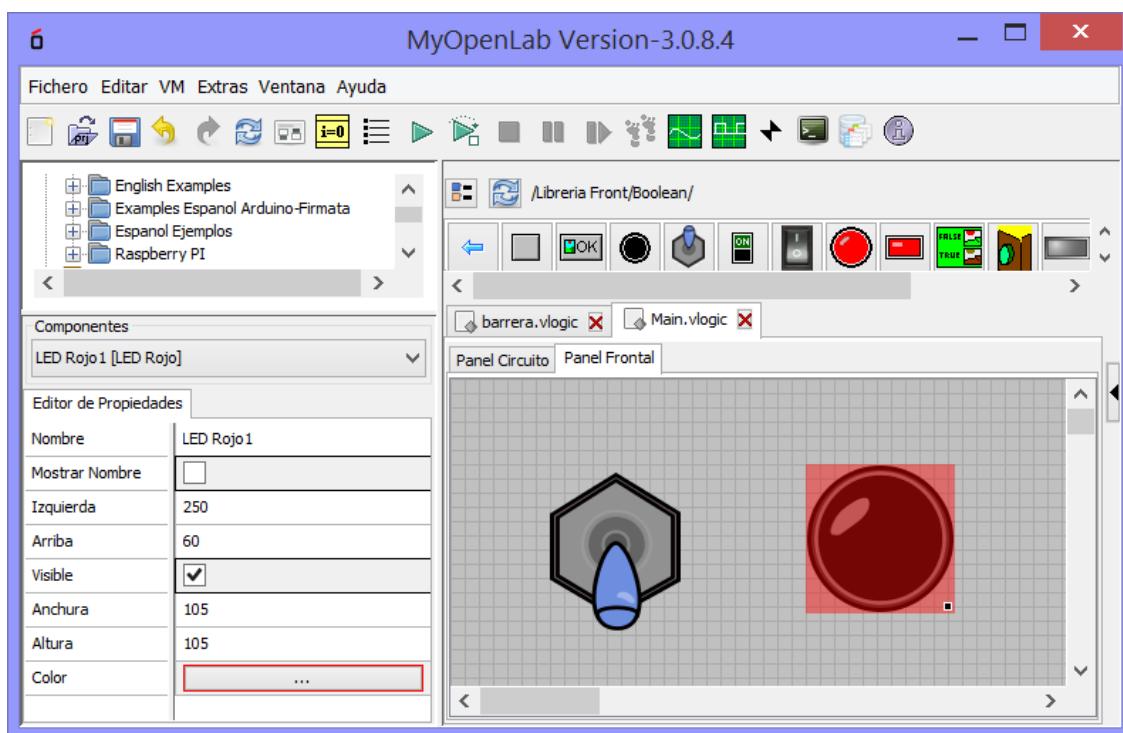




Version 3.0.8.x

User's Guide



Prof. José Manuel Ruiz Gutiérrez

Index

1. Installation of MyOpenLab
 - 1.1. System Requirements
 - 1.2. How to install MyOpenLab?
 - 1.3. How MyOpenLab works
2. Basic description of the environment
 - 2.1. Drop-down menus.
 - 2.2. Environment configuration
3. Circuit Panel
4. Element Libraries: Circuit Panel
5. Front Panel
6. Element Libraries: Visualization Panel
7. Help
8. Organization of application files:
9. Creation of VM sub-models (Sub-VM) to include within a generic VM application.
10. Library repository
11. Creation of an element within a User library
12. Creation of a distribution
13. Protection and personalization of our work
14. Plotting and tracking the execution of a VM simulation.
15. Variables Viewer.
16. Execution in Debug mode (Debugging).
17. Error and warning window
18. Console output window
19. Identification of the author of an application

1. Installation of MyOpenLab

1.1 System requirements

Before installing MyOpenLab, check the system requirements first

- Windows XP or higher 32/64 bit, 64 bit Linux
- Java 7 or higher
- For 3D: Java 3D (only up to Java 7)
- min. 50MB Free disk space
- min Screen resolution. 1024x768 pixels

1.2. How to install MyOpenLab?

- Download and install Java (R) here from <http://java.com/>
- Download the latest version from MyOpenLab <https://myopenlab.de>
- Unzip the MyOpenLab Distribution zip file and open the unzipped folder
- Start the My OpenLab application with start.bat

If you want to develop your own element (components) you must start the application with start_with_JDK.bat (Windows only). See that the Path that is noted in this file is the same one in which you have Java JDK

Notes for Linux users

1. Open your preferred shell
- 2 Go to your distribution directory
- 3 exec: chmod + x start_linux
4. and start with ./start_linux

If you want to use RS232 with MyOpenLab under Linux, you need to install the RXTX Library. For Ubuntu: sudo apt-get install librxtx -java

Notes for Mac users

RS232 function does not work with Mac.

Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and/or other countries.

The content of the MyOpenLab distribution fic hero is as follows:

Name	Änderungsdatum	Typ	Größe
Arduino+ANX7-OS	28.04.2016 17:46	Dateiordner	
Elements	28.04.2016 17:46	Dateiordner	
images	28.04.2016 17:46	Dateiordner	
lib	28.04.2016 17:46	Dateiordner	
licenses	28.04.2016 17:46	Dateiordner	
MyOpenLab_and_Arduino V	28.04.2016 17:46	Dateiordner	
Robot3D	28.04.2016 17:46	Dateiordner	
rxtx	28.04.2016 17:46	Dateiordner	
DistributionStarter.jar	28.04.2016 17:46	Executable Jar File	4 KB
myopenlab.jar	05.05.2016 18:22	Executable Jar File	1.073 KB
objects.robot3DFile	28.04.2016 17:46	ROBOT3DFILE-Datei	1 KB
readme.txt	28.04.2016 17:46	Textdokument	1 KB
rxtxParallel.dll	28.04.2016 17:46	Anwendungserwe...	83 KB
rxtxSerial.dll	28.04.2016 17:46	Anwendungserwe...	127 KB
start with JDK.bat	28.04.2016 17:46	Windows-Batchda...	1 KB
start.bat	05.05.2016 05:58	Windows-Batchda...	1 KB

When the **start.bat** file is executed for the first time, a series of MyOpenLab user files are created in the folder **C:\Users\user name\VisualLogic** with the following files.

- "Config.conf" file is for all settings under the "Option" button or the "Option" menu (Do not attempt to modify this file!).
- File "Licences1.Accepted" will be created and accepted when MyOpenLab Licenses is finished.
- "Myopenlab.path" contains your MyOpenLab path element (text file)
- "Projects.file" contains all open projects in your MyOpenLab
- "UserdefinedElements.path" is deprecated (no longer needed since version: 3.0.6.0).

Name	Änderungsdatum	Typ	Größe
Config.conf	05.05.2016 18:24	CONF-Datei	2 KB
Licences1.Accepted	17.04.2016 16:31	ACCEPTED-Datei	0 KB
myopenlab.path	05.05.2016 18:22	PATH-Datei	1 KB
projects.file	05.05.2016 18:24	FILE-Datei	1 KB
userdefinedElements.path	17.04.2016 16:31	PATH-Datei	1 KB

If you delete this folder with its files, running the star.bat file again will reinstall it. This is one way to reinstall MyOpenLab.

We take for granted that we have Java installed in our computer. Remember that we must have a version 7 or higher of the JDK tool or simply the Java SE Runtime if your PC works with 64 bits remember that you must install the Java version for 64 bits.

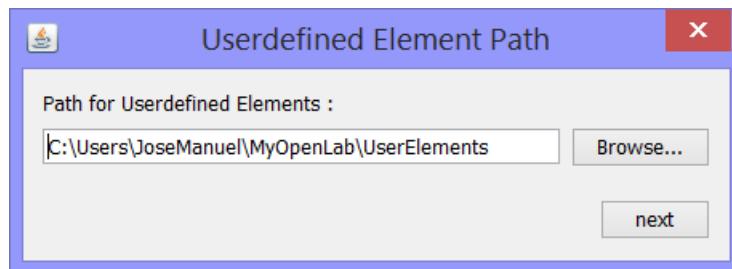
1. The first time we execute the file **star.bat** from the folder where we have unzipped the MyOpenLab distribution file, the following window will appear:



2. We read the terms of use and if we agree we select "I accept all licenses" and the following window appears in which we select the language (German, English or Spanish) and click **OK**.

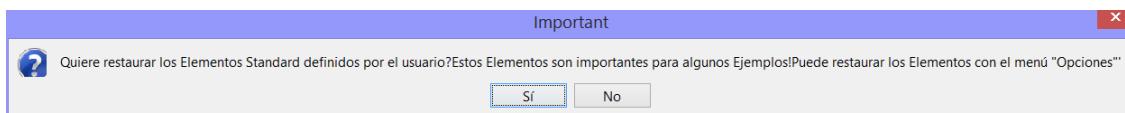


3. Next, a window will appear in which we are invited to write the path where the User Library Elements will be saved, by default the one shown in the window is selected, the location is fine, we can leave it there, click "next" and move on.



4. We will be asked if we want to save in this user libraries folder that will be created in the path of the previous window the library elements that use

Some examples The best thing to do is to say "YES" and continue. In any case, this can be defined later in the "Options" menu.



- We have just installed MyOpenlab. Do not forget that two folders have been created in the windows user folder:

One folder:

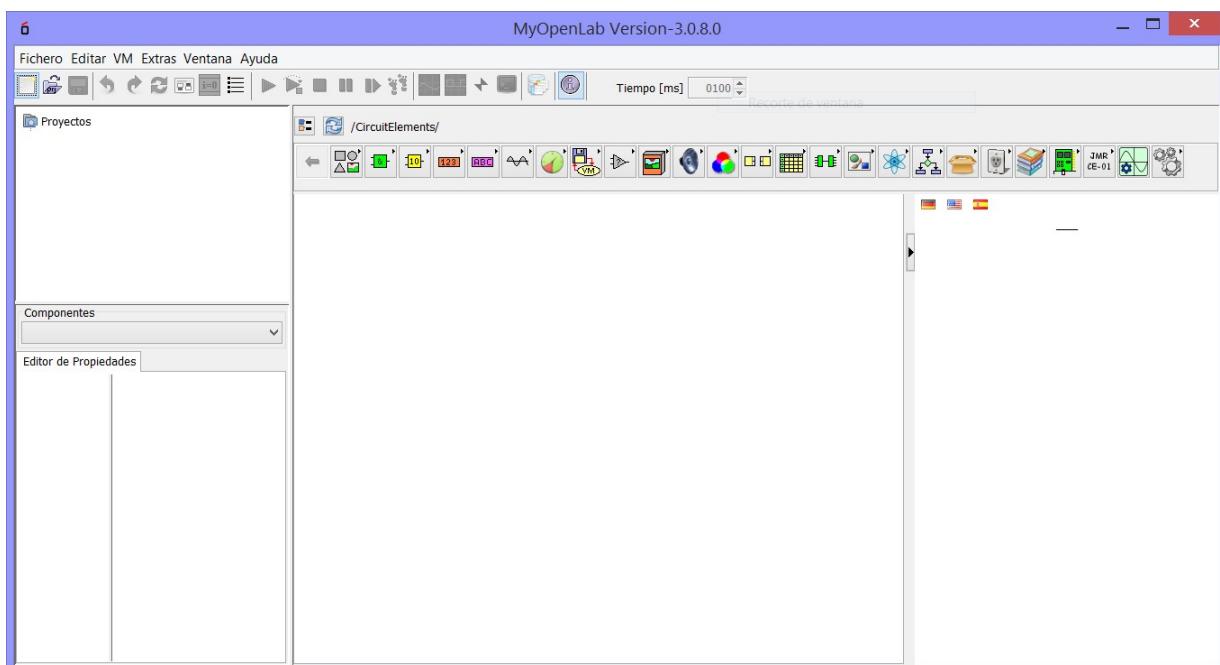
C: \Users\user\MyOpenLab -> which contains the user library elements.

C: \Users\user\MyOpenLab\UserElements

Another folder.

C: \Users\user\VisualLogic -> Containing configuration files necessary for the program.

Once the installation is finished, MyOpenlab will open automatically and the following screen will appear, in addition to the "Item Window" screen that we can close.



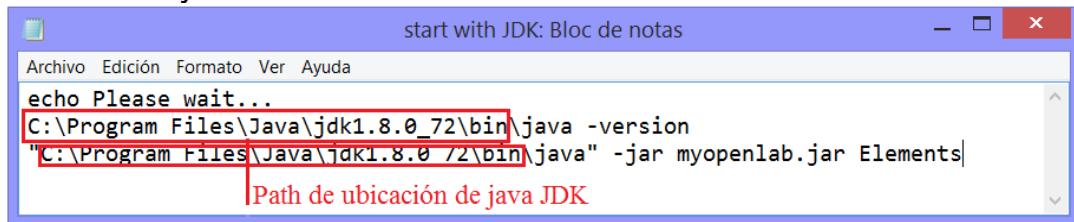
1.3 How MyOpenLab works.

MyOpenlab can be executed in two ways:

- **In Normal mode :** executed *start.bat*
- **In Compile Mode :** executing *start with JDK.bat*

With the second form we can compile those elements that we want to add or modify.

It is important that for the second way we have installed the java JDK and placed its Patch correctly in the bat file.



```
start with JDK: Bloc de notas
Archivo Edición Formato Ver Ayuda
echo Please wait...
C:\Program Files\Java\jdk1.8.0_72\bin\java -version
"C:\Program Files\Java\jdk1.8.0_72\bin\java" -jar myopenlab.jar Elements|
Path de ubicación de java JDK
```

2. Basic description of the environment

MyOpenLab is an environment oriented to the simulation of circuits and systems with a wide range of applications.

Its most important characteristics are:

- Ease of use
- Extensive library of functions for both analog and digital signal handling.
- Processing of data types and data operations.
- Realization of applications using function blocks.
- Facility to create visualization screens that collect the status of variables and events of the simulations.
- Possibility of extending its component library.
- Possibility of creating sub-models that can be encapsulated in other sub-models.

Within the Virtual Instrumentation tools (VI Virtual Instruments) this tool is an example of simplicity and power at the same time.

The applications realized with MyOpenLab are called **VM** (Visual Modeling).

Each application shall consist of two distinct parts:

Circuit (Panel Circuit)

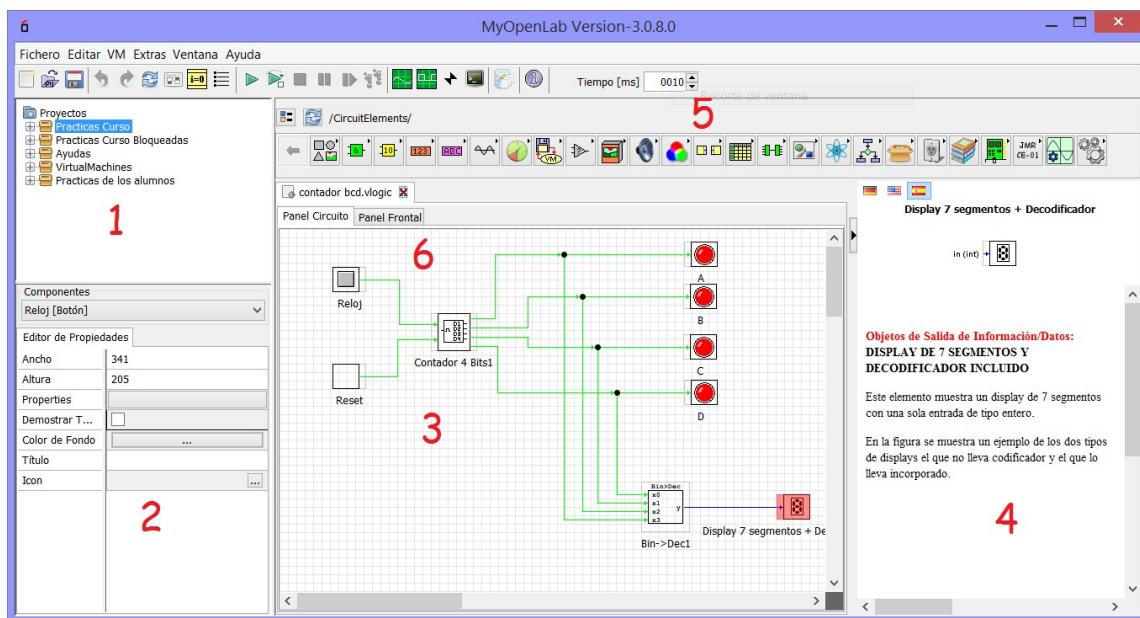
Display Panel (Front Panel)

The Circuit will be the set of functions and/or operators that conveniently linked together respond to a functionality related to an electronic circuit or a system of any kind. This set of functions is edited in the corresponding circuit editing panel, which appears in the tab labeled "Circuit Panel".

The Front Panel Display part will be the one that appears in the tab labeled "Front Panel" where the graphic display objects associated with the variables of the circuit will allow the visualization of the evolution of the simulation.

In the figure we can see how the screen looks like once the application is started and an example is loaded. In it we can easily distinguish the following parts:

After MyOpenLab is started, you can see the main window. See image n below.



MyOpenLab MainWindow

Section 1: Open Projects

Section 2: Element properties editor

(Just select a member and you can edit the Properties of the element here).

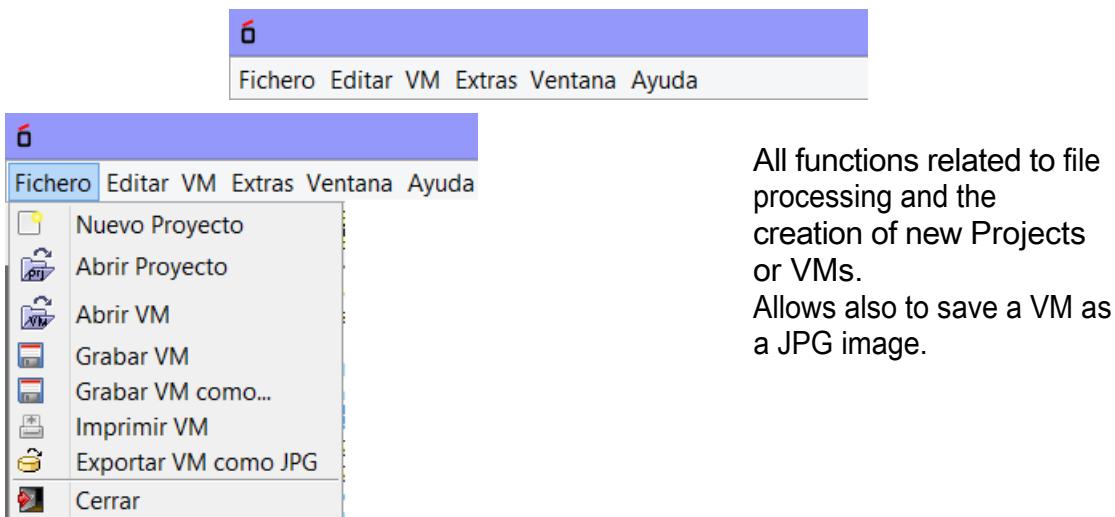
Section 3: Circuit Panel

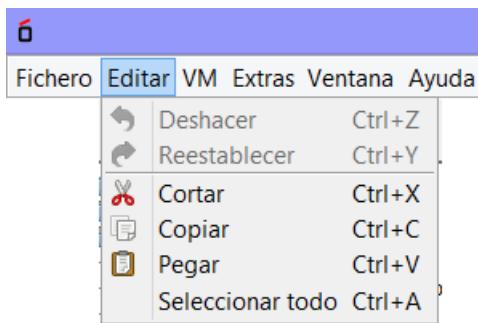
Section 4: Pin Descriptions and Element HTML help Section 5:

Panel Element Library Circuit

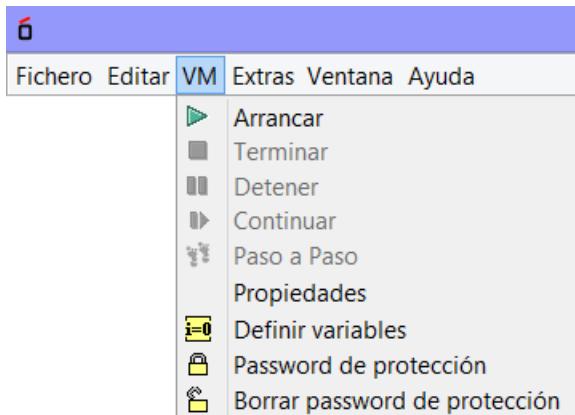
Section 6: Front panel tab

2.1. DROP DOWN MENUS

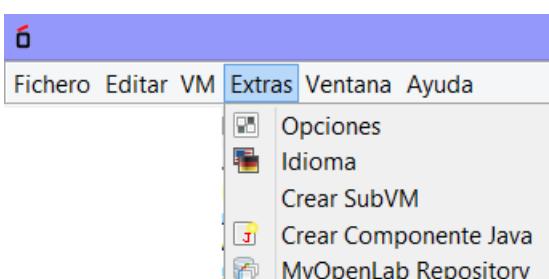




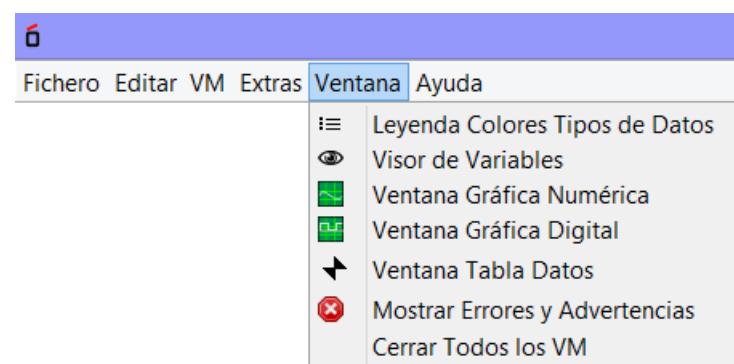
All editing functions



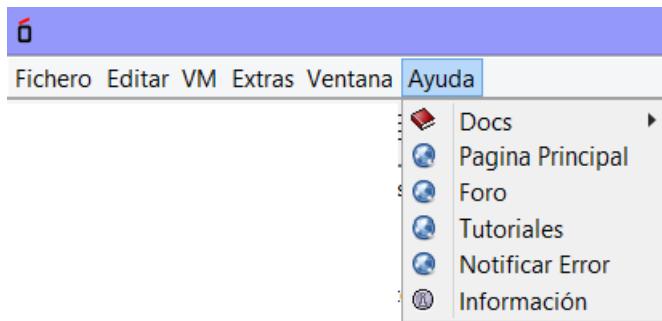
Functions related to the execution of a VM, definition of variables and processing of protection passwords.



Options configuration, language. Creation of Sub-VM, Java Components and connection to MyOpenLab Repository

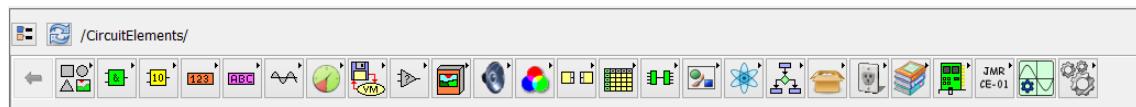


It allows to display windows with colored legend for data lines, Variable Viewer, Data Plot windows, Error Window and Close all open VMs.

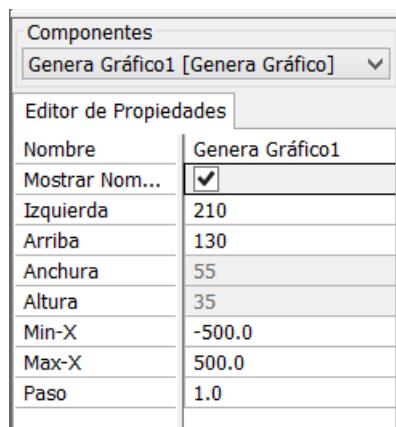


Displays help and connection to MyOpenLab website and Forum

The most important parts into which the MyopenLab main screen is divided are:

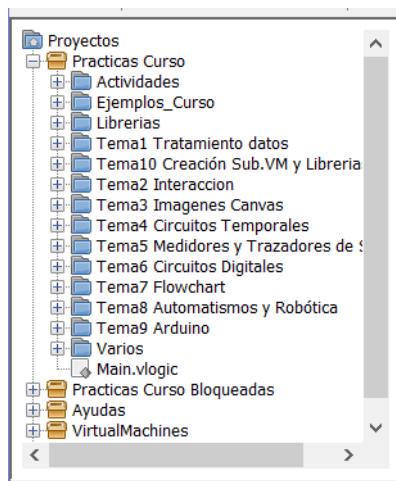


Component Area: Where the libraries of the active panel (Circuit Panel or Front Panel) are displayed. It is possible to navigate through the libraries by clicking on the icons that represent the folders where the different elements are located.



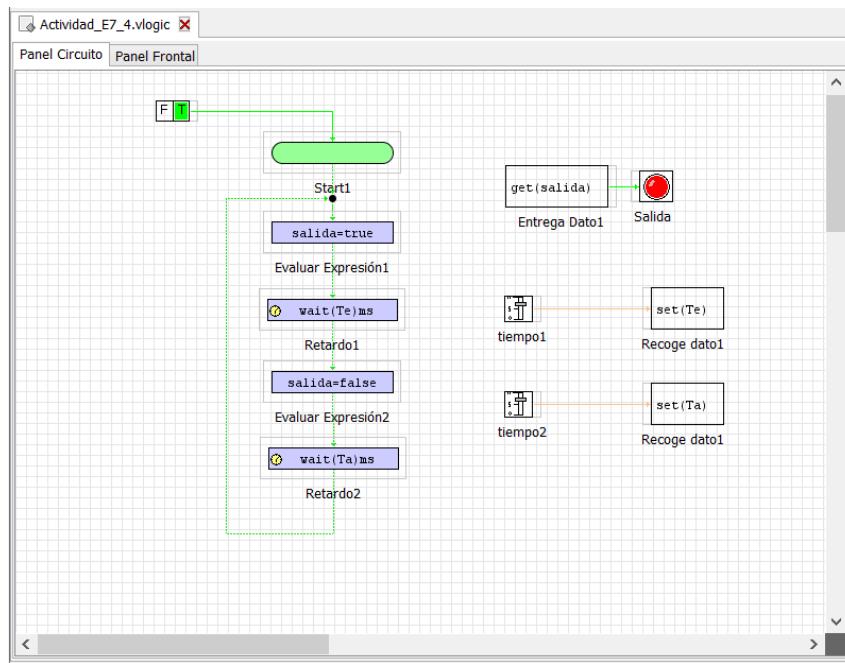
Properties Area of Components

Components: This displays a window with the properties of the object or function that we point out.

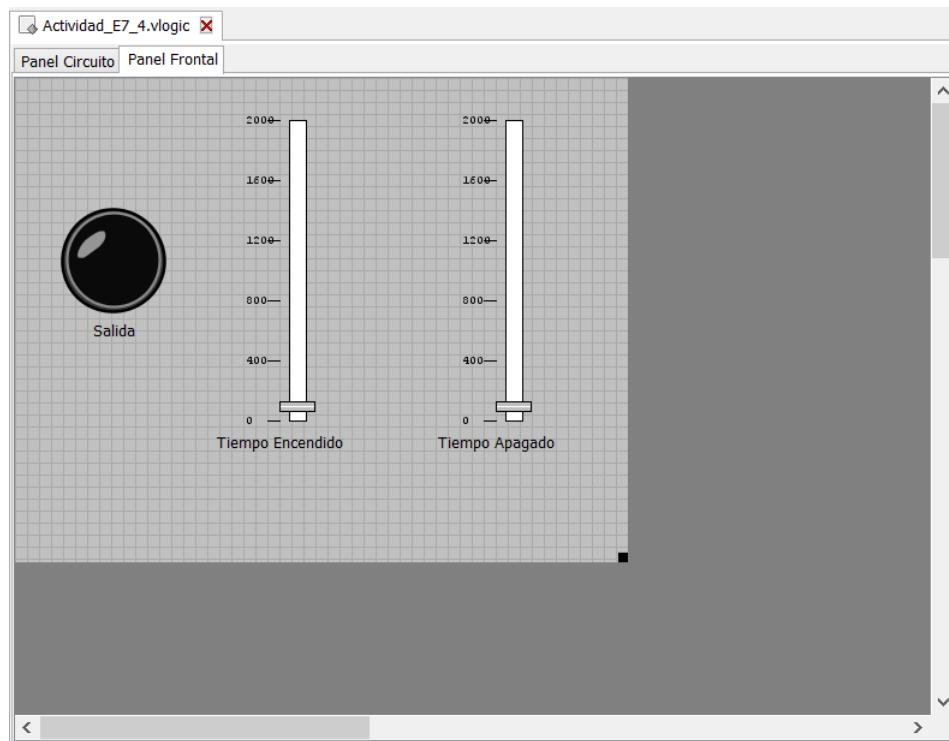


Project Area. Where the tree of folders and files of VM applications is shown.

