

# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

---

## Tercera sesión

### Circunferencias tangentes a tres rectas dadas

1. Se plantea el problema de construir una circunferencia tangente a tres rectas dadas  $r$ ,  $s$  y  $t$ . Usaremos lo aprendido. El centro de la circunferencia buscada está en el lugar geométrico de los centros de las circunferencias tangentes a  $r$  y a  $s$  y en el correspondiente para  $s$  y  $t$ . ¡Encontrémoslo!
2. ¿Cuántas circunferencias tangentes a las tres rectas dadas hay? ¿Cuál es el lugar geométrico de sus centros?

**Observación.** A estas alturas ya estamos en capacidad de plantear y resolver muchos problemas más sobre rectas y circunferencias tangentes. Lo podemos hacer usando lo aprendido y combinando el número de rectas y el número de circunferencias. Un problema histórico consiste en construir todas las circunferencias tangentes a tres circunferencias dadas. Es el problema de las circunferencias de Apolonio. Un caso particular de este problema es cuando las circunferencias dadas son tangentes entre sí. Los centros de las circunferencias tangentes se conocen como los centros de Soddy del triángulo formado por los centros de las circunferencias dadas.

### Bibliografía.

**Ministerio de Educación Nacional** (2002). *Seminario de Formación de Docentes: Uso de Nuevas Tecnologías en el Aula de Matemáticas*. Serie Memorias, p.p. 312,313.

---

Programación gráfica utilizando la calculadora TI - 92 plus (3 sesiones)

**Efraín Alberto Hoyos Salcedo**

Universidad del Quindío

**Nivel.** Intermedio

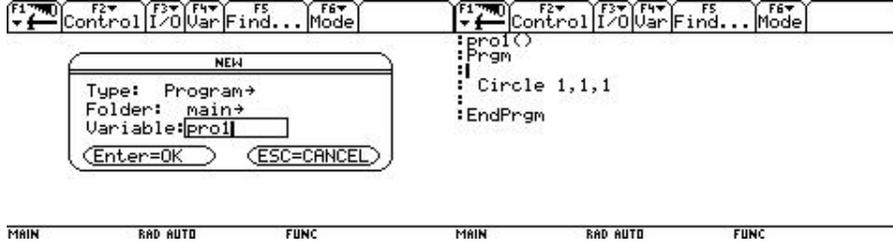
### Objetivos.

- Adquirir algunos de los elementos básicos de programación en la calculadora TI 92.
- Graficar funciones y hacer construcciones geométricas mediante la programación de la calculadora.

Descripción general del taller . Escritura de un programa sencillo en la calculadora TI 92 . Presentación de los elementos básicos de programación. Realización de gráficas utilizando las primitivas y gráficas con la primitiva línea. Transformaciones geométricas: Translación - Rotación - Reflexión. Utilización de las instrucciones de alto nivel para graficar funciones. Animación haciendo gráficas desde un menú de opciones.

**Conocimientos previos.** Conocimientos básicos de geometría y manejo básico de la TI 92 .

**Desarrollo del taller.**



al:  
 de Matemáticas

Sesión 1

**Cómo escribir un programa sencillo en la calculadora TI 92**

Para crear un programa nuevo en la calculadora TI 92, debe usar el editor de programas, el cual se activa presionando la tecla APPS. Se escoge la opción 7 Program Editor y se selecciona la alternativa 3, como se indica en la figura 1.



Figura 1

Cada programa debe tener un nombre, el cual se asigna al frente de la palabra variable como se muestra en la figura 2.

Figura 2

Al presionar ENTER dos veces, la calculadora dispone el editor, y en forma automática presenta el encabezamiento del programa y el fin del mismo como se ve en la figura 3.

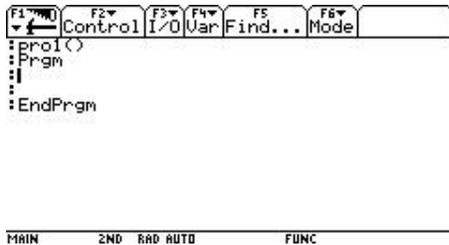
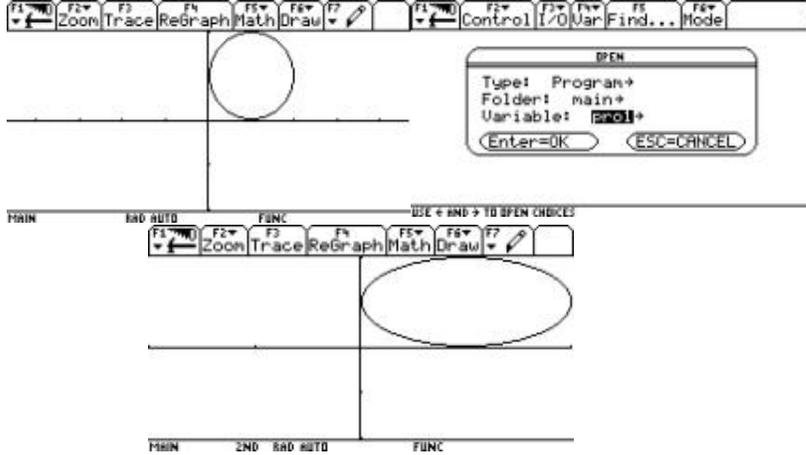


Figura 3

Observe que el cursor queda exactamente en el lugar adecuado donde debe escribirse la primera instrucción del programa que se desea hacer. En este momento se introduce la primera instrucción del programa, como por ejemplo la empleada para hacer un círculo, tal como se muestra en la figura 4.

