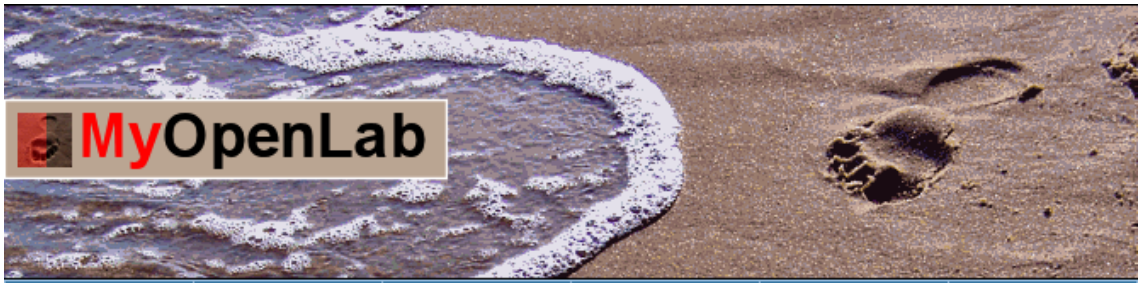


MyOpenLab

Tratamiento de Datos

Versión 2.4.8.3

Documentación para el usuario

www.MyOpenLab.de

Prof. José Manuel Ruiz Gutiérrez

Índice

FORMATOS Y TRATAMIENTO DE LOS DATOS CON MyOpenLab

1. Introducción.

En este apartado vamos a estudiar los distintos tipos de datos que puede manejar la aplicación MyOpenLab y sus formas de conversión.

La naturaleza de los datos y sus conversiones es fundamental en un entorno de simulación.
MyOpenLab tiene dos librerías en las que se abordan conversiones de datos y su tratamiento.

Antes de nada conviene tener en cuenta que en la ventana de componente se muestra en cualquiera de los componentes de MyOpenLab los datos de entrada y de salida y su naturaleza. Esto vale también para los nuevos bloques de función que podamos crear.

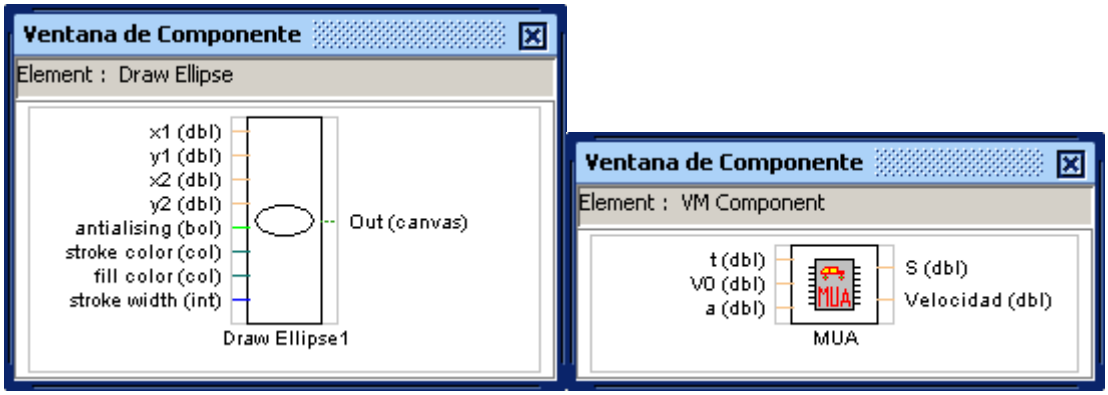


Figura1

En la figura 1 vemos los datos del componente Elipse de la librería canvas y los datos del componente VM MUA creado por nosotros.

2. Tipos de Datos en MyOpenLab.

En la aplicación podemos encontrar los siguientes tipos de datos:

- Double (**dbl**)
- Integer (**int**)
- String (**str**)
- Booleano (**bol**)
- Var (**var**)
- Grupo (**grp**)
- Fuente (**font**)
- Color (**col**)
- Matrices de valores 1D

- Matrices de valores 2D

En el diseño de los circuitos se colorean las entradas y salidas de los distintos elementos con un código de colores que nos permite saber en todo momento de que tipo son los datos en un determinado bloque de función.

En la figura 2 se muestra una imagen con los códigos.

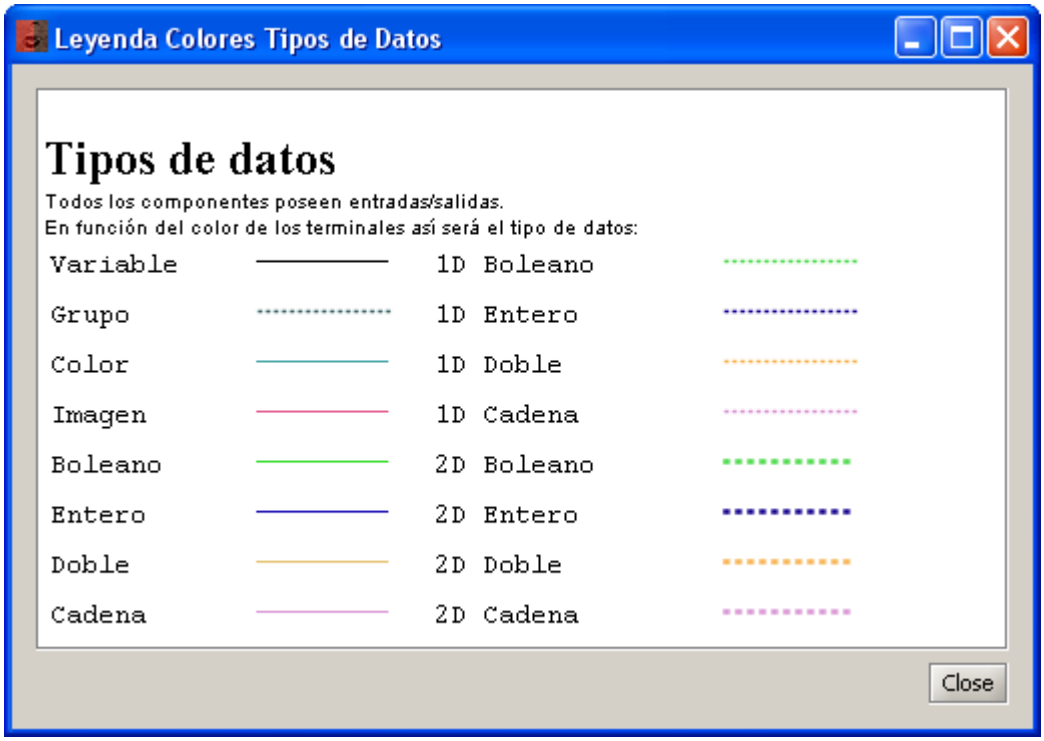


Figura 2

2.1. Datos Tipo “double”

Los tipos de datos Double son considerados por defecto con dos decimales. En el caso de aportar datos con más decimales se producirá el redondeo a la hora de mostrar el dato, si bien lo tendrá almacenado con sus decimales y operara con todos ellos:

Ejemplo

Dato Introducido	*Dato Aceptado*
12.342	12.34
12.348	12.35

2.2. Datos Tipo “integer”

Los tipos integer son variables numéricas enteras. Su utilización en la creación de modelos será en aquellos casos en los que no tiene sentido hablar de decimales por ejemplo en un contador de impulsos.