Circuit Panel: This is the one in which we design the circuit and wire all the components connecting inputs and outputs.



Front Panel : This is the panel on which the objects that will display the results or from which values will be collected in the simulation are placed.



We see that some elements are displayed in the two panels. Exactly those that we have taken from the libraries of display objects. Such is the case of the "Zoom X" and "Zoom Y" potentiometers that appear on the circuit panel and on the front panel as the "Frequency" and "Amplitude" controls.

Button Bar: This bar contains the buttons associated with the most important functions of MyOpenLab.



We see that from left to right the buttons are used for:

-  Create new Project: New Project
-  Open a Project already created: Open Project
-  Save the VM file currently being edited: Save VM
-  Undo last action: Undo
-  Restore undone action: Restore
-  Update the data of the active VM: VM Upgrade
-  Allows you to configure Options: Options
-  Allows you to define and/or edit variables: Define Variables
-  Show window with information about the colors of the data lines:
Legend colors data types
-  Start Simulation: Start VM
-  Start simulation in debug mode: Start VM (debug mode)
-  End simulation: End VM
-  Stop simulation: Stop VM
-  Continue simulation: Continue VM
-  Performing the simulation step by step: VM step/step
-  Display Numerical graphic window
-  Sample Digital Graphic Window
-  Show Data Table window
-  Data Output Console Display
-  Repository
-  Displays window with program information: Information

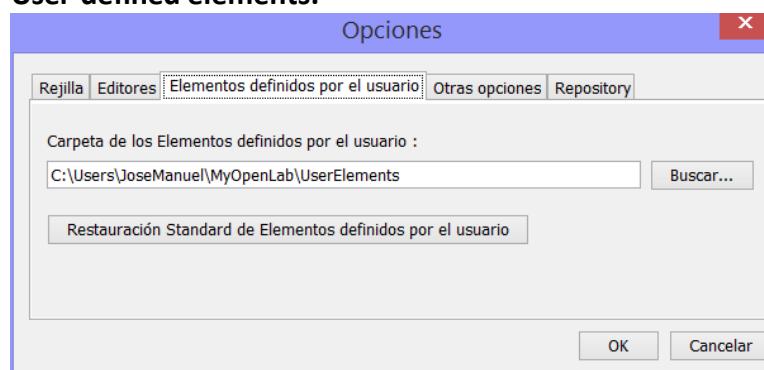
Tiempo [ms] 0010  Adjusts simulation time

2.2. Environment configuration.

MyOpenLab allows you to configure some options. To do so, go to the main menu "Extras -> Options" and select "Options", the following screen will appear.

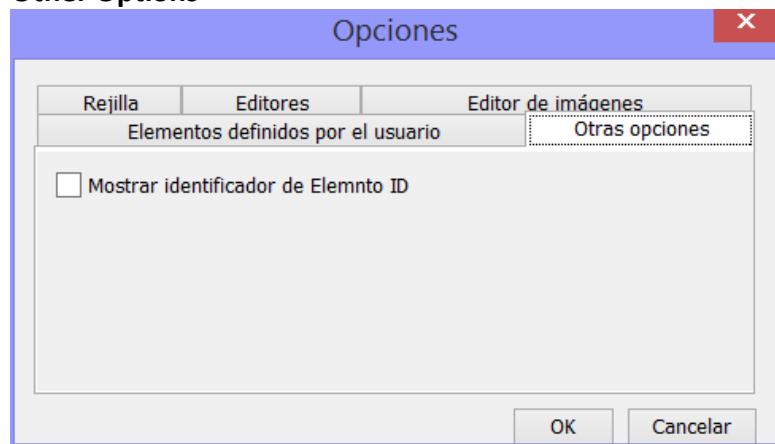


User-defined elements.

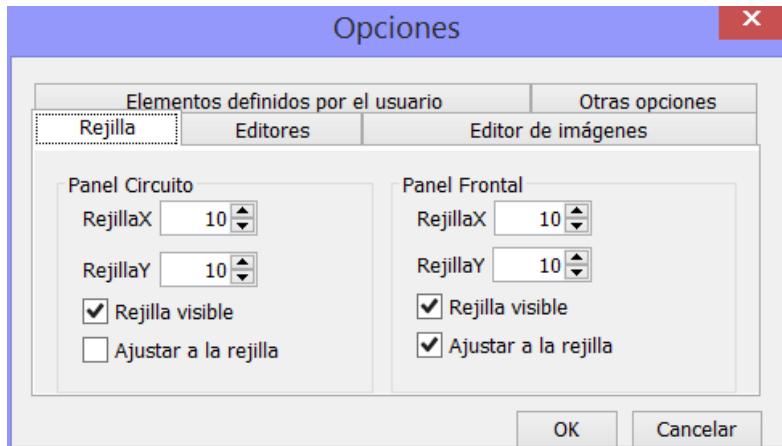


Allows to select the folder in which the library elements created by the user will be placed and also to restore the elements in the previously selected library.

Other Options



Allows you to select whether or not to display the Element identifier.

Grid

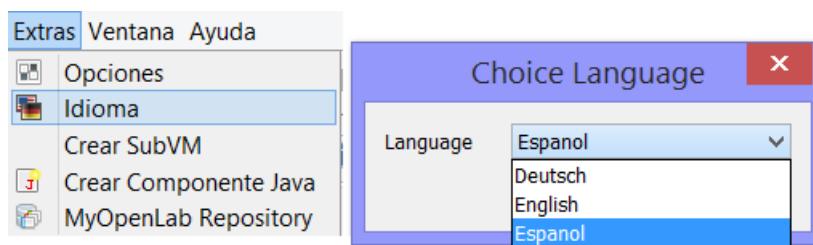
Allows to make a grid visible in the working areas of the Circuit and Front panels. It also allows to adjust the positioning of the objects on the grid and to modify the grid spacing.

Editors

Select the location of the editor to be used for editing the help files in HTLM format of the library components
Select the image editor for the library icons

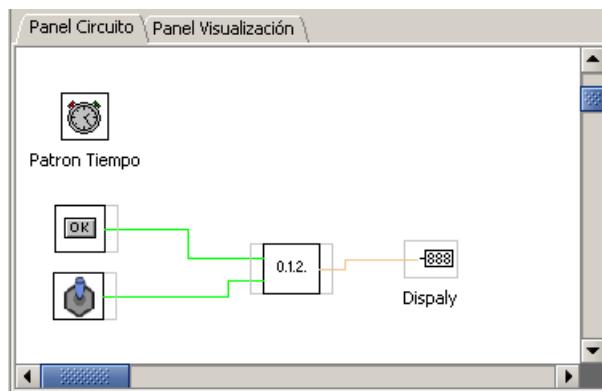
Language selection.

In the "Extras -> Language" menu

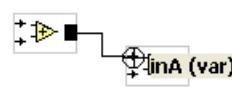
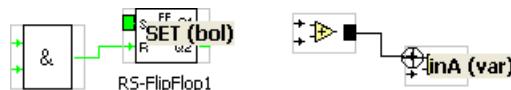


3. Circuit Panel

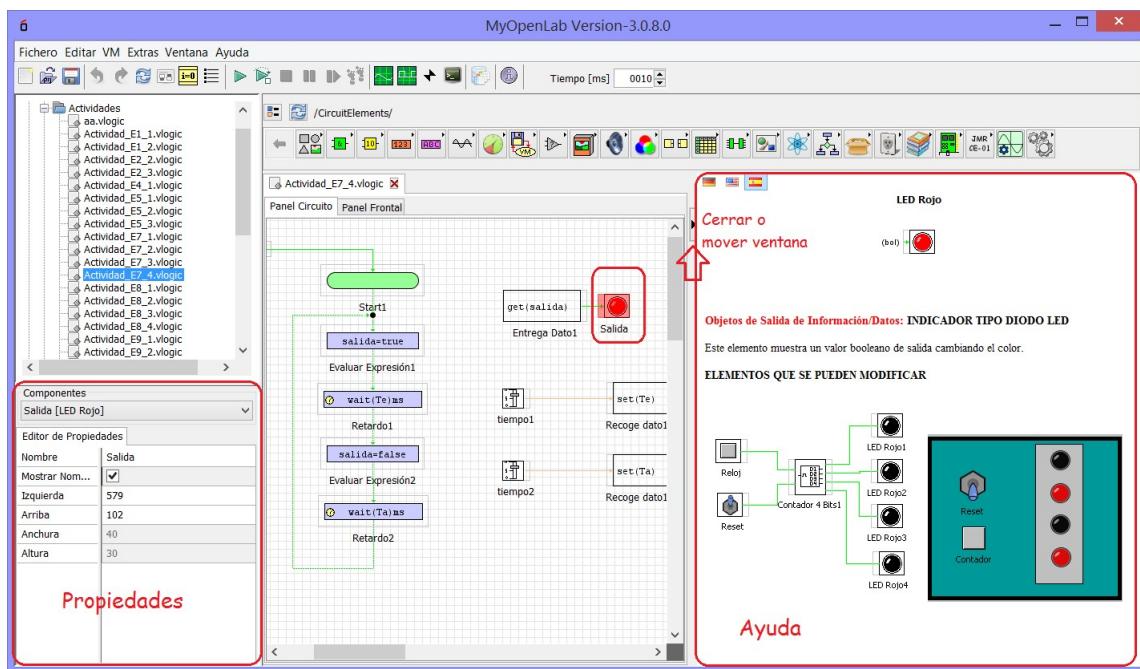
The Circuit Panel is the work area where each of the elements of the different libraries that we are going to use in our project are placed.



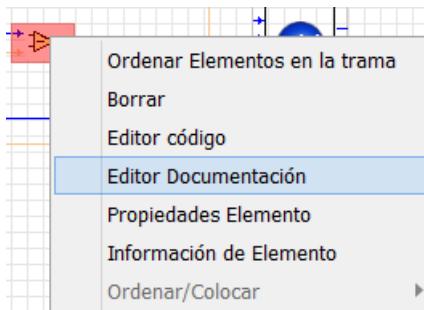
The components, once placed in this area, can be linked together using the mouse. Simply move the mouse pointer over an input or output terminal of a component to display a label identifying the name of the terminal and in parentheses the type of data it supports (if it is an input) or delivers (if it is an output).



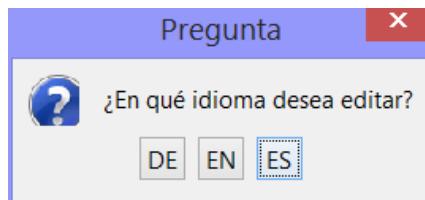
When a component is marked or selected, we see on the right side of the window the help area that shows information about the component, this window can be made larger or smaller and even closed.

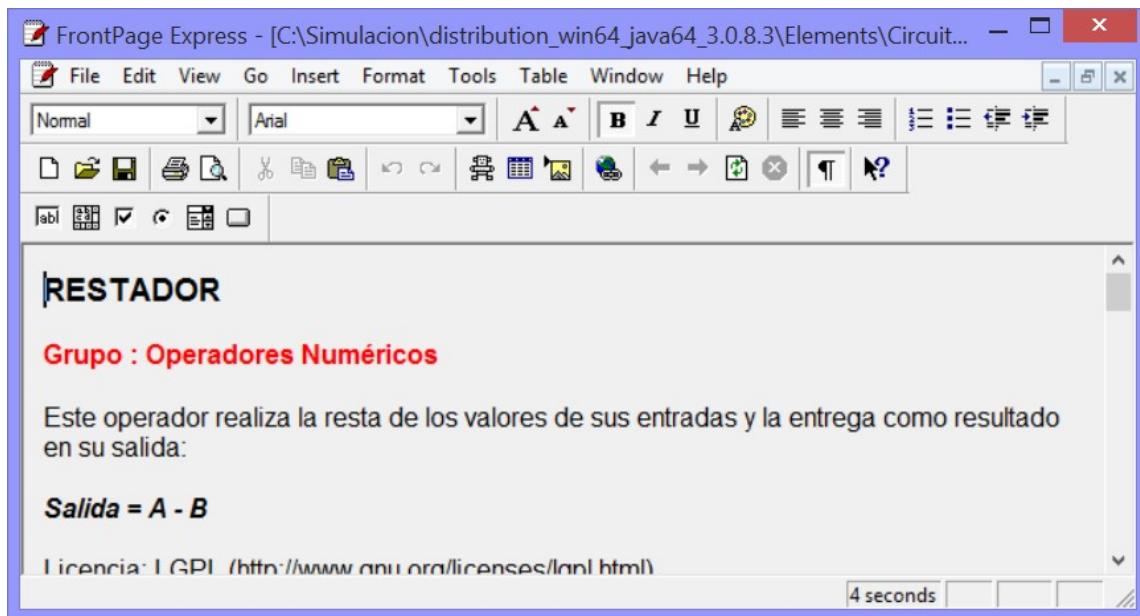


The help file of a block or component can be edited by simply clicking on the component and right-clicking on it with the mouse, a menu will appear in which we select the option "Documentation Editor" which, when selected, displays the HT LM Editor that we have selected in the configuration options (for example FrontPage).



It will ask for the language in which we want to edit the help and then open FrontPage.

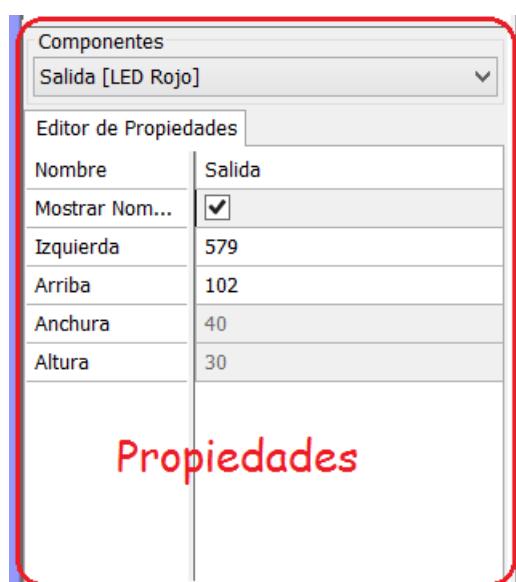
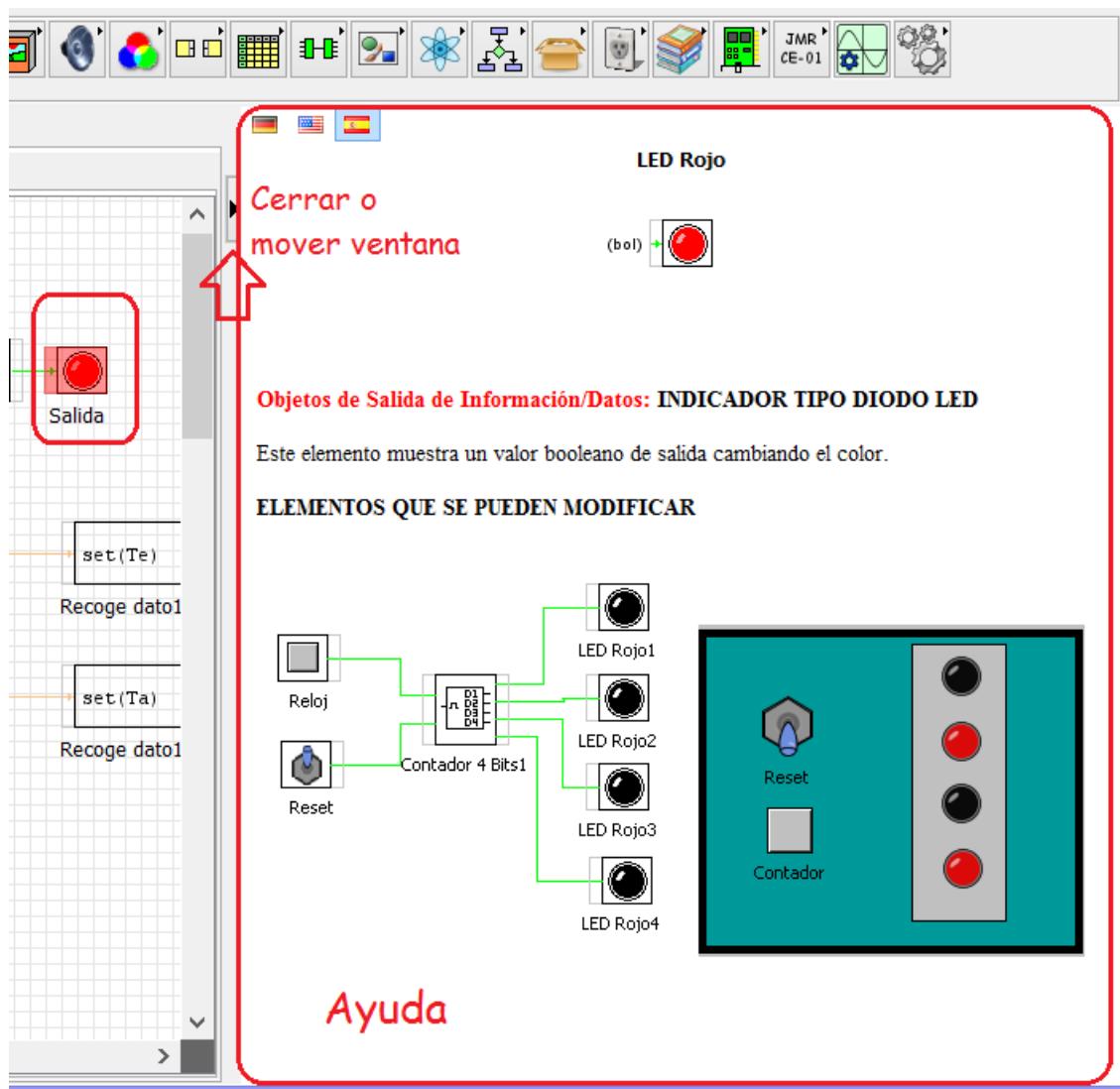




Selection of a component

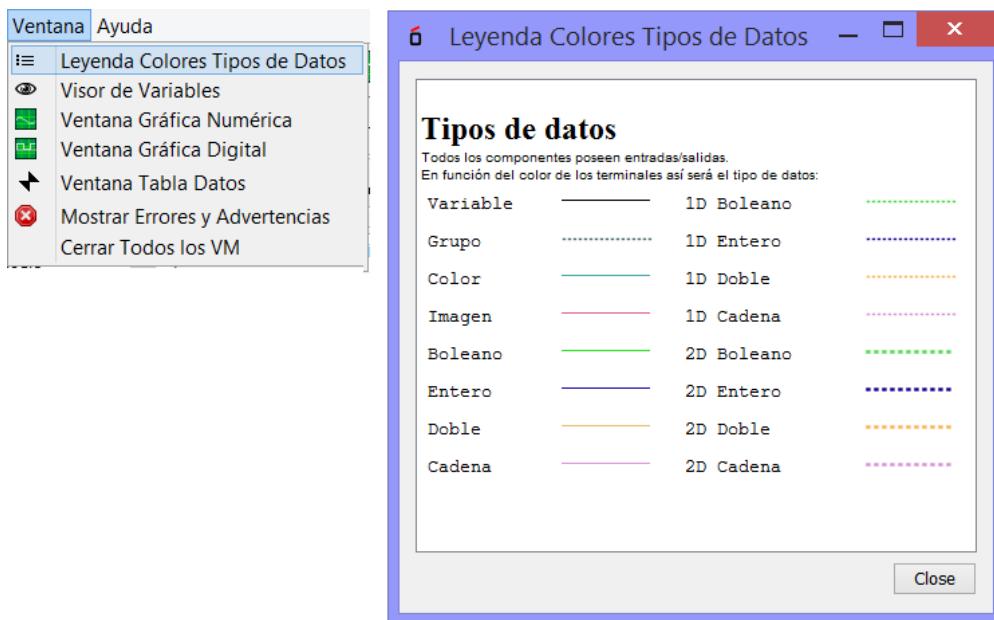
The elements are selected from the library area located at the top of the screen. The libraries are accessed by clicking on the folders and once the element we need is located, we click on it with the left button and select it, then we go to the work area of the panel and click again with the left button and the element is placed. We will be able to move it if we wish simply by holding down left-click on it and move the mouse.

In the figure we see the Led component together with its help window and its properties window.

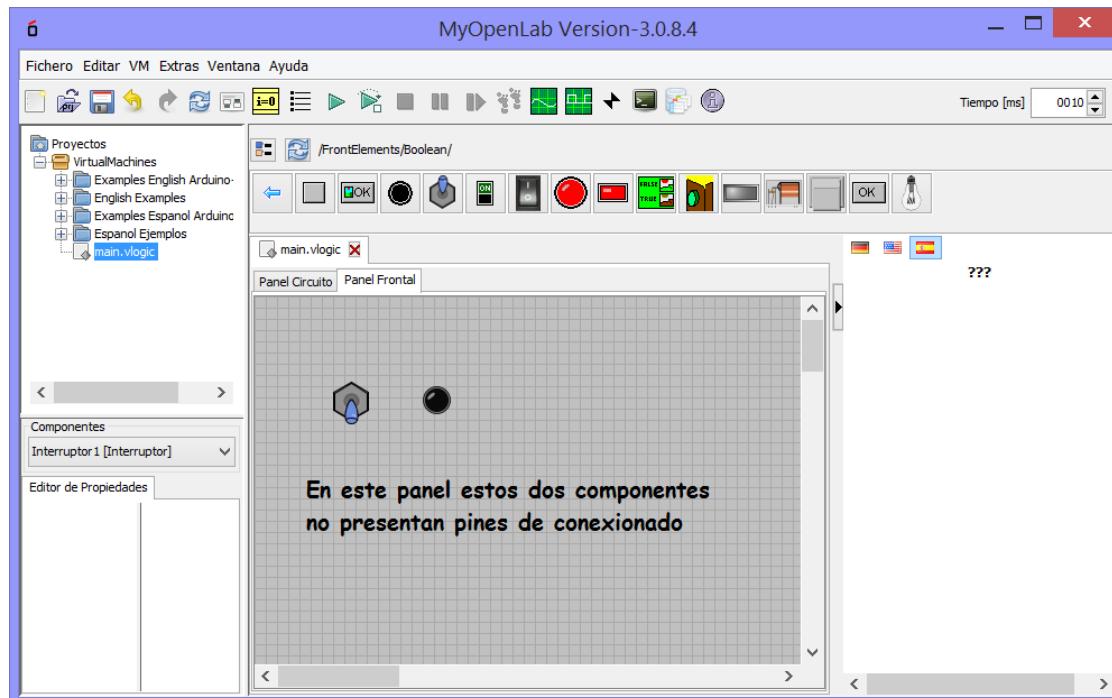


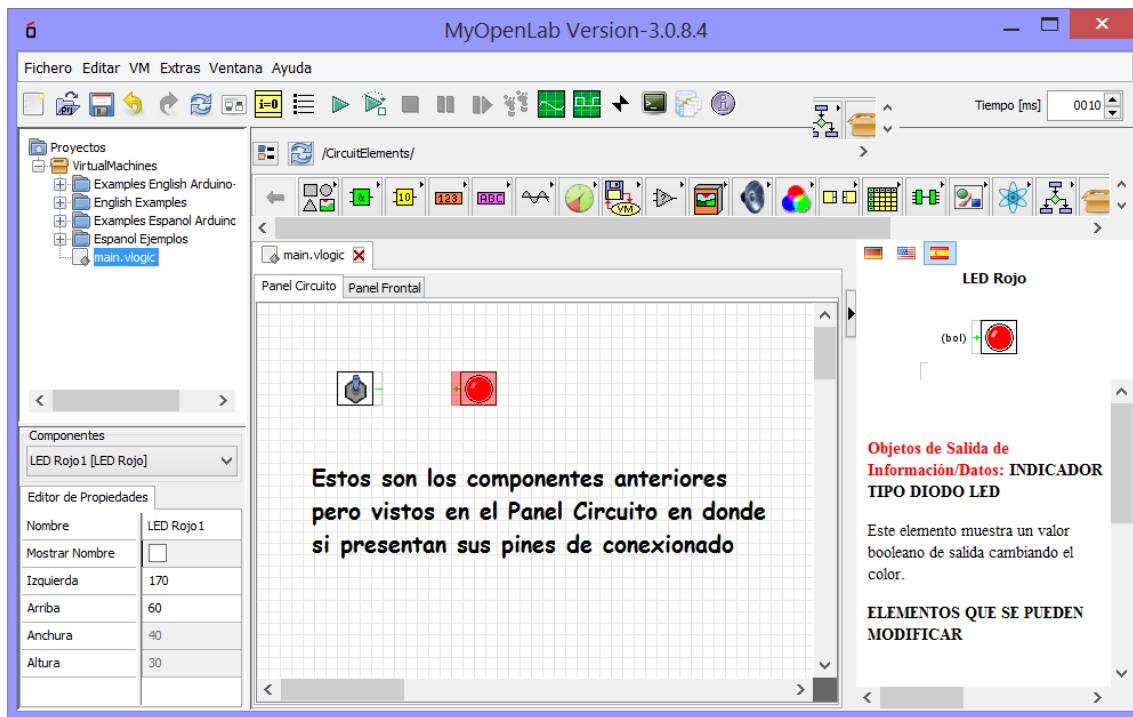
Note that this component only accepts *Boolean* type data in its inputs.

It is very important to pay attention to the colors of the input and output lines of the components, since depending on their color and type (continuous or dotted) you can determine their nature.



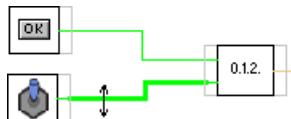
Some components when placed on the Display panel will automatically appear on this panel. For example the LED and the Switch in the following figure. In the Display Panel it is not possible to connect components, therefore the Input/Output components of this panel must appear in the Circuit Panel in order to be connected to the corresponding input or output.





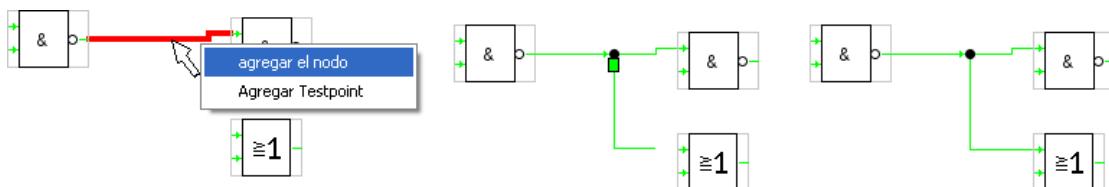
Deleting a connection

If you want to delete a connection, click on it with the mouse and then press **DELETE** or in the menu *Edit ->Cut* to delete the connection.



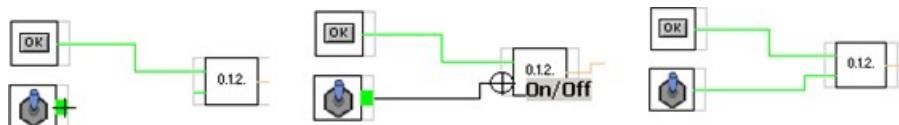
Placement of a branch or splice node

If we want to place a node to be able to derive a connection to two components, it will be enough to place the mouse over the connection line and press the right button and a menu will appear from which we will select add node.



Component connection

To make the connection between components, simply mark the output with a left-click and then mark the corresponding input of the selected component.



and keeping the left button pressed look for the entry and solar

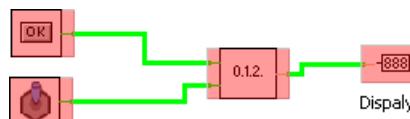
We always have to connect an output to an input or the output of a node to an input.

The On/Off label is the indication of the input terminal.

To delete components

To delete one or more components or an entire area, simply select the area or components with the mouse and then, once the area is shown in highlighted video, press the DELETE key on the keyboard or select the "Cut" option from the Edit menu.

In editing, you can mark component digits and copy and paste them, so that you can edit them more quickly.



Copy and paste Components or complete areas.

If you want to duplicate an area or component, simply select the area or component and select the "Copy" option from the Edit menu and then the "Paste" option. Do not forget that when pasting, the copied area or component is pasted over the original one.

