



# **S**IMULACIÓN DE **M**ODELOS **M**ATEMÁTICOS Y **T**EMPORALES

Para la versión 2.5 (RC1) o superiores

*Prof. José Manuel Ruiz Gutiérrez*  
*[jruiz1@platea.pntic.mec.es](mailto:jruiz1@platea.pntic.mec.es)*

# Índice

## **1. GENERADORES TEMPORALES DE ESTIMULOS: GENERADORES DE SEÑAL**

- 1.1. Retardar un evento.
- 1.2. Impulso retardado
- 1.3. Generador numérico.

## **2. CONTADORES Y TEMPORIZADORES**

- 2.1. Contador
- 2.2. Ticket Counter
- 2.3. Generador de Impulsos (Ticker Timer)
- 2.4. Contador Temporizador
- 2.5. Contador Adelante/Atrás
- 2.6. Temporizador

## **3. GENERACION DE VALORES CORRESPONDIENTES A UNA EXPRESIÓN MATEMÁTICA DEL TIPO $y=f(x)$**

- 3.1. Método 1: (Bloque Genera Gráfico)
- 3.2. Método 2: (Bloque 1D Genera Gráfico)

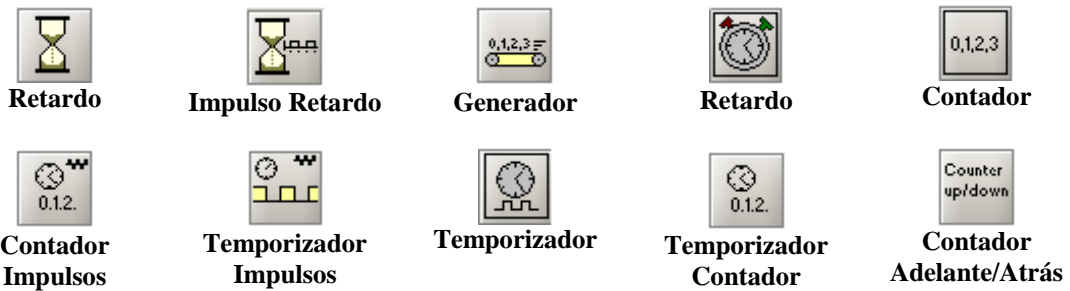
## **4. CALCULANDO CON MyOpenLab**

- 4.1. Cálculo utilizando el objeto “Calculadora”
- 4.2. Cálculo utilizando el objeto “Calculadora  $y=f(x)$ ”
- 4.3. Cálculo utilizando el objeto “Calculadora ext.”
- 4.4. Ejemplos de Cálculo
  - 4.4.1. Ejemplo1: Cálculo del valor de la resistencia de un hilo en función de su longitud y su sección.
  - 4.4.2. Ejemplo2: Cálculo del espacio recorrido por un móvil.

# 1. GENERADORES TEMPORALES DE ESTIMULOS: GENERADORES DE SEÑAL

En numerosas aplicaciones es preciso suministrar al modelo una variable independiente, normalmente el tiempo que es preciso generar. Para ello se recurre al establecimiento de circuitos contadores que con una cadencia temporal modificable se encargan de esta tarea.

MyOpenLab posee una librería de funciones que prestan este servicio de una manera fácil y potente. Se trata de la librería “Utilidades”. De la que mostramos sus elementos.



## 1.1. Retardar un evento.

En ocasiones nos interesara retardar el cambio de una variable digital de un estado a otro.

En este caso se utilizara la función “Retardo” tal como se indica en la figura,

Cuando pulsemos el botón se activará a la salida el diodo led al cabo de un tiempo que previamente hemos establecido en la ventana “Editor de Propiedades”. Mediante el valor “Delay [ticks]”

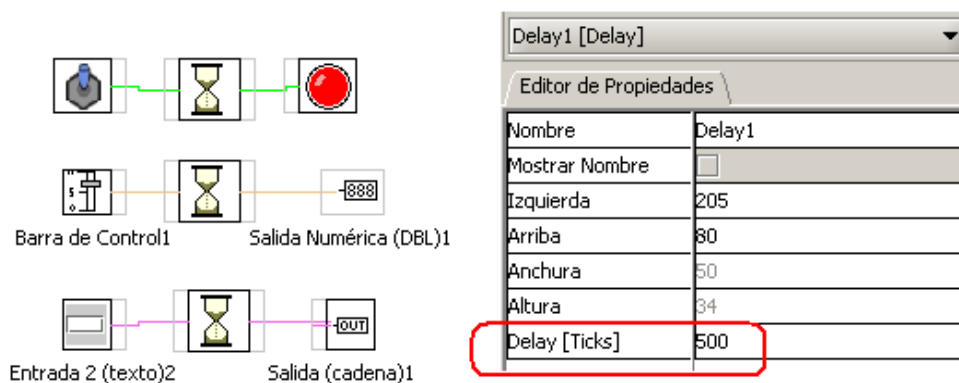


Figura 1

Este bloque admite cualquier tipo de variable, lo cual significa que podemos retardar cualquier valor (string, dbl, bol, int, etc).

### 1.2. Impulso retardado

En ocasiones nos interesará generar un impulso con un retardo. Para este fin se utiliza el bloque de función “Impulso retardo”

La forma de actuar se muestra en el diagrama de tiempo de la figura.

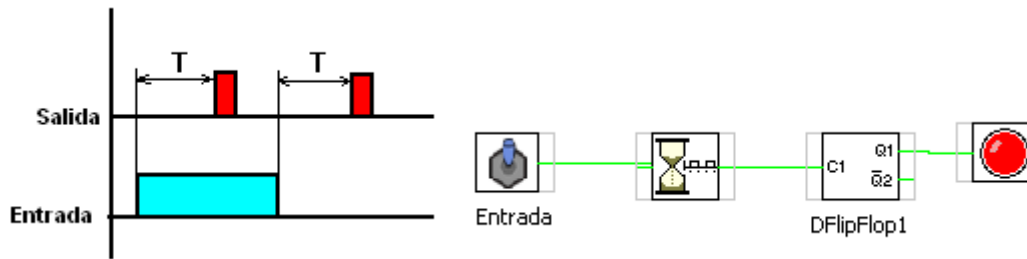


Figura2

Vemos que cuando se produce un cambio de nivel (flanco) empieza el tiempo de retardo “T” transcurrido el cual se produce un impulso en la salida.

En la figura se ha colocado un biestable tipo D con el fin de que el impulso de salida sirva para hacerle bascular y así poder comprobar esta función.

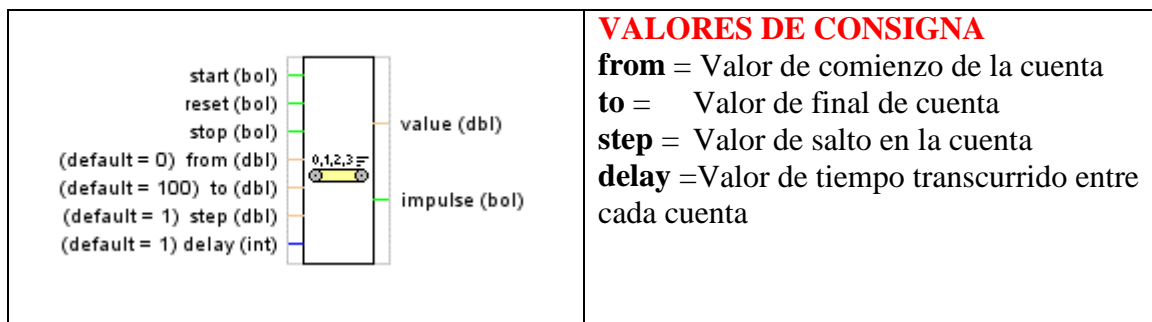
### 1.3. Generador numérico.

En ocasiones es necesario disponer de una señal que se va incrementando en pasos de un determinado valor desde un valor de comienzo hasta un valor final.

Cuando necesitemos una variable independiente que adquiera un conjunto de valores comprendidos entre un intervalo determinado y con una precisión determinada este es nuestro bloque.

El bloque como vemos trabaja con variables de tipo decimal “dbl” si bien la variable de duración del impulso de cuenta es de tipo entero “int”.

En la figura vemos el bloque y sus parámetros.



<b>VALORES DE GOBIERNO</b>	<b>VALORES DE SALIDA</b>
<b>start</b> = Señal de inicio de cuenta (booleana)	<b>value</b> = Valor de salida del contador
<b>reset</b> = Puesta a cero del contador	<b>impulse</b> = Impulso de salida cuando ha realizado un ciclo de cuenta
<b>stop</b> = Detección de la cuenta	

Supongamos que tenemos una variable X que queremos que cambie entre 50 y 150 de 3 en tres y con una cadencia (tiempo entre valor y valor) de 1,5 s.f.