La longitud máxima que admite el simulador es de +/- 999999999

Visualizar mas de dos decimales.

En el caso de que queramos visualizar más de dos decimales en un objeto “Salida Numérica” bastara con que cambiemos en la ventana de propiedades el valor del campo “Formato”. Figura 3

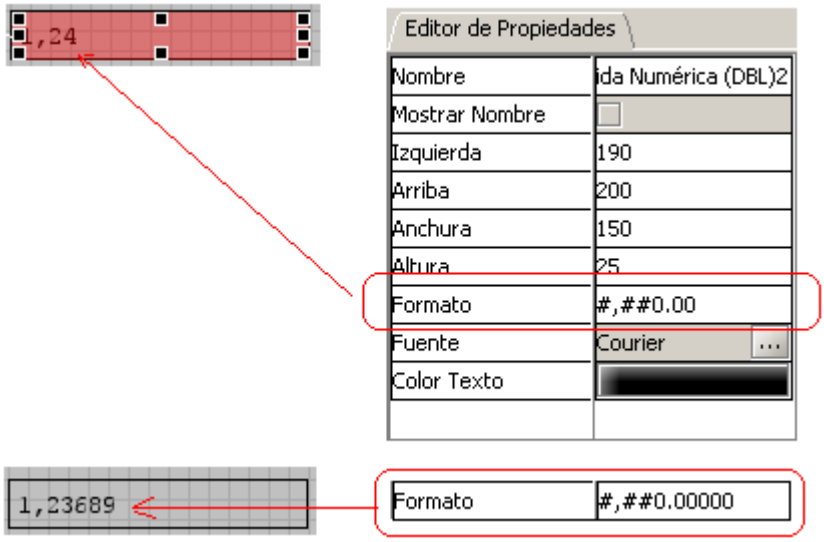


Figura 3

2.3. Datos Tipo “string”

Este tipo de datos consiste en cualquier cadena de texto que incorpore letras, números y signos.

Ejemplos de este tipo serían:

As23,89qs
_f_765^4.gg
MyOpenLab

2.4. Datos Tipo “bol” (booleano- digital).

Estos datos son los que pertenecen a los sistemas y operadores digitales que como sabemos son los elementos “0” y “1”.

El elemento  

perteneciente a la librería de Operadores digitales sirve para fijar a cero o 1 una entrada de un componente bastara con marcar F(false)=0 o T(True)=1. Figura 4.

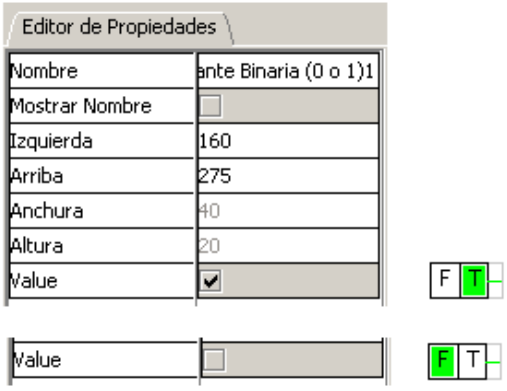


Figura 4

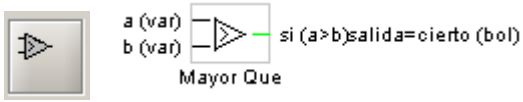
A continuación se muestran los elementos de entrada y de salida utilizados en el Panel de Visualización para este tipo de dato.

Entrada **Salida**

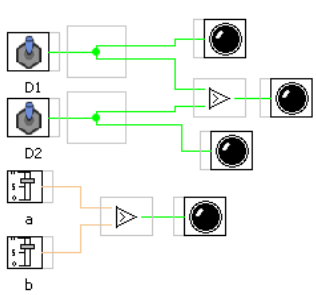
2.5. Datos Tipo "var".

Determinados bloques de MyOpenLab tienen la posibilidad de aceptar distintos tipos de datos, produciéndose una adaptación del operador a las entradas. En estos bloques sus entradas y/o salidas vienen marcadas como tipo **"var".**

En la figura 5 vemos un ejemplo de este tipo de variables, concretamente en el bloque de la librería de comparadores **"mayor que"**



Pongamos un sencillo ejemplo en el que utilizamos una función de comparación.



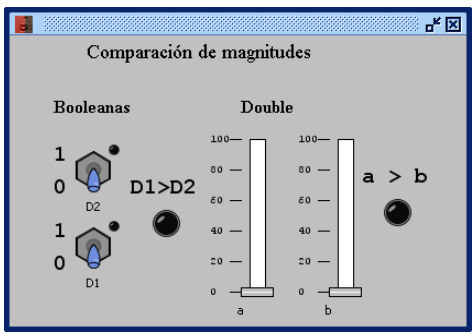


Figura 5

Vemos que en el caso del comparador de datos booleanos sus entradas son del tipo “bol” y en el caso de las variables analógicas es de tipo "double”.

2.6. Datos Tipo “grp” (grupo).

Este tipo de datos es un conjunto de datos que se transfiere de un bloque a otro y que de forma empaquetada esta formado por un número de datos individuales.

Como ejemplo de este tipo de datos tenemos: El elemento concentrador utilizado para incluir varios objetos en un área Canvas, la salida de un Bloque Generador de Frecuencia en la que se empaquetan los datos necesarios para poder visualizar la señal que genera., la entrad de un Osciloscopio para visualizar señales.

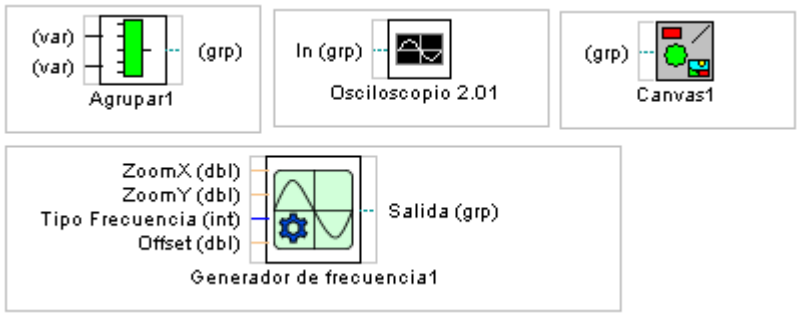


Figura 6

Las señales que forman parte de estos grupos se tratan de manera conjunta, es decir no se pueden sacar las señales y procesarlas independientemente al menos en la actual versión de MyOpenLab.