Figura 4

El anterior es un programa que ya se puede ejecutar y al hacerlo presenta un círculo en la pantalla gráfica. Para ejecutar este programa debe ir a la pantalla HOME y en la parte inferior escribir el nombre del programa seguido de paréntesis como se indica en la figura 5.



## adicional: rículo de Matemáticas

Figura 5

Observe que la apariencia de la figura es la de una elipse. Para corregir el efecto visual use la instrucción ZoomSqr antes de dibujar el círculo y el programa quedaría como el ilustrado en la figura 6. Ejecute nuevamente el programa desde la pantalla HOME y observe cómo mejora la apariencia del

círculo.

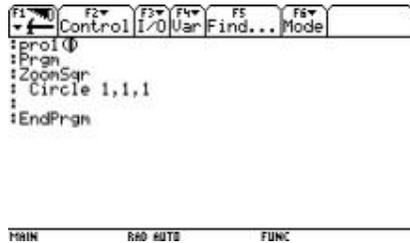


Figura 6

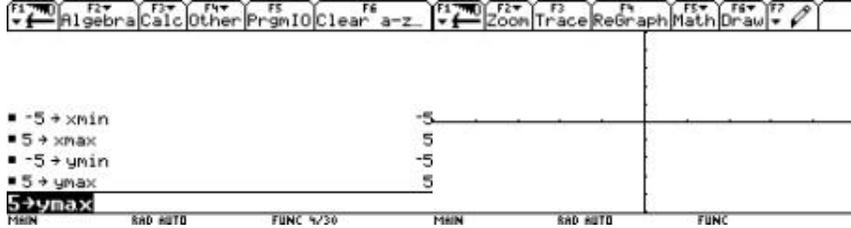
Es de anotar que la salida de un programa que muestra gráficas se efectúa en la pantalla GRAPH. Por otro lado, la calculadora no acepta un nombre de programa ya existente en la memoria.

Puede abandonar el editor de programas en cualquier momento, como por ejemplo, cada vez que vaya a ejecutar el programa lo cual se hace desde la pantalla HOME.

Los programas almacenados en la calculadora pueden abrirse desde el editor de programa con la opción open así como se muestra en la figura 7.



Figura 7



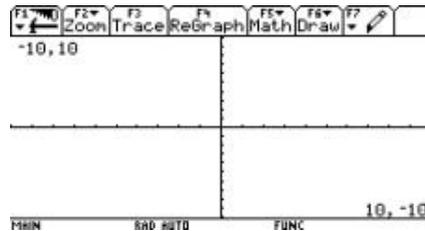
## ional: ulo de Matemáticas

### Algunas consideraciones

#### generales de la pantalla GRAPH

Esta pantalla tiene 239 pixeles de ancho por 103 de alto, los cuales se reparten en forma proporcional a la ventana que configure el usuario. La configuración estándar de la pantalla

GRAPH se puede observar en



la figura 8.

Figura 8

Como se puede apreciar, la calculadora dispone por defecto, de una ventana gráfica con un sistema cartesiano que considera 10 unidades hacia la derecha, 10 a la izquierda, 10 hacia arriba y 10 hacia abajo; claro está que el usuario puede cambiar dicha ventana de varias formas. Una manera es cambiando los valores de las variables que la controlan ( $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ). Por ejemplo, configuremos una ventana de 5 unidades hacia arriba, 5 hacia abajo, 5 hacia la derecha y 5 hacia la izquierda, así como se muestra en la figura 9.

Figura 9

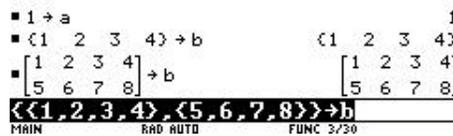
## Sesión 2

### Elementos básicos de programación

La programación de la calculadora TI 92 permite manejar tres clases de variables:

- **Variables del sistema** (no se pueden renombrar), por ejemplo  $x_{max}$ ,  $x_c$ .
- **Variables de carpeta**, se crean en el programa y permanecen en la memoria, incluidas en la carpeta donde se encuentra el programa. Se crean como en el siguiente ejemplo:  $4 > d$ .
- **Variables locales** son creadas con la orden "local" y no se almacenan en la memoria. Esto quiere decir que después de terminada la ejecución del programa, las variables desaparecen. Por ejemplo, para crear las variables locales  $f$ ,  $j$ , suma, se escribe: Local  $f, j, suma$

Cualquier lenguaje de programación permite manejar unas estructuras de datos para almacenar los datos y unas estructuras de control para determinar lo que se debe hacer con dichos datos. Las formas más comunes para almacenar datos son las siguientes:



## Congreso Internacional: Computacionales en el Currículo de Matemáticas

- almacenar un número en una variable

- almacenar un arreglo en una variable

Veamos los ejemplos en la figura 10:

Figura 10

Las principales estructuras de control son las siguientes:

If  
 If ... then... endif  
 If ... then... else ... endif

Goto eti

Lbl eti

For ... endfor

While endwhile

Loop ... endloop

Exit permite la salida de un bucle.

Cycle transfiere el control del programa a la siguiente repetición del bucle.

### **Realización de gráficas utilizando las primitivas.**

Aunque la primitiva gráfica es la instrucción para pintar un punto, también se pueden considerar como primitivas gráficas las instrucciones para dibujar una línea recta y un círculo.

