, x = 2 + 9\,z, z = z\}$$

Representa en el espacio tres planos que se cortan en una recta.

```
> with(plots):
> r1:=implicitplot3d(x+2*y-z=4,x=-8..8,y=-8..8,z=-8..8,
color=blue):
> r2:=implicitplot3d(2*x+5*y+2*z=9,x=-8..8,y=-8..8,z=-8..8,
color=red):
> r3:=implicitplot3d(x+4*y+7*z=6,x=-8..8,y=-8..8,z=-8..8,
color=green):
> r4:=spacecurve([2+9*t,1-4*t,0+t],t=-2..2,color=black):
> display(r1,r2,r3,r4);
```

Se requiere del comando ***spacecurve*** si se desea representar una recta en el espacio dada en forma paramétrica.

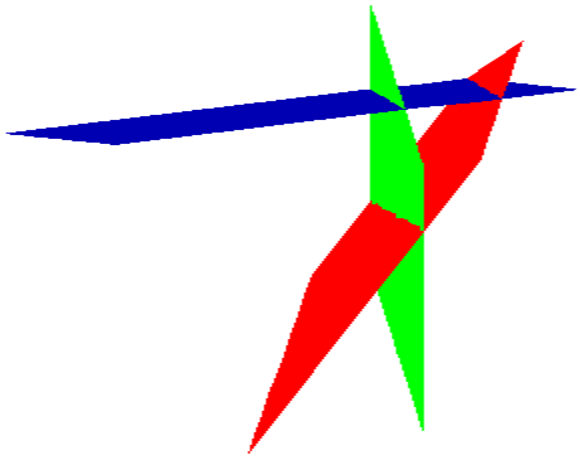


Ejemplo 6:

```
> solve({y-2*z=-5,2*x-y+z=-2,4*x-y=-4},{x,y,z});
```

Representa en el espacio tres planos con intersección vacía.

```
> r1:=implicitplot3d(y-2*z=-5,x=-3..3,y=-3..3,z=-3..3,
color=blue):
> r2:=implicitplot3d(2*x-y+z=-2,x=-3..3,y=-3..3,z=-3..3,
color=red):
> r3:=implicitplot3d(4*x-y=-4,x=-3..3,y=-3..3,z=-3..3,
color=green):
> display(r1,r2,r3);
```

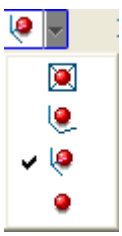


Gráficos en en el espacio

Podemos ver que si hacemos clic con el cursor sobre la gráfica, aparecen inmediatamente en la barra de menú, algunos comandos representados por los íconos que se muestran a continuación, que provocan que Maple ejecute alguna acción directa sobre la gráfica seleccionada:

Cambia el ángulo de rotación del sistema de referencia horizontalmente y verticalmente.

Cambia la gráfica al estilo **Patchnogrid (Sin rejilla).**
Cambia la gráfica al estilo **Patch (Rejilla)**.
Cambia la curva al estilo **Patchcontour.**
Cambia la curva al estilo **Contour (Contorno).**
Cambia la curva al estilo **Line (Línea).**
Cambia la curva al estilo **Hidden.**
Cambia la curva al estilo **Point (Punto).**



Cambia los ejes al estilo **Boxed (Caja).**
Cambia los ejes al estilo **Frame (Marco)**.
Cambia los ejes al estilo **Normal**.
Cambia los ejes al estilo **None (Ninguno)**.

Mover el gráfico libremente
Agrandar o achicar el gráfico
Mover el gráfico de forma

Eliminación de Gauss

Las matrices pueden definirse de formas distintas a través del comando **matrix**.

- Indicando en los dos primeros números la cantidad de filas y columnas. Seguidamente se introducen los elementos de la matriz de la siguiente forma:

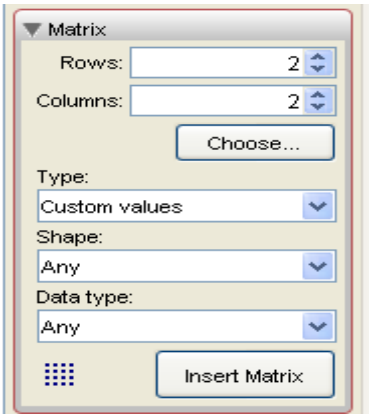
```
>A:=matrix(2,3,[1,2,3,4,5,6]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- Ingresando los elementos de cada fila:

```
> A:=matrix([[1,2,3],[4,5,6]]);
```

- O usando la opción del menú en parte izquierda de la pantalla. Primero seleccionar el tamaño de la matriz y luego ingresar los elementos.



$$> \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{bmatrix}$$

Para obtener una forma escalón o la forma escalón reducida de una matriz, se utilizan comandos que se encuentran dentro del paquete **linalg.**

```
> with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected
```

El comando **gausselim** permite obtener una forma de escalón.

El comando **rref** o **gaussjord** permite obtener la forma escalón reducida.

Ejemplo:

```
> with(linalg):
```

```
> C := matrix([[1, 0, 1, 1, -5], [1, 0, -1, 1, -1], [1, 1, 1, 1, -3], [2, 0, 2, 0, -2]]);
```

$$C := \begin{bmatrix} 1 & 0 & 1 & 1 & -5 \\ 1 & 0 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -3 \\ 2 & 0 & 2 & 0 & -2 \end{bmatrix}$$