En este caso el sistema quedaría tal como se indica en la figura:

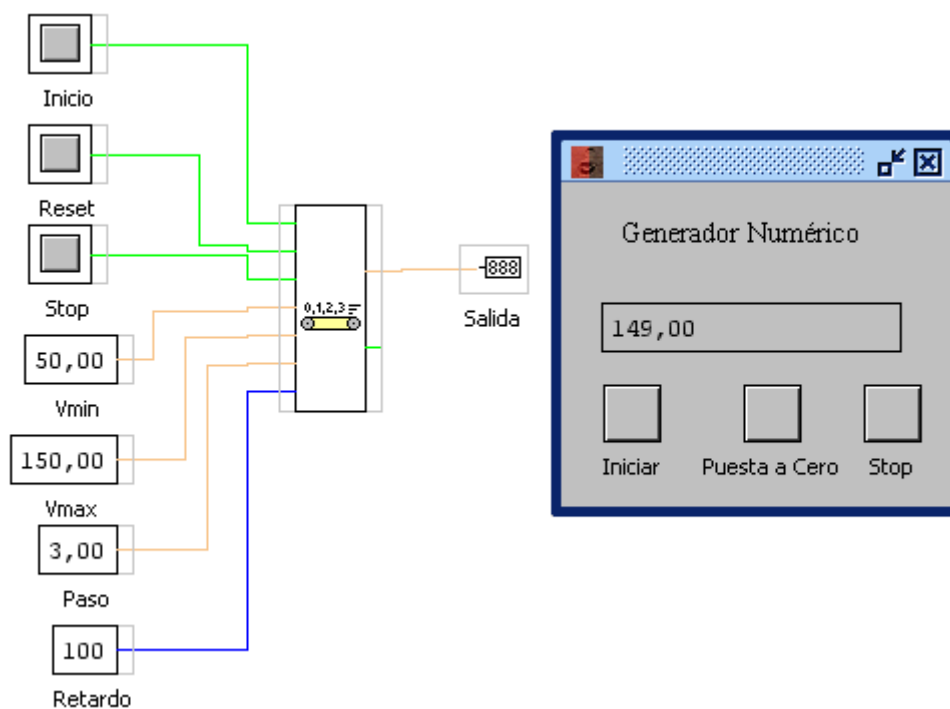


Figura 3

## 2. CONTADORES Y TEMPORIZADORES

En este apartado vamos a estudiar el resto de bloque con los que podemos generar secuencias de conteaje o simplemente generación de impulso para alimentar nuestras aplicaciones.

En la figura siguiente vemos los montajes de cada uno de estos 6 tipos de bloque de función temporal así como la pantalla de presentación de los elementos colocados en el panel de circuito.

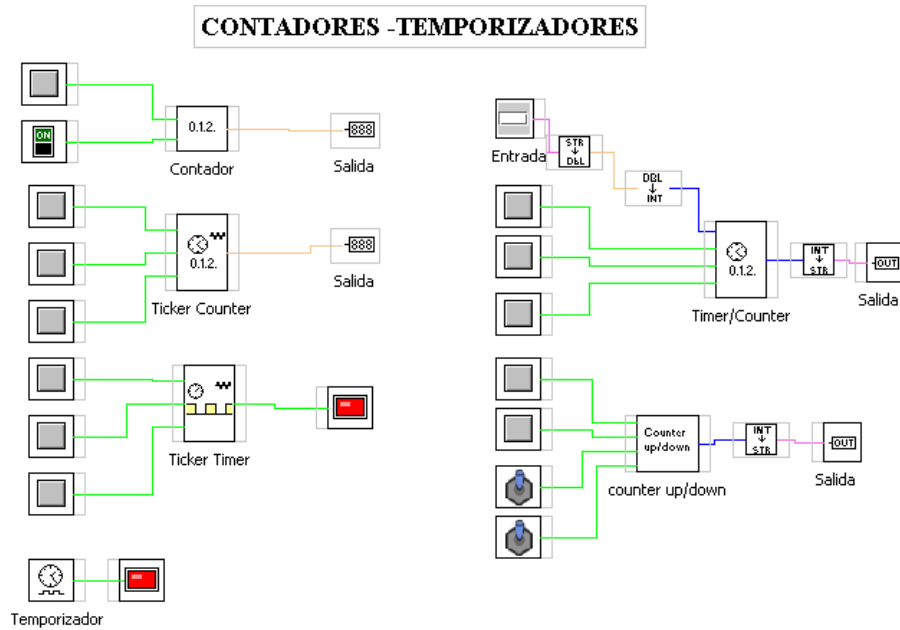


Figura 4

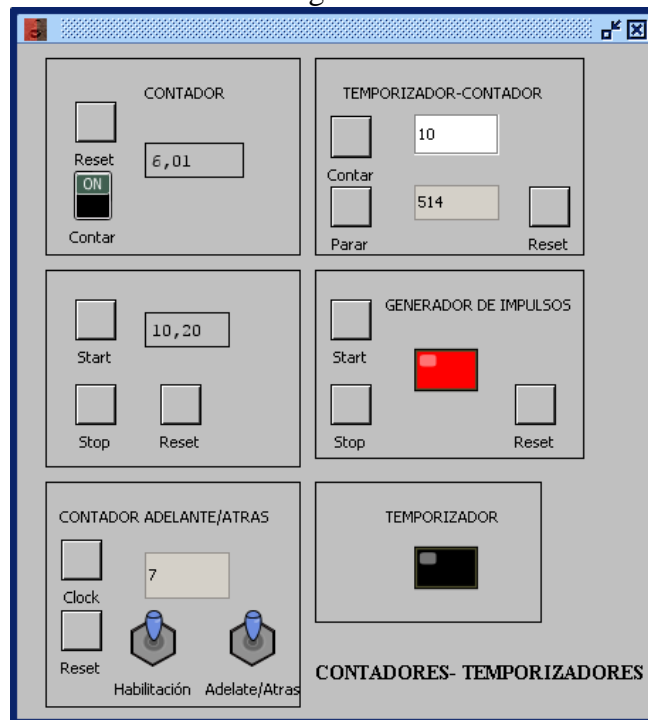


Figura 5

## 2.1. Contador

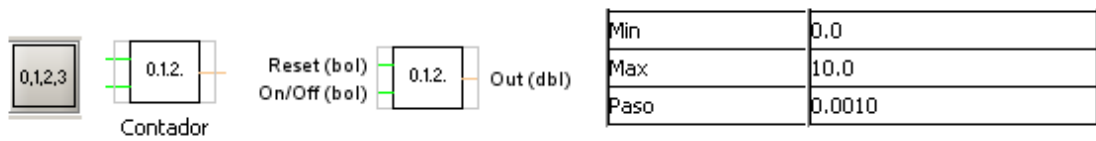


Figura 6

Esta función genera un número de tipo “dbl” de acuerdo a unos parámetros que son los siguientes:

*Min:* valor mínimo desde el que comienza la cuenta

*Max:* valor máximo en el que acaba la cuenta

*Paso:* valor del paso o incremento entre un valor y el siguiente.

El contador es cíclico. Es decir cuando llega al valor máximo empieza de nuevo a contar.

La entrada Reset lo pone a cero. El contador para inmediatamente que soltemos el botón vuelve a contar.

## 2.2. Ticket Counter



Figura 7

Este contador genera un nuevo tipo “dbl” de acuerdo con los parámetros que se especifican en el “Editor de Parámetros”. Estos son:

*Delay:* Tiempo que transcurre entre cada paso del contador (retardo de impulso de cuenta)

*Step:* Valor del incremento del contador

*From:* Valor en el comienza la cuenta

*To:* Valor final de la cuenta.

El contador una vez que ha llegado al valor final de la cuenta se detiene.

Con la entrada “Stop” se detiene la cuenta.

Con la entrada “Reset” se pone a cero el contador y si aun no había acabado la cuenta la inicia de nuevo.

La señal “Star” arranca el contador.

**Veamos un ejemplo:** Queremos generar una variable cuyos valores se den en el entorno entre 10 y 100 y que cuente de 3 en 3 unidades con un retardo de tiempo entre cada paso de cuneta de 200 ms.