### **Instrucciones para activar un punto en pantalla.**

Pton x, y . Pxlon x, y (activa el punto (x,y) en la ventana configurada)

Ptchg x, y. Pxlchg x, y (activa el punto (x,y) en la ventana configurada)

Veamos algunos ejemplos de programas que dibujan gráficas realizadas con puntos (se trabajará con la ventana: xmin = -5, xmax = 5, ymin = -2, ymax = 2):

Una recta (Figura 11)

# Tecnologías Comp

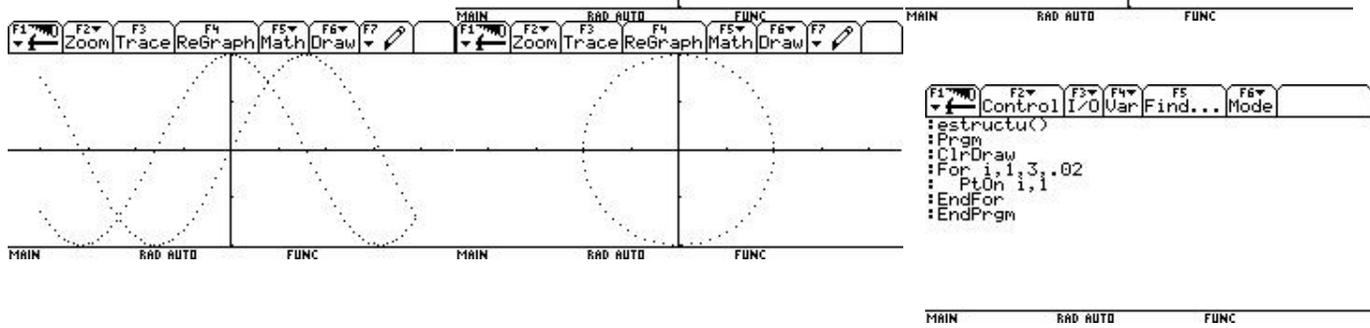


Figura 11

Puntos aleatorios (Figura 12)

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode
:ale()
:Prgm
:ClrDraw
:For i,1,300
:  PtOn rand(),rand()
:EndFor
:EndPrgm
  
```

Figura 12

Circunferencia (Figura 13)

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode
:circun()
:Prgm
:ClrDraw
:Local x,y,t
:0→t
:While t<2*π
:  2*sin(t)→x
:  2*cos(t)→y
:  PtOn x,y
:  t+.1→t
:EndWhile
  
```

Figura 13

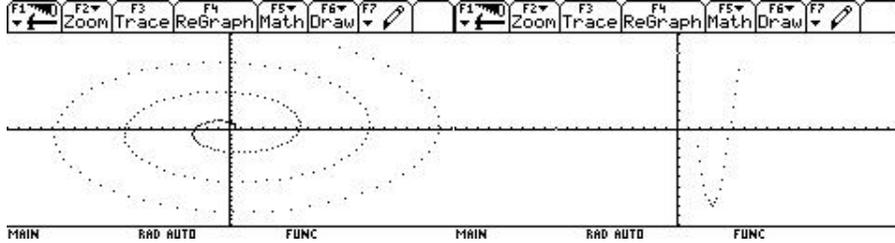
Gráfica de las funciones seno y coseno simultáneamente (Figura 14)

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode
:tri()
:Prgm
:Local t,f1,f2
:ClrDraw
:-4→t
:While t<4
:  2*sin(t)→f1
:  2*cos(t)→f2
:  PtOn t,f1
:  PtOn t,f2
:  t+.1→t
:EndWhile
  
```

Figura 14

Curvas Paramétricas (Figura 15)



al:  
de Matemáticas

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
: Prgm
: Local t,x,y
: -20→xmin:20→xmax:-20→ymin:20→ymax
: ClrDraw
: 0→t
: While t<20
:   t*cos(t)→x
:   t*sin(t)→y
:   PtOn x,y
:   t+.1→t
: EndWhile
: EndPrgm
MAIN RAD AUTO FUNC

```

Figura 15

Movimiento de varios puntos en la trayectoria de la función seno (Figura 16)

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
: Local t,x,y
: -20→xmin:20→xmax:-10→ymin:10→ymax
: ClrDraw
: -20→t
: While t<20
:   8*cos(t)→y
:   PtOn t,y
:   8*cos(t-4)→y
:   PtOff t-4,y
:   t+.2→t
: EndWhile
: EndPrgm
MAIN RAD AUTO FUNC

```

Figura 16

### Gráficas con la primitiva línea

Hay varias instrucciones que permiten realizar gráficas de líneas recta, entre ellas tenemos:

Line x1, y1, x2, y2[, m] traza una recta desde el punto (x1,y1) hasta el punto (x2,y2)

Linehorz y[, m] traza una línea horizontal por el valor de y

Linevert x[, m] traza una vertical por el valor de x

Si m = 1 dibuja la recta por omisión

Si m = 0 desactiva la recta

Si m = -1 activa o desactiva la recta según el estado inicial.

Drawslp x1, y1, p traza una recta que pasa por el punto x1,y1 y tiene pendiente p

Linetan expresión1, expresión2 traza una recta tangente a la expresión1 en el punto dado por expresión2.

Pxlline f1, c1, f2, c2[, m] opera lo mismo que line pero no lo hace sobre la ventana sino sobre los pixeles de la pantalla.

Veamos un ejemplo de gráfica de una de recta. ( Figura 17)

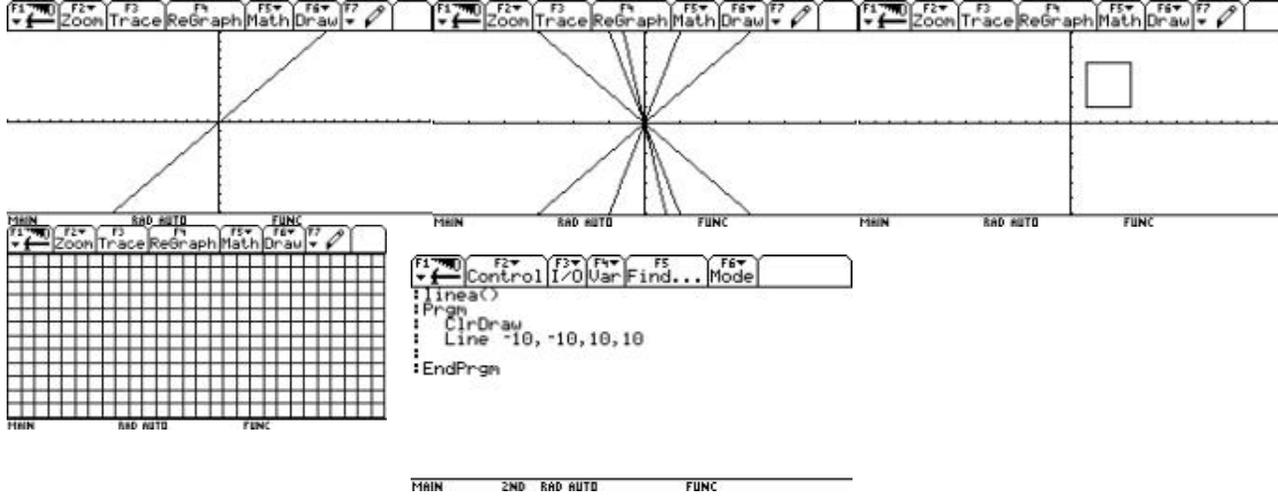


Figura 17

Ahora veamos varias rectas de diferente pendiente. (Figura 18)

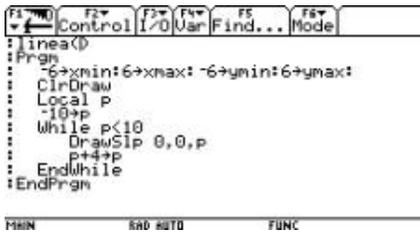


Figura 18

El programa de la figura 19 grafica una poligonal cerrada.

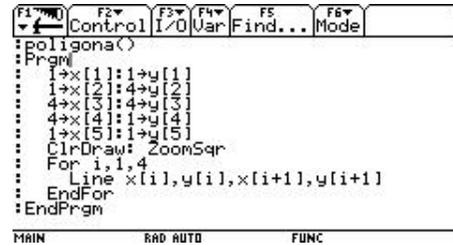


Figura 19

El siguiente ejemplo (figura 20) permite cuadrricular la pantalla de la calculadora, utilizando las instrucciones LineHorz y LineVert.

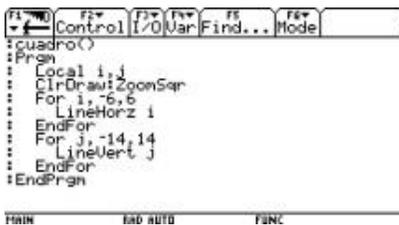
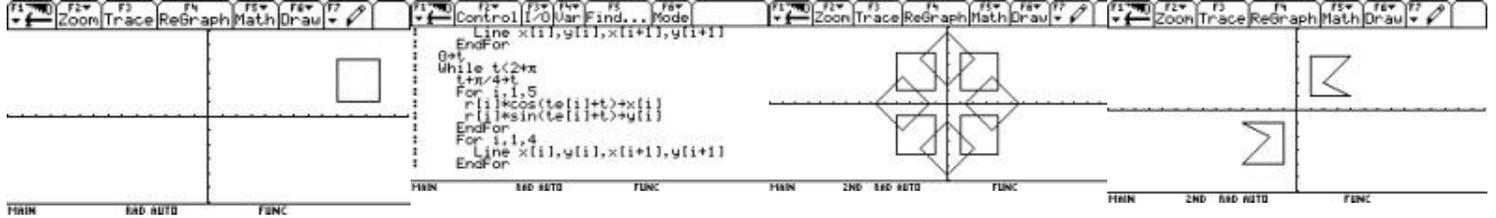


Figura 20

## Transformaciones geométricas



**Traslación:** para trasladar una poligonal a una nueva posición, se incrementan o decrementan las coordenadas de cada vértice de la poligonal. Por ejemplo, para trasladar el polígono de la figura 20, 8 unidades a la derecha, se procede como se muestra en la figura 21.

```

:poligona()
:Prgm
: 1×(1):1+y(1)
: 1×(2):1+y(2)
: 4×(3):1+y(3)
: 4×(4):1+y(4)
: 1×(5):1+y(5)
:ClrDraw: ZoonSqr
: For i,1,4
:   Line x[i]+8,y[i],x[i+1]+8,y[i+1]
: EndFor
:EndPrgm
  
```

Figura 21

**Rotación:** para rotar el polígono de la figura 21, un ángulo de 30 grados, las coordenadas de cada vértice deben ser rotadas dicha cantidad, tal como se indica en la figura 22.

```

:poligona()
:Prgm
: Local x,y,t,e,i,t
: 1×(1):1+y(1):1×(2):1+y(2)
: 4×(3):1+y(3):4×(4):1+y(4)
: 1×(5):1+y(5)
: For i,1,4
:   t=atan(y[i]/(x[i]))+e[i]
:   t=(x[i]^2+y[i]^2)+r[i]
: EndFor
:ClrDraw: ZoonSqr
: For i,1,4
  
```

```

: EndWhile
: EndPrgm
  
```

Figura 22

**Reflexión:** para reflejar una poligonal con respecto al eje  $y$  se cambia, en las coordenadas de cada vértice, el valor de  $x$  por  $-x$ . Para reflejar una poligonal con respecto al eje  $x$  se cambia, en las coordenadas de cada vértice, el valor de  $y$  por  $-y$ . Si se cambian ambos signos de las coordenadas de cada vértice, se obtiene un reflexión con respecto al origen.

En la figura 23 se ilustra un ejemplo de este último caso.

```

:poligona()
:Prgm
: Local x,y,i
: 1×(1):1+y(1):1×(2):1+y(2)
: 4×(3):1+y(3):1×(4):1+y(4)
: 4×(5):1+y(5):1×(6):1+y(6)
:ClrDraw: ZoonSqr
: For i,1,5
:   Line x[i],y[i],x[i+1],y[i+1]
: EndFor
: For i,1,5
:   Line -x[i],y[i],-x[i+1],y[i+1]
: EndFor
:EndPrgm
  
```

Figura 23

# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

---

## Sesión 3

### Utilización de las instrucciones de alto nivel para graficar funciones

Algunas de las instrucciones que la calculadora posee, las cuales permiten graficar en forma rápida funciones y relaciones, son las siguientes (casos particulares):

Setmode ( " graph ", " function ")

Drawfunc sin (x)

Drawinv cos (x)

Drawpol  $5 * \cos(\hat{e}), 5 * \sin(\hat{e}), 0, 4, .2$

Drawparm  $t * \cos (t), t * \sin (t), 0, 5, .1$

Circle 1,1,2

Pxlcrcl 10,10,10

Stopic v 1,0,0,20

Xorpic v 1,20,25

Andpic v1,30,60

Graph expresión1 [, expresión2] [, var1] [, var2]

trace

Shade sin(x),2\* sin(x)

Newplot 1,1,11,12

Disp G

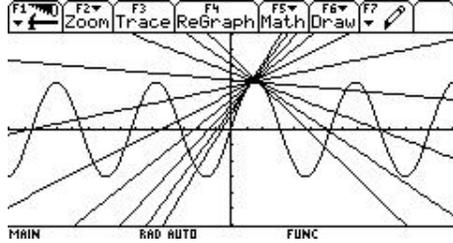
Fnon 1

Fnoff 2

Plotsoff 3

Plotson 2

Style 1, " below "



## Congreso Internacional: Computacionales en el Currículo de Matemáticas

Ejemplificación de las instrucciones inmediatamente anteriores:

En la figura 24 está escrito un programa que grafica la función  $3*\sin(x)$  y una secuencia de líneas tangentes a dicha función para  $x$  entre 1 y 2. Para ello utiliza la instrucción LineTan.

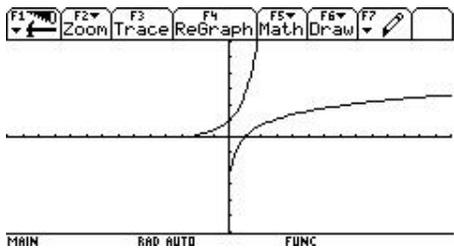
```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode
: tangen()
: Prgm
: setMode("Graph", "FUNCTION")
: ClrDraw:ClrGraph
: ZoomSgr
: Graph 3*sin(x)
: 1→x
: While x<2
:   LineTan 3*sin(x),x
:   x+.1→x
: EndWhile
: EndPrgm
MAIN RAD APPRX FUNC

```

Figura 24

El programa de la figura 25 dibuja inicialmente la función logaritmo natural y luego su inversa.



```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode
: fun1()
: Prgm
: setMode("Graph", "FUNCTION")
: ClrDraw
: DrawFunc ln(x)
: DrawInv ln(x)
: EndPrgm
MAIN RAD AUTO FUNC

```

Figura 25

El programa de la figura 26 permite dibujar una rosa de 4 pétalos en coordenadas polares y una espiral.

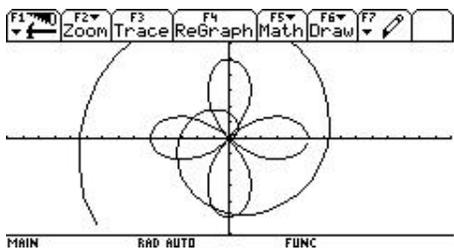


Figura 26

```

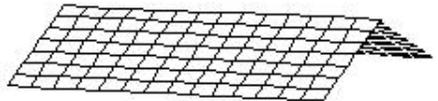
F1 Control F2 I/O F3 Var F4 Find... F5 Mode
: fun1()
: Prgm
: setMode("Graph", "FUNCTION")
: ClrDraw
: DrawPol 5*cos(2*θ),0,2*π,.1
: DrawParm t*cos(t),t*sin(t),0,10,.2
: EndPrgm
MAIN RAD AUTO FUNC

```

Las instrucciones de la figura 27, dibujan una serie de círculos sobre el eje x de izquierda a

## Congreso Internacional: Computacionales en el Currículo de Matemáticas

---



MAIN RAD AUTO 3D

derecha.

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
:func()
:Prgm
:  setMode("Graph", "FUNCTION")
:  ZoomSqr
:  ClrDraw
:  Local r
:  I→r
:  While r<6
:    Circle r,0,r
:    r+.5→r
:  EndWhile
:EndPrgm
MAIN RAD AUTO FUNC
    
```

Figura 27

El código de la figura 28 utiliza la instrucción Graph para graficar en este caso un paraboloides.

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
:func()
:Prgm
:  setMode("Graph", "3d")
:  ZoomStd
:  ClrDraw
:  Graph x^2+y^2,x,u
:  setGraph("Style","Hidden Surface")
:EndPrgm
MAIN RAD AUTO 3D
    
```

Figura 28

Utilizando la instrucción when , es posible graficar una función por trozos como se ilustra en la figura 29.

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
:func()
:Prgm
:  setMode("Graph", "3d")
:  ZoomStd
:  ClrDraw
:  Graph when(x<0,x,-x)
:EndPrgm
MAIN RAD AUTO 3D
    
```

Figura 29

La instrucción " Shade expresión 1, expresión 2" permite visualizar en forma sombreada la parte comprendida entre dos gráficas correspondientes a expresión1 y expresión2 donde la expresión1 sea menor que la expresión2. Veamos el ejemplo de la figura 30.

## Tecnología

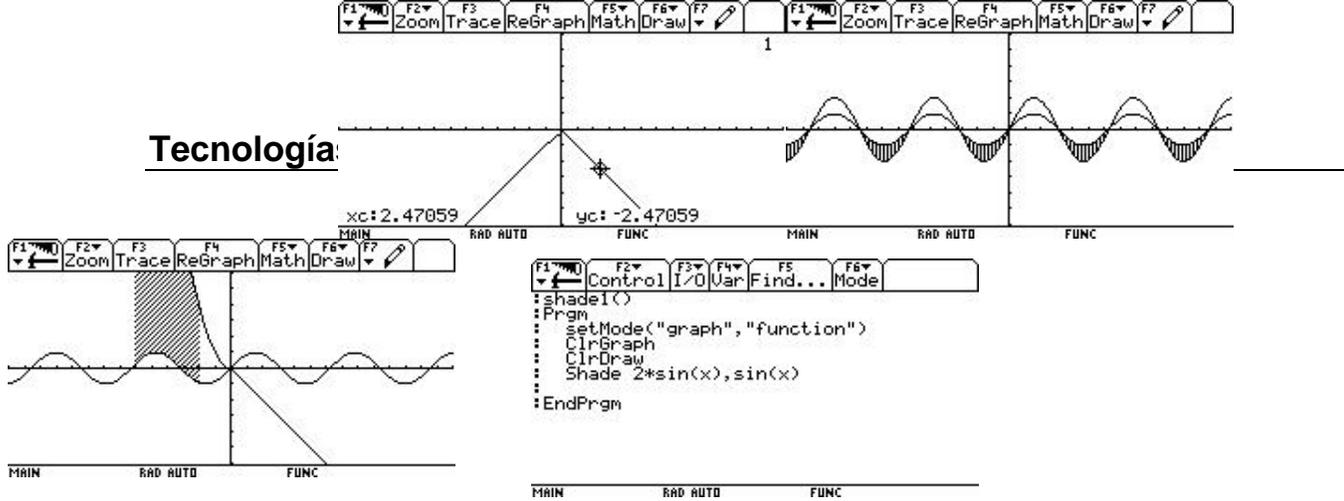


Figura 30

Otro ejemplo de aplicación de la instrucción Shade en un intervalo para dos funciones donde una de ellas se hace por trozos, se presenta en la figura 31.

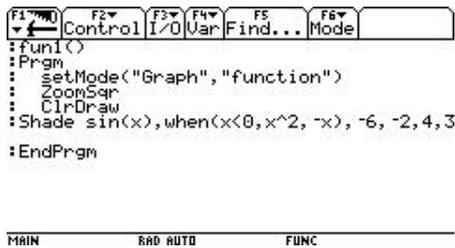


Figura 31

La opción trace es una instrucción que dispone el cursor para que el usuario lo mueva, restringido su movimiento al dominio y rango de las funciones. Veamos un ejemplo en la figura 32.

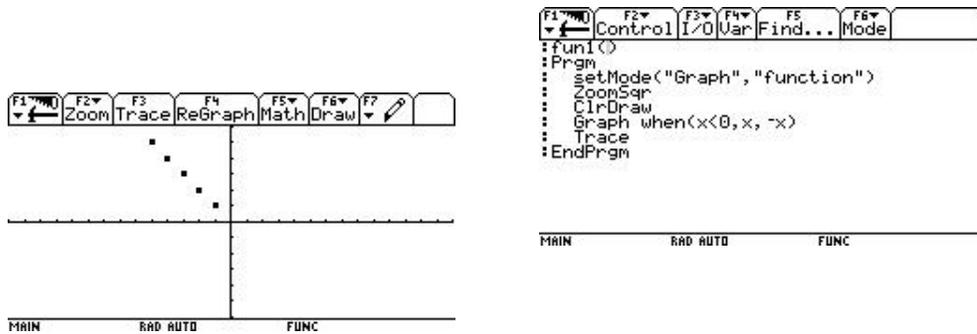
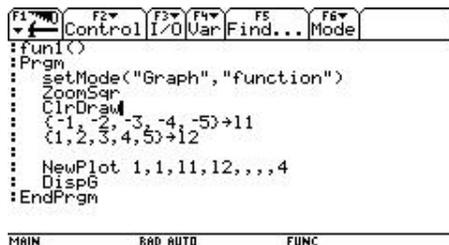


Figura 32

Podemos emplear la instrucción newplot para graficar un conjunto de puntos, como se ilustra en la figura 33.



# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

---

**Figura 33**

**Animación**

La forma corriente de simular el movimiento de una gráfica en la pantalla es mostrándola en forma consecutiva en diferentes puntos de la pantalla. Para ello, dichas imágenes deben grabarse con antelación en memoria y en formato PIC. Esto se logra realizando la gráfica en la pantalla GRAPH y luego con la opción Save Copy As se escoge el tipo Picture y se almacena la figura como se muestra en la figura 34.

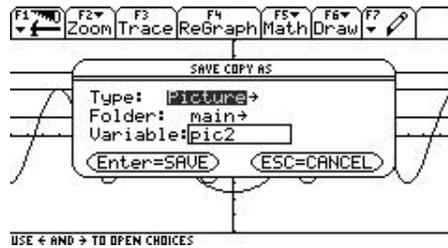


Figura 34

Ya grabadas, estas imágenes se muestran en pantalla en forma consecutiva mediante la instrucción cycle, así como se observa en la figura 35:

```

F1 F2 F3 F4 F5 F6
Control I/O Var Find... Mode
:anima
:Prgm
: DispG
: CyclePic "pic",3,.5,4,-1
:
:EndPrgm
MAIN RAD AUTO FUNC

```

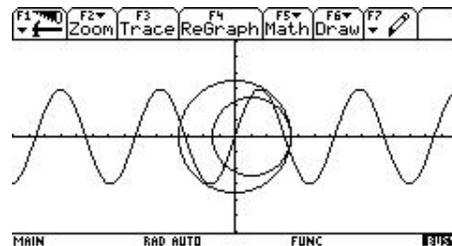


Figura 35

Mediante la instrucción stopic, se puede almacenar una porción de la pantalla gráfica, la cual se puede colocar en diferentes partes de la misma haciendo operaciones ( and, or, xor) con la parte correspondiente. La figura 36 muestra un ejemplo del lugar en el que se dibuja un círculo con la instrucción pxlcrcl y se almacena en la variable v1 mediante la instrucción stopic.

# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

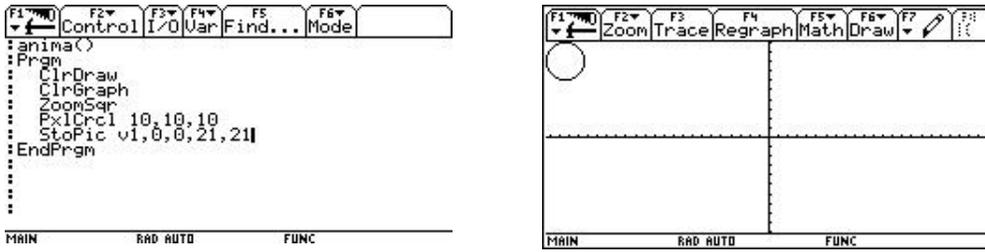


Figura 36

El código de la figura 37 pone en movimiento el círculo almacenado en la variable v1. Note que la instrucción xorpic se ejecuta 2 veces, pues inicialmente coloca la gráfica y luego la borra.

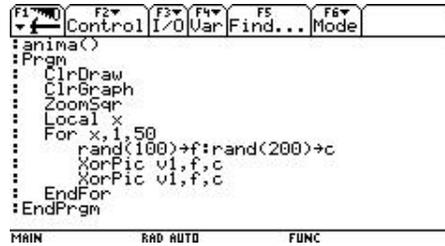
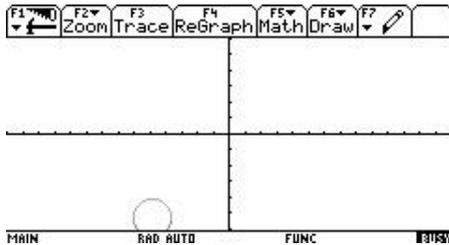
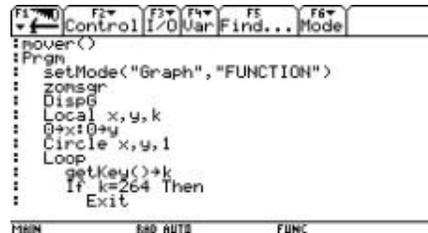
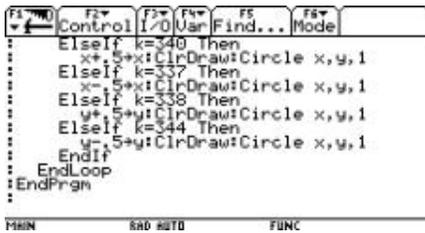


Figura 37

Simulación del movimiento de una gráfica utilizando las teclas de flechas. ( Figura 38)



# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

---

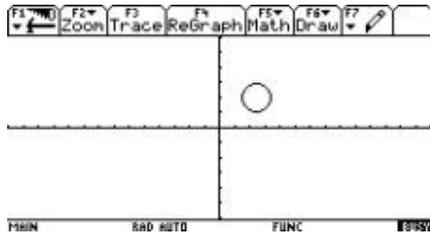


Figura 38

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
: Item "parabolas",t4
: EndTBar
: Lbl sa
: setMode("Split 1 App","Home")
: Exit
: Lbl t1
: Graph tan(x):Cycle
: Lbl t2
: Graph cos(x):Cycle
: Lbl t3
: Graph x+(1,2,3):Cycle
: Lbl t4
MAIN RAD AUTO FUNC
    
```

### Haciendo gráficas desde un menú de opciones

Es posible disponer de una barra de menú con opciones, desde donde se pueden ejecutar órdenes gráficas. Para ello existen varias posibilidades:

- Utilizar la instrucción toolbar ... endtbar. ( Figura 39)

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
: toolbar()
: Prgm
: setMode("Graph", "FUNCTION")
: DispG:FnOff :ClrDraw:ClrGraph
: Loop
:   Toolbar
:     Title "funciones"
:     Item "tangente",t1
:     Item "coseno",t2
:     Item "salir",sa
:     Title "familias"
:     Item "rectas",t3
MAIN RAD AUTO FUNC
    
```

```

F1 Control F2 I/O F3 Var F4 Find... F5 Mode F6
: Graph (1,2,3)*x^2:Cycle
: EndLoop
: EndPrgm
:
:
:
:
:
MAIN RAD AUTO FUNC
    
```

Figura 39

Como se puede observar en la gráfica que se muestra a continuación, la barra de herramientas es reemplazada por la barra programada por el usuario, y cada submenú se activa con las teclas de función. Con las teclas de cursor el usuario se puede desplazar en las opciones y escoger la que esté resaltada presionando la tecla ENTER. Por ejemplo, si se presiona ENTER en la opción coseno, la gráfica obtenida es la ilustrada en la figura 40.

# Congreso Internacional: Tecnologías Computacionales en el Currículo de Matemáticas

---

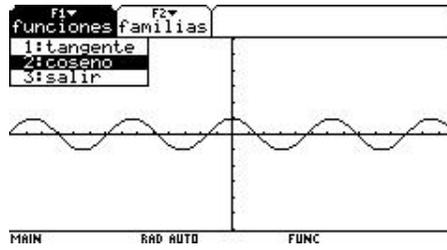


Figura 40

Si se presiona la tecla ENTER en la opción rectas , la gráfica obtenida se muestra en la figura

41.

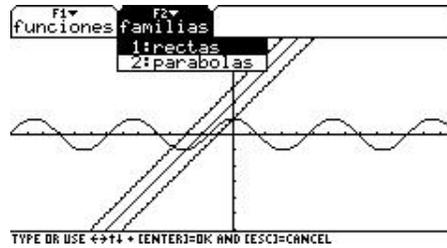
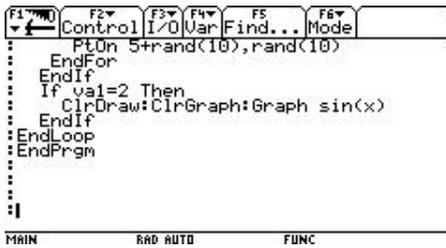


Figura 41

- Utilizar un menú flotante, para escoger de él alguna opción. El ejemplo ilustrado en la figura 42, dispone al usuario un menú flotante.

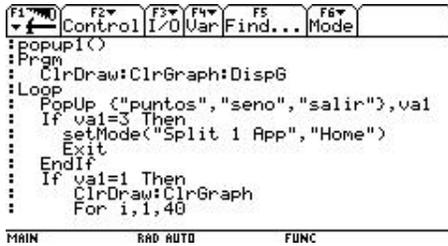


Figura 42

Al presionar la tecla ENTER en la primera opción, se muestra la gráfica de la figura 43:

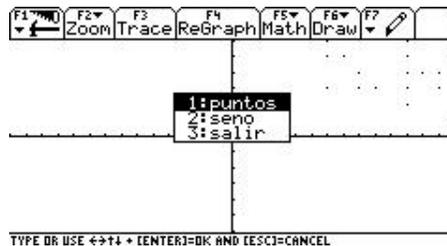
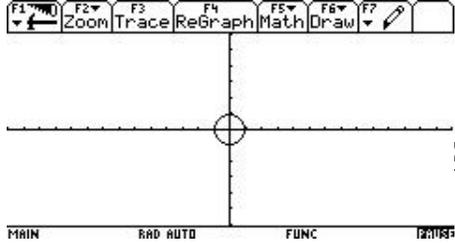


Figura 43

## Congreso Internacional: Computacionales en el Currículo de Matemáticas

---



Usar una ventana de diálogo, lo cual se ejemplifica en el

programa de la figura 44:



Figura 44

### Simulación en Cabri e integración de sistemas de representación

(3 sesiones)

**Fabiola Rodríguez García**

Instituto Pedagógico Nacional

Grupo Coordinador MEN

Incorporación Nuevas Tecnologías al Currículo de Matemáticas

**Martín Eduardo Acosta Gempeler**

Grupo Coordinador MEN

Incorporación Nuevas Tecnologías al Currículo de Matemáticas

**Nivel** . Intermedio

#### Objetivos.

- Analizar la simulación del movimiento de tres aviones que viajan paralelamente.
- Relacionar las diferentes representaciones obtenidas a partir de la toma automática de datos