

# REMOTE PROGRAMMING OF PLC'S AND HMI

## Worksheet

# Index

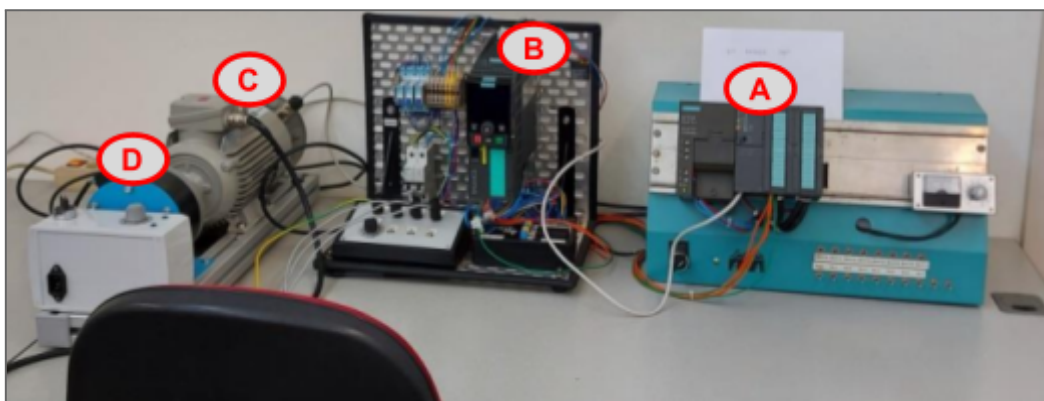
<b>Background</b>	<b>3</b>
<b>Targets</b>	<b>3</b>
<b>Necessary equipment</b>	<b>4</b>
<b>Equipment preparation</b>	<b>5</b>
<b>Reservation of equipment by each student</b>	<b>7</b>
<b>Remote connection instructions</b>	<b>8</b>
<b>Configuration and execution of Putty software</b>	<b>8</b>
<b>Configuration and execution of Sockscap software</b>	<b>11</b>
<b>Configuring PLC connectivity in the TIA Portal project</b>	<b>13</b>
<b>Description of the student's activity</b>	<b>15</b>
<b>General description</b>	<b>15</b>
<b>System description</b>	<b>15</b>
<b>Scheme of the installation</b>	<b>17</b>
<b>Description of the PLC programming and HMI panel configuration</b>	<b>18</b>
Programming and debugging of the PLC program	18
Configuring user features on the HMI panel	27

## 1. Necessary equipment

The activity is designed so that 6 students can work simultaneously from their homes, for a controlled period of time that will be managed through reservation.

Each of the 6 workstations consists of:

- A. SIMATIC PLC S7-314C v3.3
- B. Speed variator SIMANICS G120 CU250S-2PN Vector
- C. Asynchronous motor bench 4 poles 230v/0.55Kw
- D. Incremental encoder 1024pulses/rotation, coupled to the motor shaft.



Each student has for personal work:

- Win10Pro laptop with TIA Portal v17 licensed

## 2. Reservation of equipment by each student

A shared spreadsheet is made available to students to reserve equipment with a date and time.

EQUIP 2 (192.168.150.112)	02/01/2021	03/01/2021	04/01/2021	05/01/2021	06/01/2021	07/01/2021	08/01/2021	09/01/2021	10/01/2021
Hora									
0 a 1h									
1 a 2h									
2 a 3h									
3 a 4h									
4 a 5h									
5 a 6h									
6 a 7h									
7 a 8h									
8 a 9h								DAVID	Oscar
9 a 10h							DAVID	DAVID	Oscar
10 a 11h						DAVID	DAVID	DAVID	
11 a 12h						DAVID	DAVID	DAVID	
12 a 13h						DAVID			

### 3. Remote connection instructions

The remote connection between the equipment installed in the classroom and the student's PC is made through an SSH tunnel (in the future it will be done with VPN tunnel). Each student must download and install two programs on their laptop:

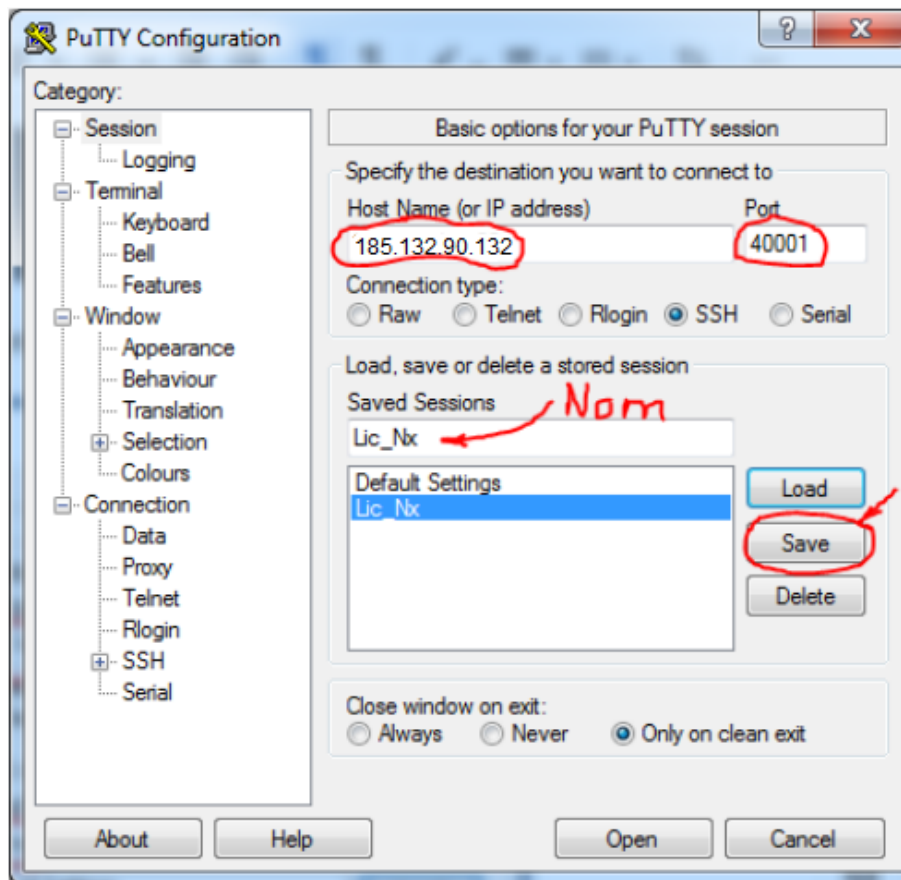
- [Putty](#)
- [Socksap](#)

#### 3.1. Configuration and execution of Putty software

Once the Putty executable file has been downloaded, it must be run:

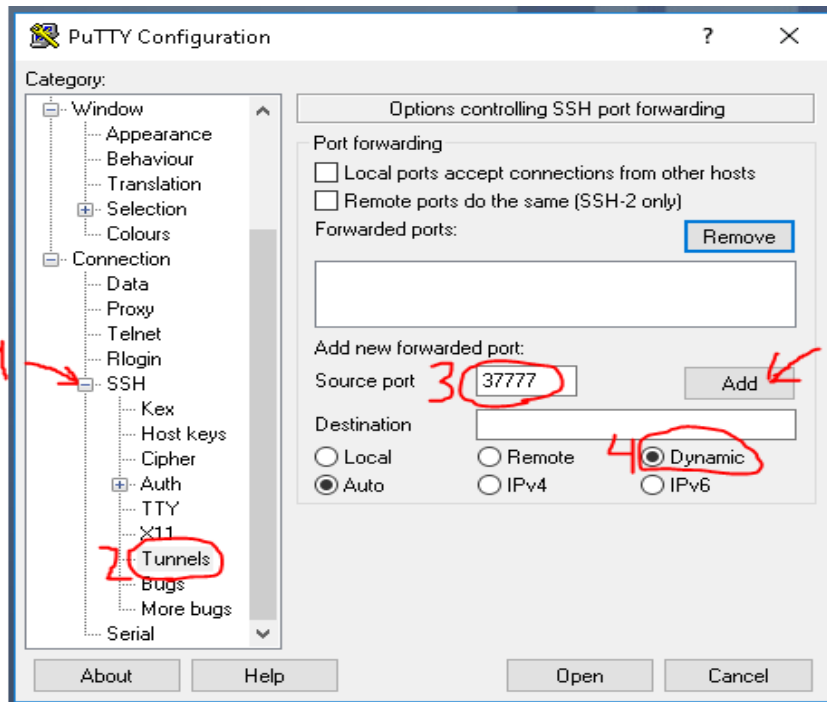


Once opened, we add the following data and save "Save":

