CLASE 11
USO DE CONDICIONALES Y VECTORES

Condicional de Descartes. Para sacar un mayor provecho a este comando es importante entender cómo funciona su estructura lógica. Para ello se recomienda escribir el seudo-código que permita construir correctamente el código en Descartes. En ese sentido, se debe tener en cuenta:

- Uso de líneas en blanco para organizar las instrucciones en parágrafos y para separar lógicamente las instrucciones que conforman una unidad.

- Uso de la tabulación (identación) para mostrar la estructura lógica del código

Para un condicional simple. La estructura lógica de un condicional simple es la siguiente:

Si ***condición*** Entonces
    Instrucción
Fin si

La condición es una expresión de tipo booleano. Es decir, su evaluación arroja un valor de verdad para la condición: falso (0) o verdadero (1). Por ejemplo, la condición x>t es verdadera para todos los valores de **x** superiores a **t.**

La instrucción se ejecuta sólo sí la condición es verdadera o, igual a 1.

Ejemplo Descartes. El condicional **dibujar-si**, del que vimos un ejemplo en la clase 4, usaba la siguiente estructura:

Si ***menú=1*** Entonces
    Dibujar la curva $f(t)=t^2$
Fin si

En el *Nippe*, dentro de la opción gráficos y para esta curva, escribimos **menú=1** en la casilla correspondiente a **dibujar-si.**

Para un condicional doble., La estructura lógica es la siguiente:

Si *condición* Entonces
    Instrucción 1
Sino
    Instrucción 2
Fin si

Su funcionamiento es similar a la estructura anterior, sólo que tiene dos bifurcaciones. La primera para la condición verdadera, tal como lo vimos en el caso anterior y, la segunda para la condición falsa. Es decir si *condición* es verdadera se ejecuta la instrucción 1, **sino** la instrucción 2

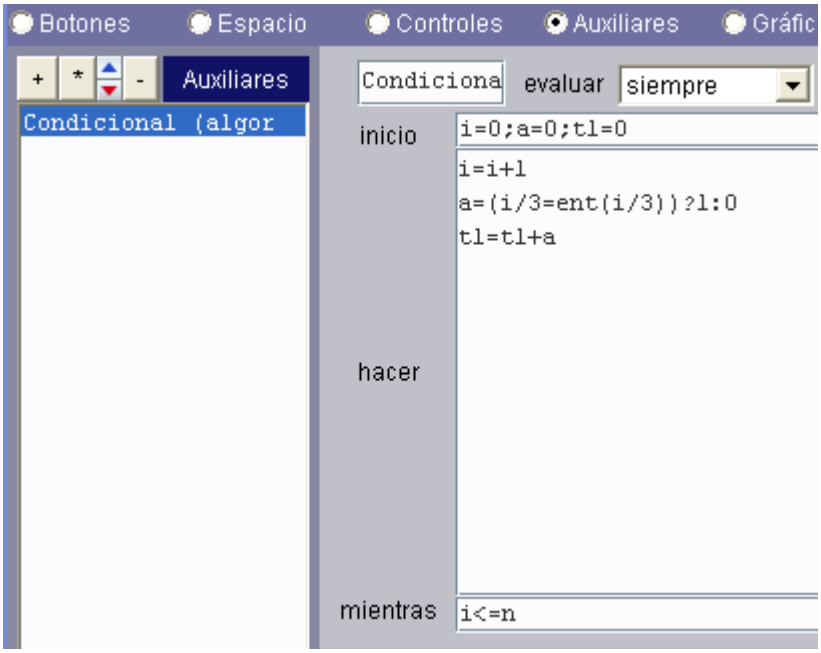
Ejemplos Descartes. Los condicionales dobles son muy útiles cuando se incluyen en un algoritmo. Supongamos que deseamos contar el número de enteros múltiplos de 3 en el rango [1, n]. El seudocódigo sería el siguiente:

```
i=0
a=0
t1=0
Hacer mientras i <= n
        i = i+1
        Si  i/3=entero(i/3) Entonces
                a = 1
        Sino
                a = 0
        Fin si
        t1 = t1 + a
Fin mientras
```

Para comprender este ejemplo, recuerda o repasa la clase 10 sobre el ciclo DO-WHILE. Inicializamos tres variables en cero, luego se ejecuta un ciclo que recorre el intervalo [1, n]. Para cada valor del intervalo (i) se evalúa si es o no múltiplo de 3; para ello, se usa la función "mayor entero que". En otras palabras, si la expresión i/3=entero(i/3) es verdadera, entonces, **a** toma el valor de 1 sino, toma el valor de cero. La variable t1 suma los valores de **a** o los múltiplos de 3. Luego puedes agregar un texto para que muestre el valor de t1.

