UNIDAD DIDÁCTICA

10 APLICACIONES DE USO EN INGENIERÍA. GEOGEBRA Y OCTAVE

Objetivos de la unidad

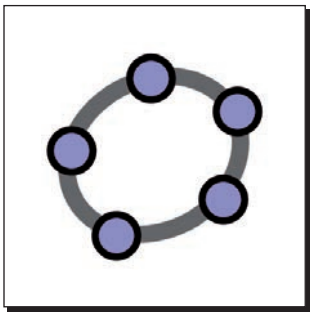
OBJETIVOS DE LA UNIDAD

En la presente unidad se aborda el estudio de algunos programas informáticos que pueden servir de ayuda para los estudiantes de ingeniería a la hora de resolver e interpretar distintos problemas matemáticos.

El primero de ellos es GeoGebra, un *software* de matemáticas dinámicas (interactivas) creado para todos los niveles educativos, y que reúne cálculo, álgebra, geometría, gráficos, hoja de cálculo y estadística en un único programa muy intuitivo y fácil de utilizar. Puede servir de apoyo en los estudios de las disciplinas de ciencias, tecnología, ingeniería y matemáticas (STEM: *Science, Technology, Engineering & Mathematics*).

El segundo es Octave, un lenguaje de programación matemático de alto nivel y a la vez el *software* de interpretación del código matemático, orientado a la computación numérica, y particularmente al cálculo algebraico y al análisis y visualización de datos. Mediante un sencillo lenguaje de programación de alto nivel, y una enorme colección de librerías, Octave sirve de apoyo al cálculo matemático avanzado en todas las áreas científicas, tanto en las relacionadas con STEM, como en otras donde puedan realizarse cálculos matemáticos, como la contabilidad y las finanzas.

1. GEOGEBRA. INTRODUCCIÓN



GeoGebra es un *software* matemático creado por Markus Hohenwarter, quien comenzó el proyecto en el año 2001 como parte de su tesis doctoral en la Universidad de Salzburgo y, actualmente, continúa desarrollándolo en la Universidad de Atlantic, Florida. Es un *software* libre, de código abierto; es interactivo y está diseñado para la enseñanza y el aprendizaje de las matemáticas en colegios y universidades. El entorno ofrece un lugar donde el álgebra y la geometría están conectados de forma plena. Las últimas aportaciones que se han realizado le proporcionan intereses fuera del ámbito de la educación. Así, reúne geometría, álgebra y cálculo y, por tanto, puede ser utilizado también en física, programación lineal, estimaciones para la decisión estratégica, así como en otras muchas disciplinas.

Empezó siendo un *software* de geometría dinámica, pero tras las aportaciones de las últimas versiones se ha convertido en un programa de matemática dinámica, donde se conjugan la geometría interactiva, el cálculo, el álgebra, así como estadísticas y los registros gráficos, con organización en tablas y la formulación simbólica.

Las sucesivas versiones de GeoGebra han ido añadiendo diferentes características así como nuevos comandos, que pueden verse en la Tabla 1. «Principales características de cada una de las versiones GeoGebra».

La primera versión apareció en enero de 2002, en inglés y alemán, aunque se tradujo rápidamente a otros idiomas, entre ellos el castellano. La facilidad en el manejo y su gran versatilidad para el uso en la educación, junto con su gratuidad, contribuyeron a la gran expansión entre el profesorado.

Las versión 2.0 aparece dos años después, mejorando la representación gráfica, el zoom y las herramientas de análisis integral y diferencial. Se produce una gran demanda del programa y de los desarrolladores, que empiezan a realizar un gran trabajo colaborativo.

Entre la versión 2.0 y la 3.0 aparece en la web de GeoGebra un foro con versiones en distintos idiomas (versión inglesa <http://forum.geogebra.org/>, con la posibilidad de elegir otros muchos idiomas) y una wiki (versión en español <http://archive.geogebra.org/en/wiki/index.php/Spanish>), que tuvieron un gra éxito entre los usuarios.

La versión 3.0 fue lanzada en marzo de 2009, con mejoras en polígonos, áreas, pendientes, funciones por partes, operaciones lógicas. También la forma de exportación mejoró al incorporar formatos como PDF, SVG, EMF, así como web dinámica. También aparece por primera vez la versión traducida al español.

Con la versión 3.2 se avanza al poder trabajar con tablas dinámicas, es decir, con la Vista de Hoja de Cálculo y comandos de funciones estadísticas. También se amplía bastante las posibilidades de uso de matrices y números complejos.

La versión 4.0 lanzada en octubre de 2011 incorpora GeoGebraTube (<http://tube.geogebra.org/>), que nos permite compartir las hojas creadas con el resto de la comunidad a través del propio portal. También aparece la vista estadística para el análisis de datos y probabilidad.

La versión 4.2 es una consecuencia de la incorporación del cálculo simbólico, con la Vista GeoGebra CAS, Computación Algebraica Simbólica, que permite operar simbólicamente y pueden emplearse literales sin ningún valor asignado en las operaciones simbólicas. Por poner un pequeño ejemplo, si se introduce (a + b) ^ 2, se evalúa como a ^ 2 + 2 * a * b + b ^ 2. También incorpora nuevos comandos de JavaScript.

La versión 5.0 nos proporciona el soporte para las tres dimensiones, creando la Vista 3D. Aunque en la web se anuncia la vista Python, en la versión 5.0.50 aún no aparece, suponemos que está en desarrollo. También permite crear libros dentro de GeoGebratube, que posibilita poder ordenar los recursos que vamos encontrando o creando.

La versión 6 se crea pensando en dispositivos móviles, con panel táctil desplegable de entrada. Se unifican todos los materiales creados en recursos, versión de examen que permite utilizar el potencial de GeoGebra durante los exámenes en papel (como utilizar una calculadora gráfica o una calculadora de bolsillo), a la vez que se restringe el acceso a internet o a otros programas instalados en la computadora que no deberían ser utilizados durante el examen.

Tabla 1. **Principales características de cada una de las versiones GeoGebra**

Versión	Lanzamiento	Características
1.0	enero 2002	• Objetos disponibles: punto, vector, recta, ángulo, número y sección cónica. • Construcciones con el ratón y la barra de entrada. • Extensión de los archivos: *.geo*. • Idiomas: inglés y alemán.
2.0	enero 2004	• Funciones en x, graficación, derivadas, integrales, tangente en un punto. • Funciones hiperbólicas. • Exportación de gráficos como EPS, PNG y JPG. • Extensión de los archivos: *.ggb* (XML comprimido). • Idiomas: inglés y alemán.
3.0	marzo 2009	• Polígonos regulares, curvas parámetricas, listas. • Nuevas herramientas: área, pendiente, longitud, perímetro. • Funciones por partes. • Operaciones lógicas binarias. • Inserción de texto (y fórmulas en LaTeX) e imágenes. • Exportación de gráficos como PDF, SVG, EMF y PSTricks. • Exportación como página web dinámica. • Ajustes almacenables. • Idiomas: 39 (incluido español por primera vez).
3.2	junio 2009	• Vista de Hoja de Cálculo. • Nuevas herramientas: compás, inversión, cónicas. • Comandos de funciones estadísticas y gráficos. • Matrices y números complejos. • Capas y colores dinámicos. • Exportación a PGF/TikZ. • Idiomas: 45.
4.0	octubre 2011	• GeoGebraTube (compartición de hojas dinámicas en línea). • GeoGebraPrim (versión para estudiantes pequeños). • Requiere Java 5. • Nuevas herramientas: análisis de datos, cálculo de probabilidades, inspección de funciones, polígonos rígidos, polilíneas.

.../...

Versión	Lanzamiento	Características
.../...		
4.0 *(cont.)*	octubre 2011 *(cont.)*	• Desigualdades, inecuaciones, ecuaciones implícitas y funciones varias. • Logaritmos en cualquier base. • Copiar y pegar. • Posibilidad de asociar guiones a cada objeto en lenguaje propio o JavaScript. • Botones, cajas de entrada y herramienta lápiz. • Exportación a GIF animado y archivo de Asymptote. • Idiomas: 50.
4.2	diciembre 2012	• Soporte para cálculo simbólico: vista algebraica CAS. • Nuevos comandos de GeoGebra, LaTex y JavaScript.
4.4	diciembre 2013	• Nuevo motor de álgebra simbólica. • Mayor integración con GeoGebraTube. • Eliminada la exportación a página web dinámica HTML. • Nuevos comandos.
5.0	septiembre 2014	• Soporte para 3 dimensiones: vista 3D. • Nuevas herramientas y objetos: plano, prisma recto, esfera; pirámide, cilindro, cono. • Ventana Python y tortugas como en logo.
6.0	septiembre 2017	• Diseñado para uso en dispositivos móviles. • Versión en App Store, Google play, Windows Store. • Versión web mejorada. • GeoGebra Realidad Aumentada. • Modo examen.

2. DESCRIPCIÓN DE LA APLICACIÓN

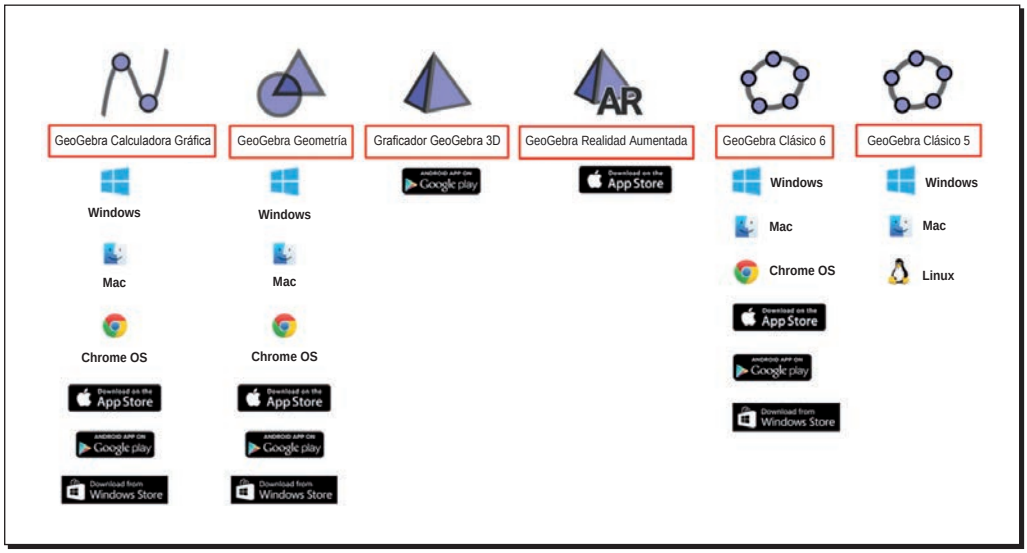
Desde la versión 5 de la aplicación no hace falta descargársela para poder utilizarla, ya que se puede usar como servicio web. Desde el lanzamiento de la versión 6, la aplicación se enfoca a dispositivos móviles. En la última versión, se han separado GeoGebra Calculadora Gráfica y GeoGebra Geometría, Graficador GeoGebra 3D y GeoGebra Realidad Aumentada como aplicaciones distintas, aunque mantienen la versión GeoGebra Clásico (tanto en versión 5 como en versión 6).

Está disponible para descargar en ordenadores personales con el nombre de «Versión para escritorio». En la versión web y en las aplicaciones para teléfonos y tabletas, aunque ambas son muy parecidas, existen ligeras diferencias en cuanto a la visualización. En este manual se va a describir la «Versión de escritorio» de la versión 5, por lo que podría haber ligeras diferencias con respecto a la última versión.

Para la descarga la instalación basta con ir a la siguiente dirección: <http://www.geogebra.org/download>.

El sistema proporcionará las distintas opciones de instalación en los diferentes sistemas operativos, para tabletas o para móviles, según la aplicación que se desee instalar.

Figura 1. **Resumen de las aplicaciones y sistemas operativos disponibles para la instalación**



Cuando se inicia el programa, aparece una ventana similar a la que se muestra en la figura 2 (versión 5) o en la figura 3 (versión 6), que consta de varias vistas que se pueden ir abriendo o cerrando a voluntad. En estas figuras se pueden ver algunas de las distintas vistas, entre las que distinguimos de izquierda a derecha, la Vista Algebraica, la Gráfica, la de Cálculo Simbólico, la de Hoja de Cálculo Algebraica. Estas vistas podemos cerrarlas o volverlas a abrir en el menú de Vista.

La disposición de estas vistas puede ser modificada a voluntad, pudiéndose mostrar en otras ventanas o en otra ubicación, adecuándola al contenido que deseemos reflejar. Pueden mostrar distintas «apariencias» o «perspectivas», según lo que queramos trabajar. Se despliega desde la pequeña flecha que hay en la barra lateral derecha.