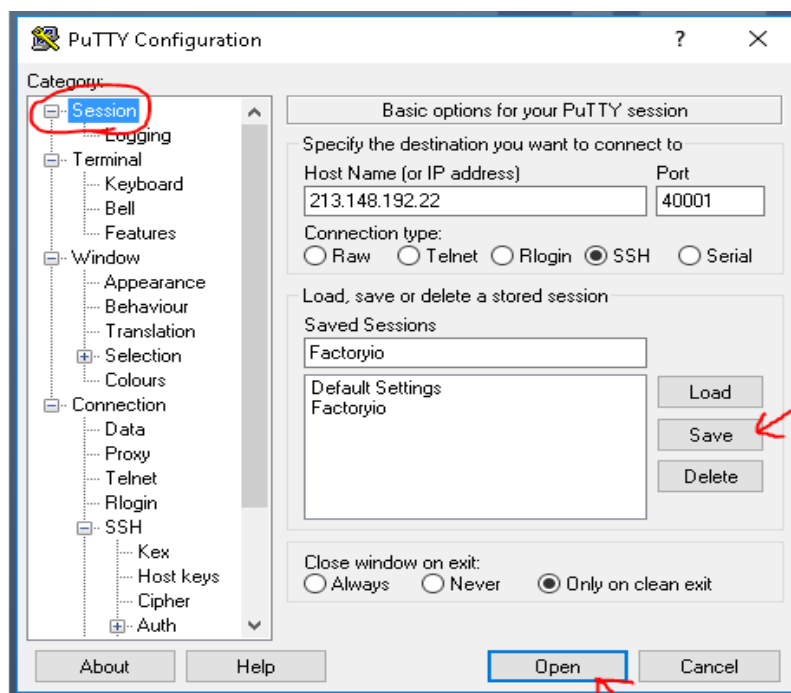


We will put a consistent name, such as: **Connexió\_Remota\_Servidor\_EEO**

Click on the SSH option in the "category" tree, left margin of the interface:

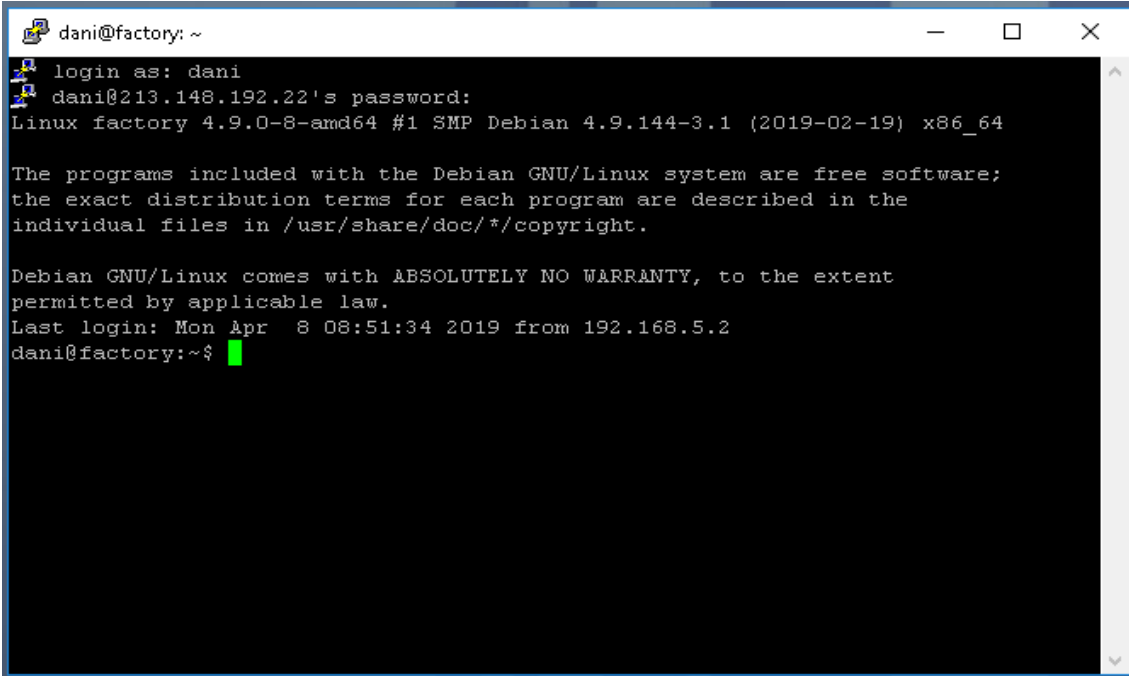


Now we return to the main screen of the "putty" and save the profile created:



To set up the SSH tunnel between the student's PC and the Dep EEO server, open Putty, load profile ( LOAD ) and open ( OPEN ) connection with your username and keyword, and leave the terminal minimised.

User Name: xxxxxxxxxxxx Password: yyyyyyyyyyyyyyyy

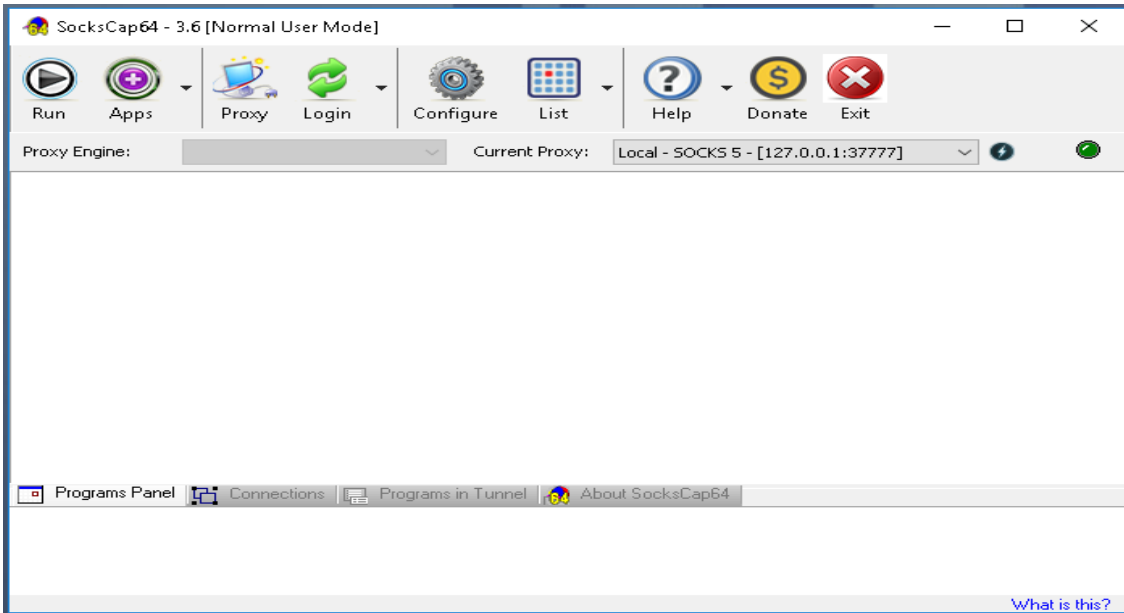


If the Putty reacts as in the picture, you already have the SSH tunnel set, you can leave the Putty minimised.

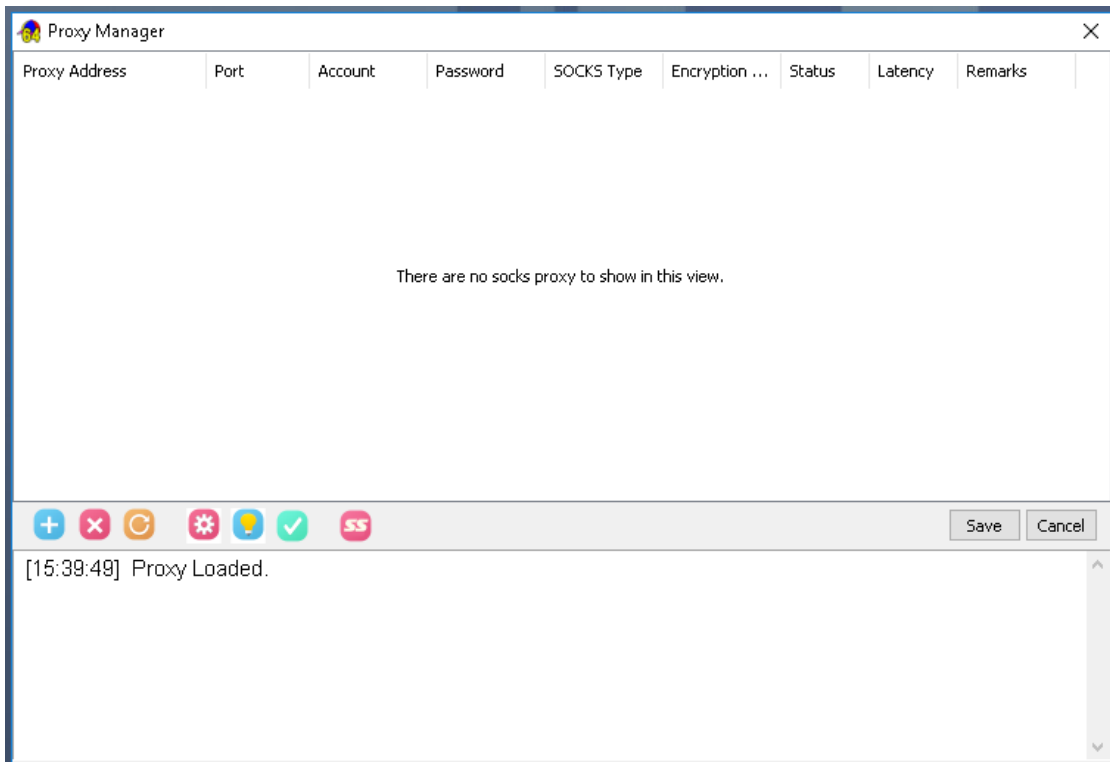
Note: The teacher will have previously registered all students with a Username and Password on the server.