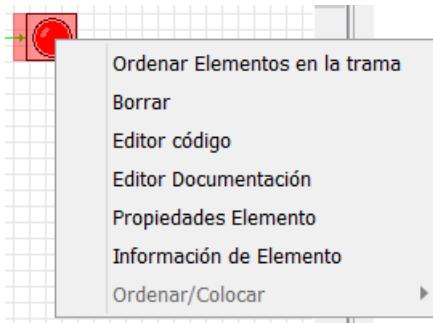
The copy operation can be done with the CTRL+C keys and the paste operation with CTRL+V.

Placement of a SubVM Element in the work area of the Circuit Panel.

When we want to place a SubVM component in the work area we just need to drag it from the project area (selecting its name) and drag it to its place. This can be done as long as the sub-element is in the same project folder as the VM we are editing.

Context menu on the component

Right-click on a component to display this menu.

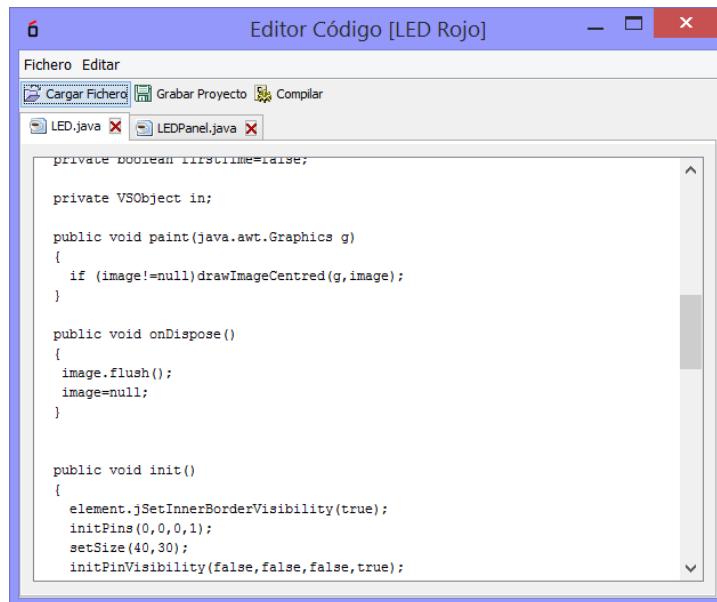


From this menu we can invoke the component's help file and also edit it. In the same way, from the "Code Editor" option we can edit the Java code of the component and modify it if we wish. This possibility makes MyOpenLab an absolutely flexible and open tool.

From this contextual menu we can, among other things:

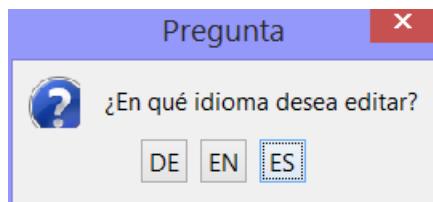
Code editor

Edit the java code of the component



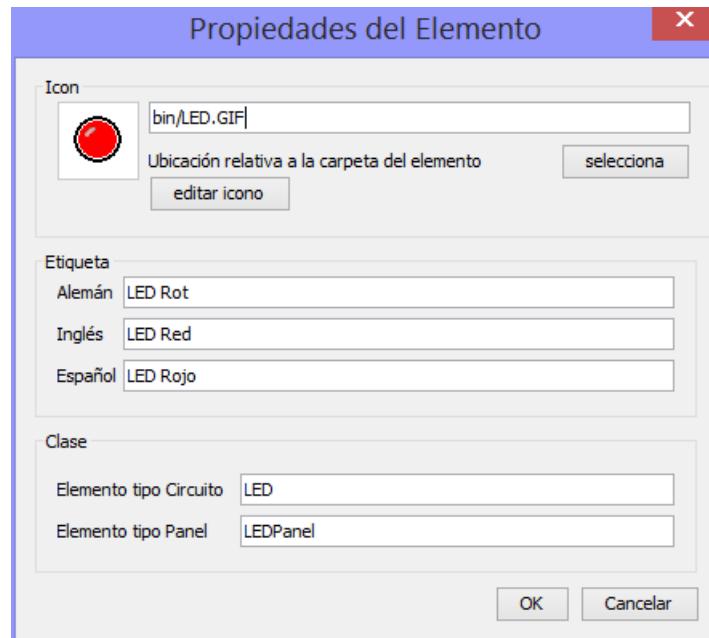
Documentation Editor

Edit help file

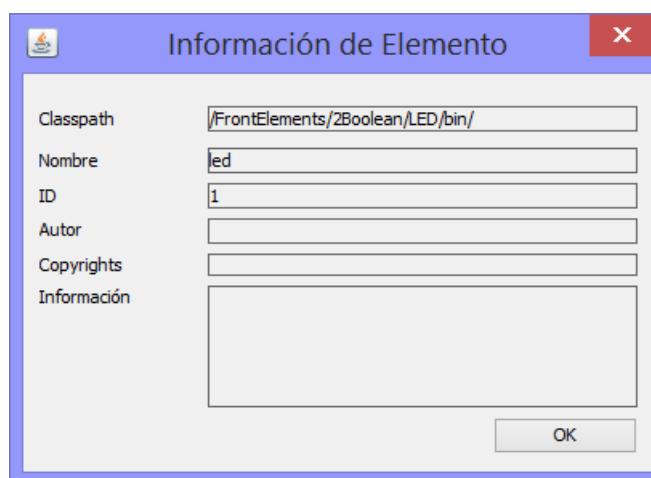


Element Properties

Show component properties

**Element Information**

Displays element information.

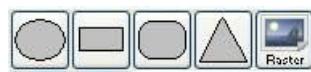


4. Element Libraries: Circuit Panel



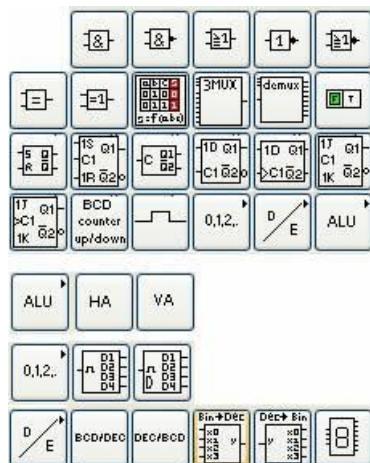
Libraries of "Logic Panel" Elements

Decoration Elements



Label

Digital Operators



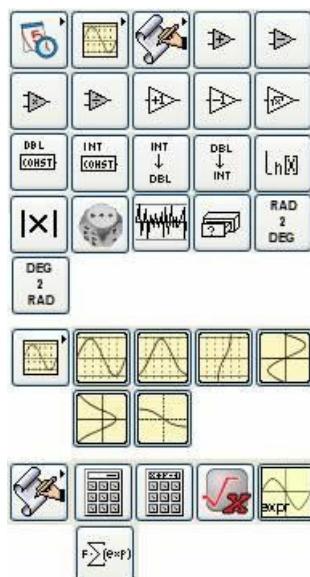
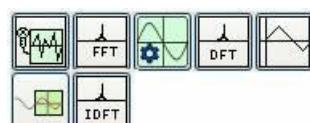
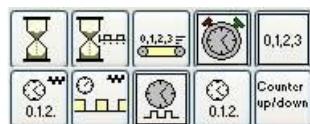
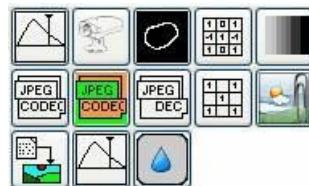
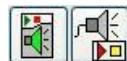
Numerical Operators**Character Processing****Analog Elements****Utilities****Input/Output Files****Comparisons**

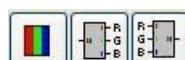
Image Processing



Sound Processing



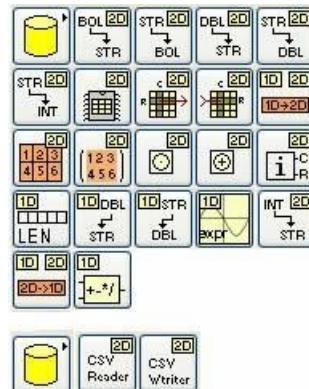
Color



I/O pins



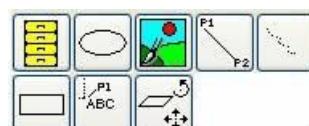
Vectors and matrices

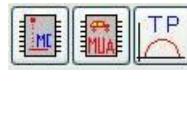
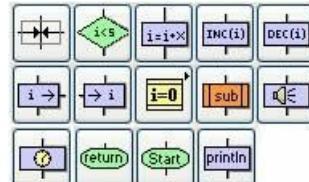


Grouping of Elements



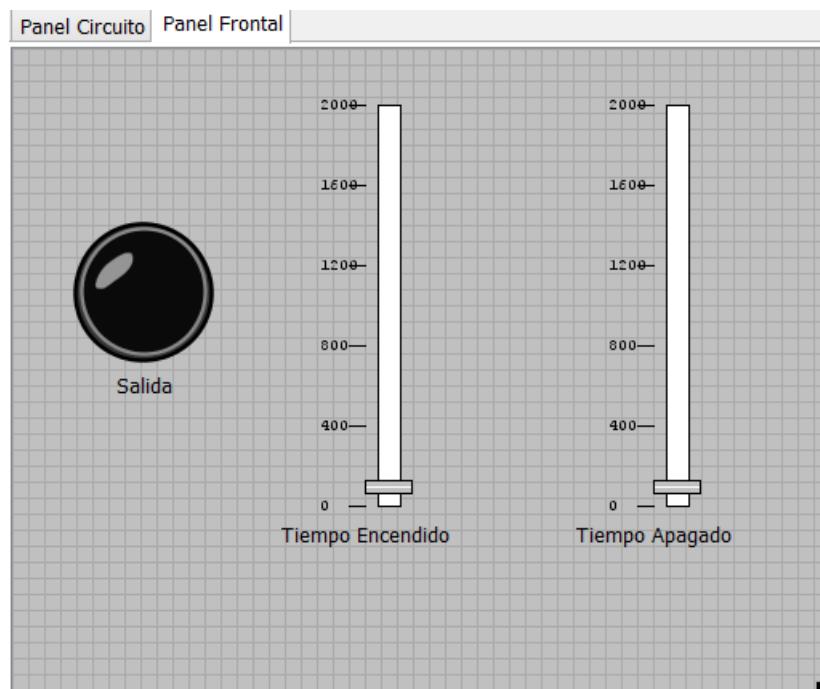
Canvas Graphic Objects



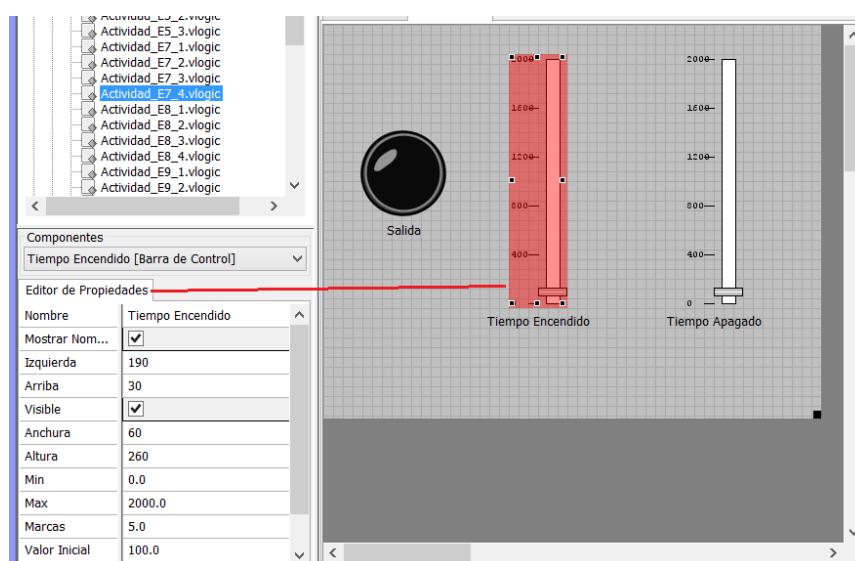
Physics Bookstore**Flow Diagram Library****Extras Library****Inter-application Connections
Library****User-defined library****Automation Library****Interfaces**

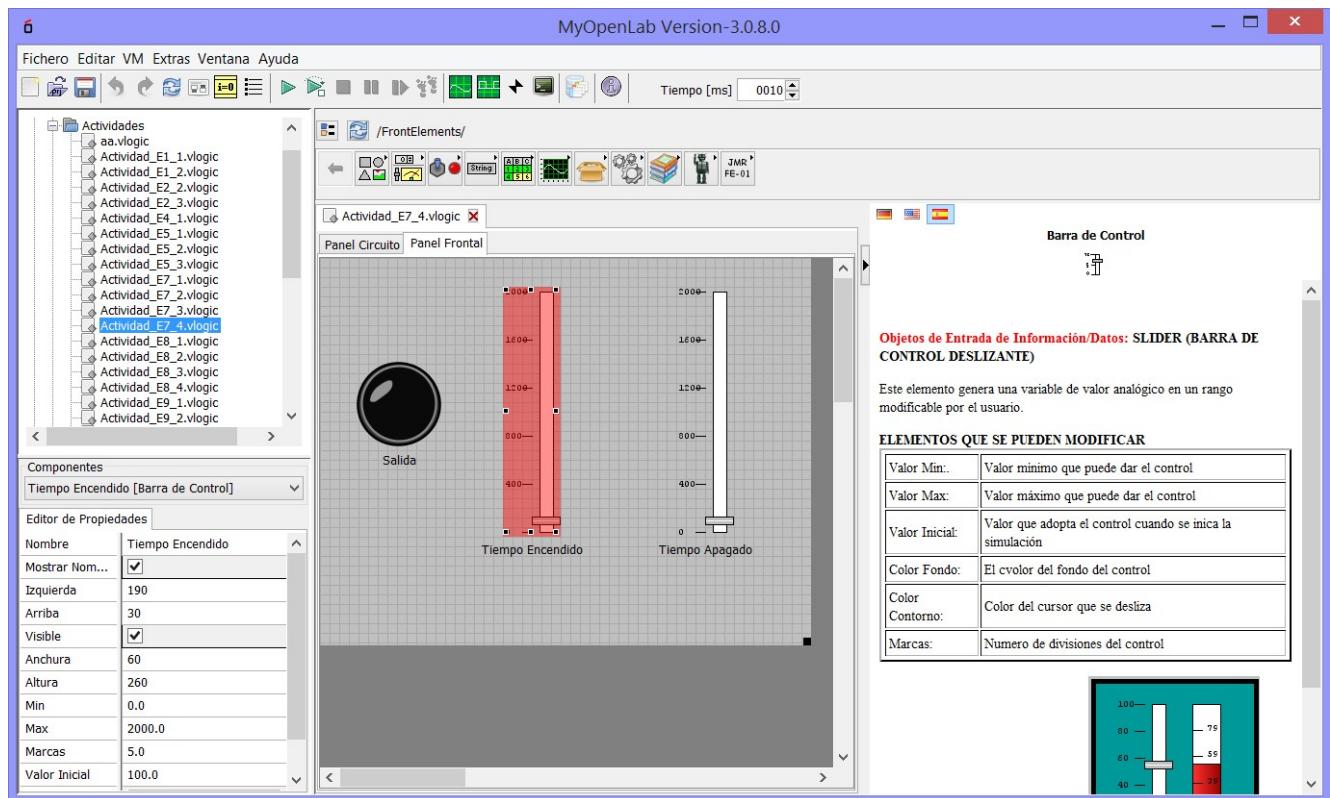
5. Front Panel

This panel is where the elements that will be used to enter the values and/or input parameters (stimuli) of the application are placed, as well as the elements that allow to see the output data (output elements). It also allows us to place on it decorative elements that have no relation with the elements that make up the application. In the figure we see the panel corresponding to the example we have used in the Circuit Panel.



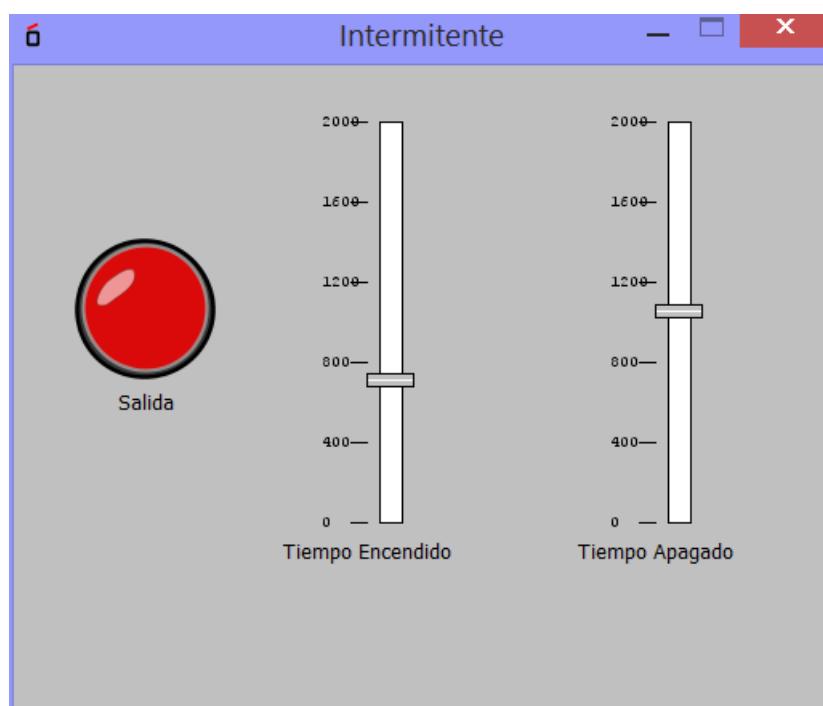
The figure shows the *numerical output* component with the component help area and the properties menu open. Do not forget that the properties shown by a component in this panel are different from those shown in the Circuit Panel.





In this panel, the components are placed looking for a harmonic visualization, since this will be the image that MyOpenLab will give us once we switch from editing mode to simulation mode. The figure shows the appearance of the

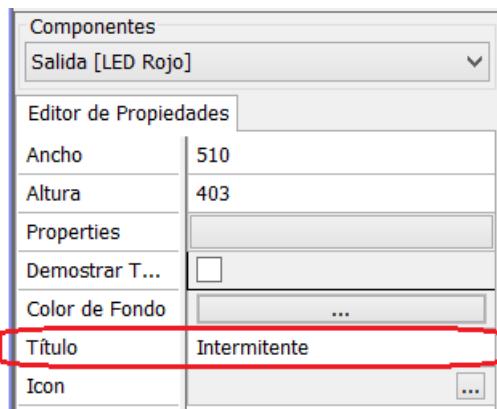
panel once we have clicked on simulation



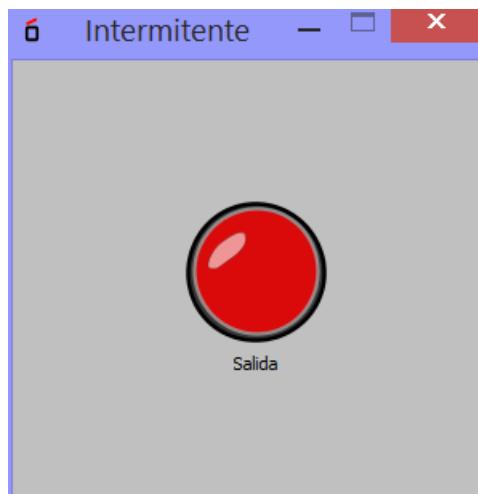
Don't forget that to stop the simulation you just need to press  eno  ende on the menu button bar.

Title for the window in simulation mode

In case we want the window to appear with a title, we only need to write it in the place indicated in the figure below

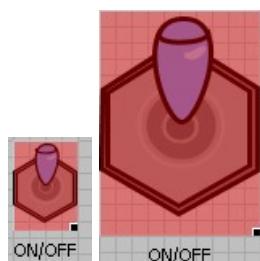


In which case the panel, when the application is run, displays the name.



Modification of the size of objects in the Front Panel

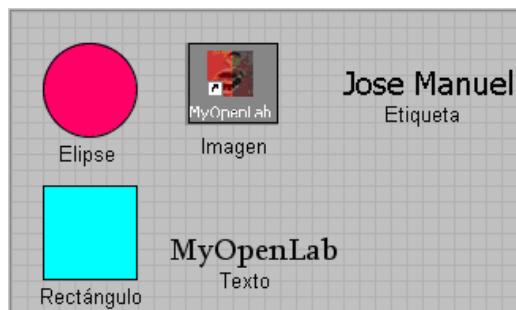
The objects displayed in the Display Panel can be resized as shown in the figure.



To do this, all we do is stretch from the black dot marked at the corners of the component when it is marked

Decorative objects.

The decorative objects are useful to make the panel since they allow us to approach the graphic reality of an instrument, system, machine, etc. See figure.



6. Element Libraries: Front Panel.



Front Panel Elements Library

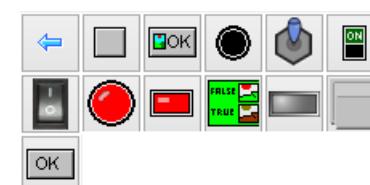
Decoration Elements



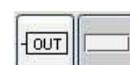
Numerical display elements



Digital activation elements



String Input and Output Elements



Input and output elements type vectors
and data arrays





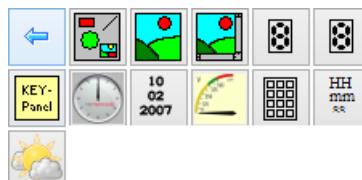
Elements of graphic visualization on coordinate axes I



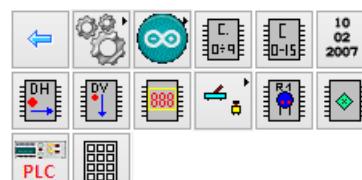
Elements of graphic visualization on coordinate axes II



Extra elements



User library elements



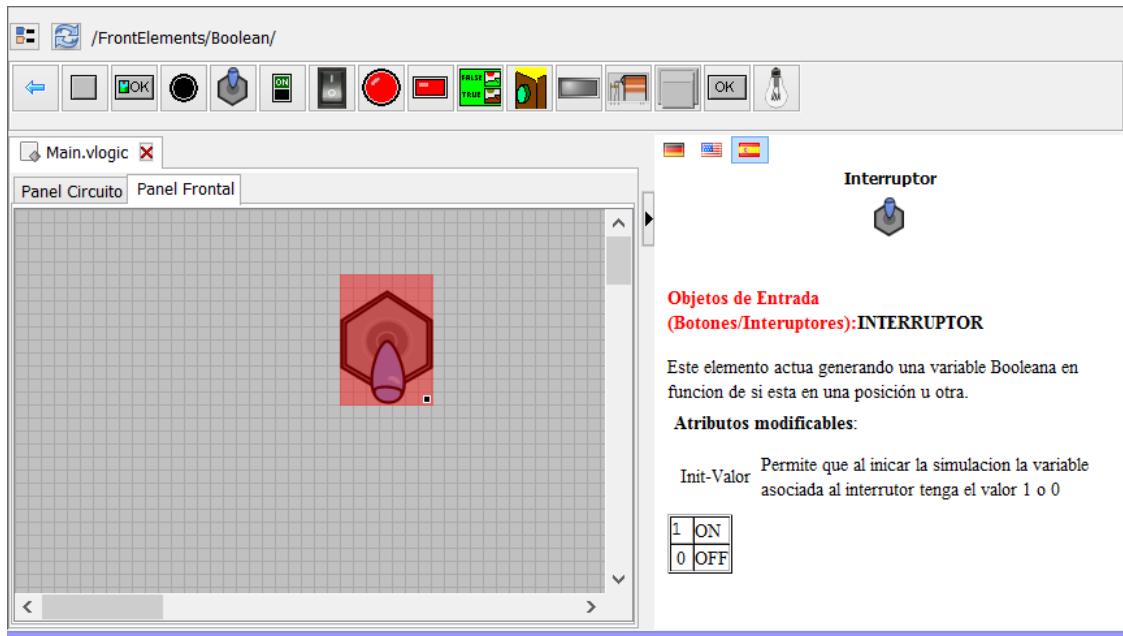
2D Robot



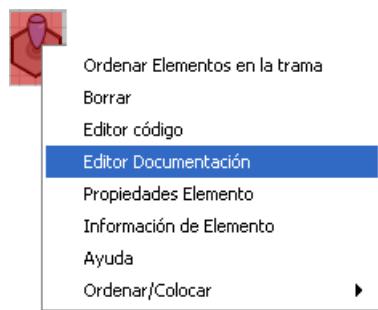
7. Help.

Simply highlight the component and you will see the help area on the right-hand side.

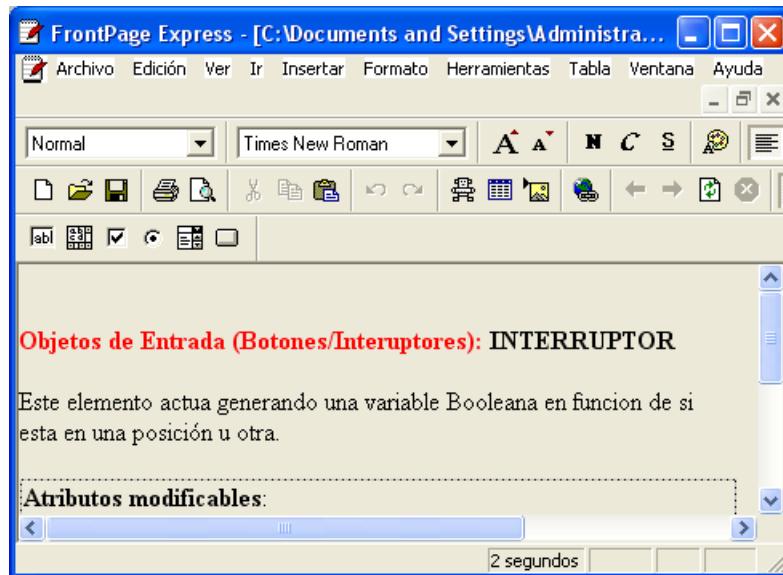
Help can be viewed in any of the three languages in which MyOpenLab is programmed.



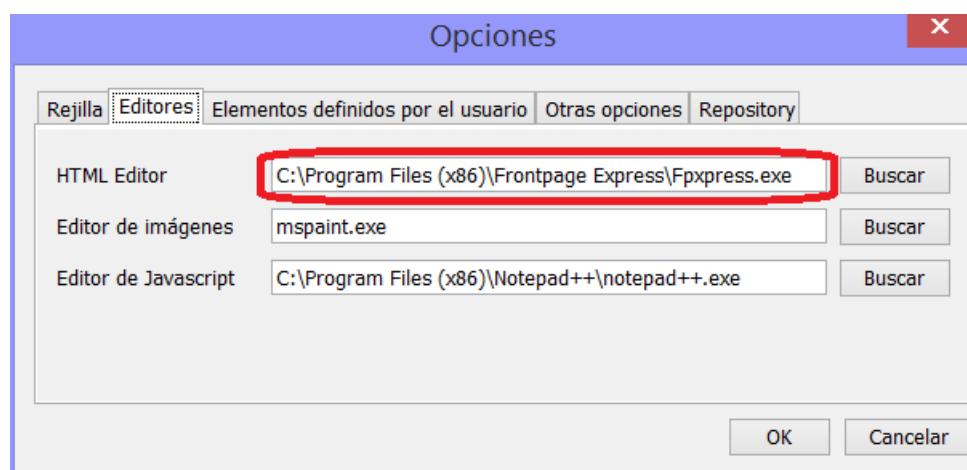
The help file is written in HTML language and can be edited at any time by clicking on the component and right-clicking on it and selecting "Documentation Editor".



The figure shows the appearance of the window for editing the help file of the switch component.



The selection of the HTML page editor for the help files can be configured, choosing the one we want. This can be done in the "**Extras -> Options**" option of the main menu.