En la figura 7 siguiente vemos un sencillo ejemplo el que se hace uso de este tipo de datos agrupados. Se trata de visualizar una función matemática que se introduce mediante una caja de entrada de texto en un bloque de tipo MathCalc.

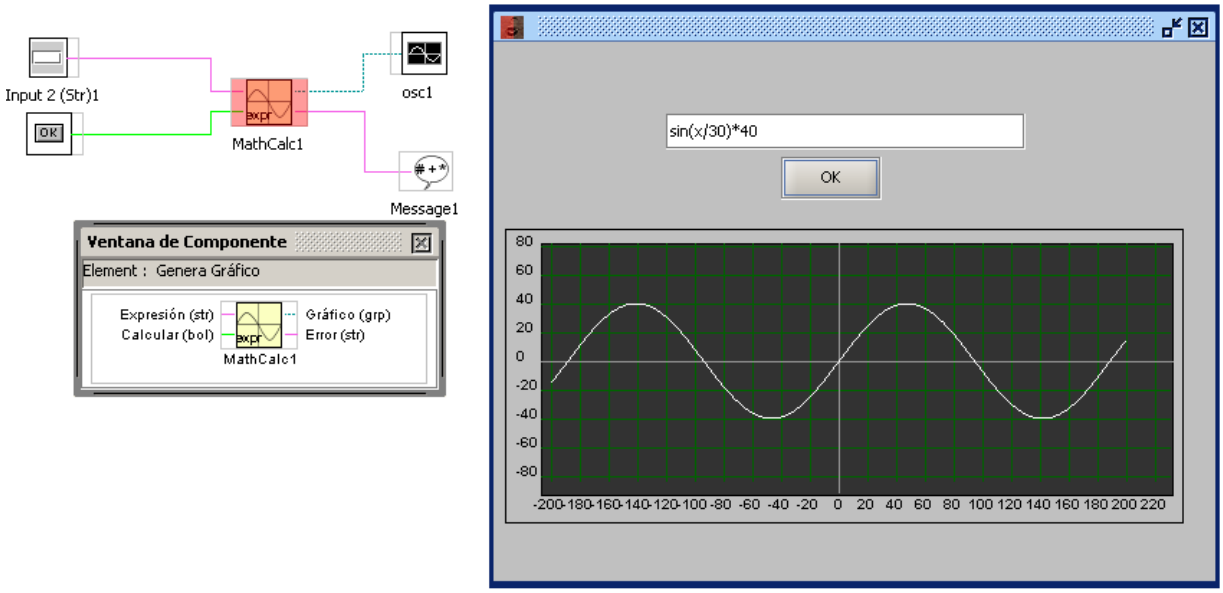


Figura 7

2.7. Datos Tipo “font” (fuente de letra).

Los datos tipo “font” se refrieren, como su nombre indica a los tipos de letra. Estos datos se utilizan para aquellos bloques funcionales que utilizan tipos de letra como es el caso del objeto Canvas.

En la figura 8 vemos como se introduce este tipo de dato al bloque de Mostrar Texto en un área Canvas.

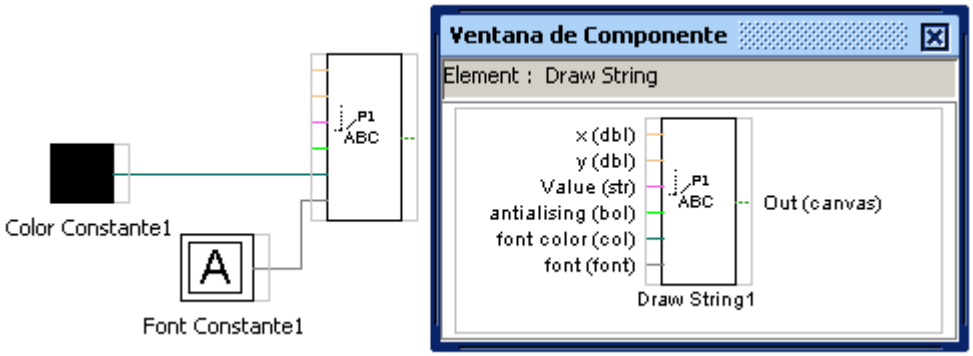


Figura 8

La selección del tipo de letra se realiza en el menú de propiedades del elemento Font Tal como se ve en la figura 9

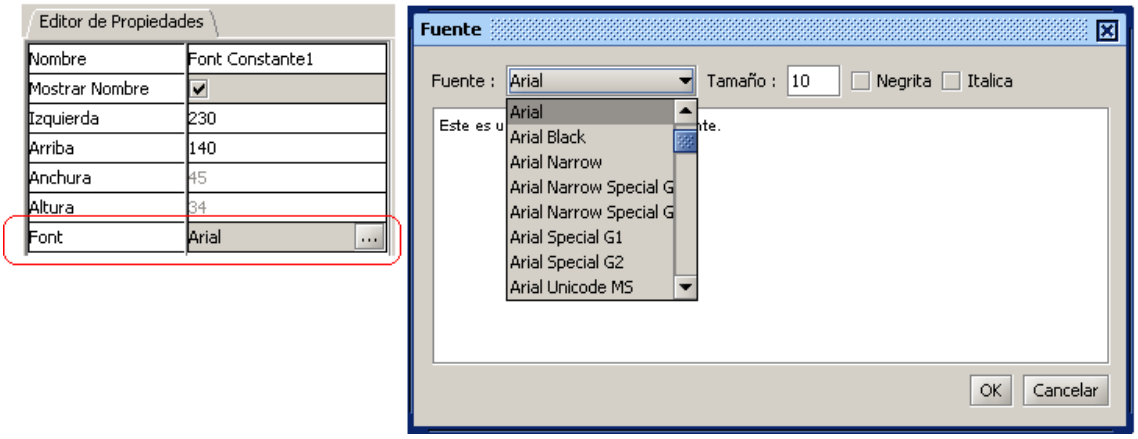


Figura 9

Al pulsar en el botón ... se despliega la ventana Fuente desde la que seleccionamos el tipo, tamaño y aspecto de la fuente.

2.8. Datos Tipo "col" (color).

Los tipos de datos "col" hacen referencia a la selección de color que mediante ellos podemos introducir en un bloque que requiera de este tipo de datos. En la figura 8 vemos que el bloque canvas se le pasa también un dato de esta naturaleza.