### 3.2. Configuration and execution of Sockscap software

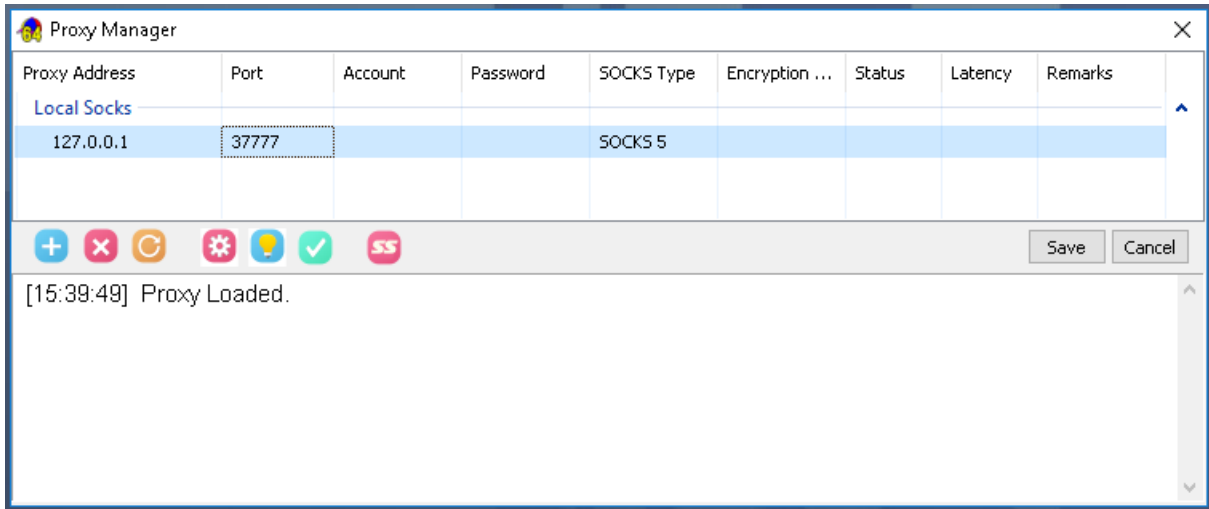
After downloading and installing the software on the PC, the following interface will be run and observed:



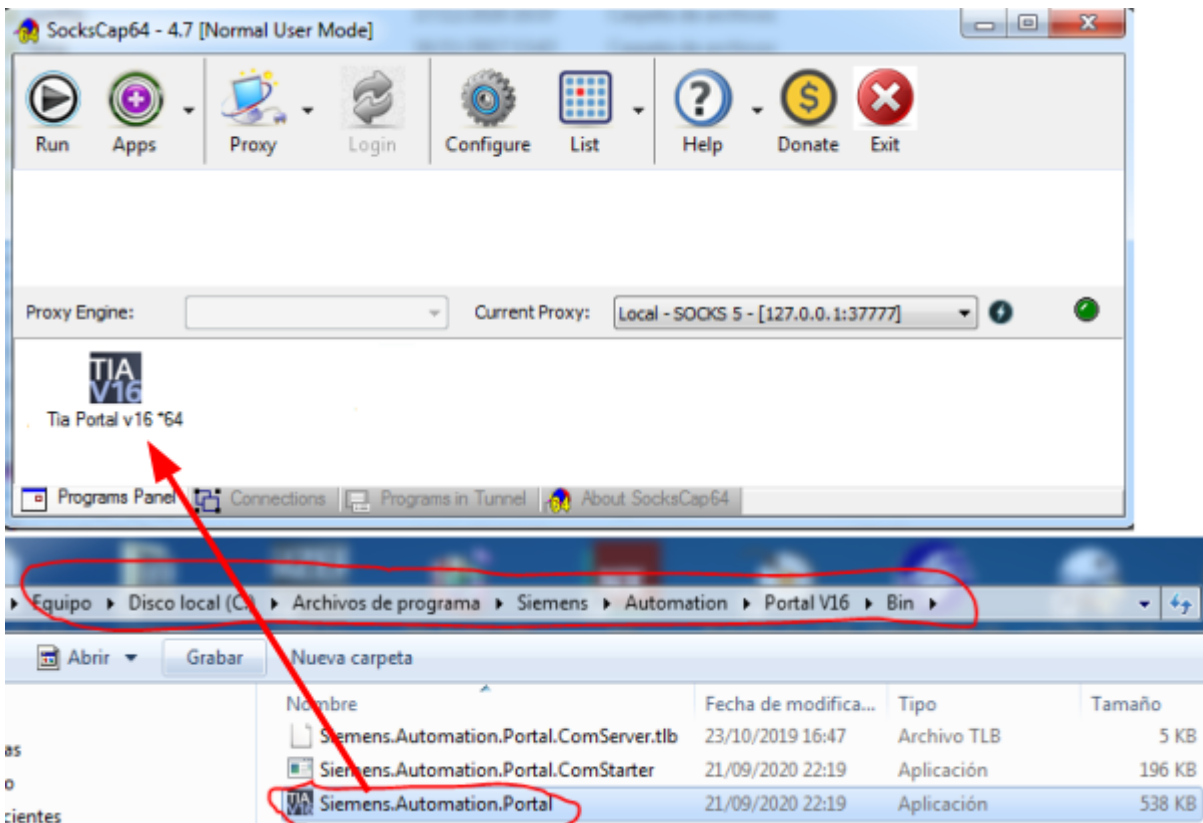
Now click on the "Proxy" icon,



By pressing the (+) icon, add the data of the image ..., and “save”



Now all you have to do is drag the TIA Portal icon into the SocksCap window, from the indicated directory:

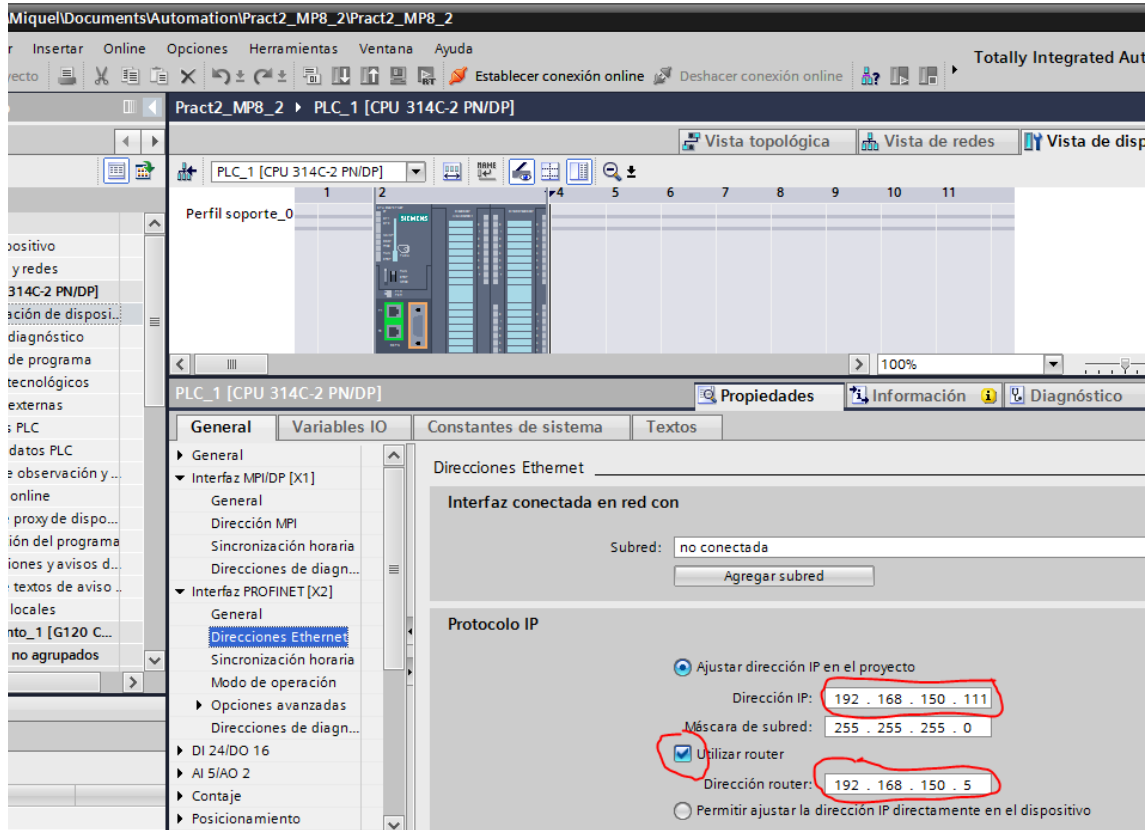


To start the TIA Portal, double-click on the icon you just inserted in the Sockscap, the work environment will open and you can work as if you were in the classroom.



### 3.3. Configuring PLC connectivity in the TIA Portal project

Define in the TIA Portal, in your project, the CPU model you want to connect to the classroom network, and assign it an IP and a router IP, as you see in the figure:



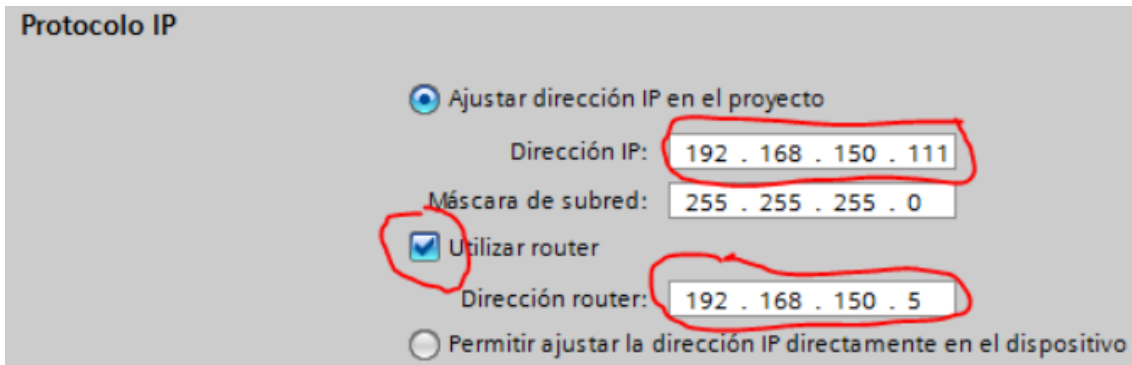
The IP of the PLC must match the range of IPs in the classroom: 192.168.150.XXX where XXX is a different number for each CPU.