Figura 2. **Vistas en GeoGebra 5**

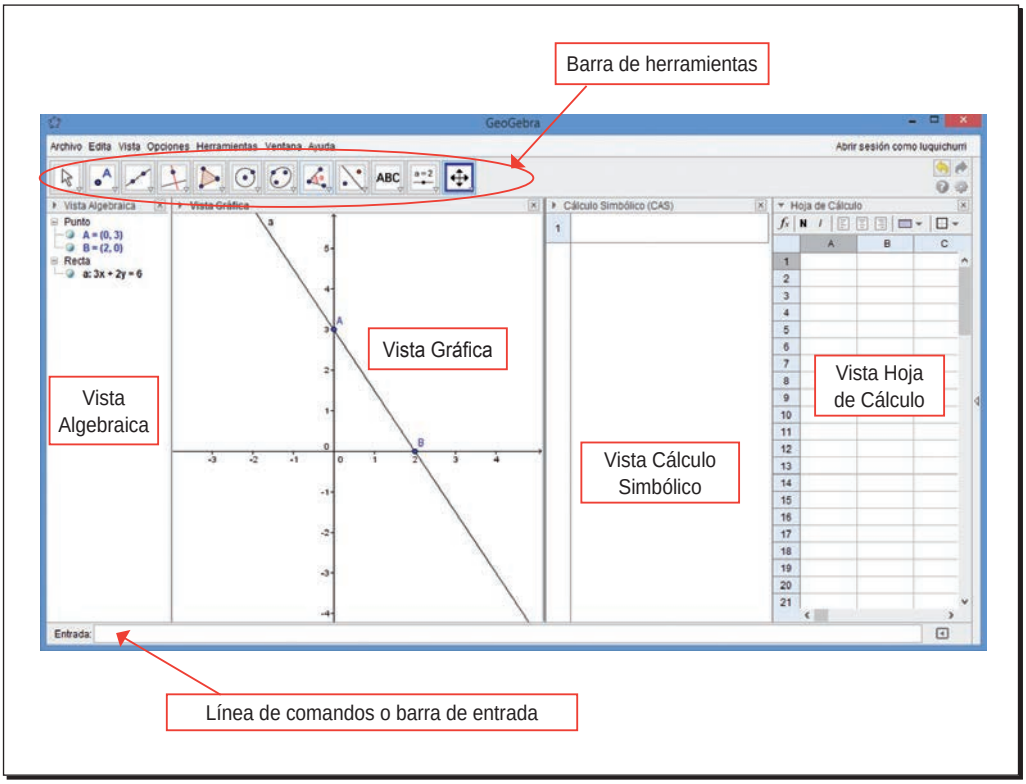
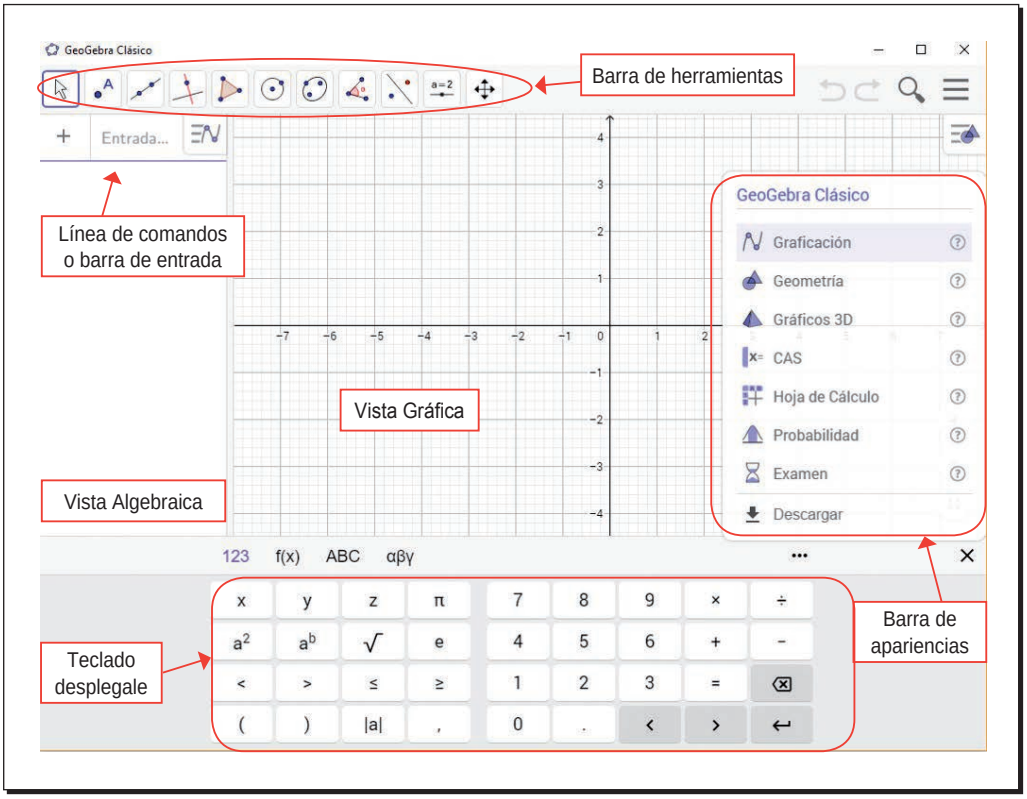


Figura 3. **Vistas en GeoGebra 6**



2.1. VISTA ALGEBRAICA Y VISTA GRÁFICA

Estas dos vistas van ligadas y casi siempre van de la mano, lo que se hace en una se refleja en la otra, por eso las vamos a explicar en el mismo apartado. En la Vista Algebraica irán apareciendo todos los elementos que vayamos creando para el desarrollo de nuestro material didáctico, y estos pueden aparecer o no en la Vista Gráfica, dependiendo de si los queremos mostrar o no, ya que habrá algunos que los obtengamos como paso intermedio pero no deseamos mostrarlos. En las figuras 5, 11, 12, 16 y 20 se muestran ejemplos de ambas vistas en distintas construcciones. En la parte izquierda está la Vista Algebraica y en la derecha, la Gráfica. No todos los objetos de la Vista Algebraica se muestran en la Gráfica, solo aquellos que están señalados en círculo en color sólido, mientras que los que aparecen con círculo en color blanco no son mostrados.

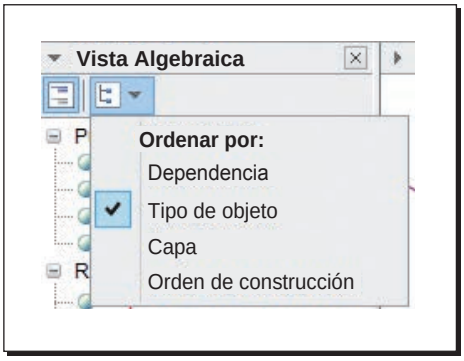
En cada uno de los objetos que creemos podremos editar y modificar sus propiedades Apretando el botón derecho del ratón sobre dicho objeto, aparecerá entonces una ventana que nos permitirá elegir algunas opciones y las propiedades de dicho objeto.

Los objetos pueden ser de tres tipos:

- **Objetos libres.** Su valor o su posición no dependen de ningún otro objeto. Se pueden crear directamente con una herramienta determinada del menú de herramientas o por ingreso directo en la barra de entrada. Normalmente, estos objetos pueden modificarse, ya que no están sujetos a restricciones.
- **Objetos dependientes.** Son objetos que dependen de otro u otros objetos. Se suelen crear con herramientas o comandos.
- **Objetos auxiliares.** Son objetos que marcamos como tales o se crean con alguna herramienta, de forma que se utilizan para la construcción de alguna otra. Son similares a los dependientes, pero con la peculiaridad de que se pueden ocultar de la vista algebraica pulsando sobre el botón Objetos auxiliares  .

Los objetos se pueden ordenar por dependencia, tipo de objeto, capa u orden de construcción, desplegando el menú que aparece en el botón, tal y como se muestra en la figura 4.

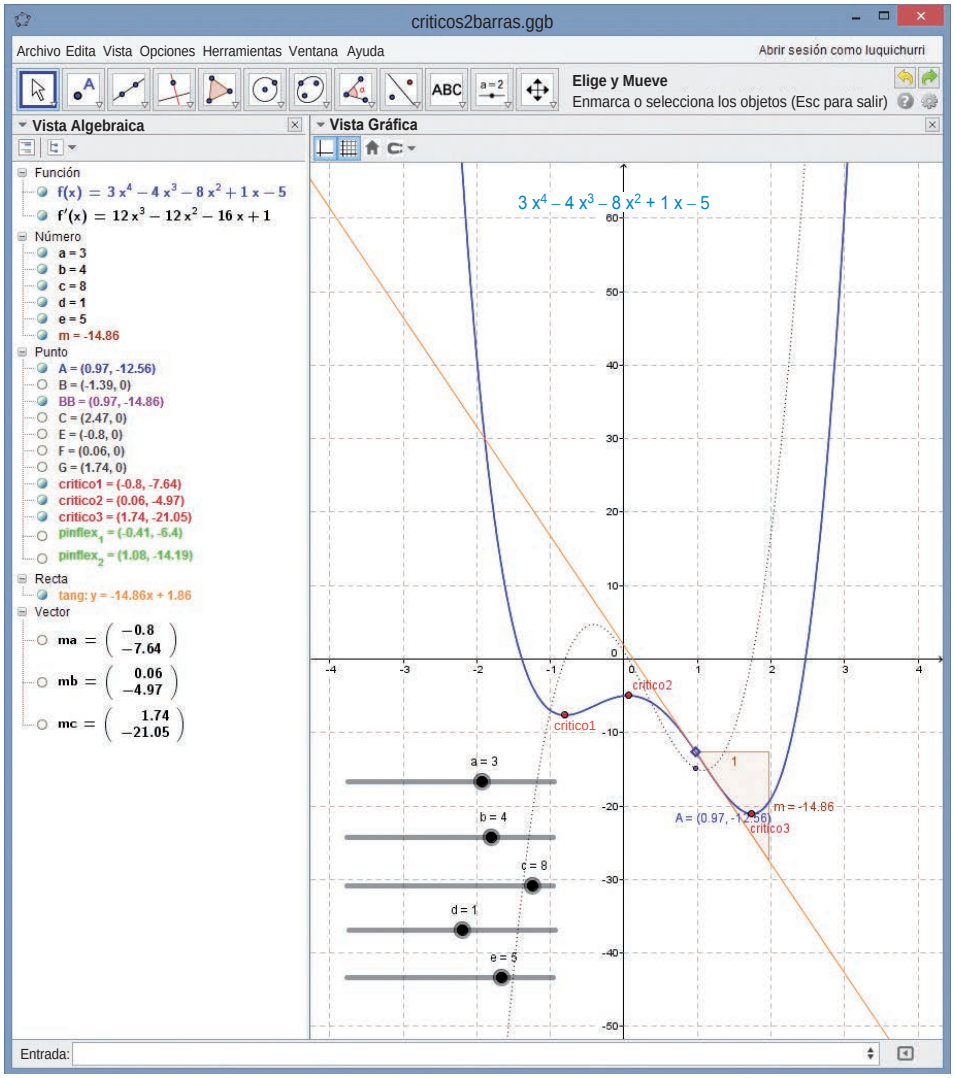
Figura 4. **Ordenación de los objetos según distintos criterios en la vista gráfica**



La Vista Gráfica va mostrando todos los objetos que tenemos visibles, pero además se puede personalizar para incluir diversos tipos de cuadrículas y modificar el formato de los ejes.

Además, existe un segunda Vista Grafica que nos permitirá que algunos objetos se muestren en esta vista y/o en la Vista Gráfica 1. Se activa desde el menú Vista y podemos hacer que los objetos se dibujen en la Vista Grafica 1, en la 2, o en ambas, desde las propiedades avanzadas de cada objeto. Por defecto, se dibujan en la que tengamos activa.

Figura 5. **Vistas Algebraica (izquierda) y Gráfica (derecha)**



2.2. VISTA CAS (Computación Algebraica Simbólica)

Como ya hemos dicho antes, desde la versión 4.2 ya se admite el cálculo simbólico.

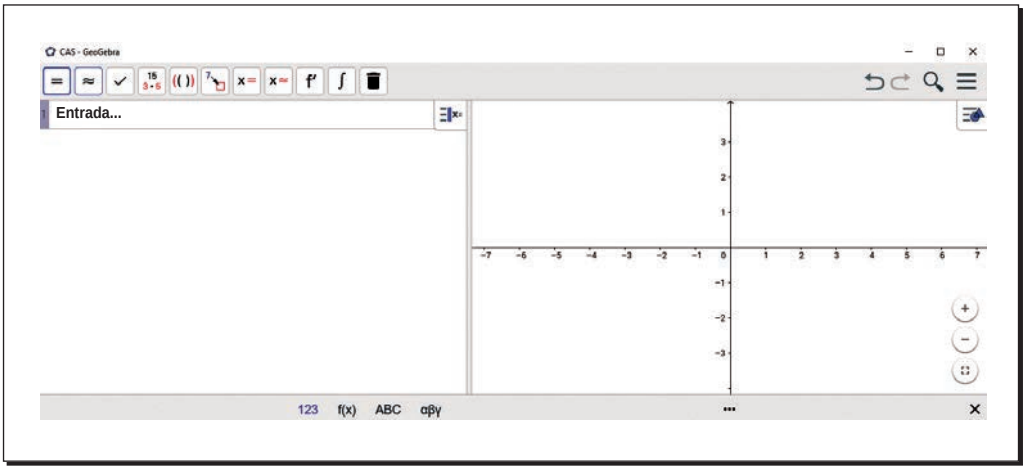
Anteriormente a esta versión se debía recurrir a otro tipo de *software* para evaluar expresiones aritméticas, factorizar, etc.; *software* tales como Derive, Mathematica, Mapple, MathLab o Wiris, entre otros.

GeoGebra CAS utiliza una programación basada en otro programa de cálculo simbólico de código abierto, el programa Maxima, que cuenta con un gran número de funciones para poder hacer uso de la manipulación simbólica.

La apariencia CAS incluye la Vista CAS y la Gráfica, alternándose en la barra de herramientas, según cuál sea la vista activa.

Los ejes coordenados siempre aparecerán visibles por defecto en la Vista Gráfica.