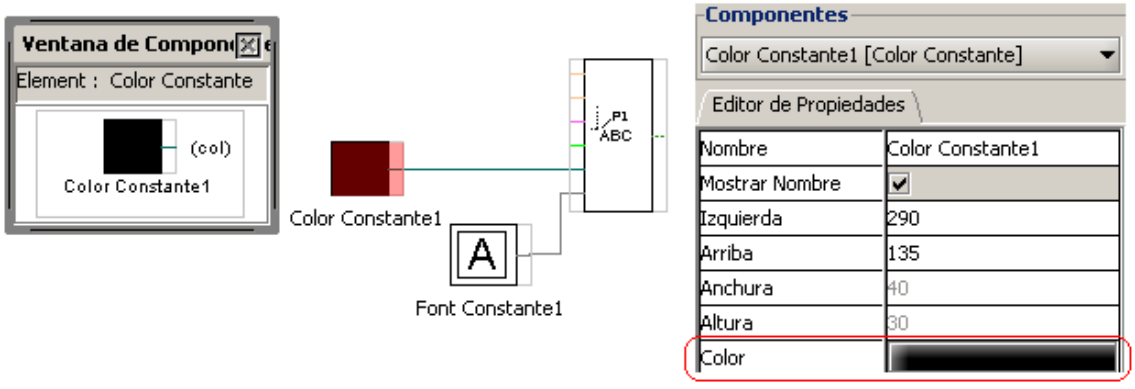


Figura 10

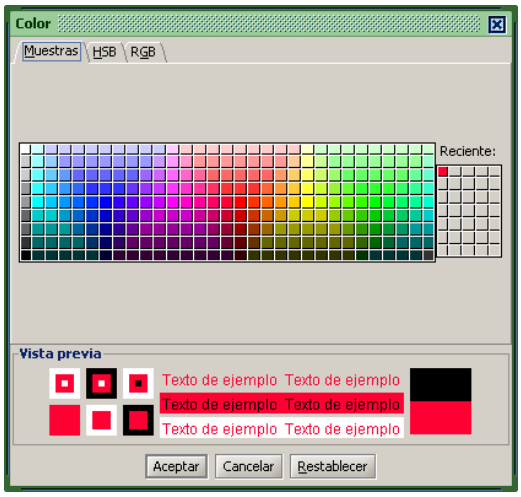


Figura 11

Cuando seleccionamos color en el menú de propiedades del objeto Color se despliega una paleta de colores de la que seleccionamos el color que queremos. Figura 11 .

Es importante destacar que la variable color se puede descomponer en tres números que corresponden con los calores RGB de la mezcla. Esto se realiza con el bloque "Color a RGB". En la figura 12 vemos como se descompone el color dando los tres valores.

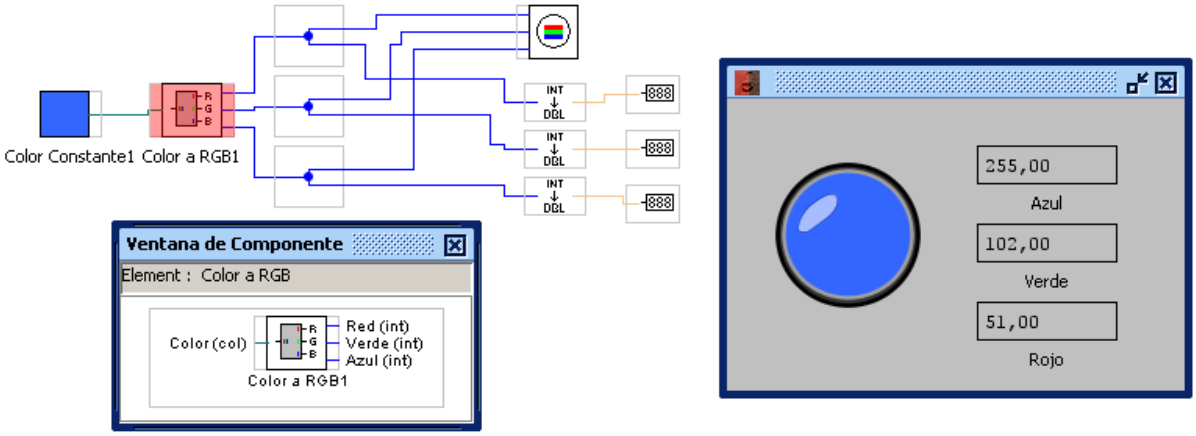


Figura 12

9.1. Datos Tipo "img" Imagen

Este tipo de datos en realidad responde a una imagen que se puede recuperar de un fichero de tipo (JPG,GIF,PNG, etc).

La mayor parte de las funciones que se presentan dentro de la librería **"Image"** del **Panel Circuit**o manejan este tipo de datos. En la figura 13 vemos algunas de estas funciones .

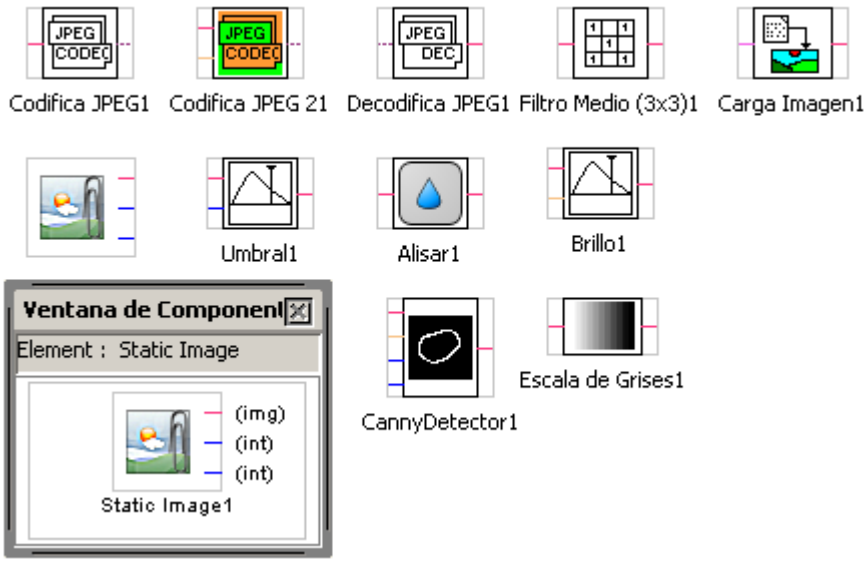


Figura 13

MyOpenLab es capaz de realizar interesantes operaciones con las imágenes, por lo que en el tema grafico adquiere una gran importancia dentro de las herramientas de modelización y simulación.

También existen bloques de función pertenecientes a la librería de elementos de salida del **"Panel Visualización "** que trabajan con datos de tipo "img". En la figura 14 vemos algunos.

Figura 14

En la figura 15 vemos una sencilla aplicación que carga tres ficheros de imagen y los muestra en el panel de visualización.