We can give a number of IP XXX = from 101 to 130. In the example it is 111. For this application from 111 to 115 depending on the workstation you want to connect to.

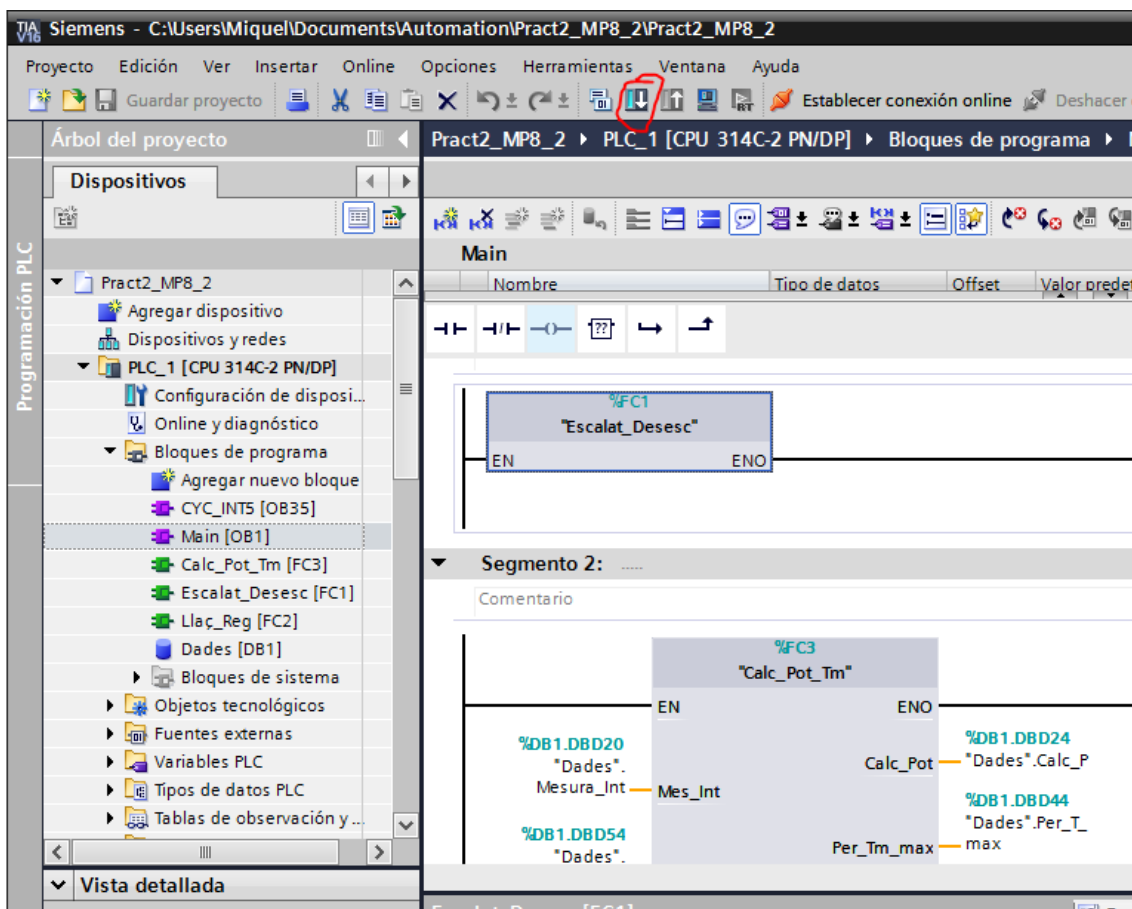
Marck **Use router** , and assign this IP: **192.168.150.5**. This should always be the same.

#### IMPORTANT

- Initially all PLCs are in STOP.
- VERY IMPORTANT: Check that the IP of the TIA matches that of the CPU you want to connect to (If you duplicate tIPs or send one out of range 150, you will leave the workstation unused) :



- You can now load the entire TIA Project (HW and program) to the PLC:



Now, you can work as if you were in the classroom.

For closed work session :

- PUT THE PLC ON STOP
- Close the TIA Portal saving before if you want.
- Close the Putty. The VPN tunnel will end.
- Close the SocksCap. Remember that it is running in the background.

## 4. Description of the student's activity

### 4.1. General description

The activity consists of the realization of a control in proportional closed loop (No PID) of an electrical machine, analyzing several parameters.

The system must enable the machine to work in different regimes, which will be determined through the operator panel.

### 4.2. System description

It is designed to control an asynchronous motor using a frequency converter controlled through its digital and analog inputs.

It is a 400/230v 0.55Kw 4-pole motor in a triangle connection, which when operated by the G120 converter can operate a machine at variable speed .

The machine in question will be emulated in the assembly by a hysteresis brake coupled to the motor shaft, which will cause variations in torque resistant ( $T_r$ ) to the shaft, so that the motor will respond by modifying the torque ( $Torque$ ) adapting it at all times to the demand for force on the shaft. As a result, variations in motor torque ( $T_m$ ) will cause changes in motor power consumption ( $T_m \uparrow \quad I_{Fase} \uparrow$ ), which will be reflected in the instantaneous power consumed.

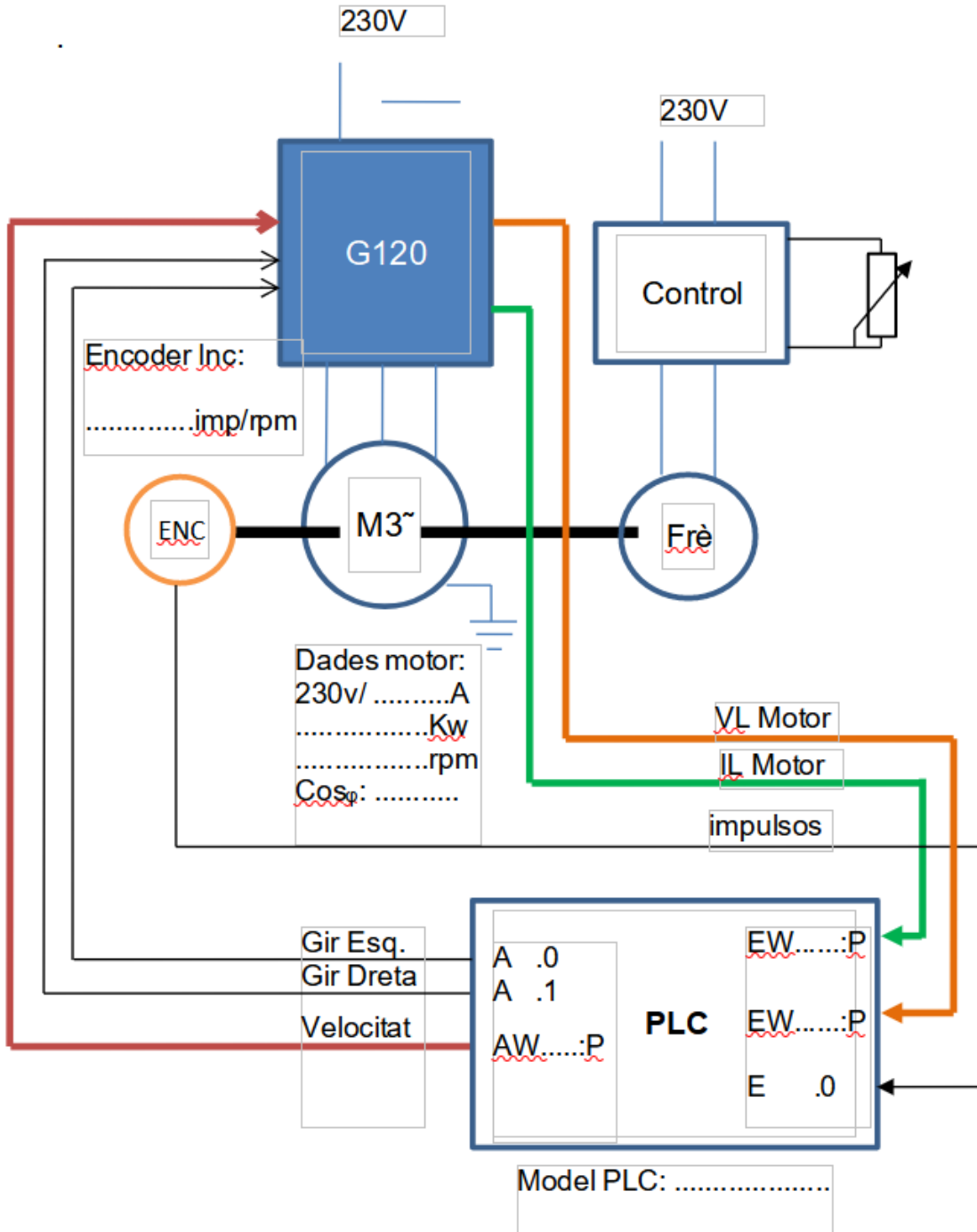
The whole system will be controlled from a PLC, which will perform the following actions:

- Variable measurement actions
  - Measurement of the actual motor speed from the pulses obtained from the encoder coupled to the shaft.
  - Measurement of the phase current supplied by the drive by connecting the analog output of this (configured to read the phase current) to an analog input of the PLC.
  - Measurement of motor line voltage (VL) in the same way.
  - Calculation of the instantaneous active power (Kw), extrapolating the value of power factor from the current value.