Los parámetros serían:

*Delay=200    Step=3    From=10    To=100*

### 2.3. Generador de Impulsos (Ticker Timer)

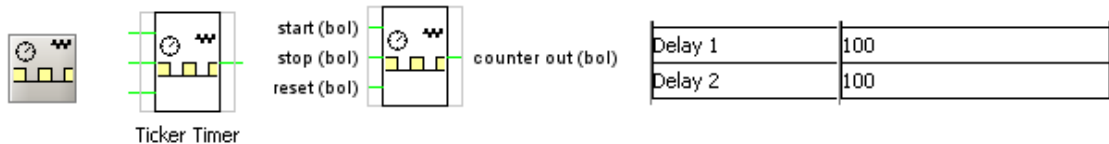


Figura 8

Este objeto es capaz de generar impulsos de tipo digital con una cadencia temporal ajustable con los parámetros:

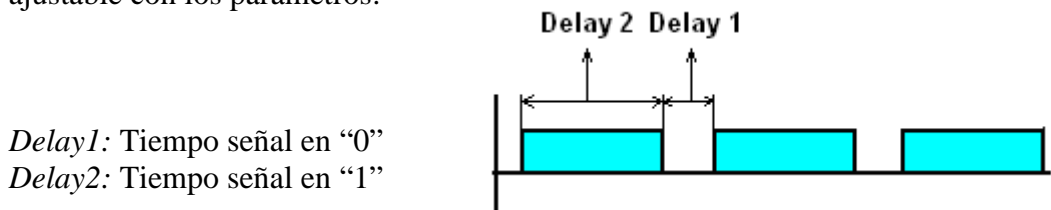


Figura 9

Las señales que gobiernan esta función son:

- Start:* Iniciar la generación de los impulsos
- Stop:* Detener la generación de los impulsos
- Reset:* Puesta a cero (comienza la secuencia)

### 2.4. Contador Temporizador

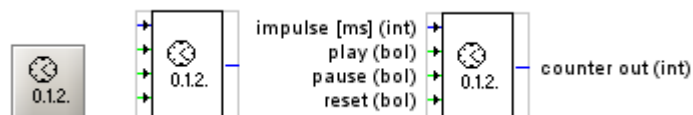


Figura 10

Este bloque de función genera una señal de tipo analógico (valor entero) que se incrementa de 1 en 1 y cuya cadencia de tiempo entre un valor y el siguiente se puede modificar con el parámetro de entrada “impulse (ms)”. Este valor es un número de tipo entero que se coloca en la entrada del contador.



Las señales de entrada que controlan el contador son de tipo booleano:

*Play:* Inicia la cuenta

*Pause:* Detiene la cuenta

*Reset:* Pone a cero el contador.

Este contador no tiene valores de comienzo y fila de cuenta, es decir siempre comienza en 0 y no tiene tope teórico de cuenta.

## 2.5. Contador Adelante/Atrás

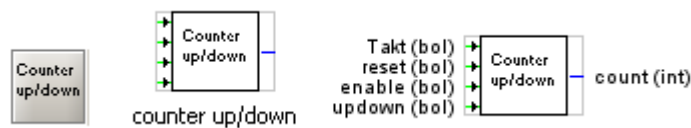


Figura 11

Este contador es capaz de contar los impulsos que se produzcan en su entrada dando en su salida un valor equivalente a la cuenta de estos impulsos de entrada.

Las señales de entrada que gobiernan este bloque contador son las siguientes:

*Takt:* Entrada de impulsos (bol)

*Reset:* Puesta a cero del contador (bol)

*Enable:* Habilitación del contador (bol)

*Updown:* Permite en un estado la cuenta hacia delante y en el otro hacia atrás. (1 adelante y 0 atrás) (bol)

## 2.6. Temporizador

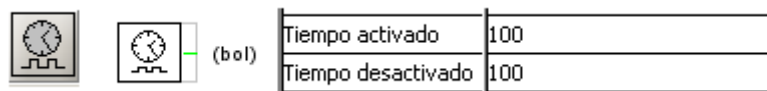


Figura 12

Este bloque genera impulsos de tipo reloj pudiendo ser modificado el tipo de “activado” y “Desactivado” de la señal mediante los valores correspondientes de la ventana “Editor de Parámetros”.

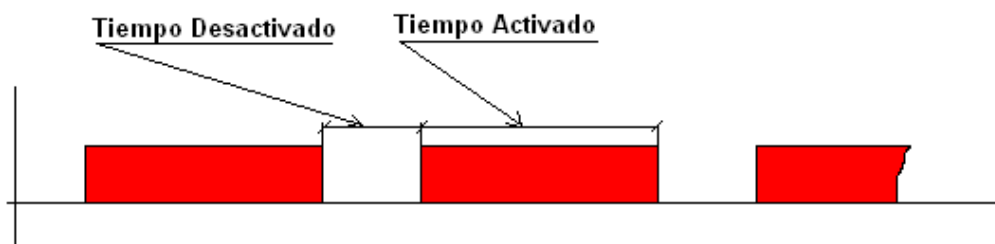
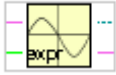


Figura 13

### 3. Generación de valores correspondientes a una expresión matemática del tipo $y=f(x)$

#### 3.1. Método 1: (*Bloque Genera Gráfico*)



En este ejemplo vamos a realizar el trazado de una función acotada entre unos valores determinados de la variable independiente ( $x$ ). Para ello recurrimos al bloque funcional “Genera Grafico” que se muestra en la figura.

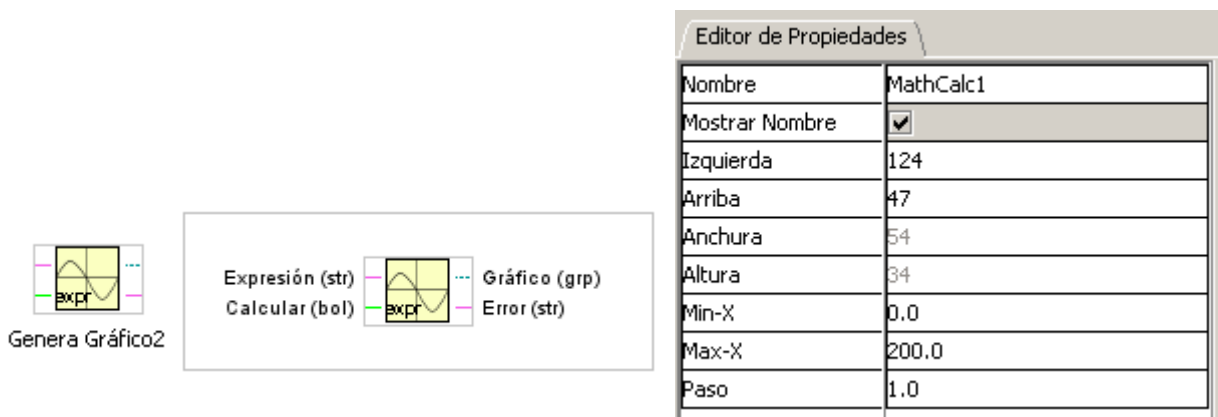


Figura 14

Este bloque realiza el cálculo de un conjunto de valores acotados mediante los parámetros “Min-X” y “Max-X” en pasos de valor indicado en el parámetro “Paso”

En la figura se trata de realizar el calculo de 200 valores de  $x$  que irían desde el valor 0 al valor 200 de 1 en 1.

La entrada expresión recoge un string que representa la expresión matemática sobre la que se calcularán los valores (tomando como variable independiente el valor  $X$ )

La entrada “Calcular” cuando se activa se realiza el cálculo de los valores.

Para representar los valores calculados hemos recurrido al objeto grafico “Osciloscopio 3.01” que es un trazador muy potente que permite incluir en las varias señales y que en su momento explicaremos detenidamente.

En la figura siguiente vemos el Panel de Circuito y el Panel de Visualización de la aplicación.

La introducción de la expresión se realiza mediante un objeto de “Entrada de texto” que en tiempo de ejecución nos permite colocar la expresión y posteriormente validar el cálculo mediante el botón “calcular valores”.

El panel de visualización recoge el cálculo de dos expresiones matemáticas.

La definición del color y el tipo de grafico se realiza haciendo uso de las funciones de “Agrupación de valores” que entregan en su salida valore agrupados que son los que necesita adquirir como entrada el elemento de representación gráfica.