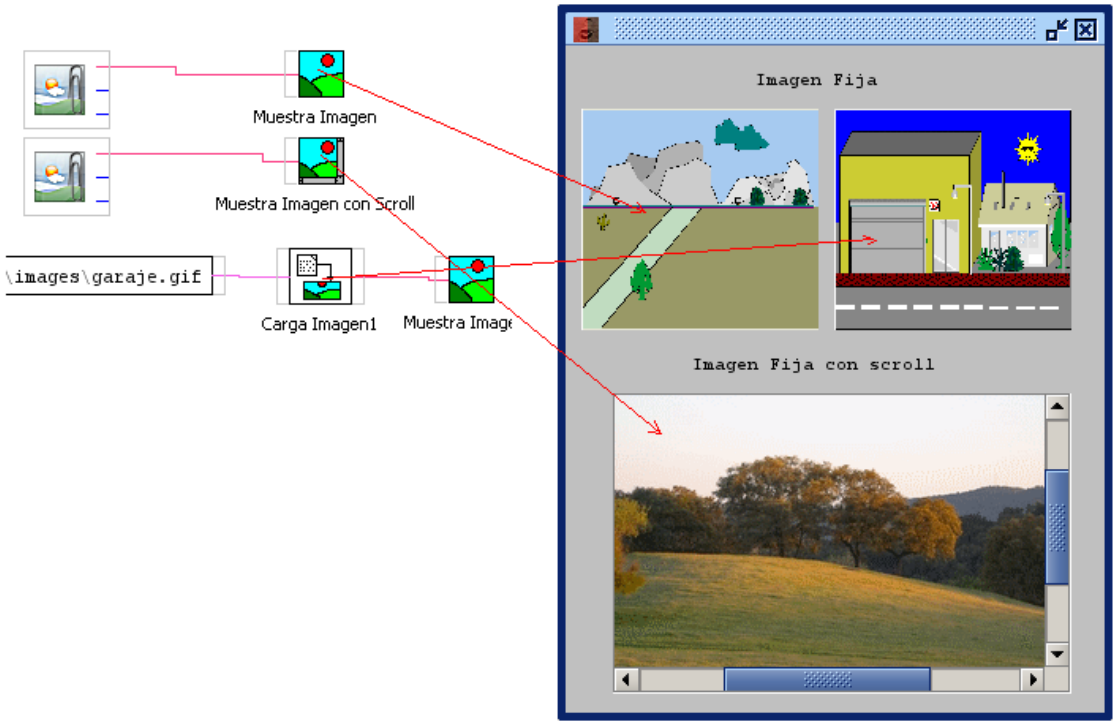


Figura 15

Conviene resaltar que las imágenes se pueden tratar mediante la utilización de los bloques de tratamiento de la imagen que se muestran en las figuras anteriores. Tratar la imagen significa actuar sobre sus condiciones y cualidades. Vemos un ejemplo a continuación (figura 16) en el que se ha recogido una imagen y se ha tratado mediante los bloques
"Escala de Grises" y "Alisar".

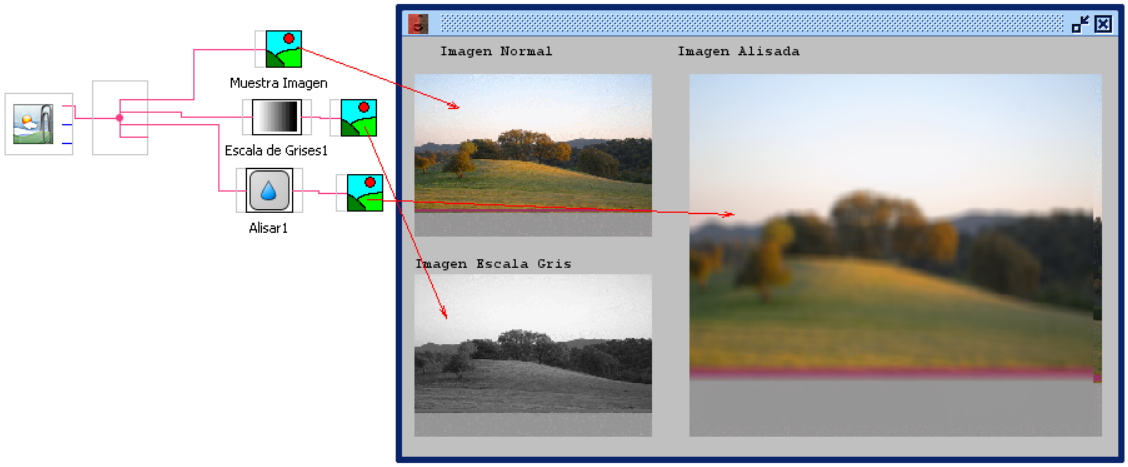


Figura 16

9.2. Array de valores 1D

Determinados bloques de MyOpenLab generan en sus salidas un Array de valores de tipo 1D, como es el caso del los bloques utilizados para generar señales

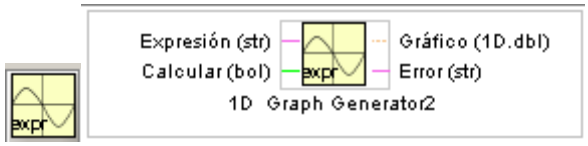


Figura 17

En realidad este bloque lo que haces es crear una tabla de valores que son los que luego recogerá un trazador grafico (osciloscopio) y mostrara.

Para ilustrar la idea de un Array de valores tipo 1D lo mejor es hacerlo con un ejemplo. A continuación se muestra un ejemplo en el que se visualizan señales provenientes de dos bloques generadores de señal (figura 18) . Estos bloques generadores de señal recogen una cadena de caracteres que representa una función (en este caso funciones trigonométricas) y una entrada de tipo booleano que sirve para habilitar o crear la tabla de valores de esa función. En este caso la tabla se crea entre 0 y 500 en pasos de 0.3