Figura 6. **Vistas CAS (izquierda) y Gráfica (derecha)**



2.3. VISTA HOJA DE CÁLCULO

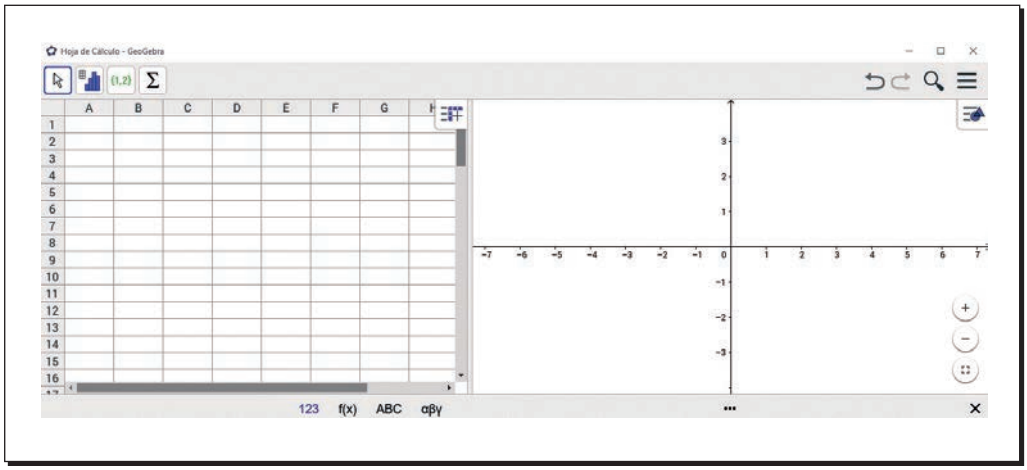
Esta vista tiene una apariencia muy similar a cualquier hoja de cálculo; las columnas están identificadas por letras y las filas, por números (similar a Excel). De esta forma cualquier celda tiene una denominación específica que nos va a permitir dirigirnos a cada una de ellas. Así, por ejemplo, la celda de la columna A y la fila 1, se llama A1 (figura 7).

Con esto, el nombre de una celda determinada puede usarse en comandos y en expresiones para referirse a su contenido. Dentro de cada celda pueden introducirse números u otros objetos, como coordenadas de puntos, funciones, comandos o textos.

La apariencia hoja de cálculo incluye la Vista Hoja de Cálculo y la Gráfica, alternándose en la barra de herramientas según cuál sea la vista activa.

Los ejes coordenados siempre aparecerán visibles por defecto en la Vista Gráfica.

Figura 7. **Vistas Hoja de Cálculo (izquierda) y Gráfica (derecha)**

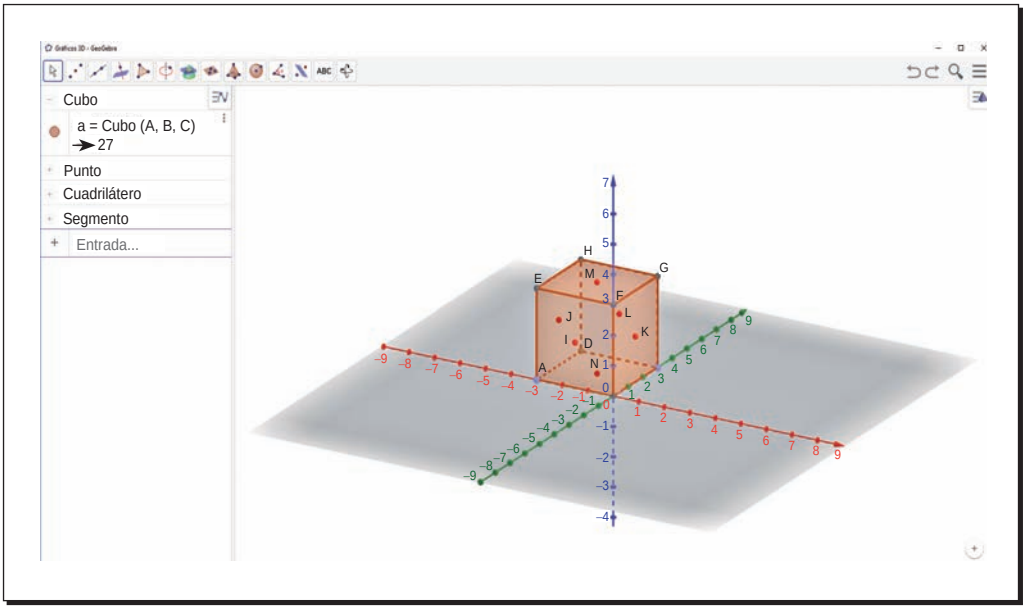


2.4. VISTA GRÁFICA 3D

Por defecto, la Vista 3D se abre junto a la Vista Algebraica, como se puede apreciar en la figura 8.

Esta vista es muy útil para geometría tridimensional, ya que nos permite ver los objetos en tres dimensiones, y desde distintas vistas.

Figura 8. **Vista Gráfica 3D**

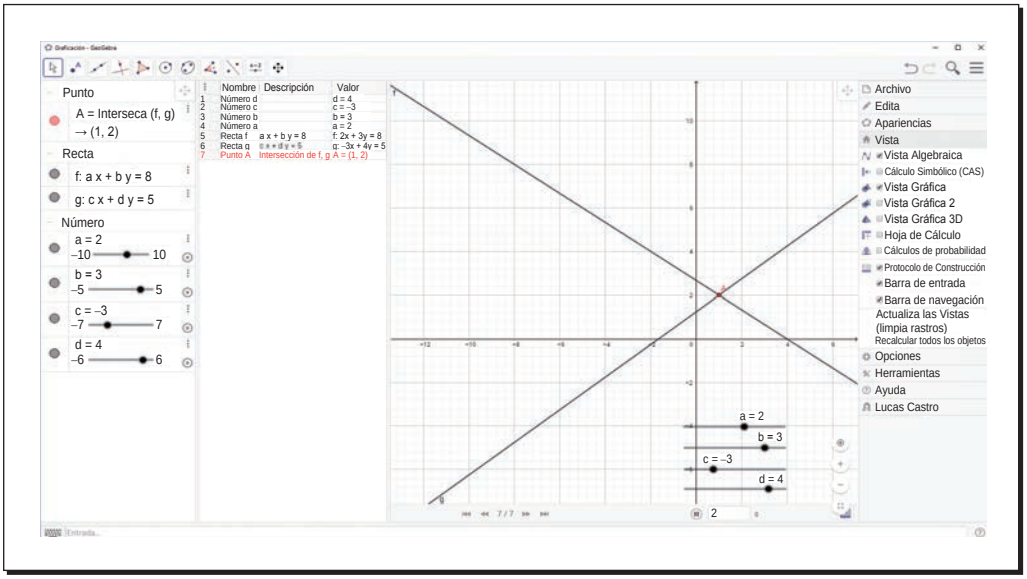


2.5. VISTA PROTOCOLO DE CONSTRUCCIÓN

El ítem Protocolo de la Construcción del menú Vista permite visualizar una tabla interactiva que indica cada uno de los pasos seguidos en la construcción, lo que permite rehacer el boceto realizado, paso a paso.

Se puede usar la barra de navegación que aparece en la parte inferior de la Vista Gráfica. Esta vista es muy útil cuando tengamos una construcción que hemos descargado y no sabemos cómo se ha realizado, o incluso para una construcción nuestra que hemos olvidado cómo fue realizada.

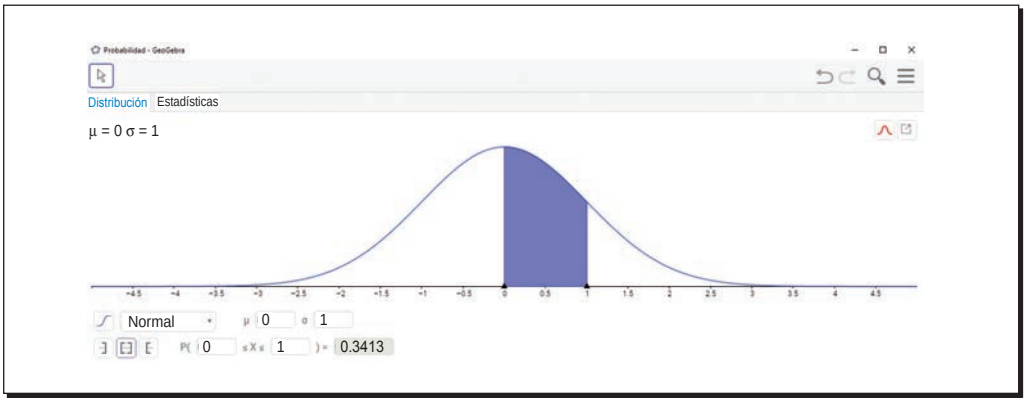
Figura 9. **Vistas Algebraica (izquierda), Protocolo de construcción (centro) y Gráfica (derecha)**



2.6. VISTA CALCULADORA DE PROBABILIDADES

Esta vista muestra la Calculadora de Probabilidades, que nos permitirá el cálculo y graficación de distribuciones de probabilidad. Es muy sencilla de utilizar, simplemente debemos seleccionar la distribución que deseemos (tanto discreta como continua); por ejemplo, distribución normal, *student*, binomial, etc, e ir ajustando los parámetros (escribiendo o moviéndonos sobre la gráfica), y por último calcular la probabilidad en el intervalo que se elija.

Figura 10. **Vista Calculadora de Probabilidad**



3. GEOMETRÍA A TRAVÉS DE GEOGEBRA

En este y en los siguientes epígrafes se van a desarrollar algunos ejemplos de la aplicación didáctica de GeoGebra para la docencia de las matemáticas en enseñanza primaria y secundaria.

Los ejemplos irán aumentando en complejidad, empezando por los más fáciles, para quienes no tengan ninguna experiencia con el programa, para más tarde ir introduciendo herramientas progresivamente. De todas formas, y debido a la envergadura del programa, no es posible abordar en profundidad todas las herramientas. Aconsejamos visitar las páginas web que se mencionan en la bibliografía, entre ellas el manual del programa (<http://wiki.geogebra.org/es/Manual>).

3.1. CONSTRUCCIÓN 1. RECTA PARALELA Y RECTA PERPENDICULAR

Dibujamos una recta. Para ello existen dos posibilidades:

a) Introduciéndonos en la barra de entrada y escribiendo la ecuación de la recta, por ejemplo, $y = 2x - 2$.

b) Introduciéndonos en la Ventana Gráfica con el botón Recta  .

   Una vez seleccionado, pulsaremos los dos puntos por los que pase la recta.

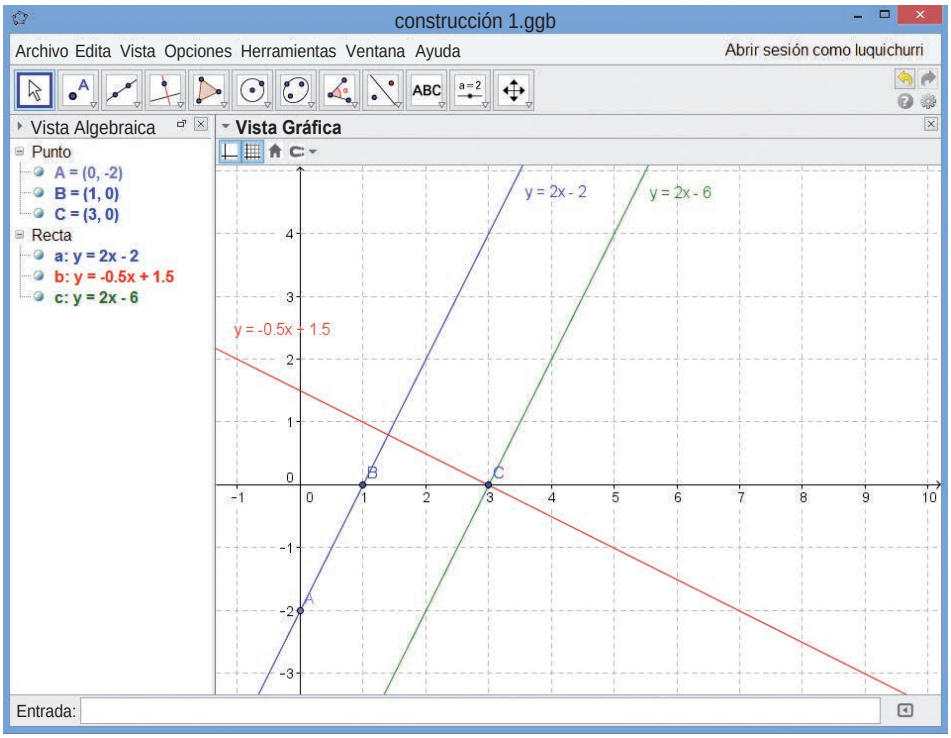
Ambas formas de construcción proporcionan la recta, pero la construcción que se obtiene es distinta, ya que la opción b) proporciona los dos puntos por los que pasa la recta y que podremos modificar en el área gráfica, modificándose a su vez la recta. De la otra forma, para modificar la recta tendremos que editar la ecuación de la recta desde la Vista Algebraica. Por tanto, es más versátil crearla como se ha realizado en el apartado b), ya que permitirá a los alumnos interaccionar con la construcción, modificándose los parámetros de dicha construcción.

Continuamos con la recta paralela, para ello seleccionaremos el botón Recta paralela  y a continuación la recta a la que tiene que ser paralela y el punto por el que ha de pasar (en verde en la figura 11).

Continuamos ahora con la recta perpendicular, para ello seleccionaremos el botón Recta perpendicular  y a continuación la recta a la que tiene que ser perpendicular y el punto por el que ha de pasar (en rojo en la figura 11).

Tras su realización debería quedar algo similar a lo que se muestra en la figura 11. Se pueden modificar las propiedades de cada elemento para cambiar el color, tamaño, estilo, el rotulo, etc. Los puntos A, B y C son dinámicos, es decir, se pueden mover y el resto de construcciones dependientes de ellos también se modificarán.