## 01

- Calculation of the instantaneous torque developed ( $T_m$ ), and of the percentage with respect to the maximum possible torque ( $T_m \text{ MAX}$ ).
- Control actions
  - Operation in two selectable modes (Manual or Stade State)  
In **Manual Mode** the operator of the machine can change at will, the setpoint of speed from the panel HMI.  
In **Mode Stade State**, the operator only chooses in a previously entered recipe, the working mode of the motor.
  - Configuration of alarm variables, and system shutdown in case they are activated.
  - Proportional closed loop motor control (both in manual mode and Stade State mode). It is decided to apply a proportional closed loop because of its simplicity of implementation and so that the student can check its rather unstable behavior when modifying the proportional action. (later activities with control by PID are realized, where the student will be able to catch the difference of behavior with respect to this system with proportional control)

### 4.3. Scheme of the installation



Write down the necessary data, the spaces in discontinuous, to be able to carry out the calculations to the PLC program.

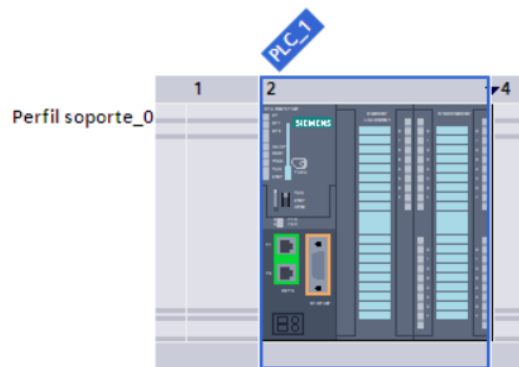
## 4.4. Description of the PLC programming and HMI panel configuration

In this point it is described the actions to be carried out by the student, organizations in two blocks:

- Programming and debugging of the PLC program
- Configuration of the user functionalities in the HMI panel

### 4.4.1. Programming and debugging of the PLC program

First of all it is to carry out the configuration of the HW of the PLC according to the PLC that you have chosen to carry out the activity.



- Configure the digital and analog I / O that you want to use.
- Cycle marks.
- IP connectivity.

### Organization of the program

The program will be organized in FC functions, each one developing specific tasks.

FC1: Scale of the measured variables and descale of the control variable

FC2: Loop of proportional regulation

FC3: Direction of rotation

FC4: Calculations of power, torque, etc.

FC5: Start-stop control and alarm management

OB1: Unconditional calls of functions

OB100: Initialization of variables

FC1 Scaling of measured variables and scaling of control action variable

Scaling of measured variables

The system measures three process variables:

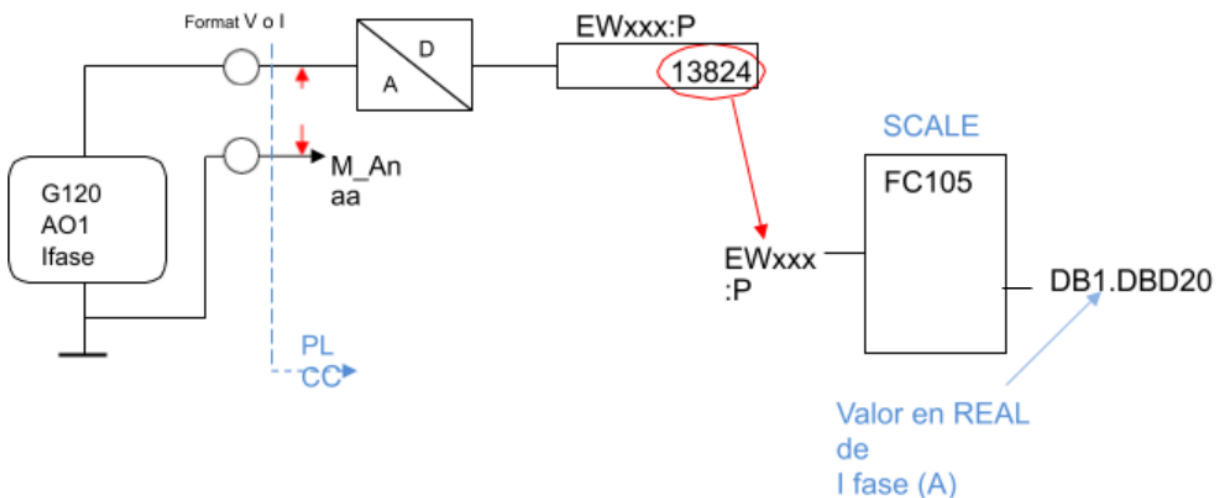
- Phase current ( $I_{FASE}$ ), via analog input.
- Motor line voltage ( $V_L$ ), via analog input.
- Real axis speed (via HSC digital input)

**Phase intensity:** EW .....: P

It is read directly from the frequency converter's analog output, so this output must be set to provide the phase current value from the motor.

In the PLC we will configure an analog input of this in the same format (V or I) of the frequency converter, and with the same range of work, and we will scale it to give units within the limits of work of the motor with 80% of margin at the upper limit.

Save the result to **"Mesura\_Int": DB1.DBD20**



**Motor line voltage:** EW .....: P

Use the 2nd analog output of the PLC to extract the line voltage applied at all times to the motor windings, and scale it as you did with the current.

Save the result to **"Mesura\_VL\_Motor": DB1.DBD54**

**Motor speed:** E ..... 0 (AWL)

In this case, the reading is based on the frequency measurement of encoder pulses using digital fast counting inputs. (HSC)

Later we will scale within this same function the frequency measured at rpm, performing a unit change by applying the conversion factors with the appropriate data according to the available devices. (encoder pulses / rev)

$$xxx \frac{imp}{s} \cdot \frac{1}{yyy \frac{imp}{rev}} \cdot \frac{60s}{1min} = rpm$$

Check if the maximum frequency measured at 1500rpm is within the measurement limits of the PLC Fast Counter (HSC).

Motor working range: 0 to 1500rpm.

Save the measured speed to **“Mesura\_Vel”**: DB1.DBD16

**Control of the G120 speed shifter from the analog output of the PLC:** Take advantage of the same function to de-escalate the control action on the frequency converter. The speed (rpm) value to be sent to the frequency converter will be located in **“Accio\_RPM”**: DB1.DBD48, in REAL format, and this value must be scaled (UNSCALE S7-300) to be provided by the free analog output of the PLC (**AWxxx: P**)

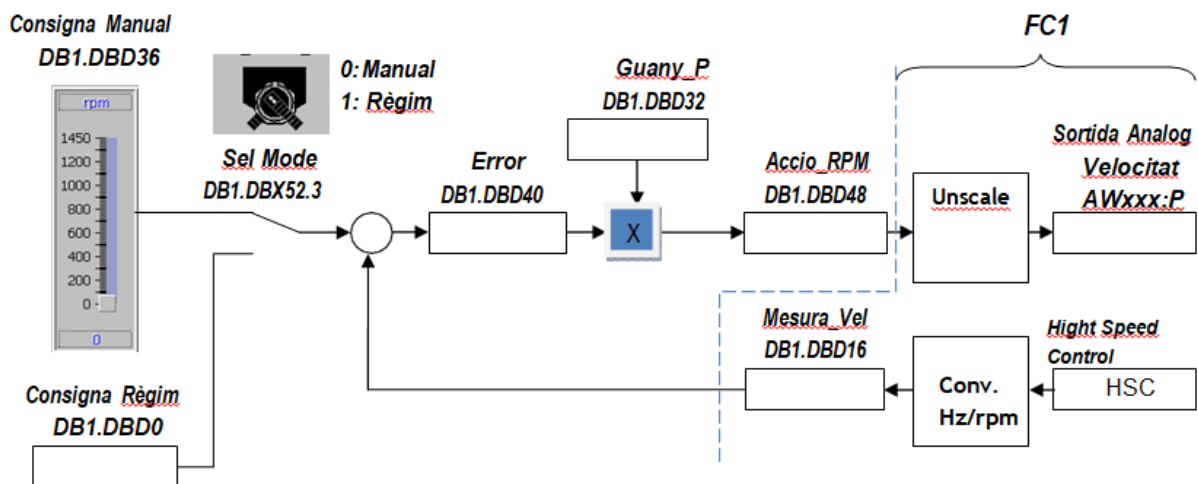


### FC2: Loop of proportional regulation

This function will define the speed control loop that will govern the control action on the frequency converter.

All operations are defined in the following block diagram, just follow it carefully.