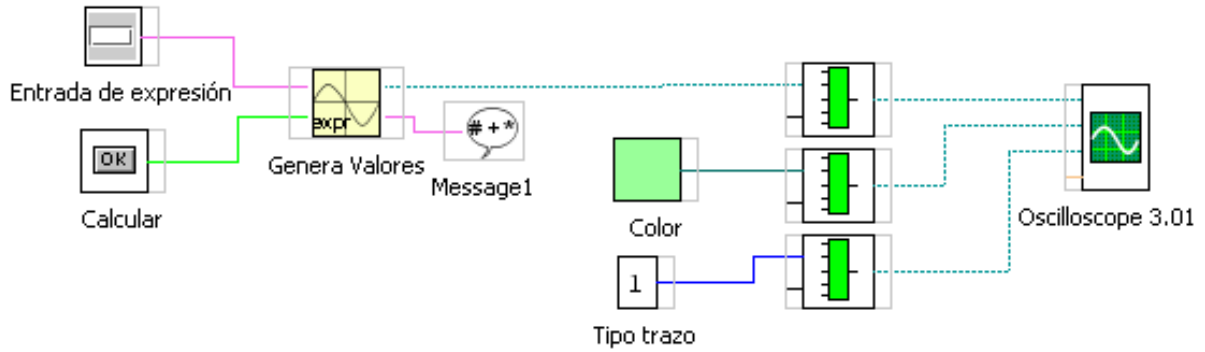


Figura 15

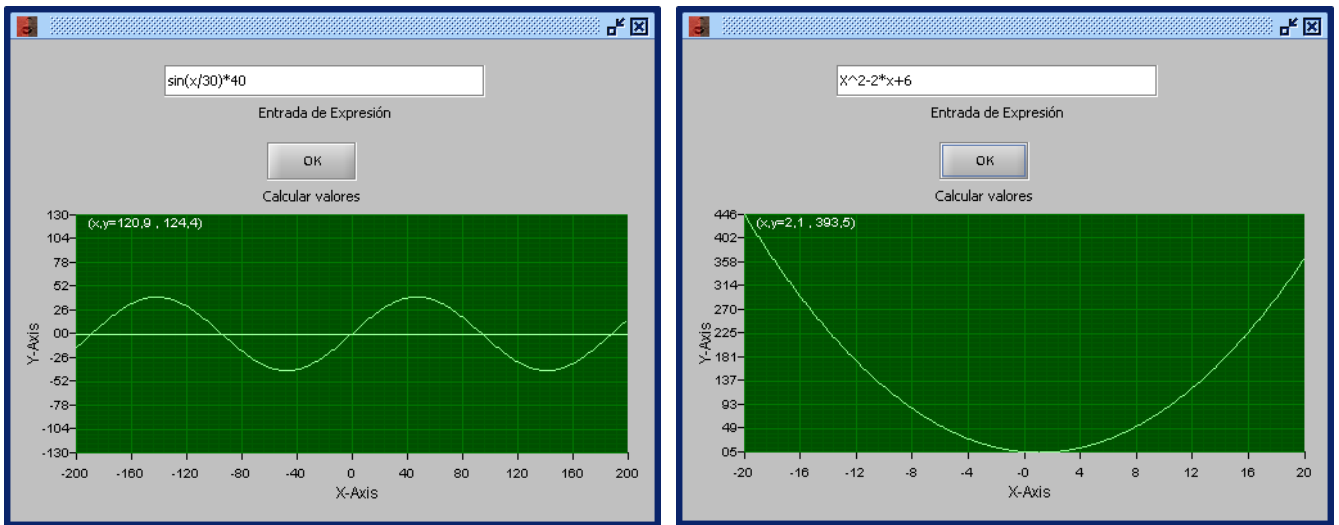


Figura 16

La sintaxis a utilizar en la entrada de la expresión es la propia de MyOpenLab y se muestra a continuación.

#### Operadores Unarios

#### Operadores Artiméticos

+	Suma Operador Unario
-	Resta Operador Unario

#### Paréntesis

)	Cerrar paréntesis
(	Abrir paréntesis

#### Constantes

E	e	2.718281828459045
PI	p	3.141592653589793

#### Funciones Trigonométricas

Sin(x)	Seno
Cos(x)	Coseno
Tan(x)	Tangente
aSin(x)	arcSeno
aCos(x)	ArcCoseno
aTan(x)	Arc tangente
Sinh(x)	Seno Hiperbólico
Cosh(x)	Coseno Hiperbólico
Tanh(x)	Tangente Hiperbólica

+	Suma
-	Resta
*	Multiplicación
/	División
^	Potenciación. Ejemplo: $3^2 = 3^2 = 9$
%	Modulo (Resto División): $7\%2=1$

#### Otras Funciones

Abs(x)	Valor Absoluto $ x $
Log(x)	Logaritmo decimal (base 10)
Ln(x)	logaritmo Neperiano (base e)
Exp(x)	Exponencial $e^x$
Sqrt(x)	Radical x.
Round(x)	Parte entera
Fak(x)	Factorial $x!$
toDeg(x)	Convierte radianes a grados
toRad(x)	Convierte grados a radianes

**TABLA 1**

### Varias funciones en un mismo sistema de representación.

En el caso de que queramos representar varias funciones en un mismo grafico podremos hacerlo simplemente haciendo uso de la función “Agrupar”.

En la figura vemos el caso de la representación de dos señales.

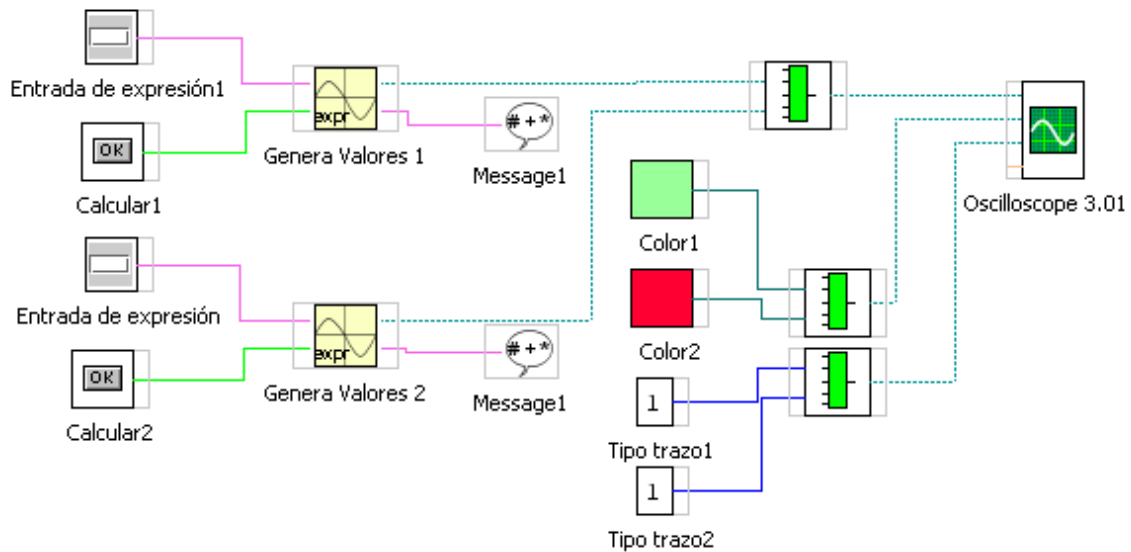


Figura 17

El aspecto del panel de visualización sería el que se muestra en la figura. Cada uno de los dos botones “Calcular1 y Calcular2” sirven para validar la ecuación que coloquemos en las cajas de edición de texto, de acuerdo con la sintaxis establecida.

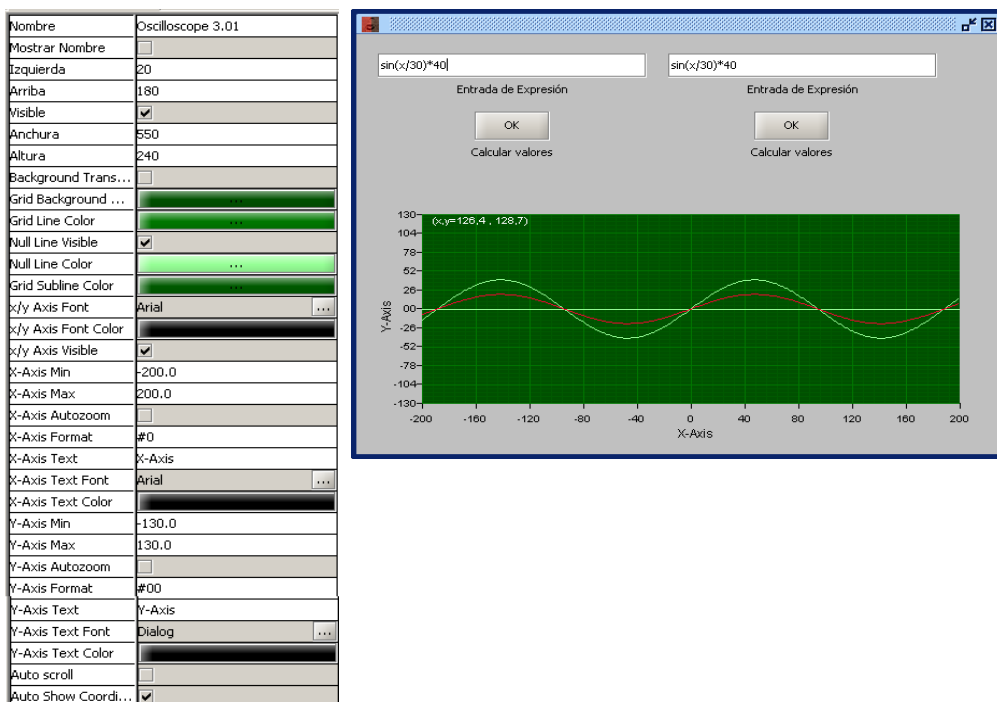


Figura 18

A la hora de representar las graficas se debe prestar atención a los parámetros del trazador grafico “Osciloscopio 3.01” Con la actuación en su parámetros se consigue situar el grafico en el punto adecuado y modificar el zoom, escalado. Rejilla. Etiquetas de ejes, etc.