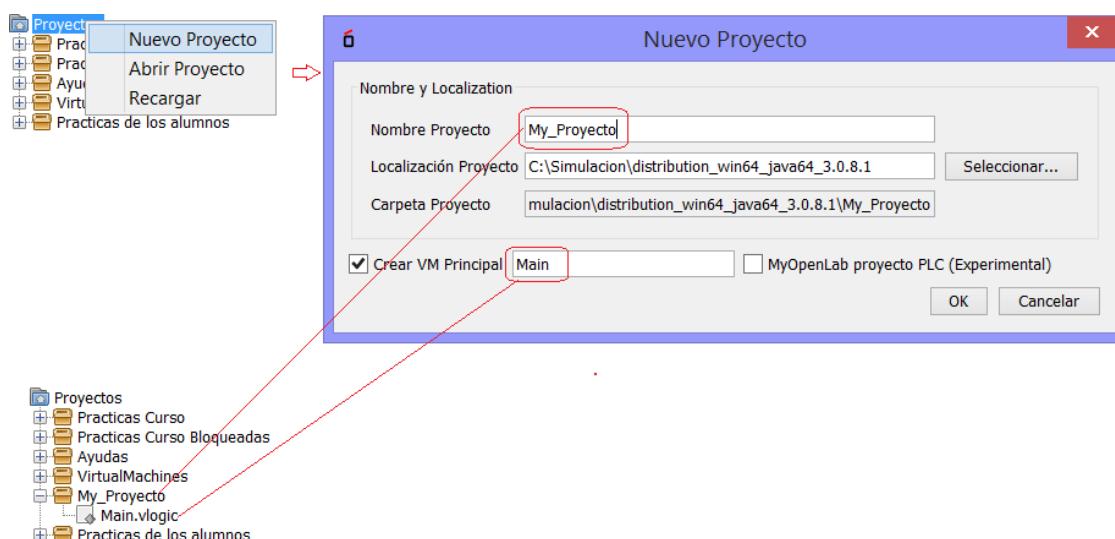


8. Organization of application files.

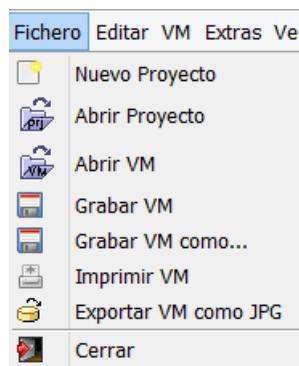
In MyOpenLab the organization of application files, also called VM files (Visual Models), is done through file structures called "**Projects**".

The projects physically constitute folders inside which the VM applications are located.

When creating a new project (by right-clicking on the "Projects" icon) a menu appears, select "New Project" and a window appears asking for the name of the new project, type the name of the project and if desired change the name of the main VM of the project, which is called "Main.vlogic" by default.



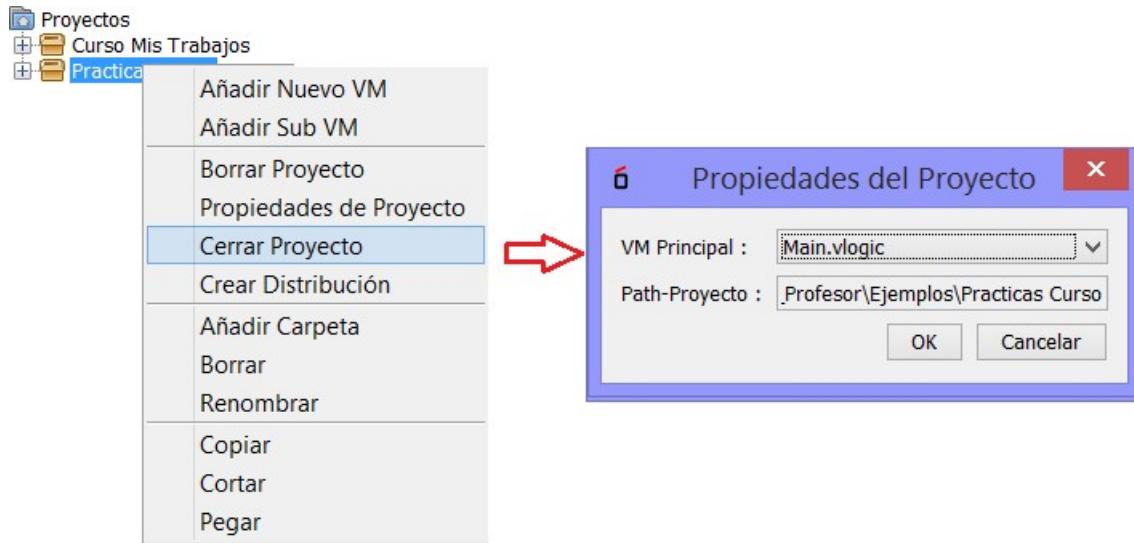
This first VM created is blank (without any element neither in the Circuit Panel nor in the Front Panel) and in it we can design our first application.



A new project can also be created by selecting the option from the application drop-down menu (New Project).

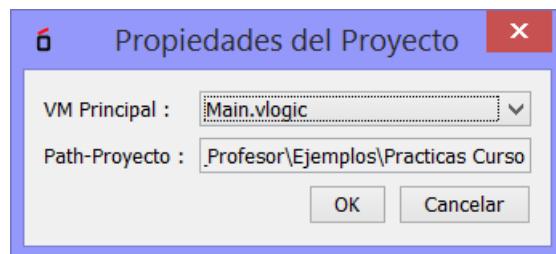
Project folders can be edited by simply placing the mouse over the folder and using the context menu (right mouse button).

We can see that the operations that can be performed with a project folder basically allow us to manipulate the VM application files that we have there.



It is important to highlight among the operations that appear the "**Create Distribution**" and "**Add Sub VM**" operations, which we will deal with later on.

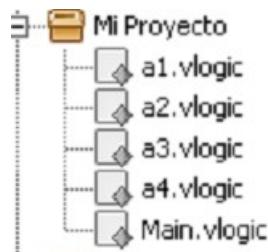
With the "**Project Properties**" option we can select the main VM of the application, as shown in the figure.



With the "Copy", "Cut" and "Paste" options we can change the folder of the different VMs we have created.

Example of the creation of a "Project Structure" with several associated VMs.

In this example, we will see the possibilities of the Sub -VM Element component.

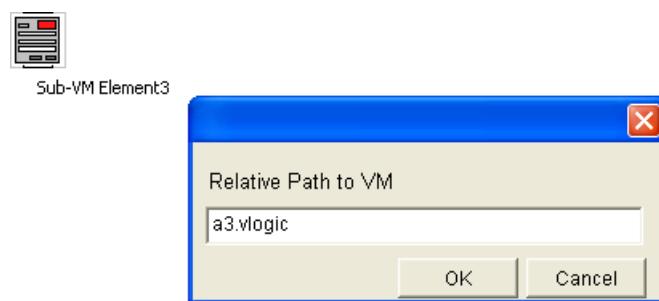


The following is an example of a project we have created in which 4 VM applications named a1, a2, a3, a4 are contemplated.

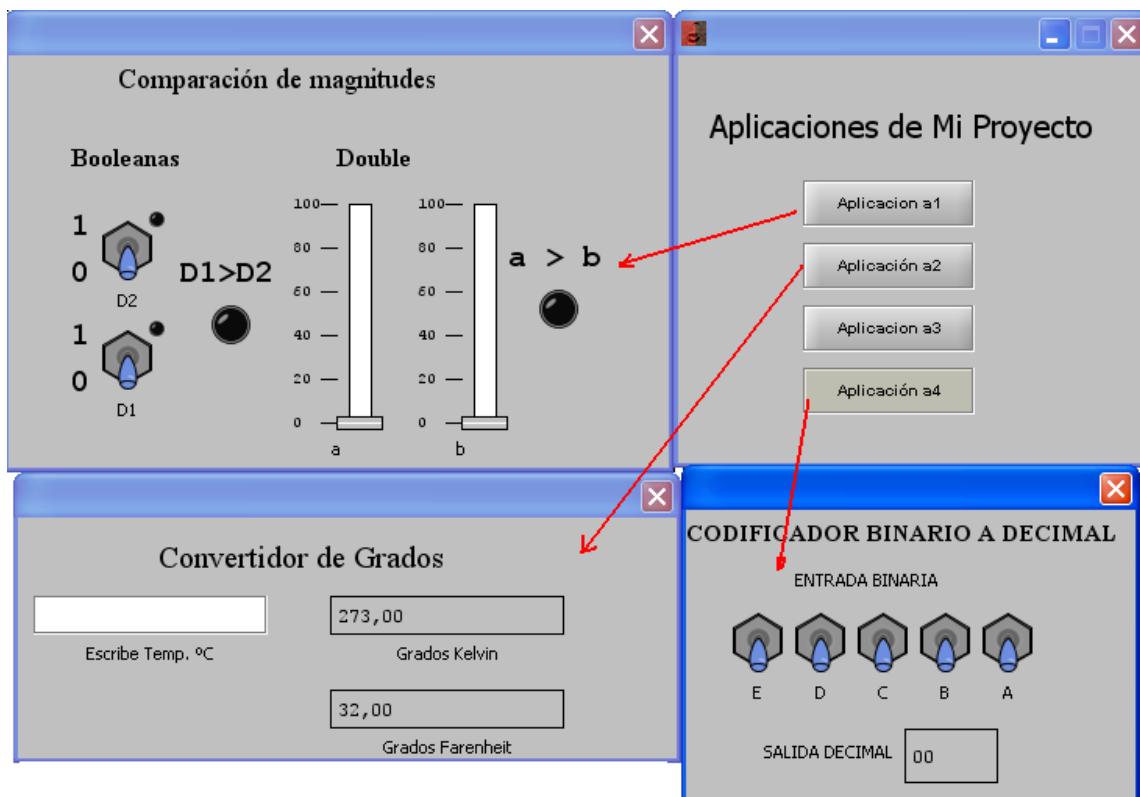
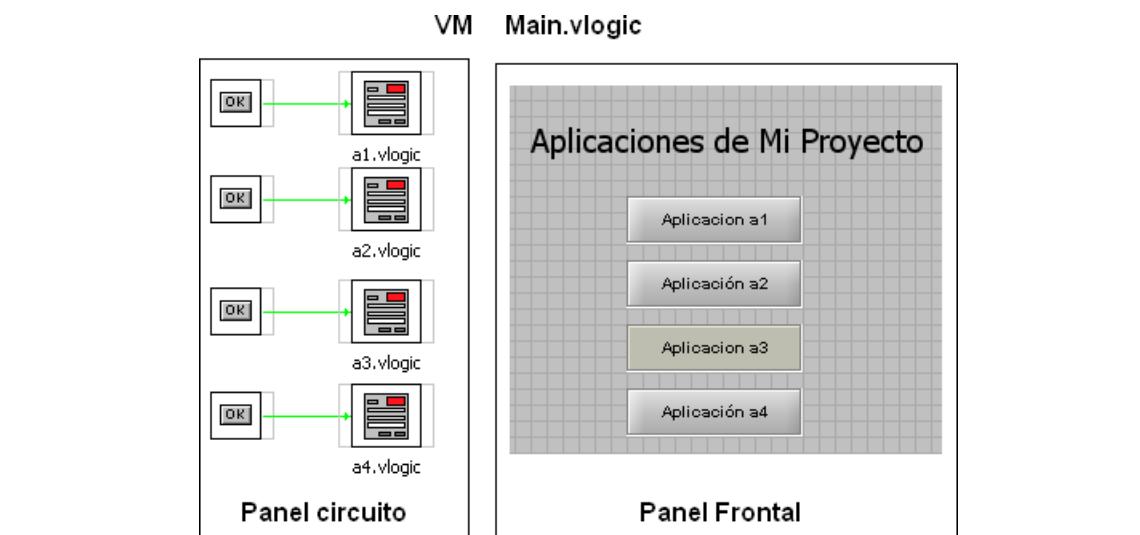
In the VM Main.vlogic we have placed four action buttons that invoke through the function "Sub-VM Element" to each one of them.

The figure shows the circuit and front panels of the VM Main, showing each of the invoked applications.

Once the configuration has been created, a "Distribution" of the application can be created by using the corresponding option in the project context menu.



The figure shows the window for collecting the name of the VM that we want to associate to the "Sub-VM Element".



In the figure we can see the definition of the VM Main in which the four buttons that activate the applications can be seen.

In the figure we see in "execution mode" the displayed screens of the applications.

This method of project creation allows in a quite comfortable way to structure our VM applications in themes and to be able to group them all in the same screen from which each one of them is opened.

9. Creation of VM sub-models (Sub-VM) to include within a generic VM application.

MyOpenLab, in cases where the model to be designed is very large, allows the creation of sub-models that respond to various functions of the model and that are encapsulated in a single component (Sub-VM).

These submodels are incorporated into the example folder tree and can then be incorporated into other new simulations (VM).

To explain the procedure, it is best to use an example.

The aim is to create a simulation of a digital circuit that we will integrate in two submodels: SubDig1 and SubDig2.

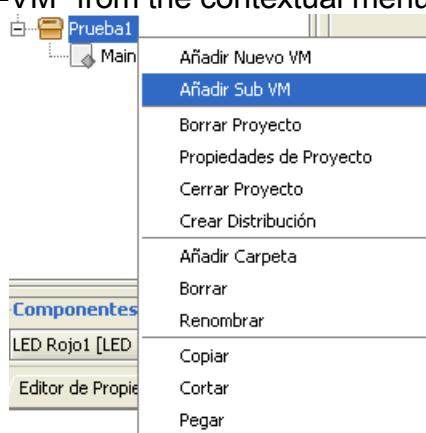
The creation of a submodel can be done in several ways depending on where we resort to within the environment.

Creation of a SubVM SubModel.

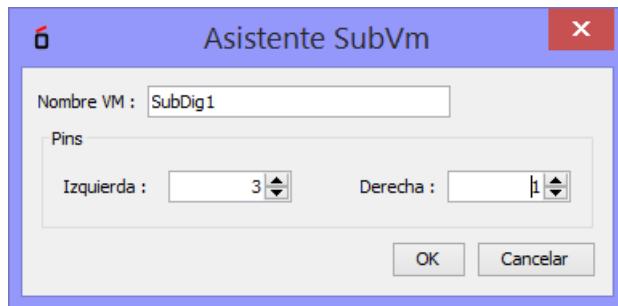
Method 1

In our example we will do it by creating the submodels directly from the Project tree. Steps to follow:

- 1.- It will be enough to place ourselves on the name of the project folder in which we are going to save the application and pressing the right button we select "Create Sub-VM" from the contextual menu.



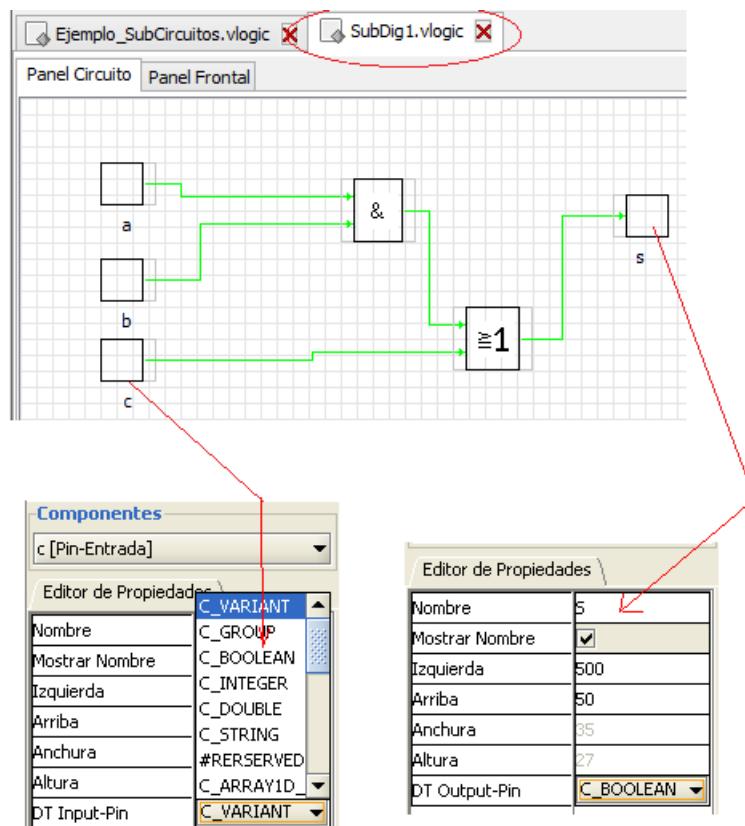
- 2.- Next we select the number of input and output pins that our Sub-VM will have, on the left (inputs) and on the right (outputs).



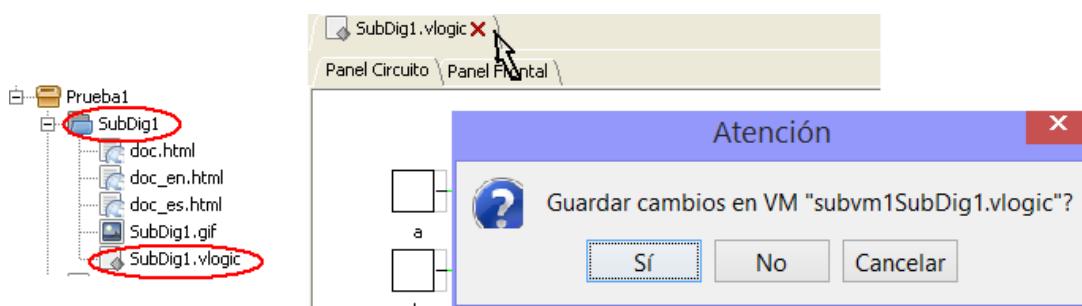
When accepting, once the number of pins has been selected, the following structure will appear in the "Circuit Panel".



- 3.- The next step is to select the components that will be part of the Sub-VM and designate the nature of the input/output pins. When defining the pins we can also give each one the name "terminal label" that will be used to identify each input/output.

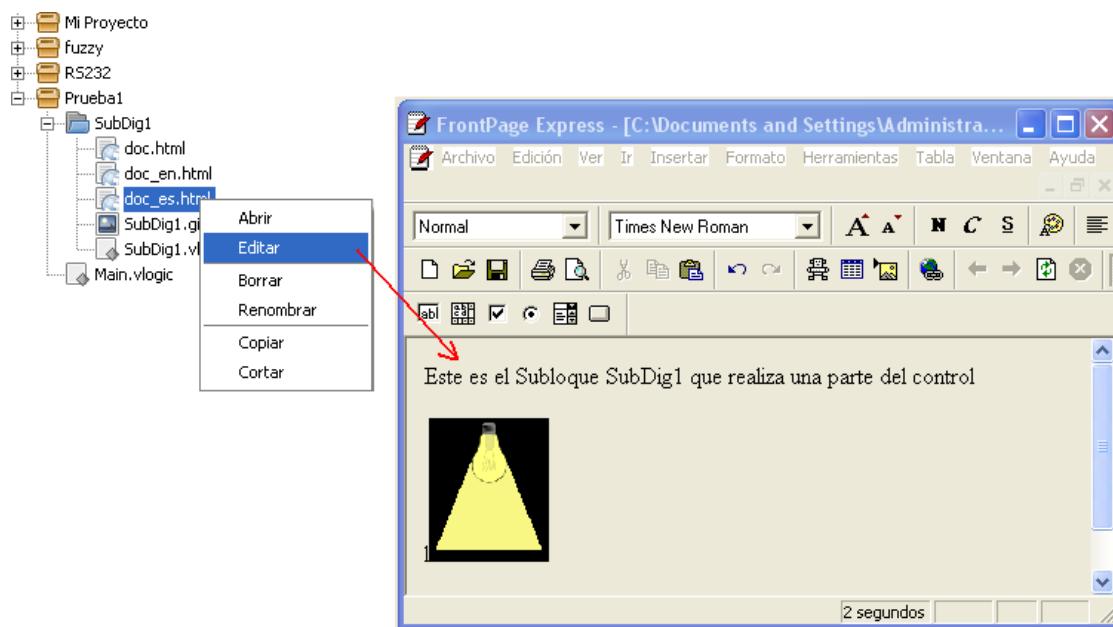


- 4.- Once the complete schematic of the Sub-VM is created, we proceed to save it. Click on the red cross on the tab and press "Yes" and the file will be saved as shown in the figure.



We see that the Sub-VM has placed it in a folder that receives the same name as the file itself (SubDig1) and also created three .html files to be able to place in them the help information that we want in the three languages of MyOpenLab, as well as a file (SubDig1.gif) that will be the icon that we can put to this Sub-VM.

Editing the SubDig1.html file



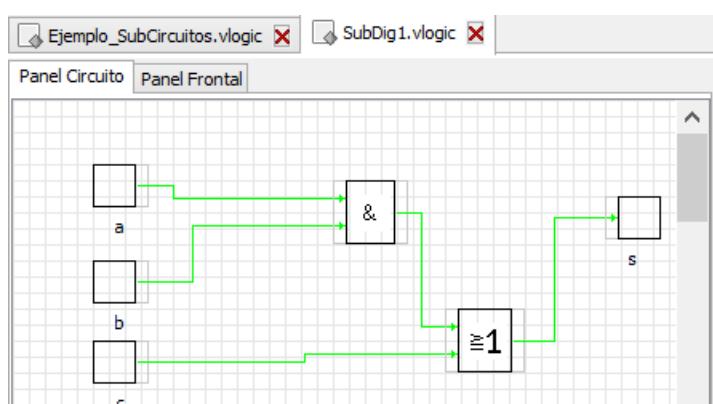
It is also possible to edit the icon and change it for another one.

Editing the icon associated to the Sub -VM SubDig1

SubDig1 icon



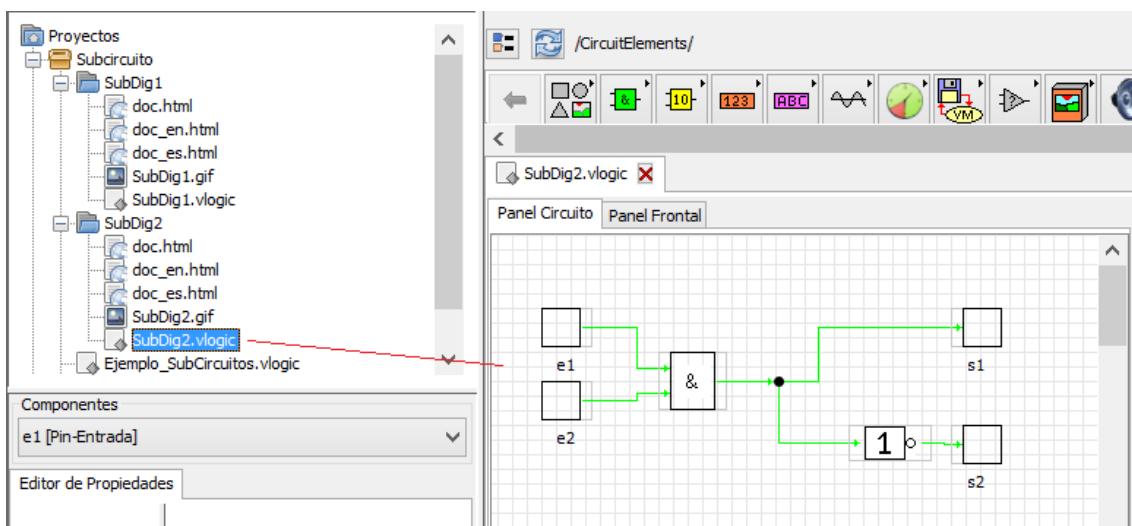
Finally we see the aspect of how the created submodel would look like. It would be enough to create a new VM and dragging the file SubDig1.vlogic to the area of the "Circuit Panel" it would be there symbolized with its new icon and as we see its pins that would be perfectly identified in the "Help Window". If we want to see the content of the SubDig1.vlogic element, we just have to right click on the mouse button, show the contextual menu that appears in the figure and select "Show VM", opening the complete schematic of this figure.



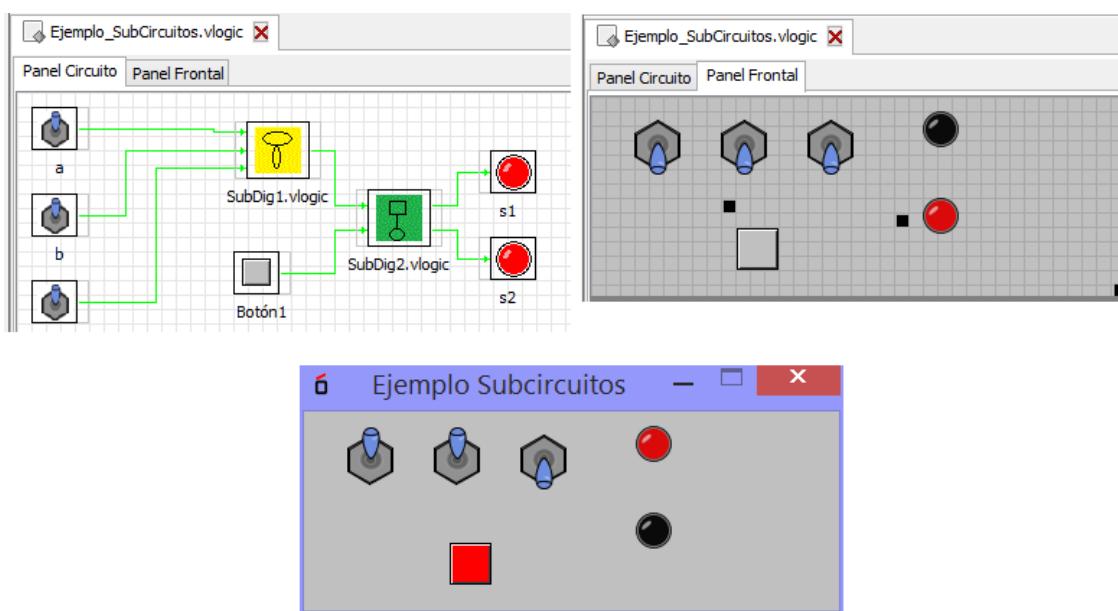
By clicking on "Show VM" we see the content of SubDig1.vlogic

In the created Sub-VM, display elements of the "Front Panel" can also appear, such as an LED.

- 5.- Once the SubDig1 and SubDig2 subcircuits have been created, to incorporate them into a design it will be enough to create a new VM and drag each of the elements onto it.

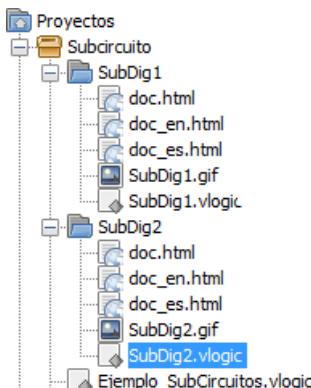


The figure shows how the VM Example_Subcircuits .vlogic would look like.



Appearance of VM Digi tal1.vlogic application

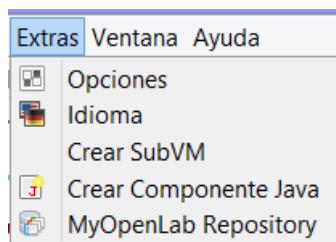
In the following figure we can see how the "Projects" tree would look like.



Figra 49

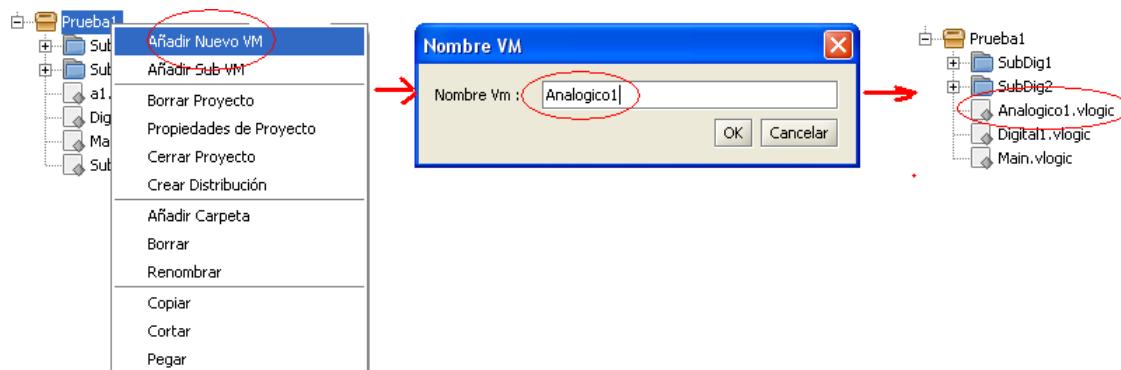
Method 2

Another way to create a Vm submodel structure is to use the "Create SubVM" option in the Extras menu of MyOpenLab, Figure.



The procedure would be as follows.

