PRÁCTICA Nº 7

Matrices y determinantes. Sistemas de ecuaciones lineales

En esta práctica veremos cuestiones generales sobre matrices, las operaciones definidas y cómo los determinantes nos proporcionan una herramienta que mejora distintos cálculos. En concreto, haremos un resumen de los distintos métodos que hemos visto para obtener el rango de una matriz y el estudio de matrices regulares y el cálculo de sus inversas. La utilización de determinantes tendrá también una aplicación importante a la resolución de sistemas de ecuaciones lineales. Este es uno de los problemas que se encuentra con más frecuencia en los distintos campos de la Ciencia. En particular, nosotros tendremos que resolver sistemas de ecuaciones en bastantes ocasiones, como pasos previos a la resolución de distintos problemas. En primer lugar veremos los comandos que Mathematica incorpora para discutir y resolver sistemas de ecuaciones, y posteriormente, otras alternativas a este estudio, calculando el sistema escalonado reducido o la fórmula de Cramer.

1. GENERALIDADES SOBRE MATRICES.

Una matriz de dimensión $m \times n$ puede ser considerada como una lista formada por un conjunto de listas de la misma longitud siendo m el número de listas que la componen y n la longitud de las mismas. Por ejemplo, la lista:

In[]:= **a={{1,2,3},{2,3,4},{3,4,6}};**

puede considerarse que representa a la tabla o matriz:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{pmatrix}$$

Si se desea que Mathematica la represente en forma tabular, escribiremos:

MatrixForm[a]

La orden **Dimensions[nombre]** genera una lista formada por un único número igual a su longitud, si “nombre” es una matriz fila o columna, o por dos números en caso contrario que representan el número de filas y el de columnas, respectivamente.

In[]:= **v={-1,2,3};**

In[]:= **Dimensions[v]**

Out[]= `{3}`

In[]:= **Dimensions[a]**

Out[]= `{3,3}`

Si se desea seleccionar una fila de una matriz, bastará con escribir el nombre de la misma y, entre dobles corchetes, la posición que ocupa dicho elemento. La *i*-ésima componente de *v* se selecciona mediante **v[[i]]**, y el elemento que ocupa la posición (i, j) de la matriz a mediante **a[[i,j]]**.

Las órdenes

In[]:= **m=Dimensions[a][[1]]**

In[]:= **n=Dimensions[a][[2]]**

asignan a *m* y *n* el número de filas, 3 y el de columnas, 3 de la lista *a*, pues Dimensions[a] es la lista formada por estos dos valores.

Las siguientes funciones permiten construir algunos tipos especiales de matrices:

a) **DiagonalMatrix[{x,y,z,...}]**

Genera la matriz diagonal cuyos elementos de la diagonal son sus argumentos.

In[]:= **DiagonalMatrix[{1,-2,4}]**

Out[]= `{{1, 0, 0}, {0, -2, 0}, {0, 0, 4}}`

b) **IdentityMatrix[n]**

Matriz identidad de orden *n*.

In[]:= **IdentityMatrix[4]**

Out[]=
```
{{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0},
 {0, 0, 0, 1}}
```

2. OPERACIONES CON MATRICES.

La suma y el producto de matrices con las correspondientes condiciones sobre sus dimensiones, se expresan mediante los operadores "+" y "." respectivamente. Consideremos las siguientes matrices:

In[]:= **a=Table[1/(i+j),{i,5},{j,5}];**
b={{2,3,4,5,6},{3,4,5,6,7},{4,5,6,7,8},{5,6,7,8,9},{6,7,8,9,10}};

MatrixForm[a]
MatrixForm[b]

Out[]= $\begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \end{pmatrix}$ $\begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \\ 6 & 7 & 8 & 9 & 10 \end{pmatrix}$

