Curso Iberoamericano de formación permanente de profesores de matemática

GeoGebra III

Álgebra lineal

GeoGebra III
Álgebra lineal

Contenido de este documento:

Listas y secuencias

Una lista será un conjunto de datos que debemos representar entre llaves, separando por comas sus elementos.

$L_1 = \{-2, 3, 6, 3, 9, 1, 9, -7\}$

$L_2 = \{(0, 3), (-4,1), (1, 3), (2, 0)\}$

Para generar elementos de una lista que cumplan una determinada condición, ya hemos utilizado en ocasiones anteriores el comando **Secuencia**.

Este comando admite la sintaxis siguiente

Secuencia[f(t), t, t_1, t_2]

para crear una lista de elementos utilizando la expresión *f(t)* para valores de *t* comprendidos entre t_1 y t_2.

Además, se podrá incluir un argumento más, para indicar el incremento de valores que tomará la variable *t*.

Por ejemplo, para crear una lista con los cuadrados de los diez primeros números naturales, escribiremos:

Secuencia[n^2,n,1,10]

El resultado, como era de esperar será:

lista1 = {1,4,9,16,25,36,49,64,81,100]

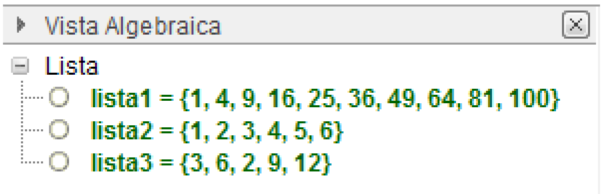
Este comando admite como alternativa la opción:

Secuencia[n]

para crear una lista formada por los números naturales 1, …, n.

Cuando los elementos no sigan una ley de formación, será necesario crear una lista con los elementos, separados por comas y encerrados entre llaves.

En la imagen siguiente aparecen ejemplos con las listas creadas siguiendo los tres modos descritos anteriormente para crearlas.



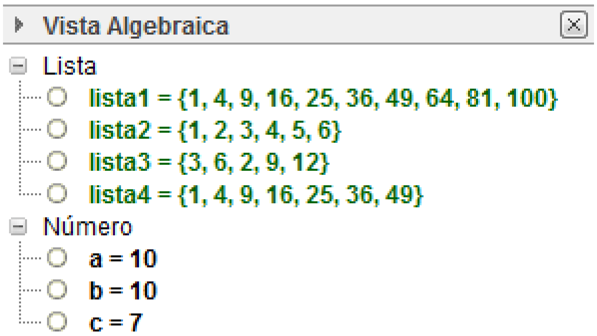
Para trabajar con listas, disponemos de algunos comandos que relacionamos a continuación:

- Longitud de una lista: **Longitud[L]** o **Dimensión[L]**, siendo *L* una estructura de datos del tipo lista. Además, podremos contar el número de elementos de una lista que cumplen una determinada condición, utilizando el comando **CuentaSi[condición, L]**.

  Así, al escribir CuentaSi[x<50,lista1] obtendremos como resultado 7. Este valor aparecerá en la Vista algebraica.

  Si nuestra intención es obtener una nueva lista con los elementos que cumplen una condición, tendremos que utilizar el comando **ConservaSi[condición, L]**.

  Para el ejemplo anterior, al ejecutar el comando ConservaSi[x<50,lista1] obtendremos una nueva lista con los siete elementos que cumplen la condición indicada.



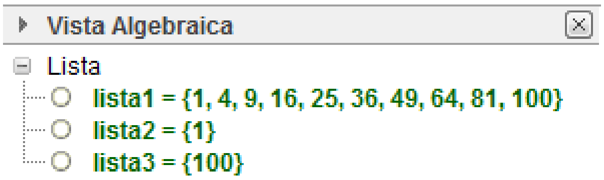
- Máximo de una lista: **Máximo[L]**.

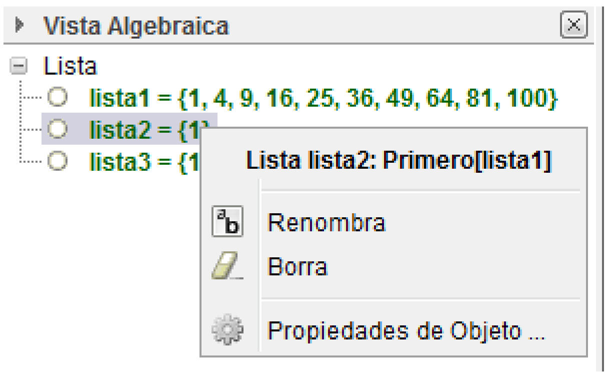
- Mínimo de una lista: **Mínimo[L]**.

- Primer elemento de una lista: **Primero[L]**. Si escribimos **Primero[L, n]** devuelve una lista con los *n* primeros elementos de la lista *L*.

- Último elemento de una lista: **Último[L]**. Al igual que en el comando anterior, **Último[L,n]** devuelve una lista con los *n* últimos elementos de la lista *L*.

  El resultado de los elementos anteriores es una nueva lista.





- Elemento que ocupa la posición *n* en la lista *L*: **Elemento[L, n]**. El resultado no es una lista, es el valor que corresponde al elemento seleccionado.

Si se utiliza **ElementoAleatorio[L]** el resultado será un valor igual a uno de los elementos de la lista elegido de manera aleatoria.

- Extraer elementos de una lista: **Extrae[L, m, n]** devuelve la lista formada por los elementos comprendidos entre las posiciones *m* y *n* de la lista *L*. Cuando se omite el tercer argumento, este comando devuelve una lista con los elementos de la lista original a partir de la posición *m*.

La mayoría de estos comandos se pueden utilizar sobre un texto que puede encontrarse en la Vista gráfica o se puede introducir como argumento encerrado entre comillas.

Además, tenemos otros comandos para realizar operaciones en una lista o para efectuar operaciones con estos datos. Algunos de estos comandos son:

- Ordenar una lista: **Ordena[L]**.

- Invertir el orden de los elementos de una lista: **Invierte[L]**.