Figura 11. **Construcción 1, recta paralela y recta perpendicular por un punto**



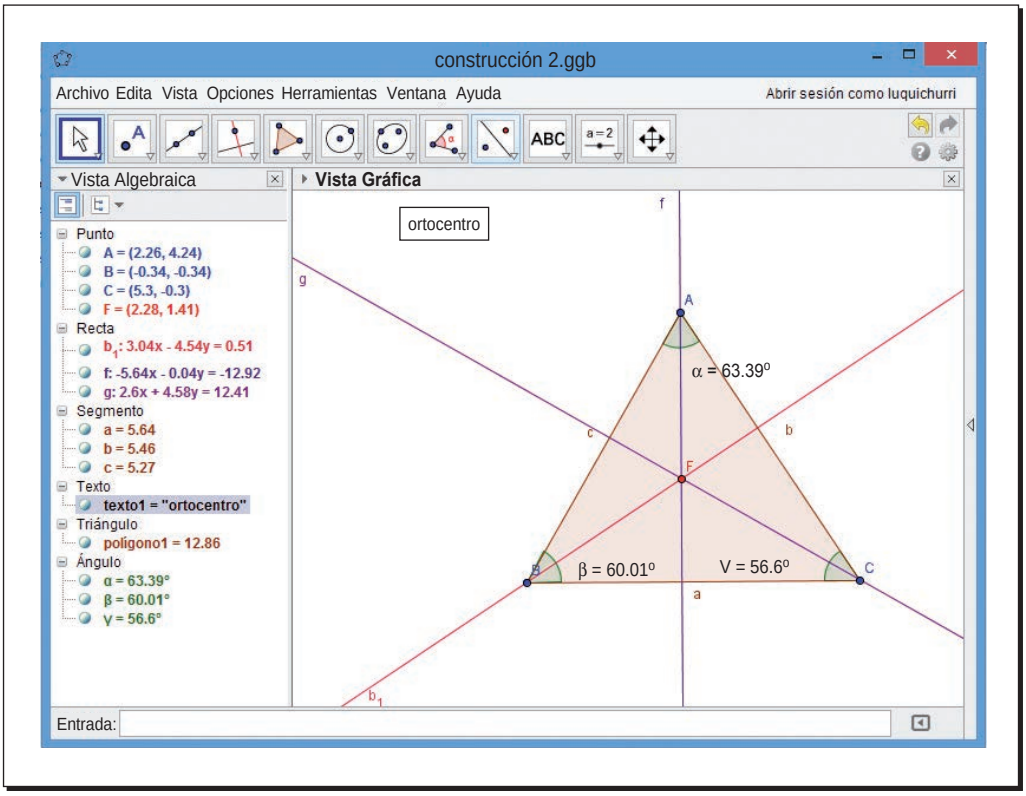
3.2. CONSTRUCCIÓN 2. TRIÁNGULO Y ORTOCENTRO

Introduciremos un triángulo mediante la herramienta Polígono, y definiremos los tres vértices que componen el triángulo.

A continuación, mediante la herramienta Recta perpendicular, trazaremos las rectas que van perpendiculares a los lados del triángulo y que van hasta el vértice opuesto.

Dibujaremos un punto en la intersección de las tres rectas. El punto de intercesión de las tres rectas es el baricentro, que no lo podemos desplazar ya que es un elemento dependiente, pero sí que podemos mover los puntos de los vértices, desplazándose todo la construcción.

Figura 12. **Construcción 2, ortocentro de un triángulo**

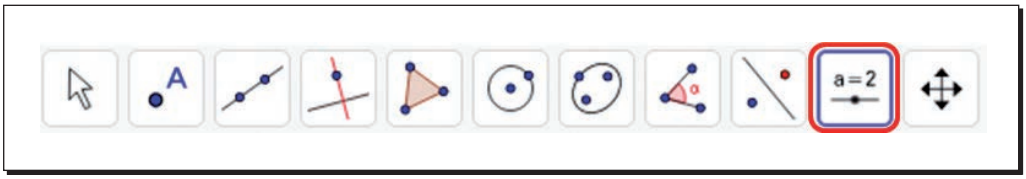


3.2.1. Deslizadores

Los deslizadores nos sirven para poder variar dinámicamente y de forma sencilla algún parámetro, de manera que podamos ver de forma rápida cómo influye dicho parámetro en algún objeto o función al que se asocie.

Vamos a ver estos deslizadores con algunos ejemplos de uso. El botón para introducir deslizadores es el que se muestra realzado en la figura 13.

Figura 13. **Barra de herramientas indicando botón de deslizador**



Una vez que se inserta el deslizador, tenemos que asociarlo a algún objeto que tengamos en nuestra construcción. Veamos un ejemplo para resolver sistemas de ecuaciones introduciendo deslizadores para variar los coeficientes de las rectas.

Primeramente trataremos de resolver el sistema de ecuaciones:

$$\begin{cases} 2x + 3y = 8 \\ -3x + 4y = 5 \end{cases}$$

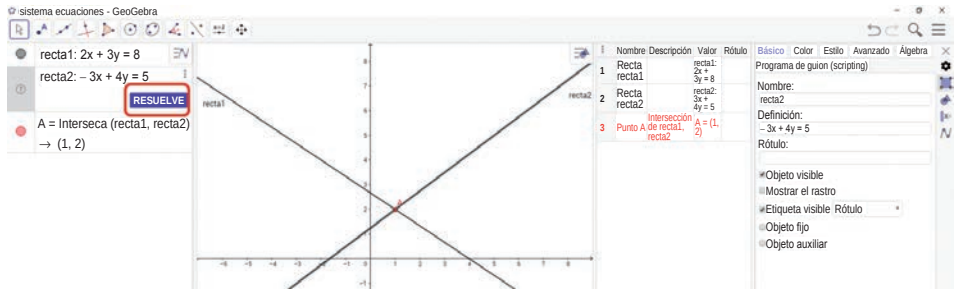
Posteriormente asignaremos los parámetros «a», «b», «c» y «d» a los coeficiente de la «x» y la «y» para ver cómo varía la solución del sistema de ecuaciones.

Para introducir la ecuación, deberemos ir a la barra de entrada e introducir primero una ecuación y posteriormente la segunda. Por defecto, les asignaremos un nombre a cada una de las ecuaciones, normalmente «a», la primera y «b», la segunda, pero podremos modificarlo haciendo doble click sobre el nombre y editándolo, para llamarlos, por ejemplo, «recta1» y «recta2». Podremos cambiar el color, grosor de la línea, etc.

Una vez introducidas las ecuaciones, vamos a ver cuál es el punto de intersección de ambas, es decir, el punto que cumplen ambas ecuaciones y que será el resultado del sistema de ecuaciones. Para ello introduciremos el comando Interseca (recta1, recta2), donde recta 1 y recta2 son los nombre que les hemos dado a las ecuaciones. También podremos hacerlo de forma parecida haciendo click sobre recta2 y presionando sobre el botón Resuelve (figura 14).

Introduciremos deslizadores para poder cambiar de forma dinámica los valores de coeficientes de la «x» y la «y» en cada una de las rectas. Para ello introduciremos cuatro deslizadores mediante el botón mostrado en la figura 13. Cuando los introduzcamos, debemos indicar la posición donde queremos que aparezcan. A continuación se abrirá una ventana emergente en la que introduciremos nombre, definiremos el intervalo con el valor mínimo, máximo y el incremento, dependiendo si elegimos que sea un número, un ángulo o un entero (figura 15).

Figura 14. **Solución de un sistema de ecuaciones con GeoGebra**



Fuente: <https://ggbm.at/brhC8ZvV>.

Una vez que hayamos introducido los cuatro deslizadores, deberemos a asociarlos a un objeto para que varíe cuando se desplace el deslizador. Para ello, en las ecuaciones que teníamos vamos a cambiar los coeficientes que multiplican a la «x» y a la «y» de cada una de las rectas por los nombres que hemos dado a los deslizadores.

Una vez realizado, podremos mover los deslizadores en la Vista Gráfica o Algebraica y el punto A, que es la solución del sistema de ecuaciones, también lo hará, con lo cual tendremos la solución del sistema de ecuaciones para cualquier valor que vayamos poniendo.

Figura 15. **Ventana para definir el deslizador**

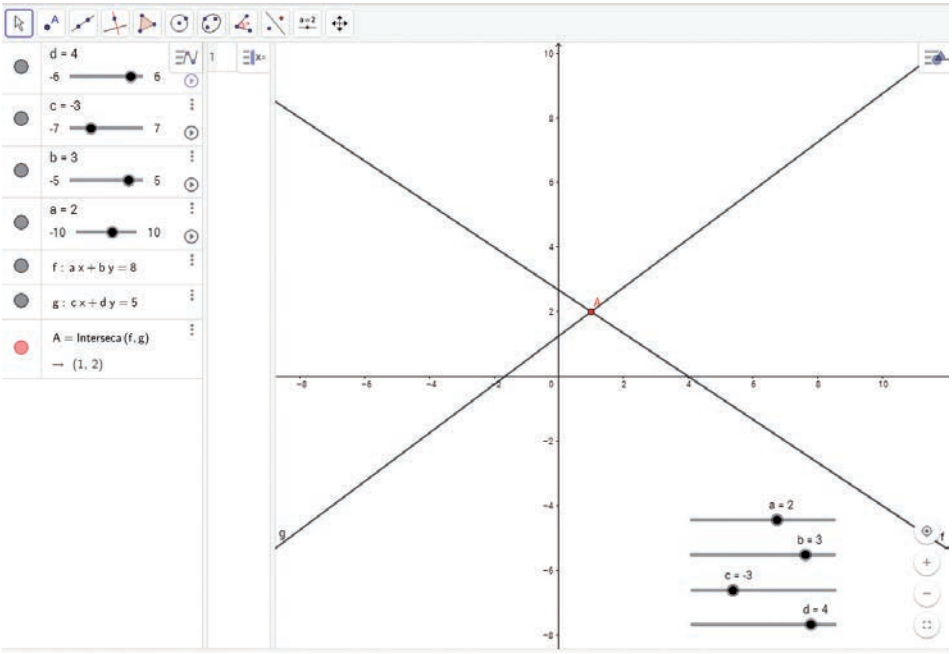
Nota. Si hubiésemos intentado introducir directamente la rectas como:

$$ax + by = 8$$

$$cx + dy = 5$$

GeoGebra habría detectado automáticamente que se pueden tratar de parámetros y nos preguntará si queremos crear los deslizadores para «a», «b», «c» y «d».

Figura 16. **Solución de un sistema de ecuaciones con GeoGebra**



Fuente: <https://ggbm.at/dWSzJ7Bk>.

4. PROGRAMACIÓN LINEAL CON GEOGEBRA

Normalmente, la resolución de los problemas de programación lineal consiste en maximizar o minimizar una función llamada *función objetivo* sobre una región del plano (solución de un sistema de inecuaciones lineales en dos variables), o del espacio (solución de un sistema de inecuaciones lineales en tres variables). A esta región se la denomina *región factible*.

La solución de estos problemas siempre se va a encontrar en alguno de los vértices de esta región.

5. ÁLGEBRA CON GEOGEBRA

Existen múltiples utilidades para el uso de GeoGebra aplicado al álgebra, como son las identidades notables, la factorización de polinomios, matrices, etc.

6. FUNCIONES CON GEOGEBRA

Dentro de las posibilidades que ofrece la nueva Vista CAS, tenemos varias operaciones que pueden resultar muy interesantes para su aplicación en el aula.

6.1. LÍMITES

Introduciremos en la primera fila la función f: = (x – 5)/x y conservamos la entrada ✓ Escribiremos el comando Límite[f,3] y con evalúa, = , obtendremos el límite de f cuando x → 3.

En la siguiente fila escribiremos Límite[f,∞] y obtendremos el valor cuando x → ∞.

También podremos calcular los límites superiores e inferiores (laterales), cuando se acerquen a un punto, con los comandos LímiteInferior[f, 0], LímiteSuperior[f,0].

Figura 17. **Ejemplo de límites**

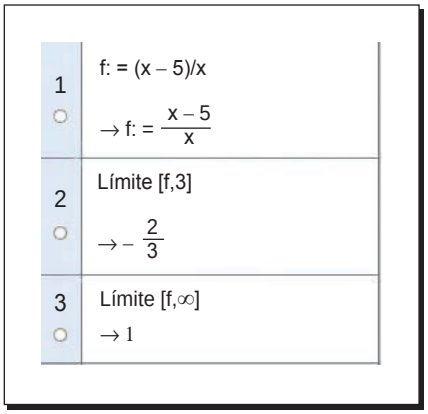
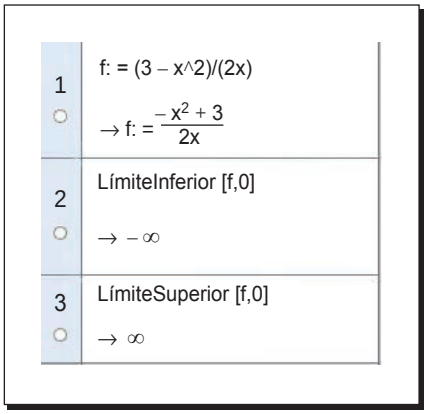


Figura 18. **Ejemplo de límites superiores e inferiores**



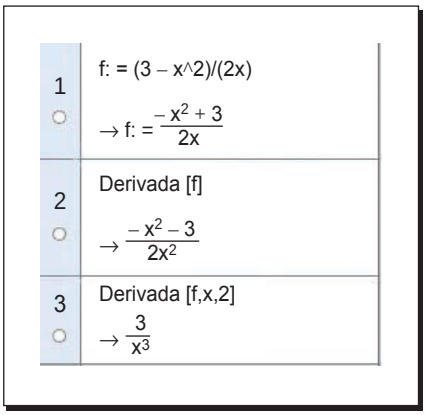
6.2. DERIVADAS

Para el cálculo de derivadas se introducirá una función y seleccionaremos Evalúa: f: = (3 – x^2)/(2x).