```
> gausselim(C);
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 & -5 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & -2 & 0 & 4 \\ 0 & 0 & 0 & -2 & 8 \end{bmatrix}$$

```
> rref(C);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix}$$

Advertencia: En sistemas donde alguna ecuación tiene como coeficiente un valor desconocido, aplicar el comando ***rref*** puede acarrear errores ya que el software **no** tiene en cuenta que éste puede ser cero.

Para sistemas con coeficientes complejos escribir I en lugar de i.

Ejemplo: (Ejercicio 4 de la práctica adicional de sistemas de ecuaciones lineales).

Analizar la compatibilidad del sistema según los valores de a y b.

$$\begin{cases} 2x + y + 2z = 1 \\ x + y + z = a \\ 2x + 3y + b\,z = -1 \end{cases}$$

Ingresemos la matriz ampliada del sistema:

```
> C := matrix([[2, 1, 2, 1], [1, 1, 1, a], [2, 3, b, -1]]);
```

$$C := \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & a \\ 2 & 3 & b & -1 \end{bmatrix}$$

Utilizando el comando ***rref*** obtenemos:

```
> with(linalg):
```

```
> rref(C)
```

$$\begin{bmatrix} 1 & 0 & 0 & -\dfrac{-b+2-6a+ab}{b-2} \\ 0 & 1 & 0 & -1+2a \\ 0 & 0 & 1 & -\dfrac{4a}{b-2} \end{bmatrix}$$

De lo que concluiríamos una respuesta equivocada.

Si utilizamos el comando ***gausselim*** obtenemos:

```
> gausselim(C)
```

$$\begin{bmatrix} 2 & 1 & 2 & 1 \\ 0 & 2 & b-2 & -2 \\ 0 & 0 & \dfrac{1}{2}-\dfrac{1}{4}b & a \end{bmatrix}$$

De lo que concluimos que el sistema tiene solución única si $b \neq 2$ y $\forall a$,

Tiene infinitas soluciones si $b = 2$ y $a = 0$ y resulta incompatible si $b = 2$ y $a \neq 0$.

Vectores

Existen distintas formas de definir vectores:

```
> u:=<1,2,3>;
```

$$u := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> with(LinearAlgebra):v:=Vector([1,2,3]);
```

$$v := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> Vector[row]([1,2,3]);
```

(vector fila)

[1, 2, 3]

```
> Vector[column]([1,2,3]);
```
(vector columna)

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

```
> u:=([1,2,3]);
```

$u := [1, 2, 3]$

O como una matriz:

```
> u:=[1 2 3];
```

```
>
```
$$u := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Operaciones con matrices (tener en cuenta que según el tamaño de las matrices a operar puede resultar que alguna operación no se pueda realizar)

```
>
```
$$A := \begin{bmatrix} 1 & 0 \\ -1 & 3 \end{bmatrix} : B := \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} :$$

Suma de matrices:

```
> A+B
```

$$\begin{bmatrix} 2 & -1 \\ 0 & 5 \end{bmatrix}$$

Resta de matrices:

```
> A-B
```

$$\begin{bmatrix} 0 & 1 \\ -2 & 1 \end{bmatrix}$$

Multiplicación de un escalar por una matriz: (la operación correspondiente es * asterisco)

```
> 3*A
```

$$\begin{bmatrix} 3 & 0 \\ -3 & 9 \end{bmatrix}$$

Multiplicación de matrices: (la operación correspondiente es el punto)

```
> A.B
```

$$\begin{bmatrix} 1 & -1 \\ 2 & 7 \end{bmatrix}$$

O pueden utilizarse los comandos:

```
> with(linalg):multiply(A,B);
```

```
> u := [ 2 ; -2 ]:
```

```
> A.u
```

$$\begin{bmatrix} 2 \\ -8 \end{bmatrix}$$

Matriz transpuesta

Se utiliza el comando ***transpose.***

Ejemplo:

```
> E := [ 1  0  2 ; -1  3  -2 ]:
```

```
> transpose(E);
```

$$\begin{bmatrix} 1 & -1 \\ 0 & 3 \\ 2 & -2 \end{bmatrix}$$

Matriz inversa

Para obtener la matriz inversa, se utiliza el comando ***inverse*** que se encuentran dentro del paquete ***linalg.***

Ejemplo:

```
>with(linalg):
```

```
> A := [ 1  2  1 ; 1  2  -1 ; 1  -2  1 ]:
```

```
> with(linalg);
```

> $B := inverse(A);$

$$B := \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & 0 & \frac{-1}{4} \\ \frac{1}{2} & \frac{-1}{2} & 0 \end{bmatrix}$$

O también se puede utilizar: $evalm(A\,\hat{}\,(-1));$

Verificación:

> $multiply(A, B);$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Determinante de una matriz

Se utiliza el comando ***det*** que se encuentran dentro del paquete ***linalg*.**

Ejemplo:

> $B := \begin{bmatrix} 1 & 3 \\ -2 & 4 \end{bmatrix}:$

```
> with(linalg):
> det(B);
```

10