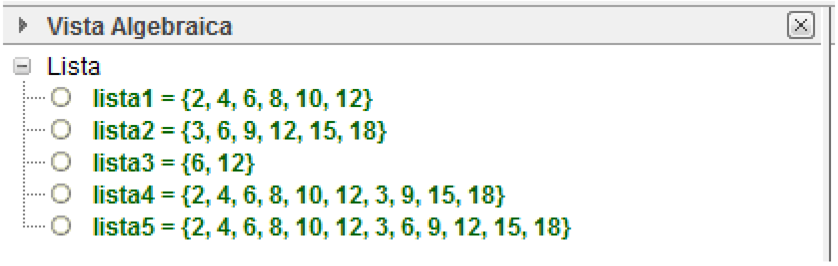
- Añadir elementos a una lista: **Añade[L, elemento]** o **Añade[elemento, L]** para añadir el elemento indicado al final o al principio de la lista *L*.

- Intercalar un elemento en una lista: el comando **Intercala[elemento, L, n]** o **Intercala[L1, L, n]** se utilizará para intercalar el elemento indicado en la posición *n* en la lista *L* o para intercalar la lista *L1* en la lista *L*, a partir de la posición *n*.

- Intersección y unión de listas: **Intersección[L1, L2]** y **Unión[L1, L2]** devuelven la lista intersección y unión, respectivamente, de las listas *L1* y *L2*.

- Encadena: el comando **Encadena[L1,L2]** crea una nueva lista con los elementos de las dos listas, en la que a diferencia de Unión no elimina las repeticiones.

- Eliminar repeticiones: el comando **Único** aplicado sobre una lista devuelve una nueva lista ordenando los elementos en orden creciente, eliminado las repeticiones.



En la imagen anterior, las distintas listas corresponden al resultado de los comandos siguientes:

lista1 = Secuencia[2n,n,1,6]

lista2 = Secuencia[3n,n,1,6]

lista3 = Intersección[lista1,lista2]

lista4 = Unión[lista1,lista2]

lista5 = Encadena[lista1,lista2]

Y aún hay más comandos con los que realizar operaciones en una lista, como son:

- Suma de los elementos de una lista: **Suma[L]**. También admite la sintaxis **Suma[L, n]** para obtener la suma de los *n* primeros elementos de la lista *L*. Este comando admite distintos formatos y posibilidades cuando se combina con el comando **Secuencia**.

  Por ejemplo **Suma[Secuencia[k2, k, 1, 10]]** devuelve el valor de la suma de los cuadrados de los diez primeros números naturales.

- Producto de los elementos de una lista: **Producto[L]**.

- Máximo común divisor y mínimo común múltiplo: se obtiene a partir de los comandos **MCD** y **MCM**, respectivamente.

Ejemplo 1

Genera una lista con los números primos menores que 25.

Para trabajar con números primos disponemos de dos comandos **PrimoAnterior** y **PrimoSiguiente** que devuelven el número primo anterior o siguiente, respectivamente, al valor indicado.

Por ejemplo PrimoAnterior[5] será 3 y PrimoSiguiente[5] devolverá 7.

Por tanto, podemos crear una lista con los siguientes números primos a cada número natural comprendido entre 1 y 25.

Para ello, ejecutamos la instrucción:

Secuencia[PrimoSiguiente[n],n,1,25]

El resultado será la lista de valores siguiente:

Lista
lista1 = {2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, 17, 17, 17, 17, 19, 19, 23, 23, 23, 23, 29, 29, 29}

A continuación, eliminamos las repeticiones utilizando el comando **Único**.

Único[lista1]

Devolverá una nueva lista con los números primos.

Vista Algebraica
Lista
lista1 = {2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, 17, 17, 17, 17, 19, 19, 23, 23, 23, 23, 29, 29, 29}
lista2 = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}

Por último, solo nos queda eliminar el número 29 que no cumple la condición de ser menor que 25. Lo conseguiremos con ayuda del comando **ConservaSi**.

ConservaSi[x<25,lista2]

Vista Algebraica
Lista
lista1 = {2, 3, 5, 5, 7, 7, 11, 11, 11, 11, 13, 13, 17, 17, 17, 17, 19, 19, 23, 23, 23, 23, 29, 29, 29}
lista2 = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
lista3 = {2, 3, 5, 7, 11, 13, 17, 19, 23}

Otra forma de resolver este ejemplo sería utilizando el comando **PrimoAnterior**, ejecutando la instrucción:

Secuencia[PrimoPrevio[n],n,1,25]

El resultado será la lista siguiente, en la que observamos algunos elementos indefinidos que aparecen representados por **?**, que son los que corresponden a los números primos previos al 1 y al 2 que no existen.

lista4 = {?, ?, 2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, 13, 13, 17, 17, 19, 19, 19, 19, 23, 23}

A continuación, utilizamos el comando **EliminaIndefinidos** para obtener una nueva lista en la que estos elementos desaparecerán.

Y por último, solo nos queda aplicar el comando **Único** a la nueva lista para eliminar los primos que aparecen repetidos.

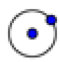
lista4 = {?, ?, 2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, 13, 13, 17, 17, 19, 19, 19, 19, 23, 23}
lista5 = {2, 3, 3, 5, 5, 7, 7, 7, 7, 11, 11, 13, 13, 13, 13, 17, 17, 19, 19, 19, 19, 23, 23}
lista6 = {2, 3, 5, 7, 11, 13, 17, 19, 23}

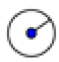
La línea de entrada permite el trabajo con comandos; lo que supone que cualquier objeto ya conocido que utiliza GeoGebra también se puede definir a través de la línea de entrada utilizando el comando adecuado.

Por ejemplo, ya sabemos dibujar una circunferencia utilizando la herramienta adecuada, pero supongamos que deseamos dibujar un conjunto de circunferencias concéntricas cuyo radio vaya aumentando una unidad con respecto a la anterior.

Evidentemente todas las circunferencias estarán relacionadas, por lo que podemos aprovechas las posibilidades del comando **Secuencia** para crearlas.

El comando que permitirá crear circunferencias desde la línea de entrada es **Circunferencia**, que utiliza distintos argumentos según la opción que deseamos utilizar.

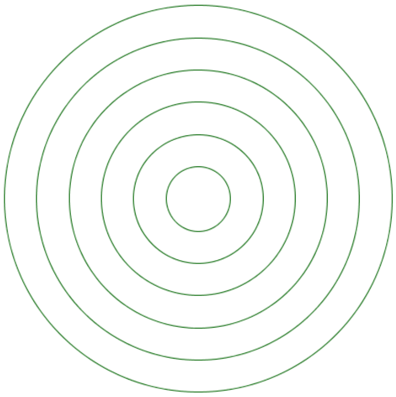
Circunferencia[punto centro, punto] equivale a la herramienta  .

Circunferencia[punto centro, número] equivale a la herramienta  .

Utilizamos esta segunda opción para ir cambiando el valor del radio; escribiremos en la línea de entrada la expresión siguiente:

Secuencia[Circunferencia[(0,0),r],r,1,6]

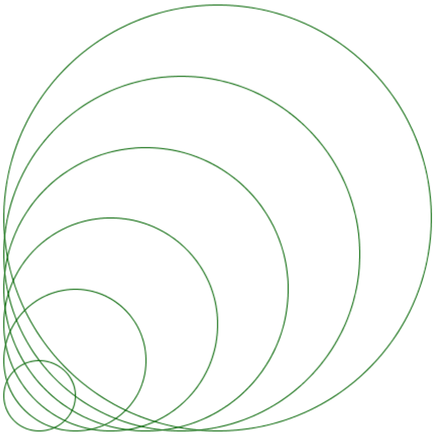
Obtendremos la representación de las seis circunferencias concéntricas, con centro en el punto (0,0), de radios 1, 2, …, 6.



Si en lugar del comando anterior, ejecutamos:

Secuencia[Circunferencia[(r,r),r],r,1,6]

la figura obtenida será la que aparece a continuación, en la que además de ir variando el radio, también cambia el centro.



Continuamos con otros ejemplos en los que además de utilizar **Secuencia** seguiremos trabajando con comandos a través de la línea de entrada.

Como actividad previa que necesitaremos para otras construcciones, resolvemos el siguiente ejemplo.

Ejemplo 2

Divide un segmento en un número de partes iguales.

Dibujamos un segmento AB que tendremos que dividir en un número de partes iguales, siendo este número una variable que definiremos a través de un deslizador, del tipo entero, al que hemos asignado valores entre 2 y 10.

Antes de continuar, indicaremos que cualquier punto tiene unas coordenadas, con respecto a los ejes que hasta ahora hemos ocultado en las distintas construcciones que hemos realizado. En GeoGebra las coordenadas de un punto, por ejemplo A, se representan por **(x(A), y(A))**.

Por tanto, para hacer referencia a la primera coordenada (abscisa) de A, tendremos que escribir **x(A)** y para la segunda (ordenada), escribiremos **y(A)**.

Para mostrar las coordenadas bastará con activar la opción muestra valor en las propiedades del punto.

Como el número de partes en las que dividir el segmento es variable, no es posible realizar los cálculos con los valores numéricos. Tenemos que definir un proceso que sirva para cualquier valor del deslizador.

El valor n del deslizador determina el número de partes en las que tenemos que dividir el segmento, cuyo nombre es a. Este nombre además tiene el valor de la longitud del segmento.

Por tanto, la longitud de cada una de las partes en las que lo dividiremos será $\frac{longitud.segmento}{númeropartes}$, es decir $\frac{a}{n}$.

El punto correspondiente a la primera división estará $\frac{a}{n}$ a la derecha de A, el segundo estará a $\frac{2a}{n}$ unidades a la derecha de A, el tercero a $\frac{3a}{n}$, y así sucesivamente hasta completar todas las divisiones.

Estos puntos se generan a través de una *lista*, para lo que necesitamos un comando denominado **Secuencia**.

Recordemos que este comando se ejecutará a través de la línea de entrada.

Para conseguir la división del segmento en n partes, tenemos que obtener los puntos que corresponden a las divisiones, relacionando las coordenadas de A, con la cantidad que en cada caso (partes) hay que añadir.

Las coordenadas de A están representadas por x(A) e y(A). La segunda coordenada no cambia, por lo que solo tenemos que ir sumando a la primera los valores correspondientes a cada parte.

Tendremos que los puntos correspondientes a la división serán:

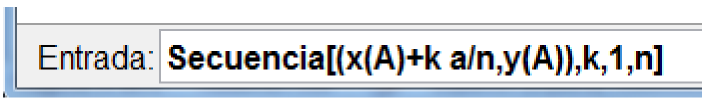
$$x(A)+\frac{a}{n}, x(A)+\frac{2a}{n}, x(A)+\frac{3a}{n}, \ldots$$

Por tanto, podemos definir una expresión que nos permita obtener todos los valores que será $x(A)+k\frac{a}{n}$, siendo *k* una variable que tomará valores desde 2 hasta el número de partes.

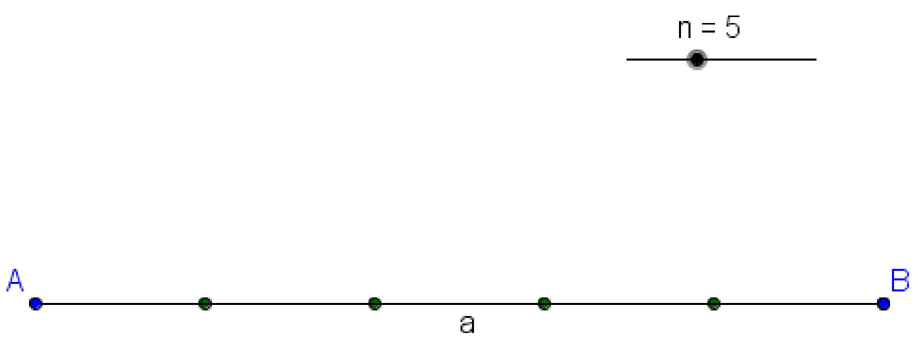
Por tanto, tendremos que definir la lista de puntos utilizando el comando **Secuencia** como:

Secuencia((x(A)+k a/n,y(A)),k,1,n)

Lo comprobamos ejecutando la expresión anterior a través de la línea de entrada.

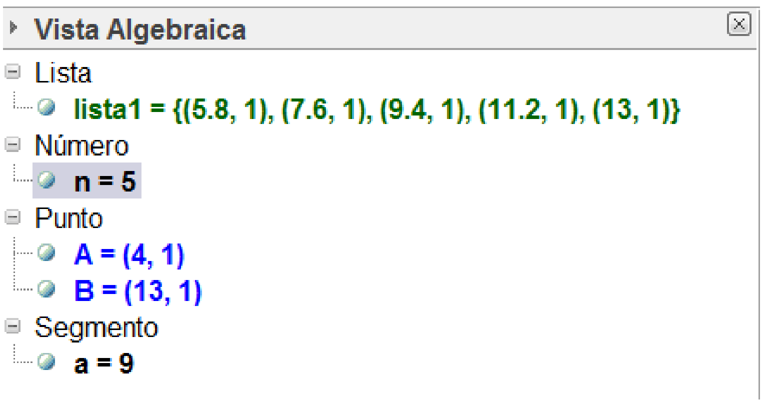


El resultado será una lista de puntos que corresponden a las divisiones buscadas.



Podemos comprobar que al cambiar el valor del deslizador aparecerán los puntos correspondientes al número de partes necesarias.

Al activar la vista algebraica aparecerá la lista con las coordenadas de los puntos que acabamos de obtener.



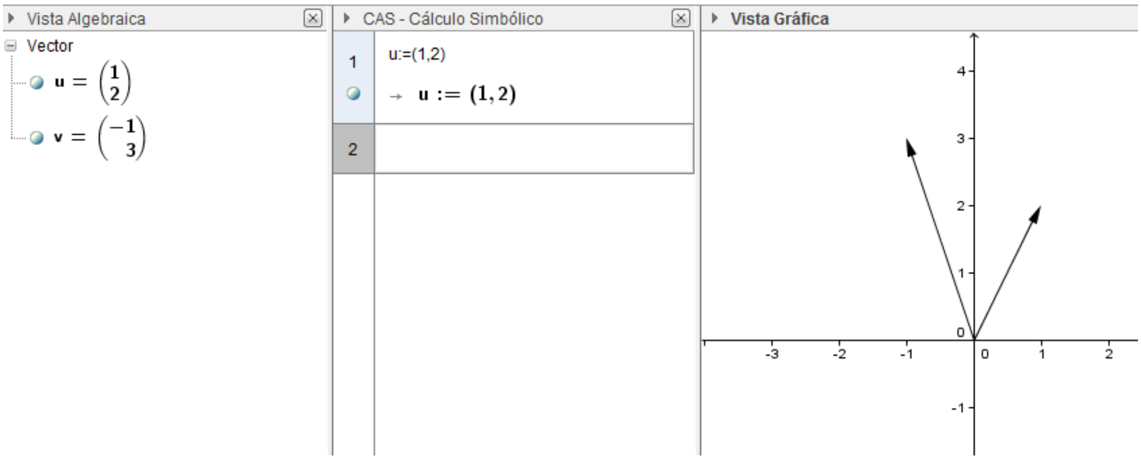
Continuamos que otro ejemplo que también resultará de utilidad para representar más adelante una fracción.

Vectores y matrices

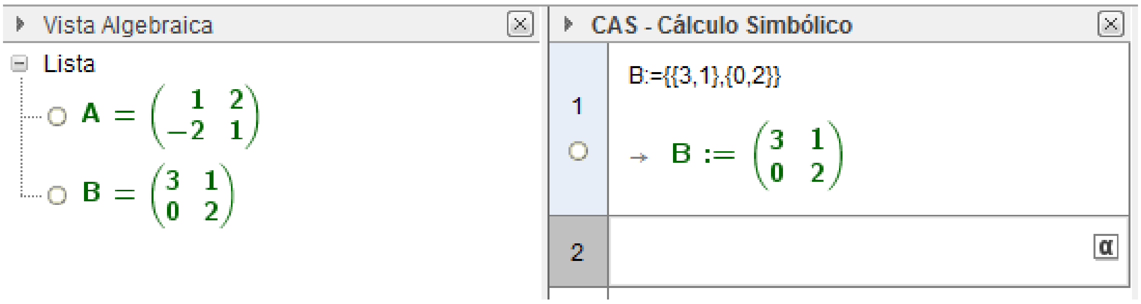
En primer lugar, indicaremos que un vector es una estructura del tipo lista y una matriz será una lista de listas. Por ejemplo si escribimos *{{1,0,1},{0,1,1}}* estaremos representando la matriz $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

La expresión de un vector como lista se hará para cualquier dimensión, aunque en el plano un vector se expresará mediante u=(x,y), utilizando un carácter en minúscula para el nombre ya que la misma notación, cuando el nombre es en mayúsculas, representará un punto.

Cuando la definición del vector se hace desde la vista CAS será necesario utilizar el signo **:=**, escribiendo u:=(x,y).



La definición de matrices se puede hacer desde la línea de entrada, en cuyo caso las expresiones resultantes irán a la Vista Algebraica, o bien se pueden introducir directamente desde la Vista CAS.

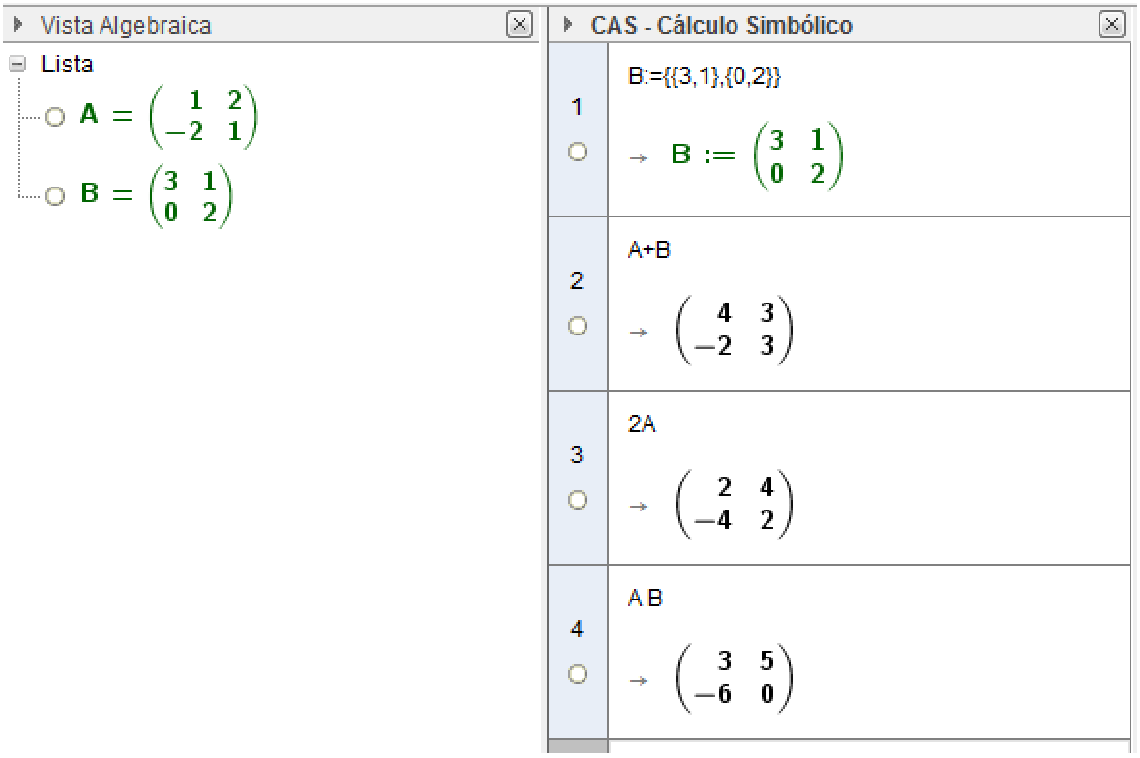


Una vez definidas las matrices, podremos obtener la suma, el producto por un número real o el producto de matrices.

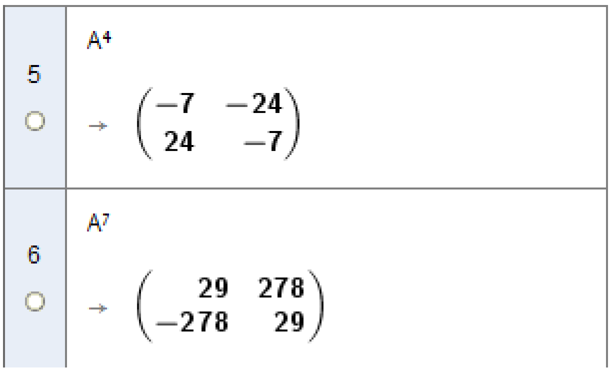
Los comandos expuestos para trabajar con lista se podrán utilizar con matrices ya que su estructura es la de una lista de listas.

Así, al ejecutar **Dimensión[A]** obtendremos {2,2}, siendo el primer valor el número de filas y el segundo el número de columnas.

Así, para dos matrices *A* y *B*, escribiremos *A* + *B*, *2A* o *A B* (el producto se representa con un espacio) para obtener la suma, el doble de *A* o el producto, respectivamente.



También se puede introducir a través de la línea de comandos *A^n* para obtener A^n.



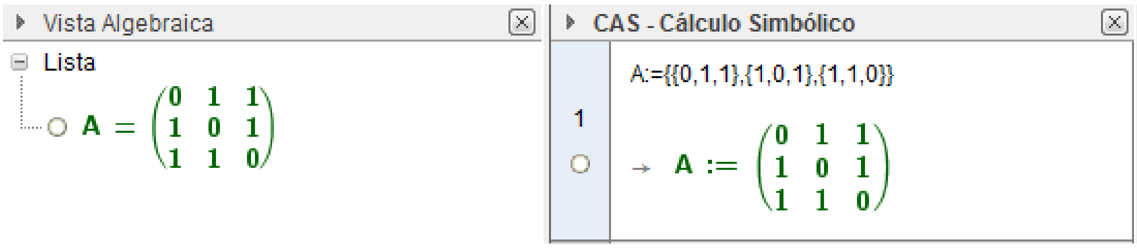
Ejemplo 3

Calcular A^2, A^3, A^4 *y* A^5.

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

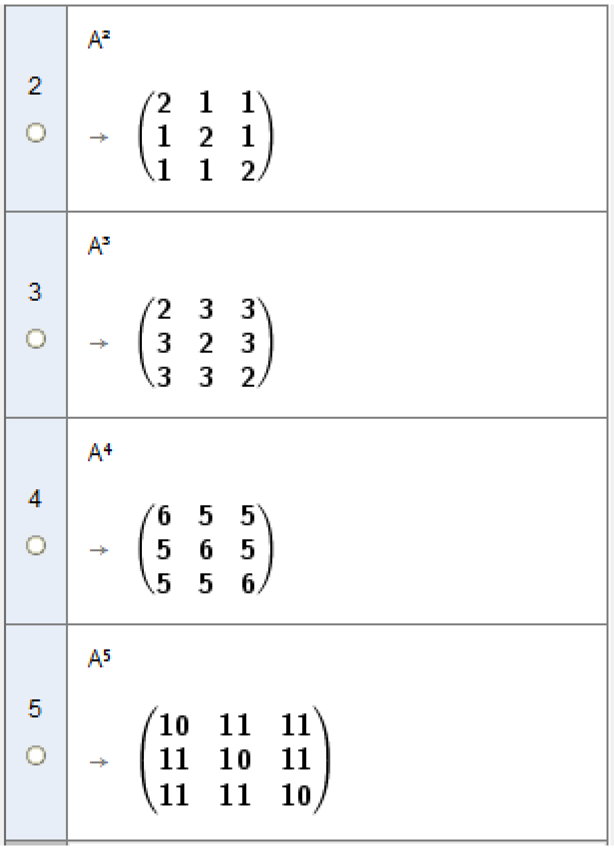
En primer lugar, definimos la matriz *A* introduciendo, a través de la línea de comandos o desde la vista CAS, la expresión siguiente:

A={{0,1,1},{1,0,1},{1,1,0}}



A continuación, basta escribir *A^2*, *A^3*, *A^4* y *A^5*, para obtener las potencias de la matriz *A*.

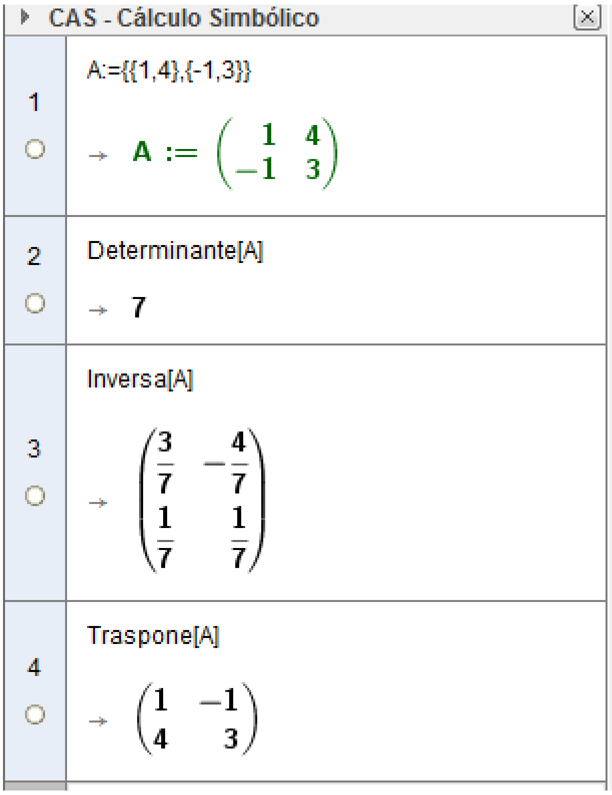
Los resultados aparecen en la imagen siguiente:



Otros comandos para trabajar con matrices:

- Matriz identidad de orden n: **Identidad[n]**
- Determinante de una matriz: **Determinante**.
- Inversa de una matriz: **Inversa**.
- Transpuesta de una matriz: **Traspone**.
- Rango de una matriz: **RangoMatriz**.

En la imagen siguiente aparece el valor del determinante de la matriz, la matriz inversa y la transpuesta de *A*.

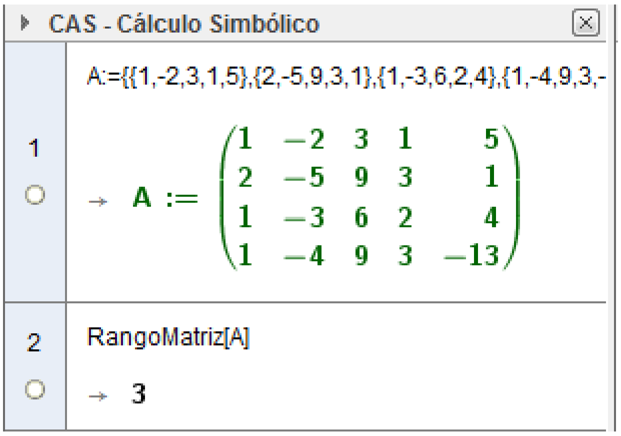


Ejemplo 4

Calcular el rango de la matriz

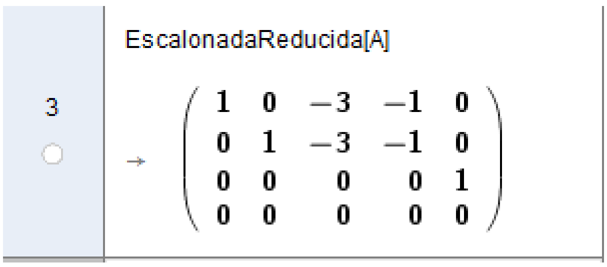
$$\begin{pmatrix} 1 & -2 & 3 & 1 & 5 \\ 2 & -5 & 9 & 3 & 1 \\ 1 & -3 & 6 & 2 & 4 \\ 1 & -4 & 9 & 3 & -13 \end{pmatrix}$$

Definimos, en primer lugar, la matriz anterior, utilizando, a continuación, el comando **RangoMatriz** para calcular su rango.



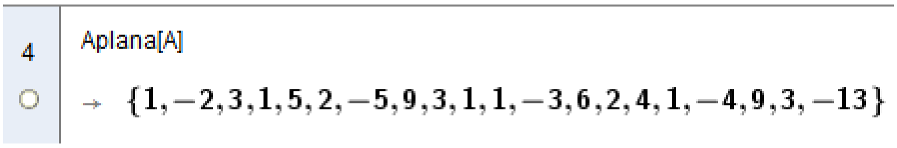
Tendremos, pues, que rango(A) = 3

La forma escalonada de la matriz se podrá obtener con la función **EscalonadaReducida**.



En ocasiones, interesa convertir una matriz a una lista, para lo cual disponemos del comando **Aplana**.

Para la matriz A del ejemplo anterior, este comando nos devolverá:



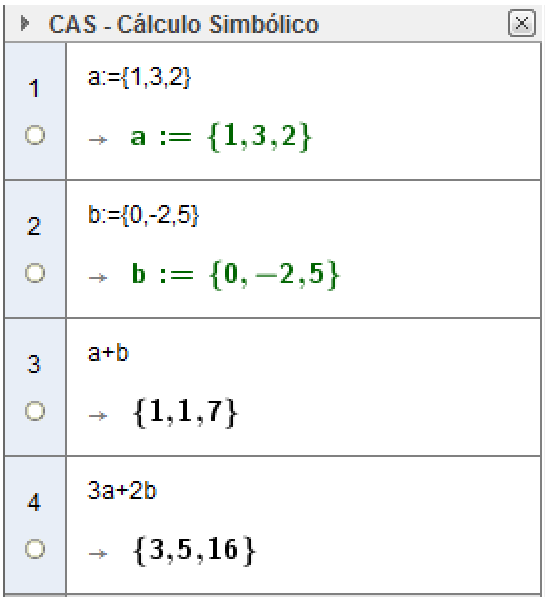
Operaciones con vectores

En general, un vector se puede considerar una lista, por lo que todas las funciones expuestas al comienzo de este tema se pueden aplicar sobre ellos.

Un caso particular será cuando los vectores sean de dimensión dos ya que admiten otra notación y sobre todo, permiten su representación en la vista gráfica.

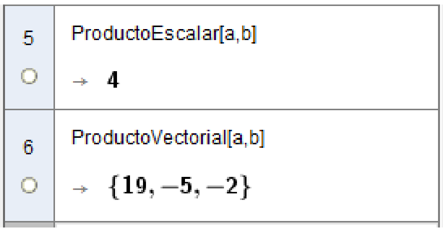
En este caso, ya hemos expuesto que su definición se puede hacer tanto desde la línea de entrada como desde la vista CAS.

La suma de vectores y el producto por un escalar se realizará utilizando los operadores **+** y *****, de manera similar a como ha quedado expuesto para matrices. El operador * se podrá omitir o sustituir por un espacio.

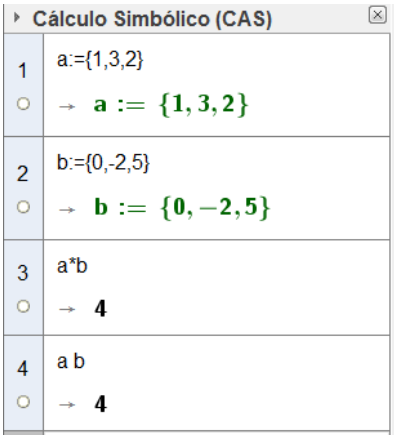


Para obtener el producto escalar y el producto vectorial utilizaremos las funciones **ProductoEscalar** y **ProductoVectorial**, respectivamente.

Para los dos vectores anteriores, los resultados del producto escalar y vectorial aparecen en la imagen siguiente:



Si escribimos *a*b* o *a b*, el resultado será el producto escalar de los vectores *a* por *b*.



Ejemplo 5

Dados los vectores a y b. Hallar k para que sean perpendiculares.

$$\vec{a} = (1, k, 3) \qquad\qquad \vec{b} = (-2, 2, 1-k)$$

Una vez definidos los dos vectores, calculamos su producto escalar y hallamos los valores de k para los que este producto es igual a 0.

Los resultados aparecen en la imagen siguiente

1	a:=(1,k,3) → $\mathbf{a := (1, k, 3)}$
2	b:=(-2,2,1-k) → $\mathbf{b := (-2, 2, -k + 1)}$
3	ProductoEscalar[a,b] → $\mathbf{3\ (-k + 1) + 2\ k - 2}$
4	Resuelve[(3 (-k + 1) + 2k - 2)] → $\mathbf{\{k = 1\}}$

:

Vectores en el plano

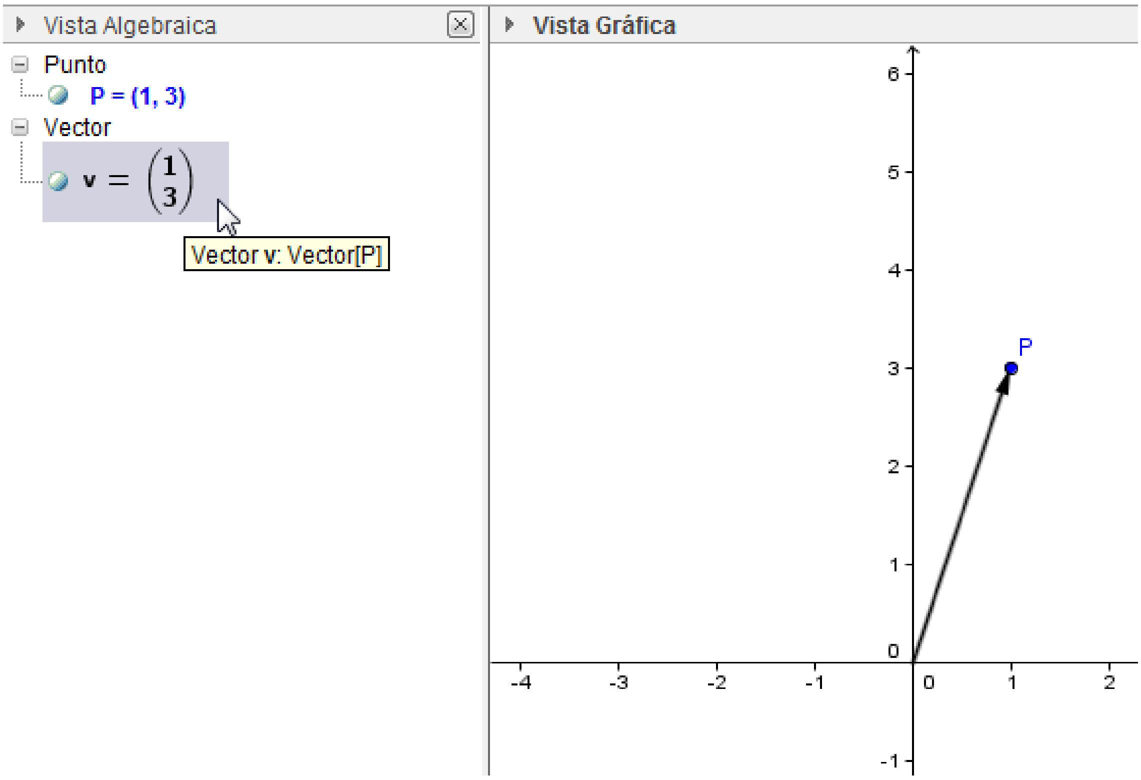
Además de las opciones expuestas anteriormente, un vector en el plano se podrá definir con el comando **Vector** para definirlo.

Su sintaxis es:

Vector[P] define el vector de posición del punto P.

Vector[(x,y)] define el vector de posición del punto de coordenadas (x,y).

Vector[P,Q] define el vector $\overrightarrow{PQ}$.



Además, de los comandos y herramientas disponibles para aplicar en la vista gráfica, para vectores en el plano GeoGebra ofrece algunos comandos que describimos a continuación.

AComplejo: aplicado sobre un vector, devuelve su expresión binómica y representa el vector. Puede aplicarse sobre un vector o sobre un punto.

AComplejo[{1,2}] devuelve la expresión 2i +1.

APolar: aplicado sobre un vector devuelve una lista con dos valores que corresponden al módulo y al argumento.

También se puede aplicar sobre un complejo expresado en forma binómica para obtener su expresión en polares.

1	a:=(1,2) → $\mathbf{a := (1, 2)}$
2	AComplejo[a] → $1 + 2\,\imath$
3	APolar[a] → $\left(\sqrt{5}; \arctan(2)\right)$
4	APolar[1-í] → $\left(\sqrt{2}; -\frac{\pi}{4}\right)$

Es conveniente observar en la imagen anterior, que para aplicar este comando el vector tiene que definirse utilizando la notación específica para el plano.

APunto: aplicado sobre un número complejo, devuelve las coordenadas de su afijo.

6	APunto[2-3í] → $(2, -3)$

La unidad imaginaria se introduce pulsando **Alt – i**.

Otros comandos disponibles para aplicar sobre vectores en el plano son:

VectorUnitario: devuelve el vector con la misma dirección y sentido con módulo igual a 1.

Para el vector b anterior, tendremos:

7	VectorUnitario[b] → $\left(\frac{1}{\sqrt{5}}, \frac{2}{\sqrt{5}}\right)$

Este comando se puede aplicar sobre una recta, semirrecta o sobre la ecuación de una recta para obtener el vector unitario correspondiente a su dirección.

VectorNormal: devuelve un vector perpendicular al vector sobre el que se aplica.

2 VectorNormal[b]

$\rightarrow (-2, 1)$

VectorNormalUnitario: devuelve un vector perpendicular al vector sobre el que se aplica, de módulo 1.

3 VectorNormalUnitario[b]

$\rightarrow \left(-\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right)$

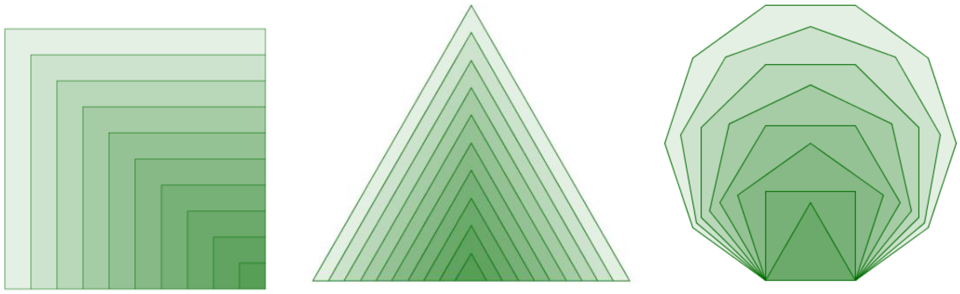
Actividades propuestas

1. Generar una lista correspondiente a la función $y = x^3 - 1$ para valores de la variable comprendidos entre 1 y 2, con un incremento de 0´2.

2. Generar dos listas con veinte números enteros comprendidos entre 1 y 20. Hallar la unión y la intersección de las dos listas. (El comando **AleatorioEntre[m,n]** genera un número entero aleatorio en el intervalo [m,n]).

3. Generar un vector de dimensión 20 cuyos elementos corresponden a la sucesión $a_n = 3n^2 - n$.

4. Utilizando el comando **Secuencia** realiza las siguientes construcciones. Para dibujar un polígono regular puedes utilizar el comando **Polígono[punto,punto,n]** que dibujará el polígono regular de n lados, siendo uno de los lados el segmento establecido por los dos puntos indicados como argumentos.



5. Dadas las matrices *A* y *B*, calcular *A + 2 B*, *3 A – B* y *A x B*.

$$A = \begin{pmatrix} 0 & -4 \\ 2 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 3 \\ -2 & 1 \end{pmatrix}$$

6. Encontrar una expresión para A^n.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

7. Hallar la matriz inversa de *B + I*, siendo

$$B = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

8. Hallar la matriz $2A^2 - 3A - I$. $A = \begin{pmatrix} 0 & -1 & -2 \\ -1 & 0 & -2 \\ 1 & 1 & 3 \end{pmatrix}$

9. Calcular A^2 y A^3. Encontrar la expresión de A^n. $A = \begin{pmatrix} p & p \\ p & p \end{pmatrix}$

10. Comprobar si la matriz A verifica $(A^{-1})^t = (A^t)^{-1}$

$$A = \begin{pmatrix} -1 & 3 \\ 2 & -5 \end{pmatrix}$$

11. Determinar los valores de x para los cuales la matriz A es singular. Hallar la matriz inversa para x=1.

$$A = \begin{pmatrix} x & 1 & 0 \\ 1 & x & 2 \\ 1 & 0 & -1 \end{pmatrix}$$

12. Sean a y b dos números reales. Hallar para que valores de a y b, la matriz A es singular.

    Determinar la inversa de A para cada valor de a y de b para los cuales la matriz es invertible.

$$A = \begin{pmatrix} 0 & a & 1 & 0 & b \\ 1 & 0 & 0 & b & 0 \\ 0 & 1 & b & 0 & 1 \\ b & 0 & 0 & 1 & 0 \\ 0 & b & 1 & 0 & b \end{pmatrix}$$

13. Para cada número natural n, se define la matriz cuadrada A_n. $A_n(i,j) = \begin{cases} 0 & si \;\; i = j \\ 1 & si \;\; i \neq j \end{cases}$. Deducir cuál es el valor del determinante de A_n.

14. Calcular los valores de t para los que el determinante $\begin{vmatrix} 2 & t & 0 \\ t & 2 & 1 \\ 3 & 0 & 1 \end{vmatrix}$ es positivo.

    Determinar el mayor valor que alcanza.

15. Resolver la ecuación $\begin{vmatrix} 1 & -1 & 2 \\ 2 & x & 1 \\ 1 & 3 & x \end{vmatrix} = 10$

16. Determinar los elementos de la matriz A. Hallar su inversa.

$$a_{ij} = (-1)^{i+j}\,(2+j) \quad \mathrm{i} = 1{,}2 \quad \mathrm{j} = 1{,}2\,.$$

17. Dada la matriz A, determinar si es posible, un valor k, tal que (A - K I)^2 sea la matriz nula. $A = \begin{pmatrix} 0 & -1 & -2 \\ -1 & 0 & -2 \\ 1 & 1 & 3 \end{pmatrix}$

18. Hallar el rango de las matrices:

$$\begin{pmatrix} 1 & -1 & 3 \\ -2 & 2 & -6 \\ 3 & -3 & 9 \end{pmatrix} \qquad \begin{pmatrix} 2 & 1 \\ -5 & 0 \\ 4 & -2 \end{pmatrix}$$

19. Hallar el rango de la matriz

$$\begin{pmatrix} 5 & 0 & -3 & 4 \\ 1 & -1 & 0 & 2 \\ -2 & 0 & 3 & 3 \\ 3 & 1 & -2 & 0 \end{pmatrix}$$

20. Calcular el determinante de las matrices

$$\begin{pmatrix} x^2+1 & x & 0 & 0 \\ x & x^2+1 & x & 0 \\ 0 & x & x^2+1 & x \\ 0 & 0 & x & x^2+1 \end{pmatrix} \qquad \begin{pmatrix} x^2+1 & x & 0 & 0 & 0 \\ x & x^2+1 & x & 0 & 0 \\ 0 & x & x^2+1 & x & 0 \\ 0 & 0 & x & x^2+1 & x \\ 0 & 0 & 0 & x & x^2+1 \end{pmatrix}$$

¿Es posible deducir una fórmula para el determinante de las matrices anteriores de orden n?

Comprobar la expresión obtenida para los determinantes de orden 6, 7 y 8.