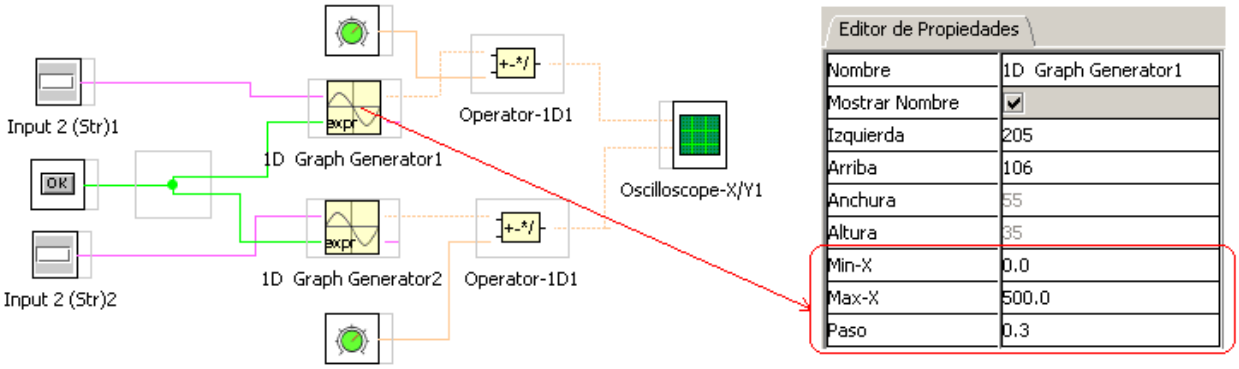


Figura 18

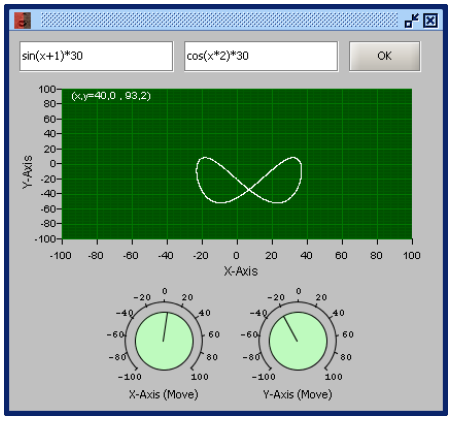


Figura 19

Los controles tienen por misión desplazar la señal trazada en el trazador en sentido vertical y horizontal.

9.3. Matrices de valores 2D

Una matriz de datos tipo 2D es una lista o array formado por un conjunto de datos que MyOpenLab será capaz de tratar de una sola vez haciendo uso de funciones específicas para ello. En la figura 20 vemos tres bloques que operen con este tipo de matrices de datos 2D.

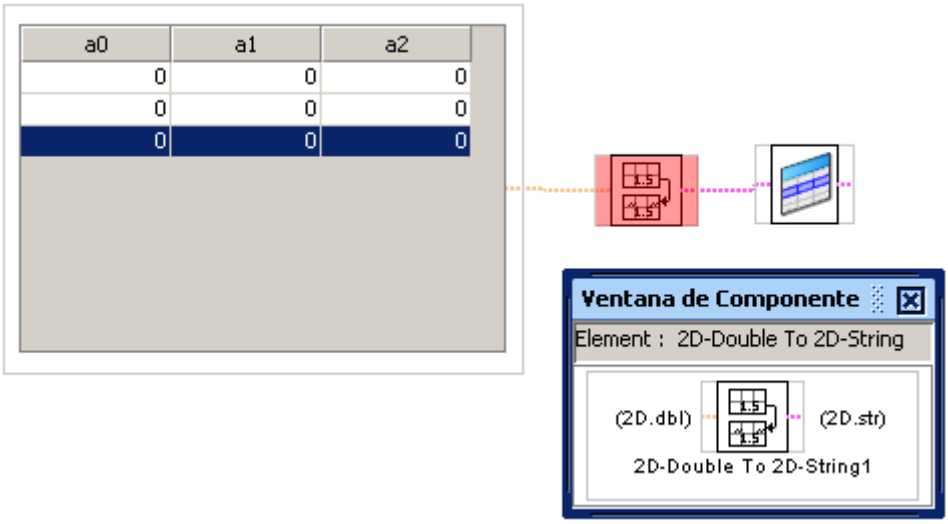
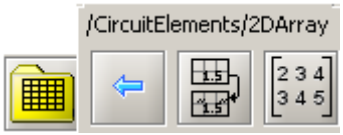


Figura 20

Lo que vemos en la anterior figura básicamente es un bloque de datos perteneciente a la librería 2D Array 

en el bloque se escribe los datos en forma de matriz n*m del tipo "dbl". A continuación pasa a un bloque que convierte los datos **2D-Double** a datos **2D-String** para posteriormente sacarlos mediante el bloque perteneciente al la librería de salida del Panel Visualización llamado **"table".** El ejemplo muestra claramente como MyOpenLab es capaz de tratar matrices 2D de datos**.**

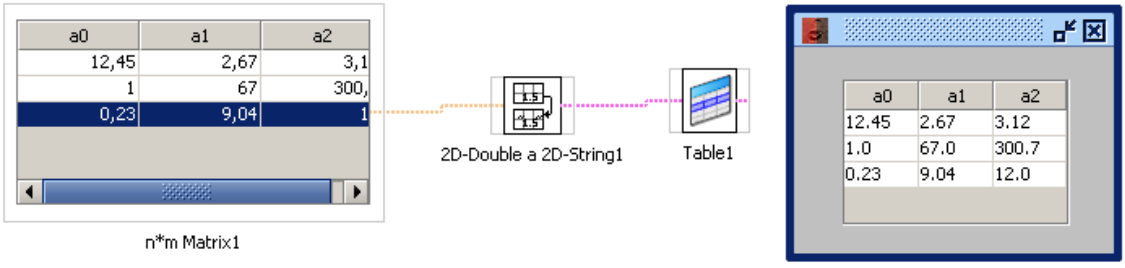


Figura 21

3. Conversiones de tipos double, integer y string

Las conversiones que se pueden realizar teniendo en cuenta este tipo de datos son:

Convierte una variable “integer” en double”

Convierte una variable “double” en “integer”

Convierte una variable “double” en un “string”

Convierte una variable “string” en una variable “double”

Convierte una variable “integer” en una variable “string”

Recordemos que estas funciones se encuentran en las librerías 

Los datos de este tipo se pueden introducir en una simulación en forma de constantes mediante las funciones:

Bloques de función de entrada de datos

Introduce una variable tipo “string”. Ejemplo: hola, 123asd, <>@#mm

Introduce una variable del tipo “double”: Ejemplo: 12,32

Introduce una variable del tipo “integer”. Ejemplo: 12


Genera un numero aleatorio entre 0 y 1 de tipo “double”

4. Consideraciones relativas a la visualización de los datos en el “Panel de Visualización”.

Es muy importante tener en cuenta que los elementos de salida de datos deben tener su entrada compatible con el tipo de datos que queramos ver. En este debemos considera que los elementos que permiten ver los tipos de variables son los clasificados en la tabla

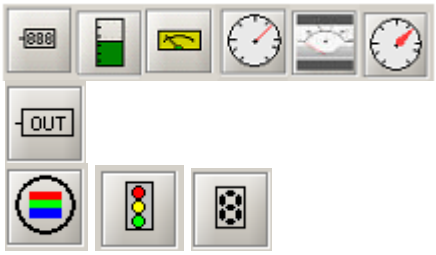
En la figura vemos la compatibilidad en la visualización de elementos (salida de datos).

Elementos de Visualización

Double

String

Integer



De la misma manera los datos generados por los elementos de entrada de datos de la librería correspondiente del “Panel de Visualización” también entregan un tipo concreto de datos que hemos de tener en cuenta ciando hagamos un diseño.