- 1.- We create a new application having marked the project folder in which we want to deposit it, selecting



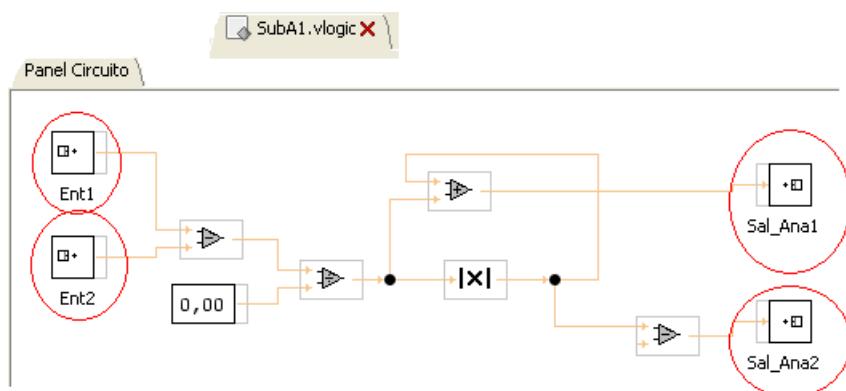
- 2.- On the "Circuit Panel" we place all the elements we need and we make the schematic as what later will be the SubA1.

The selection of the area that we make will be the one that will host the components that we are going to integrate in the new Sub-VM that we will call SubA1.

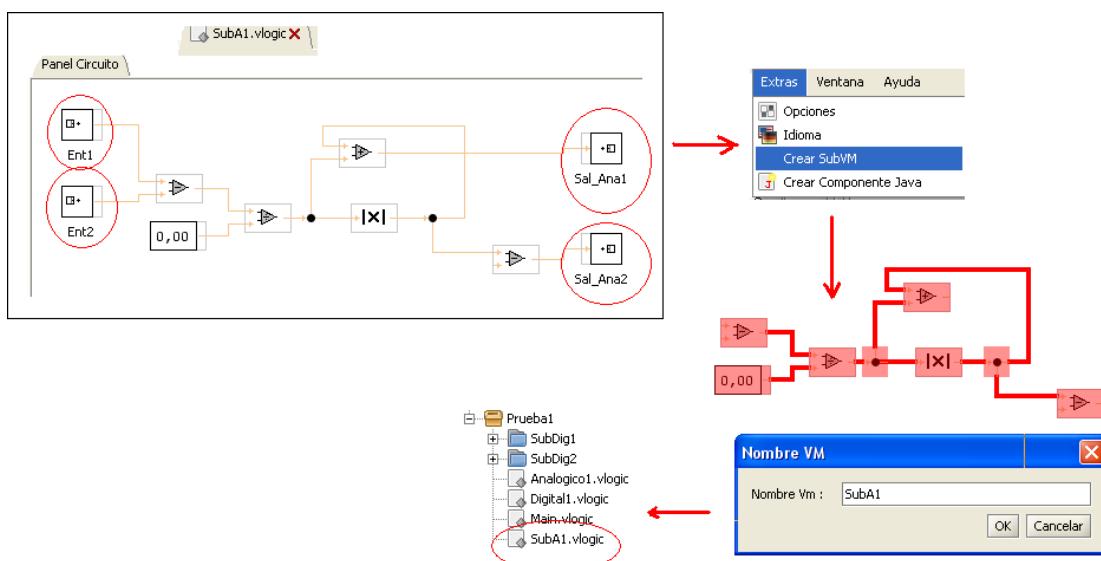
The next step is to define the input and output pins of the new element created, taking care to define the pin name and the nature of the signal to be managed, whether it is an input or output signal.

- 4.- Once the pins have been defined, the next step is to save the new Sub-VM with the name SubA1. To do it we go to the menu **Extras -> Create SubVM** we move the mouse to the work area of the Circuit Panel (the cursor has become a cross) we mark all the components area including the input/output pins and when we release a window will appear for us to put the name of the file that is going to be saved. It will be saved in the project folder where we are, not in the possible subfolders inside the project folder.

Designation of the input and output pins of the Sub -VM SubA1.vlogic



Next select the option **Extras ->CreateSubVM**



This new Sub-VM will be available for when we need to use it.

Simply drag the Sub-Vm to the work area of the "Circuit Panel" and we will have our new SubA1.vlogic.

Note that the component appears with its inputs and outputs labeled perfectly.

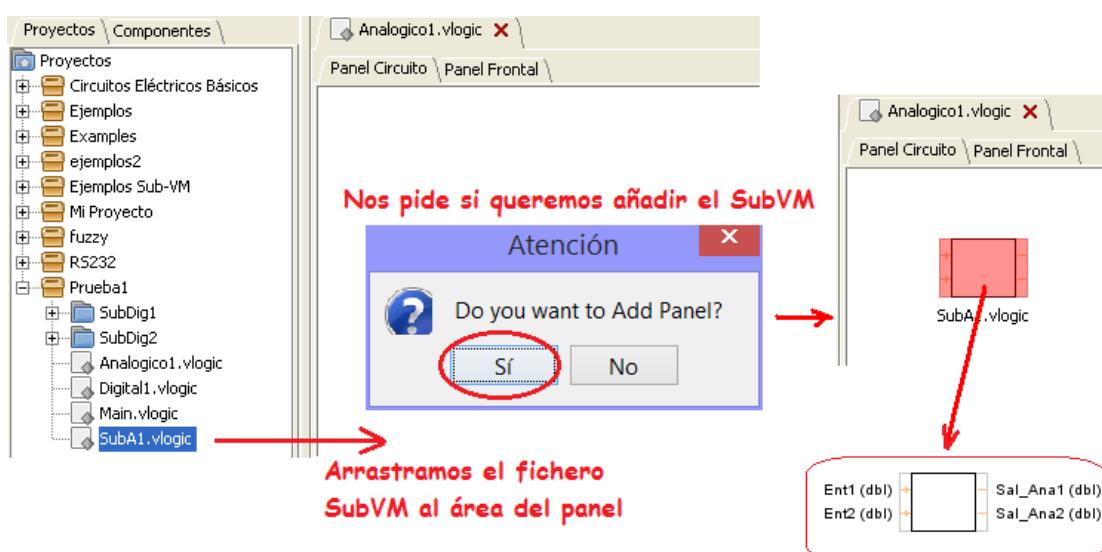
Integration (loading) of a SubVM in a VM application

To load any Sub_VM created in our application we can do this in two ways

1st Form.

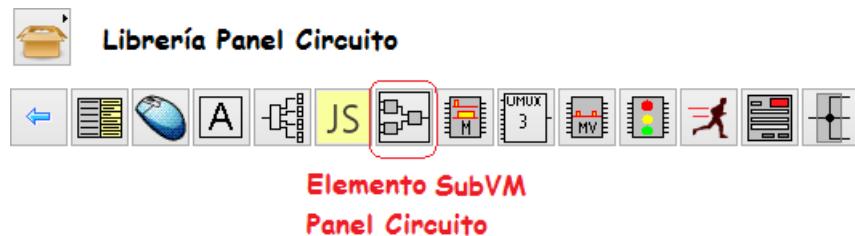
Simply drag the Sub-VM to the "Circuit Panel" work area and we will have our new SubA1.vlogic.

Note that the component appears with its inputs and outputs labeled perfectly.

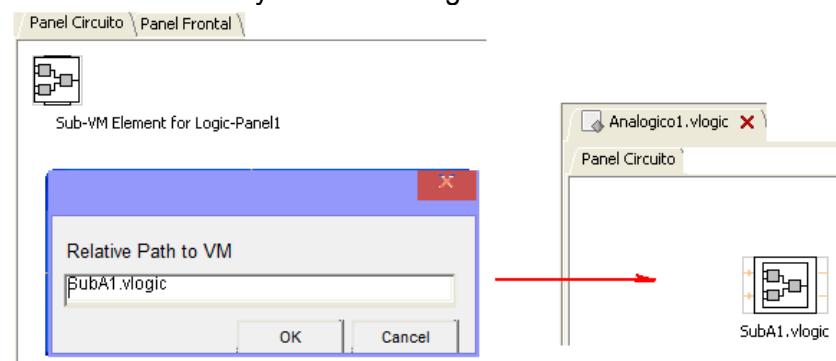


2nd Form of Loading a SubVM

This way will consist of making use of the library component marked in the following figure.



This component allows you to directly write the name of the file associated with Su b-VM and include it in the model you are creating.



10. Repository.

This tool allows you to manage the libraries we work with in MyOpenLab as well as the collections of VM examples we can create.

The tool is prepared to work in offline and online mode.

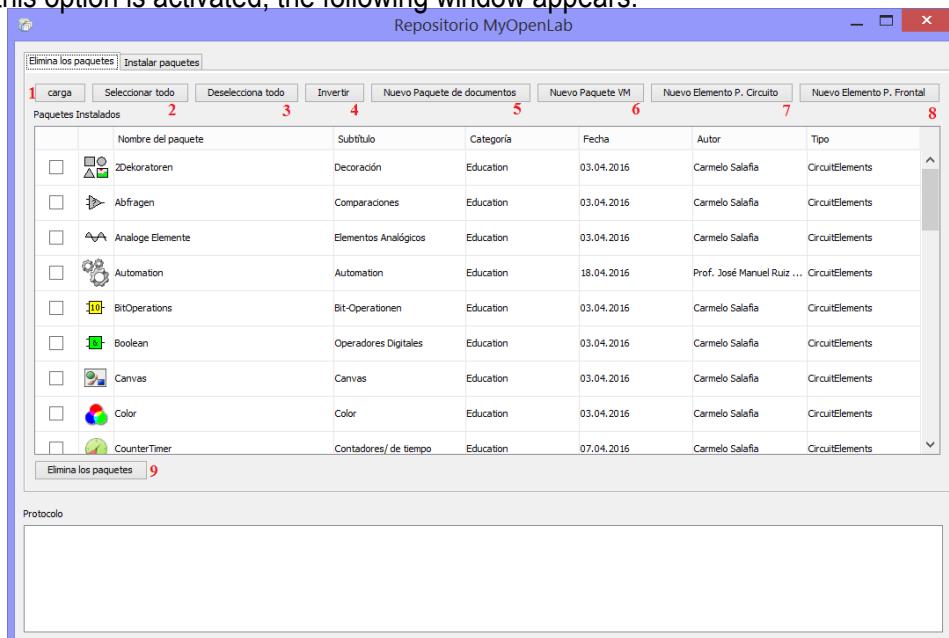
The utility can be accessed in two ways: through the icon in the toolbar or in the drop-down menu

Acceso a MyOpenLab Repositorio



Create or remove packages from our application.

When this option is activated, the following window appears:



What is displayed on the screen is the set of library packages, documents and VMs installed on our machine in the MyOpenLab folder.

Options of the tab "Remove packages".

- 1 Reload libraries
- 2 Select all libraries
- 3 Deselects all libraries.
- 4 Switches selection/deselection in switch mode
- 5 Creates a document package

- 6 Create a VM package
- 7 Creates a new user library of type Circuit Panel
- 8 Create a new Front Panel user library
- 9 Removes the selected packages.

Each library contains the following identifiers

	Nombre del paquete	Subtítulo	Categoría	Fecha	Autor	Tipo
<input type="checkbox"/>	2Dekoratoren	Decoración	Education	03.04.2016	Carmelo Salafia	CircuitElements

Package name

Subtitle

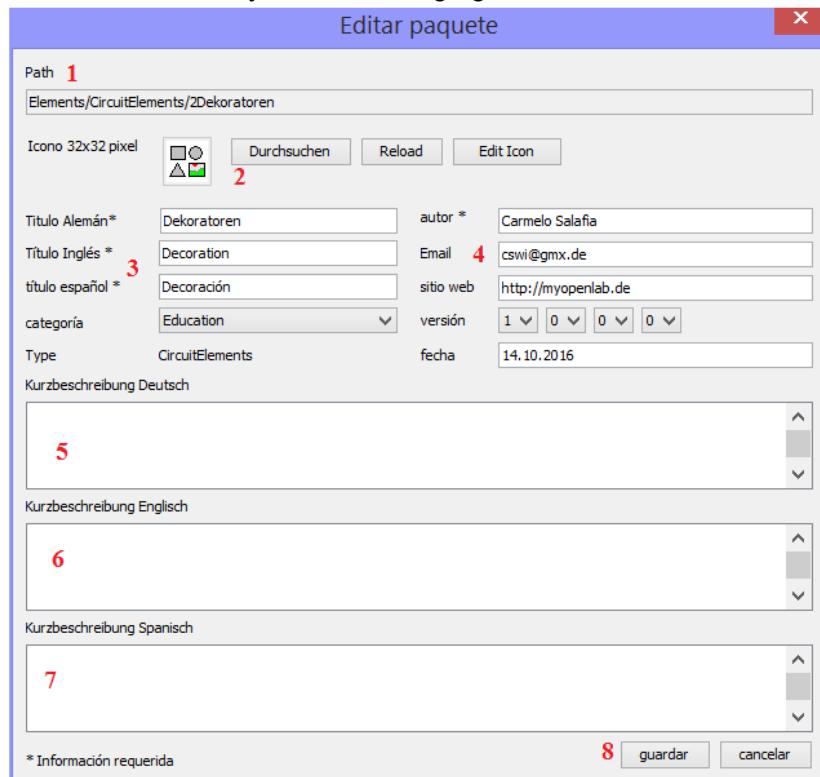
Category

Date

Author

Type

If we click twice on the library line with the mouse, a window appears, which is the parameter window of that library. The following figure shows it.



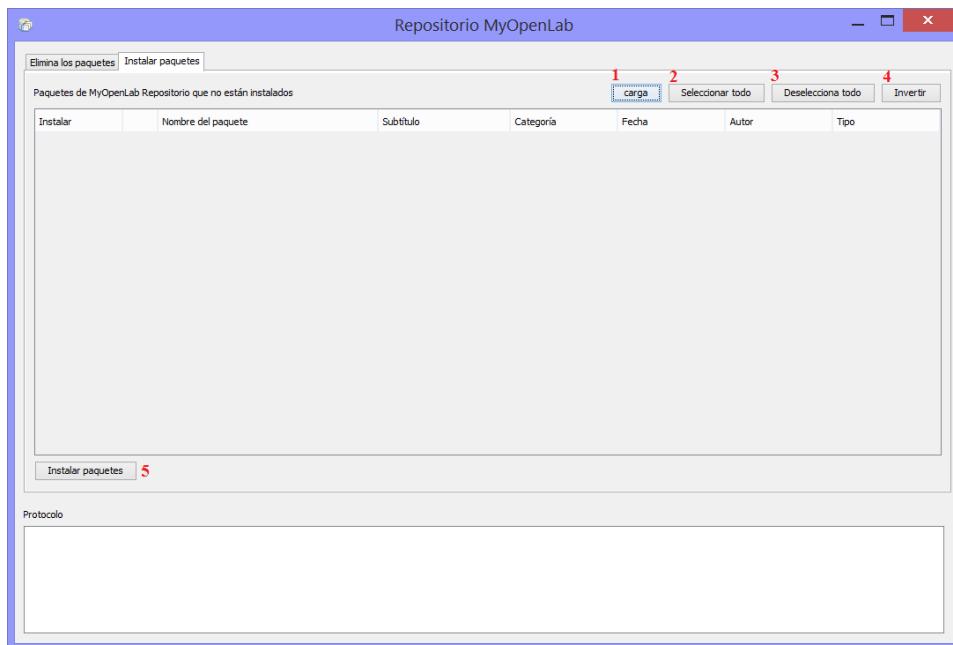
- 1 Patch in which the freeara is located in the MyOPenLab folder
- 2 Editing the library icon. Allows you to load, reload or update and edit the icon.
- 3 Text identifying the bookstore in each language and Category of the bookstore
- 4 Data of the library author, date and version
- 5,6,7 Detailed description of the library

Download or upload a library:

To download a library to our machine or take it to the remote repository, we can use the two options shown in the context menu that appears when we right-click on the library.



Installing packages from remote repository



Install Packages tab.

1 Load packages that are not installed

2 Select all packages

3 Deselects all packages

4 Switches selection/deselection in switch mode

With the Load option will appear the libraries that we do not have loaded in our application and that appear in the repository. If they are all installed, logically none of them will appear here.

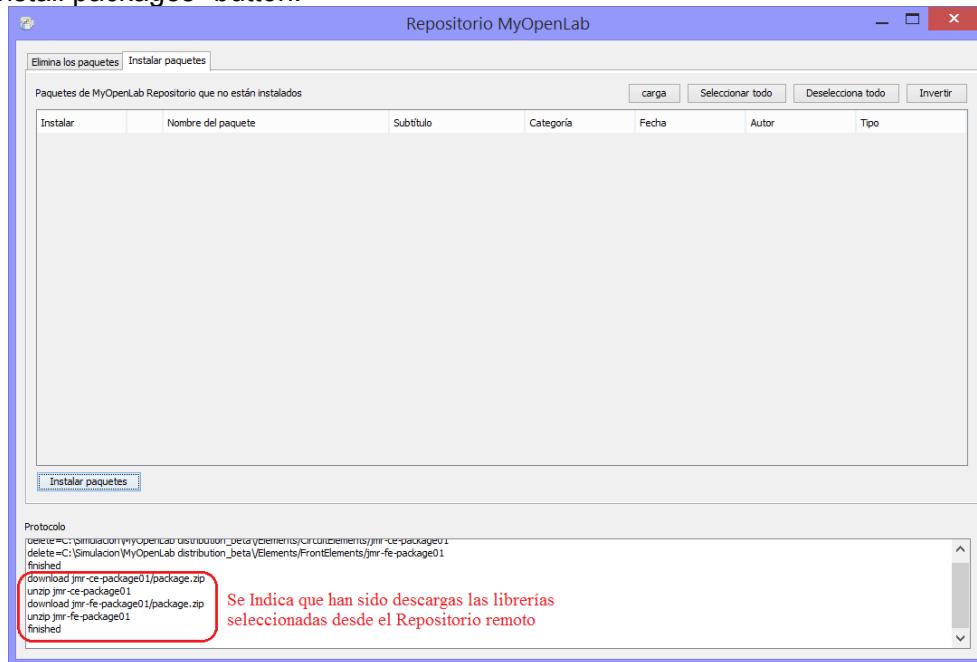
When we select delete packages, they disappear from our environment and this window is displayed



If we click on Load now, the list of libraries is refreshed and the two deleted libraries are no longer displayed.

NOTE: It is important to make sure of the deletion because if we cannot connect to the repository via Internet we will not be able to rescue the library.

If we now go to the Install packages tab and click on Load we will get the list of the two libraries that are no longer installed and that can be reloaded if we select them and click on the "Install packages" button.



Creation of new library packages, VMs or Documents.

We will now explain how to create a library that will be incorporated into our MyOpenLab distribution installed on our machine.

We are going to create a library of the Circuit Panel type. To do so, click on the "New Circuit Panel Element" button and the window shown below will appear.

Editar paquete

Path
C:\Simulacion\MyOpenLab distribution_beta\Elements\CircuitElements\ce-bb2bfbcc-5645-4bd3-adf3-920629f92794

Icono 32x32 pixel Durchsuchen Reload Edit Icon

Título Alemán*	<input type="text"/>	autor *	<input type="text"/>
Título Inglés *	<input type="text"/>	Email	<input type="text"/>
título español *	<input type="text"/>	sitio web	<input type="text"/>
categoria	Education	versión	1 ▾ 0 ▾ 0 ▾ 0 ▾
Type	CircuitElements	fecha	14.10.2016

Kurzbeschreibung Deutsch

DE
DE
DEdfsdfs

Kurzbeschreibung English

EN
EN
EN

Kurzbeschreibung Spanish

ES
ES
ES

* Información requerida

Click on the "Save" button. We will have previously filled in the data.

Editar paquete

Path
Elements/CircuitElements/ce-bb2bfbcc-5645-4bd3-adf3-920629f92794

Icono 32x32 pixel Durchsuchen Reload Edit Icon

Título Alemán*	<input type="text" value="Mi Librería"/>	autor *	<input type="text" value="Jose Manuel"/>
Título Inglés *	<input type="text" value="Mi Librería"/>	Email	<input type="text"/>
título español *	<input type="text" value="Mi Librería"/>	sitio web	<input type="text"/>
categoria	Education	versión	1 ▾ 0 ▾ 0 ▾ 0 ▾
Type	CircuitElements	fecha	14.10.2016

Kurzbeschreibung Deutsch

Esta es una librería de Ejemplo

Kurzbeschreibung English

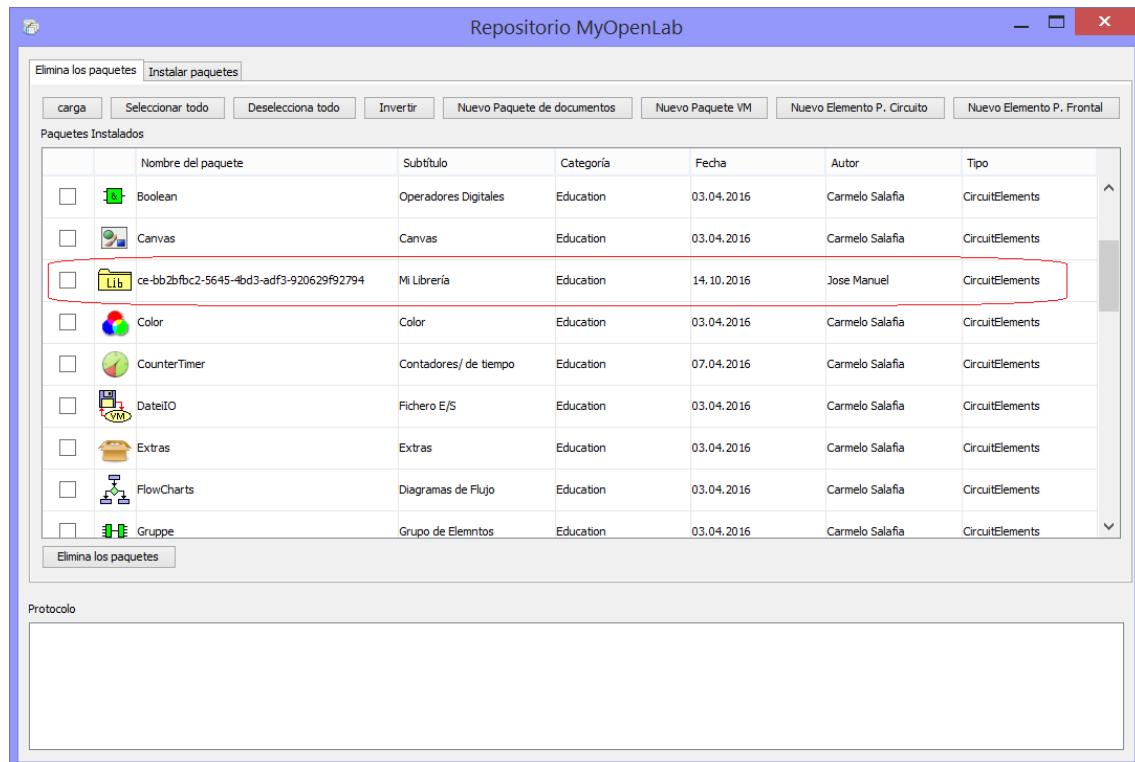
Esta es una librería de Ejemplo

Kurzbeschreibung Spanish

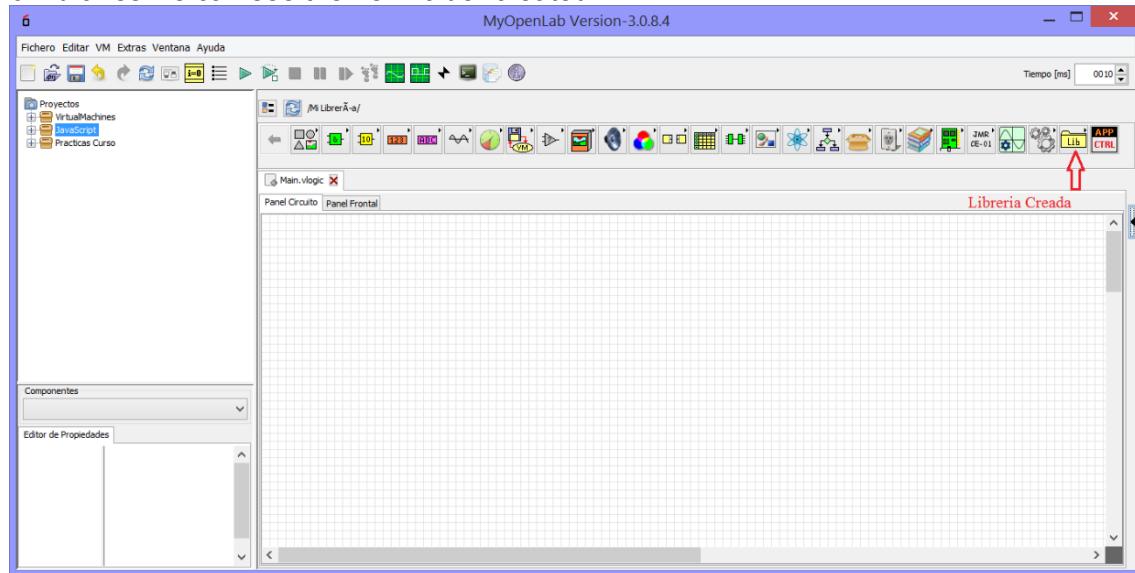
Esta es una librería de Ejemplo

* Información requerida

In this way, the folder in which we will place our library elements is created.



If we go to the environment and change tab Circuit Panel Front Panel to refresh the list of libraries we can see the new folder created.



11. Creation of a new element in the User library

Once we have created our user library folder, we can start including new library items as we create them, so that we can extend the MyOpenLab base libraries with our own contributions.

We will now explain how to create a new library element that will calculate the arithmetic mean of two numbers.

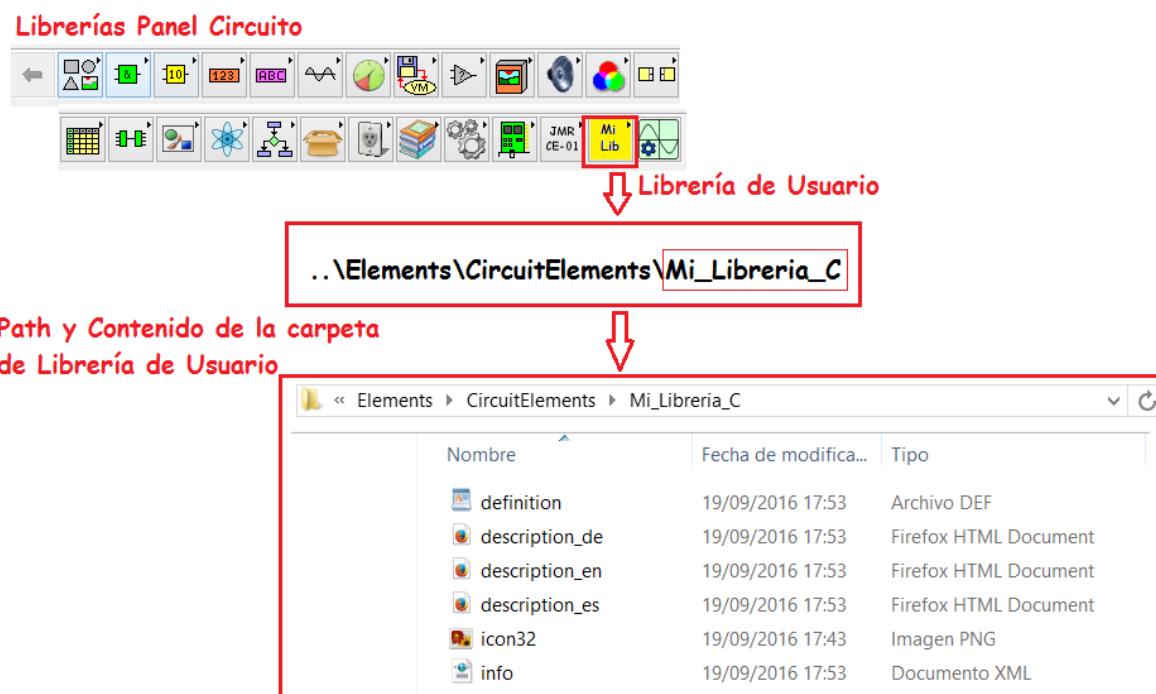
Method 1

IMPORTANT: In case of using a version prior to MyOpenLab 3.0.8.4 version

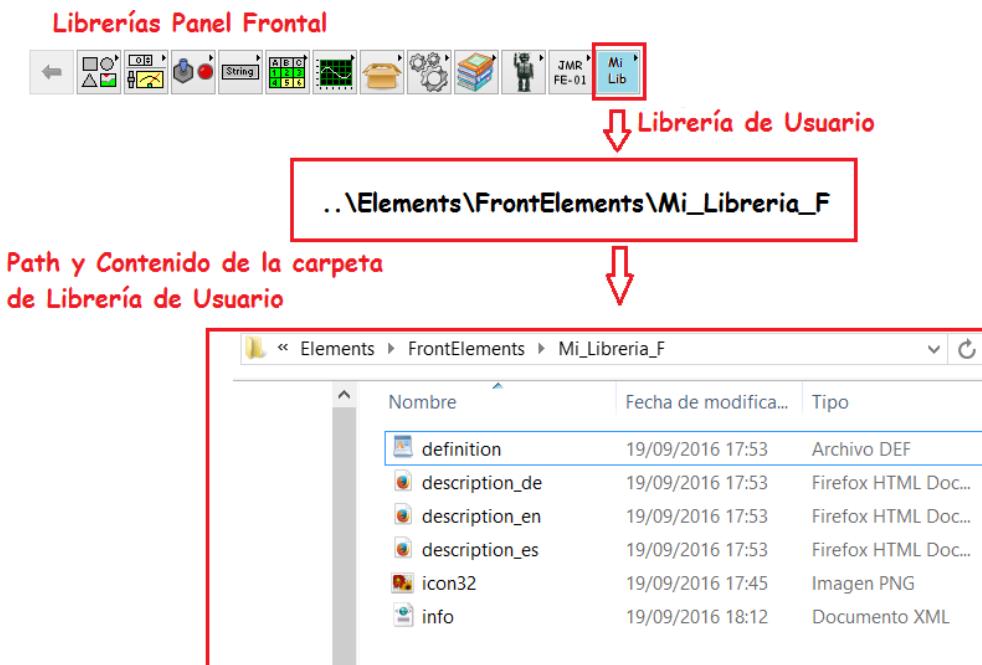
Without prejudice that in the next versions of MyOpenLab there will be another procedure for the creation of a library that is more automated, in our case we are going to describe the steps to create two new library folders in each of the main libraries: Circuit Library and Panel Library.