Para el cálculo de su derivada escribiremos el comando Derivada[f] y si lo que queremos obtener es la derivada de orden 2, escribiremos el comando Derivada[f,x,2].

Con este comando podremos realizar la derivada del orden que necesitemos cambiando el 2 por el orden que deseemos.

Figura 19. **Ejemplo de derivadas**



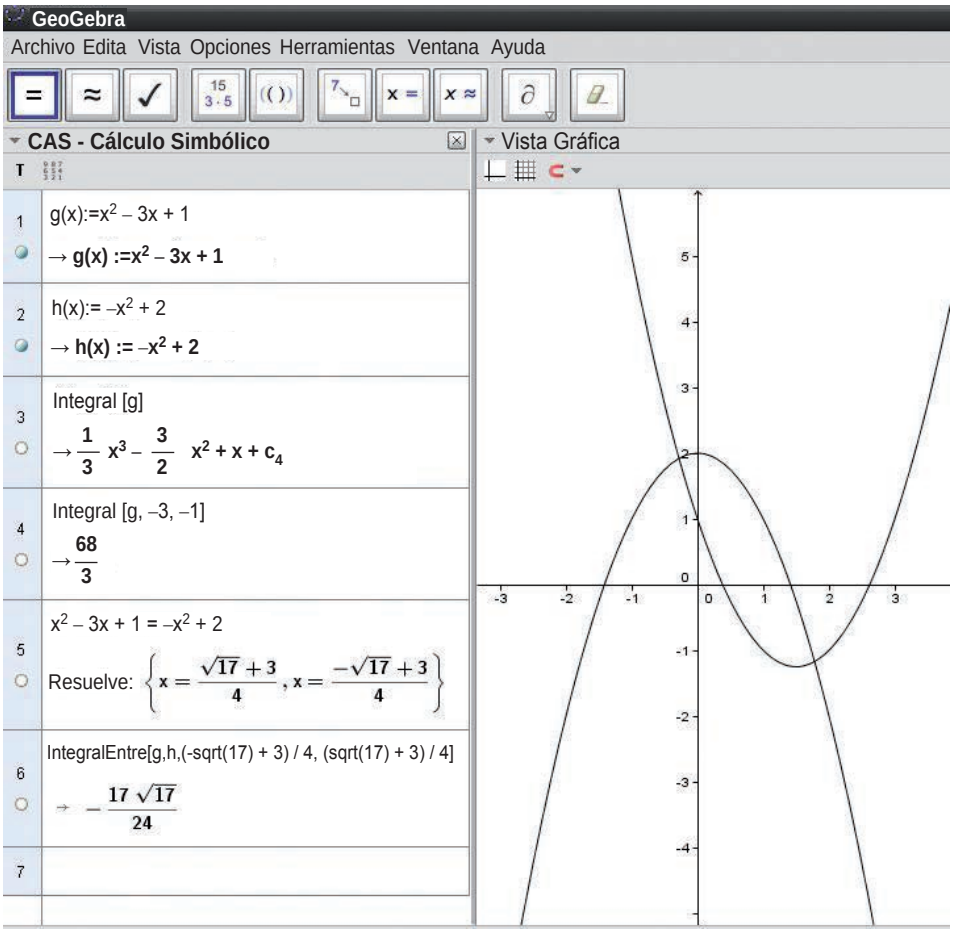
6.3. INTEGRALES

Dentro de los comandos propios de integrales, podremos realizar los siguientes:

- Integral[<Función>] → Devuelve la función primitiva de la función.
- Integral[<Función>, <Valor Inicial de x>, <Valor Final de x>] → Devuelve el valor de la integral definida entre dos valores de «x».

  IntegralEntre[<Función>, <Función>, <Valor Inicial de x>, <Valor Final de x>] → El resultado es, el área definida entre las dos funciones y los valores de «x».

Figura 20. **Ejemplo de integrales**



7. RESUMEN DE LOS COMANDOS PRINCIPALES DE GEOGEBRA

A continuación se resumen de los principales comandos usados en GeoGebra.

Tabla 2. **Principales comandos usados en GeoGebra**

Álgebra	CAS	Cálculo y Funciones	GeoGebra	Geometría
ComúnDenominador	FactorC	FactorC	AComplejo	Recta
Desarrolla	ResoluciónC	ResoluciónC	APolar	Interseca
Divisores	SolucionesC	SolucionesC		Longitud
División	Racionaliza	Racionaliza		PuntoMedio
EsPrimo	APunto	APunto		Tangente
FactoresPrimos	FraccionesParciales	FraccionesParciales		
Factoriza	IntegralN	IntegralN		
CompletaCuadrado	ResoluciónN	ResoluciónN		
AExponencial	SolucionesN	SolucionesN		
ListaDivisores	ValorNumérico	ValorNumérico		
Máximo	ProductoEscalar	ProductoEscalar		
SegundoMiembro	ProductoVectorial	ProductoVectorial		
PrimerMiembro	Resuelve	Resuelve		
Mínimo	Soluciones	Soluciones		
MCD	Sustituye	Sustituye		
MCM		PolinomioTaylor		
PrimoAnterior		Raíz		
PrimoSiguiente		RaízCompleja		
Cociente		ListaRaíces		
Resto		ResuelveEDO		
Simplifica		TrigCombina		
SumaDivisores		TrigDesarrolla		
		TrigSimplifica		

.../...

.../...

Listas	Lógica	Probabilidades	Guiones	Estadísticas	Vectores y matrices
Añade	Si	AleatorioEntre	Elimina	AjusteExp	Determinante
Aplana		BinomialAleatorio		AjusteLog	Dimensión
Invierte		NormalAleatorio		AjustePolinómico	Identidad
Elemento		PoissonAleatorio		AjustePotencia	Inversa
Elemento Aleatorio		nPr		Covarianza	EscalonadaReducida
Extrae		DistribuciónBinomial		DE	RangoMatriz
ListaPuntos		Cauchy		Comando DEMuestra	Traspone
Primero		NúmeroCombinatorio		Muestra	VectorNormal
Producto		Exponencial		Mediana	VectorUnitario
Último		DistribuciónF		Mezcla	VectorNormalUnitario
Secuencia		Gamma		Media	
Único		Hipergeométrica		Suma	
Encadena		ChiCuadrado		Varianza	
		Normal		VarianzaMuestra	
		Pascal			
		Poisson			
		PolinomioAleatorio			
		DistribuciónT			
		Weibull			
		Zipf			

8. OCTAVE. INTRODUCCIÓN

En ocasiones, las aplicaciones que se han visto en el presente manual y que nos facilitan el cálculo matemático, como son GeoGebra o MS Excel, pueden no resultar suficientemente flexibles para realizar los algoritmos que necesitamos, o en el procesado de grandes volúmenes de datos, para la generación de los mismos, o puede que carezcan de los procesos y de las fórmulas necesarias para resolver el problema con el que tratamos.

Esto suele suceder en materias como la ingeniería, pero también en otras áreas donde se requiere del análisis y la visualización avanzada de datos, como pueden ser en psicología, marketing, finanzas, etc.

Estos casos requieren del uso de *software* de simulación matemática. En la actualidad, existen muchas soluciones de este tipo, tanto soluciones *software* privativas con algún tipo de licencia, como soluciones de *software* libre. El caso más conocido es la suite de cálculo de la empresa MathWorks, llamado Matlab. Matlab es el conocido *software* de cálculo matemático y también el lenguaje de programación con el que se definen las operaciones y cálculos a realizar. La sintaxis con la que se trabaja en Matlab ha trascendido hasta el punto de ser considerado un lenguaje de programación en sí mismo a la altura de C/C++, Java, Python, etc.

Entre las opciones de *software* libre existen varios ejemplos, como pueden ser el propio Python o el motor de cálculo R (que cuenta con diverso *software* de apoyo, como puede ser RStudio), que últimamente están ganando importancia frente al lenguaje de programación de Matlab.

GNU Octave[1] es, de la misma forma que Matlab, un lenguaje de programación matemática de alto nivel, y además, también es el *software* o entorno de programación para interpretarlo, escribirlo y ejecutarlo para realizar los cálculos programados. Por lo tanto nos referiremos con el nombre de Octave tanto al entorno de programación como al lenguaje con el que se programa indistintamente.

El entorno de desarrollo generado por el proyecto GNU Octave es una buena opción de *software* libre para trabajar con este lenguaje de programación matemática, y su sintaxis es compatible en un alto porcentaje con el propio Matlab, puesto que la comunidad de programadores de GNU Octave lo ha diseñado de forma que se interpreten igual las operaciones y herramientas programadas con la sintaxis de Matlab y generen resultados similares. Esto le da a GNU Octave una ventaja competitiva importante frente a otras opciones, puesto que es posible reutilizar la inmensa base de conocimiento que tanto los programadores de Matlab como los de Octave han ido produciendo y compartiendo en línea. Asimismo, permite reutilizar el código creado entre uno y otro, aunque a veces sea necesario alguna pequeña adaptación o la instalación de librerías específicas en GNU Octave, debido a las limitaciones que imponen las licencias privativas de Matlab y sus paquetes de herramientas (o *ToolBoxes*).

9. INSTALACIÓN DE OCTAVE

La instalación de GNU Octave es sencilla y solo requiere la descarga del *software* para el sistema operativo deseado desde la página web del proyecto GNU Octave[2]. Una

1 <https://www.gnu.org/software/octave/>.

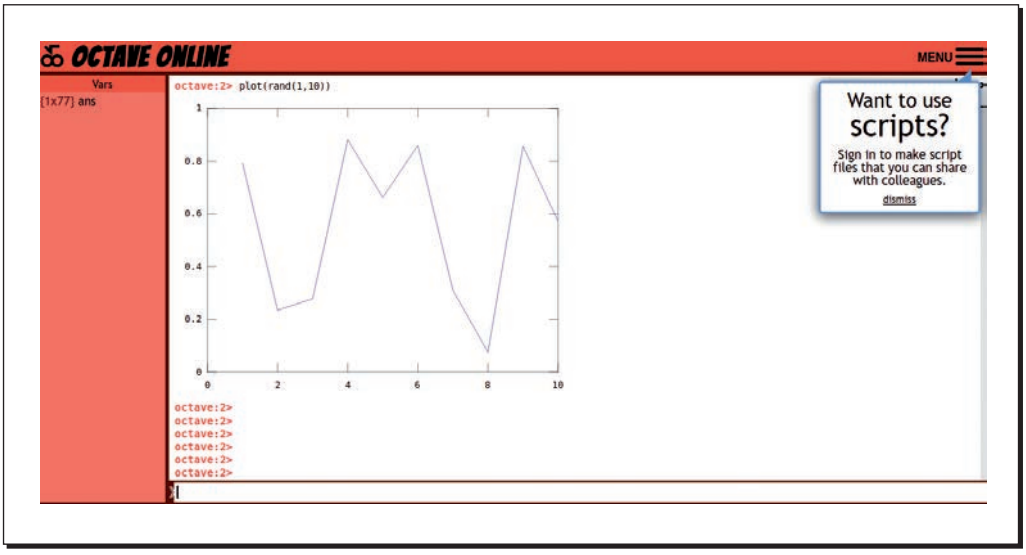
2 < https://www.gnu.org/software/octave/download.html>.

vez instalado en nuestro PC, dispondremos de un centro de trabajo como el de la figura 21, o similar (las versiones evolucionan bastante rápido).

Existen además opciones en línea desde donde poder ejecutar los comandos del lenguaje de Octave, como Octave-online.net[1], que nos permiten ejecutar comandos desde un navegador. Estas soluciones suelen estar limitadas a lo que podamos realizar en la nube, sin acceso al espacio físico del dispositivo desde donde las ejecutemos, pero son una buena alternativa para poder ejecutar código Octave en la nube. En el caso de Octave Online, por ejemplo, dispone de la posibilidad de crearse una cuenta en el sistema para poder guardar los ficheros de código (*scripts*) y poderlos ejecutar posteriormente.

En el caso de necesitar comandos o funciones más avanzadas o específicas de un área de conocimiento concreto, tanto GNU Octave como su homólogo con licencia Matlab, disponen de paquetes con aplicaciones, utilidades y funciones adaptados a las más diversas materias. Estos paquetes, que también se conocen como *toolboxes* por ser el nombre técnico utilizado en Matlab, son también de código abierto y pueden encontrarse en el repositorio central de Octave Forge[2].

Figura 21. **Imagen de la pantalla principal de la aplicación Octave Online**

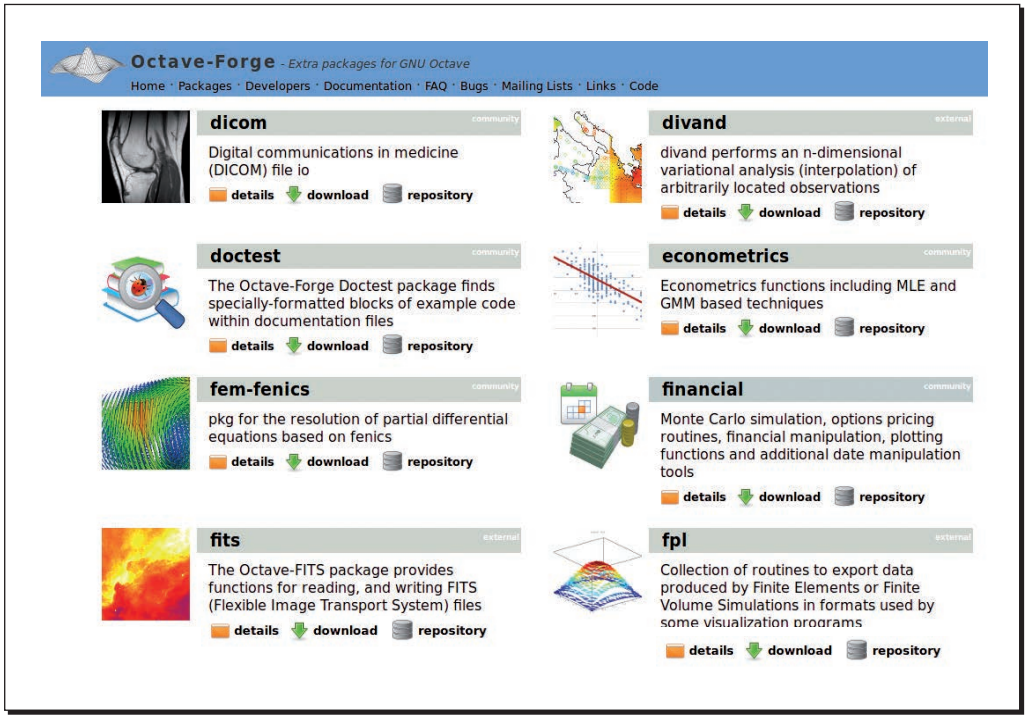


1 < https://octave-online.net>.