Algunos ejemplos de operaciones son:

a) *In[]:=* **a+b//MatrixForm**

Out[]= $\begin{pmatrix} \frac{5}{2} & \frac{10}{3} & \frac{17}{4} & \frac{26}{5} & \frac{37}{6} \\ \frac{10}{3} & \frac{17}{4} & \frac{26}{5} & \frac{37}{6} & \frac{50}{7} \\ \frac{17}{4} & \frac{26}{5} & \frac{37}{6} & \frac{50}{7} & \frac{65}{8} \\ \frac{26}{5} & \frac{37}{6} & \frac{50}{7} & \frac{65}{8} & \frac{82}{9} \\ \frac{37}{6} & \frac{50}{7} & \frac{65}{8} & \frac{82}{9} & \frac{101}{10} \end{pmatrix}$

b) *In[]:=* **a b**

Out[]= `{{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1}}`

c) *In[]:=* **a*b**

Out[]= `{{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1},{1,1,1,1,1}}`

d) *In[]:=* **a.b//MatrixForm**

Out[]= $\begin{pmatrix} 5 & \frac{129}{20} & \frac{79}{10} & \frac{187}{20} & \frac{54}{5} \\ \frac{547}{140} & 5 & \frac{853}{140} & \frac{503}{70} & \frac{1159}{140} \\ \frac{1357}{420} & \frac{3457}{840} & 5 & \frac{4943}{840} & \frac{2843}{420} \\ \frac{2321}{840} & \frac{4421}{1260} & \frac{10721}{2520} & 5 & \frac{14479}{2520} \\ \frac{1523}{630} & \frac{2573}{840} & \frac{4673}{1260} & \frac{10973}{2520} & 5 \end{pmatrix}$

Las potencias de una matriz las calcularemos a través de la función **MatrixPower[matriz, n]**, quien realiza la potencia n-ésima de la matriz "matriz".

In[]:= **M={{2,3,4},{3,4,5},{4,5,6}}**
MatrixPower[M,4]

Out[]= `{{4493, 5916, 7338}, {5916, 7788, 9660}, {7338, 9660, 11982}}`

sin embargo de la forma siguiente no calculamos las potencias

In[]:= **M^4**

Out[]= `{{16, 81, 256}, {81, 256, 625}, {256, 625, 1296}}`

El producto de un escalar por un vector o una matriz se expresa mediante el operador "*" o un espacio:

a) *In[]:=* **0 a**

Out[]= `{{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}, {0,0,0,0,0}}`

b) *In[]:=* **0*a**

Out[]= `{{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}, {0,0,0,0,0}}`

La transpuesta de una matriz la calculamos a partir de la función **Transpose[matriz]** quien nos devuelve la transpuesta de "matriz".

In[]:= **M2={{1,0},{3,2},{4,-1}};**
Transpose[M2]

Out[]= `{{1, 3, 4}, {0, 2, -1}}`

Para matrices cuadradas disponemos de la función **Det[matriz]** que calcula el determinante de "matriz". Por ejemplo:

In[]:= **M1={{1,2,3},{2,3,4},{3,4,6}};**
Det[M1]

Out[]= `-1`

3. CALCULO DEL RANGO DE UNA MATRIZ.

Dada una matriz $A \in M_{n\times m}(K)$, nos proponemos calcular el rango de A. Como sabemos el **rango** de A, **rg(A),** es el número de filas no nulas de su forma normal de Hermite por filas (= número de pivotes). Además si A tiene n filas y m columnas sabemos que rg(A) $\leq$ mín{m, n}.