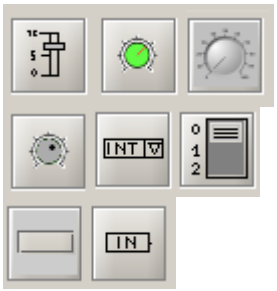
En la siguiente figura vemos cada uno de estos elementos que se colocan en el Panel de Visualización” para introducir datos en el modelo con los tipos de datos de cada uno de ellos:

Elementos de Entrada de datos

Double

Integer

String



En los casos que se requiera sacar un dato en un elemento de salida que no sea compatible con el se pude recurrir a realizar una conversión de datos. En la figura 22 vemos un sencillo ejemplo en el que mostramos la forma de visualizar un valor Integer generado por el potenciómetro en un instrumento de aguja que recoge valores tipo double.

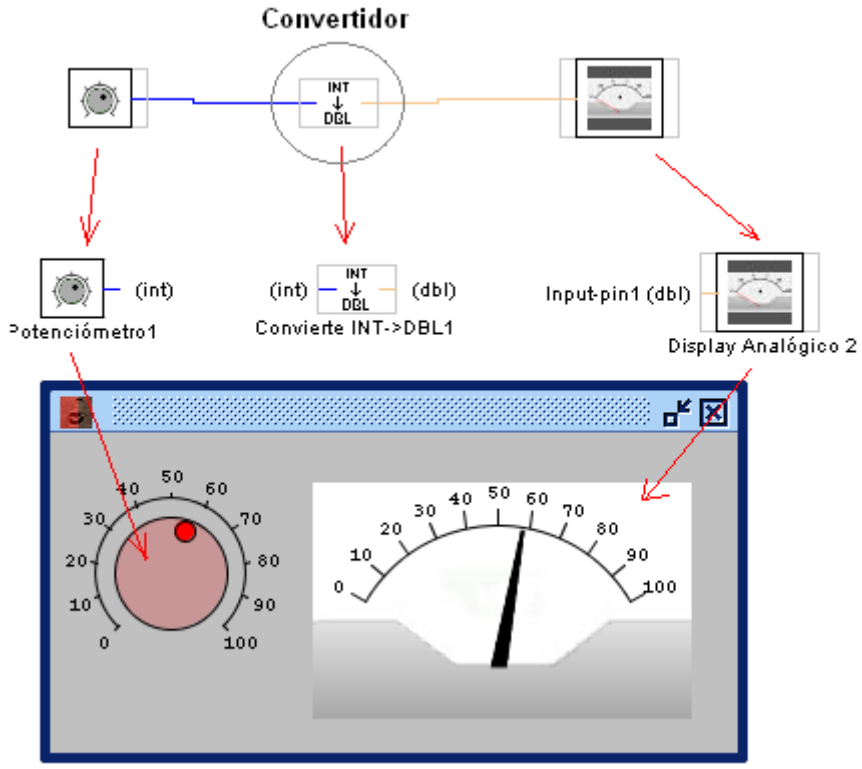


Figura 22

5. Establecimiento de formato para un dato tipo “double”.

En algunos casos nos puede interesar modificar el formato de un dato de tipo “double “ para poner o quitar más decimales. Para resolver esta cuestión se recurre ala función.

que mediante su entrada de formato (string) permite definir el formato que le damos al valor que pongamos en su entrada “value” de dato.

En el ejemplo de la figura 23 se introduce el dato por la entrada “value” 3.14166666 y se dice que solo tome cinco decimales por lo tanto la salida que visualiza es 3.14167 (vemos que el primer digito que se desprecia al ser mayor que 6 al anterior le aumenta una unidad.

Figura 23

6. Averiguar la longitud de una cadena “length”

Esta función 

es muy útil dentro del tratamiento de los datos de tipo string. La función recoge la variable cadena cuanta el numero de caracteres y lo devuelve en su salida.

En la figura 24 vemos un ejemplo en el que mediante una caja de entrada de texto colocamos la cadena en el la entrada de la función y la salida de esta (numero de caracteres) que es un dato tipo Integer lo pasamos a un convertidor de formato del tipo explicado anteriormente y de este, una vez dado el formato lo sacamos mediante una caja de visualización de cadenas.

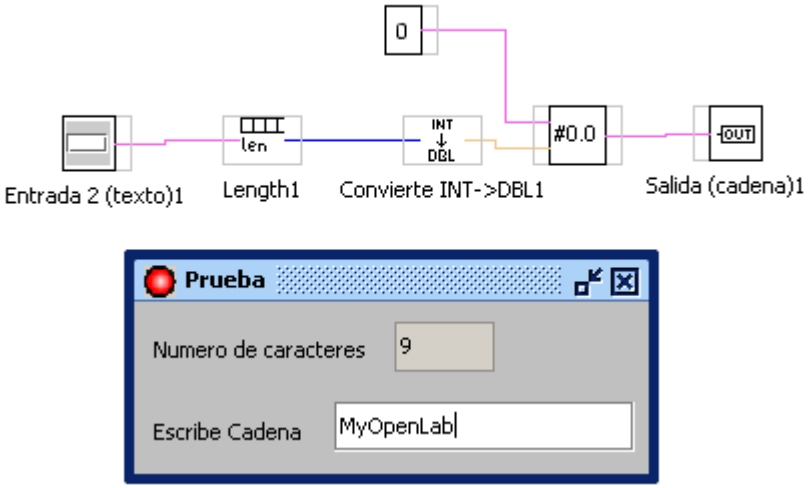


Figura 24

7. Extraer elementos de una cadena “Substring”

Esta función es muy útil para cuando queramos extrae de una cadena un conjunto de caracteres y convertirlo en otra cadena distinta.

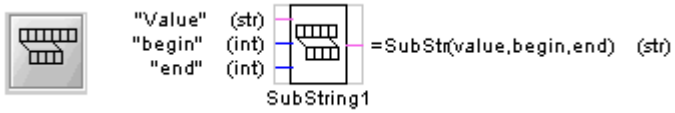


Figura 25

El bloque tiene una entrada “value” que es por donde se pone la cadena de entrada y después dos entradas “begin” y “end” que son los valores tipo integer que delimitan la cadena. La salida lógicamente es una cadena.

En la figura 26 vemos un ejemplo de esta función en el que se pone una cadena y se delimita el comienzo y el final de la cadena a cortar.

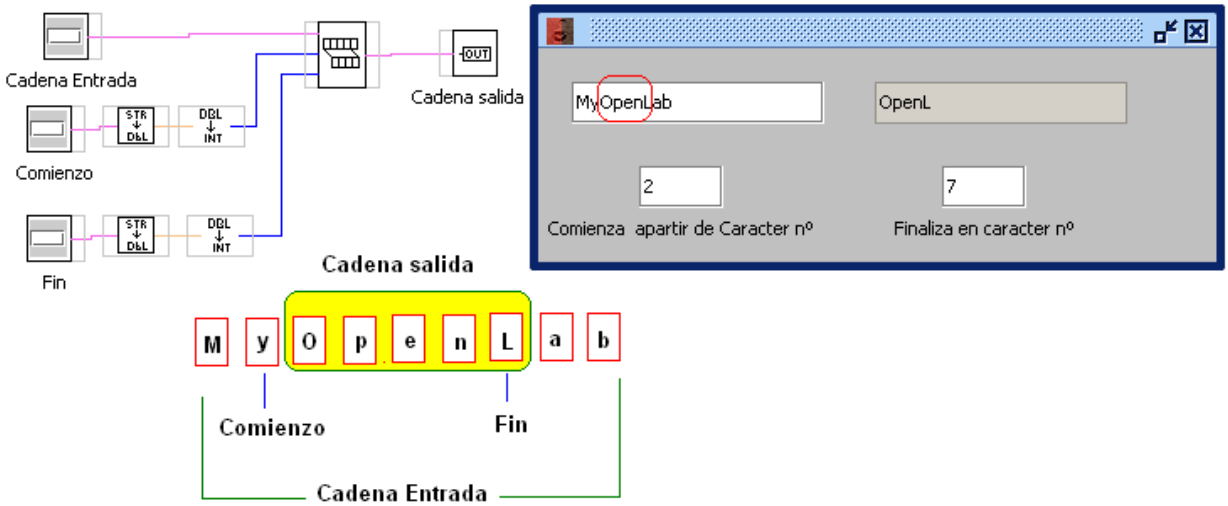


Figura 26

8. Sumar los elementos de dos cadenas

Esta función nos permite sumar los elementos de dos cadenas. Las dos variables de entrada son dos cadenas (tipo string) y la salida también lo es.

En las figura 27 y 28 vemos un ejemplo en el que se suman tres cadenas poniendo en cascada dos bloques. Del mismo modo se han colocado contadores de caracteres a la salida de las cadenas para comprobar que efectivamente se suman los caracteres.

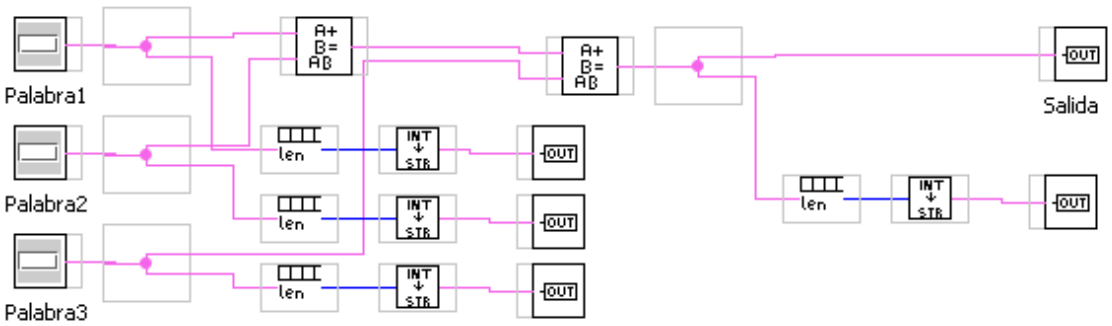


Figura 27

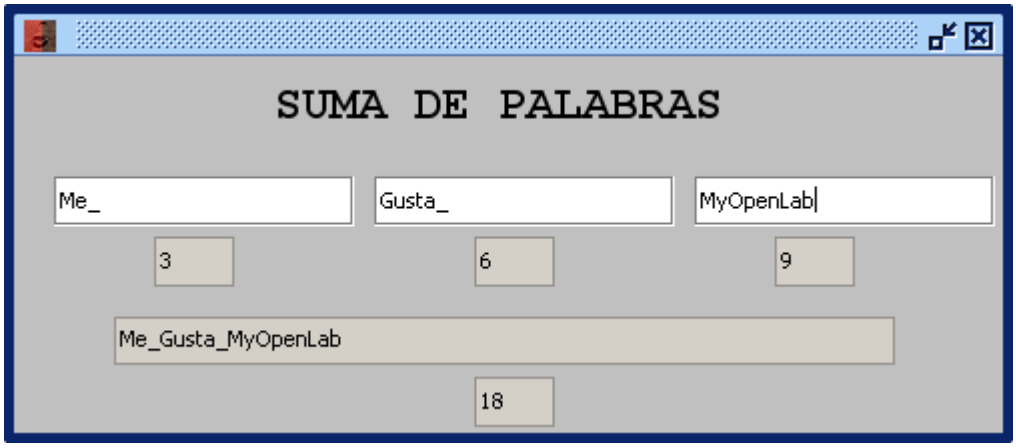


Figura 28

9. Registro de un dato en memoria.

La función Memoria  permite almacenar un dato y sacarlo solo cuando se de la orden por su entrada de gobierno In1(señal booleana).

Figura 29

En la figura 30 vemos una aplicación de un detector de máximo valor. Se trata de que se almacene el valor máximo alcanzado por una variable. El botón de reset es para situar el valor máximo en el actual.

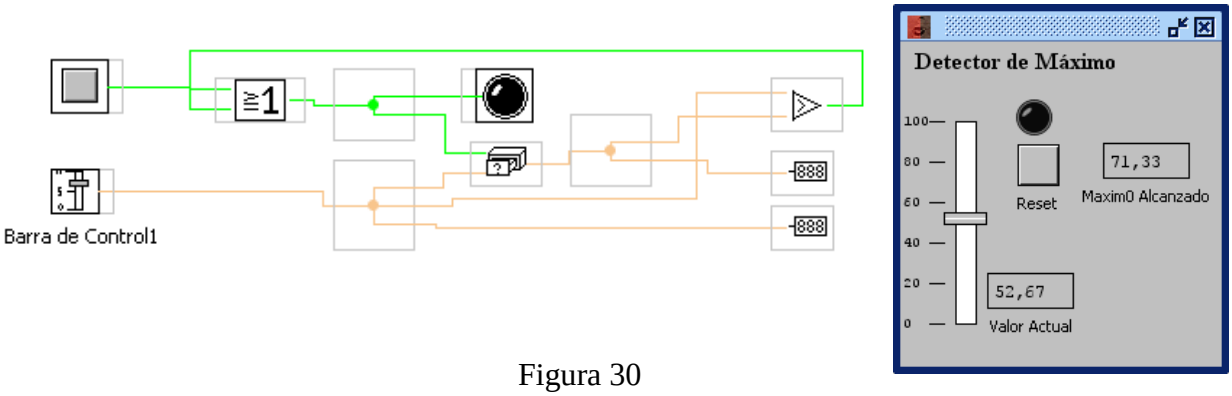


Figura 30