The only thing we have to do to make this new folder appear in each of the libraries is to go to:

- The ...**Elements** folder in the case of the Panel Library Circuit and copy the folder My_Bookstore_C to that folder



- The ...\\Elements\\FrontElements folder in the case of the Front Panel Library and copy the My_F_Library folder to that folder.



These folders, once they are copied into place, will appear in the environment, each one in its place, and we are ready to create library components in them according to the instructions that will be explained later.

The contents of these folders are as follows:

- icon32.png** This is the icon file that will show the folder in the library bar.
- description_de.html** Help file about the content of the folder, it is empty and it is not necessary to write anything in it, it is for German language.
- description_en.html** Same as above but for English language.
- description_es.html** Same as above but for Spanish language.
- info.xml** It contains information about the folder name, author and library. No need to touch anything
- Definition.def** Contains information necessary for the system, it is not necessary to touch it at all.

Creation of a library block inside the User library in case of having a MyOpenlab version prior to 3.0.8.4

For the case of once we have the folders in their corresponding locations if we open the environment we will see that these folders appear in color

yellow and the other in blue.

The user can then change the icon to the one he/she likes best.

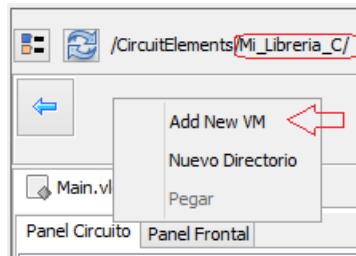
We are going to create a library within the User library in the set of Circuit libraries to calculate the arithmetic mean of two numbers.

If you have version 3.0.8.4 you have already created the library folder with the method explained in the previous point (pinto 10 Repository).

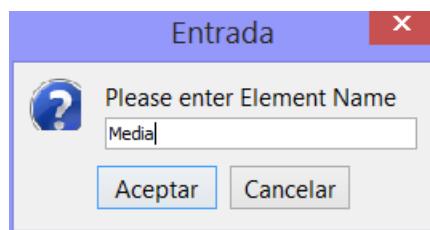
Let's see the steps to follow:

Go to the library bar of the Circuit Panel and open the folder **Mi.L** or the folder we created and we have the version 3.0.8.4.

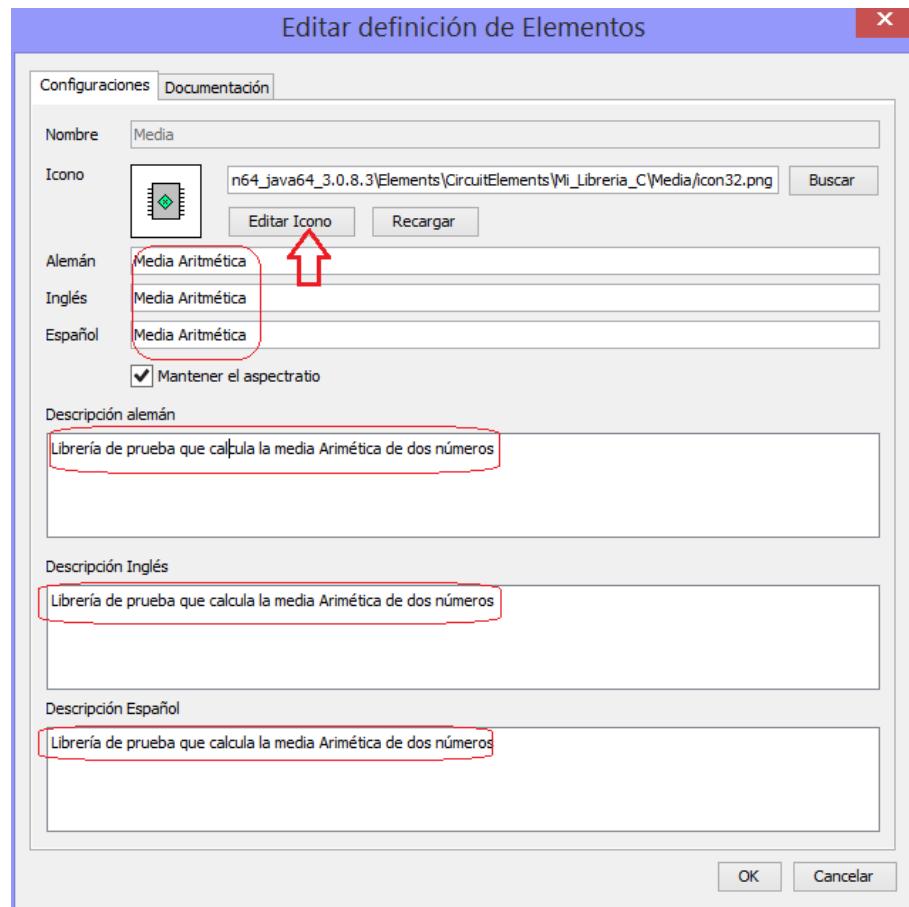
We will see that it is empty. Right-click on the element bar of the library and the context menu shown in the figure appears.

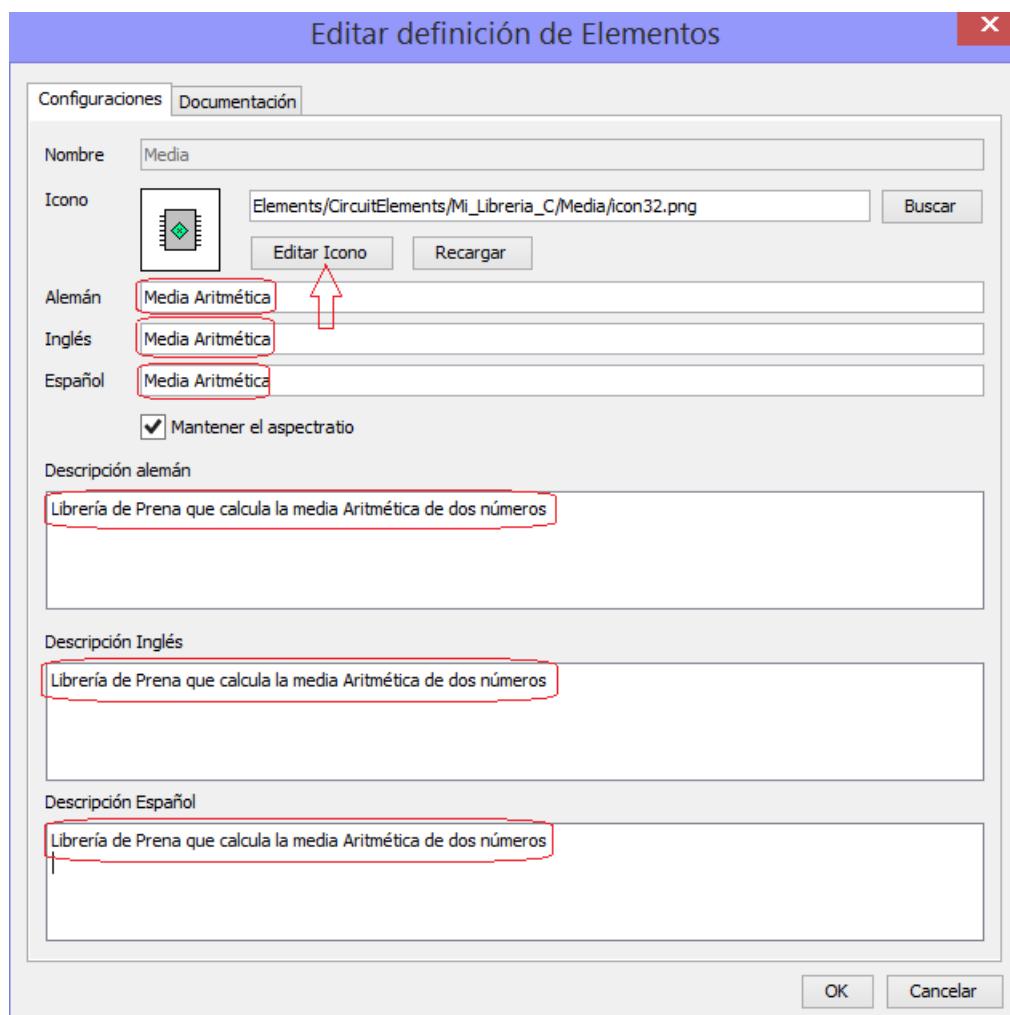


select **Add New VM** and the following window will be displayed

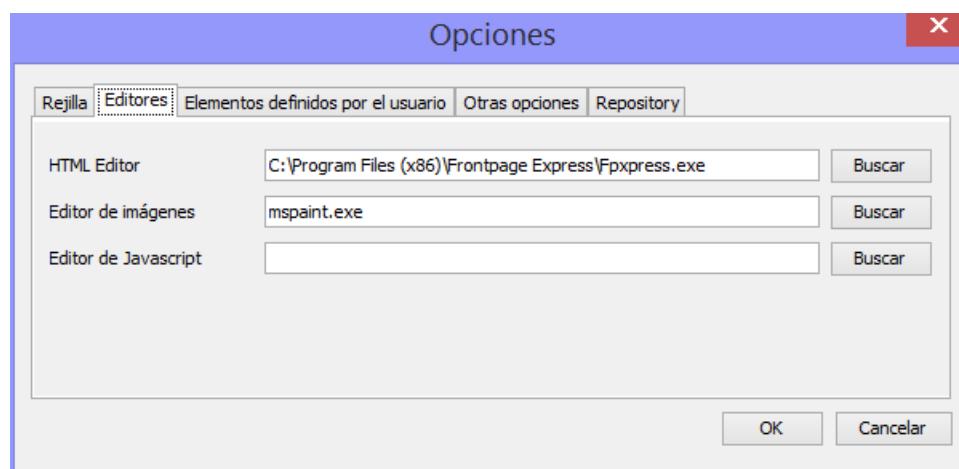


Type the name of the file to be created "Media" and click OK. The following window will appear on which we write the data of the library and design the icon .

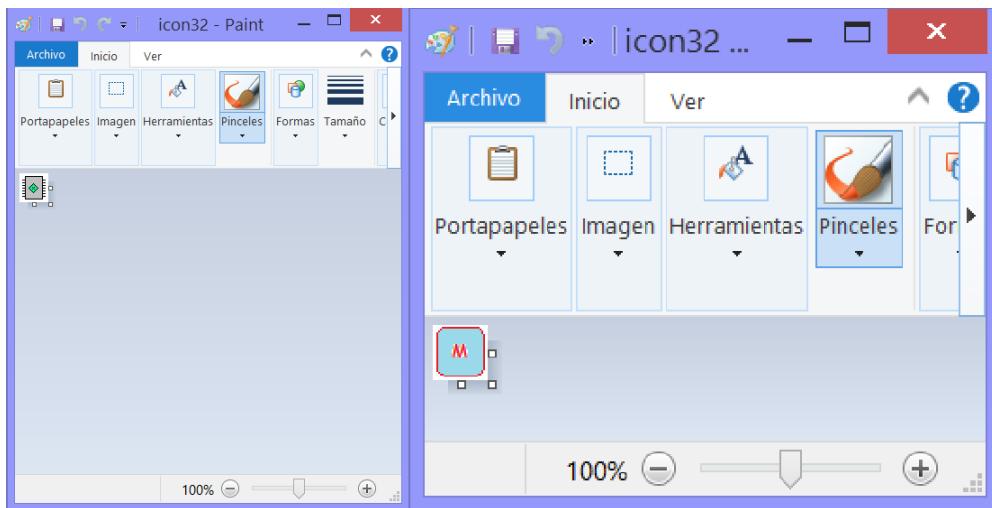




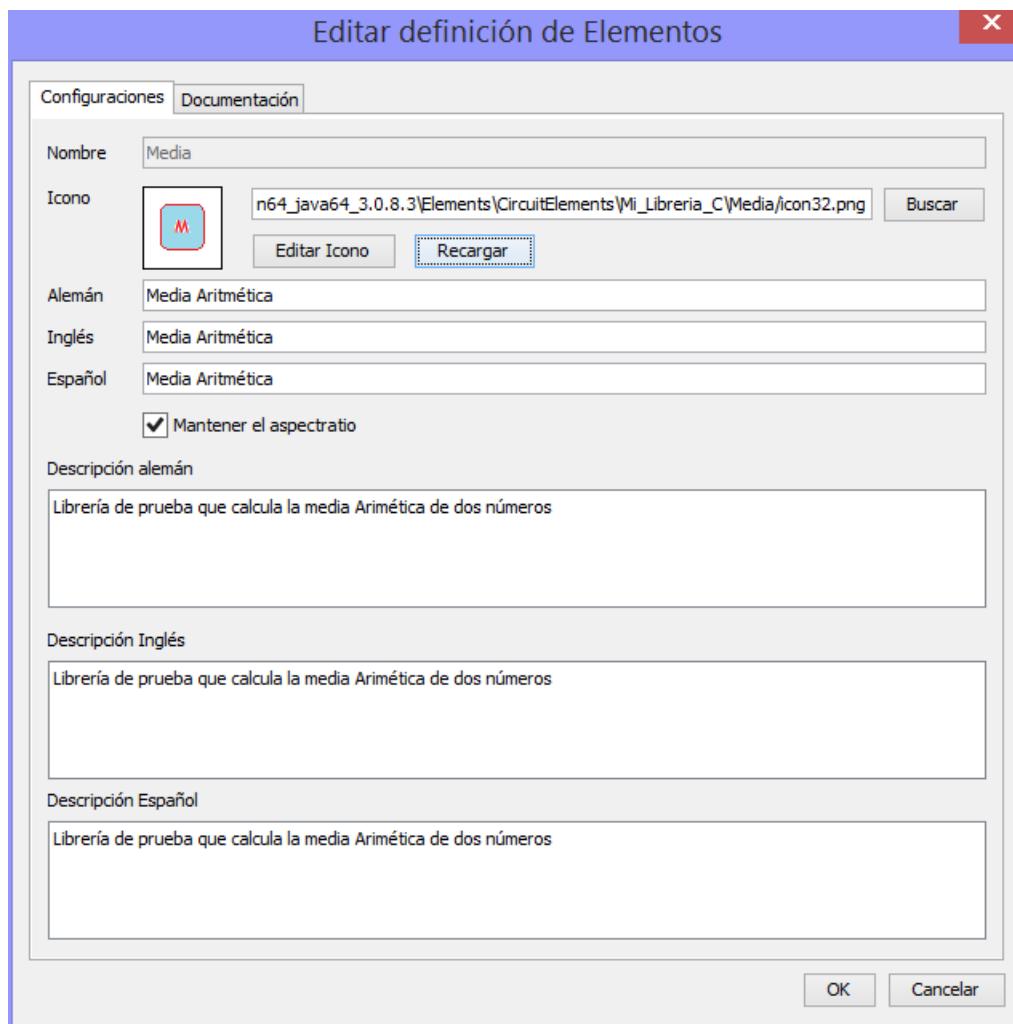
We edit the library icon by clicking on the **Edit Icon** button using the **mspaint** program. This program opens because previously in the configuration options we have already indicated that this will be the program that opens.



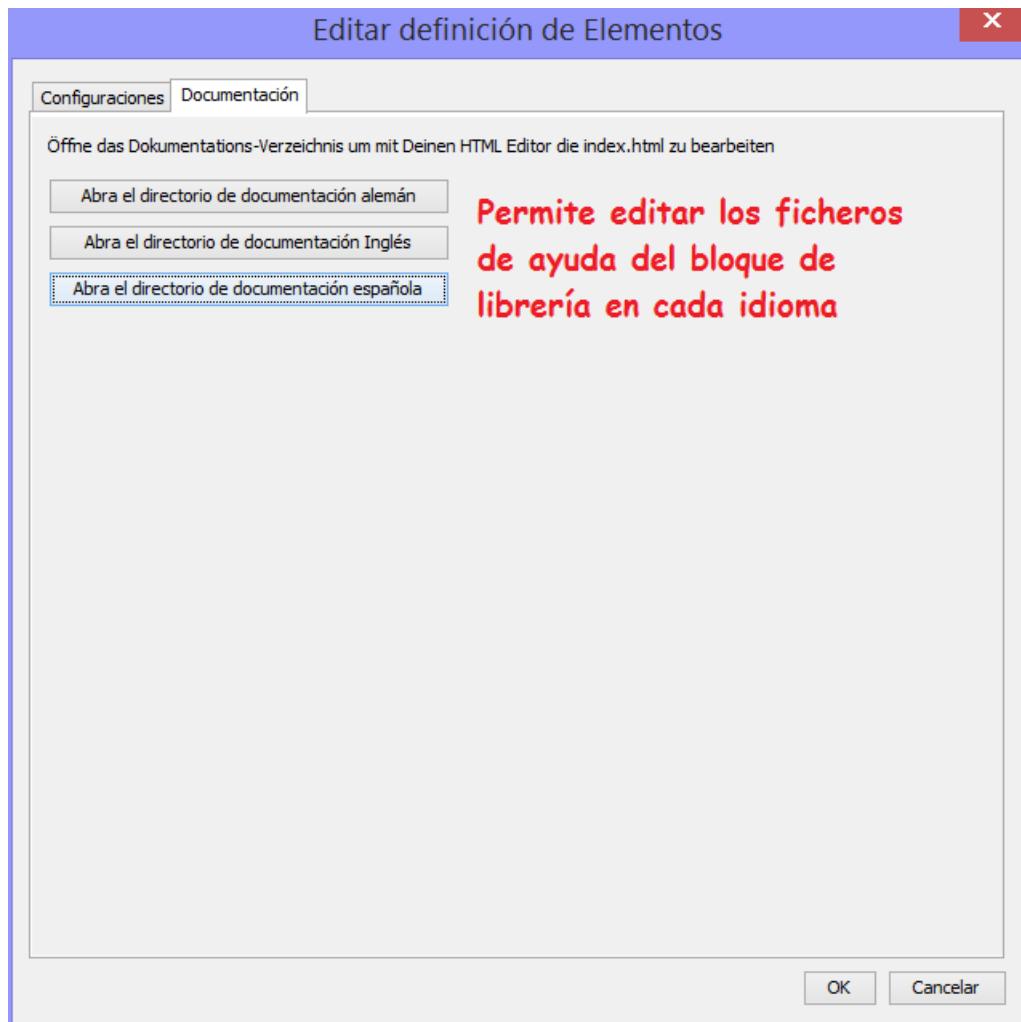
With the program we edit the icon and leave it more or less as shown in the figure.



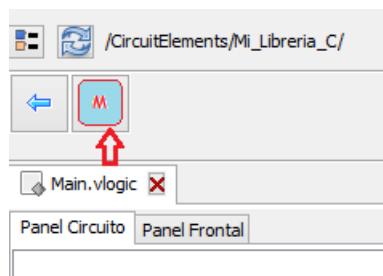
Save the image and click on the "Reload" button of the sale to load it and put it in its place.



Edit the library files of the block if desired



Finally we select "OK" and the information is saved.



The new block will look as shown.

Once the new library block has been created, the next step is to edit it and include in it the elements that we are going to need. We will now explain the procedure for editing a library block that calculates the "average" of two numbers

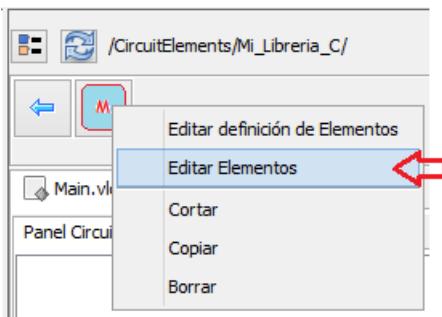
1. Creation of an Element that calculates the average of two numbers.

Now we are going to integrate into the library block its content, which will be none other than the necessary to calculate the average value of two numbers. The numbers will be in the format

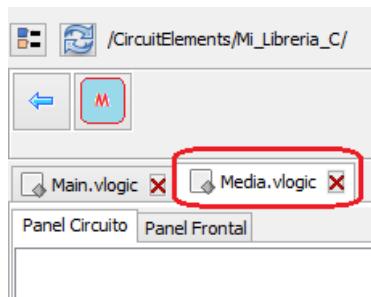
"dbl". It is therefore a purely mathematical calculation library and therefore will not have any "visual" component.

Creation of the corresponding VM

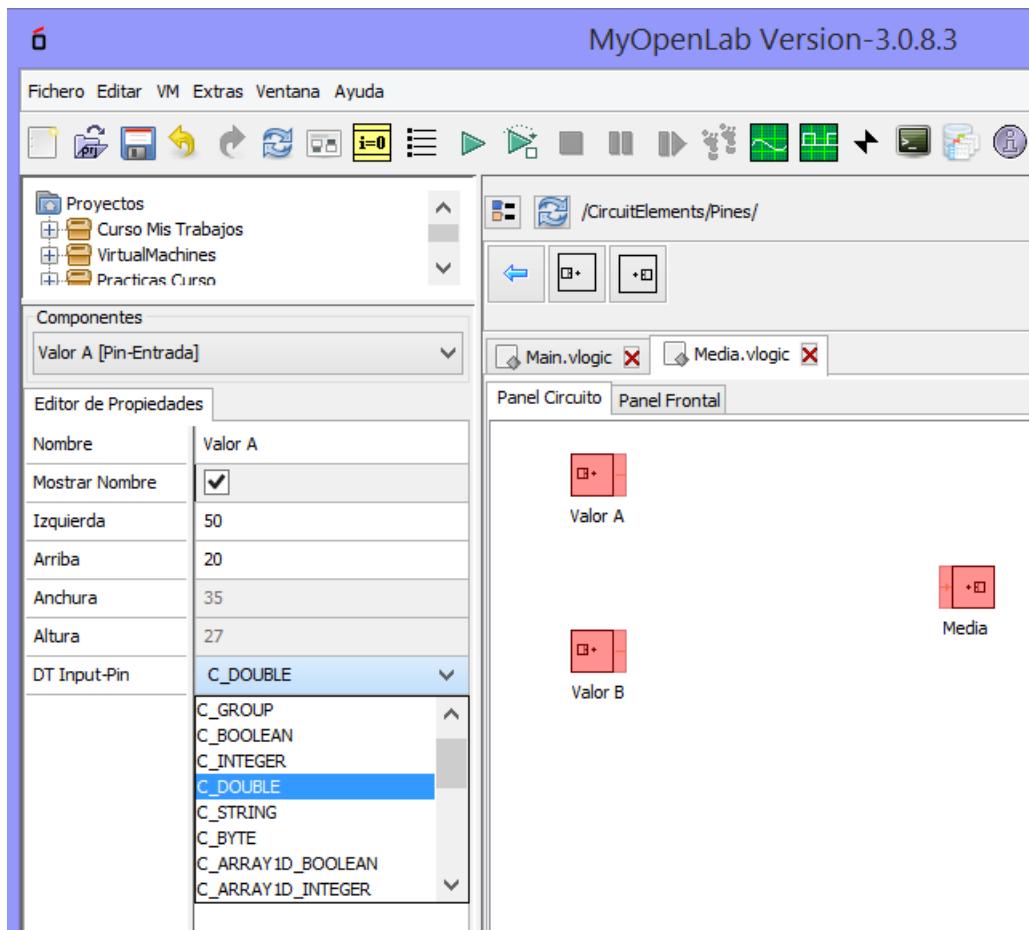
We place ourselves on the component icon and click on the right mouse button and select "Edit Elements".



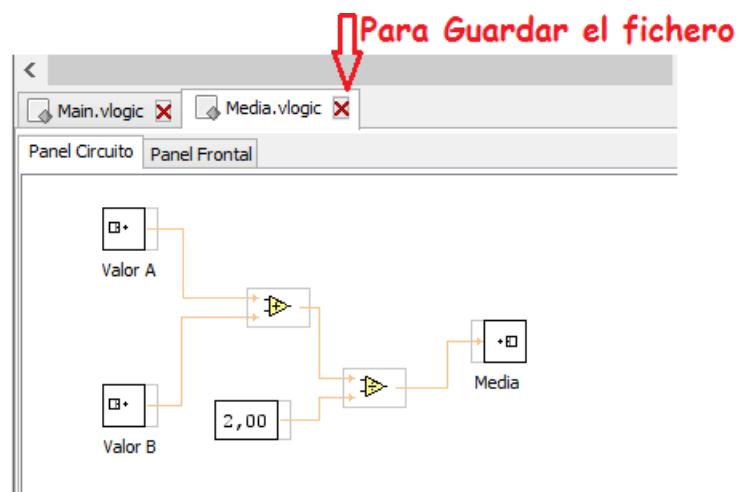
Then a tab with the name of the **Media.vlogic** file will open and we can place the elements that constitute it in its Pan Circuit tab.



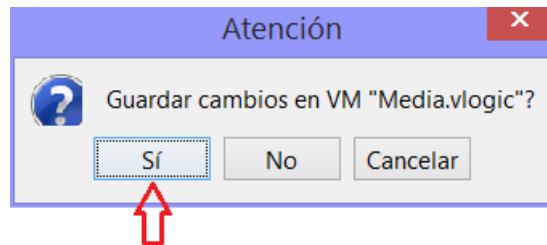
Note that the inputs and outputs will necessarily be connected to "Pins" type elements in which we define the nature of the data (in this case they are all of type "dbl" (by selecting C_DOUBLE in the pin properties menu).