En el capítulo correspondiente a la descripción de este objeto grafico explicaremos sus atributos de forma pormenorizada.

### 3.2. Método 2: (Bloque 1D Genera Gráfico)



En ocasiones necesitamos no solo mostrar el gráfico de una función sino también poder sacar en una tabla los valores calculados.

Para esta necesidad se puede recurrir al bloque “1D Genera Gráfico” que es muy similar al anterior pero presentando en su salida un array de datos tipo 1D que contiene todo los valores calculados y que se podrá representar gráficamente y además mostrar los valores en una tabla. A continuación mostramos el esquema funcional de esta aplicación.

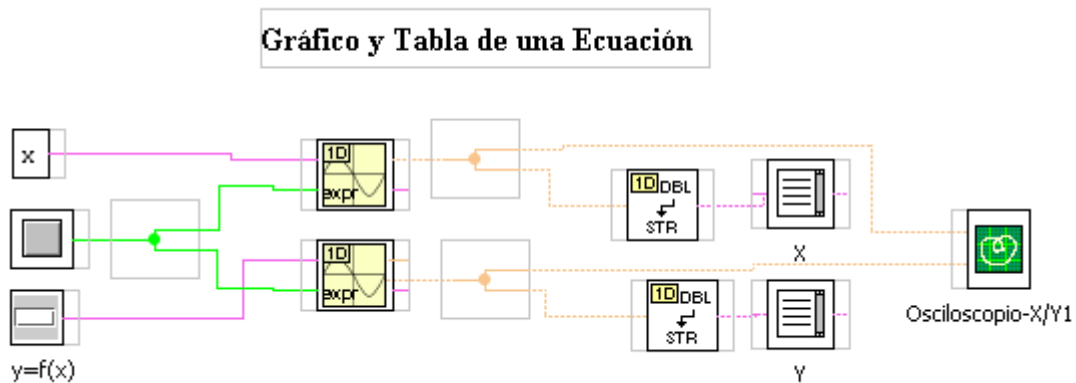


Figura 19

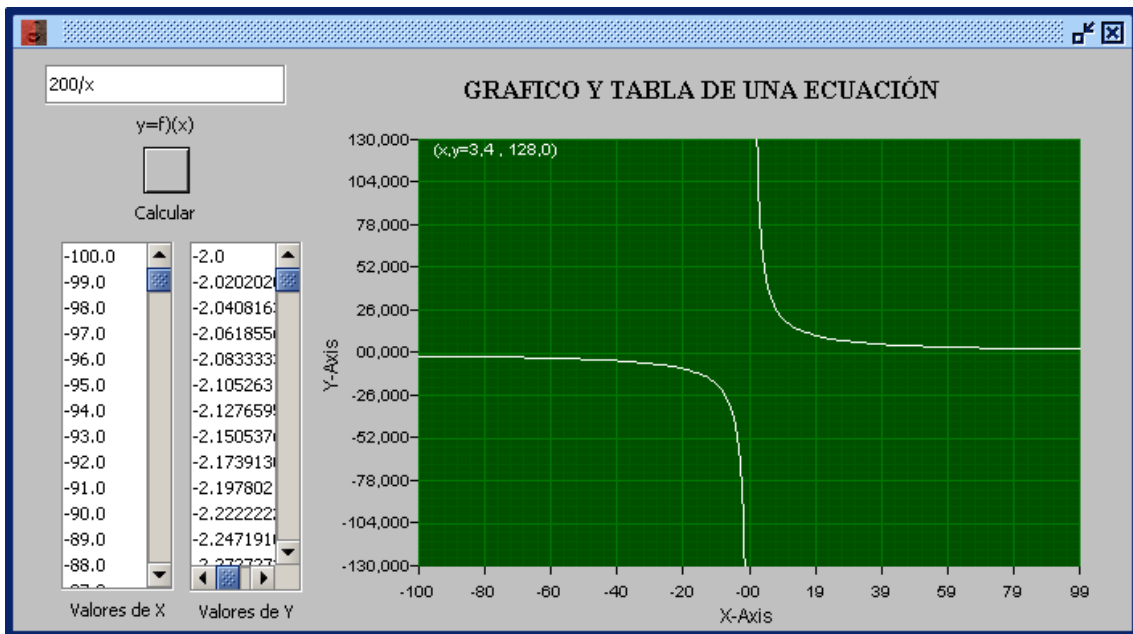


Figura 20

LOS PARAMETROS DE LOS BLOQUES “1d Genera Función” son los mostrados en la figura. Evidentemente los valores del entorno de cálculo han de ser los mismos ya que el valor de X es sencillamente el valor de la variable independiente.

1D Genera Gráfico1 [1D Genera Gráfico]	
Editor de Propiedades	
Nombre	1D Genera Gráfico1
Mostrar Nombre	<input type="checkbox"/>
Izquierda	185
Arriba	90
Anchura	55
Altura	36
Min-X	-100.0
Max-X	100.0
Paso	1.0

Observar que los valores generados por el bloque de calculo son del tipo “dbl” y los que se escriben en las tablas x e y son de tipo “string”, por lo tanto se debe realizar una



conversión mediante el bloque

Figura 21

## 4. CALCULANDO CON MyOpenLab

### 4.1. Calculo utilizando el objeto “Calculadora”

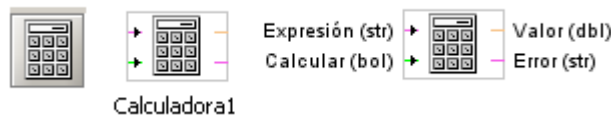


Figura 22

Este bloque de función recoge una expresión numérica en su entrada “Expresión” y cuando damos la orden en su entrada booleana “calcular” se realiza el calcula y se entrega el resultado en la salida “Valor”

La sintaxis para escribir la expresión debe ajustarse a la que se especifica en la Tabla 1

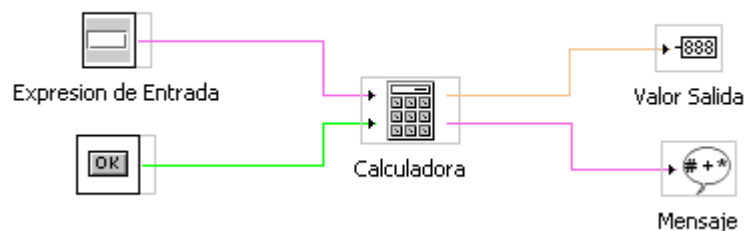


Figura 23

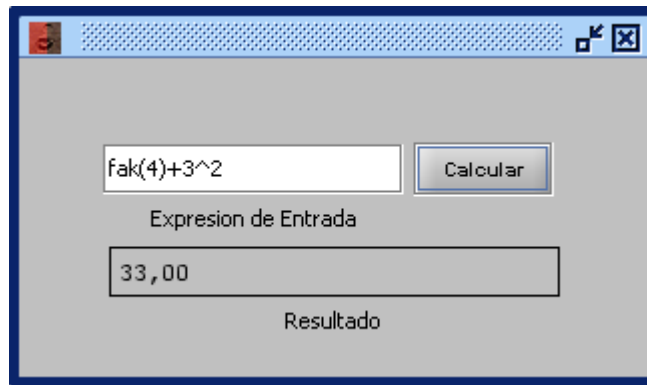


Figura 24

En la figura 25 vemos el esquema del modelo para realizar cálculos. La caja de entrada de texto nos permite escribir la expresión matemática y el botón “OK” nos permite ejecutar la orden de cálculo.

La función fak(n) calcula el factorial del numero n

En la figura 24 vemos el aspecto de la aplicación en modo ejecución.

#### 4.2. Cálculo utilizando el objeto “Calculadora y=f(x)”

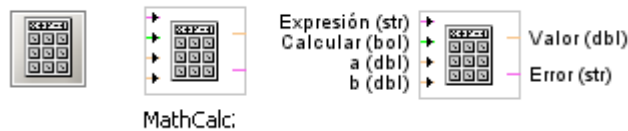


Figura 25

Esta calculadora a diferencia de la anterior permite el cálculo con un número determinado de variables de entrada que se podrá definir en la ventana de edición de parámetros a través del valor “Pins” que representa el número de variables de entrada que tendremos.

Las variables se nombran de manera fija empezando por la a y continuando con las letras del alfabeto.

**Variables de entrada:** a, b, c, ....  
**Variable de salida:** Valor

La habilitación del cálculo se realiza mediante la entrada booleana “**Calcular**”.

Veamos un ejemplo en el que se trata de resolver las soluciones de una ecuación de segundo grado dados sus coeficientes:

$$a*x^2 + b*x + c = 0$$



Sabemos que las soluciones serían

$$x1 = \frac{-b + (b^2 - 4ac)^{1/2}}{2a}$$

$$x2 = \frac{-b - (b^2 - 4ac)^{1/2}}{2a}$$

En este caso tendremos 3 parámetros de entrada: a , b , c

Como tenemos dos soluciones utilizaremos un bloque de calculo para cada una de ellas.

