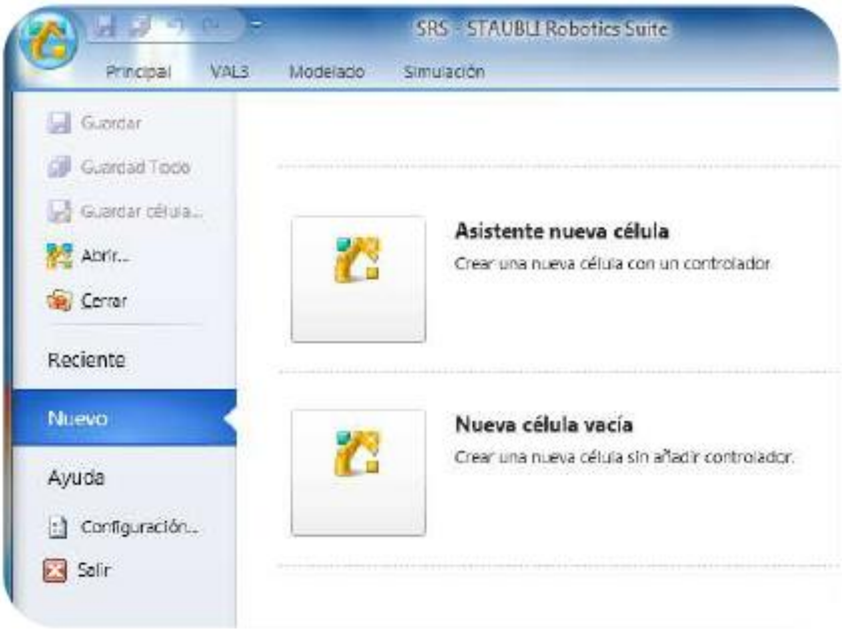
Creating a new cell, choosing a controller and a robot

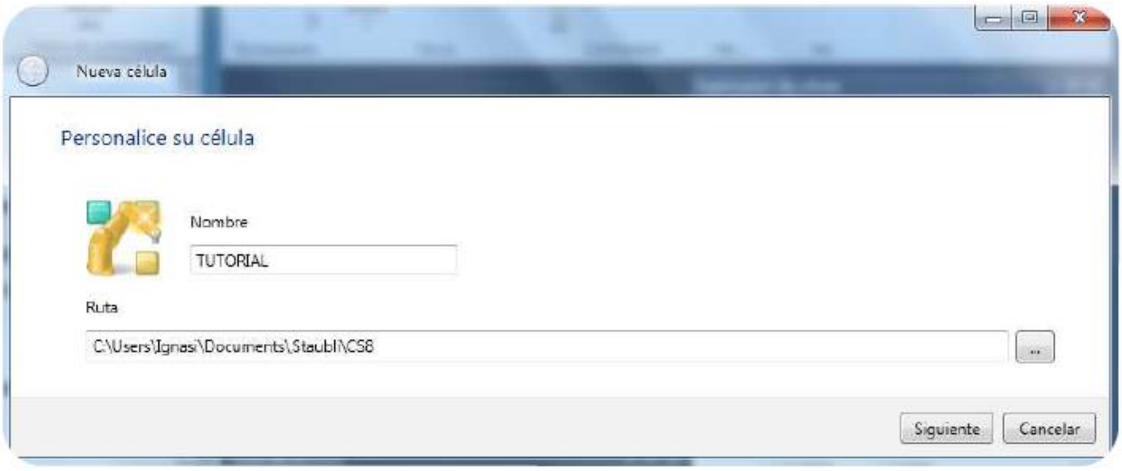
We are going to start with the process of creation of a new cell. First click over “Nuevo” (New) and then “Asistente nueva célula” (New cell wizard)



Asistente nueva célula
Crear una nueva célula con un controlador

Nueva célula vacía
Crear una nueva célula sin añadir controlador.

Then, a new window will appear that will allows us to edit the name of the Cell and the path. We recommend using the path by default and editing the name of the Cell (TUTORIAL, in this specific case)

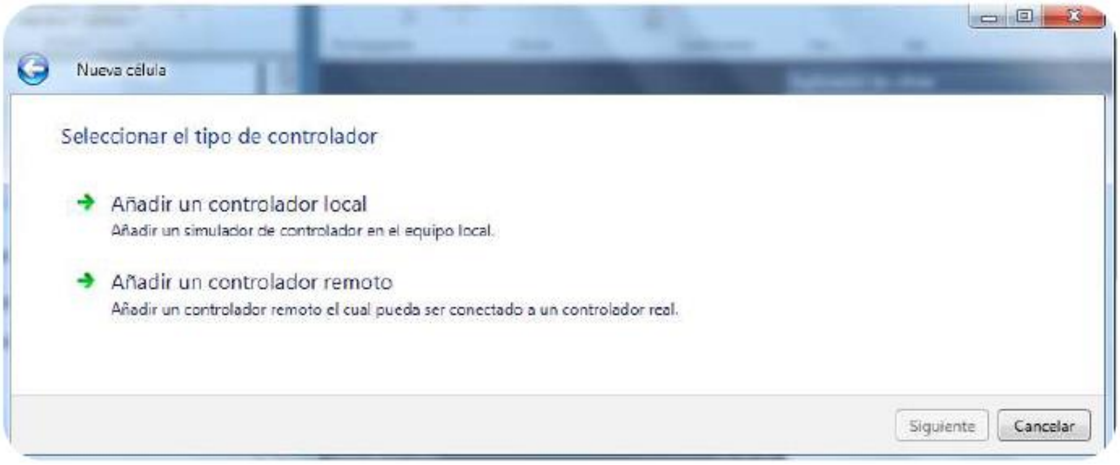


Personalice su célula

Nombre: TUTORIAL

Ruta: C:\Users\Ignasi\Documents\Staubli\CS8

Then we have to select the controller. We are going to add a local controller, because we are going to do a simulation from the computer: “Añadir un controlador local”.



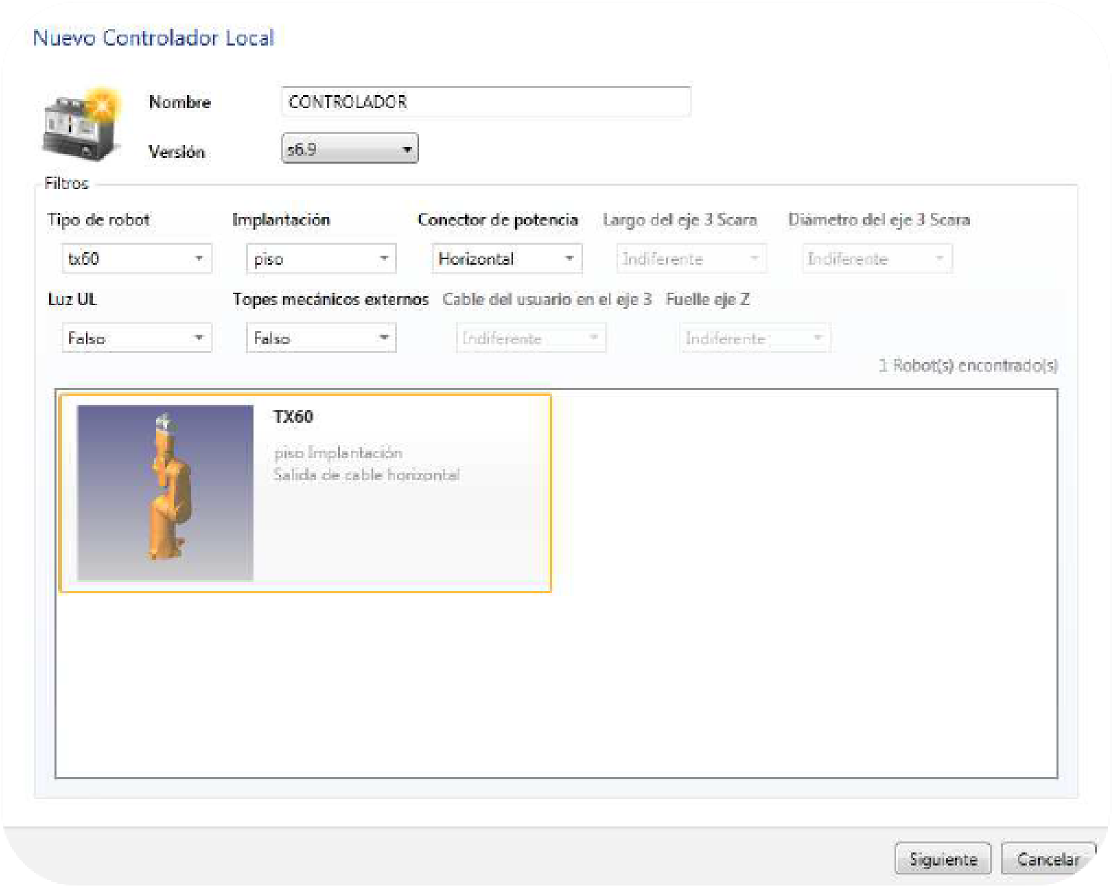
Nueva célula

Seleccionar el tipo de controlador

➔ Añadir un controlador local
Añadir un simulador de controlador en el equipo local.

➔ Añadir un controlador remoto
Añadir un controlador remoto el cual pueda ser conectado a un controlador real.

Once selected, a new window will appear that allows us to select the version of the controller, type of robot, and other parameters we need to select. We are going to work with the two robots we have in our lab. Staubli TX60 (controller version 5.3.2) or Staubli Scara TS60 (controller version 6.8 or 6.9) and with the following parameters: L200 (length of axis 3) and D25 (diameter of axis 3).
In our case, the name of the controller is CONTROLADOR.



Nuevo Controlador Local

Nombre: CONTROLADOR
Versión: s6.9

Filtros

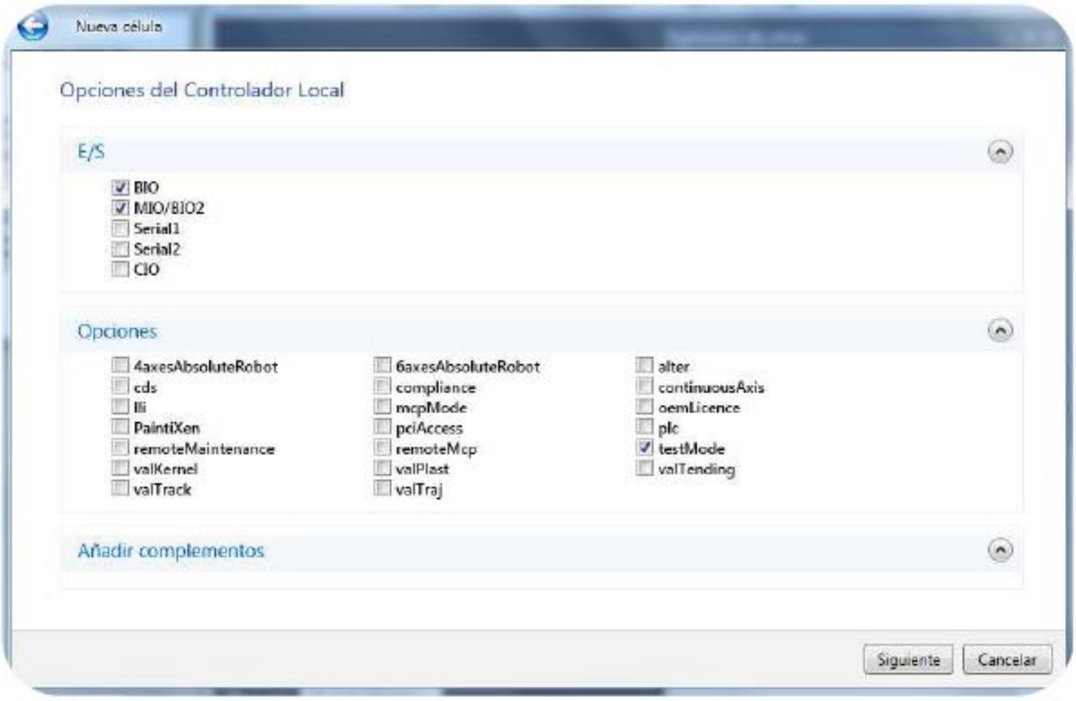
Tipo de robot	Implantación	Conector de potencia	Largo del eje 3 Scara	Diametro del eje 3 Scara
tx60	piso	Horizontal	Indiferente	Indiferente

Luz UL	Topes mecánicos externos	Cable del usuario en el eje 3	Fuelle eje Z
Falso	Falso	Indiferente	Indiferente

1 Robot(s) encontrado(s)

TX60
piso Implantación
Salida de cable horizontal

Then, we have to delect other kind of options for our controller. Please, follow the pictures!!



Opciones del Controlador Local

E/S

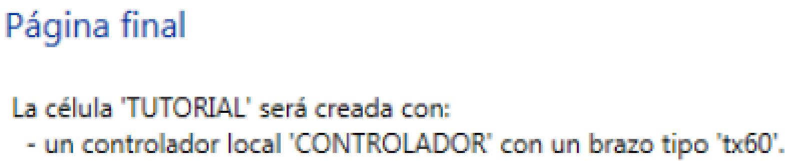
☑ BIO
☑ MIO/BIO2
□ Serial1
□ Serial2
□ CIO

Opciones

□ 4axesAbsoluteRobot
□ cds
□ lli
□ PaintiXen
□ remoteMaintenance
□ valKernel
□ valTrack
□ 6axesAbsoluteRobot
□ compliance
□ mcpMode
□ pciAccess
□ remoteMcp
□ valPlast
□ valTraj
□ alter
□ continuousAxis
□ oemLicence
□ plc
☑ testMode
□ valTending

Añadir complementos

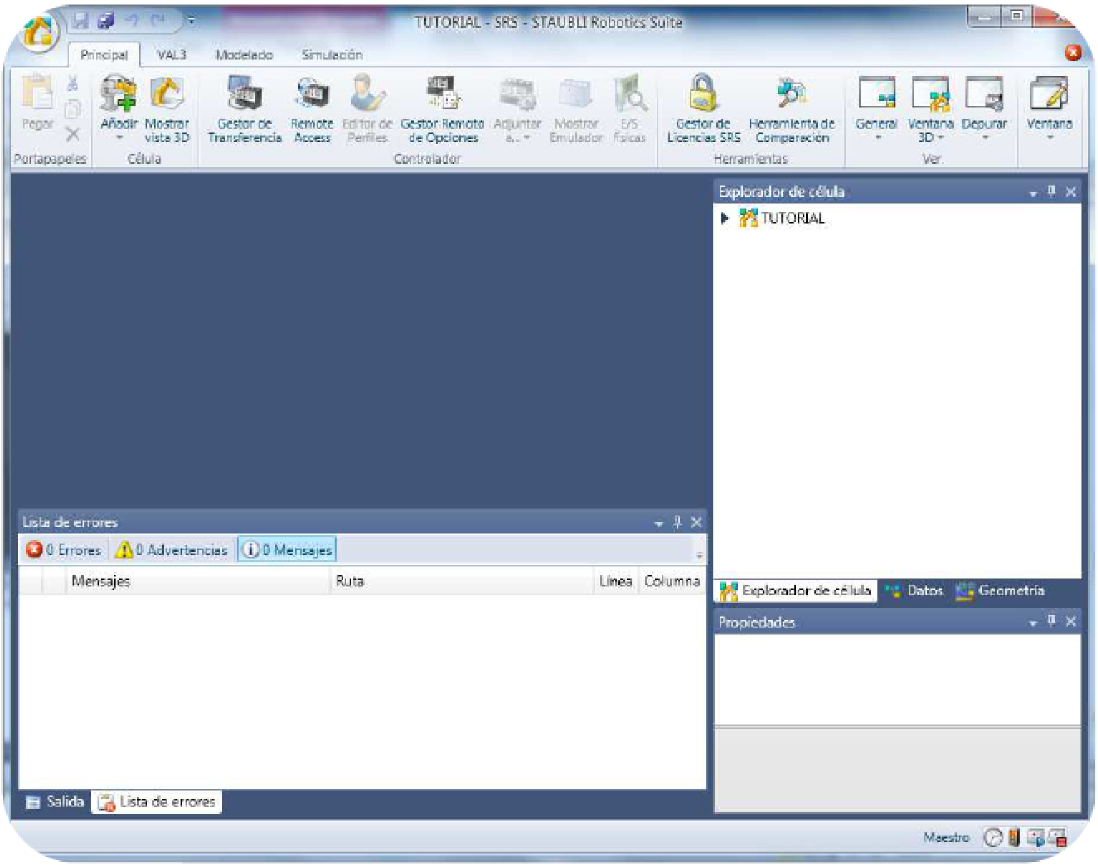
At the end, a new window will appear “Página final” with the results of all of this process.



Página final

La célula 'TUTORIAL' será creada con:
- un controlador local 'CONTROLADOR' con un brazo tipo 'tx60'.

When we click “Finalizar” (Finish) the Cell will be created. SRS software shows us automatically the window with different folders with tools. The Cell is already created, and in the window “Explorador de Célula” (Cell Explorer) we can see the controller and the robot. Now is time to start working with this new Cell. We can see a new Cell called TUTORIAL. We can unfold the Cell and we can edit the “Explorador de Célula” (Cell Explorer), the “Datos” (Data) and the “Geometría” (Geometry).
We have to be familiar with these things related to the robot cell.

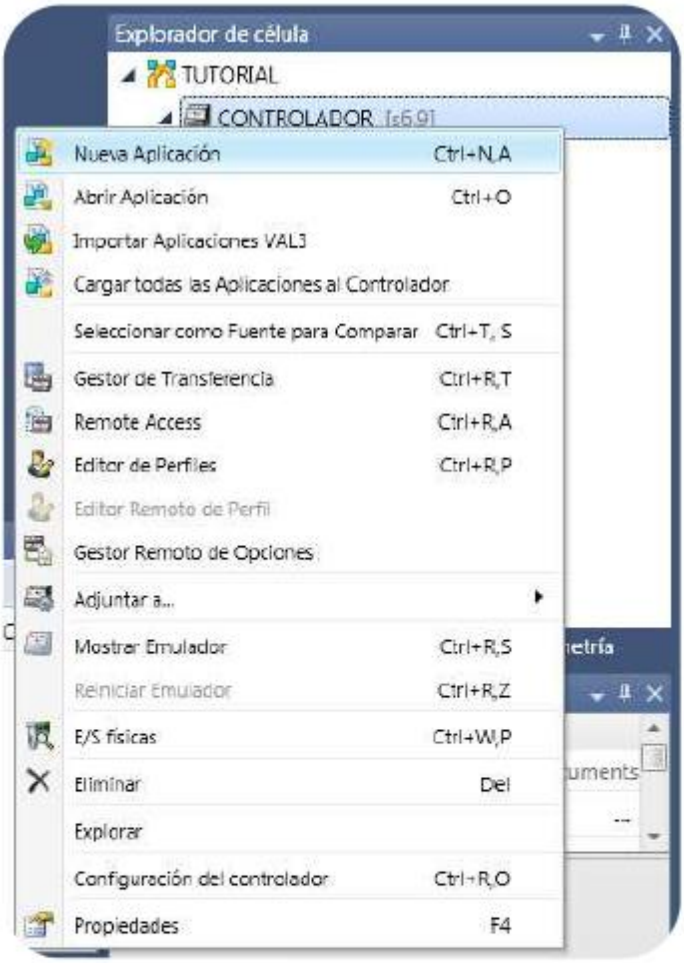


Creation of VAL3 Applications

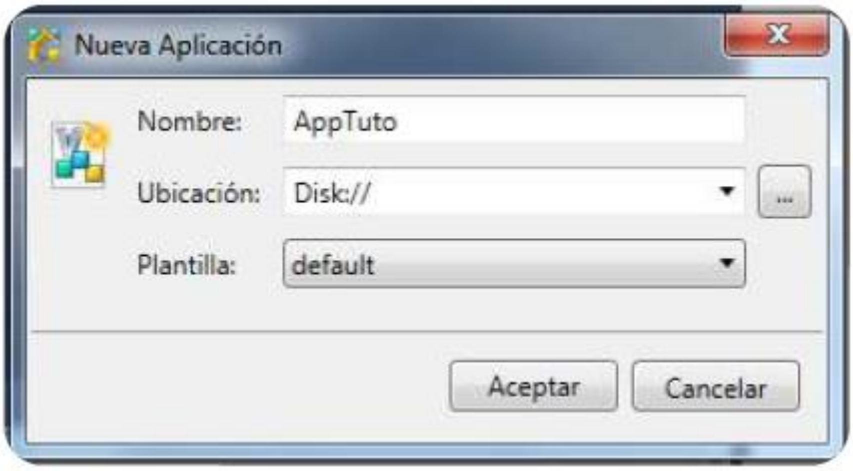
To create a VAL3 application a Cell is required with a controller and a robot associated to it. We have the conditions to proceed because we have all we need.

Assigning a new application to a controller

We are going to create an application for a robot Staubli TX60 associated with the controller CONTROLADOR. Double-click over the name of controller (CONTROLADOR), and a new menu will appear. You must choose “Nueva Aplicacion” (New Application)


Explorador de célula
TUTORIAL
CONTROLADOR [s6.9]
Nueva Aplicación Ctrl+N,A
Abrir Aplicación Ctrl+O
Importar Aplicaciones VAL3
Cargar todas las Aplicaciones al Controlador
Seleccionar como Fuente para Comparar Ctrl+T, S
Gestor de Transferencia Ctrl+R,T
Remote Access Ctrl+R,A
Editor de Perfiles Ctrl+R,P
Editor Remoto de Perfil
Gestor Remoto de Opciones
Adjuntar a...
Mostrar Emulador Ctrl+R,S
Reiniciar Emulador Ctrl+R,Z
E/S físicas Ctrl+W,P
Eliminar Del
Explorar
Configuración del controlador Ctrl+R,O
Propiedades F4
etría
uments

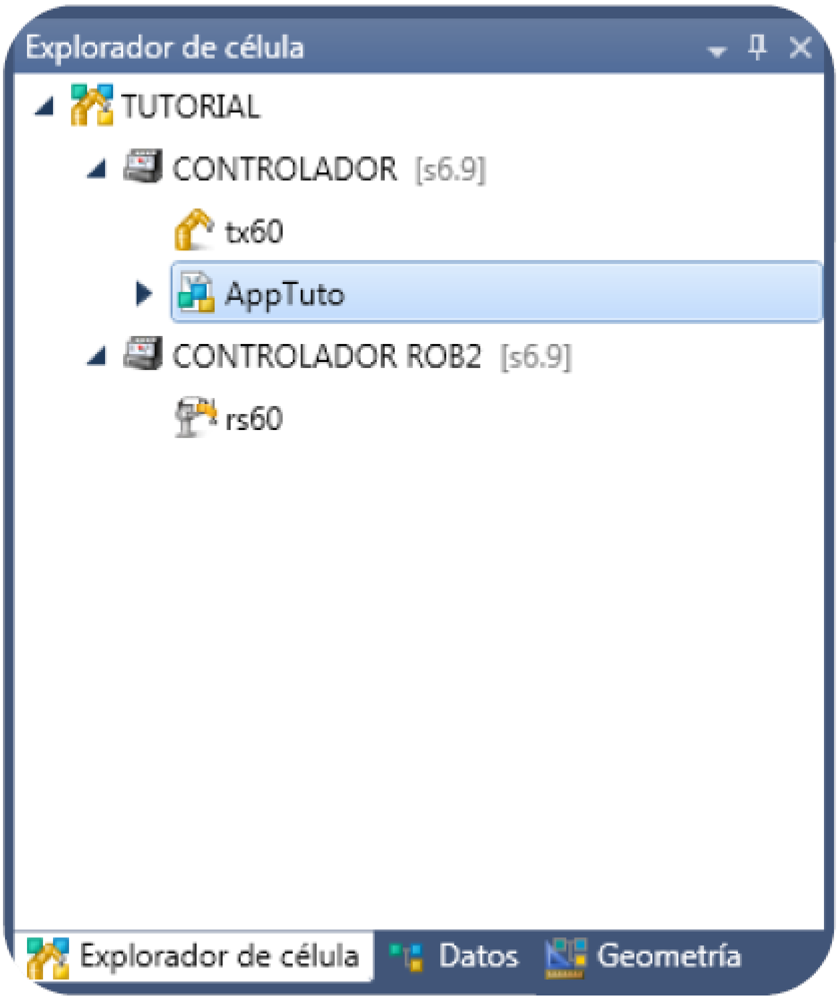
An emergent window will appear allowing us edit the “Nombre” (name) of the application, the “Ubicación” (path), and a “Plantilla” (template) (we will not use it). We will only name the application: “AppTuto”


Nueva Aplicación

Nombre: AppTuto

Ubicación: Disk://

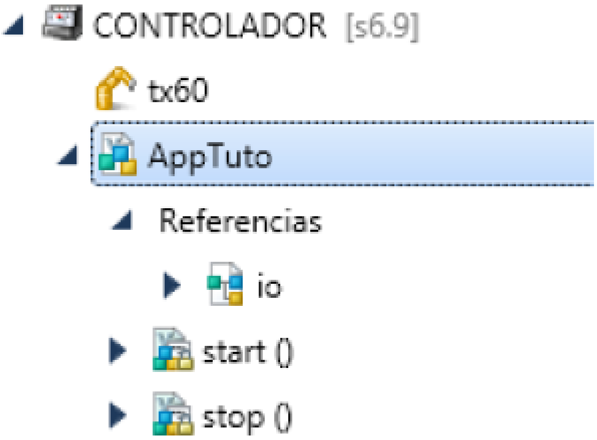
Plantilla: default



Contents of the new application in the “Explorador de Célula” (Cell Explorer)

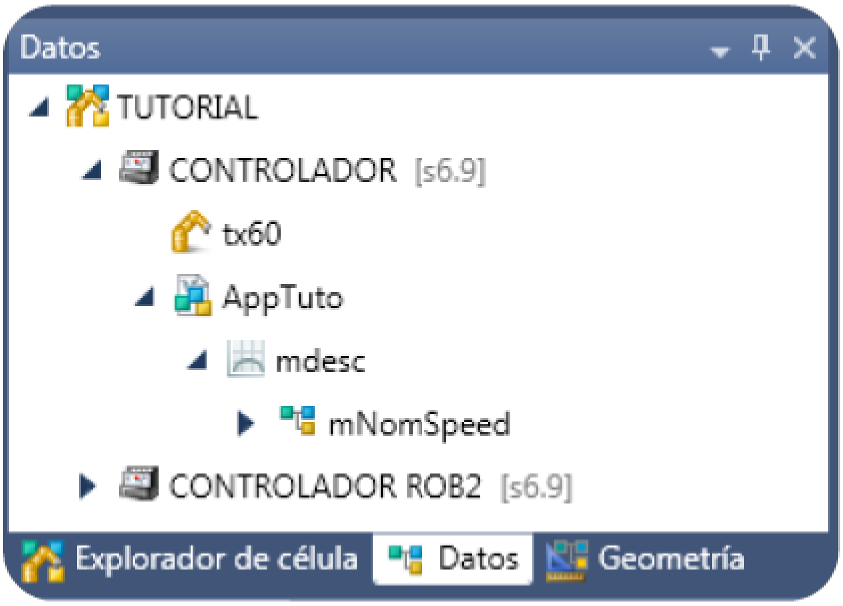
New application has two routines called “start” and “stop”. “start” is the routine that executes automatically when an application is started. “stop” is the routine that it is executed when the application finish.

An unfolded list of available inputs and outputs (io) also appears.



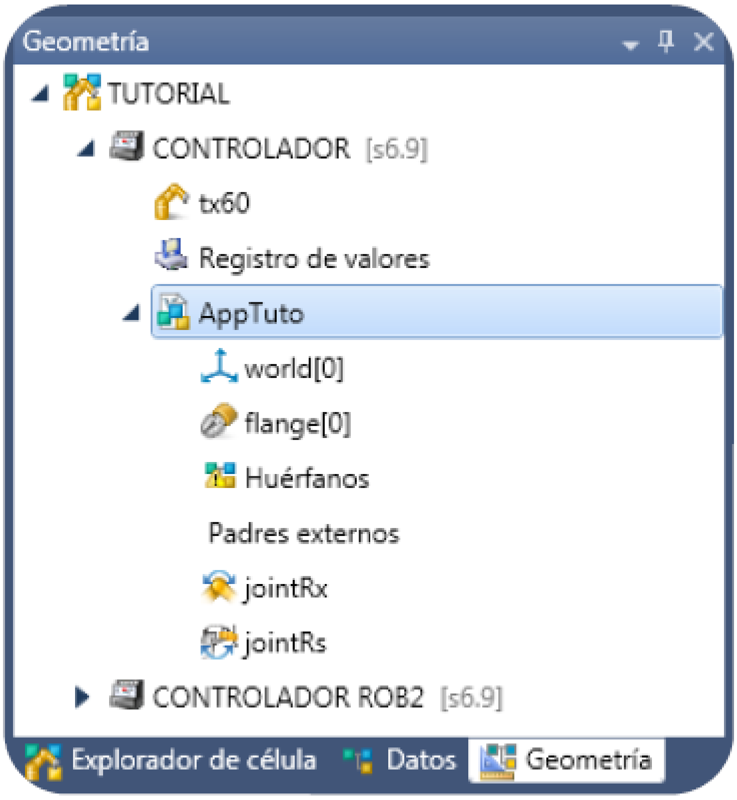
Contents of the new application in “Datos” (Data)

In folder “Datos” (Data) also appears the new application, with all the data belonging to the application. As we can see, there is the variable mNomSpeed (type mdesc) that holds the value of the speed of the arm by default.



Contents of New Application in “Geometría” (Geometry)

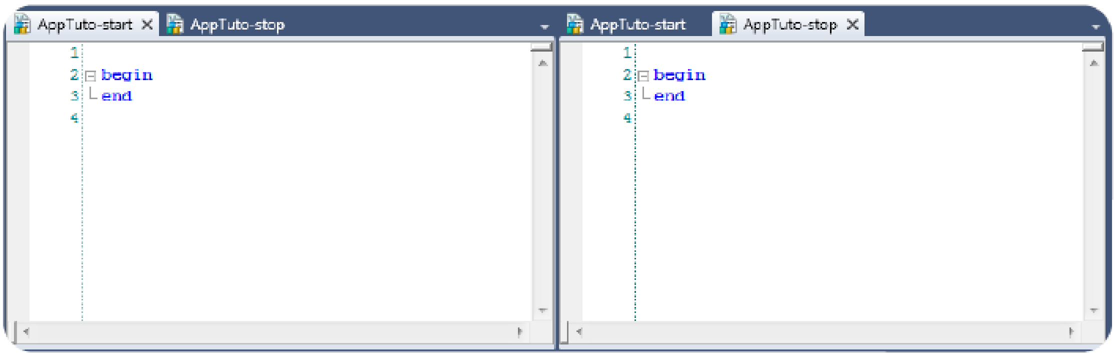
Also, the application appears in this folder, with all the geometrical data.



Edition of the application (Start and Stop)

To edit application AppTuto we are going to edit routines start and stop.

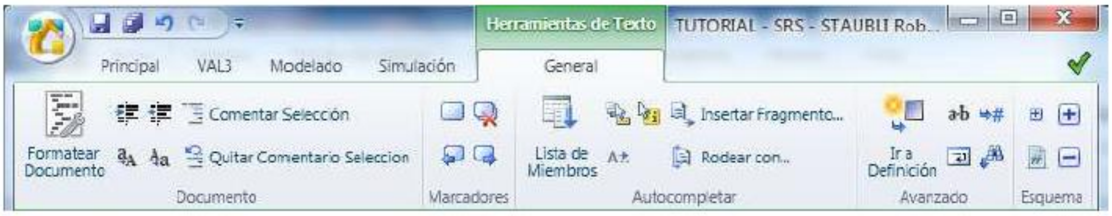
We have to go to “Explorador de Célula” (Cell Explorer) and double-click over Start() and Stop(). A new window will be opened where we can edit these routines.



```
begin
end
```

```
begin
end
```

In the bar above, a new folder appears “Herramientas de texto” (Text tools)



We will use the following pieces of code that we are going to copy in Start() and Stop() routines.
Code for Start() is the following. It is possible to change the comments, and we have to change the tool of the robot we are using in the example (called “eina0”) by the name of the tool of the robot by, *flange*, that it is not necessary to define (ii is the one by default)

```
Begin
userPage()
cls()
putln("Inici del programa")
enablePower()
mNomSpeed.vel=30
movej(posicio2[0],eina0[0],mNomSpeed)
close(eina0[0])
putln("Posicio 2a assolida. Element terminal tancat.")
open(eina0)
movej(posicio2[1],eina0[1],mNomSpeed)
close(eina0[1])
putln("Posicio 2b assolida. Element terminal tancat.")
end
```

For Stop() routine, we will use the following:

```
begin
movej(posicio0,eina0,mNomSpeed)
open(eina0)
putln("Posicio 0 assolida. Element terminal obert. Programa finalitzat")
popUpMsg("Pending movement commands have been canceled")
resetMotion()
end
```

All of these instructions belong to a very simple piece of code to show you how to use the Staubli Robotics Suite 2013 software.

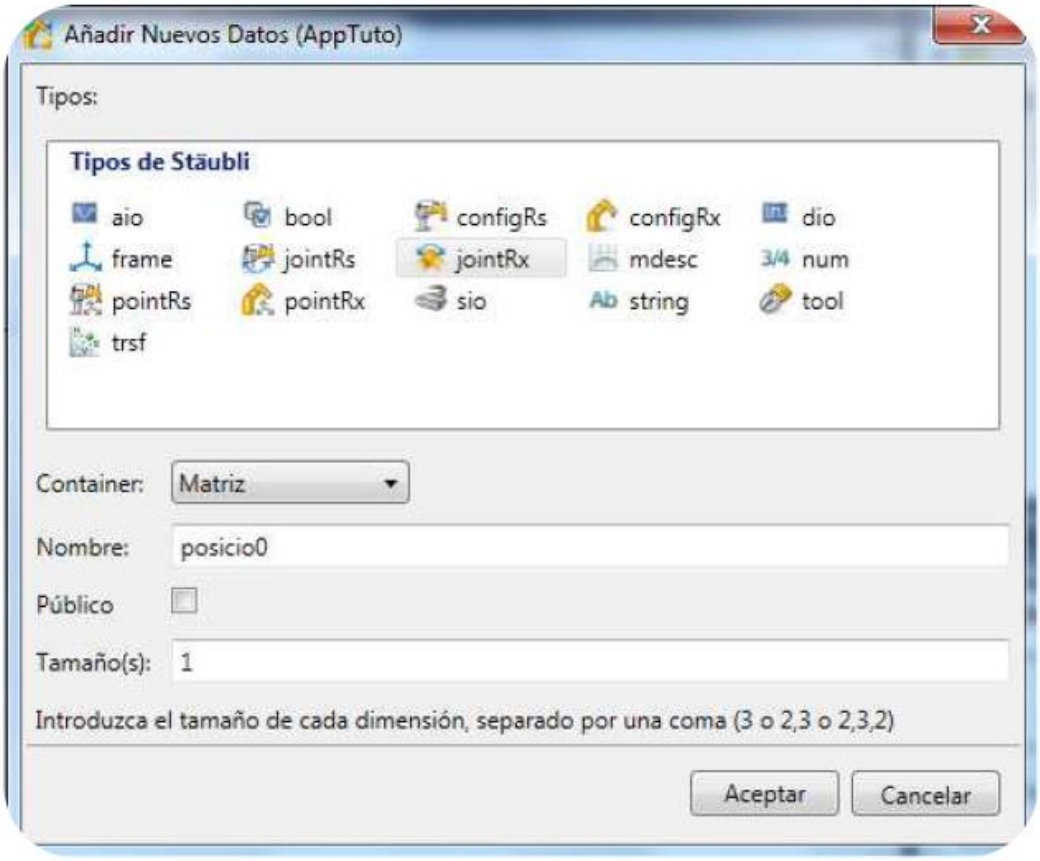
The elements that are still not defined and that are completely needed to compile the application are the points that we are using: posicio0 and posicio2.

To define posicio0 we can use the following procedure:

Going to “Datos” (Data), click with the right button of the mouse over the name of the application (AppTuto), press over “añadir” (add), “Nuevos datos” (New Data).

Then, configure the new point: “Nombre” (name): posicio0 and “Tipos” (type of data): jointRx, without adding anything more.

Type of data we are choosing in this case is jointRx (values of the joints to specify the point)



Container: Matriz

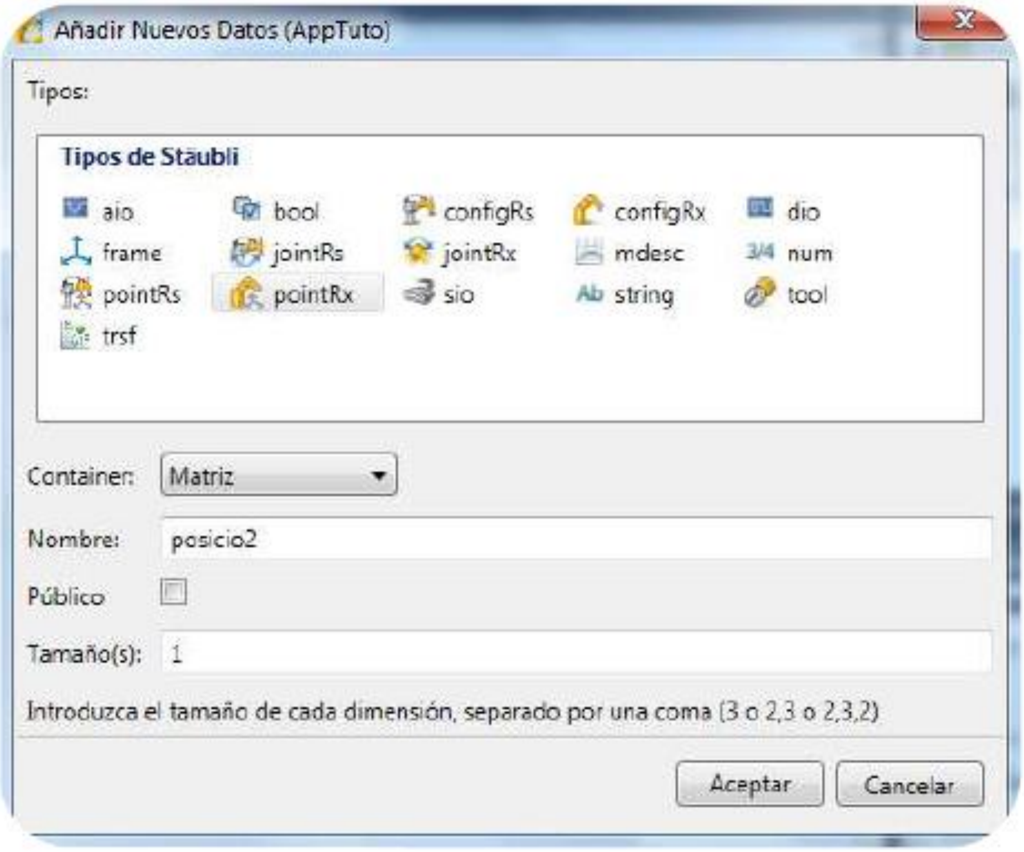
Nombre: posicio0

Público ☐

Tamaño(s): 1

Introduzca el tamaño de cada dimensión, separado por una coma (3 o 2,3 o 2,3,2)

To define posicio2, we have to repeat the previous process and we have to configure the new point. In this specific case, choose type of data pointRx (cartessian values to specify the point)



Container: Matriz

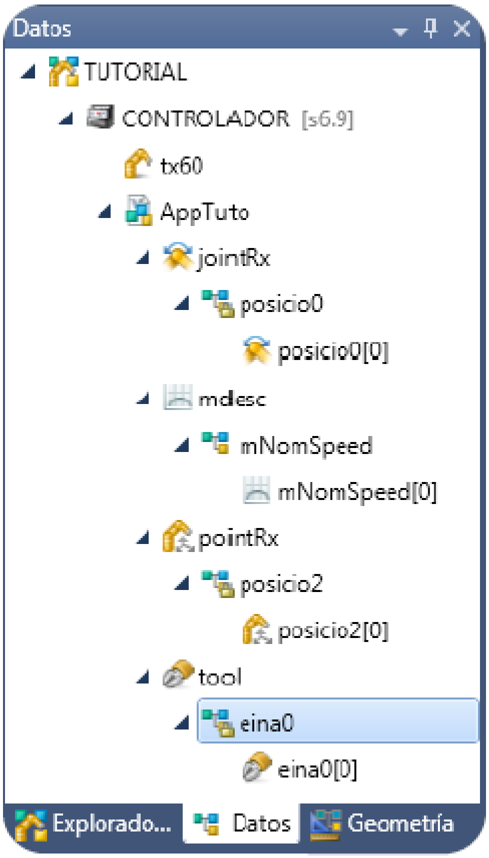
Nombre: posicio2

Público □

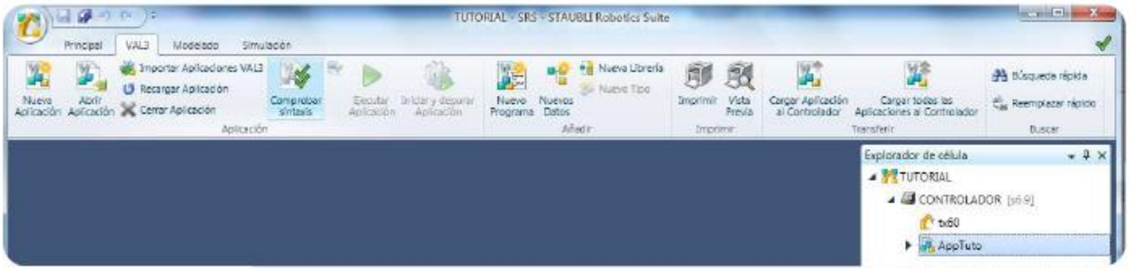
Tamaño(s): 1

Introduzca el tamaño de cada dimensión, separado por una coma (3 o 2,3 o 2,3,2)

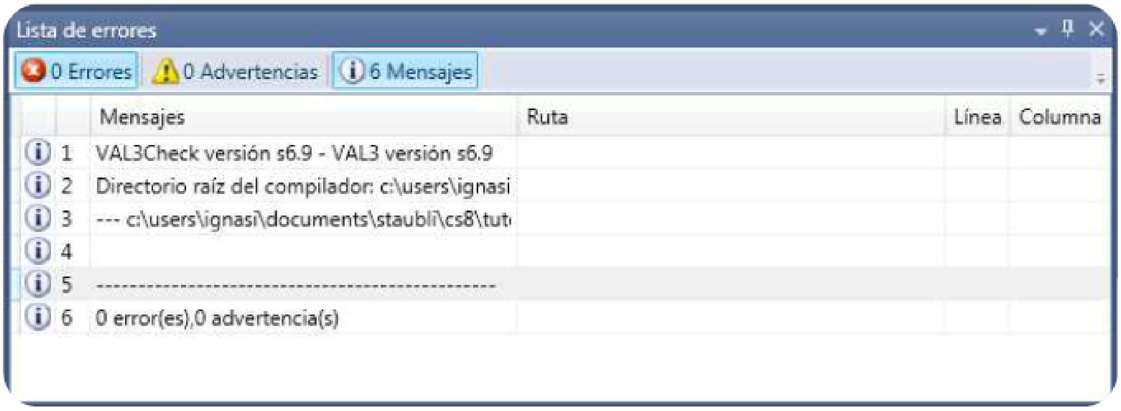
In the following figure we can see how the folder Data is after declaring all the required Data to allow the proper compilation of the application AppTuto (remember that we are not using the tool “eina0”).



Now, it is time to check the code. Close the folders of the editor of programs, saving the modifications we did. Now, we can go to Cell Explorer, click over the application AppTuto, and click on “comprovar sintaxis” (Check sintax) at the folder VAL3 on the bar.



There are not errors nor warnings.

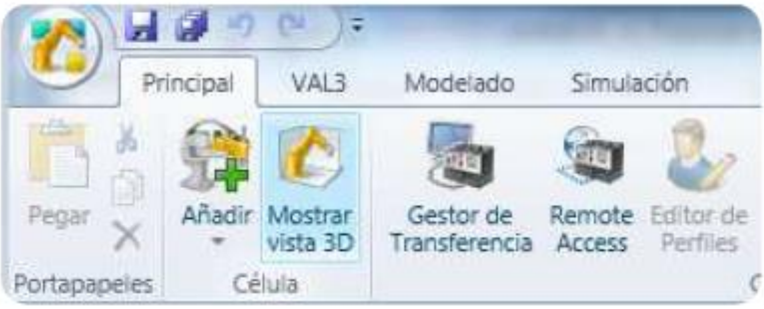


		Mensajes	Ruta	Línea	Columna
ⓘ	1	VAL3Check versión s6.9 - VAL3 versión s6.9			
ⓘ	2	Directorio raíz del compilador: c:\users\ignasi			
ⓘ	3	--- c:\users\ignasi\documents\staubli\cs8\tut			
ⓘ	4				
ⓘ	5	---			
ⓘ	6	0 error(es),0 advertencia(s)			

At this point, our application AppTuto is ready. We can then simulate it using the Emulator and link it to the 3D View to see how the robot is moving according to the program.

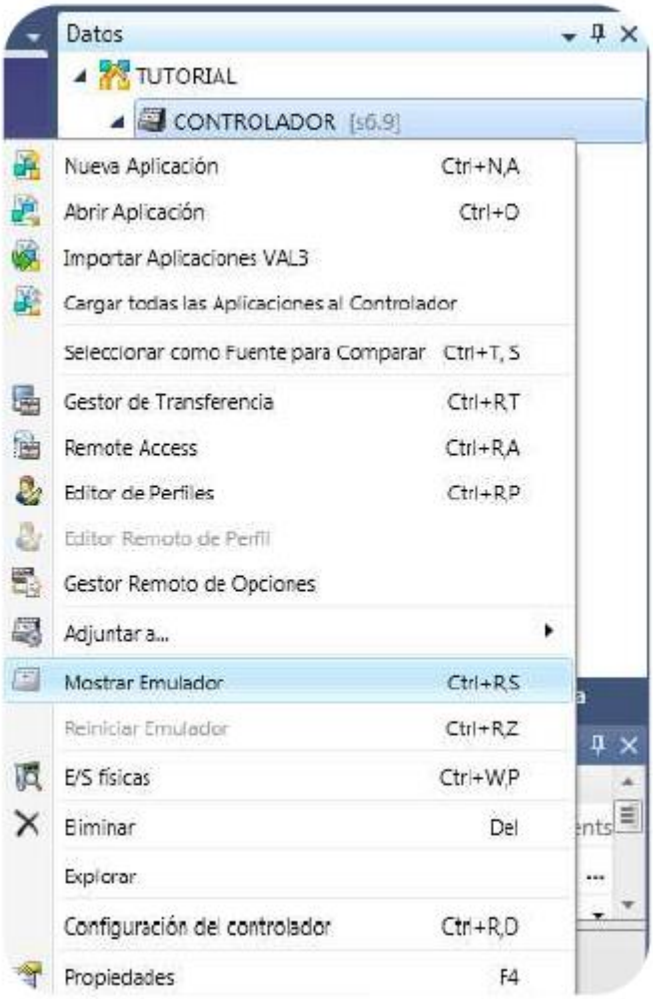
3D Model

To visualize the 3D model we have to click over “Mostrar vista 3D” (Show 3D view) folder “Principal” (Home) of the bar.

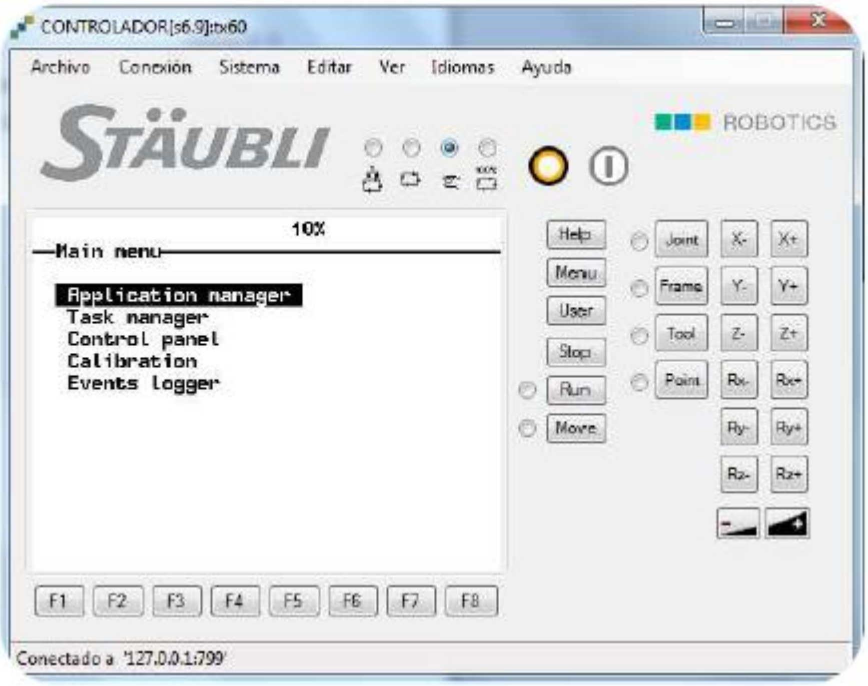


Emulator

Emulator is the software in charge of simulate the operation of the Control Unit and MCP of the Robot. We have to right-click with the mouse over the controller and select “Mostrar emulador” (Show emulator).


Datos
TUTORIAL
CONTROLADOR [s6.9]
Nueva Aplicación Ctrl+N,A
Abrir Aplicación Ctrl+O
Importar Aplicaciones VAL3
Cargar todas las Aplicaciones al Controlador
Seleccionar como Fuente para Comparar Ctrl+T, S
Gestor de Transferencia Ctrl+R,T
Remote Access Ctrl+R,A
Editor de Perfiles Ctrl+R,P
Editor Remoto de Perfil
Gestor Remoto de Opciones
Adjuntar a...
Mostrar Emulador Ctrl+R,S
Reiniciar Emulador Ctrl+R,Z
E/S físicas Ctrl+W,P
Eliminar Del
Explorar
Configuración del controlador Ctrl+R,D
Propiedades F4

An emergent window will appear that emulates faithfully the real MCP attached to the real Control Unit of the robot.



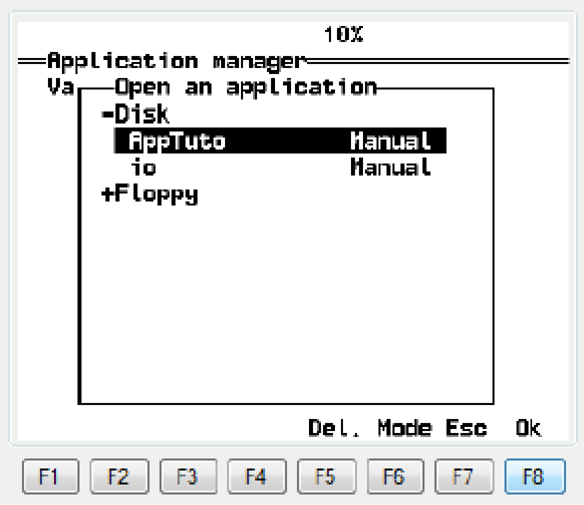
```
                      10%
—Main menu——————————————————————

  Application manager
   Task manager
   Control panel
   Calibration
   Events logger
```

Conectado a '127.0.0.1:799'

The arrows allow us to move through the menus of the MCP. The right arrow allows us to select the menu that is enhanced on the screen.

To come back to the main menu we can click over the key “Menu”.
Now, we can open the application AppTuto to simulate its execution. Choose the option “Application manager”, “Val3 Applications”, “Disk”. Then click F8 (OK).



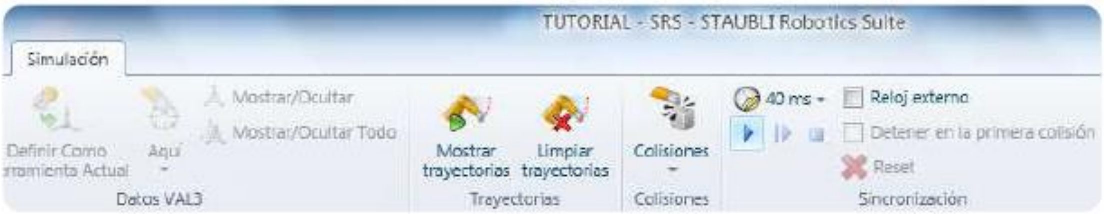
```
                    10%
==Application manager==========================
  Va—Open an application——
      -Disk
       AppTuto          Manual
        io              Manual
      +Floppy

                          Del. Mode Esc   Ok
```

We are now ready to start the simulation.

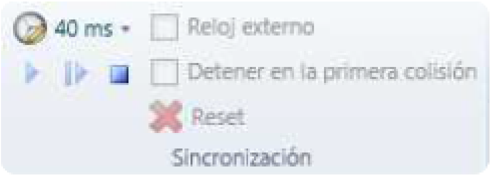
3D Simulation

To simulate the application the emulator must be active and synchronized with 3D Model. We have to go to options bar and select “simulación” (simulation). Then we have to use the controls available in “sincronización” (synchronization).



It is recommended to use a synchronization time of 40 ms (click on “Reloj externo” (external clock)) to see the simulation without problems.

Also, “play” symbol must be clicked (set, in sky blue) to synchronize the emulator with 3D view.



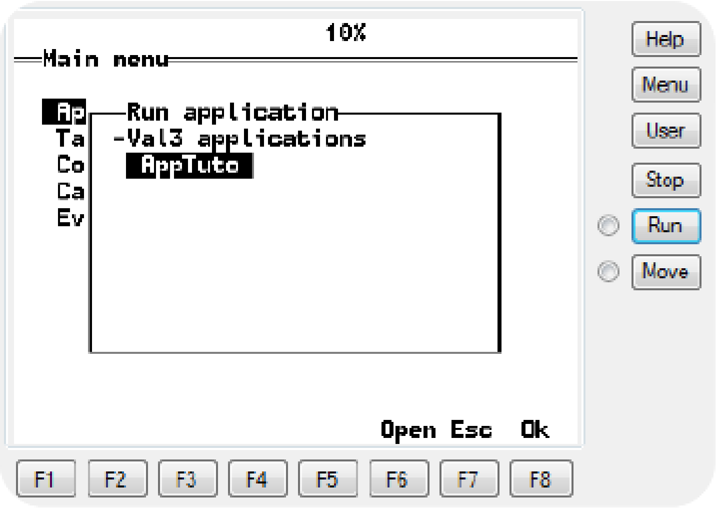
Now, we can move the robot, or simulate the movement of the robot corresponding to the code of the application. If we would like to see the trajectory of the robot we can set, “mostrar trayectorias” (show traces).

When we move the robot using the emulated MCP, we can look on the values of the different joints and Cartesian coordinates when the robot reach any interesting position in the space. We already have to define “posicio0” and “posicio2”, and we could use the values of two interesting positions of the robot in space to define these positions.

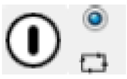
To move the robot on the screen, we have to select the manual mode of operation of the robot (the symbol of a hand), set one of the modes of movement (“joint”, for example), and to apply power to the arm, following the instructions that appear on the screen.

Power button is the one with a vertical bar that will light green when the power is applied.

If we want to execute an application, we have to click “Run” in the emulator of MCP. Previously, we have to unselect any mode of movement. We will see a new window on the screen with the name of all the applications already loaded in the emulator. We have to select AppTuto using the arrows keys, and click F8 (OK)



We have to enable the power again, pressing the power button. As you know, that button will light green showing that the arm is powered and ready. Then, we have to choose the mode of operation Mode 100 % (closed cycle with a 100%).



We can now press “Move” in the emulator
And the robot should move!!!!!!!!!!!!!!!!!!!!