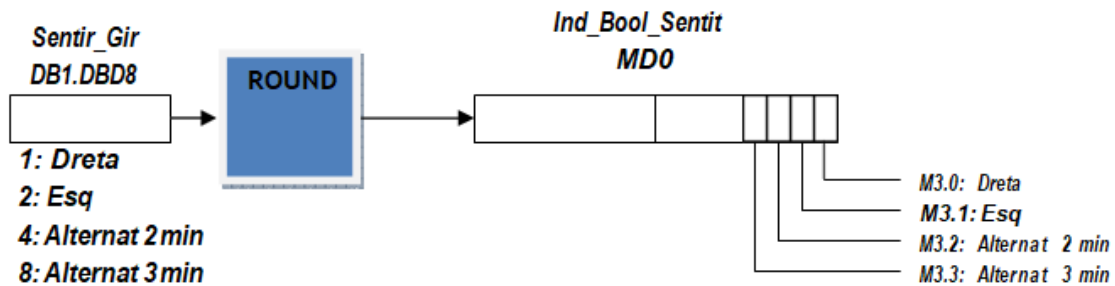
Please note that it must be possible to choose the type of speed setpoint (manual or stade) using the **Sel\_Mode** bit, which will act as a switch, and that all operations are performed in REAL format.



FC3: Direction of rotation

This function has to produce the Boolean signals that will activate the direction of rotation of the machine, bearing in mind that this may be determined by the working regime defined in the recipe, or operated manually when the system is under this mode of operation.

**1st** We will create some Boolean indicators from the data of direction of rotation indicated in the stade.



The data is REAL, using the ROUND function we get an integer DINT of 32 bits, that if we save them to MD0 we will obtain boolean indicators when testing bits M3.0 to M3.3. The numbers (1,2,4,8) that the panel stores in DB1.DBD8 have already been thought of so that a 1 appears in the corresponding bit once the data is converted to DINT. Check it out.

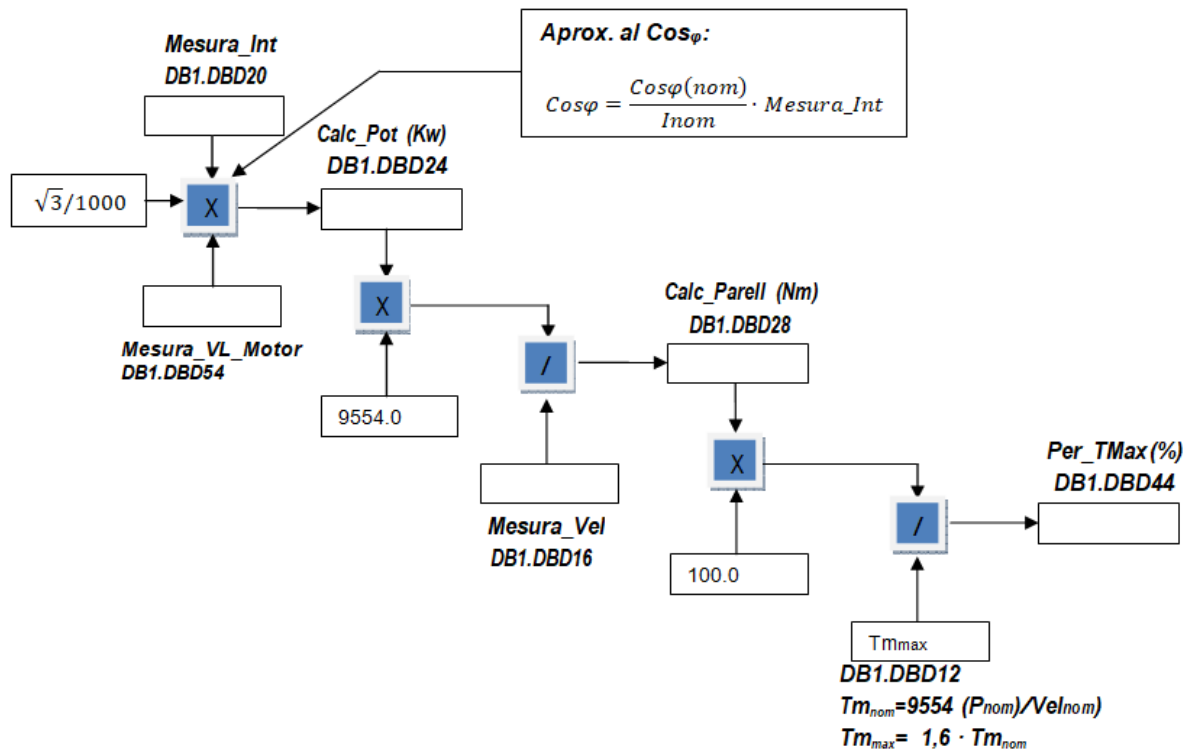
**2nd** Create a 4 min clock with two timers, and repeat it to create another 6 min clock, which will be used for alternated modes.

**3rd** Think and edit the Boolean logic that activates the outputs of the direction of rotation (A124.0, A124.1), and the pilot indicators of the panel (DB1.DBX52.0, DB1.DBX52.1), using the variables:

- Sentit\_Man DB1.DBX52.5)
- Sel\_Mode (DB1.DBX52.3)
- Ind\_Bool Sentit (M3.0 a M3.3)
- Clock timers for alternating modes 2 min and 3min.

FC4: Calculations of power, torque, %T<sub>MAX</sub>

This function will be responsible for performing the calculations from the original data of **Mesura\_Vel (DB1.DBD16)** and **Mesura\_Int (DB1.DBD20)**, to reach the values of power (Kw) in **DB1.DBD24**, motor torque (T<sub>m</sub> in Nm) in **DB1.DBD28**, and %T<sub>MAX</sub> in **DB1.DBD44**.



### FC5: Start-stop control and alarm management

Interlock a boolean from the START button, and reset it from the STOP button, and use it so that the control action on the converter does not execute when stopped. Edit protection so that the system shuts down automatically when:

**Mesura\_Int > I\_Lim\_Max** for 60 seg in a row

OB100: Initialization of variables

Make sure that every time the PLC goes to RUN, it is always in:

- Mode Manual
- Guany = 1
- Sentit Dreta
- I\_Lim\_Max=3A

**Operating data under regime from the panel (Recipe):**

REGIME	Speed setpoint	Direction of rotation	Value	I_Lim_Màx
Regime Low	300 rpm	Right	1	1.0A
Regime Midium	500 rpm	Left	2	1.2A
Regime High	900 rpm	Alternate every 2 min	4	1,8A
Regime Max	1400 rpm	Alternate every 3 min	8	2.0A

**List of variables in DB1 “Dades”**

All data assigned by panel, and the results of measurements and calculations to be displayed in this panel, will be saved in the DB1 with symbolic name “Dades”

Name	Address	Variable type	
Consigna_Vel	DB1.DBD0	REAL	
I_Lim_Max	DB1.DBD4	REAL	
Sentit_Gir	DB1.DBD8	REAL	
Parell_Max	DB1.DBD12	REAL	
Mesura_Vel	DB1.DBD16	REAL	
Mesura_Int	DB1.DBD20	REAL	
Calc_P	DB1.DBD24	REAL	
Calc_Parell	DB1.DBD28	REAL	
Guany_P	DB1.DBD32	REAL	
Consigna_Vel_Man	DB1.DBD36	REAL	
Error_Vel	DB1.DBD40	REAL	
Per_T_Max	DB1.DBD44	REAL	
Accio_RPM	DB1.DBD48	REAL	
Ind_Gir_Dreta	DB1.DBX52.0	BOOL	
Ind_Gir_Esq	DB1.DBX52.1	BOOL	
Sel_Mode	DB1.DBX52.3	BOOL	
Botó Inici	DB1.DBX52.4	BOOL	
Sentit_Man	DB1.DBX52.5	BOOL	
Aturada	DB1.DBX52.6	BOOL	
Mesura_VL_Motor	DB1.DB54	REAL	

**Variable E/S**

Name	Address	Variable type	
Int_Motor	EW.....:P	INT	Entrada de perifèria
VL_Motor	EW.....:P	INT	Entrada de perifèria
Impulsos Encoder	E.....0	BOOL	DI
Vel_Conv_Freq	AW.....:P	INT	Sortida de perifèria
Arranc_Dreta	A.....0	BOOL	Sortida digital
Arranc_Esq	A.....1	BOOL	Sortida digital

#### 4.4.2. Configuring user features on the HMI panel

The system is fully controlled by the PLC, and monitored by TP700 Comfort operator panel. From the root window (**Cover**), we have three options



- **Paràmetres**, gives access to a data entry window.
- **Control**, switches to the motor control window in both Manual and Stade mode.
  - Manual mode is to operate the engine (speed and direction of rotation), using manually operated operator controls.
  - Stade Mode allows you to define between 4 possible predefined forms of action using Recipe.
- **Règim Treball**, gives direct access to the recipe settings window.