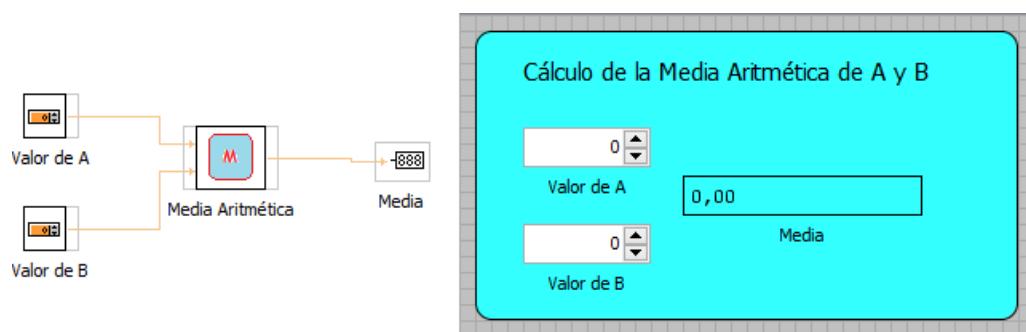
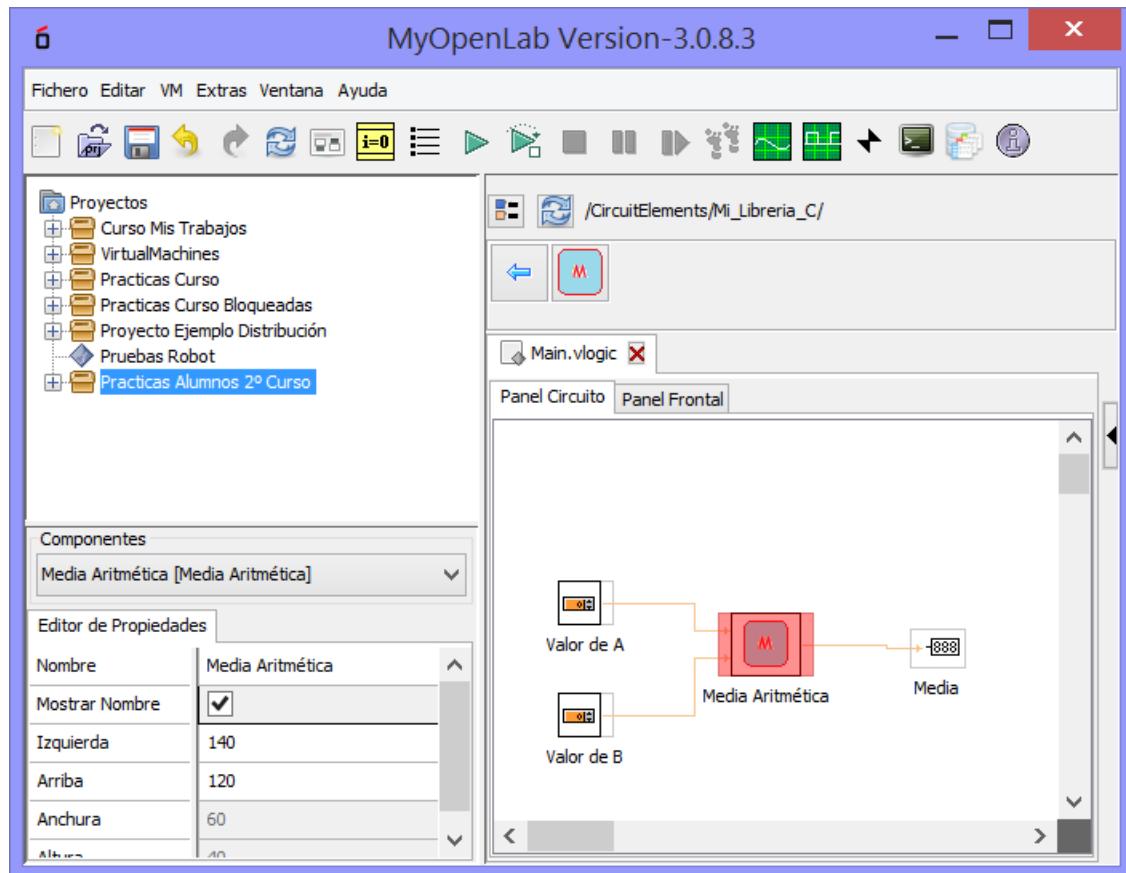
Then we finish the assembly

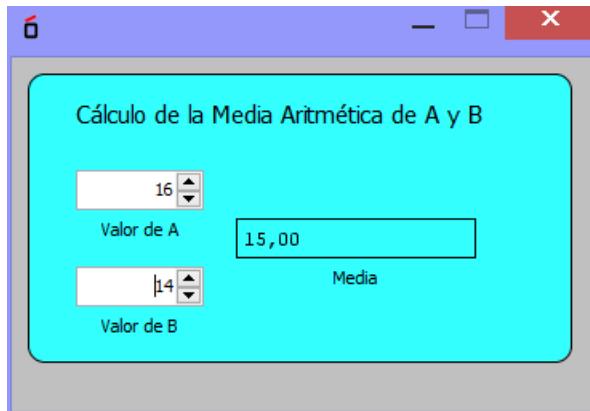
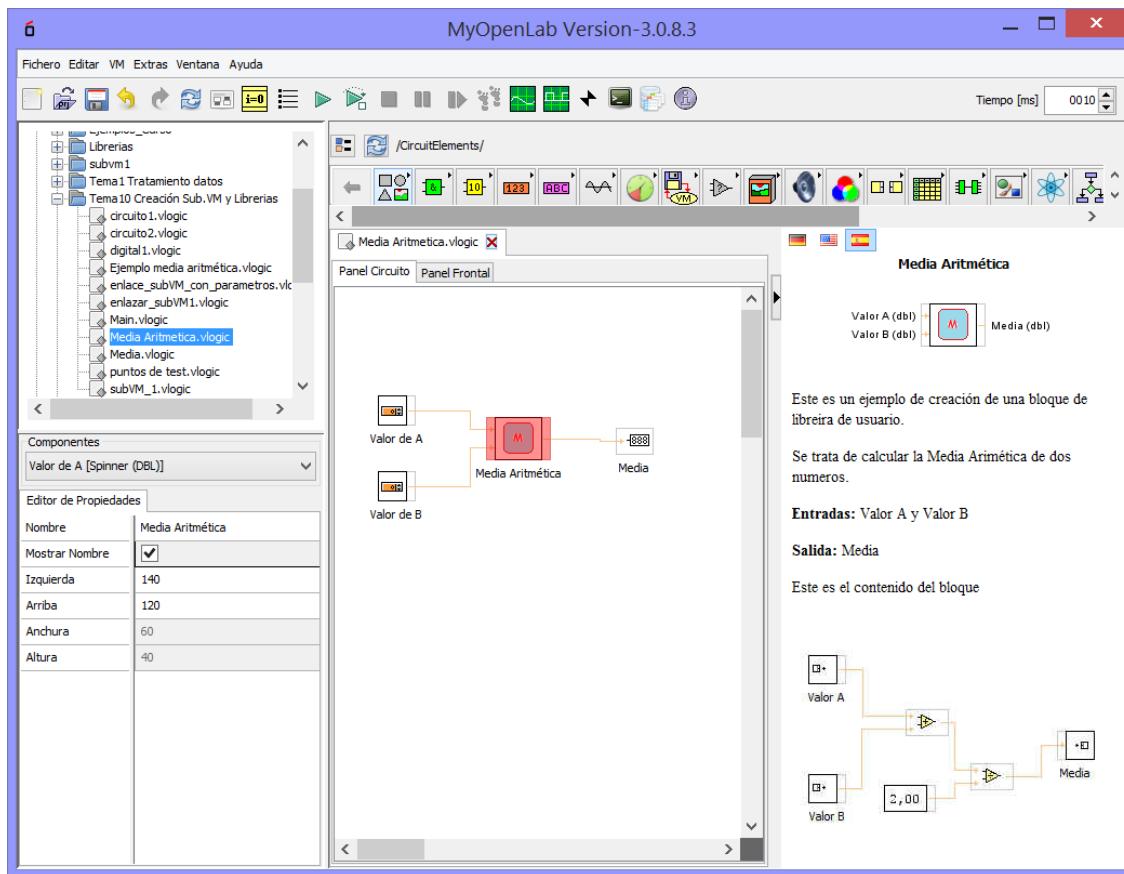


Next, we save the file Media.vlogic, for this we just need to press the
A red "x" appears next to the name and a window appears asking for permission to save.



Incorporation of our new library to a new VM.





The figure shows a simple to use application. If we wanted to change the icon we could do it simply by editing it.

Method 2.

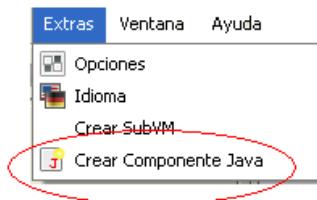
Creation by writing the component code in JAVA.

The component creation method explained below is oriented to those who know the Java language and are capable of creating functions that relate the

input variables and parameters involved in a component. We must start MyOpenlab in edit mode with JDK (*start with JDK.bat*)

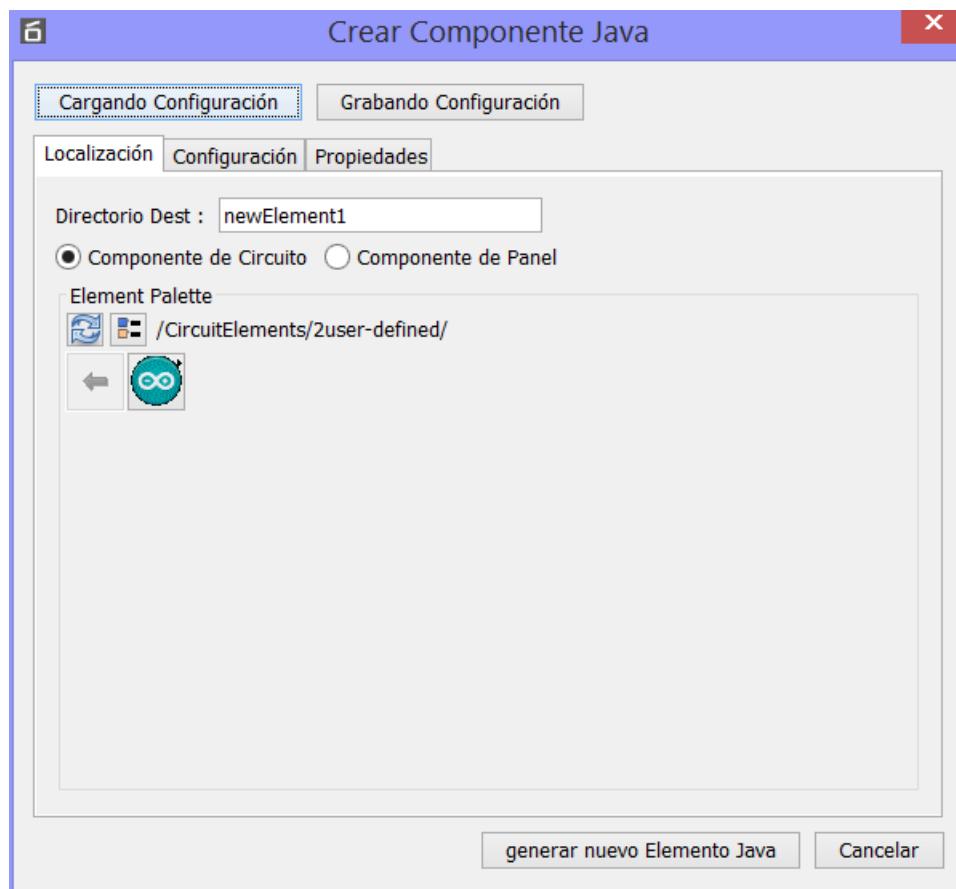
We will follow the steps below:

1. Select from the menu the option Create Java Component

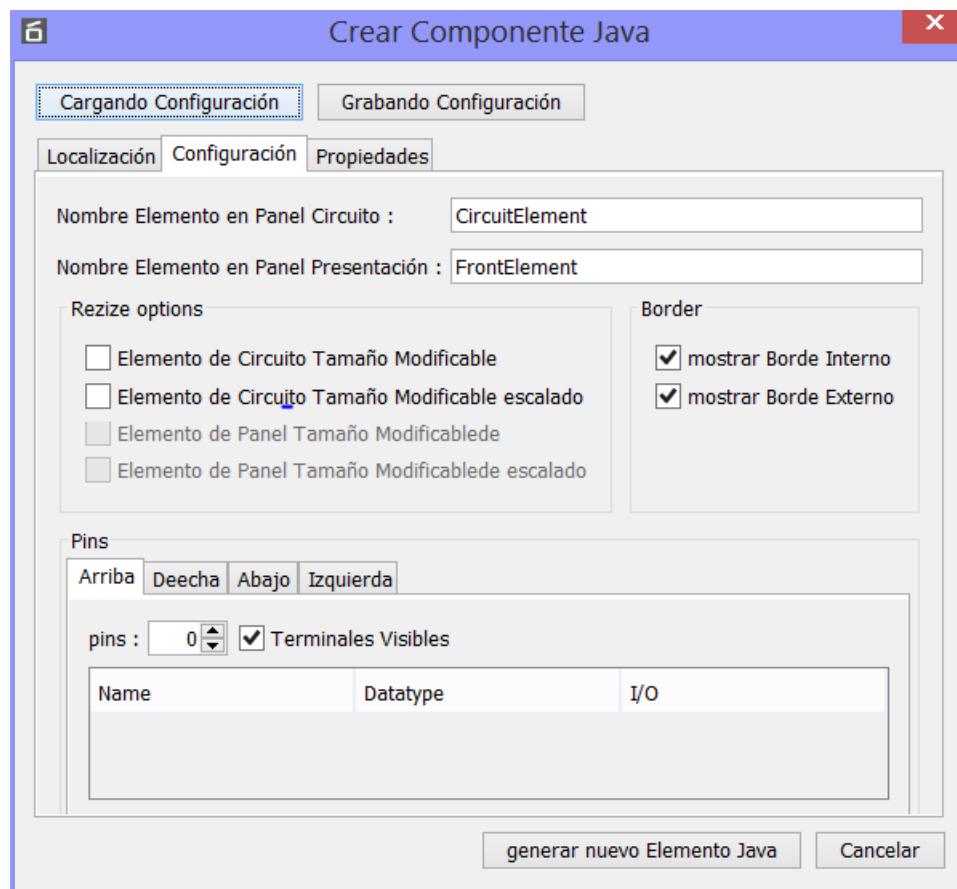


2. Configuration of component parameters

The component parameters and configuration window shown below appears immediately.



From this window we name the Destination Directory of the component in the "Location" tab.



The "Configuration" tab is used to define the parameters that refer to:

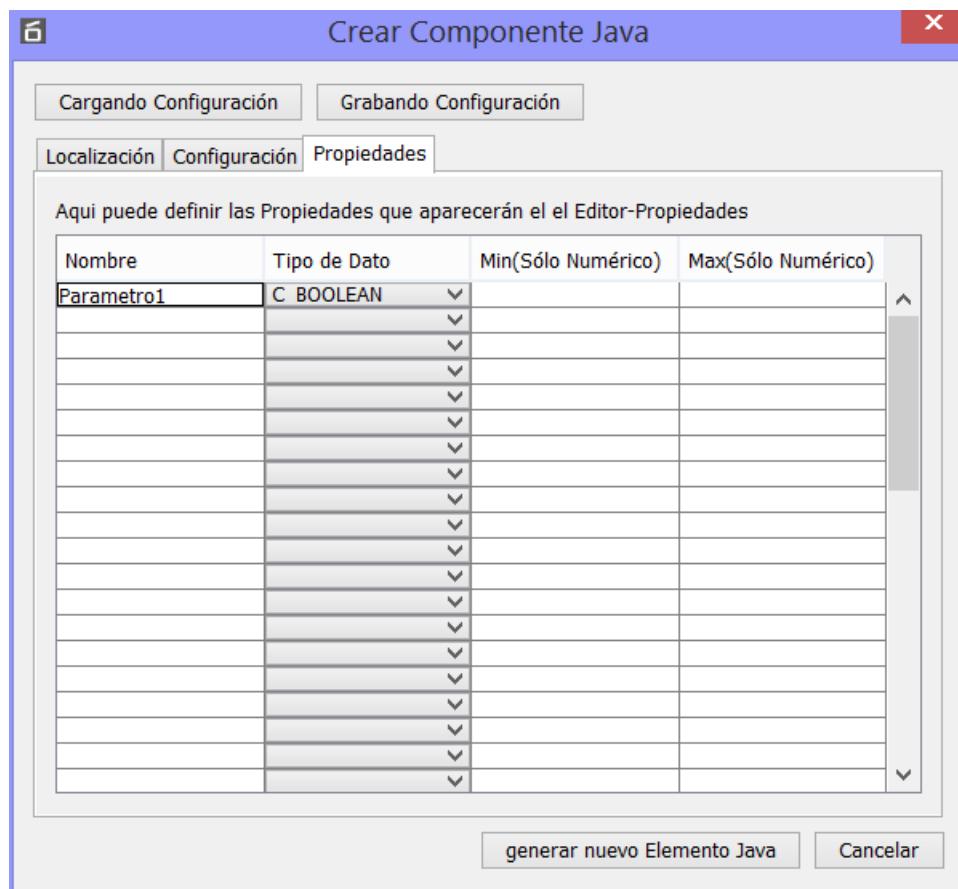
Name: "Name of the component" with which it will appear on the Circuit Panel and on the Front Panel.

Resice Options: Component resizing options when in the panels

Border: Appearance of the component border once placed on the panels.

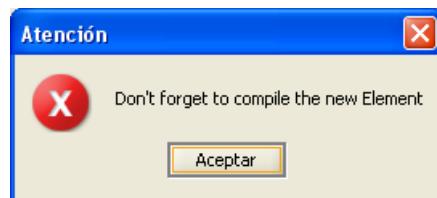
Pins: From here you can configure the pins (inputs/outputs) of the component, which can be on the right. Left, above and below the box with which the component is represented. In the same way the data types of each pin are defined and if they are inputs or outputs (I/O).

In the Properties tab we can designate the Parameters of our component by its "Name", "Data type" and "Min" and "Max" values.



Once we have defined the parameters of the component we can save them in order to be able to use the same model in other future components we want to create: Options "Loading Configuration" and "Saving Configuration".

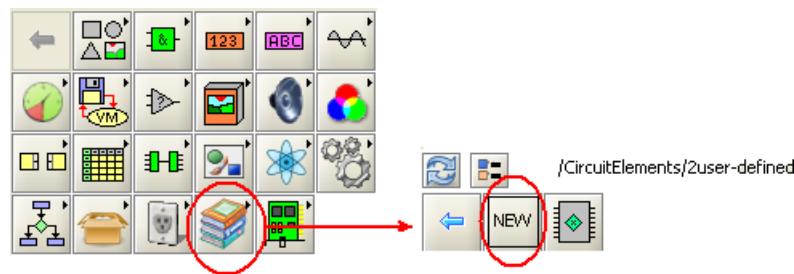
Then we select the option "generate new Java element" and the base model of the component is created, and a warning window appears indicating that we must compile it so that it can be available.



3. Edition and Compilation of the created component.

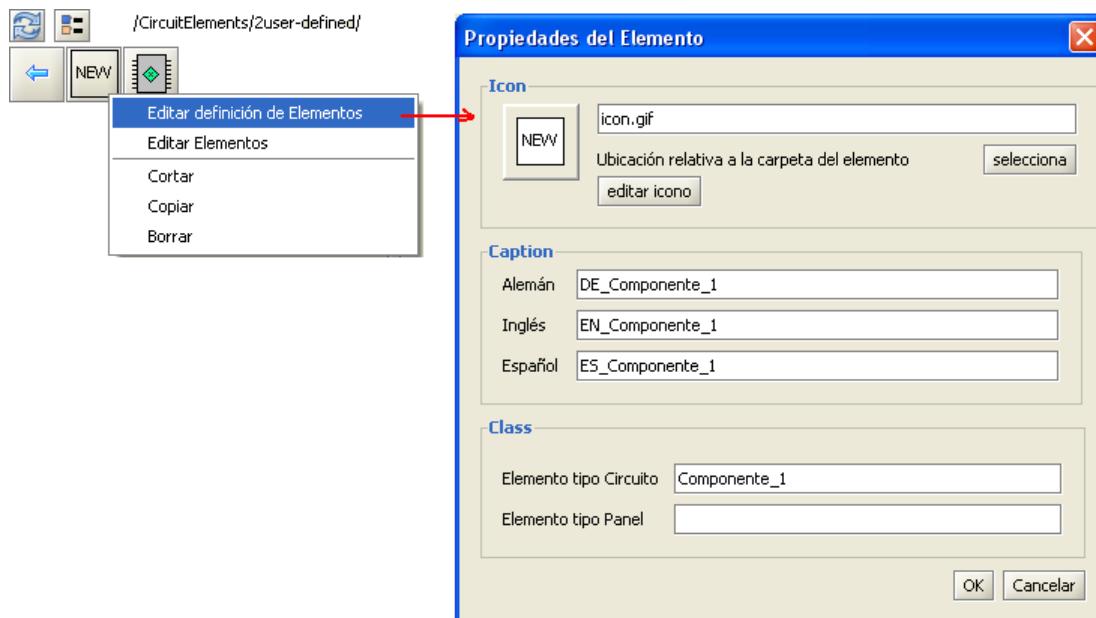
The component we have just defined is not yet available to be used since all we have done is to create the Java program structure of the component. Now what we have to do is to write the Java code according to the function we want the component to perform and then compile it.

The component has been created in the group of user libraries as shown in the following figure

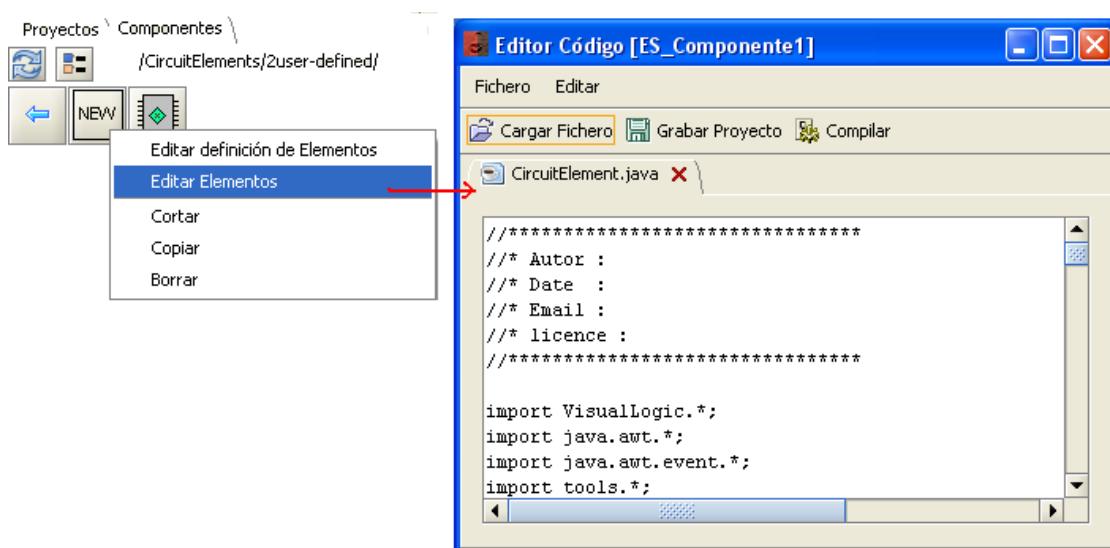


We only need to place ourselves on the component icon and click on the right mouse button and the menu shown in the figure will appear, from which we can select each of the editing options shown.

The following image shows the "Element definition editor" window from which several things can be modified, among them the icon of the created Element. The Caption option allows you to write the text that will appear when we pass the mouse over its icon.



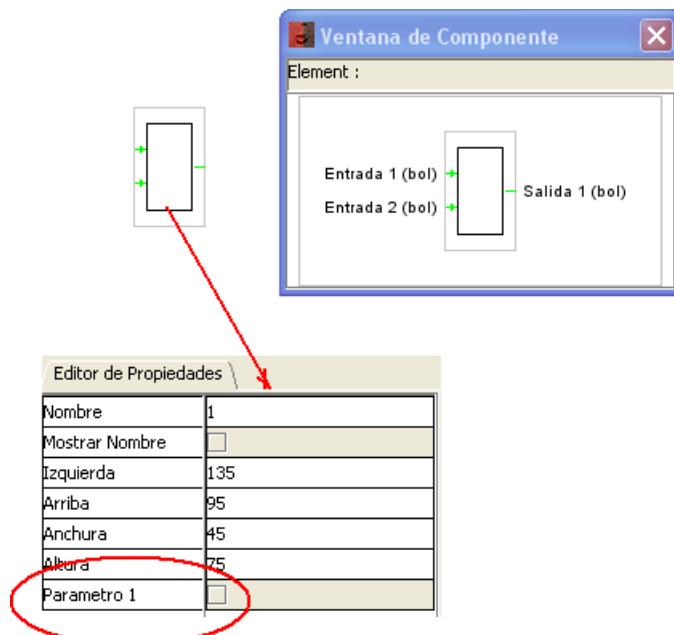
The Edit Elements option displays the java code editing window from which we can write, adding to the basic code already written, the code that allows the realization of our function for this component.



Once the code has been written, we must click on the "Compile" option, which will definitely generate the usable component.

The code of the component can be edited as many times as we want.

In the following figure we can see how the component we have just created will look like, where the identification of its terminals (Component Window) and the Properties Editor appear, where the parameter we defined previously "Parameter 1" appears.



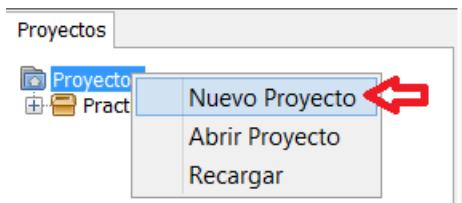
12. Creation of a Distribution

It is possible to create a distribution of a VM application in such a way that it can be run without opening the programming environment, something similar to running in "runtime" mode.

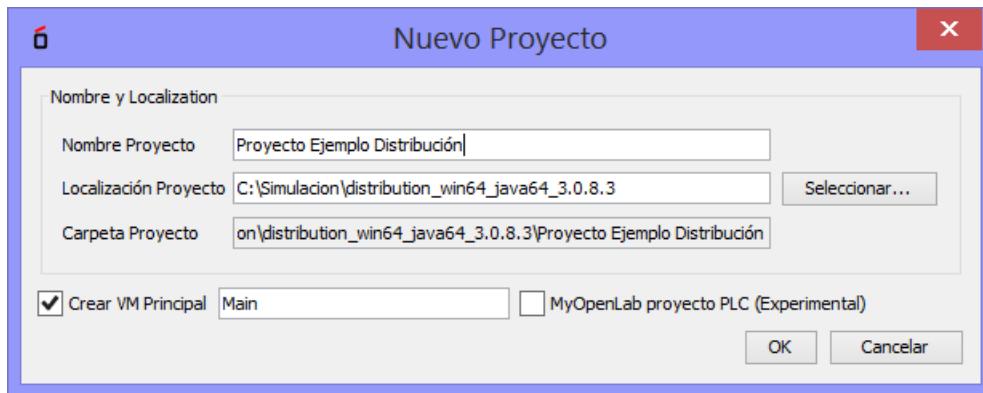
IMPORTANT NOTE: *The created application will be able to run as long as we have MyOpenLab installed on the computer since libraries and applications of the tool are invoked, i.e. it is not an "entirely independent" distribution.*

For the realization of this example we will follow the following steps.

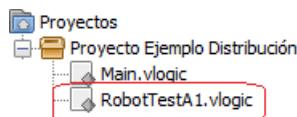
1. A new project is created with the name Test_Distribution



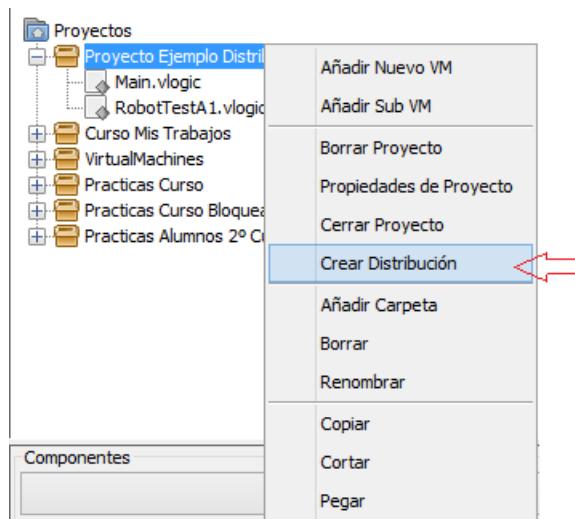
The data in the window that opens will be filled in.