Vemos que las entradas de los parámetros a, b y c se realizan mediante cajas de texto y después conversión de este (string) a un dato de tipo dbl. No olvidemos que la calculadora trabaja con parámetros de tipo “dbl”.

En las figuras 26 y 27 se muestran la ventana de circuito (modelo) y la ventana de presentación (en modo ejecución) respectivamente.

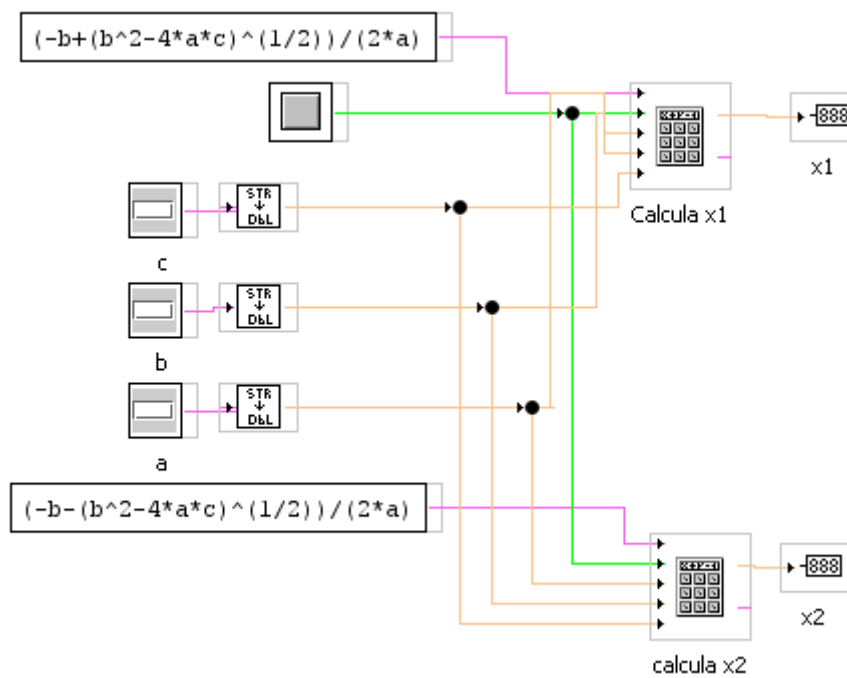


Figura 26

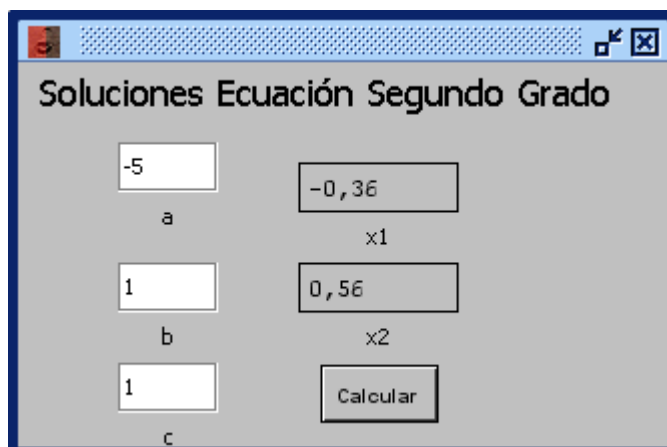


Figura 27

### 4.3. Cálculo utilizando el objeto “Calculadora ext.”

MyOpenLab tiene una función que nos permite calcular expresiones matemáticas con varias variables independientes.

En el ejemplo de la figura vemos como el operador “Calculadora ext1” tiene por entradas dos variables “x” e “y” y en su entrada tiene colocada una caja de entrada de texto a través de la cual escribiremos la expresión que deseamos calcular.

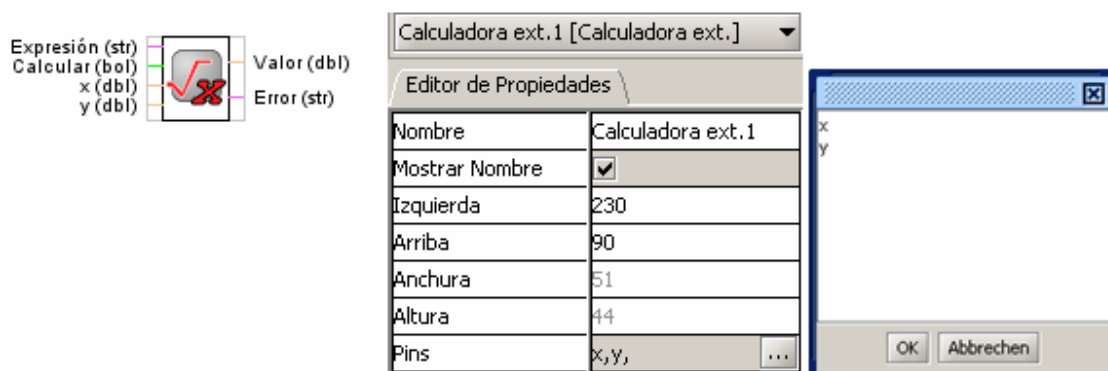


Figura 28

La función calculadora puede tener varias variables de entrada, todas ellas de tipo dbl (entero decimal) y en su salida presenta un valor de acuerdo con la expresión matemática que coloquemos en su entrada “Expresión”. El número de entradas se pone a través de la ventana “Editor de Propiedades” mediante el valor “Pins”. Si pulsamos sobre esta propiedad se despliega la ventana de edición de entradas y ponemos cada entrada en una línea.

En el ejemplo se realiza un calculador genérico que utiliza dos variables “x e y” mostrándose el resultado en la salida. Las entradas x e y las simulamos mediante dos sliders de valores 0 a 100.

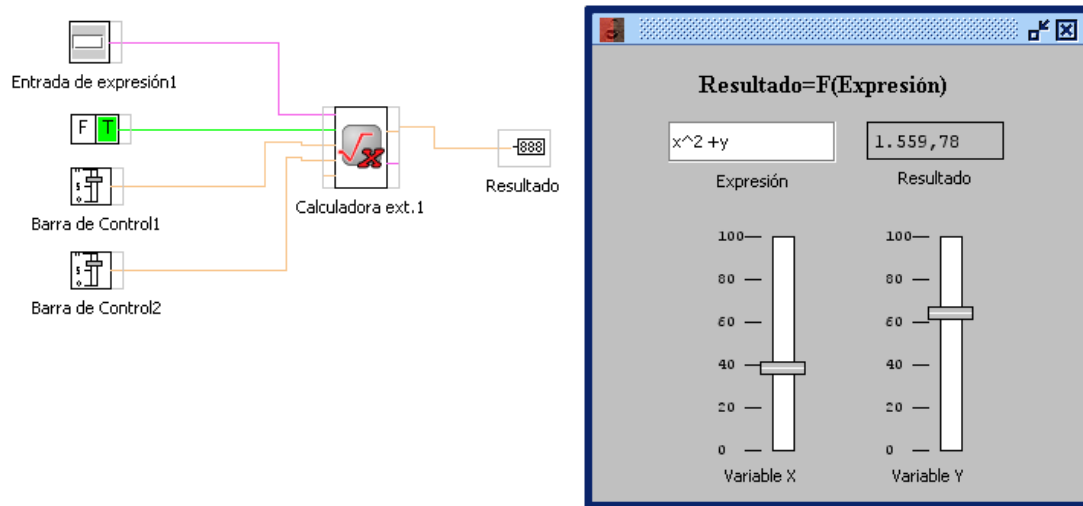


Figura 29

También se pueden realizar cálculos de expresiones con otros objetos de la librería de cálculo. Veamos como.

#### 4.4. Ejemplos de Cálculo:

##### ***Ejemplo1: Conversión de temperaturas***

Con este ejemplo vamos a implementar un modelo de cálculo muy sencillo en el que nos vamos a valores de los operadores de cálculo básico que se incorporan en la librería numérica de MyOpenLab.

Se trata de realizar una aplicación que transforme los grados centígrados en grados Fahrenheit y en grados Kelvin.

El dato de entrada del modelo será:

**T**= Temperatura en °C

Los datos de salida serán :

**F**= Grados Fahrenheit

**K**= Grados Kelvin

Las fórmulas que se utilizan para estas conversiones son las siguientes:

$$K = C + 273$$

$$F = C * (9/5) + 32$$

En la figura 30 se muestran la ventana de circuito con el modelo completo. En la figura 31 se muestra la ventana de visualización estando el modelo en modo ejecución.

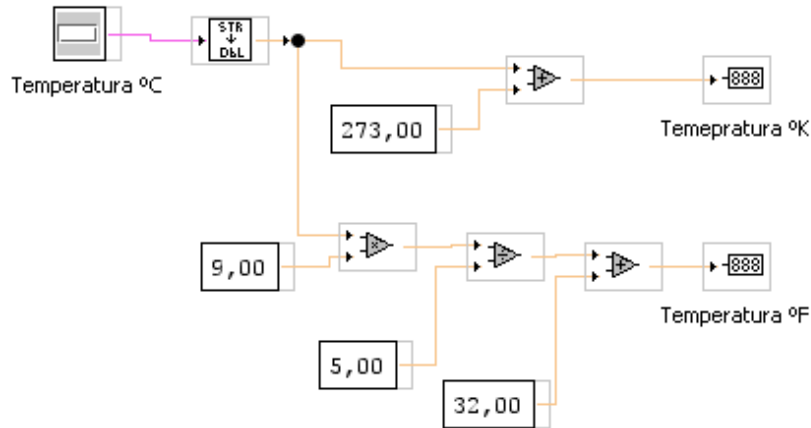


Figura 30

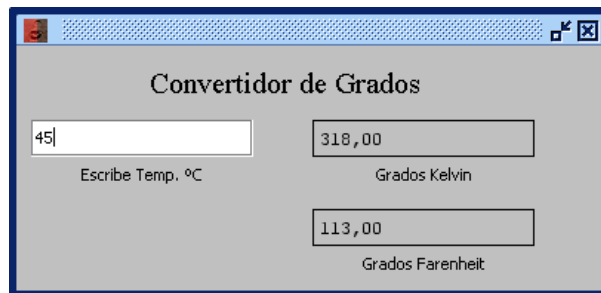


Figura 31

#### 4.4.1. **Ejemplo1: Cálculo del valor de la resistencia de un hilo en función de su longitud y su sección.**

En este ejemplo vamos a estudiar la variación del valor de la resistencia eléctrica de un conductor en función de su longitud, de su sección y de su resistividad.

Las variables independientes del modelo son:

- L = Longitud del hilo conductor (m)
- D = Diámetro del hilo (mm)
- r = Resistividad en ( $\Omega$ mm/m)

La variable dependiente en este caso es una sola:

R= Resistencia del conductor  $R = (r * L) / \pi * (d/2)^2$

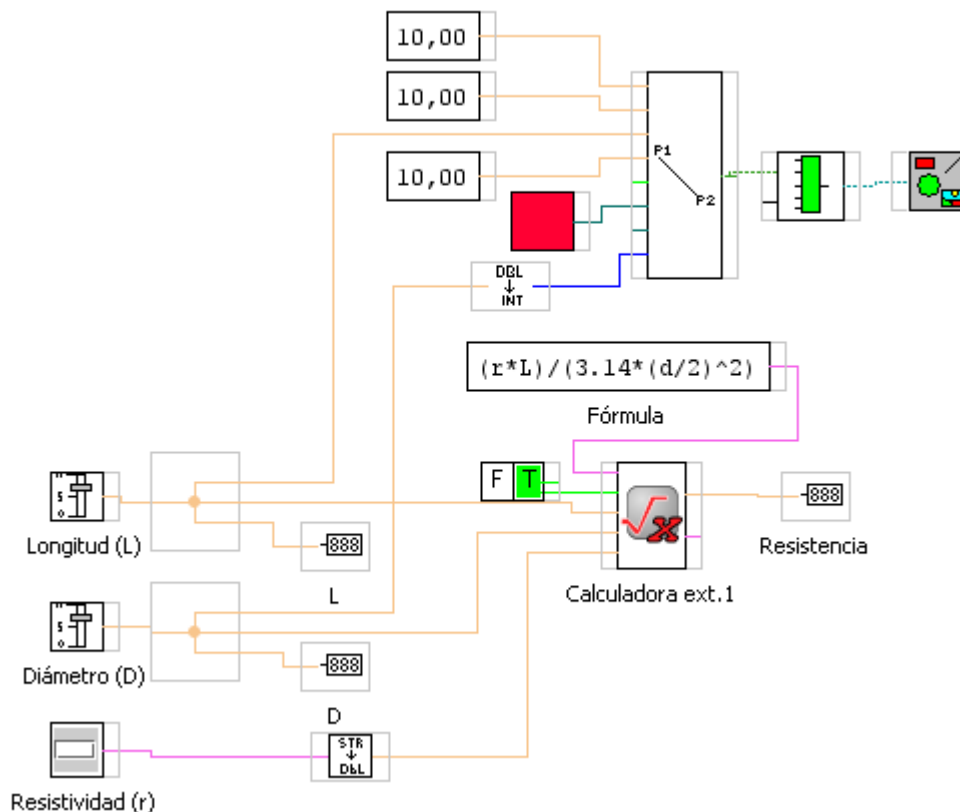


Figura 32

Vemos en el Panel de Circuito la “Calculadora ext.1” a la que se conectan las tres variables independientes que se simulan con dos sliders (L y D) y una caja de entrada de texto ( r ) que antes de conectar a la calculadora previamente convertimos de str-> dbl.

La formula se la ponemos a la calculadora mediante un objeto de entrada de texto (str).

De cara a facilitar la comprensión del fenómeno se ha dibujado una línea con el correspondiente objeto canvas “Draw Line” sobre la que controlamos la longitud (variable X2 que simula la longitud) y su espesor (variable stroke width que simula el diámetro del hilo). Estas variables lógicamente se recogen de los correspondientes sliders que controlan las variables L y D del modelo.

En la siguiente figura vemos el aspecto del Panel de Visualización en modo simulación.

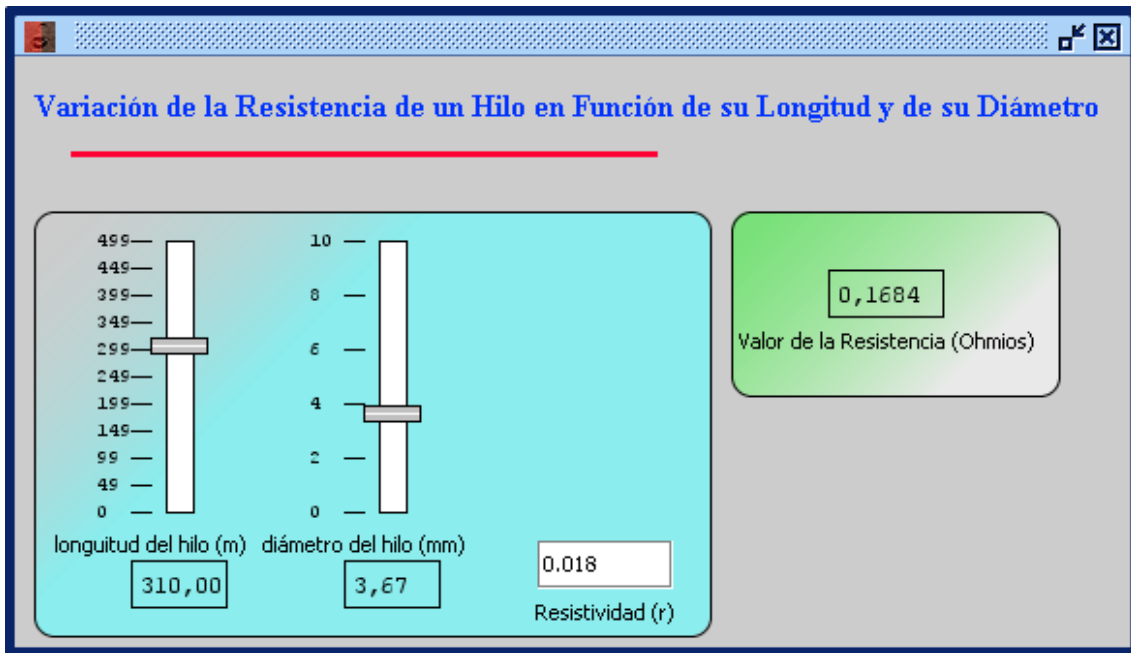


Figura 33

#### 4.4.2. **Ejemplo2: Cálculo del espacio recorrido por un móvil.**

En el siguiente ejemplo vamos a simular el comportamiento a lo largo del tiempo de un móvil afectado por un movimiento uniformemente acelerado.

Para abordar el problema distinguiremos dos variables a calcular: El espacio  $X$  y la velocidad  $V$ . Como sabemos ambas variables se calculan de acuerdo a las siguientes ecuaciones:

$$X=V_0*t+1/2*a*t^2$$

$$V=V_0+a*t$$

Distinguiremos claramente las variables dependientes y las independientes así como los parámetros o valores que consideremos constantes.

**Variables Independientes:**

t=tiempo

**Variables dependientes:**

v= Velocidad instantánea


X= Espacio recorrido

**Parámetros constantes:**


V= Velocidad inicial

a= Aceleración

Para realizar el modelo matemático del fenómeno haremos uso de dos bloques

funcionales “Calculadora ext.”  Y cada uno de ellos se encargara del cálculo de cada una de las dos variables dependientes.

La variable independiente (el tiempo) la vamos a simular haciendo uso de un bloque

generador de variable analógica “Generador” .

Este bloque nos permitirá generar un valor analógico del tipo decimal (dbl) de acuerdo a sus parámetros de entrada. En este caso decidimos que de sus parámetros de configuración solo sea configurable el valor máximo de cuenta denominado “to” en el componente. Los parámetros del generador quedarían de la forma: from=0 ; step= 1 ;delay= 100

Esto significa que contaremos el tiempo empezando desde 0 saltando de 1 en 1 con un tiempo entre un número y el siguiente de 100 ms.

En la figura vemos como quedan los dos bloques de cálculo. La variable “t” les llega a los dos del mismo generador como es lógico. Los bloque generadores tiene tres señales de gobierno que se generan mediante tres botones pulsadores:

**Start** = Inicio de la cuenta de tiempo

**Stop** = Parada de la cuenta

**Reset** = Puesta a cero del contador (inicio de la variable tiempo)

Los valores de los parámetros “a” y “Vo” los podremos modificar en modo simulación mediante una caja de entrada de texto. Este texto luego lo pasamos a variable tipo dbl y las conectamos a los bloques de cálculo.

El modelo quedaría totalmente definido con estos bloques, pero será útil poder mostrar los valores de las variables. Para ello hemos recurrido a varios objetos de visualización:

- Barra de Color:** Para visualizar el espacio
- Indicador de aguja:** Para la Velocidad
- Trazador Grafico temporal:** Para el espacio
- Salida Numérica:** Para el espacio

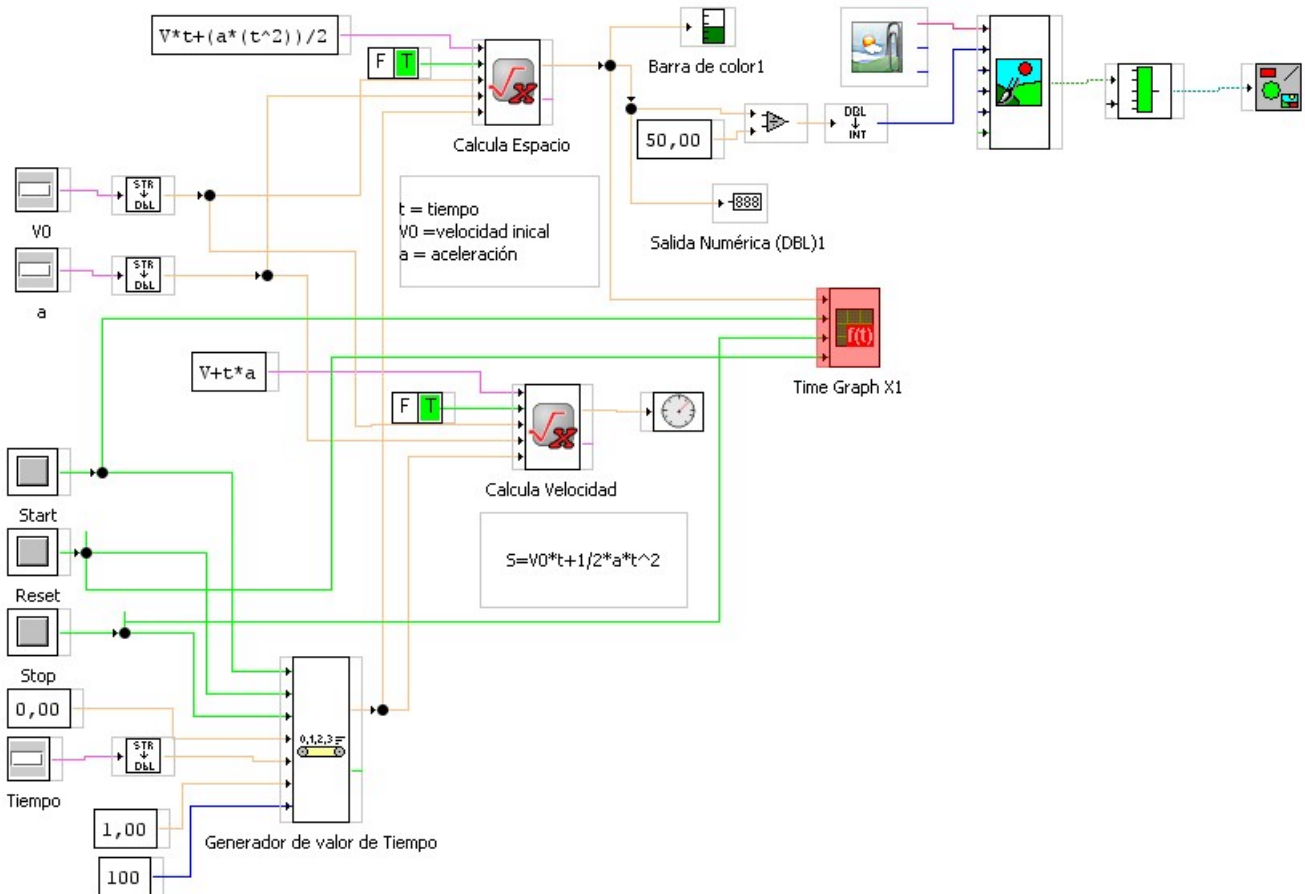


Figura 34

En el modelo se ha colocado también un objeto (carrito) que se desplazará en función del espacio en el sentido positivo del eje x (horizontalmente), para ello hemos recurrido a la librería de objetos canvas de donde se ha utilizado el objeto "Draw Image" así como el preceptivo objeto que crea el espacio canvas "Canvas" y el necesario objeto de agrupación "Agrupar"

En la figura vemos los objetos que se encargan de mostrar el carrito en movimiento.



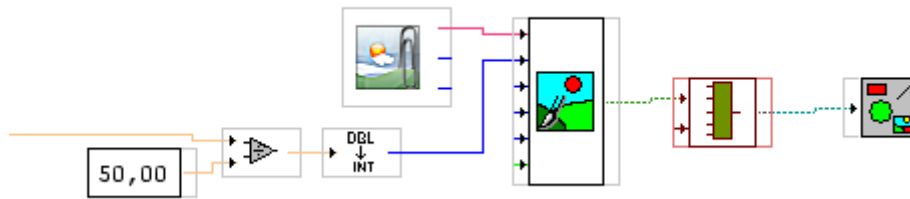


Figura 35

Finalmente la pantalla de visualización en modo ejecución es la que se muestra a continuación.

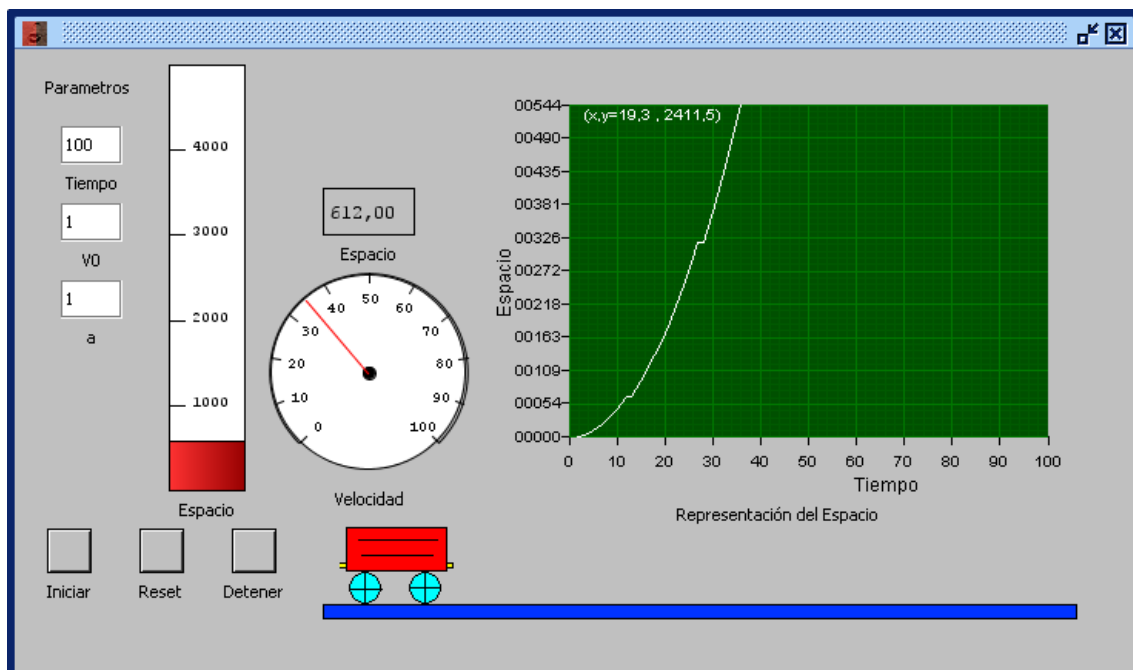


Figura 36