2 < https://octave.sourceforge.io/>.

La lista completa de paquetes disponibles para la versión con la que se esté trabajando, así como aquellos que hayamos podido instalar y configurar, puede obtenerse escribiendo en la línea de comandos: `pkg list` y para instalar algún paquete de la lista del proyecto Octave Forge, solo tendríamos que escribir: `pkg install -forge nombre_del_paquete`.

Figura 22. **Ejemplo de paquetes disponibles en Octave para áreas de conocimiento específicas**



10. EL ENTORNO DE TRABAJO

En el entorno instalado por defecto ya es posible comenzar a trabajar dado que cuenta con una gran cantidad de operadores, fórmulas y comandos predefinidos en la instalación estándar.

Las partes principales del entorno de trabajo son:

- **La ventana de comandos.** Es la ventana principal y es el lugar en el que vamos a ejecutar los comandos y operaciones que queramos realizar. Para usar la ventana de comandos solo tenemos que escribir el comando que queremos calcular o ejecutar y pulsar la tecla Enter. Inmediatamente aparece el resultado en la misma ventana de comandos, o en caso de no haberlo escrito bien, nos aparecerá la información asociada al error cometido. Por ejemplo, si escribimos `help --list`, debería aparecer como resultado todo el listado de operadores matemáticos, las palabras clave para crear algoritmos y los comandos disponibles en la instalación por defecto para poder ser utilizados.

  En cambio, si escribimos: `3*15`, aparecerá el resultado de esta operación matemática como: `ans = 48`.

- **El editor de código.** En Octave podemos trabajar de dos maneras: usando la línea de comando para ejecutar las funciones y operaciones deseadas, o generando *scripts*: ficheros de texto con las órdenes a ejecutar, a los que llamaremos en la línea de comando y que se grabarán en un fichero con extensión *.m*.

  Ambas formas son igualmente válidas, pero la segunda nos permite guardar las operaciones deseadas para poder ejecutarlas posteriormente y editar y corregir las órdenes fácilmente.

  En el editor de código es desde donde podemos escribir el código, sin ejecutarlo inmediatamente, para poder ejecutarlo posteriormente en bloque. Se ejecutan todos los comandos que haya en el fichero secuencialmente si no se genera ningún error. Al editor de código se accede pulsando la pestaña de la parte inferior de la ventana de comandos.

  Desde el propio editor de código tendremos acceso a su barra de herramientas, con opciones para ejecutar línea por línea y depurar el código, como ocurre en los compiladores de otros lenguajes de programación.

- **El explorador de archivos.** Desde donde podemos seleccionar el directorio de trabajo donde queremos situarnos. Esto nos permite acceder a ficheros con código previamente guardado para ejecutarlos o guardarnos los resultados que vayamos generando, así como importar datos desde ficheros al espacio de trabajo de Octave para procesarlos.

Mediante la barra del explorador y los botones situados en ella podemos cambiar de directorio de trabajo. Esto también podemos hacerlo escribiendo en la ventana de comando: `cd directorio_destino`, donde directorio_destino será el camino completo del directorio al que queremos cambiar, o el nombre de uno de los directorios contenidos en el directorio actual.

- **El espacio de trabajo.** Es una ventana en la que se van mostrando los diversos objeto (matrices en general) que se vayan creando cuando estemos trabajando en la ventana de comandos, junto con su tamaño y dimensiones.
- **El historial de comandos.** Es una ventana en la que van quedando guardados todos los comandos que vayamos ejecutando en la ventana de comandos. Esto permite volver atrás y ejecutar comandos escritos con anterioridad o copiar y pegar comandos ejecutados con anterioridad al espacio del editor de código para guardarlos para otra ocasión o para crear un fichero ejecutable o *script*.

Figura 23. **Entorno de trabajo de la aplicación GNU Octave**

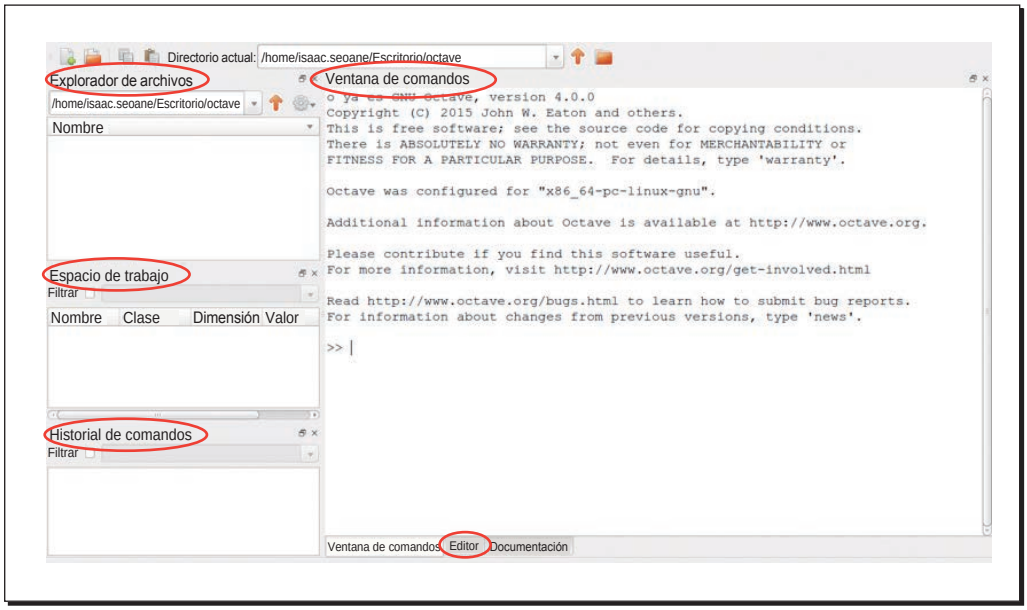
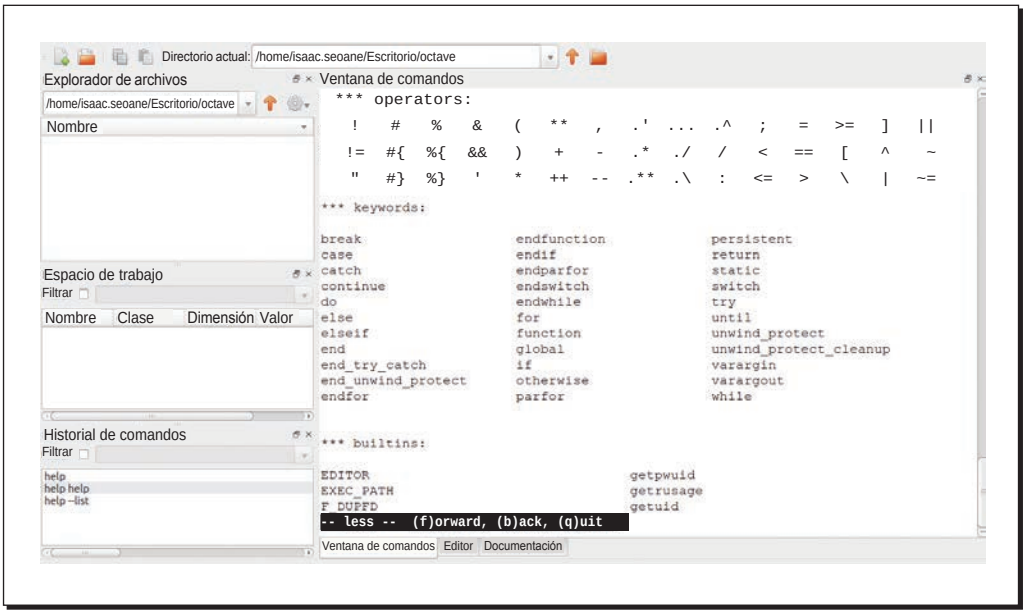


Figura 24. **Lista de operadores y funciones predefinidas**



11. PRIMEROS PASOS CON OCTAVE PARA NO PROGRAMADORES

El primer y más importante concepto que hay que tener en cuenta para trabajar con Octave es que la sintaxis y el motor de cálculo están orientados principalmente al cálculo matricial y el álgebra lineal. Para Octave, los datos introducidos se tratan como matrices de N dimensiones, a cuyas posiciones podemos acceder indistintamente o manejarlas de forma íntegra. Por lo tanto, permite diseñar incluso operaciones de cálculo más avanzadas que las que podamos realizar por escrito o con una hoja de cálculo con matrices de dos dimensiones (MxN – M filas x N columnas).

Las operaciones que escribamos en la ventana de comandos se calculan al pulsar la tecla Enter, siempre y cuando se haya escrito correctamente. En caso contrario, aparecerá la información correspondiente al error cometido. La única excepción serán aquellas líneas precedidas del símbolo %, que servirán como un comentario escrito que no se calculará ni se procesará.

Un número escrito en la línea de comando genera en el espacio de trabajo una variable matricial de tamaño 1x1 (1 fila x 1 columna).

```
>> 7
ans = 7
```

La indicación `ans` se refiere al último resultado que se ha calculado en el espacio de trabajo, y puede observarse en la ventana del espacio de trabajo que, al escribirse el número 7, se ha creado un objeto de nombre ans, que tiene como tamaño 1x1.

Si no queremos que aparezca este resultado almacenado en el objeto ans por cada línea que escribamos, podemos terminar la línea con el símbolo `;`. Esto hace que la línea se ejecute o calcule, pero no se muestre el resultado.

Lo habitual es ir guardando los resultados de las operaciones en variables, de forma que luego podamos recuperar los resultados parciales calculados en operaciones anteriores simplemente escribiendo el nombre de la variable donde se guardaron. Por ejemplo:

```
>> A=7;
>> B=2;
>> C=6;
>> D=A*B+C
   ans = 20
```

El ejemplo anterior crea en el espacio de trabajo cuatro objetos, denominados A, B, C y D de dimensiones 1x1, donde habremos ido almacenando los números 7, 2, 6 y 20, respectivamente.

A la hora de crear matrices de más de un elemento, tendremos que echar mano de estos símbolos:

`[]` → delimitan el contenido de una nueva matriz.

`;` → genera un salto de fila.

`,` → genera un salto de columna.

`:` → indica todos los elementos de la dimensión en la que aparezca.

`end` → indica el elemento final de la dimensión donde aparece.

Veamos algunos ejemplos a continuación.

En primer lugar, podemos crear matrices vacías, que nos servirán para reservar el nombre de un objeto y posteriormente añadirle datos.

```
>> V = [ ]; % crea una matriz vacía.
```

Crear un vector columna o un vector fila se puede conseguir escribiendo una serie de valores numéricos entre `[ ]`, separados por `;` o por `,` respectivamente:

```
>> C = [ 1 ; 3; 5; 7]; % almacena en C un vector columna
>> F = [ 1 , 3, 5, 7]; % almacena en F un vector fila
```

Tanto el vector fila como el vector columna son matrices de dos dimensiones, de tamaños [Mx1] y [1xN], respectivamente. En general, podremos crear matrices de dos dimensiones de tamaño [MxN] compuestas por valores numéricos o por otras matrices previamente creadas.

```
>> A = [ 1, 2 ; 3, 4]; % crea una matriz A cuadrada de dos dimensiones de tamaño [2x2]
>> B = [ 5, 6 ; 7, 8 ;9, 10]; % crea una matriz B de dos dimensiones de tamaño [3x2]
>> C = [ A ; B]; % crea una matriz C de dos dimensiones: 5 filas y 2 columnas [5x2] con los elementos de A y B yuxtapuestos.
>> D = [ A , B]; % genera un error debido a los tamaños de A y B
```

La última operación es una operación errónea, dado que no podemos crear una matriz usando como partida una matriz [2x2] y yuxtaponerla con una matriz [3x2], dado que las dimensiones no coinciden. Los errores más comunes que ocurren cuando programamos en Octave tienen que ver con errores cometidos con los tamaños de las dimensiones de las matrices.