In[]:= **A={{1,2,3,4,1},{2,2,1,1,1},{3,4,4,5,2},{1,0,3,2,0}};**
RowReduce[A]//MatrixForm

Out[]=
$$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
Por tanto el rango de A es 3.
```

Como es conocido, los determinantes nos proporcionan un nuevo método para calcular el rango de una matriz.

Teorema: Sea A $\in M_{m\times n}(K)$, entonces el rango de A coincide con el mayor orden de una submatriz cuadrada regular de A.

Este teorema nos proporciona el método para calcular el rango de A. Tomamos k = mín{m, n}, luego rg(A) $\leq$ k, tomadas todas las submatrices cuadradas de A de orden k, si alguna es regular (su determinante es no nulo), entonces rg(A) = k. En otro caso, rg(A) $\leq$ k-1 y se repite el proceso con k-1 y así sucesivamente. Buscamos así menores (o determinantes de submatrices) no nulos de orden máximo.

Esto lo haremos con el Mathematica usando la orden **Minors[matriz,k]** que nos genera una matriz de orden $\binom{p}{k}\times\binom{q}{k}$ cuyos elementos son los menores de orden k de "matriz" siendo p y q el número de filas y columnas respectivamente, de "matriz".

In[]:= **M2={{1,0},{3,2},{4,-1}};**
Minors[M2,2]

Out[]= `{{2}, {-1}, {-11}}`

Así, comenzaremos calculando los menores de orden k = min{m, n} y si todos son ceros reduciremos k hasta encontrar algún menor distinto de cero. Por ejemplo:

In[]:= **A={{1,2,3,4,1},{2,2,1,1,1},{3,4,4,5,2},{1,0,3,2,0}};**
n=Dimensions[A][[1]];
m=Dimensions[A][[2]];
k =Min[n,m];
Minors[A,k]

Out[]= `{{0,0,0,0,0}}`

In[]:= **Minors[A, k-1]**

Out[]=
```
{{0,0,0,0,0,0,0,0,0,0},
 {-10,-10,0,10,5,5,10,0,0,5},
 {-10,-10,0,10,5,5,10,0,0,5},
 {10,10,0,-10,-5,-5,-10,0,0,5}}
```

Como ya hemos encontrado menores de orden k-1 distintos de cero se tiene que el rango de A es k-1:

In[]:= **k-1**

Out[]= 3

Vamos ahora a usar la orden Module para crear un nuevo comando que llamaremos **rango** y que nos calculará el rango de una matriz usando el cálculo de menores como en lo anterior:

In[]:=

```
rango[A_]:=Module[{n,m,s},
 n=Dimensions[A][[1]];
 m=Dimensions[A][[2]];
 k=Min[n,m];
 If[n==m  && Det[A]!=0, Print[" El rango de esta matriz es ", k ],
 For[i=k, i>=1,i--,s=Minors[A,i];
If[s!=Table[0,{j1,Dimensions[s][[1]]},{j2,Dimensions[s][[2]]}],
Print[i];Break[]]
]]]
```

In[]:=

```
A={{1,2,3,4,1},{2,2,1,1,1},{3,4,4,5,2},{1,0,2,3,0}};
rango[A]
```

Out[]= 3

Mathematica dispone de la función MatrixRank[matriz] para calcular el rango. Veamos:

In[]:=

```
A={{1,2,3,4,1},{2,2,1,1,1},{3,4,4,5,2},{1,0,2,3,0}};
MatrixRank[A]
```

Out[]= 3

4. MATRICES REGULARES. INVERSE[A].

Como sabemos no todas las matrices tienen inversa. Llamamos matriz regular a toda matriz cuadrada que admite inversa; es decir, es aquella que es una unidad dentro del anillo de las matrices cuadradas. Así, para que una matriz sea regular es necesario que sea cuadrada y de rango máximo, con Mathematica ambas cosas se pueden comprobar con facilidad. En primer lugar, si fuese necesario, usaremos la orden

Dimensions[A]

para calcular el número de filas y columnas que tiene nuestra matriz, la salida será un vector de dos coordenadas que indican el número de filas y de columnas por tanto si las dos coordenadas coinciden la matriz A será cuadrada, además usando la orden

Det[A]

calcularemos el determinante de la matriz A y si es distinto de cero la matriz A es regular y podremos calcular su inversa con la orden

Inverse[A]

Por último para comprobar que la matriz obtenida es realmente la inversa de A basta con multiplicar A y la matriz obtenida y ver que el producto es la identidad. Veamos el siguiente ejemplo:

In[]: = **A={{1,2,-1},{0,-2,1},{2,1,-1}};**
Dimensions[A]

Out[] = `{3, 3}`

In[]: = **Det[A]**

Out[] = `1`

In[]:= **Inverse[A]**

Out[]= `{{1,1,0},{2,1,-1},{4,3,-2}}`

In[]:= **Inverse[A].A==IdentityMatrix[3]**

Out[]= `True`

Otro método que vimos para el cálculo de la inversa fue el que utilizábamos transformaciones elementales: Teniendo en cuenta que para toda matriz regular, su forma de Hermite por filas es la matriz identidad, para calcular la inversa de A podemos seguir un procedimiento análogo al de la práctica de matrices elementales. Para ello tomaremos la matriz (A/I) resultante de pegar la matriz A y la matriz identidad y aplicando transformaciones elementales por filas a dicha matriz, obtendremos a la izquierda, la forma de Hermite por filas H, que será la identidad, y a la derecha Q, una matriz que verifica la siguiente ecuación: Q.A = H = Id, por tanto, Q = A^{-1}. También podríamos hacer:

In[]: = **A={{1,2,-1},{0,-2,1},{2,1,-1}};**
n=Dimensions[A][[1]];
B= Transpose[Join[A,IdentityMatrix[n]]];
Bt=Transpose[RowReduce[B]];
MatrixForm[Table[Bt[[i]],{i,n+1,2n}]]

Out[]=

$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 4 & 3 & -2 \end{pmatrix}$$

Los determinantes nos proporcionan también un nuevo método para calcular la inversa de una matriz A, que como sabemos es

$$A^{-1} = \frac{\mathrm{Adj}(A)^t}{\det(A)}$$

donde $\mathrm{Adj}(A) = (\alpha_{ij})_{i,j}$ es la **matriz adjunta**, esto es la matriz que en cada posición tiene el correspondiente menor adjunto de A, siendo el **ij-ésimo menor adjunto** de A

$$\alpha_{ij} = (-1)^{i+j} \det(A_{ij})$$

donde A_{ij} es la matriz que se obtiene de A eliminando la fila i-ésima y la columna j-ésima.

Mathematica no tiene definida la matriz adjunta de A, por lo que previamente tendremos que definirla y para ello usaremos la orden **Minors[A, n]** que calcular los menores de A de orden n y los expresa mediante una matriz. Veámoslo con un ejemplo:

In[]:= **A={{1,2,-1},{0,-2,1},{2,1,-1}};**

Calculamos ahora los menores de orden 2, mediante el comando Minors:

In[]:= **S = Minors[A, 2];**
MatrixForm[S]

Out[]=
$$\begin{pmatrix} -2 & 1 & 0 \\ -3 & 1 & -1 \\ 4 & -2 & 1 \end{pmatrix}$$

Para calcular la matriz adjunta tenemos que darnos cuenta que Minors nos da como salida los menores de orden 2 de forma que el menor que resulta de quitar la fila 1 y la columna 1 lo coloca en el lugar s_{33} y así sucesivamente, por tanto la matriz adjunta de A se puede definir mediante:

In[]:= **adj=Table[(-1)^(i+j)*S[[4-i, 4-j]],{i,1,3},{j,1,3}];**
MatrixForm[adj]

Out[]=
$$\begin{pmatrix} 1 & 2 & 4 \\ 1 & 1 & 3 \\ 0 & -1 & -2 \end{pmatrix}$$

Aplicando ahora la fórmula, es decir, adjunta transpuesta partido por el determinante obtenemos la inversa de A:

In[]:= **inv =Transpose[adj]/Det[A];**
MatrixForm[inv]

Out[]=
$$\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ 4 & 3 & -2 \end{pmatrix}$$

5. RESOLUCIÓN DE SISTEMAS DE ECUACIONES.

La resolución de sistemas de ecuaciones lineales en Mathematica puede hacerse de dos maneras diferentes:

(a) Mediante la función LinearSolve[], cuya sintaxis es:

LinearSolve[A, b]

donde A es la matriz de coeficientes y b el vector de términos independientes y nos devolverá un vector que cumple la ecuación matricial $A \cdot x = b$.

In[]:= **a={{1,2,3},{2,3,4},{1,0,3}};**
b={1,0,5};
LinearSolve[a, b]

Out[]: = `{-1,-2,2}`

Otra forma de resolver el sistema anterior es invirtiendo la matriz de coeficientes como se muestra en el siguiente ejemplo:

In[]:= **Inverse[a].b**

Out[]:= `{-1,-2,2}`

Destacar que en sistemas compatibles indeterminados, este comando sólo calcula una de sus soluciones. Para reconocer si un sistema es determinado o no podemos usar el comando NullSpace que determina si el correspondiente sistema homogéneo tiene solución no nula:

NullSpace[A]

donde A es una matriz, nos devuelve una base del las soluciones del sistema $A \cdot x = 0$. En el ejemplo anterior la solución que nos daban es la única pues el sistema homogéneo tiene como solución el espacio vectorial 0.

In[]:= **NullSpace[a]**

Out[]:= `{}`

Sin embargo en el siguiente ejemplo el sistema es compatible indeterminado y la orden LinearSolve sigue dándonos solo una solución del sistema:

In[]:= **LinearSolve[{{1,1,0},{0,-1,2},{1,0,2}},{1,-1,0}]**

Out[]:= `{0,1,0}`

In[]:= **NullSpace[{{1,1,0},{0,-1,2},{1,0,2}}]**

Out[]:= `{{-2,2,1}}`

(b) Escribiendo las ecuaciones y utilizando para resolverlas el comando Solve.

Mathematica entiende por ecuación una expresión lógica de igualdad,

expresión1==expresión2

y un sistema de ecuaciones por una lista de ecuaciones de la forma:

{expresión1==expresión2, expresión3==expresión4,...}

La sintaxis del comando Solve es:

Solve[{ecuación1,ecuación2,...},{var1,var2,...}]

O bien de forma matricial:

Solve[A.{var1,var2,...}=={b1,b2,...},{var1,var2,...}]

In[]:= **Solve[{x+y==1,-y+2z==-1,x+2z==0},{x,y,z}]**

Out[]:=
```
Solve::svars:
Equations may not give solutions for all "solve"
variables.

{{x->-2 z, y->1+2 z}}
```

Observar que Mathematica advierte que las ecuaciones pueden no dar solución a todas las variables como ocurre aquí con la variable z, que sería la que jugaría el papel de parámetro en la solución del sistema compatible indeterminado.

Otros comandos que incorpora Mathematica para resolver ecuaciones son los comandos Reduce y NSolve[] con sintaxis:

Reduce[{ecuación1,ecuación2,...},{var1,var2,...}]
Nsolve[{ecuación1,ecuación2,...},{var1,var2,...}]

El primero reduce el sistema de ecuaciones a otro más sencillo equivalente al primero, y el segundo resuelve numéricamente el sistema de ecuaciones dando una solución aproximada:

In[]:= **Reduce[{x+y==1,-y+2z==-1,x+2z==0},{x,y,z}]**

Out[]:= `x==-2 z&&y==1+2 z`

In[]:= **NSolve[{2x+y-z==1,x-y+2z==-1,x+y+2z==0},{x,y,z}]**

Out[]:= `x{{x->0.1,y->0.5,z->-0.3}}`

Además con el comando Reduce Mathematica realiza un estudio de casos en función de los parámetros que aparezcan en el sistema:

In[]:= **Reduce[{p*x+y==0,x-y==0},{x,y}]**

Out[]:= `y==x&&p==-1 || 1+p≠ 0&&y==0 && x==0`

Observar por último que Solve[], Reduce[] y NSolve[] resuelven también sistemas de ecuaciones no lineales.

6. SISTEMAS DE ECUACIONES LINEALES ESCALONADOS REDUCIDOS.

Según hemos visto al principio del tema 2, un primer método para la resolución de ecuaciones lineales será el obtener, a través del método de Gauss-Jordan, el sistema escalonado reducido equivalente al sistema de partida. Sin embargo, la siguiente proposición resume el método anterior:

Proposición. Dado un sistema de ecuaciones lineales con matriz ampliada (A|B), si H es la forma normal de Hermite por filas de (A|B), entonces el sistema cuya matriz ampliada es H es un sistema escalonado reducido equivalente al de partida.

■

Según este resultado podemos obtener el sistema escalonado reducido, calculando la forma normal de Hermite por filas de la matriz ampliada. Para ello utilizaremos RowReduce[A|B] o trabajaremos paso a paso usando las matrices elementales de la práctica 5.

In[]: = **a={{1,2,3},{2,3,4},{1,0,3}};**
b={1,0,5};
ampliada={{1,2,3,1},{2,3,4,0},{1,0,3,5}};

In[]: = **RowReduce[ampliada]//MatrixForm**

Out[]: =
$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Como vemos en el ejemplo, el sistema es compatible determinado y su única solución {-1,-2, 2}

7. REGLA DE CRAMER.

Los determinantes nos han proporcionado un método mejorado para la resolución de sistemas de ecuaciones: se trata de aplicar la regla de Cramer.

Dado un sistema de ecuaciones A.X = B donde $A = (a_{ij})_{i,j}$ es la matriz de coeficientes, X y B son las matrices columna de incógnitas y de términos independientes, respectivamente:

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

diremos que es un **sistema de Cramer** si A es cuadrada y regular.

Teorema. (Regla de Cramer). Dado un sistema de Cramer:

$$\begin{cases} a_{11}x_1 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + ... + a_{2n}x_n = b_2 \\ \qquad .... \\ a_{n1}x_1 + ... + a_{nn}x_n = b_n \end{cases}$$

la solución (única) del sistema viene dada por:

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & \cdots & a_{1n} \\ b_2 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_n & a_{n2} & \cdots & a_{nn} \end{vmatrix}}{|A|};...;\ x_n = \frac{\begin{vmatrix} a_{11} & a_{12} & \cdots & b_1 \\ a_{21} & a_{22} & \cdots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & b_n \end{vmatrix}}{|A|}$$

■

Ejemplo. Consideremos el sistema de Cramer:

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

Introducimos la matriz de coeficientes A y el vector de términos independientes:

In[]:= **A={{1,2,-1},{0,-2,1},{2,1,-1}};**
b={1,2,3};

El siguiente paso seria calcular la matriz que se obtiene cambiando la primera columna por el vector de términos independientes y calcular la primera coordenada del vector solución, esto lo hacemos del siguiente modo:

In[]:= **A1=Transpose[A];**
A1[[1]]=b;
x=Table[0,{i,Dimensions[A][[1]]}];
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= 3

De igual forma lo haríamos para la segunda y tercera coordenada:

In[]:= **A1=Transpose[A];**
A1[[2]]=b;
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= 1

In[]:= **A1=Transpose[A];**
A1[[3]]=b;
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= `4`

Vamos ahora a hacerlo todo de una vez usando para ello la orden Module con la que crearemos un nuevo comando **cramer** que nos resolverá los sistemas de Cramer:

In[]:=

```
cramer[A_,b_]:=Module[{A1,x,n},
A1=Transpose[A];
n=Dimensions[A][[1]];
x=Table[0,{i,1,n}];
Do[A1[[i]]=b;
x[[i]] =Det[Transpose[A1]]/Det[A];A1=Transpose[A],{i,1,n}];
Print["La solución del sistema de Cramer es ", x];
]
```

In[]:= **cramer[A,b]**

Out[]= `La solución del sistema es {3,1,4}`

■