El código Descartes para condicionales dobles tiene la siguiente sintaxis:

var = (***condición***)?:instrucción 1:instrucción 2

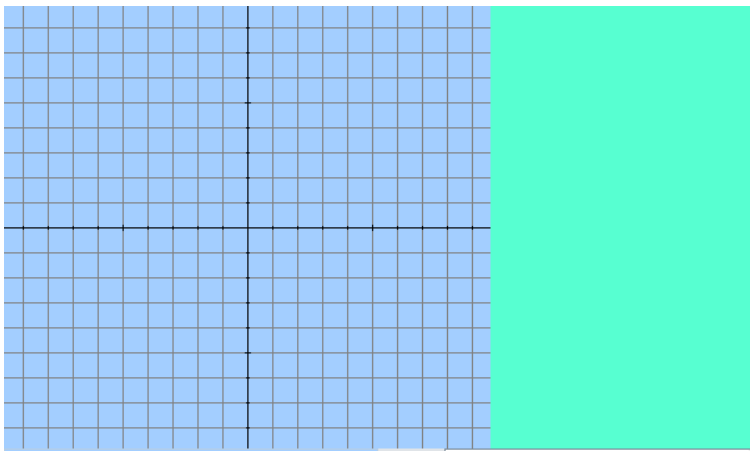


Lo cual significa que la variable **var** toma el valor de la instrucción1 si la condición es verdadera o el de la instrucción 2 si es falsa. Para el ejemplo lo codificaríamos como se ve en la siguiente en la imagen izquierda. Con esta introducción podemos ejecutar nuestra primera actividad.

Actividad 1. Hallar el número de puntos máximos o mínimos locales para una función dada en el intervalo [inicio, fin].

Si bien el objetivo son los máximos y mínimos, presentaremos la gráfica para verificar los resultados y, adicionalmente, mostraremos la ecuación y la gráfica de la recta tangente en un punto cualquiera de la función.

1.1 Para juntar las dos actividades de esta clase, crearemos dos espacios E1 y E2. El primero, en el que diseñaremos la actividad uno, tendrá un ancho de 65% y una escala de 20, y el segundo con un ancho de 35%, ubicado en x = 390. Las dimensiones del *applet* serán de 600x400. Una propuesta de diseño de la escena se observa en la imagen siguiente (puedes escoger los colores que desees):



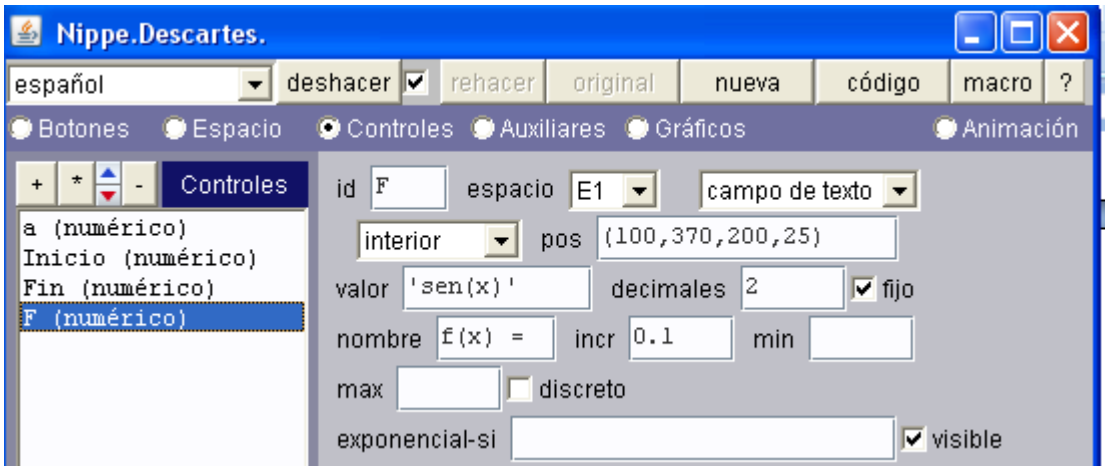
1.2 **Controles**. Usaremos cuatro controles:

a: Control tipo barra, al interior de la escena, con un valor inicial de 1. Este control nos permitirá cambiar el valor de la abscisa del punto de la función en el cual hallaremos la recta tangente, el intervalo es [-10, 10]; es decir; **min = -10** y **max = 10.** Este control tendrá un sólo decimal y **pos = (0, 340, 130, 22)**.

Inicio: Control tipo barra, al interior de la escena, con un valor inicial de -5. Es el valor especificado en el intervalo [-1, 9]. También le asignaremos un decimal, además de la posición y tamaño definidos por **pos = (135, 340, 130, 22).**

Fin: Control tipo barra, al interior de la escena, con un valor inicial de 5. Es el valor especificado en el intervalo, el cual varía de un mínimo igual al control **Inicio** hasta 10, con un decimal y **pos = (270, 340, 130, 22)**

F: Control numérico tipo campo de texto, al interior de la escena, con un valor inicial de 'sen(x)', que llamaremos **f(x) =**. La identificación del control deber ser **Id = F** (observa la imagen siguiente); debes, además, activar la opción **visible** para poder ingresar otras funciones. Puede ubicar los controles en otras partes de la escena, es aquí donde debes practicar el diseño que más se ajuste a tus gustos.



1.3 **Auxiliares.** Usaremos las siguientes auxiliares:

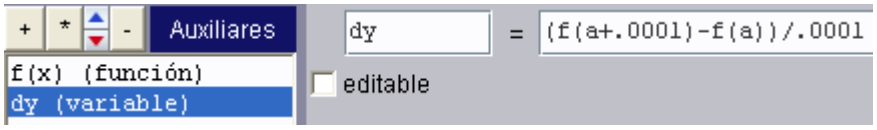
f(x): Es una función, cuyo valor es F. Es decir, f(x) = F. Esta forma de asignar la función obedece a que F no es una gráfico (es una variable definida por el control anterior) y a que f(x) no se puede editar (recuerda la clase anterior).

dy: es una variable que nos permite calcular la derivada en el punto (a, f(a)). Recordemos la definición del cálculo diferencial:

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \left(\frac{f(x + \Delta x) - f(x)}{\Delta x} \right)$$

Como Δx tiende a cero, podemos usar un número muy pequeño. Para el caso usaremos Δx = 0.0001. Es decir,

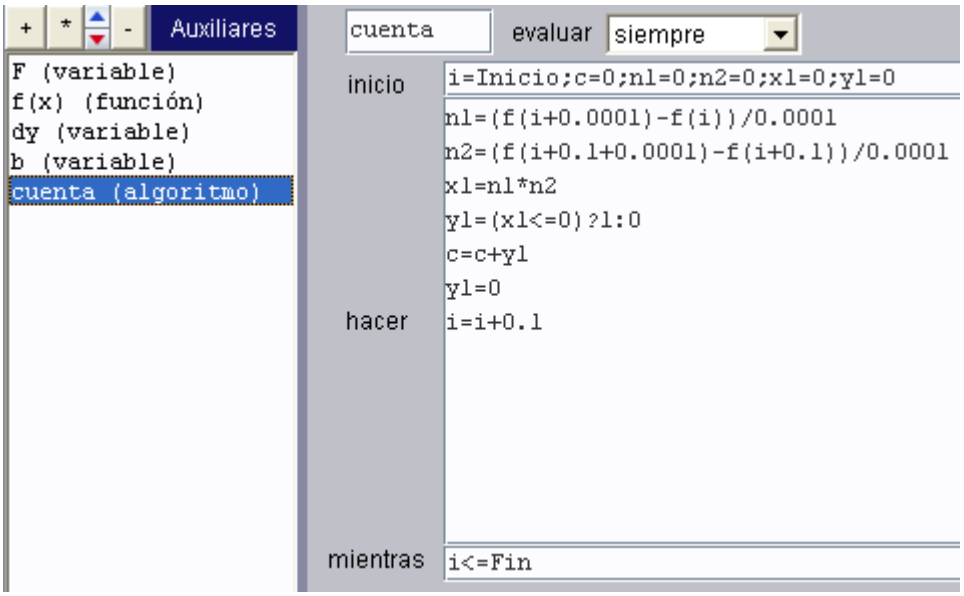
dy = (f(a+0.0001)-f(a))/0.0001



b: Variable que representa el valor de b en la ecuación de la recta tangente y = mx + b. Es decir: b = f(a)-dy*a

cuenta: Es un algoritmo que nos contará el número de puntos máximos o mínimo locales.

Escribe los datos tal como aparecen en la siguiente imagen:

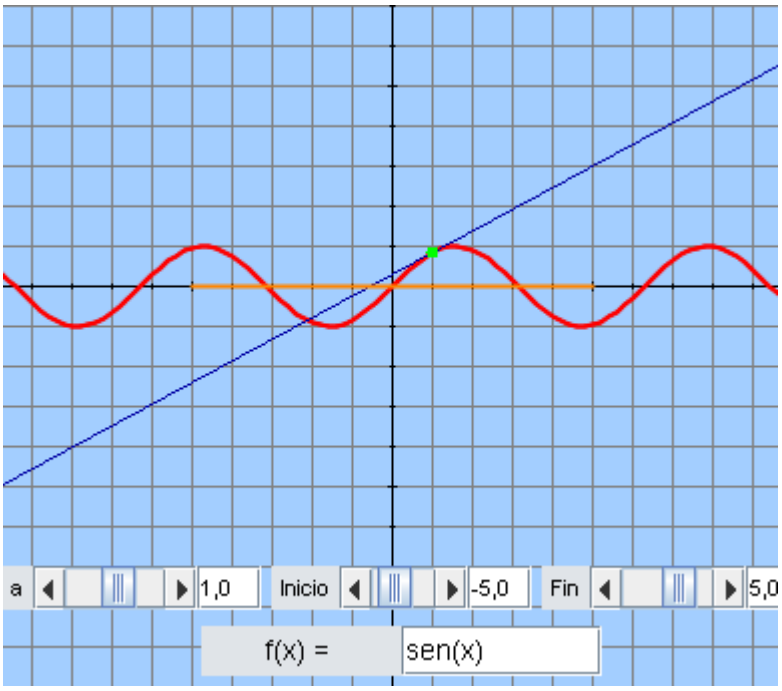


Observa que n1 es la derivada en la abscisa i, n2 es la derivada en la abscisa i+0.1. Si el producto de estas dos derivadas es negativo o cero, es porque hubo cambio de sentido en la curva. Es decir, hay un mínimo o un máximo local. Observa el uso del condicional.

1.4 **Gráficos.** Usaremos los siguientes gráficos en el espacio E1:

curva (t,f(t)). Nos permite dibujar la función f(x) dada o reingresada en el intervalo [-10, 10] y con pasos = 100. Desactiva la opción **visible**. Nosotros hemos usado un color rojo y ancho 2.

ecuación y=dy*(x-a)+f(a). Representa la recta tangente en el punto (a, f(a)). Para esta recta hemos usado color azul obscuro.



punto (a, f(a)): Es el punto sobre el cual se dibujará la recta anterior. Usa color diferente y un tamaño de 3 para destacar el punto.

segmento (inicio,0)(fin,0). Se usa sólo para destacar el intervalo utilizado. Usa un color y tamaño que permita destacar el segmento

1.5 **Texto dinámico.** Si bien se añade en la opción gráficos, lo enuncio en otro apartado porque vamos usar el condicional simple **dibujar-si** para darle mejor forma al texto dinámico.

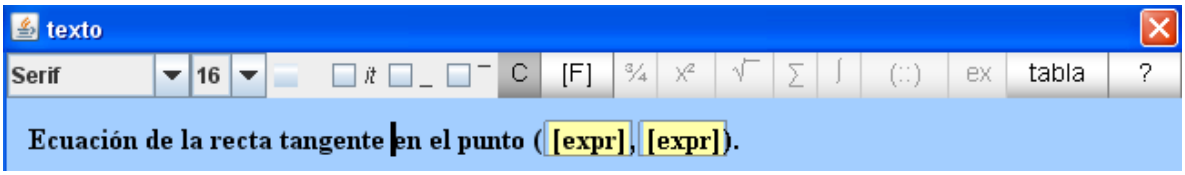
Añadamos el siguiente texto en la posición [20, 60] **y = dyx + b.** La parte en rojo se inserta como expresiones (recuerda las clases 7 y 10) y el texto sólo se dibuja si b>0.

Añadamos el siguiente texto en la posición [20, 60] **y = dyx - abs(b).** La parte en rojo se inserta como expresiones (recuerda la clase 7) y el texto sólo se dibuja si b<0.

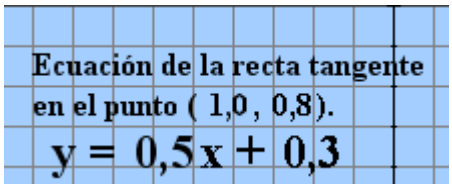
Las dos expresiones anteriores nos evitan que aparezcan textos como **y = 3x +- 2.**

Añadamos el siguiente texto en la posición [20, 60] **y = dyx,** texto que sólo se dibuja si b=0. Esto nos evita expresiones como **y = 3x + 0.**

Añadamos el siguiente texto: **Ecuación de la recta tangente en (a, f(a)).** La parte en rojo se inserta como expresiones.

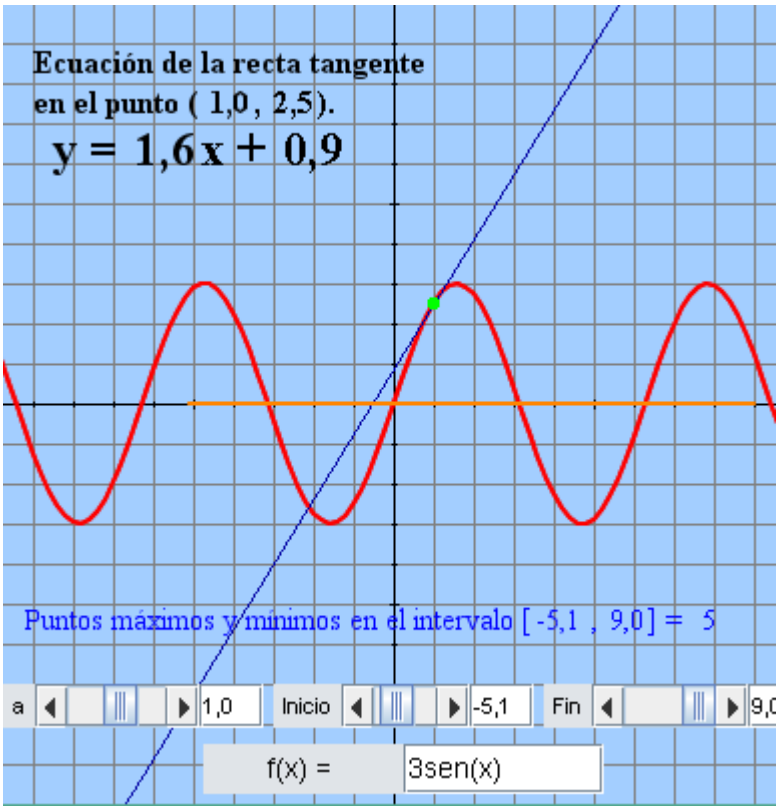


Aparecerán expresiones como (3, 3.5) donde aún se utiliza la coma para decimales y no el punto. Quizá en la próxima versión del Descartes se habilite el punto para los decimales.



Agreguemos el siguiente texto en la posición [5, 300]: **Puntos máximos y mínimos en el intervalo [inicio, fin] = c** (revisa el algoritmo)**.** La parte en rojo se inserta como expresiones.

Ahora si… revisa tu diseño. Debe aparecer una figura como la siguiente:



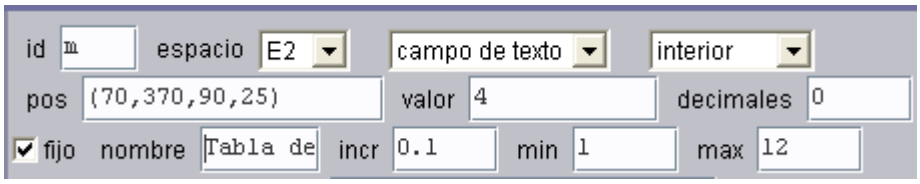
Actividad 2. Usar vectores para mostrar la tabla de multiplicar de un número dado m.

De la documentación de Descartes se obtiene:

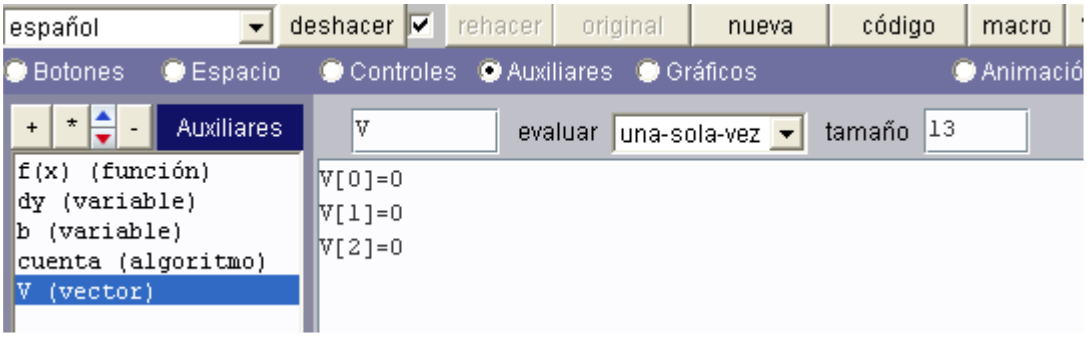
> Un **vector** es una **lista de constantes**. Un vector tiene un identificador y un tamaño, que es su longitud o número de elementos. La manera de hacer referencia a un elemento de un vector es escribir el identificador del vector seguido del número del elemento entre corchetes. Los elementos de un vector se numeran siempre comenzando por **0**. Los elementos no inicializados de un vector tienen por defecto el valor 0.

Es cierto que las tablas que diseñaremos en el espacio 2 (E2) no tienen relación alguna con los máximos y mínimos del espacio 1. Nuestro propósito es sólo didáctico y orientado a comprender el funcionamiento de los condicionales, algoritmos y, para esta actividad, los vectores.

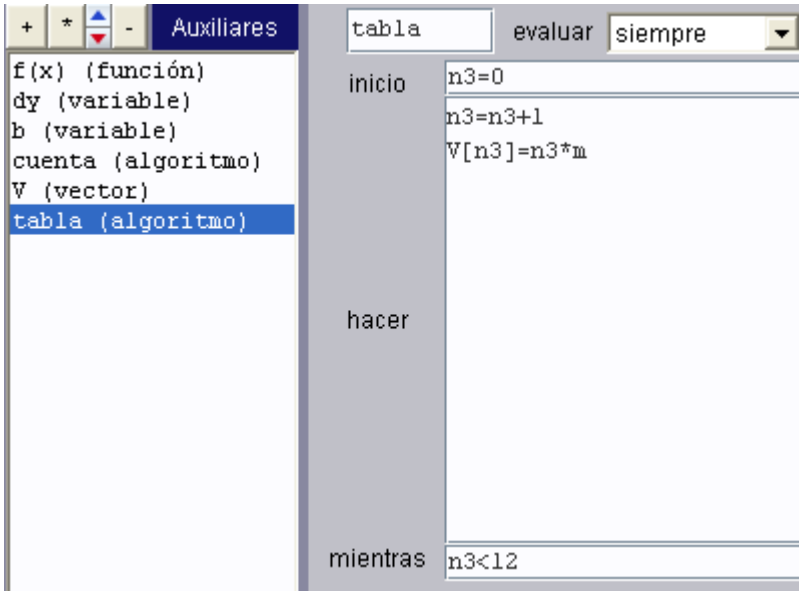
2.1 Añadamos inicialmente el control numérico tipo **campo de texto** identificado por **m** (id = m), con nombre **tabla del**, y con valor inicial de 4. El incremento debe ser 1, su mínimo 1, y su máximo 12.



2.2 Añadamos, entonces, un vector V de 13 elementos (ver imagen) desde la opción **Auxiliares**. Podemos inicializar todos los elementos del vector en cero ó, podemos usar un algoritmo con una expresión como V[i] = 0, ó simplemente lo dejamos así como se observa en la imagen, que por defecto inicia todos los elementos en cero. El tamaño de 13 obedece a que Descartes asigna un elemento V[0] a todo vector que creemos.



2.3 Ahora añadamos un algoritmo para que almacene en los elementos del vector los productos de la tabla de multiplicar a mostrar (ver imagen). Es decir: V[1] = 1*m, V[2] = 2*m, …, V[12] = 12*m.



2.4 Finalmente, en la posición [20, 40] del espacio 2 (E2) escribiremos el siguiente texto:

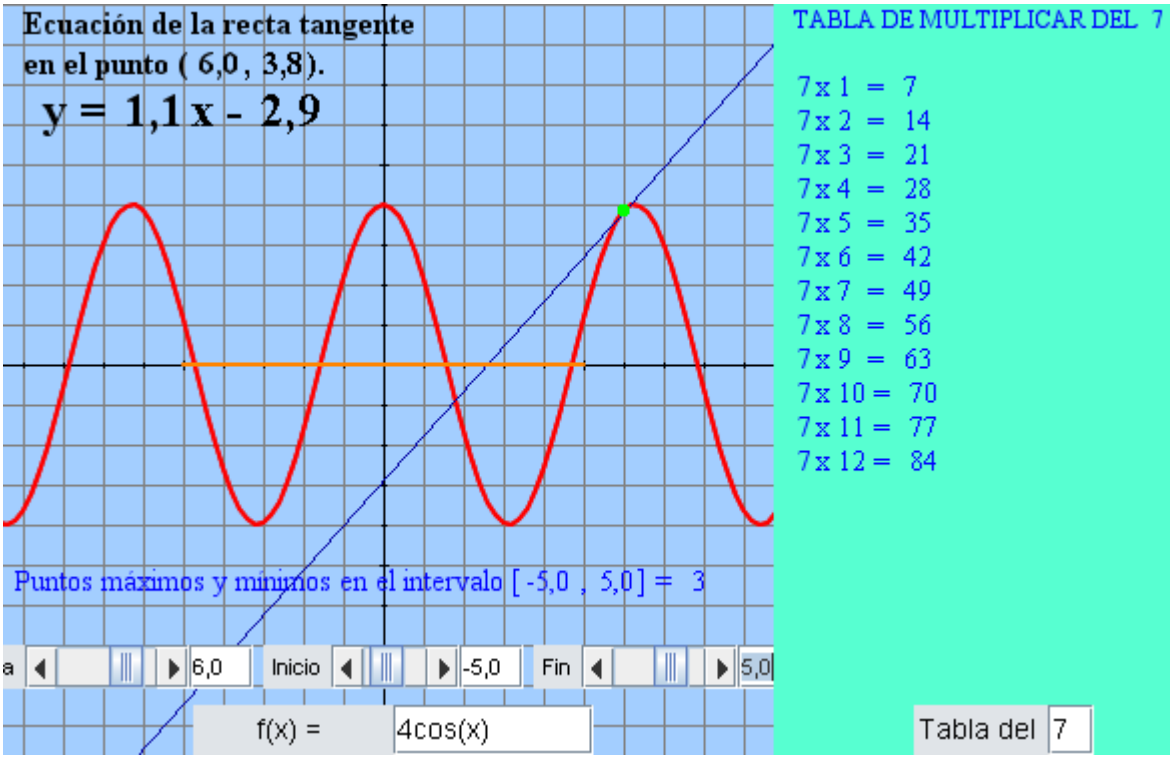
TABLA DE MULTIPLICAR DEL **m**

mx1 = **V[1]**
mx2 = **V[2]**
mx3 = **V[3]**

Hasta terminar con **m**x12 = **V[12].** La parte en rojo se inserta como expresiones.

Es cierto que no era necesario recurrir a vectores para mostrar estas tablas. Con expresiones como **m**x7 = **mx7** hubiese dado el mismo resultado. Sin embargo, el objetivo era practicar con una lista, vector o matriz unidimensional de elementos.

Realizado lo anterior debes obtener un *applet* final, como el de la siguiente imagen:



Hasta pronto. En www.descartes3d.blogspot podrás practicar con el *applet* final

Juan Guillermo Rivera Berrío