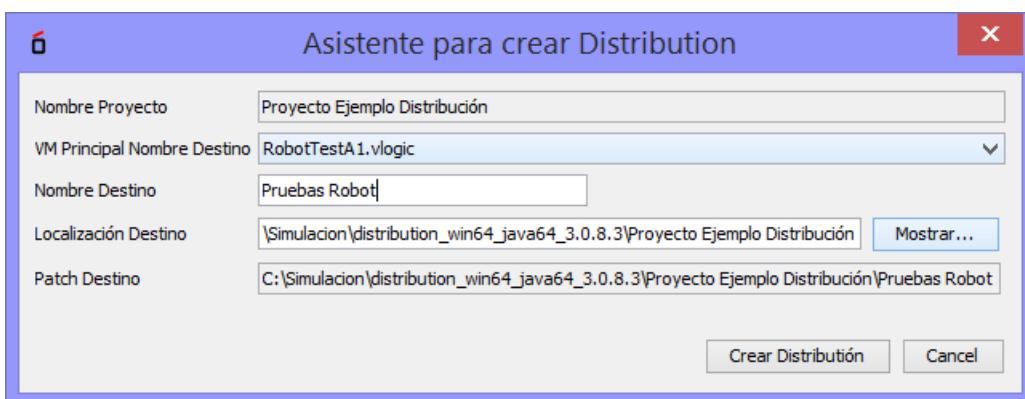
2. Copy the Robot_Test.vlogic file to the new folder we have created.



3. We place ourselves on the folder *Sample Distribution Project* and with the right mouse button we open the contextual menu from which we select Create Distribution.



The following window is displayed in which we can edit the data



It is important that we select as Main VM the file "**RobotTest A1.vlogic**" because the file **Main.vlogic** will be blank since when creating a project folder this file is always created by default, in which we could naturally design an application, since it is a normal VM. In our case it will be blank and will have no role.

Regarding the Destination name, it will be the name that MyOpenLab will give to the main distribution file. In our case it will be "**Robot Tests**".

The destination location will be the folder where the distribution files will be saved, by default it will be in the same folder of the project.

4. Next we will be able to go to the folder where the distribution has been saved and in it we will find a folder called "Robot Tests" inside which are the files of the distribution. For the execution it will be enough to execute the file **start.bat**

Localización del proyecto que hemos creado**..\\distribution_win64_java64_3.0.8.3\\Proyecto Ejemplo Distribución\\Pruebas Robot**

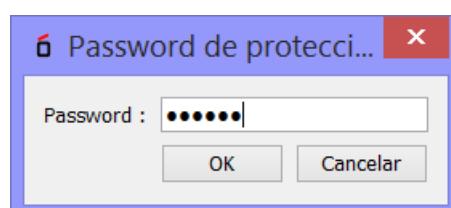
	DistributionStarter	14/09/2016 16:42	Executable Jar File	4 KB
	Main.vlogic	14/09/2016 16:42	Archivo VLOGIC	1 KB
	myopenlab.executable	14/09/2016 16:42	Archivo EXECUTA...	0 KB
	project.myopenlab	14/09/2016 16:42	Archivo MYOPENL...	1 KB
	RobotTestA1.vlogic	14/09/2016 16:42	Archivo VLOGIC	13 KB
	rxtxParallel.dll	14/09/2016 16:42	Extensión de la ap...	83 KB
	rxtxSerial.dll	14/09/2016 16:42	Extensión de la ap...	127 KB
	start ← Ejecuta el Proyecto	14/09/2016 16:42	Archivo por lotes ...	1 KB
	start_linux_distribution	14/09/2016 16:42	Archivo	1 KB

13. Protection and personalization of our work

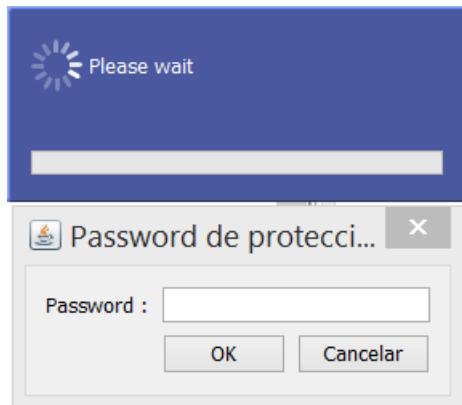
In the new version of Myopenlab there is the possibility of protecting our VM developments by means of a password so that when opening them only the corresponding screen of the Front Panel appears and not the Circuit Panel. In this way, the user who opens the VM application will only be able to perform the simulation but not to modify it.



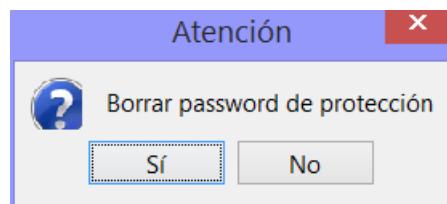
To protect a VM application proceed as follows: We open the VM file and select the "Password Protection" option. A window will appear asking for the password (minimum 6 characters), which you must enter. type and select "OK". The file is now protected, but do not forget to save it when you close or exit the program.



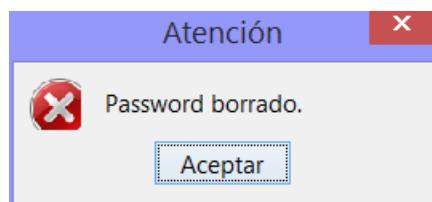
When we try to open a protected file the first thing we will be asked for is its password in the corresponding window. If we write it well we open it complete if we give to "CANCEL" we will open only the Front Panel...



If we want to remove the protection of a file we must select "Delete Password Protection" from the menu Select "YES" in the window and the file will be completely open.



The following message window appears.



14. Plotting and tracking the execution of a VM simulation.

When a model is designed and the simulation is performed, it is sometimes necessary to monitor (trace) the values that certain signals of the model are adopting. To achieve this objective MyOpenlab has the possibility to add test points "Add Testpoint".

How to add a signal test point?

To add a signal test point, simply place yourself on the line from which you want to collect the value of the variable and click the right mouse button. A contextual menu will appear and we select "Add Testpoint", leaving a mark in the indicated position that will indicate that we have a test point there.

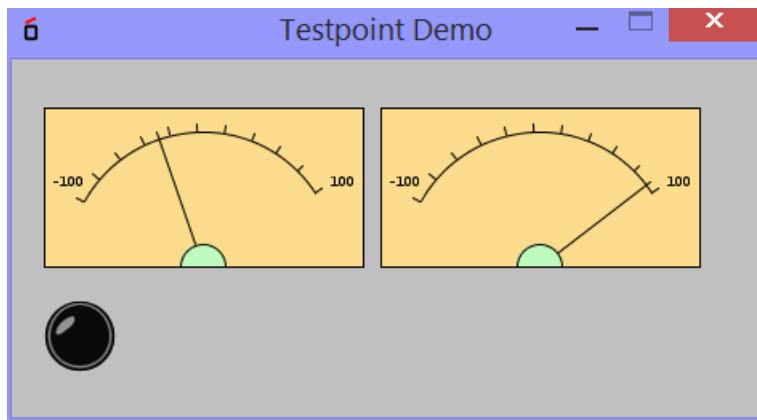
The figure shows the steps to be performed:



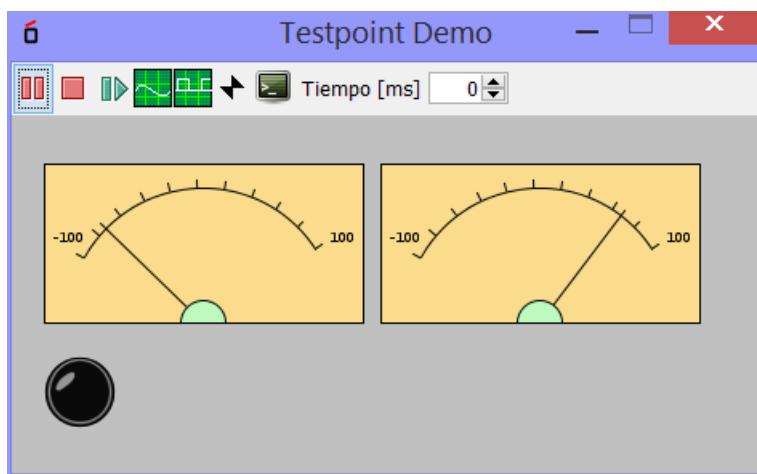
Enable the option to display Testpoint signal graphs in simulation mode.

To be able to see the behavior of the signals we have marked with the Testpoint option in simulation mode, we must enable the "Demonstrar Toolbar" option in the properties menu of the Front Panel.





With the "Show Toolbar" option disabled



With the "Show Toolbar" option disabled

If this option is enabled, the following types of displays will appear on the simulation screen of the Front Panel:



Numerical Graphic Window



Digital Graphic Window



Data Table Window



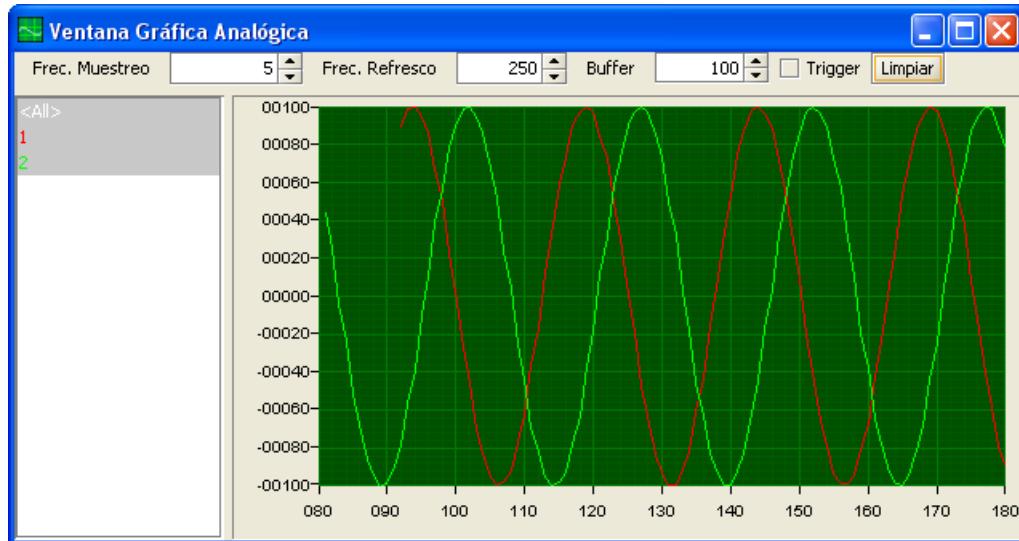
Console

	Stop VM Simulation
	Terminate VM Simulation
	Continue VM Simulation
Tiempo [ms] <input type="button" value="0"/>	Setting the simulation time delay

The following figures show the information windows of the Testpoint points located in a VM simulation.

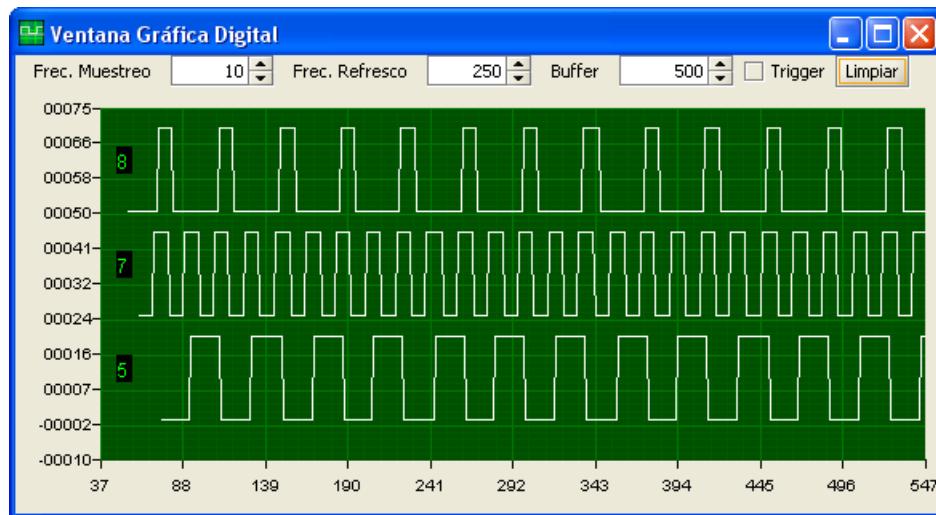
It is also possible to open these windows from the application toolbar.

Numerical Graphic Window



By means of the controls Sampling Freq. Freq. Refresh Buffer Trigger and Clear controls can be used to configure the display of both analog and digital data.

Digital Graphic Window



Data Table Window

Ventana Test de Puntos

Reiniciar

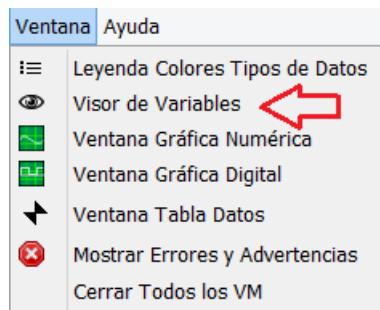
1	2	5	7	8
73.2462941...	-68.080690...	true	true	false
66.0836398...	-68.080690...	true	true	false
66.0836398...	-75.052998...	true	true	false
58.2606996...	-75.052998...	true	true	false
58.2606996...	-81.275401...	true	true	false
49.8556377...	-81.275401...	true	true	false
49.8556377...	-86.685727...	true	true	false
49.8556377...	-86.685727...	false	false	false
40.9524348...	-86.685727...	false	false	false
40.9524348...	-91.229918...	false	false	false

Fila nº = 169

15. Variables viewer

Sometimes it will be interesting to see the value that certain variables take during the simulation of a VM application. We have already described the possibilities of plotting data through the analog, digital and tabular data graphing windows. In the explained form the simulation will be able to run in continuous mode.

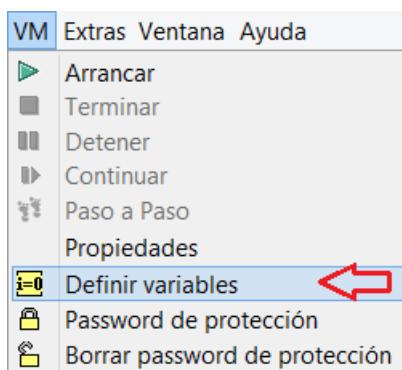
In the case we are going to explain, we will make use of the data display window called "Variables Viewer", which appears in the "Window" pull-down menu, as shown in the following figure.



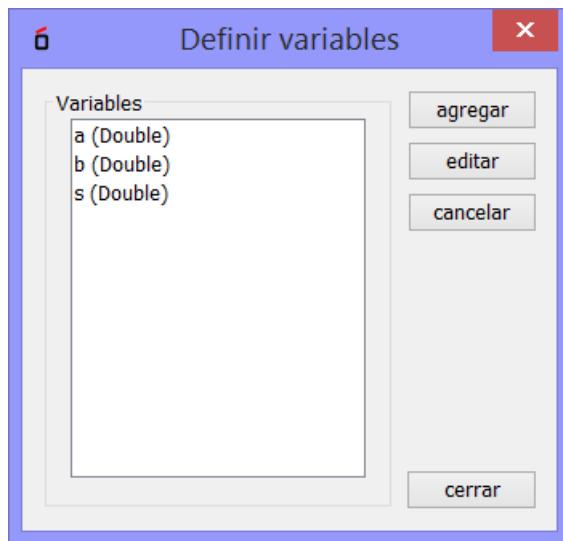
Let us say beforehand that in this case the simulations should be done in "**debug mode**" as shown in the following figure.

When defining variables, it is necessary to do so explicitly (the variables defined by MyOpenlab are not accessible in this mode of work) by using the editor

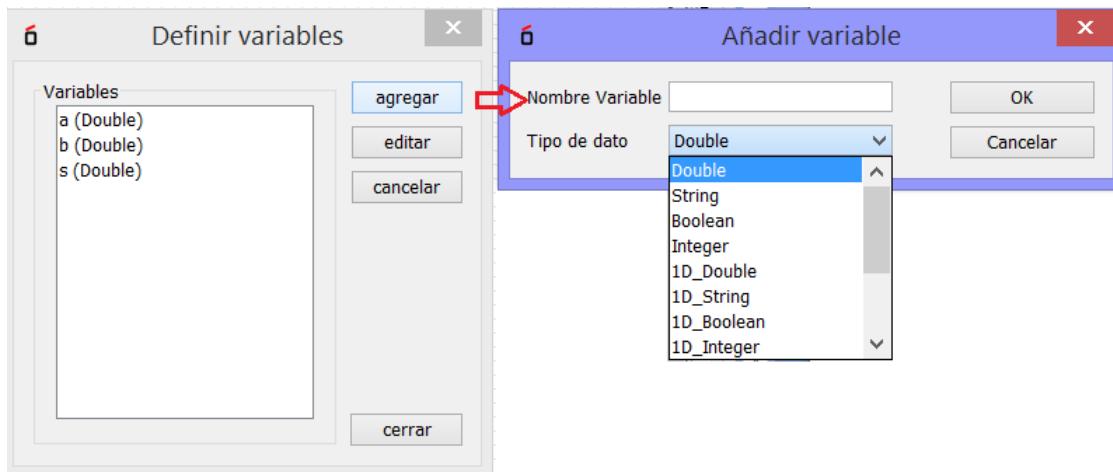
The variable list can be activated by clicking on the button in the button bar of the main window or through the menu:



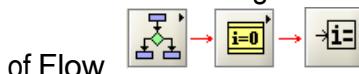
With this tool we define the variables by giving them a name and a data type. Note that we can only define three data types: **Integer, String and Double**.



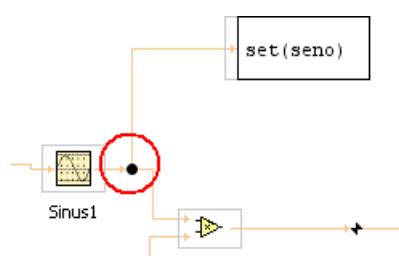
When we select add, the window shown in the following figure appears and in it we define the variable.



Once we have defined the variables we need, we will proceed to extract them from the "Circuit Panel" using the "Get(i)" function found in the Diagrams library.

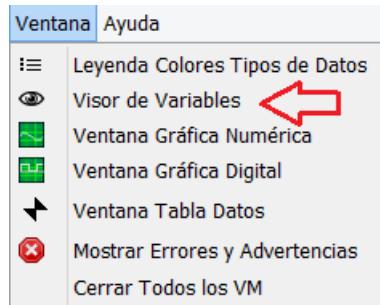


To extract the value of the variable from the corresponding line of the VM schematic, simply "add node" (right-click on the line where you want to include the node in the context menu function).



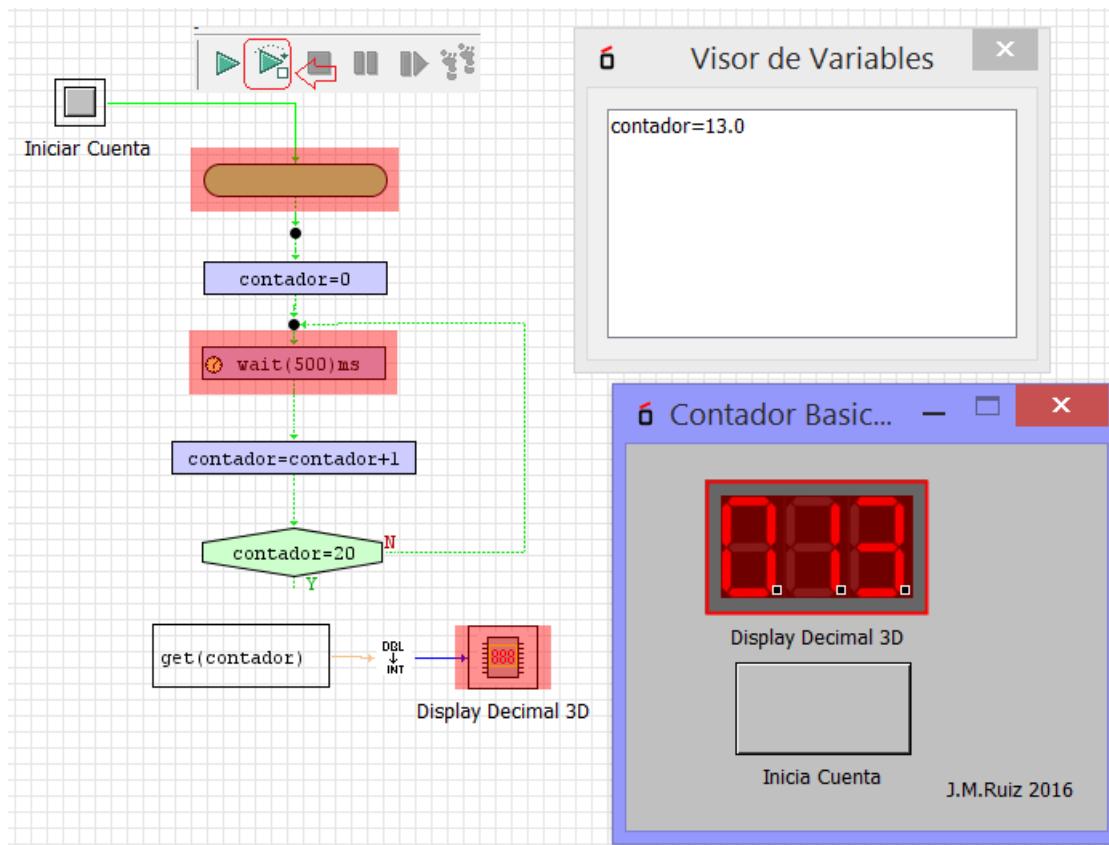
Once we have defined the variables we want to see, we just need to start the VM simulation always in "Debug Mode".

We open the "Variables Viewer" window and we can see a list of all the variables defined and their values evolving according to the stimuli received by the model.



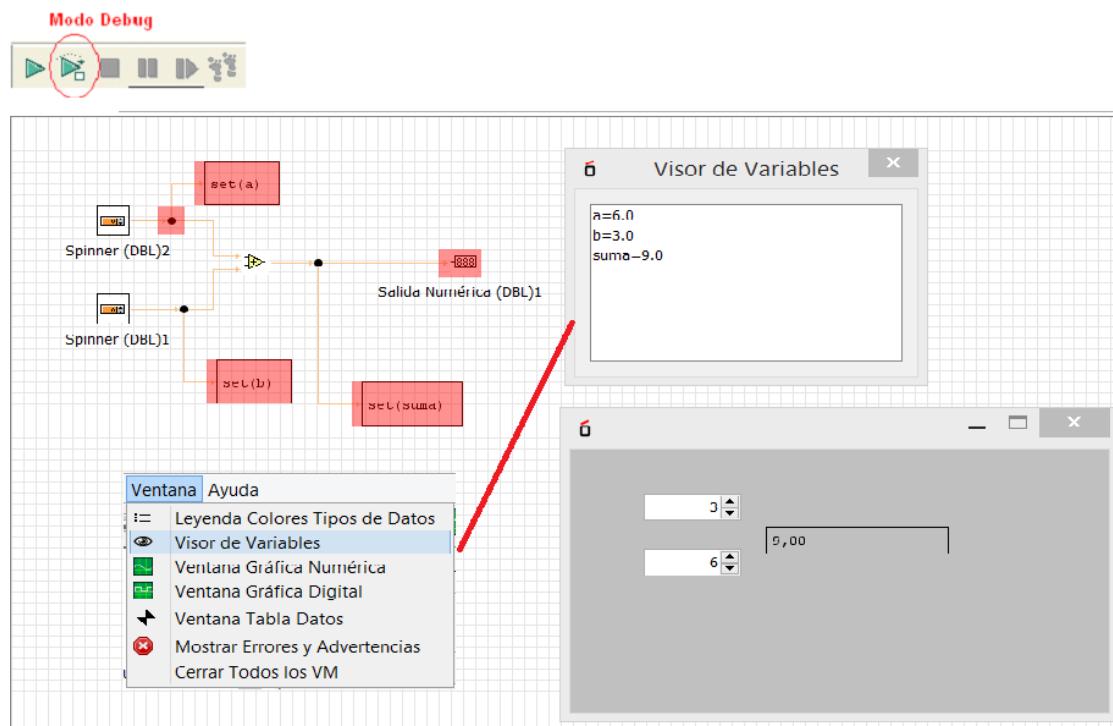
In the following figure we see the evolution of a simple example of a Flowchart application that implements a counter from 0 to 10.

The variable in this example is "counter".



In the following example we see the simulation of an addition operation in which we collect the values of the input variables "a" and "b" and the output variable "sum".

The "Variables Viewer" panel next to the simulation screen is shown in the figure.

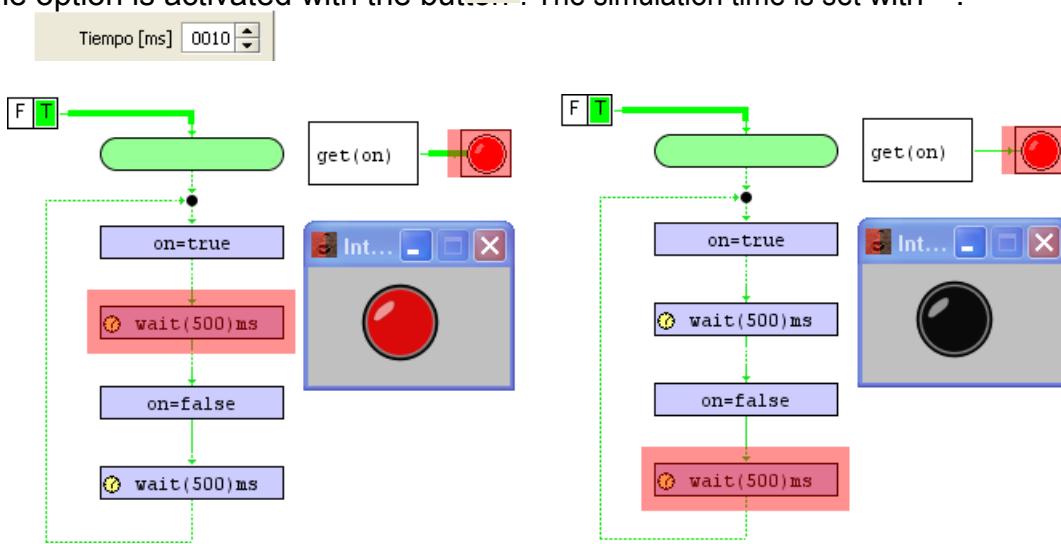


16. Execution in debug mode

In some applications, at the time of simulation, we will be interested in running the simulation in "debug mode". This means that we will be able to see graphically how the input and output elements are activated and how the data flow is explored.

To understand this form of simulation it is best to use a copy or in which there is a flowchart because you can clearly see how the various blocks that constitute it are activated in sequence....

The option is activated with the button . The simulation time is set with .

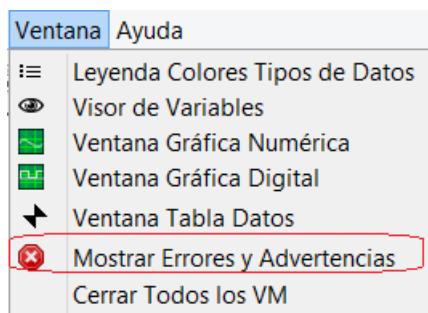


17. Errors and Warnings" window.

This window shows possible errors or warnings during the simulation or in the case of invalid operations.



It can be accessed through the "window" menu, as shown in the figure below.



18. Console Output" window.

This window allows you to view the data sent to the console.



The commands that send data to the console are "Println" and "Print line on console":

The screenshot shows the MyOpenLab software interface. On the left, there is a workspace titled "Main.vlogic" with two tabs: "Panel Circuito" and "Panel Frontal". In the workspace, there is a red rectangular block labeled "println(i)". To the right of the workspace is a detailed description of the "Println" element:

Println

(Flow)

Elemento Imprime

Este elemento imprime el valor de una variable en la ventana se salida

Salida

```
5.0
Hello World!!!
3.0
20.0
```

Uso: : Println(<expression>)

Ejemplo :

```
Println("Hello World!")
Println(a) // a = double
```

The screenshot shows the MyOpenLab software interface. On the left, there is a workspace with a flowchart. It starts with a green decision block (IF), followed by a green connector block, and ends with a purple "println" block containing the text "Hola Mundo". To the right of the workspace is a "Salida Consola" window:

Salida Consola

```
Hola Mundo
```

The appearance of the console window is shown in the figure below, which corresponds to a simple example.

The following is an example with the command "Print line on console".



19. Identification of the author of a VM application



When creating an application with MyOpenlab, it is possible to indicate the author's name, e-mail address and web page address. This is done through the VM -> "Properties" option in the Menu. The following figure shows this option and the properties window that appears when it is activated.

Very important note:

Important syntax modifications for the new Flowchart library

Formerly	Now
=	==
Value type dbl 10	Value dbl rate 10.00
sin(x)/cos(x)/...	Math.sin(x)/Math.cos(x)/..
AND & operator	AND operator &&
Operator OR	Operator OR