12. MANIPULACIÓN DE MATRICES

Para poder manipular correctamente las matrices en Octave hay que prestar atención a sus tamaños y a las reglas de manipulación de matrices en álgebra lineal. Para poder

hacerlo correctamente hace falta utilizar comandos ya predefinidos en Octave que nos den información sobre las dimensiones de los objetos creados:

```
>> size(A) % calcula el tamaño de las dimensiones de a y las devuelve en un vector fila
ans =
       2   2
>> length(C)% calcula la longitud de un objeto vector, ya sea fila o columna
ans = 4
```

Otra operación habitual es la transposición de la matriz. La matriz transpuesta será una matriz en la que se han intercambiado filas por columnas respecto de la original. El operador de transposición se indica mediante el signo ' (comilla simple).

```
>> A
A =
       2
       4
>> A'
ans =
       1   3
       2     4
```

A los valores en cada una de las posiciones de una matriz se puede acceder indicando el valor de la posición en cada dimensión. Esto es, para acceder al primer elemento de la segunda columna de A escribiremos:

```
>> A(1,2)
   ans = 2
```

También podemos acceder a un rango de valores, para ello debemos usar el operador de Octave que hace referencia a un rango, similar a los rangos en otros sistemas de

cálculo, como por ejemplo en Excel, y que es el signo : Cuando le precede y le siguen valores, indicarán el principio y final del rango:

```
>> C(3:5,1:2)
ans =
       5   6
       7   8
       9   10
```

Cuando el signo : aparece solo, indica «todo el rango», o lo que es lo mismo, todos los valores de esa dimensión.

```
>> C(2,:)
ans =
       3   4
```

Una operación muy común es la generación de una secuencia de valores equiespaciados. Para ello escribimos en la consola de Octave:

```
>> t = [ -0.5: 0.01: 0.5];
```

Esto nos genera una matriz `t` cuyos valores son el cálculo de la serie aritmética de valores entre –0.5 y 0.5, contando de 0.01 en 0.01. Por lo tanto, la sintaxis para generar una serie de valores será:

```
>> t = [valor_inicial:tmaño_intervalo:valor_final];
```

También se puede usar la función *linspace*, cuya sintaxis es:

```
>> t = linspace (valor_inicial, valor_final, número_de_puntos)
```

linspace es una función que ya está integrada dentro del *software* de Octave, con lo que no es necesario programarla, sino simplemente usarla poniendo los valores ade-

cuados en sus parámetros de entrada. Para conseguir ayuda sobre una función podemos escribir en la línea de comando:

```
>> help linspace
```

13. OPERACIONES MATEMÁTICAS BÁSICAS

En Octave están predefinidos los operadores matemáticos más habituales. Podemos acceder a una lista de ellos escribiendo `help --list` en la línea de comandos. La lista de operadores matemáticos y lógicos que obtendríamos sería:

Figura 25. **Listado de operadores matemáticos y lógicos**

```
*** operators:
  !    #    %    &    (    **   ,    .'   ...  .^   ;    =    >=   ]    ||
  !=   #{   %{   &&   )    +    -    .*   ./   /    <    ==   [    ^    ~
  "    #}   %}   '    *    ++   --   .**  .\   :    <=   >    \    |    ~=
```

Estos operadores funcionan exactamente igual con escalares (matrices de un solo valor) que si de números escritos en una calculadora se tratase, como hemos visto antes. Pero en el caso de que estemos trabajando con matrices, los operadores actúan como operadores algebraicos matriciales. Por ejemplo, el operador * para multiplicar no se comporta igual con números escalares que con matrices:

```
>> n = 7;
>> m= 3;
>> m*n
   ans = 21
>> n*m
   ans = 21
```

```
>> A * B
   error: operator *: nonconformant arguments (op1 is 2x2, op2
is 3x2)
>> B * A
   ans =
         23   34
         31   46
         39   58
```

Como se puede ver en el ejemplo, el uso del operador de multiplicar con los números escalares almacenados en m y n permiten multiplicarlos directamente y funciona la propiedad conmutativa de la multiplicación. Sin embargo, al usarlo para multiplicar A y B, nos da un error cuando multiplicamos A*B, mientras que nos da un resultado correcto cuando lo hacemos en el orden contrario, B*A. Esto es debido a que en álgebra lineal es importante el tamaño de las matrices para poder multiplicarlas.

La multiplicación de matrices no consiste en la multiplicación de los elementos de una de ellas por los elementos de la otra. Consiste en que cada posición deseada de la matriz multiplicación, descrita por su fila y columna, será el resultado de la suma de la serie de valores que se obtienen al multiplicar cada uno de los elementos de la fila buscada en la matriz de la izquierda por cada uno de los elementos de la columna deseada en la matriz de la derecha del operador multiplicación. Esto genera una serie de multiplicaciones y sumas que complican el cálculo cuando aumenta el tamaño de la matriz, mientras que usando Octave es una multiplicación más.

Por lo tanto, la condición para poder multiplicar dos matrices del tamaño que sea es que el número de columnas de la matriz a la izquierda del operador sea igual que el número de filas de la matriz a la derecha del operador. Y el resultado será una matriz del número de filas de la primera y del número de columnas de la segunda. Esto es, las dimensiones de las matrices en una multiplicación han de ser:

$$[p \times M] * [M \times q]$$

donde «p» y «q» pueden ser dos números cualesquiera.

Por ejemplo, no podemos multiplicar dos vectores fila o dos vectores columna, pero sí podemos multiplicar un vector fila por un vector columna y viceversa y el resultado será el siguiente:

```
>> p = [ 1 3 5 7];
>> q = [ 2 4 6 8];
>> p*q error: operator *: nonconformant arguments (op1 is 1x4,
op2 is 1x4)
>> p*q'
  ans = 100
>> p'*q
  ans =
        2    4    6    8
        6   12   18   24
       10   20   30   40
       14   28   42   56
```

En el primer caso estamos multiplicando matrices de dimensiones [1x4] * [4x1], resultando una matriz de [1x1] y viceversa, en el segundo caso, multiplicamos matrices de dimensiones [4x1] * [1x4], resultando una matriz de [4x4].

En otras ocasiones, lo que buscamos es multiplicar uno a uno los elementos de ambas matrices. En este caso, podemos anteponer al operador el signo . Entonces, no estaremos operando según las reglas del álgebra matricial, sino elemento a elemento:

```
>> p.*q
  ans =
        2   12   30   56
```

Casos como el del operador multiplicación ocurren con otros operadores, por ejemplo / o ^ (división y potencia respectivamente) cuando de matrices se trata. Por lo que, en general, usaremos el operador de forma sencilla cuando queramos usar el álgebra matricial para calcular el resultado de dicho operador, o le antepondremos un . para indicar que queremos realizar las operaciones elemento por elemento. En este segundo caso hay que ser cuidadoso con los tamaños, y que las matrices implicadas tengan el mismo número de elementos y colocados en la misma distribución en cada una de las dimensiones de la matriz.

14. REPRESENTACIÓN GRÁFICA DE DATOS

Otras de las funcionalidades interesantes de Octave es la representación gráfica de grandes volúmenes de datos. Para ello podemos utilizar la función plot().

Este comando de Octave nos permite generar una gráfica en dos dimensiones mediante los datos almacenados en dos objetos: el primero será los puntos del eje «X» donde queremos marcar un punto y el segundo será el valor en el eje «Y» correspondiente a cada uno de los puntos del eje «X» del primer objeto. Veamos un ejemplo para ilustrarlo.

Supongamos que queremos representar la función $y = x^2 – 5x + 3$. Lo primero de todo es crear un vector con los valores del eje «X» donde queremos representar la función:

```
>> x=[-4:0.1:8];% dibujamos la función entre -4 < x <8
```

A continuación, calculamos los valores de «y» para los valores de «x» antes calculados:

```
>> y = x.^2-5*x+3;% es importante recordar poner .^ para calcular la potencia elemento a elemento
```

Y a continuación dibujamos la función con:

```
>> plot(x,y);
```

Para cambiar el color o el tipo de línea, podemos usar un tercer parámetro en el paréntesis:

```
>> plot(x,y,»r+-«) % dibuja la gráfica en color rojo y con una línea de guiones
```

Los colores y el tipo de línea se construyen mediante una terna de signos de la forma: «color, marcador, tipodelinea», de forma que la línea resultante tendrá el color indicado por una letra y se construirá mediante un marcador en el punto de coordenadas «X», «Y» correspondiente, unidas por el tipo de línea elegido.

Los posibles tipos de líneas, colores y marcadores son:

Tipos de línea:

- `-` → Línea continua (por defecto).
- `--` → Línea discontinua.
- `:` → Línea punteada.
- `-.` → Línea punto-guion.

Marcador:

- `+` → Cruz.
- `o` → Círculo.
- `*` → Estrella.
- `.` → Punto.
- `x` → Equis.
- `s` → Cuadrado.
- `d` → Diamante.
- `^` → Triángulo hacia arriba.
- `v` → Triángulo hacia abajo.
- `>` → Triángulo hacia la derecha.
- `<` → Triángulo hacia la izquierda.
- `p` → Pentágono.
- `h` → Hexágono.

Color:

- `k` → Negro.
- `r` → Rojo.
- `g` → Verde.
- `b` → Azul.
- `y` → Amarillo.
- `m` → Magenta.
- `c` → Cian.
- `w` → Blanco.

Si queremos etiquetar la gráfica resultante, podemos usar los siguientes comandos:

```
>> xlabel (" etiqueta eje X");
>> ylabel (" etiqueta eje Y") ;
>> title (" título del gráfico") ;
```

Si queremos poner una rejilla para marcar líneas de ubicación en el gráfico, podemos usar el comando `grid`:

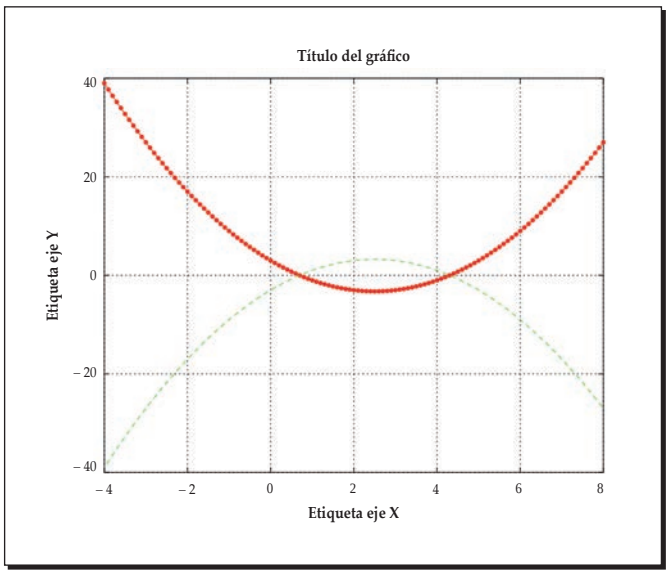
```
>> grid on % dibuja las líneas de la rejilla
>> grid off % las quita
```

Otra situación habitual es querer dibujar una gráfica superpuesta sobre otra. Para ello podemos usar el comando `hold`. Por ejemplo, en los mismos valores de «X» podríamos estar interesados en dibujar la función –y. Para ello escribiríamos:

```
>> hold on
>> plot(x,-y, 'g - -')
```

El resultado de todos los comandos anteriores sería la siguiente gráfica.

Figura 26. **Ejemplo de visualización gráfica de datos**



15. BUCLES Y CONDICIONALES PARA CREAR ALGORITMOS MÁS AVANZADOS

El uso de órdenes condicionales y bucles nos permite crear algoritmos más avanzados para nuestros cálculos. La sintaxis de un bucle `for` permite repetir una operación un número dado de veces, con la siguiente sintaxis:

```
for i = 1:10, % en la variable i marca el número de repeticiones, y el valor de i en cada vuelta
        .
            comandos a ejecutar
        .
        .
        .
endfor
```

Por ejemplo, supongamos que queremos ir guardando el cálculo de la función «y» del epígrafe anterior para una serie de valores variables del término independiente «k» en la forma:

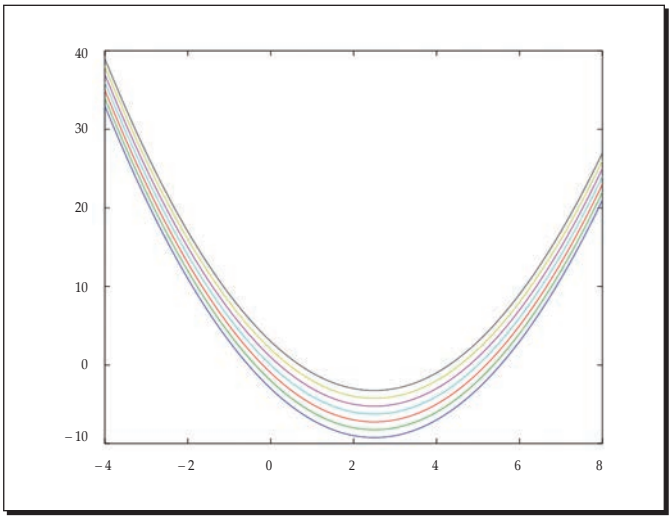
$$y = x^2 - 5x + k$$

$$-3 < k < 3$$

Para ello podríamos usar la siguiente secuencia de comandos:

```
ys=[];
for    k=-3:3,
       y=x.^2-5*x+k; % calculamos y con el valor de k que toque en cada vuelta
       ys=[ys;y]; % almacenamos la y calculada en esta vuelta
endfor
plot(x,ys') % dibujamos todas las gráficas obtenidas
```

Figura 27. **Gráfica obtenida con el ejemplo anterior**



De la misma forma, podemos hacer uso de secuencias de comandos condicionales, para elegir en función de si se ha cumplido una determinada condición. La sintaxis del comando condicional puede verse en el siguiente ejemplo:

```
x = 1;
if (x == 1) % condición de que deberá cumplirse
        disp ("one"); % comando y operaciones a ejecutar si se cumple
    elseif (x == 2) % en caso de no cumplirse puede añadirse otra condición
        disp ("two"); % comandos que se ejecutan si no se cumplió la primera pero sí la segunda
else
    disp ("not one or two"); % comandos que se ejecutan solo si no se cumplen las condiciones
endif
```

16. CREACIÓN DE *SCRIPTS*

Una forma habitual de usar el entorno de Octave es mediante el editor de código en lugar de directamente en la ventana de comando. Esto permite escribir todas las líneas que vamos a ejecutar en un mismo fichero, que será nuestro *script*, y que guardaremos dándole un nombre y que tomará extensión *.m*.

Un *script* por tanto no es más que un fichero de texto con una serie de comandos que deberán calcularse de forma secuencial para conseguir el efecto deseado. De esta forma, el uso de *scripts* nos permite hacer una reutilización eficiente del código anteriormente programado.

Por ejemplo, podríamos guardar los ejemplos del epígrafe anterior en un fichero llamado «ejemplo.*m*».

A continuación, en la ventana de comando, bastaría con escribir la palabra «ejemplo» (el nombre del *script* sin la extensión) para que el intérprete de Octave ejecutase todas las líneas una a continuación de la otra, llegando al mismo resultado que si hubiésemos escrito cada una de ellas una por una y pulsado la tecla Enter cada vez.

17. ÁLGEBRA CON OCTAVE

Como hemos visto, Octave está preferentemente indicado para cualquier operación de álgebra lineal como, por ejemplo, manipulación y operaciones con matrices bidimensionales, resolución de sistemas de ecuaciones lineales, inversión de matrices, cálculo de determinantes, etc.

Veamos un ejemplo de uso en el cálculo de operaciones de álgebra lineal. Sea la matriz «Q», vamos a realizar operaciones básicas como es la transposición, la inversa de la matriz y el cálculo de su determinante y sus autovalores.

$$Q = \begin{bmatrix} 0 & 5 & 4 & 1 \\ 6 & 2 & 1 & 6 \\ 9 & 5 & 7 & 6 \\ 4 & 7 & 4 & 1 \end{bmatrix}$$

```
>> Q'
   ans =
         0   6   9   4
         5   2   5   7
         4   1   7   4
         1   6   6   1
>> Q^-1
   ans =
         -0.282178   -0.056106   0.075908   0.163366
         0.064356   0.112211   -0.151815   0.173267
         0.108911   -0.194719   0.204620   -0.168317
         0.242574   0.217822   -0.059406   -0.193069
>> det(Q)
   ans = 606
>> eig(Q)
   ans = 16.6202   -7.0155   -2.0907   2.4859
```

También podemos utilizar Octave para resolver sistemas de ecuaciones lineales polinómicas. Para ello construimos una matriz con los coeficientes del polinomio y un vector columna con los términos independientes de cada ecuación:

$$3x + 4y - 6z = 1$$
$$4x + y = 2$$
$$-5y + 7z = -3$$

El código que resuelve este sistema de ecuaciones y con el que obtendremos los valores de las variables «x», «y» , «z», lo podemos construir de tres formas distintas: mediante el operador \, invirtiendo la matriz de coeficientes y multiplicando por la de términos independientes, o mediante el comando `linsolve`.

```
>> coefs=[3, 4, -6; 4, 1, 0; 0, -5, 7];
>> tindep= [1; 2; -3];
>> xyz=linsolve(coefs,tindep)
```

```
   xyz =
         -0.24138
         2.96552
         1.68966
>> xyz=coefs^-1*tindep
   xyz =
         -0.24138
         2.96552
         1.68966
>> xyz=coefs\tindep
   xyz =
         -0.24138
         2.96552
         1.68966
```

18. FÍSICA Y ELECTRÓNICA CON OCTAVE

En materias como la física y la electrónica es habitual el uso de sistemas de ecuaciones lineales o de ecuaciones diferenciales como herramientas para la resolución de problemas. En ese caso, Octave nos puede ayudar tanto como sistema de cálculo algebraico, como para crear un análisis del resultado de un problema cuando tengamos un dato variable, y para otras muchas aplicaciones.

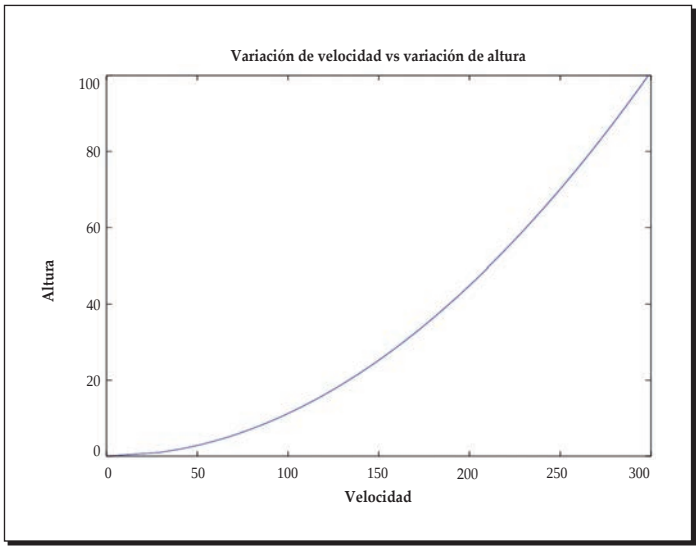
Por ejemplo, podríamos estar resolviendo un problema de balance energético en el cual un elemento móvil transforma su energía potencial en cinética. Este equilibrio se puede calcular como:

$$E_p = mgh = \frac{1}{2} m v^2 = E_c$$

Si dibujamos la gráfica de la velocidad máxima que conseguimos en función de la altura inicial del objeto, tendríamos una gráfica similar a esta, generada con el siguiente código:

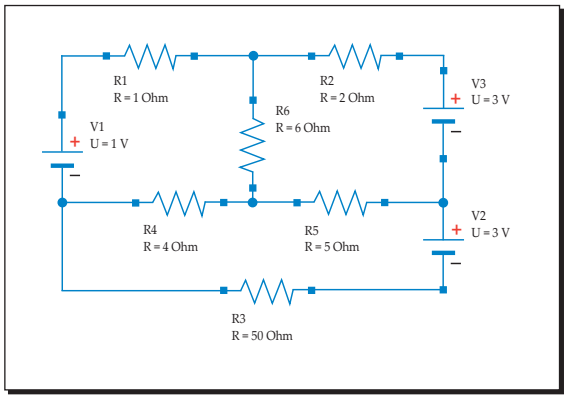
```
h=[0:100]; % serie de alturas iniciales
m=10; % masa del objeto
g=9.8; % gravedad en m/s^2
Ep = m * g * h; % serie de energías potenciales en función de las alturas iniciales
v=sqrt(2^g*h); % fórmula de la velocidad final del objeto al caer desde la altura inicial
Ec = 0.5 * m * v^2; % serie de energías cinéticas finales en función de las velocidades calculadas

plot(v,h); % dibujar la altura frente a la velocidad
title("variación de velocidad vs variación de altura");
xlabel("velocidad");
ylabel("altura")
```

Figura 28. **Gráfica de velocidad final en función de altura inicial**



De la misma forma, y sin salirnos del ejemplo de la materia de física, podríamos usar el ejemplo de la resolución de circuitos eléctricos. En ese caso, Octave puede ayudarnos a resolver los sistemas de ecuaciones que se generan en el análisis de circuitos, las variaciones respecto a alguno de sus componentes y la visualización de los efectos de los componentes sobre las entradas y salidas del circuito.

Figura 29. **Circuito de ejemplo**



El circuito de la figura puede resolverse utilizando las Leyes de Kirchhoff y un análisis por mallas para conformar el siguiente sistema de ecuaciones lineales, cuyas incognitas serán las corrientes de malla y por medio de las cuales podremos calcular las caídas de tensión en cada componente y las corrientes de cada rama.

$$11I_1 + 4I_2 + 6I_3 = 1$$
$$4I1 + 12I_2 - 5I_3 = 2$$
$$4I_1 - 5I_2 + 13I_3 = 3$$

Resolver el sistema de ecuaciones consistirá en separar este sistema de ecuaciones en su forma matricial, de forma que tengamos:

$$I = R^{-1} * V$$

Donde «V» es un vector columna con el valor de los tres generadores de tensión continua, «R» es la matriz inversa de las constantes que multiplican a las corrientes incógnitas, e «I» es un vector columna con las tres corrientes incógnita I^1, I^2 e I^3. El código sería el siguiente:

```
>> R = [11, 4, 6; 4, 12, -5; 4, -5, 13];
   >> V= [1;2;3];
>> I = R^-1 * V
   I =
         -0.41477
         0.54094
         0.56644
```

19. ESTADÍSTICA Y PROBABILIDAD CON OCTAVE

Finalmente, cabe reseñar que Octave nos puede ayudar en el análisis estadístico gracias a la gran cantidad de funciones predefinidas que contiene en su paquete básico, así como en los paquetes avanzados para esta materia en especial.

Una de las funcionalidades más interesantes es la posibilidad de generar números aleatorios siguiendo una distribución concreta. Por ejemplo, la función `rand`(m,n) permite la creación de una matriz de MxN posiciones, con valores aleatorios entre 0 y 1, siguiendo una distribución uniforme:

```
>> x=rand(3,3)
   ans =
         0.691668   0.962390   0.132944
         0.450171   0.785747   0.025230
         0.671864   0.547752   0.382998
```

Con las series de datos de que dispongamos, podemos calcular sus estadísticos de primer y segundo orden, por ejemplo, su media y su varianza con los comandos `mean`() y `var`(), respectivamente.

```
>> mean(x(:))
   ans = 0.51675
>> var(x(:))
   ans = 0.092224
```

CONCEPTOS BÁSICOS

De esta unidad didáctica es imprescindible conocer los siguientes conceptos clave:

- Comando.
- Deslizador.
- Determinante.
- GeoGebra.
- GeoGebraPrim.
- Matriz.
- Octave.
- Operador.
- *Script.*
- Transpuesta.
- Vector.
- Vista Gráfica.
- Vista Gráfica 3D.
- Vista Protocolo de Construcción.
- Vista Algebraica.
- Vista CAS.
- Vista Hoja de Cálculo.
- Utilización.

ACTIVIDADES DE REPASO

1. Inicie el programa. Puede optar por una de estas opciones

a) Instale GeoGebra en un ordenador o tableta a partir del enlace: <http://www.geogebra.org/download>.

Siga los pasos que vaya dando la web y verifique que se ha realizado correctamente.

b) También puede abrirlo directamente a través de la web, pero no podrá trabajar con GeoGebra si no tiene conexión.

2. Cree un triángulo inscrito en un círculo de radio 2 y con centro en el origen.

3. Repita el ejercicio anterior pero creando un deslizador que permita variar el radio del círculo entre 0 y 10.

4. Cree con GeoGebra la construcción de una parábola $y = a\ x^2 + b\ x + c$, de forma que los parámetros «a», «b» y «c» sean modificables dinámicamente mediante deslizadores cuyo valor pueda variar entre 0 y 10 en intervalos de 1.

5. Cree un recurso que permita obtener el área y el perímetro de un polígono regular de «n» lados y longitud de lado «m». Tanto «n» como «m» deben construirse con deslizadores.

6. Construya el siguiente recurso:

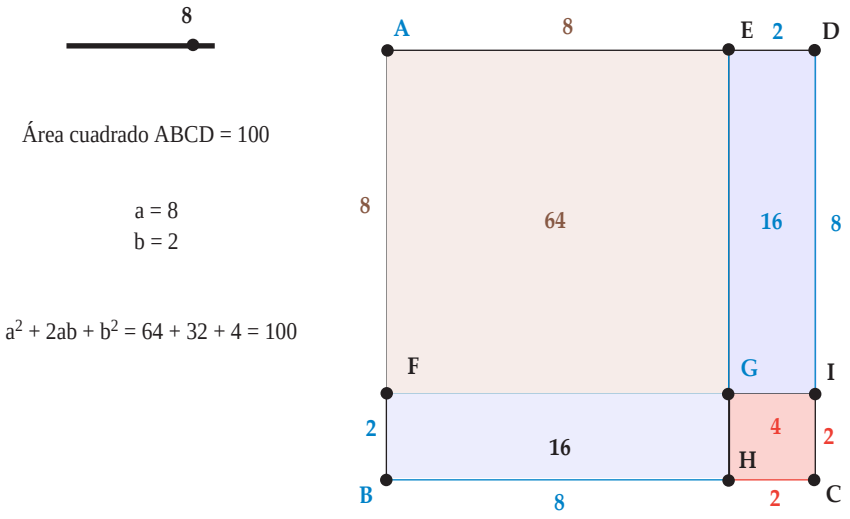
a) Cree un cuadrado (ABCD) de lado 10.

b) Cree un deslizador que se desplace de 1 a 9, con incrementos unitarios.

c) Divida cada lado del cuadrado ABCD, de tal manera que un segmento tenga el valor del deslizador, y el otro tenga el valor de (10 – valor del deslizador).

d) Trace los cuadrados y los rectángulos internos de acuerdo con la figura anterior.

e) Trace el texto dinámico como aparece en la siguiente figura:



f) Guarde el archivo.

7. Instale GNU Octave. Descargue el fichero de instalación adecuado para el sistema operativo de que se disponga.

8. Ejecute en la línea de comandos los ejemplos de código descritos en la unidad didáctica.

9. Cree cada ejemplo en un *script .m* diferente. Pruebe a ejecutar directamente los *scripts* creados y compruebe que se obtiene el mismo resultado que en los ejemplos de la unidad didáctica.

REFERENCIAS BIBLIOGRÁFICAS

En la red

<http://wiki.geogebra.org/es/Manual>.

<http://geogebra.es/cvg/manual/index.html>.

<http://www.jupenoma.es/igg2011/Taller_GG4_1.pdf>.

<https://www.academia.edu/1131388/Uso_de_Deslizadores_en_Geogebra>.

<https://www.academia.edu/1149171/Manual_B%C3%A1sico_de_Geogebra_2010>.

<https://www.academia.edu/1131405/Manual_Geogebra_nivel_intermedio>.

<https://www.gnu.org/software/octave/>.

<https://www.gnu.org/software/octave/download.html>.

<https://octave-online.net>.

<https://octave.sourceforge.io/>.

<http://www.linuxhispano.net/cursos/aprende-octave-con-ejemplos/>.