**Control Windows**

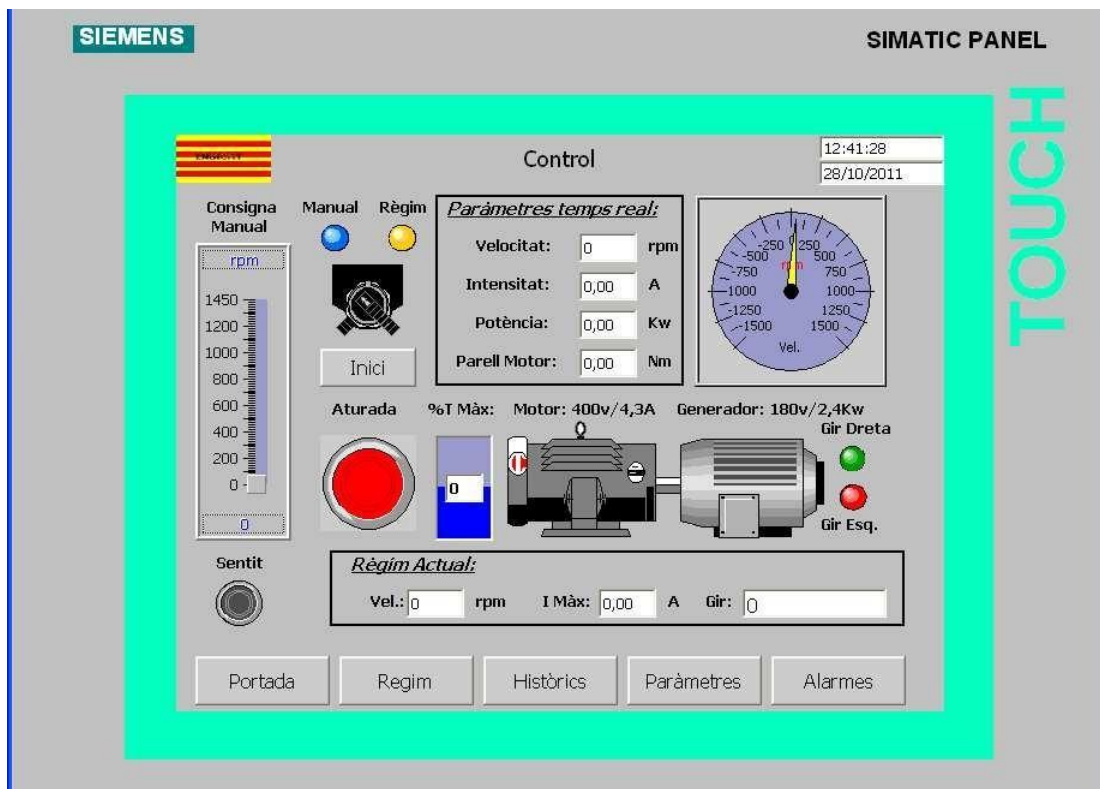
This window is the one that enables the operation of the system, and informs the operator of the working conditions of the system.

The system can be operated in two operating modes depending on the status of the corresponding switch, and the indication of the state status(Manual or Regime). The mode switch and the direction of rotation button are made by playing with the visibility of two superimposed images, and to create the switching event an invisible button is placed over them:

- Manual operation: The Manual setpoint slider and the direction of rotation button are activated and remain sensitive to touch. In the Regime position both remain disabled.  
Related PLC variables: Consigna\_Vel\_Man, Sentit\_Man, Sel\_Mode
- Regime operation: In this mode of operation, manual controls are disabled, and operating conditions are determined by the operating mode selected by the recipe tool located in the Règim window.

The selected current mode is indicated in the E/S fields configured as information output, and included under the Règim Actual area.

Related PLC variables: Consigna\_Vel, I\_Lim\_Max, Sentit\_Gir.





The window also shows the operating conditions of the system in real time:

- Speed is obtained from DB1 (**Mesura\_Vel**). It will also be associated with a graphical speed indicator.
- The intensity is obtained from DB1 (**Mesura\_Int**).
- Power is also related to Data DB1 (**Calc\_P**).
- Motor Torque (**Calc\_Parell**) is calculated by the PLC in FC4, according to the function:

$$Tm(Nm) = 9554 \cdot \frac{P(Kw)}{n(rpm)}$$

The %T<sub>MAX</sub> (Per\_T\_Max) is calculated as indicated in the FC4  $Tm_{Max} = 1.6 \cdot Tm_{nom}$ . This data is associated with the graphics bar and E/S field next to the motor.

Direction indicators assigned to two light pilots Ind\_Gir\_Dreta and Ind\_Gir\_Esq.

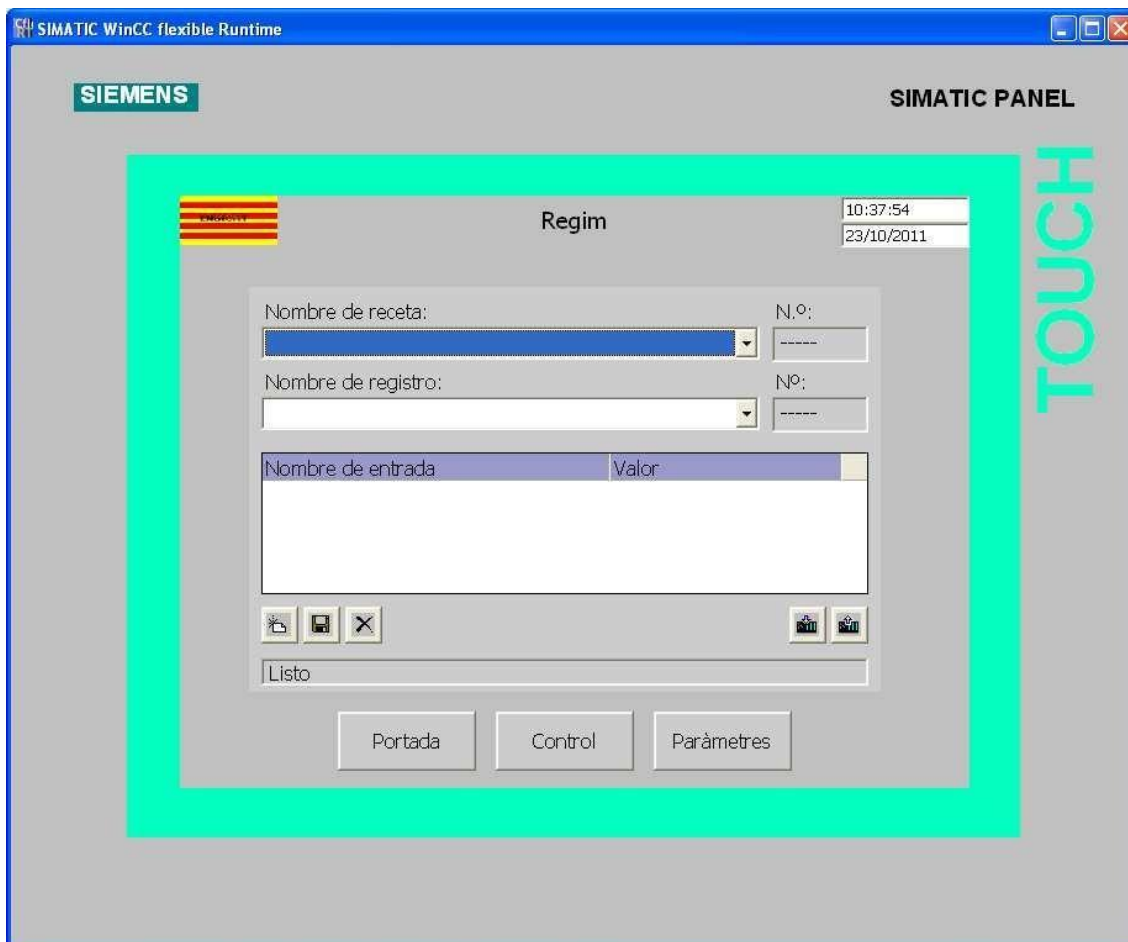
The Start and Stop buttons assigned to the corresponding DB data.

**Regim Window**

In this window is the recipe definition tool as shown in the grid in point 5, which is also listed below:

REGIME	Speed setpoint	Direction of rotation	Value	I_Lim_Màx
Regime Low	300 rpm	Right	1	1.0A
Regime Midium	500 rpm	Left	2	1.2A
Regime High	900 rpm	Alternate every 2 min	4	1,8A
Regime Max	1400 rpm	Alternate every 3 min	8	2.0A

The mode determines the setpoint speed, the direction of rotation (left, right, alternate), and the maximum allowable current intensity (**I\_Lim\_Max**) from which an error event will be triggered. The direction of rotation data is a numerical data so that the PLC will cause the indicated turn according to its value, in addition to turning on the direction indicator lights.



**Històrics Window**

In this window three curves will be drawn, which will graphically show the behavior of the system.

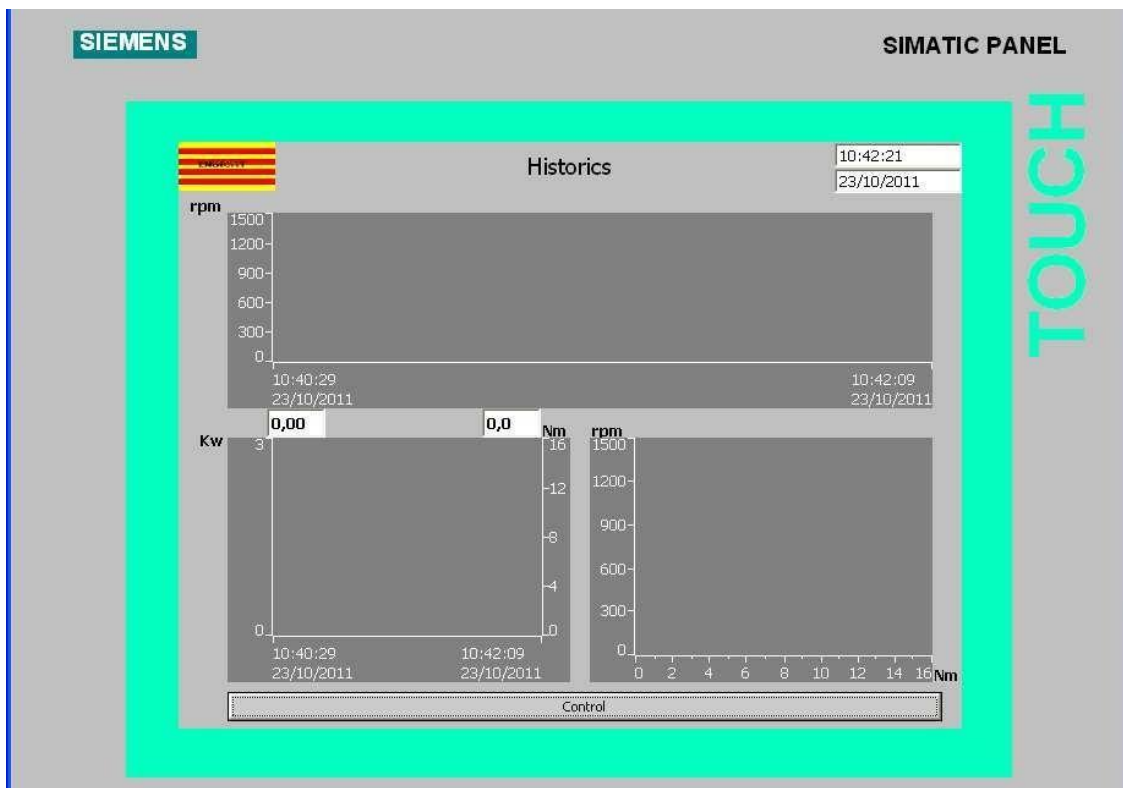
**Speed/t curve:** RT curve with respect to the time when two traces of different color are to be shown:

- Setpoint speed
- Current speed

**Power/Torque (motor/t) curve:** RT time curve where three traces of different color should be displayed:

- Power in Kw
- Trace of  $T_m$
- $T_{m_{MAX}}$

**Tm-speed curve:** HT (historic) curve with a single trace that relates to the Y axis the  $T_m$  and the X axis the speed in rpm. Note that in the image the axes are changed by mistake. Adjust this for a small sampling time.



The three curves can be distributed in different windows to improve quality, include captions about the meaning of each trace, and HT curve controls.

### Alarmes o Errors Window

The alarms are controlled by the PLC, and it is necessary to define variables Bool, to mark each alarm, and to make the ACK.

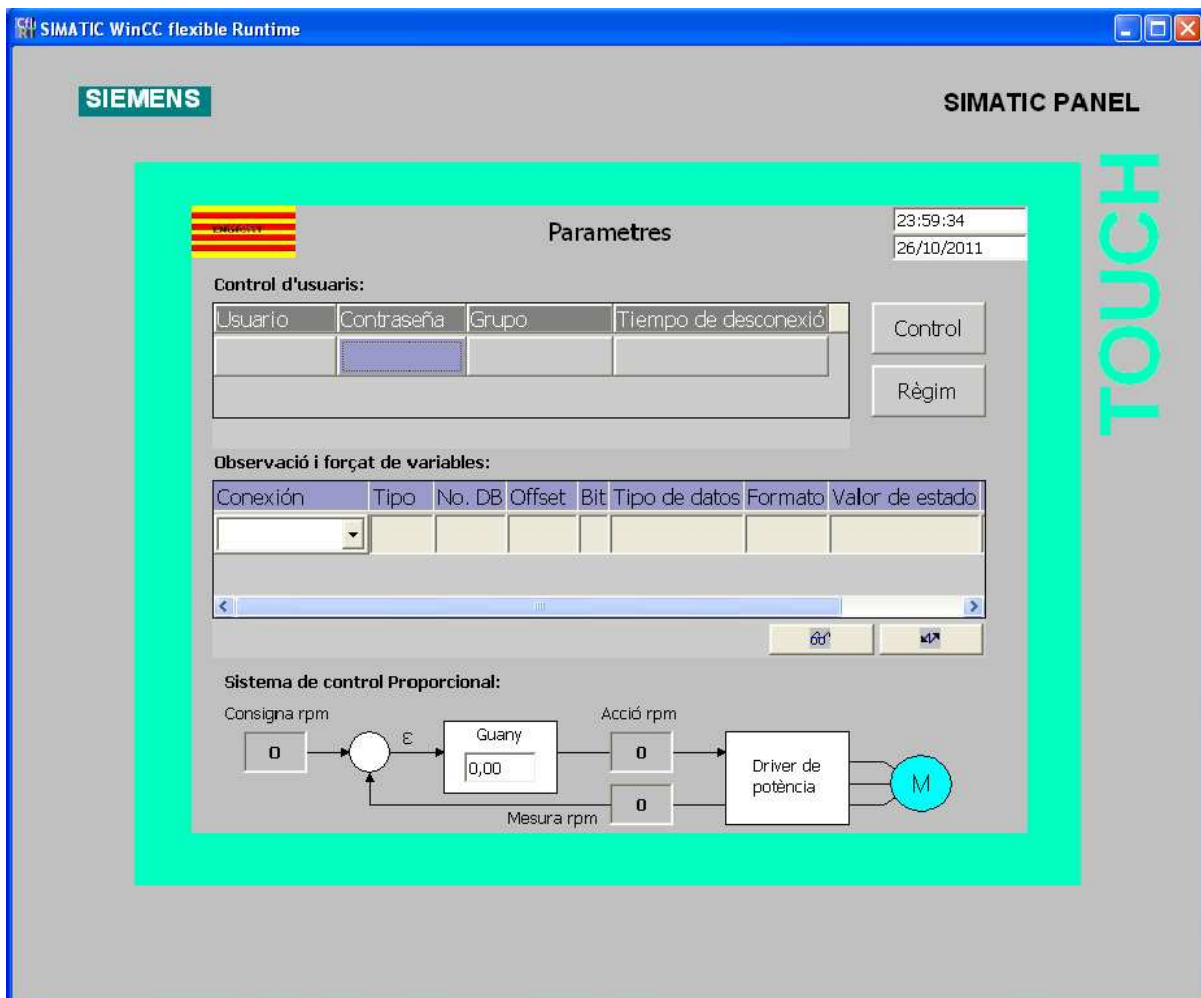
- Configure the **Intensitat** variable to cause an error event when the value exceeds **I\_Lim\_Max** for 60 seconds in a row.
- Configure the variable  $\%T_{MAX}$ , so that it triggers an error when the variable exceeds 120%.
- Assigns each variable to different alert groups.
- Configure the system so that the user can perform the recognition (ACK) of each group separately.
- Indicates in the process window that an alarm has occurred changing the color of the button. When the alarm goes off, it automatically turns gray again.
- The alarm window contains the alert view grid, and the buttons for making the VAC of the variable groups.
- Set the application template to show the triangle when an alarm is triggered.



**Paràmetres Windows**

This window contains three technical objects.

- User control settings are set from this window, but the object is only activated when the user has administrator permission.
- The State/Force grid provides access to process variables to make it easier to debug the PLC program.
- The window also incorporates a diagram of the proportional adjustment loop of the machine, and the possibility of adjusting the proportional gain of the system by using an E/S field. You can also see the setpoint, the regulation action on the power driver, and the speed measurement. (These E/S fields are configured as outputs)



**User Control**

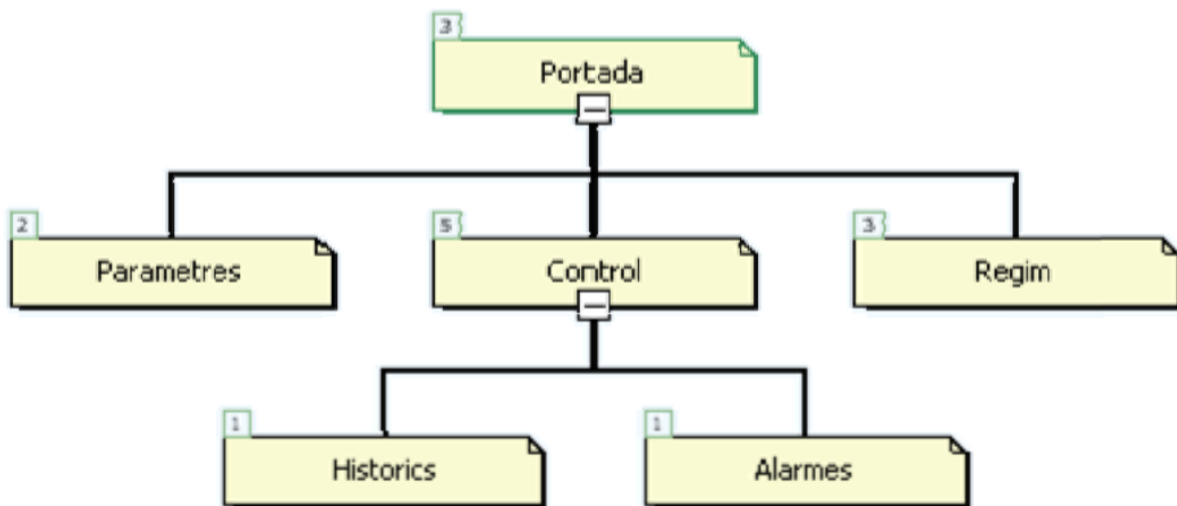
Don't configure this setting until you have verified all the operations.

Configure Login and Logout to request credentials when starting Runtime, log off due to inactivity, and be able to log out manually.

You can use the slider to do this.

**Browsing between windows and user access**

Window structure, which you can modify as you wish:



User-friendly operation options:

User	Windows Access	Observations
Operator	Control only, Alarms	
Head of plant	Historical, Control, Alarms	
Head of production	Only in Regime	
Maintenance	Only in Settings	You do not have access to config. users
Administrator	All	

Remember that although the Settings window can be accessed by the Maintenance window, it cannot operate the user management grid.



"The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein."