

**Treball Final de Carrera**

*Implantació d'un sistema SCADA en  
una cèl·lula flexible de fabricació*

Enric Rodà Sau

**Enginyeria en Organització Industrial**

Director: Juli Ordeix Rigo

Vic, juny de 2013

<b>Índex</b>	<b>Pàg.</b>
0.- RESUM -----	1
1.- INTRODUCCIÓ -----	3
1.1.- Motivació i abast d'aquest projecte -----	3
1.2.- Objectius i metodologia -----	3
2.- DESCRIPCIÓ DEL PROCÉS -----	12
3.- EL CONTROLADOR LÒGIC PROGRAMABLE (PLC) -----	19
3.1.- PLC SIEMENS S7-300 -----	19
3.2.- La programació de l'estació magatzem -----	22
3.2.1.- OB1- Principal -----	25
3.2.2.- FB2- Ventosa -----	25
3.2.3.- FB3- Ordre a SPC200 -----	26
3.2.4.- FB24- Magatzem recompte -----	27
3.2.5.- FB25- Magatzem buscar posició -----	30
3.2.6.- FB26- Magatzem escriu codi -----	31
3.2.7.- FB51- Identifica peça -----	31
3.2.8.- FC4- Eixos control -----	32
3.2.9.- FC8- Permisos -----	32
3.2.10.- FC11- Modes marxa -----	32
3.2.11.- FC13- Reposició -----	34
3.2.12.- FC18- Defectes -----	34
3.2.13.- FC19- Pilots -----	34
3.2.14.- FC21- Seqüència d'entrades -----	35
3.2.15.- FC22- Seqüència de sortides -----	36
3.2.16.- FC50- Desglossa codi -----	39
3.2.17.- FC103- Eixos posicions -----	39
3.2.18.- DB113- Existències magatzem -----	39
4.- EL SISTEMA SCADA -----	40
4.1.- Prestacions d'un sistema SCADA -----	42
4.2.- El sistema SCADA (VIJEO CITECT) -----	43
4.2.1.- Procés de creació d'aquest projecte amb el programa Vijeo Citect-----	52
4.2.2.- Definició de clústers i servidors -----	56

4.2.3.- Definició de privilegis, rols i usuaris -----	56
4.2.4.- Configuració del dispositiu de comunicació d'E/S -----	59
4.2.5.- Definició dels <i>tags</i> de variable d'aquest projecte -----	63
4.2.6.- La programació de les pantalles del procés -----	64
4.2.6.1.- Pantalla Magatzem -----	66
4.2.6.2.- Pantalla de comandes en procés -----	71
4.2.6.3.- Pantalla d'entrada de comandes -----	72
4.2.7.- Les funcions CIDODE -----	77
4.2.8.- Configuració de l'ordinador -----	83
5.- CREACIÓ DE LA BASE DE DADES DE COMANDES I INTERCANVI DE DADES ENTRE AQUESTA I EL VIJEO CITECT -----	89
5.1.- Creació de la base de dades -----	89
5.2.- L'intercanvi de dades entre el Vijeo Citect i l'Access -----	90
5.2.1.- Configuració d'un nou origen de dades en el PC -----	91
5.2.2.- Configuració de l'intercanvi de dades des del Vijeo Citect -----	93
6.- LA COMUNICACIÓ ENTRE PLC i PC VIA OPC -----	95
6.1.- La comunicació entre el PLC i el sistema SCADA (via OPC) ---	98
6.1.1.- Configuració d'un servidor OPC amb l'OPCEditor de - IBH OPC Server -----	100
7.- CONCLUSIONS -----	103
8.- BIBLIOGRAFIA -----	106
ANNEX I. <i>Tags</i> de variable definits en aquest projecte -----	107
ANNEX I.1.- <i>Tags</i> de variable de comunicació amb el PLC ----	108
ANNEX I.2.- <i>Tags</i> de variables locals -----	111
ANNEX II. Configuració IBH OPC Server -----	112
ANNEX III. Programa del PLC S7-300 de l'Estació Magatzem-----	132
ANNEX IV. Manual/guia per la docència del Vijeo Citect -----	184

## **Resum de Treball Final de Carrera Enginyeria en Organització Industrial**

**Títol:** Implantació d'un sistema SCADA en una cèl·lula flexible de fabricació

**Paraules clau:** Sistema SCADA, automatització, Vijeo Citect, monitorització, control, intercanvi dades Access-VijeoCitect

**Autor:** Enric Rodà Sau

**Direcció:** Juli Ordeix Rigo

**Data:** juny de 2013

### **Resum**

La Universitat de Vic disposa, entre altres equips, d'una cèl·lula flexible de fabricació, del fabricant Festo, que simula un procés de formació de palets amb els productes que es disposen en un magatzem intermedi. Aquesta cèl·lula està composta de quatre estacions de muntatge diferenciades (càrrega de palets, càrrega de plaques, magatzem intermedi i transport). Cada una disposa d'un PLC SIEMENS S7-300 per la seva automatització, i tots aquests es troben interconnectats amb una xarxa industrial Profibus.

L'objectiu d'aquest projecte és implantar el sistema SCADA Vijeo Citect pel control i supervisió de l'estació magatzem d'aquesta cèl·lula flexible de fabricació, establint també un intercanvi de dades entre l'SCADA i el Microsoft Access, per poder ser utilitzat per la docència.

Aquest projecte s'ha desenvolupat en cinc fases diferents:

1. La primera fase s'ha dedicat a l'automatització pròpiament de l'estació magatzem a partir de l'autòmat programable Siemens S7-300 i complint amb les necessitats plantejades.
2. En la segona fase s'ha programat i establert la comunicació per l'intercanvi de dades (lectura i escriptura) entre el sistema SCADA Vijeo Citect i la base de dades de Microsoft Access.
3. En la tercera fase s'ha elaborat i programat l'entorn gràfic de supervisió i control del procés a partir del sistema SCADA Vijeo Citect.
4. En la quarta fase s'ha instal·lat un OPC Server en el PC i s'ha establert la comunicació entre el PLC i el sistema SCADA.
5. Finalment s'ha anat revisant i depurant les diferents programacions i comunicacions per tal de que el sistema funcioni com a un conjunt.

## Summary of Final Year Project Work Engineering in Industrial Organization

**Title:** Implementation of a SCADA system in a flexible manufacturing cell.

**Key words:** SCADA System, automation, VijeoCitect, monitoring, control, exchange of data Access-VijeoCitect

**Author:** Enric Rodà Sau

**Tutor:::** Juli Ordeix Rigo

**Date:** June 2013

### Summary

Vic's University offers, among other engineering educational modules, a flexible manufacturing cell, from the manufacturing firm Festo, which simulates the process of stacking pallets with products available from a warehouse. This cell is composed of four different work stations (pallet loading, plaque loading, temporary storage and then transport). Each one of these stations has a PLC SIEMENS S7-300 for its automation and all these systems are interconnected with an industrial network Profibus.

The goal of this project is to implement the VijeoCitect SCADA system for controlling and monitoring the storage station of this flexible manufacturing cell, establishing a data exchange between the Vijeo Citect SCADA system and Microsoft Access to be used as a training tool for students.

This project was developed in five different phases:

1. The first phase was dedicated to the automation of the storage station which involved the Siemens S7-300 programmable logic controller.
2. The second phase involved the programming and establishment of the communicational interchange of data (read and write) between the VijeoCitect SCADA system and a Microsoft Access database.
3. The third phase involved the programming and design of the user interface and graphic display for the VijeoCitect SCADA system.
4. The fourth phase involved the installation of an OPC Server on a PC to establish the communication between the PLC and the SCADA system.
5. The final phase was to revise and debug different programs and communicational systems in order that all the systems worked correctly together.

## **1.- INTRODUCCIÓ**

### **1.1.- Motivació i abast d'aquest projecte**

L'automatització ocupa cada vegada una posició més preferent en el món de l'empresa i dels processos industrials. En l'automatització de processos industrials, els Controladors Lògics Programables (PLC) són els dispositius preferentment utilitzats i continuen evolucionant a mesura que les noves tecnologies es van afegint a les seves capacitats.

Superat el primer pas de l'automatització del procés, és important disposar d'una adequada interfície home-màquina per tal de dur-ne a terme una monitorització, supervisió i control. Per aquest fet, un programa SCADA (Supervisory Control And Data Acquisition, o Sistema de Control i Adquisició de Dades) és un software d'aplicació industrial orientat a la visualització de processos i control de sistemes remots. Entre les seves funcions es troben la monitorització de processos, adquisició de dades, visualització d'estats i execució d'ordres d'una planta de producció, i supervisió del funcionament.

Aquests programes, a més, pretenen millorar i simplificar la comunicació entre l'operador i el procés industrial, de forma que no sigui necessària una formació molt especialitzada per l'operari que ha de manejar l'aplicació (concepte HMI: Human Machine Interface, Interfície Home-Màquina).

La Universitat de Vic disposa, entre altres equips, d'una cèl·lula flexible de fabricació, del fabricant Festo, que simula un procés de formació de palets a partir dels productes que es disposen en un magatzem intermedi. Aquesta cèl·lula està composta de quatre estacions de muntatge diferenciades (càrrega de palets, càrrega de plaques, magatzem intermedi i transport) que poden funcionar cada una d'elles separadament, o bé es pot automatitzar i fer funcionar totes les estacions com a un conjunt. Cada estació disposa d'un PLC SIEMENS S7-300 per la seva automatització, i tots ells es troben interconnectats amb una xarxa industrial Profibus.

El Departament de Tecnologies Digitals de la Universitat de Vic ha adquirit el sistema SCADA Vijeo Citect, i vol utilitzar aquest software en la docència per ampliar els coneixements dels alumnes en el camp de l'automatització, control i supervisió d'un procés industrial.

L'objectiu principal d'aquest projecte és aprendre el funcionament del sistema SCADA Vijeo Citect i elaborar un manual/guia per la seva docència. Per aquest fet, d'una banda s'ha implantat aquest sistema SCADA Vijeo Citect pel control i supervisió de l'estació magatzem de la cèl·lula flexible de fabricació (per tal de disposar d'una aplicació d'un procés real), i per altra banda s'ha elaborat un manual/guia per la docència dels alumnes en el funcionament del Vijeo Citect (veure annex IV).

## 1.2.- Objectius i metodologia

La cèl·lula flexible de fabricació disposa de quatre estacions diferents controlades cada una d'elles pel seu propi PLC SIEMENS S7-300 (veure figures 1 i 2). Aquestes estacions funcionen cada una d'elles autònomament amb el seu propi programa del PLC, però també estan interconnectades entre elles a través d'una xarxa industrial per tal d'intercanviar-se informació i funcionar com un conjunt.

Tal i com es veu en la figura 1, el layout de la cèl·lula flexible de fabricació està distribuït al voltant de les cintes que formen part de l'estació transport. Aquesta estació disposa de quatre cintes (dues de llargues i dues de més curtes) que es troben disposades formant un rectangle.

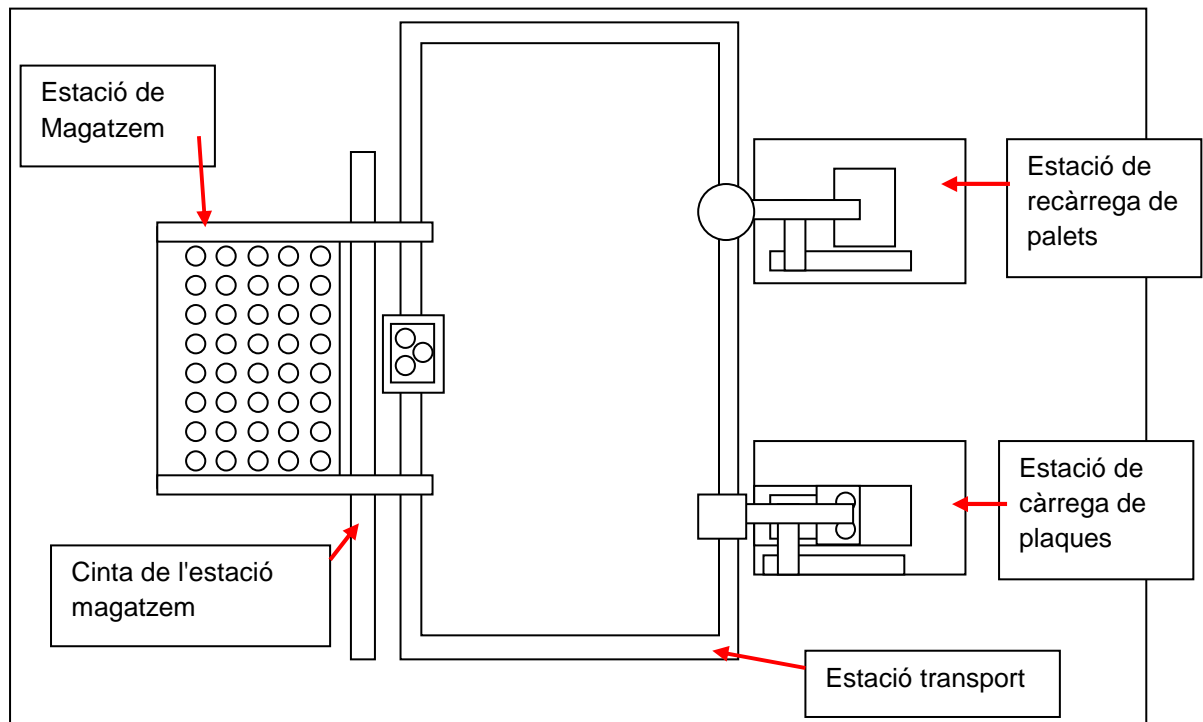


Figura 1. Esquema de les quatre estacions de la cèl·lula flexible de fabricació

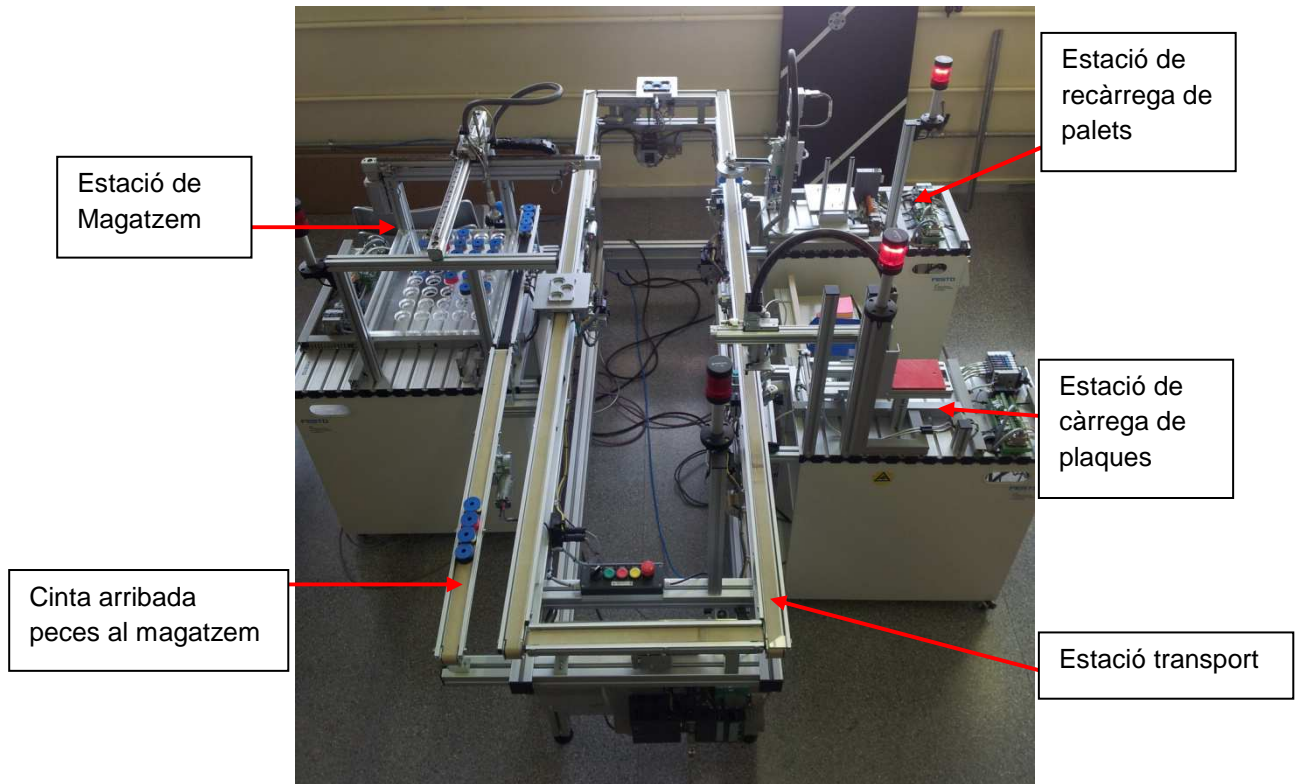


Figura 2. Les quatre estacions de la cèl·lula flexible de fabricació

La finalitat d'aquesta cèl·lula flexible de fabricació és la formació de palets que continguin tres peces cada un (veure figura 3.) En l'apartat Estació Magatzem, de la pàgina 6 d'aquest mateix punt 1.2, es defineixen el tipus de peces que es manipulen en la cèl·lula flexible de fabricació. Aquestes peces s'extreuen del magatzem i es col·loquen sobre una placa que disposa de tres orificis (que prèviament haurà estat col·locada sobre el palet per l'estació corresponent).

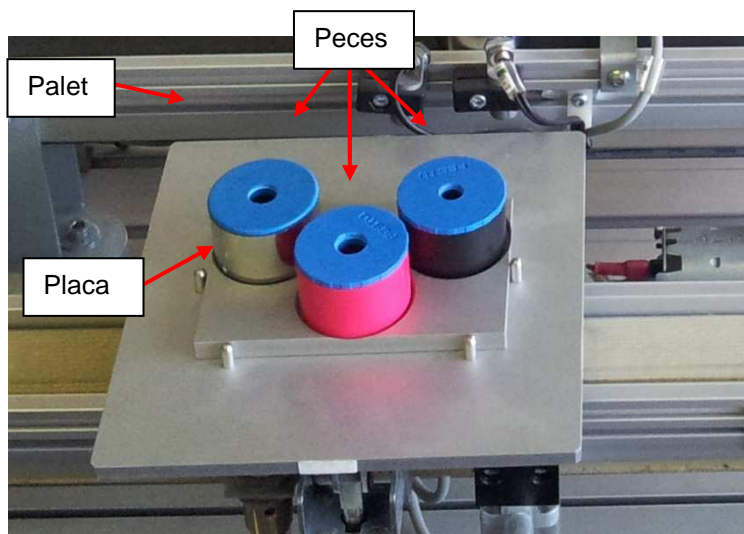


Figura 3. Detall d'una comanda completada



En els apartats següents es presenten les diferents estacions que conformen la cèl·lula flexible de fabricació:

a. Estació de transport

Aquesta estació uneix els processos de les diferents estacions de treball i dona com a resultat el procés conjunt de fabricació i control que anomenem Cèl·lula flexible de fabricació. Com es pot observar en la figura 4, l'estació de transport està formada per quatre cintes (dues de llargues i dues de més curtes) que estan unides formant un rectangle.

La cèl·lula flexible de fabricació disposa d'uns carros que van recorrent per sobre les cintes de l'estació transport cíclicament i, d'aquesta manera, van passant per cada una de les altres tres estacions per tal de que:

- l'estació de càrrega de palets col·loqui un palet damunt el carro,
- l'estació de càrrega de plaques col·loqui una placa sobre el palet,
- l'estació magatzem col·loqui les tres peces sol·licitades sobre la placa.

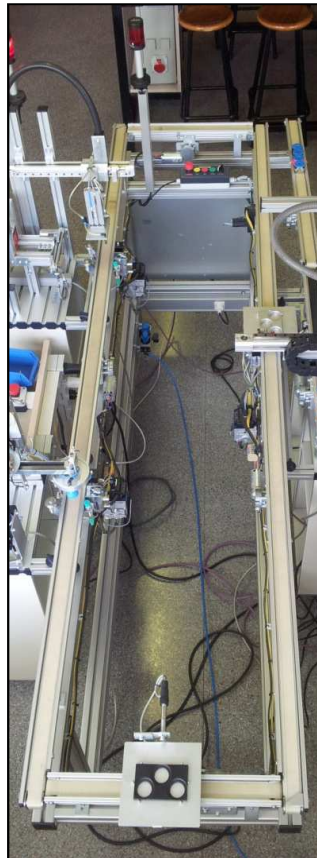


Figura 4. Estació transport

b. Estació de recàrrega de palets

Aquesta estació és un dispositiu alimentador que s'encarrega de proporcionar palets sobre els carros que circulen per la cinta transportadora del sistema (veure figura 5). Aquesta estació emmagatzema els palets en un carregador vertical i, mitjançant un manipulador amb ventoses, aquests són extrets per la part superior i es transfereixen a un dels carros de l'estació de transport quan es detecta presència d'aquest en la zona de càrrega.

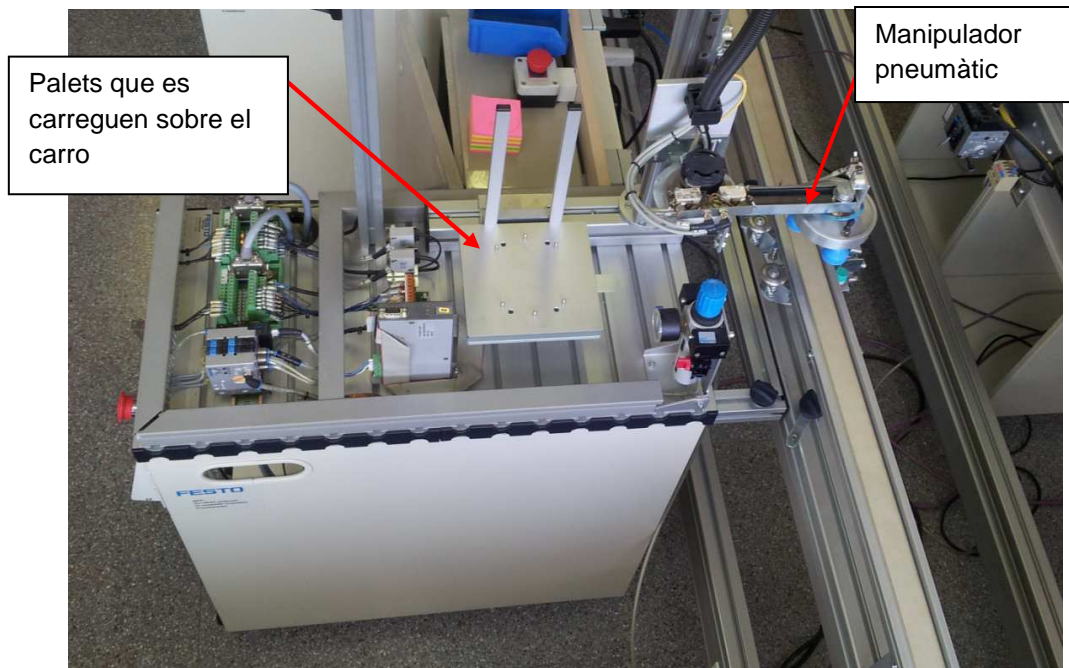


Figura 5. Estació de recàrrega de palets

c. Estació de càrrega de plaques

Aquesta estació també es tracta d'un dispositiu alimentador. Una vegada s'ha col·locat el palet sobre el carro de transport i aquest s'ha posicionat davant l'estació de càrrega de plaques, aquesta col·loca sobre el palet la placa on hauran d'anar disposades les peces que s'extreguin del magatzem (veure figura 6). Aquestes plaques poden ser negres, vermelles o metàl·liques i seran col·locades sobre el palet corresponent en funció del que estableixi l'ordre de la comanda.

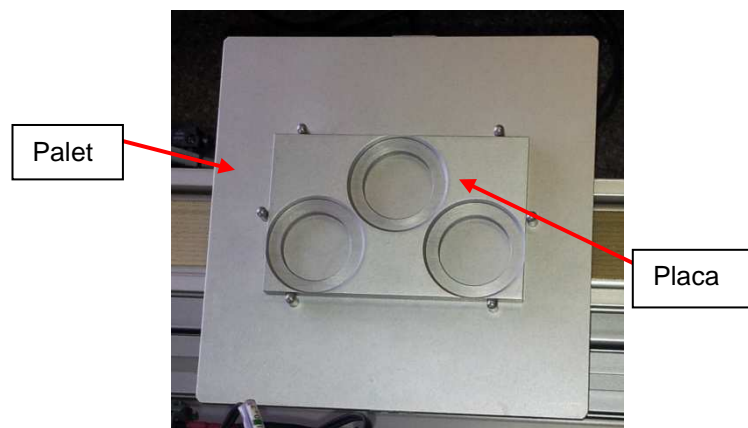


Figura 6. Detall d'una placa sobre un palet

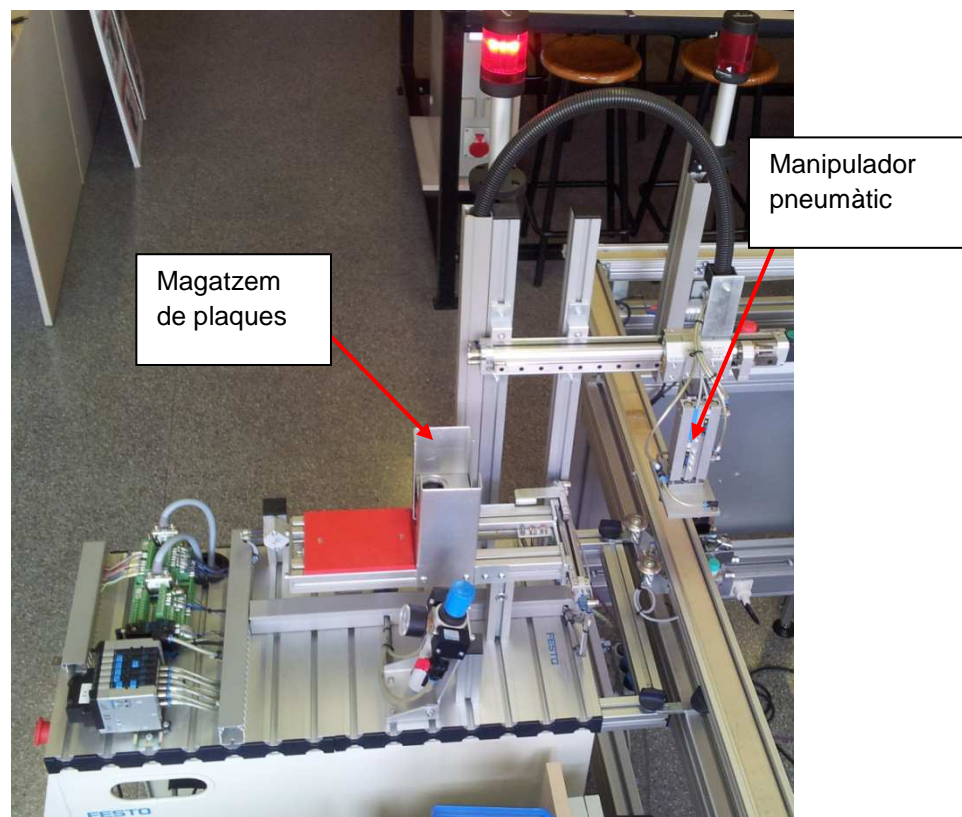


Figura 7. Estació de càrrega de plaques

d. Estació magatzem

L'objectiu d'aquest projecte és adequar el programa d'automatització de l'estació magatzem de la cèl·lula flexible de fabricació a les necessitats plantejades, i desenvolupar el programa de control i supervisió d'aquesta

estació a partir del sistema SCADA Vijeo Citect. Per tant, el desenvolupament d'aquest projecte es centra en l'estació magatzem.

S'ha escollit aquesta estació com a bon exemple d'aplicació industrial per a la docència dels alumnes per les possibilitats que ofereix tant a nivell d'automatització, com a nivell de supervisió i control.

Aquesta estació es tracta d'un magatzem automàtic de 40 posicions el qual es governa per mitjà de tres eixos, dos d'ells amb motors pas a pas i un tercer (de moviment vertical) amb un cilindre pneumàtic.

Aquest magatzem està dissenyat per emmagatzemar peces cilíndriques que simulen cilindres pneumàtics. Com s'observa en la figura 8, aquestes peces poden ser de tres tipus diferents:

- peces de plàstic de color negre
- peces de plàstic de color taronja
- peces metàl·liques

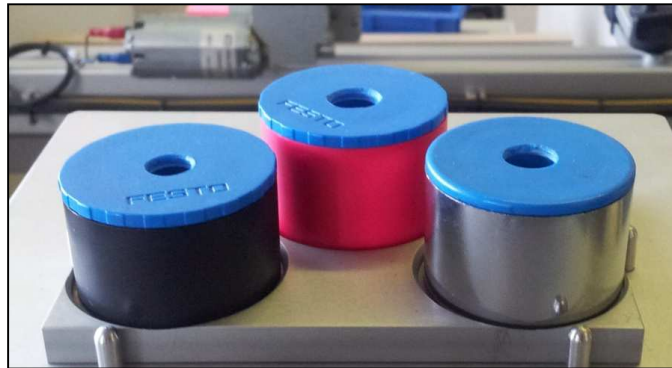


Figura 8. Detall del tipus de peça que es manipula

Aquesta estació disposa d'un quadre de comandament en el que s'hi ubiquen:

- Un polsador de "Marxa"
- Un posador de "Parada"
- Un polsador de "Reinici"
- Una parada d'emergència

L'estació magatzem té dos estats de funcionament:

- Manual. Per cada activació del polsador de "Marxa" es retira una peça del magatzem i es diposita en la zona de sortida (veure figura 9).
- Automàtic. Es retiren tres peces del magatzem, segons comanda, i es dipositen en la placa sobre la cinta de l'estació transport (veure figura 9).

L'estació magatzem té com a objectiu:

- Emmagatzemar les peces que provenen de la zona d'entrada de la cinta transportadora (veure figura 9). Aquesta cinta simula que prové de l'àrea de producció. Sempre que arriba una peça a la zona d'entrada per aquesta cinta, es recull per mitjà del manipulador amb ventosa i es diposita al magatzem (veure apartat 2.- *Descripció del procés* on es descriu com es fa la seva distribució).
- Subministrar tres peces en el palet situat a la cinta de l'estació de transport segons una comanda del client, quan l'estació es troba en estat automàtic, o bé, dipositar-les una en una (per cada pulsació del polsador de marxa) en la zona de sortida, quan l'estació es troba en estat manual (veure figura 9).

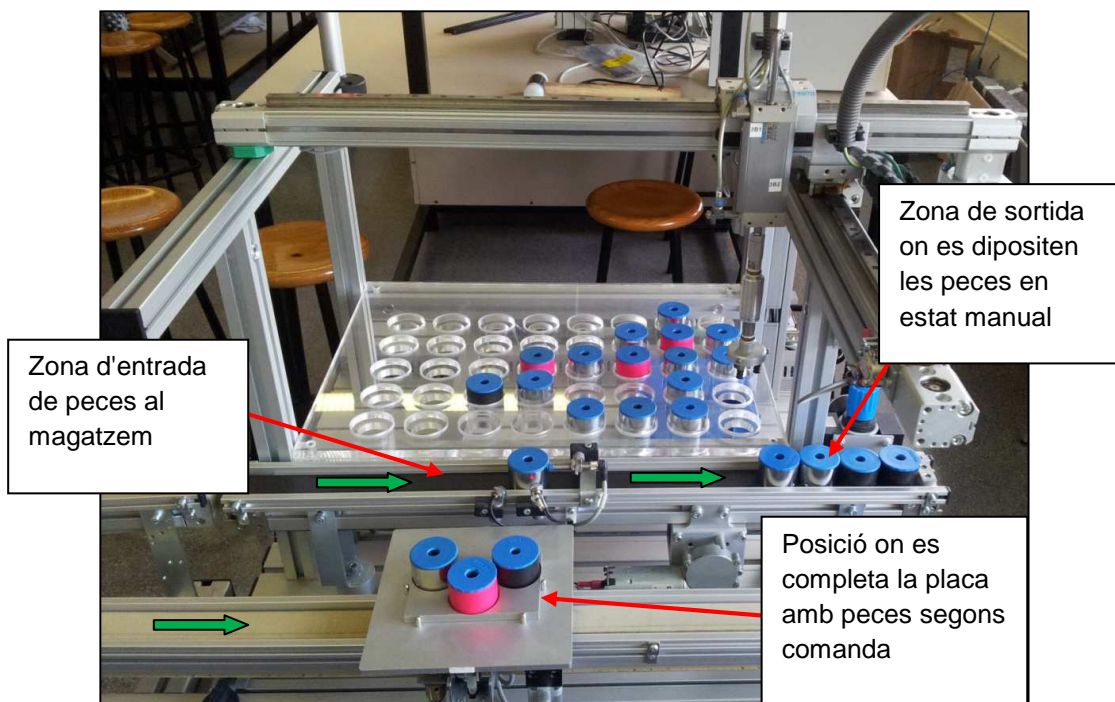


Figura 9. Estació magatzem

Un dels requeriments que formen part de les condicions d'aquest projecte és la necessitat de comunicar el sistema SCADA Vijeo Citect amb una base de dades (ACCESS) formada per dues taules: una d'elles on s'hi emmagatzemaran les comandes a realitzar pel magatzem i la segona on s'hi emmagatzemaran les comandes ja servides.

Aquest projecte s'ha desenvolupat en cinc fases diferents:

1. La primera fase s'ha dedicat a l'automatització pròpiament de l'estació magatzem a partir de l'autòmat programable Siemens S7-300 i de les necessitats plantejades.
2. En la segona fase s'ha programat i establert la comunicació per l'intercanvi de dades (lectura i escriptura) entre el sistema SCADA Vijeo Citect i la base de dades de Microsoft Access.
3. En la tercera fase s'ha elaborat i programat l'entorn gràfic de supervisió i control del procés a partir del sistema SCADA Vijeo Citect.
4. En la quarta fase s'ha instal·lat un OPC Server en el PC i s'ha establert la comunicació entre el PLC i el sistema SCADA.
5. Finalment s'ha anat revisant i depurant les diferents programacions i comunicacions per tal de que el sistema funcioni com a un conjunt.

Paral·lelament a l'elaboració d'aquest projecte s'ha confeccionat un manual/guia per al desenvolupament de la docència del programa Vijeo Citect. En aquest manual/guia s'han incorporat aspectes del Vijeo Citect que no utilitzats en aquest projecte però que s'ha cregut convenient introduir-les com a formació per als alumnes.

## 2.- DESCRIPCIÓ DEL PROCÉS

En aquest apartat es fa una descripció del funcionament de l'estació magatzem.

Aquest projecte s'ha centrat en l'estació magatzem de la cèl·lula flexible de fabricació, per tant, en tot moment aquesta funcionarà aïllada del conjunt. Tot i això, per tal d'establir comunicació amb el PLC SIEMENS S7-300 s'utilitzarà la xarxa Profibus que es troba instal·lada en la cèl·lula flexible de fabricació, i una comunicació via Ethernet amb el PC (veure apartat 6.- La comunicació entre PLC i PC via OPC).

L'estació magatzem té com a objectiu emmagatzemar les peces que li arriben per una cinta transportadora en la zona d'arribada i subministrar-les segons els requeriments de les comandes guardades. En la figura 10 es pot observar un esquema conceptual de l'estació magatzem i de la distribució de les 40 posicions d'emmagatzematge de peces.

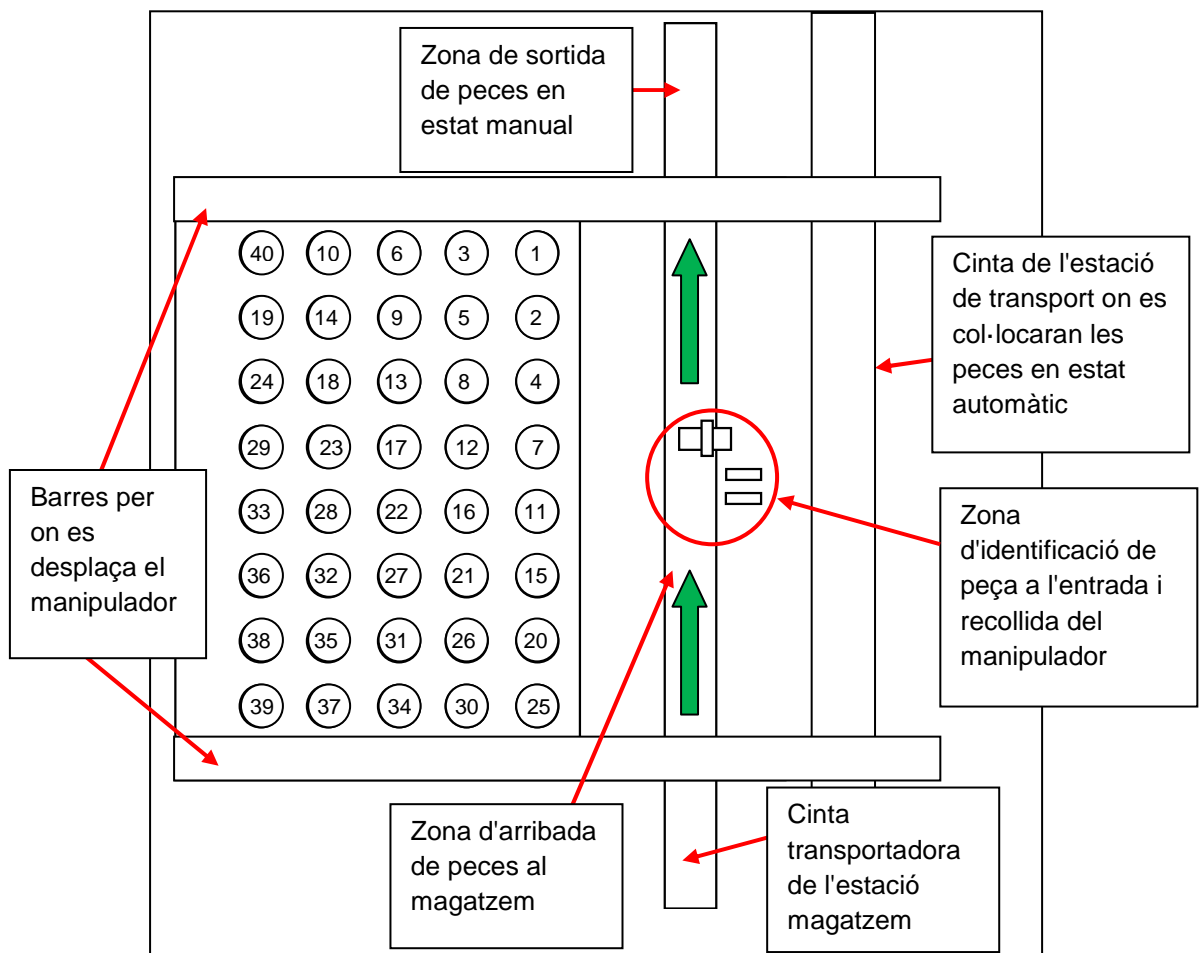


Figura 10. Esquema conceptual en vista de planta del magatzem

Les quaranta posicions del magatzem no disposen de cap tipus de sensor de presència ni identificació de peça, sinó que el contingut del magatzem es troba mapejat segons les entrades i sortides que s'han efectuat en una taula de memòria del PLC (veure apartat 3.2.18.- DB113- Existències magatzem). D'aquesta manera, si s'ha entrat una peça en el magatzem a través del manipulador i aquesta és retirada manualment, caldria anar a esborrar la posició de memòria corresponent per tal que correspongui amb l'ocupació real del magatzem. Igualment, si es col·loca manualment una peça en alguna posició del magatzem, caldrà actualitzar la posició de memòria corresponent a aquesta posició també manualment (en el funcionament normal de l'estació no caldrà fer aquestes operacions però cal tenir en compte que pel bon funcionament del sistema, no es poden manipular manualment les peces emmagatzemades).

Com es pot observar en la figura 10, les quaranta posicions del magatzem es troben numerades correlativament, començant pel vèrtex superior dret amb la posició 1 i avançant en diagonal fins a la posició 40. La numeració de les posicions del magatzem s'incrementa físicament en forma diagonal per tal d'afavorir la presència del manipulador prop de la zona de sortides i augmentar el rendiment en l'extracció del material. Tant les entrades com les sortides de les peces es fan de manera que sempre es comença a emplenar el magatzem per la posició més baixa que estigui lliure i es comença a buidar igualment començant per la posició més baixa ocupada.

Descripció del procés d'entrada de peça al magatzem:

- Quan arriba una peça per la cinta transportadora a la zona d'entrada, aquesta s'atura a la zona de recollida del manipulador (veure figura 10).
- En aquest punt, tal i com s'observa en la figura 11, es disposa de tres detectors per tal de poder fer la discriminació del tipus de peça de què es tracta: dos detectors òptics (alta i baixa sensibilitat) per la detecció de presència de peça i diferenciació de color (negre - no negre) i un detector de tipus inductiu per diferenciar les peces metàl·liques de les no metàl·liques.



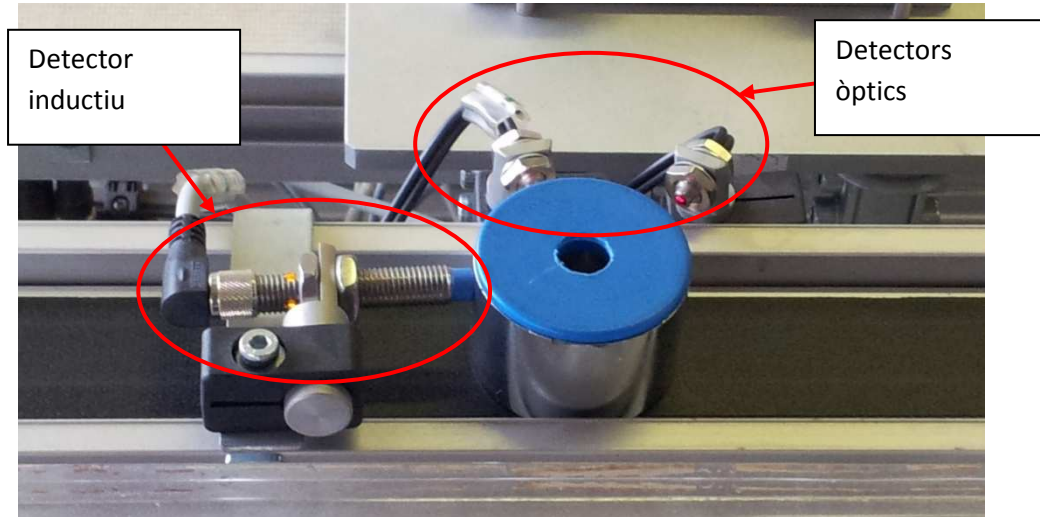


Figura 11. Detall de la zona de discriminació del tipus de peça

- Mitjançant la combinació de les tres senyals es distingeix el tipus de peça que arriba a la zona d'entrada (veure taula 1).

	Sensor òptic d'alta sensibilitat	Sensor òptic baixa sensibilitat	Sensor inductiu
Peça Negra	X		
Peça Taronja	X	X	
Peça Metà·lica	X	X	X

Taula 1.- Combinació de les senyals per la discriminació de peces

- Una vegada la peça està identificada, i suposant que no hi ha cap petició de sortida de peça en curs (les sortides són prioritàries davant les entrades), el sistema busca en la taula de memòria del PLC (veure apartat 3.2.18.- DB113- Existències magatzem) si hi ha una posició lliure en el magatzem, començant sempre pel nombre més baix de posició.
- Quan el sistema troba la primera posició lliure del magatzem (veure apartat 3.2.5.- FB25- Magatzem buscar posició), el manipulador va a recollir la peça a l'entrada i la diposita en aquesta posició. Com es pot observar en la figura 12, cada posició del magatzem ve referida per una fila i una columna per tal de ser identificades inequívocament.

- Una vegada efectuada la transacció, el sistema actualitza la taula interna de mapejat de posicions del magatzem amb un codi identificatiu de la peça entrada, de manera que:
  - Peça negra: valor numèric 2
  - Peça taronja: valor numèric 3
  - Peça metàl·lica: valor numèric 4
- En el cas de que no hi hagi cap posició lliure en el magatzem, el manipulador recull la peça de l'entrada i la diposita en la posició 40 (posició de rebuig). Aquesta posició es considera com a una rampa de sortida.

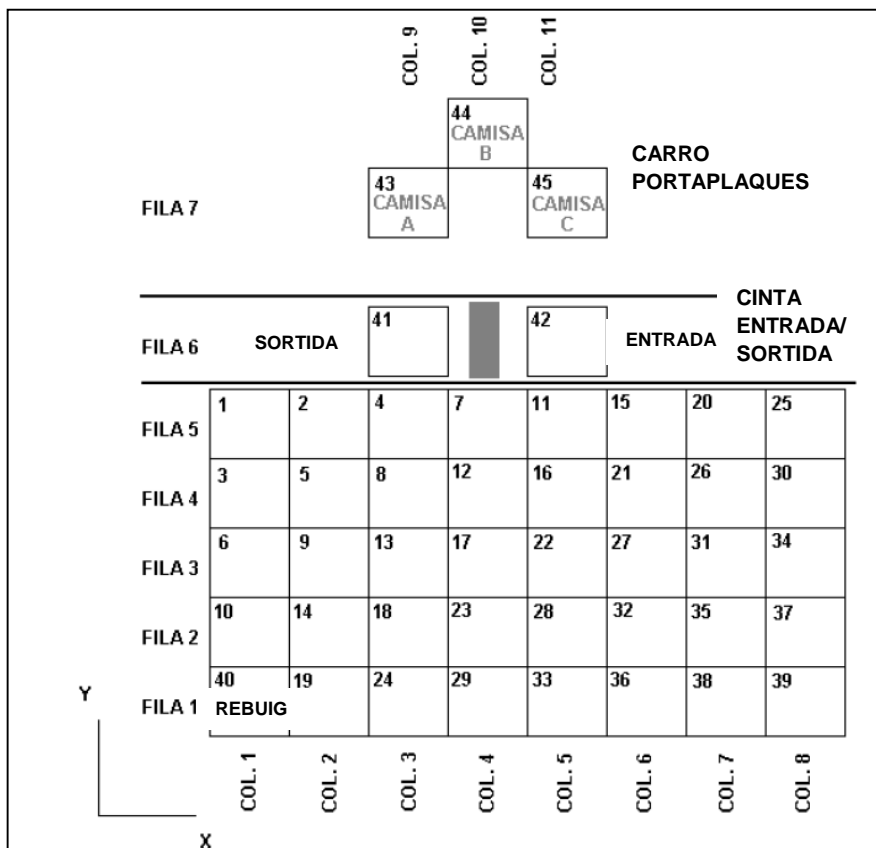


Figura 12. Distribució de files i columnes de l'estació magatzem

L'estació Magatzem pot funcionar en estat manual o en estat automàtic:

- Estat Manual. Quan l'estació es troba en estat manual, per cada acció sobre el polsador de marxa de l'estació, o bé, per cada acció sobre el botó corresponent de la pantalla del sistema SCADA (veure apartat

4.2.6.1.- Pantalla Magatzem), es recollirà una peça del magatzem i es dipositarà en la zona de la cinta que representa una sortida de recollida manual (veure figura 13). Tal i com s'ha comentat, l'ordre en què es retiren les peces és començant des de la posició més baixa ocupada del magatzem, seguida de la següent que es troba ocupada i així successivament. Si una posició no està ocupada, es salta i es continua per la següent.

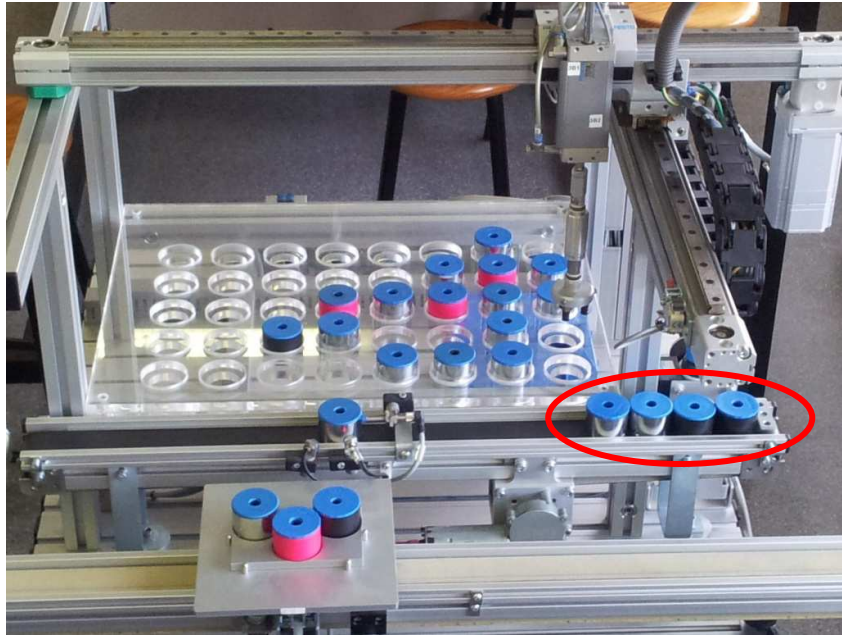


Figura 13. Detall de peces extretes en cicle manual a la cinta de sortida

- Estat automàtic. Quan l'estació es troba en estat automàtic, es retiraran tres peces del magatzem i es col·locaran en la placa que es troba sobre el palet en l'estació de transport. Per saber quines són les peces que cal dipositar en el palet, s'ha elaborat una base de dades amb el programa Access on es guarden les comandes que caldrà servir. Aquestes comandes es podran entrar des del mateix sistema SCADA Vijeo Citect (veure apartat 4.2.6.3.- Pantalla d'entrada de comandes).

Per executar una comanda, el sistema SCADA la llegeix de les que es disposen en la base de dades d'Access i la transfereix al PLC per tal de ser executada (veure apartat 5.2.- L'intercanvi de dades entre el sistema SCADA i l'ACCESS). Aquesta comanda consta d'un codi numèric (veure

figura 14) amb el qual el PLC sabrà quins són el tipus de peça que ha d'extreure en primer, segon i tercer lloc.

1	X1	X2	X3
X1= primera peça de la comanda X2= segona peça de la comanda X3= tercera peça de la comanda X1, X2, X3 poden agafar els valors 2 (negre), 3 (taronja) o 4 (metàl·lica) segons la codificació comentada anteriorment			

Figura 14. Codi numèric que conforma una comanda

El número 1 que es troba al inici de la cadena de la comanda es descarta (aquest número s'utilitzaria si l'estació treballés en conjunt amb la cèl·lula flexible de fabricació i serviria per identificar el color de la placa).

Mentre l'estació es troba en estat automàtic es van executant per ordre correlatiu totes les comandes de la base de dades. Si s'arriba a la última posició de la taula i no hi ha més comandes, l'estació es queda a l'espera que l'usuari entri una nova comanda.

Les peces que es lliuren a la sortida són extreïdes del magatzem seguint l'ordre de la posició que ocupen, de manera que si la comanda és d'una peça de cada color, el sistema extraurà la que es troba en la posició més baixa del magatzem de cada color.

En cas que no es disposi d'una peça demandada en una comanda:

- els eixos es dirigiran a la zona d'arribada de peça (veure figura 10) a l'espera que n'entri una;
- si la peça que entra és del color desitjat segons la comanda, el sistema la recull, la col·loca en la posició corresponent del palet i continua amb la comanda;
- si la peça que entra no és del color requerit, el sistema la recull, la col·loca en la posició que li toqui del magatzem i torna a la posició d'entrada a l'espera d'una altra peça.

Una vegada la comanda ha estat executada, i com que l'estació treballa aïllada del conjunt de la cèl·lula flexible, caldrà retirar el palet

manualment i confirmar que es torna a disposar d'un palet buit per tal de transferir al PLC la següent ordre de comanda.

Una vegada la comanda ha estat executada, aquesta es guarda en una altra taula de la mateixa base de dades per tal de tenir constància de les comandes servides.

Des del sistema SCADA no es pot fer el canvi de manual a automàtic ni al revés ja que en el quadre de comandament de l'estació es disposa d'un commutador de clau que podria quedar falsejat per l'ordre donada des del PC.

Tot aquest sistema està monitoritzat i controlat pel sistema SCADA Vijeo Citect. Des de la pantalla de l'ordinador, s'han dissenyat una sèrie de pantalles gràfiques per tal de seguir en tot moment el desenvolupament del procés (veure apartat 4.2. El sistema SCADA Vijeo Citect). Es té constància en tot moment i en temps real del mapa de posicions del magatzem i de l'ocupació d'aquest (amb un recompte específic del nombre de peces de cada color, així com del total de peces i del % d'ocupació).

### **3.- EL CONTROLADOR LÒGIC PROGRAMABLE (PLC)**

En aquest apartat es fa una descripció de l'autòmat programable utilitzat per l'automatització de l'estació magatzem així com del programa que s'ha elaborat.

#### **3.1.- PLC SIEMENS S7-300**

Per tal de dur a terme l'automatització d'aquesta estació magatzem s'ha utilitzat un Controlador Lògic Programable (PLC) Siemens S7-300.

El SIMATIC S7-300 és el controlador més venut de la plataforma *Totally Integrated Automation* i compta amb nombroses aplicacions de referència satisfactòries en els sectors industrials més diversos en tot el món. Per aquest fet, també es creu molt oportú introduir a l'alumne en aquest fabricant i gamma de producte.

Aquest PLC està concebut per tal de donar solucions de sistema innovadores amb especial èmfasi en tecnologia de fabricació, i com a sistema d'automatització universal, constitueix una solució òptima per aplicacions en estructures centralitzades i descentralitzades. Disposa d'una configuració modular, de manera que no hi ha cap regla d'assignació dels slots.

El Simatic S7 disposa d'una sèrie de mòduls que divideixen la memòria de programa i la de dades en seccions, permetent una programació estructurada i un accés ordenat a les dades. El nombre de mòduls dependrà del tipus de CPU utilitzada, i en el cas que ens ocupa s'utilitzaran els següents:

➤ *Mòduls d'organització i estructura del programa (OB. Organization Block)*

Els blocs d'organització constitueixen la interfície entre el sistema operatiu i el programa de l'usuari. Són cridats pel sistema operatiu i controlen el processament cíclic i controlat per alarmes del programa, el comportament d'arrencada del sistema d'automatització i el tractament dels errors. Programant els blocs d'organització es defineix el comportament de la CPU.

Els blocs d'organització determinen la seqüència en la que s'hauran d'executar les diferents parts del programa.

L'execució cíclica de programes és l'execució normal en els sistemes d'automatització. El sistema operatiu crida cíclicament a l'OB1 i arrenca l'execució cíclica del programa de l'usuari. A partir d'aquí es poden fer les crides a altres blocs de programa. Cada cop que es fa una crida, el programa va al bloc cridat, l'executa, i torna al punt d'origen de la crida.

En aquest projecte, existeixen 3 tipus d'OB:

- OB1 (cicle lliure): és el mòdul principal, el que s'executa cíclicament i del que parteixen tots els salts a altres mòduls.
- OB d'error i alarma (81, 82 i 86): són els que contenen la seqüència d'accions a realitzar en cas de que es produeixi una alarma o error programat.
- OB d'arranc (100): en aquest mòdul podem introduir valors per defecte que permeten l'arranc definit a la instal·lació, ja sigui en un arranc inicial o després d'una fallada en l'alimentació.

➤ Mòduls de codi (FC. Function Configurable)

Són mòduls en els que podem incloure part del programa de l'usuari de manera que podrem obtenir un programa molt més estructurat. A aquests mòduls s'hi pot accedir des d'un altre mòdul FC o des d'un mòdul OB.

Aquestes funcions són mòduls totalment programables. Una funció és un bloc lògic sense memòria. Les variables temporals de les FCs es memoritzen en la pila de dades locals. Aquestes dades es perden quan finalitza el tractament de les FCs. La FC conté un programa que s'executa cada vegada que la FC és cridada per un altre bloc lògic. Les funcions es poden utilitzar per:

- retornar un valor de funció al bloc invocant (ex: funcions matemàtiques)
- executar una funció tecnològica (ex: control individual amb combinació binària).

➤ Mòduls de funcions (FB. Function Block)

Són mòduls de programa especials. Aquí s'introdueixen les parts de programa que apareixen amb freqüència o que posseeixen una gran complexitat i serveixen per simplificar la programació d'aquestes funcions. Disposen d'una zona de memòria assignada per tal de guardar variables (mòduls de dades d'instància). El que es fa és enviar paràmetres al FB i guardar algunes de les dades locals resultants en el mòdul de dades d'instància.

➤ Mòduls de dades (DB. Data Block)

Són àrees de memòria destinades a contenir dades del programa de l'usuari. Existeixen mòduls de dades globals i d'instància. A les dades contingudes en un mòdul de dades és possible accedir-hi de forma absoluta o de forma simbòlica. Les dades complexes o compostes es poden dipositar en forma d'estructura.

- Mòduls de dades globals: es poden utilitzar per qualsevol mòdul del programa.
- Mòduls de dades d'instància: s'assignen a un determinat mòdul de funció i només es poden utilitzar des d'aquest mòdul. Poden assignar-se varis mòduls de dades d'instància a un mòdul de funció.

Per la programació del PLC, el fabricant ofereix tres tipus de llenguatge:

- KOP (esquema de contactes). És un llenguatge de programació gràfic. La sintaxis de les instruccions és similar a la d'un esquema de circuits. KOP permet observar la circulació del corrent a través dels contactes, elements complexos i bobines.
- AWL (llista d'instruccions). És un llenguatge de programació textual orientat a la màquina. En un programa creat en AWL, les instruccions equivalen en gran mesura als passos amb els que la CPU executa el programa. Per tal de facilitar la programació, AWL s'ha ampliat amb estructures de llenguatge d'alt nivell (tals com accessos estructurats a



dades i paràmetres de blocs). Aquest és el tipus de llenguatge de programació que s'ha utilitzat en aquest projecte.

- FUP (diagrama de funcions). És un llenguatge de programació gràfic que utilitza els quadres de l'àlgebra booleana per tal de representar la lògica. Igualment, permet representar funcions complexes (funcions matemàtiques) mitjançant quadres lògics.

### 3.2.- La programació de l'estació magatzem

El software utilitzat per la programació del PLC ha estat el Siemens STEP 7 v5.2. El programa de control ha estat creat de forma modular i en llenguatge AWL per tal de poder fer més fàcil tant el diagnòstic de potencials averies com la seva posterior revisió i/o modificació.

Tal i com s'ha comentat, l'estació utilitzada per aquest projecte forma part d'una cèl·lula flexible de fabricació que funciona amb les 4 estacions connectades amb una xarxa Profibus. En la figura 15 es mostra la pantalla de configuració de la xarxa formada per les estacions de la cèl·lula flexible de fabricació en el software SIMATIC S7-300. En aquest cas, s'ha utilitzat tan sols l'estació de magatzem intermedi per tal d'implantar-hi el sistema SCADA Vijeo Citect.

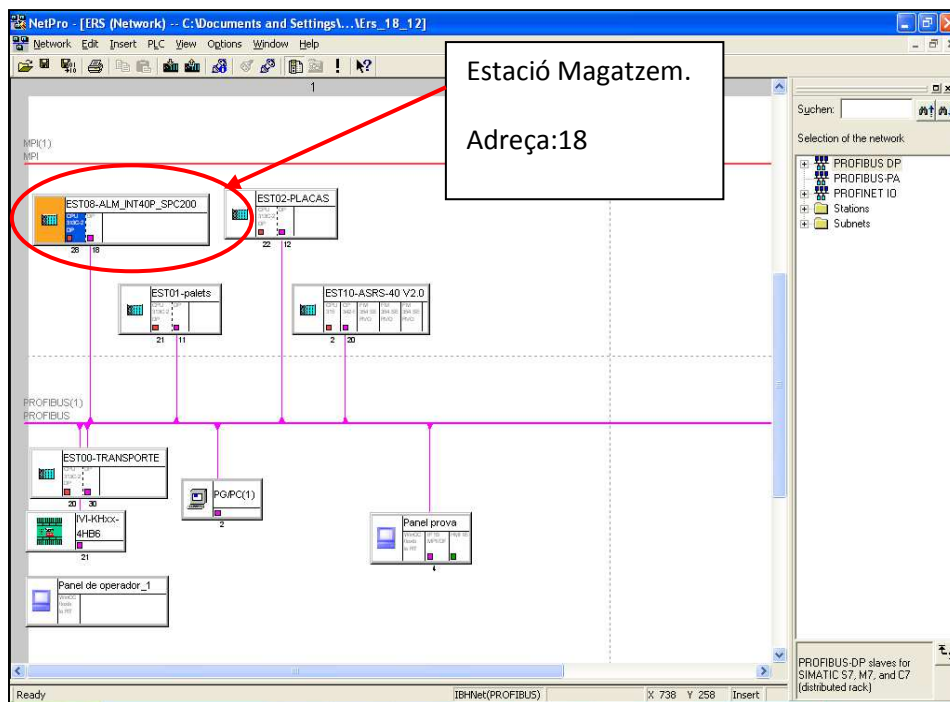


Figura 15. Pantalla de configuració de la xarxa de la cèl·lula flexible de fabricació

A continuació es mostren els elements que formen el hardware del PLC SIEMENS S7-300 de l'estació magatzem utilitzat:

- CPU 313-2DP, referència: 6ES7 313-6CE01-0AB0. CPU compacta amb MPI (Multi Point Interface), 16 Entrades Digitals/16 Sortides Digitals, 3 comptadors ràpids (30kHz), línia PROFIBUS-DP integrat, amb font d'alimentació integrada de 24V DC, memòria de 32 kbytes.
- Mòdul digital SM 323, referència: 6ES7 323-1BH00-0AA0. Amb aïllament òptic, 8 Entrades Digitals/8 Sortides Digitals, 24V DC, 0,5 Ampers, corrent agregada de 4 ampers, PIN 1x40.

En la figura 16 es pot observar la pantalla de configuració del hardware que ens ofereix el software SIMATIC S7-300.

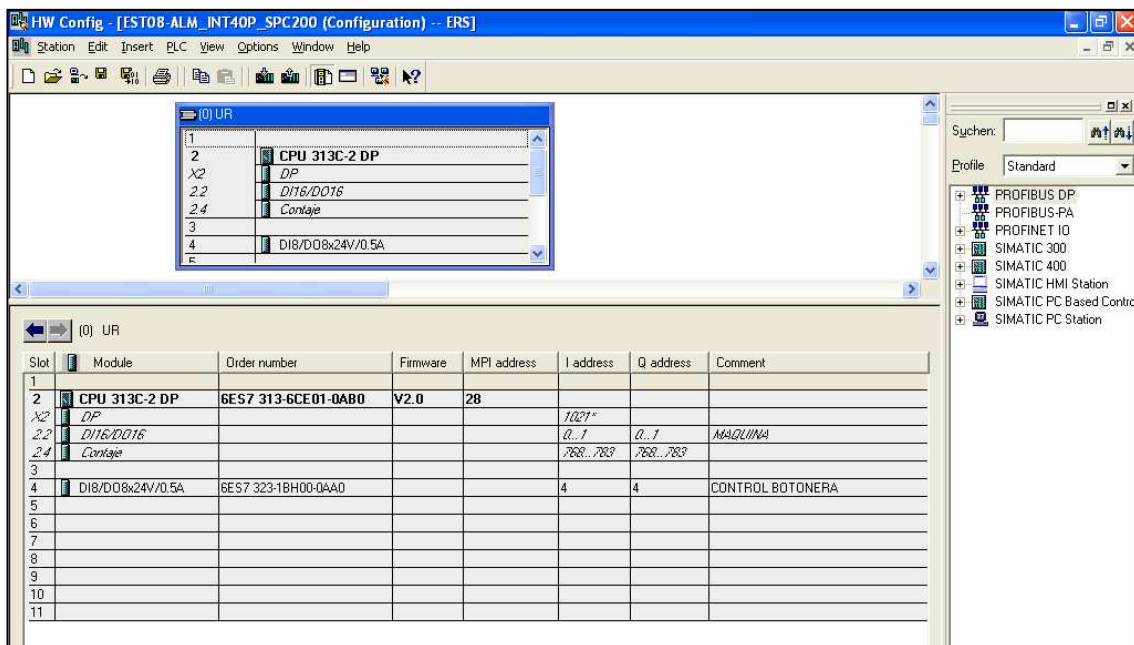


Figura 16. Pantalla de configuració del hardware del PLC

Per tal de desenvolupar l'automatització de l'estació magatzem s'ha utilitzat com a base el programa original desenvolupat per FESTO, i el qual es trobava instal·lat en el PLC i feia funcionar l'estació juntament amb el conjunt de tota la cèl·lula flexible de fabricació. Partint d'aquest programa, i com que es tracta d'un tipus de programació elaborada de forma modular, s'han modificat els mòduls oportuns per tal de complir amb els objectius de funcionament d'aquest projecte. En la Taula 2 es llisten els mòduls programats per dur a terme

l'automatització de l'estació magatzem. La programació d'aquests mòduls es descriu a partir del punt 3.2.1 i s'adjunten tots ells en l'annex III.

MÒDUL	NOM	DESCRIPCIÓ
OB1	Principal	Coordina la resta de programes
FB2	Ventosa	Control de la seqüència de la ventosa
FB3	Ordre a SPC200	Diàleg amb el SPC200
FB24	Mag_Recompte	Compte posicions del magatzem
FB25	Mag_Buscar_Pos	Busca posició amb un codi específic
FB26	Mag_Escriu_Codi	Esriptura de posició amb un codi especificat
FB51	Identifica_Peça	Identificació del tipus de peça
FC4	Eixos_Control	Control dels dos eixos
FC8	Permisos	Es defineixen les condicions de treball de l'estació així com els permisos de moviment de determinats elements
FC11	Modes Marxa	Es defineixen els modes de funcionament en funció de l'estat dels elements del panell de control
FC13	Reposició	Seqüència de posada a zero després d'una parada d'emergència o de donar tensió
FC18	Defectes	Monitoritza els defectes de l'estació i activa les alarmes corresponents
FC19	Pilots	Control dels pilots de l'estació
FC21	Seq_Entrades	Control de la seqüència d'entrada de peces
FC22	Seq_Sortides	Control de la seqüència de sortida de peces
FC50	Desglosa_Codi	Desglossa el tipus de peces d'una comanda
FC103	Eixos_Posicions	Desglossa una posició en fila + columna
DB113	Existencies_Mag	Ocupació del magatzem

Taula 2. Mòduls de programació del PLC utilitzats

En la figura 17 es mostra un esquema amb la seqüència de crides dels blocs programats que es fa en cada cicle de treball del PLC.

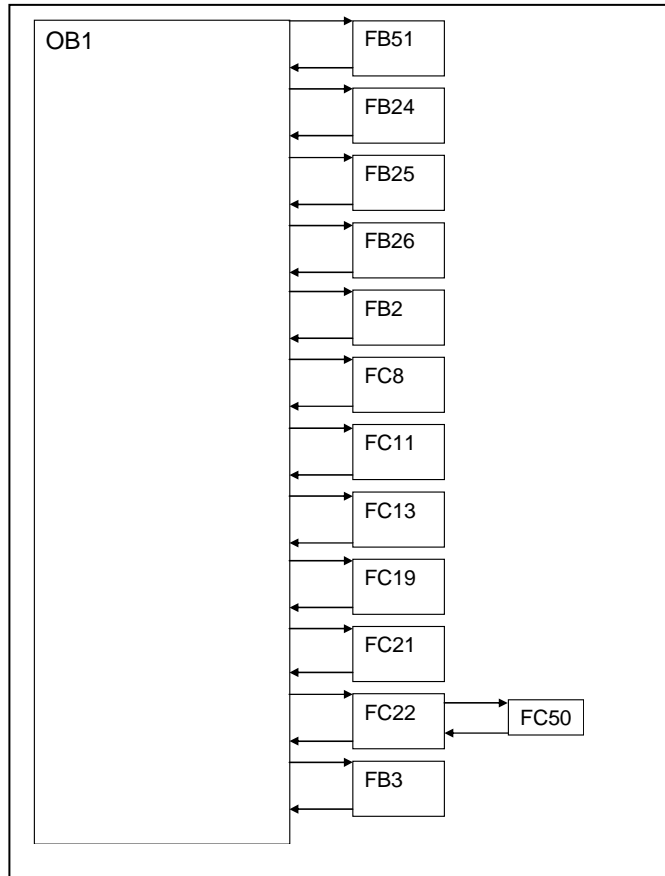


Figura 17. Seqüència d'execució del programa del PLC

### 3.2.1- OB1- Principal

Aquest mòdul és executat cíclicament per la CPU de l'autòmat i des d'aquest es fan les crides a la resta de mòduls de programació de manera seqüencial per tal de poder executar tot el programa d'una manera ordenada.

### 3.2.2.- FB2- Ventosa

Rutina de control de la ventosa. Aquesta rutina és cridada des d'altres mòduls de control de programa sempre que sigui necessari fer una acció de recollir o deixar peça. Una vegada es fa la crida a aquest mòdul, es passa a fer una inicialització de la seqüència programada.

El mòdul que fa la crida a aquesta rutina cal que:

- disposi de la necessària instrucció de crida,
- activi el bit amb l'etiqueta "ActivarVentosa", per tal d'inicialitzar-lo,

- posi el bit amb l'etiqueta "Mode" a 0 si es desitja recollir una peça o a 1 si el que es vol és deixar una peça.

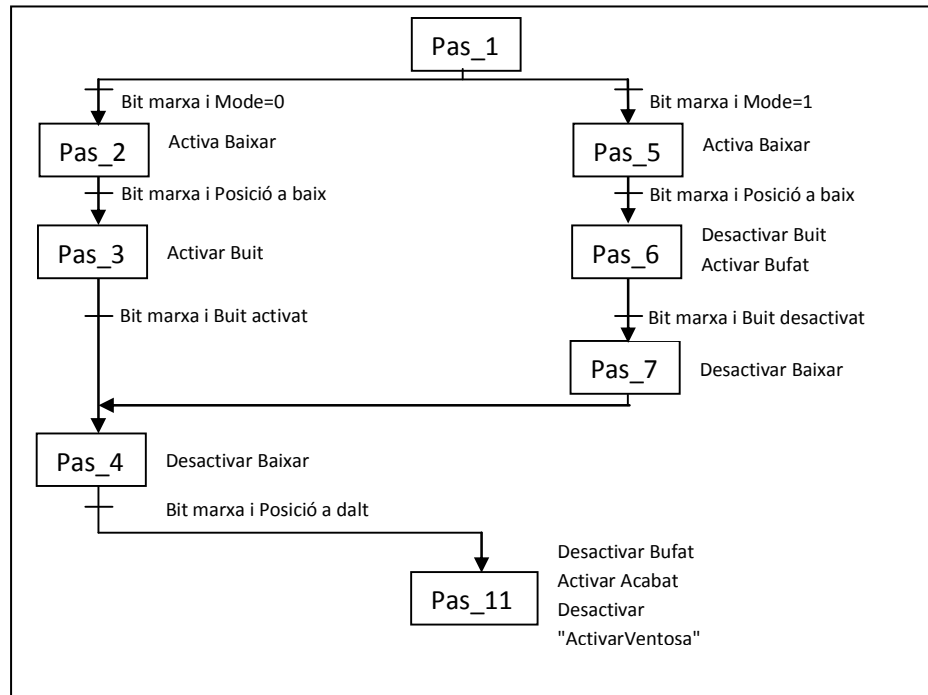


Figura 18. Diagrama de flux corresponent a la programació del mòdul FB2.

Tal i com es pot observar en la figura 18, una vegada s'entra en aquest bloc de funció hi ha dues opcions:

- Recollir peça. En aquest cas es segueix la seqüència de la branca esquerra del diagrama de flux.
- Deixar peça. En aquest cas es segueix la seqüència de la branca dreta del diagrama de flux.

### 3.2.3.- FB3- Ordre a SPC200

Rutina de diàleg amb el SPC200. S'encarrega de l'enviament d'ordres des del PLC al controlador SPC200 (senyals ACK\_x i CLK\_x). Per al desenvolupament d'aquest bloc s'ha seguit el diagrama de diàleg proposat pel fabricant FESTO i que es pot observar a la figura 19.

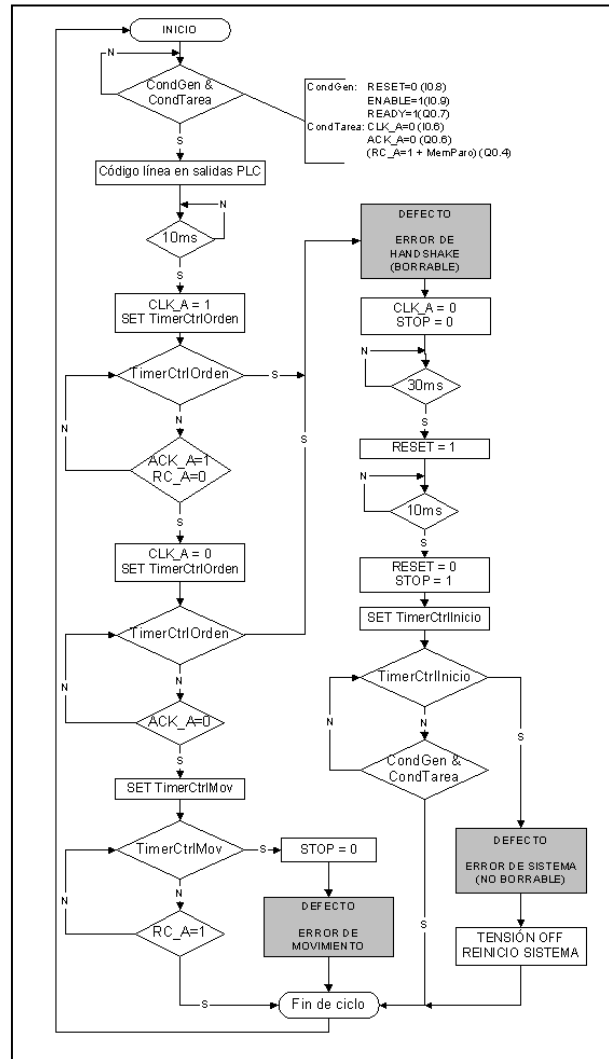


Figura 19. Diagrama de flux corresponent (FESTO. Control de ejes con SPC 200)

Aquesta rutina es crida des dels mòduls que necessiten donar ordre de deixar o recollir una peça de qualsevol posició del magatzem i per la seva programació s'ha seguit estrictament la seqüència proposada pel fabricant FESTO en el seu diagrama de flux.

### 3.2.4.- FB24- Magatzem recompte

Mitjançant un punter anem recorrent la taula d'existències del magatzem, DB113, i anem verificant el nombre de posicions lliures i les ocupades. Aquest mòdul ens proporciona informació de l'ocupació total del magatzem i de la quantitat de peces de cada color de què disposem.

Aquesta rutina es crida des del mòdul OB1 i s'executa sempre, en cada cicle d'Scan.

En el diagrama de flux de la figura 20 es pot observar la seqüència:

- En primer lloc s'inicialitza la variable *Punter* amb el valor 39. Aquesta variable, tal i com indica el seu nom, ens servirà de punter per anar resseguint tota la taula de dades de memòria corresponent a l'ocupació del magatzem utilitzant un direccionament del tipus relatiu.
- A partir d'aquí, es llegeix el tipus de peça que hi ha emmagatzemada en la posició que assenyala el *Punter*. Aquest valor pot ser 0 (no hi ha cap peça), 2 (peça negra), 3 (peça taronja) o 4 (peça metàl·lica).
- Aquest valor es guarda a la variable *Tipus\_peça*.
- Si el valor de la variable *Tipus\_peça* és zero, es salta a decrementar el valor del *Punter* amb una unitat. Si no s'ha acabat tota a taula (*Punter=0*) es retorna al punt de lectura relativa de la posició que assenyala el *Punter*.
- Si el valor de la variable *Tipus\_peça* val 2, s'incrementa en una unitat la variable corresponent a *Peces negres*, i es passa a fer la mateixa seqüència que si *Tipus\_peça* era zero.
- Si el valor de la variable *Tipus\_peça* val 3, s'incrementa en una unitat la variable corresponent a *Peces taronges*, i es passa a fer la mateixa seqüència que si *Tipus\_peça* era zero.
- Si el valor de la variable *Tipus\_peça* val 4, s'incrementa en una unitat la variable corresponent a *Peces metàl·liques* i es passa a fer la mateixa seqüència que si *Tipus\_peça* era zero.

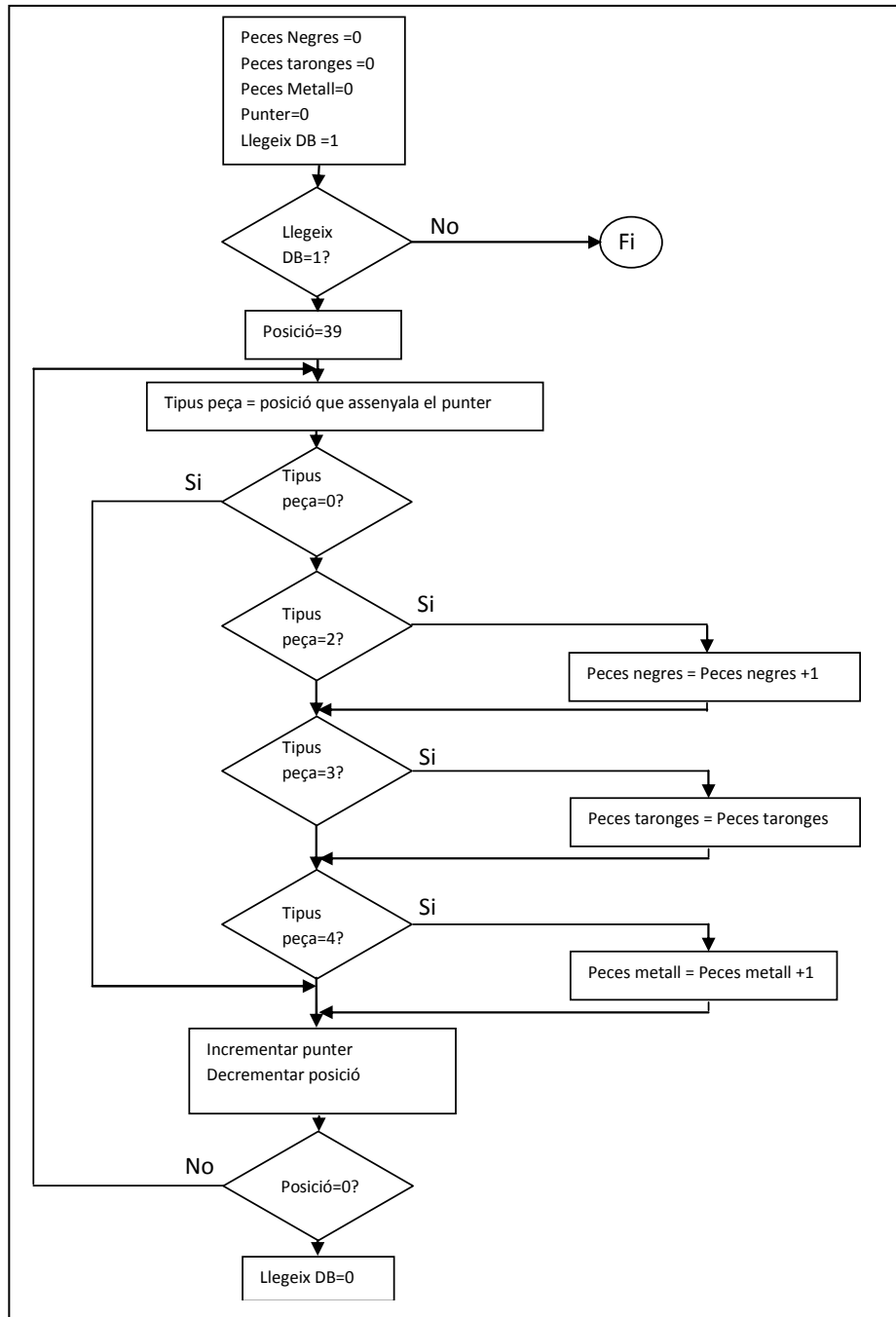


Figura 20. Diagrama de flux corresponent a la programació del mòdul FB24.



### 3.2.5.- FB25- Magatzem buscar posició

A aquesta funció li hem d'indicar quin tipus de peça volem buscar i una vegada definit el codi de la peça que es busca, es recorre la taula DB113 i es retorna el nombre de posició que contingui el codi especificat.

Si s'ha trobat el codi, s'indica mitjançant el bit *Trobat=1*.

- En mode automàtic, cal indicar a la variable "CodiBuscar" quin és el tipus de peça que es requereix: 2 (peça negra), 3 (peça taronja), 4 (peça metàl·lica).
- En mode manual, es col·loca un 1 a la variable "CodiBuscar" i aquest fet vol dir que qualsevol peça que es trobi és bona, per tant, a l'hora de buscar, tan sols caldrà veure si en la posició determinada hi ha peça o no.

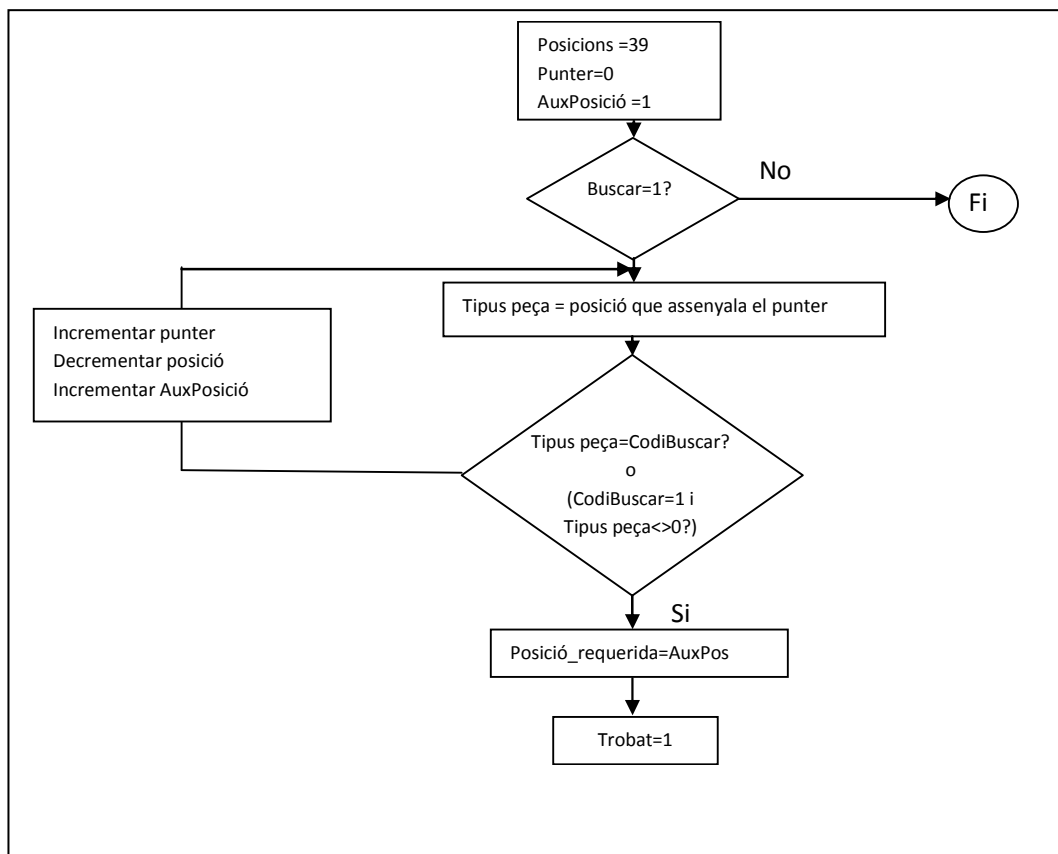


Figura 21. Diagrama de flux corresponent a la programació del mòdul FB25.

En el diagrama de flux de la figura 21 es pot observar la seqüència:

- Primer s'inicialitza la variable *Punter* a zero.
- Si la variable *Buscar* és igual a 1, vol dir que cal activar la seqüència de buscar.
- S'actualitza la variable *Tipus\_peça* amb el valor de la posició de memòria assenyalada per la variable *Punter*.
- Si el valor de la variable *Tipus\_peça* és igual al valor de la variable *CodiBuscar* (que és el tipus de peça requerit) o és diferent de zero i l'estació està funcionant en estat manual, llavors es passa el valor d'aquesta posició a la variable de sortida i s'acaba la seqüència.
- Si no, s'incrementa la variable *Punter* en una posició i torna a fer la comparació. I així consecutivament fins a recórrer tota la taula de memòria.

### 3.2.6- FB26- Magatzem escriu codi

Es proporciona a la funció un nombre de posició i un codi i aquesta col·loca aquest codi en la posició indicada. S'utilitza per tal d'esborrar les ubicacions d'extracció de peces.

### 3.2.7.- FB51- Identifica peça

Amb la lectura de les senyals provinents de les entrades dels tres detectors instal·lats en la posició d'entrada, aquest funció ens retorna el tipus de peça que s'ha de recollir a la variable "CodPeça".

A l'entrada es disposa de dos detectors fotoelèctrics per la detecció de presència de peça i diferenciació de color (negre - no negre) i un detector de tipus inductiu per diferenciar les peces metàl·liques de les no metàl·liques.

- Peça negra (codi 2): sensor òptic d'alta sensibilitat (presència de peça)
- Peça taronja (codi 3): òptic d'alta i baixa sensibilitat
- Peça metàl·lica (codi 4): els dos sensors òptics i l'inductiu

### **3.2.8.- FC4- Eixos control**

Mitjançant crides a la funció FC103, aquesta funció permet controlar el moviment dels dos eixos mitjançant un nombre de posició.

### **3.2.9.- FC8- Permisos**

Aquesta funció determina l'estat de les variables implicades en alguns moviments de l'estació. La senyal més habitual és:

PosInici: es posa a "1" quan els elements de control de l'estació es troben en les seves posicions inicials. Aquesta situació s'aconsegueix per primera vegada quan hi ha una reposició després d'una parada d'emergència.

### **3.2.10.- FC11- Modes Marxa**

Aquesta funció s'ocupa de la gestió de la botonera, realitzant el diàleg home-màquina, interpretant els senyals dels pulsadors i selectors i mostrant l'estat operatiu de l'estació mitjançant els pilots incorporats.

- Marxa. En pulsar START s'activa una petició de marxa que permetrà l'activació de la seqüència de treball.
- Parada. La parada és del tipus de Final de Cicle.

La botonera de l'estació disposa de (veure figura 22):

- un pulsador de marxa,
- un pulsador de parada,
- un pulsador d'inicialització,
- un pulsador de parada d'emergència
- i d'un selector amb clau per tal de commutar els estats de manual i automàtic.

En arrancar l'estació:

- En primer lloc cal pulsar el pulsador d'inicialització.

- A partir d'aquí, s'inicia una seqüència de comprovació de l'estat de l'estació i es situa el manipulador en la posició "zero", definida en la posició 14 del magatzem.
- Una vegada ha acabat el cicle d'inicialització, ja es pot activar el pulsador de marxa.
- Quan s'activa el pulsador de marxa de l'estació, automàticament es posa en funcionament la cinta transportadora d'entrada de peces.
- En cas de que l'estació estigui en mode manual, cada vegada que s'activi el pulsador de marxa o el pulsador de petició de sortida de peça de la pantalla corresponent del sistema SCADA, el manipulador extreu la peça que està ocupant la posició més baixa del magatzem, independentment del color que sigui, i la deixa a la cinta de sortida.
- En cas de que l'estació estigui en mode automàtic, el sistema SCADA llegeix la taula de comandes de l'Access i transmet l'ordre de la primera de la llista al PLC per tal de que s'executi.
- Una vegada executada la sortida de les tres peces corresponents a la comanda en curs (les peces es deixen en el palet que es troba en la cinta de l'estació transport), cal donar l'ordre des de la pantalla del sistema SCADA de que el carro ha estat alliberat i, automàticament, s'executa la següent comanda de la llista, i així successivament fins al final de la taula.

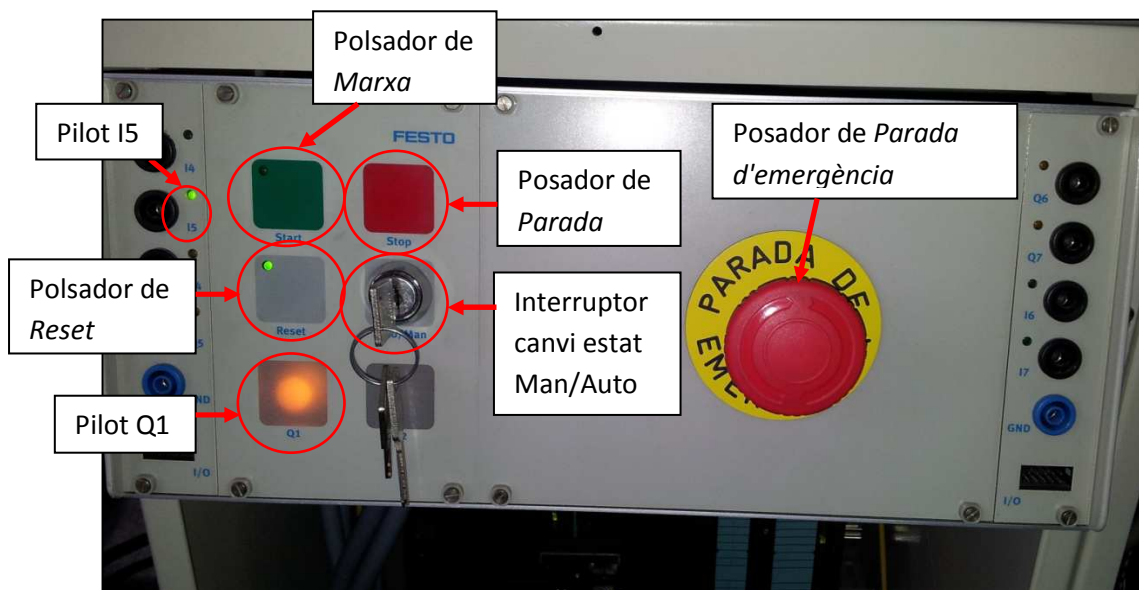


Figura 22. Detall de la botonera de l'estació

### **3.2.11.- FC13- Reposició**

Aquest programa realitza la seqüència de posada a zero després d'una parada d'emergència o després de connectar l'equip. S'activa en detectar una fallada en la tensió d'alimentació de l'estació, ja sigui accidental o provocat pel polsador de parada d'emergència. Es controla mitjançant la connexió de la tensió d'alimentació a l'entrada digital del PLC: "*Tensió\_OK*".

Es realitza una sèrie de moviments ordenats que permeten a l'estació col·locar els seus elements en la posició d'inici, quedant llesta per començar un nou cicle de treball.

En polsar el botó "*Reset*" s'activa aquesta seqüència de moviments. Quant s'ha acabat de forma satisfactòria, la senyal "*Est\_Inicialitzada*" es posa a "1".

Durant la seqüència de reposició, es prohibeix l'entrada de material a l'estació (el pilot Q1 es manté encès).

### **3.2.12.- FC18- Defectes**

Es fa una monitorització contínua de l'estat de l'estació de manera que s'activi un avís davant qualsevol situació irregular. La majoria de les alarmes que es generen són activades per temporitzadors de vigilància associats a moviments o condicions que s'han de complir dins un temps màxim.

Les alarmes queden memoritzades i s'esborren polsant el botó Reset si la causa del defecte ha desaparegut.

### **3.2.13.- FC19- Pilots**

Seqüència:

- 1.- en connectar l'estació, el pilot de l'entrada I5 s'encén, indicant que hi ha tensió d'alimentació a les sortides.
- 2.- El pilot Reset s'encén fix, indicant que l'estació necessita fer una posada a zero.
- 3.- Polsant Reset l'estació realitza la posada a zero (el pilot Reset parpadeja en seqüència de 3 intermitències mentre dura la maniobra).

4.- Quan acaba la seqüència, el bit "*Est\_inicialitzada*" es posa a "1".

Amb l'estació en la posició inicial, el Reset s'apaga i el polsador de marxa s'encén.

### **3.2.14.- FC21- Seqüència d'entrades de peces**

La seqüència a seguir per entrar una peça al magatzem segueix el diagrama de flux de la figura 23 i és la següent:

- 1.- Bloquejar les sortides de material activant una senyal d'entrada activa, si no hi ha una sortida en curs.
- 2.- Determinar el tipus de peça present en la zona d'entrades.
- 3.- Per determinar la primera posició buida, s'efectua un escombrat de la taula de posicions DB113 mitjançant un punter, que comença a llegir els codis des de la posició 1.
- 4.- Es notifica la posició al control d'eixos i s'activa.
- 5.- Es va a buscar la peça i es diposita en la posició indicada.
- 6.- Una vegada dipositada, s'actualitza la posició on s'ha dipositat.
- 7.- Es desbloca la sortida de material.

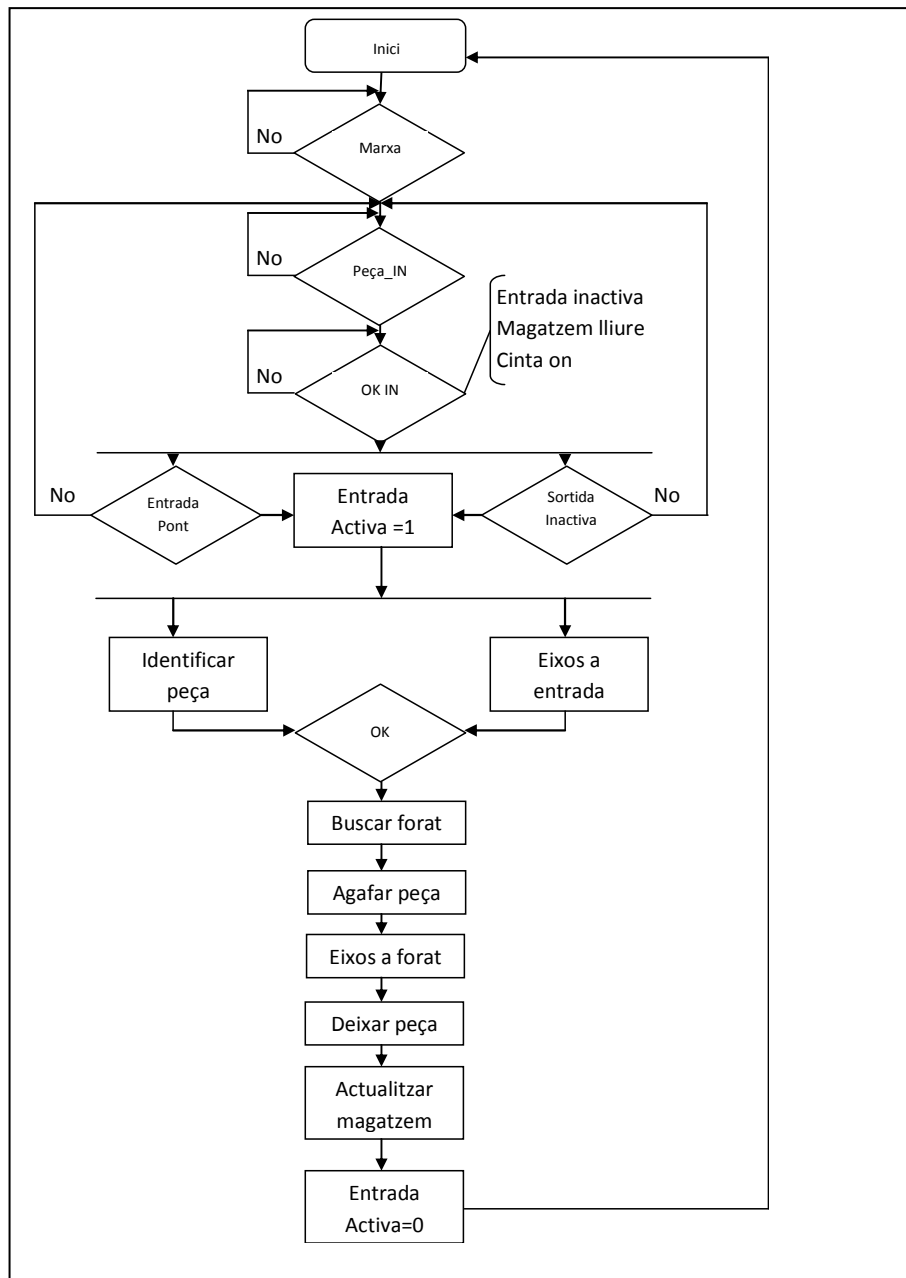


Figura 23. Diagrama de flux corresponent a la programació del mòdul FC21.

### 3.2.15.- FC22- Seqüència de sortides de peces

L'algoritme de control de sortides s'ocupa de la gestió de les comandes de material.

- Sortides en estat manual.

Si l'estació es troba en estat manual, per cada pulsació del polsador de *Marxa* de l'estació, o del botó corresponent de la pantalla de l'SCADA, es donen totes les peces del magatzem com a vàlides per la sortida i s'executa la funció FC22 (Seq\_Sortides). A partir d'aquí, la seqüència d'extracció de peça en estat manual és:

1. Bloquejar les entrades de material activant la senyal de sortida activa *Sortida\_ON*.
2. Buscar la primera posició ocupada del magatzem fent una crida a la funció FB25.
3. De la posició ocupada, determinar el tipus de peça que s'hi emmagatzema.
4. Notificar la posició al control d'eixos.
5. Extreure la peça.
6. Esborrar la posició extreta de la taula DB113 fent una crida a la funció FB26.
7. Esborrar la senyal de sortida activa *Sortida\_ON*.

Si el magatzem està sense existències, el manipulador dirigeix les ventoses a la zona d'entrades on s'esperarà l'entrada d'una peça per traslladar-la directament a la cinta de sortida manual.

➤ Sortides en estat automàtic.

Si l'estació es troba en estat automàtic, les comandes que cal servir vindran determinades per les ordres que arribin del sistema SCADA. En estat automàtic, el sistema SCADA llegeix la primera comanda de la base de dades d'Access on es guarden totes les comandes a executar i la transfereix al PLC. A partir d'aquí, es desglossa la variable i s'extreuen les xifres referents a les peces (de les quatre xifres que componen un codi, les tres de la dreta), mitjançant la funció FC50.

Per exemple, si el codi de la comanda és el 1342 significa que aquesta ha d'estar composta per (veure apartat 2.- Descripció del Procés, on es defineix la correlació entre el tipus de peça i el número que se li assigna):



peça taronja + peça metàl·lica + peça negra

La rutina del FC50 extreu les xifres corresponents a les peces i les guarda en una pila FIFO. D'aquesta manera, la seqüència d'extracció serà en el mateix ordre del codi requerit. A partir d'aquí, la seqüència d'extracció de peça en estat automàtic és:

1. Bloquejar les entrades de material activant la senyal de sortida activa *Sortida\_ON*.
2. Determinar quin tipus de peça és la primera a extreure.
3. Buscar la primera posició ocupada per la peça que cal extreure fent una crida a la funció FB25.
4. Notificar la posició al control dels eixos.
5. Extreure la peça.
6. Esborrar la posició d'extracció de la taula DB113 mitjançant la funció FB26.
7. Buscar la primera posició ocupada per la peça corresponent al segon tipus que cal extreure fent una crida a la funció FB25.
8. Notificar la posició al control dels eixos.
9. Extreure la peça.
10. Esborrar la posició d'extracció de la taula DB113 mitjançant la funció FB26.
11. Buscar la primera posició ocupada per la peça corresponent al tercer tipus que cal extreure fent una crida a la funció FB25.
12. Notificar la posició al control dels eixos.
13. Extreure la peça.
14. Esborrar la posició d'extracció de la taula DB113 mitjançant la funció FB26.
15. Esborrar la senyal de sortida activa *Sortida\_ON*.

Si la seqüència de cerca de posició ocupada per una peça determinada indica que no hi ha existències en el magatzem, el manipulador es dirigeix a la zona d'entrades a esperar una peça. Si la peça que entra correspon a l'esperada, aquesta es trasllada directament a la seva

posició sobre el palet de sortida; sinó, s'introdueix al magatzem (si no hi ha posicions lliures, la peça es diposita a la posició 40 de rebuig). L'operari té la opció de saltar la comanda en cas de que no hi hagi l'existència en el magatzem d'alguna de les peces requerides. en aquest cas, mitjançant un polsador en la pantalla corresponent del sistema SCADA es salta la comanda en curs i es passa a la següent.

### **3.2.16.- FC50- Desglossa codi**

Quan s'introdueix un codi en la variable corresponent, aquesta funció el desglossa en tots els seus components, identificant d'aquesta manera les tres peces que formen la comanda.

### **3.2.17.- FC103- Eixos posicions**

Totes les posicions del magatzem intermedi estan numerades. Mitjançant el número de posició, aquesta funció determina les coordenades que cal introduir als controls dels eixos perquè el manipulador es situï sobre d'aquesta.

### **3.2.18.- DB113- Existències magatzem**

Àrea de memòria destinada a contenir les dades d'ocupació del magatzem. En aquest DB s'han definit les 40 posicions del magatzem (40 paraules internes), més 4 bytes corresponents al nombre total de peces que hi ha al magatzem, el nombre de vermelles, el nombre de negres i el nombre de metàl·liques. Cada vegada que hi ha una entrada o sortida de peces del magatzem s'actualitza la paraula corresponent (dins el DB113) amb el codi identificador de la peça en qüestió (2: peça negra, 3: peça taronja i 4: peça de metall).

#### 4.- EL SISTEMA SCADA

En aquest apartat es descriu què és un sistema SCADA de forma general i quines són les prestacions que ens ofereix un software d'aquest tipus. Seguidament, es presenta i es descriu el sistema Vijeo Citect que és el que s'ha utilitzat per l'elaboració d'aquest projecte.

Es dona el nom de SCADA (*Supervisory Control And Data Acquisition o Control amb Supervisió i Adquisició de Dades*) a qualsevol software que permeti l'accés a dades remotes d'un procés i permeti, utilitzant les eines de comunicació necessàries en cada cas, el control d'aquest.

Els sistemes SCADA provenen d'una perspectiva integrada de tots els recursos de control i informació de la planta. D'aquesta manera, els enginyers, supervisors o operadors poden visualitzar i interactuar amb els processos mitjançant representacions gràfiques d'aquests.

Si ens atenem a la definició, veiem que no es tracta purament d'un sistema de control, sinó d'una utilitat software de monitorització o supervisió que realitza la tasca d'interfície entre els nivell de control (PLC) i els de gestió a un nivell superior. Un sistema SCADA proveeix de tota la informació que es genera en el procés productiu a diversos usuaris, tant del mateix nivell com d'altres sectors dins de l'empresa: control de qualitat, supervisió, manteniment... en un marc de model d'automatització piramidal.

Les característiques perquè la seva utilització sigui perfectament aprofitada són:

- Funcionalitat completa de maneig i visualització en el sistema operatiu Windows sobre qualsevol PC estàndard.
- Arquitectura que permeti combinacions amb aplicacions estàndard i d'usuari, que permetin als integradors crear solucions de comandament i supervisió optimitzades (ActiveX per ampliació de prestacions, OPC per a comunicacions amb tercers, OLE-DB per comunicacions amb bases de dades, llenguatge estàndard integrat com VB o C, accés a funcions i dades mitjançant API).
- Senzillesa d'instal·lació, sense exigències de hardware elevades, fàcil d'utilitzar i amb interfícies amigables amb l'usuari.

- Permetre la integració amb les eines ofimàtiques i de producció.
- Fàcilment configurable i escalable, ha de ser capaç de créixer o adaptar-se segons les necessitats canviants del sector i la tecnologia.
- Funcions de comandament i supervisió integrades. D'aquesta manera, el programa SCADA pot trobar-se instal·lat en els PC que siguin necessaris en els diferents departaments de l'empresa i, per mitjà de restriccions per privilegis, es dóna accés als objectes gràfics als usuaris segons les seves necessitats. Així, per exemple, per el rol de gerència es pot donar el privilegi d'accedir a qualsevol pantalla del procés però tan sols poder manipular aquelles pantalles corresponents a dades de tendències.
- Comunicacions flexibles per poder comunicar-se amb total facilitat i de forma transparent a l'usuari amb l'equip de planta i amb la resta de l'empresa.

Els objectius dels sistemes SCADA es conceben principalment com una eina de supervisió i comandament. Anteriorment algunes empreses disposaven de grans parets en les sales de control amb el gràfic del procés i ple de llumetes, interruptors, polsadors... Amb un sistema SCADA, tan sols és necessari disposar d'un PC i una o diverses pantalles per tal de poder visualitzar les diferents parts del procés. Les diferents pantalles gràfiques, totalment programables, poden incloure, fins i tot, un fons amb la fotografia del procés i disposar de tots i cada un dels dispositius de camp configurables i modificables des del mateix PC. Entre els objectius d'un sistema SCADA podem destacar:

- Economia: és més fàcil veure el que està succeint a la instal·lació des d'una sala de control i a través d'un PC que enviar a un operari a fer aquesta tasca o haver de disposar de tota una paret per simular el procés.
- Accessibilitat: totes les dades d'un parc eòlic, per exemple, les podem disposar amb un sol clic en una taula. Serà possible modificar els paràmetres de funcionament de cada aerogenerador, posant fora de servei els que donin indicis d'anomalies; consultar l'estat de les estacions transformadors del parc, parar els molins que no siguin necessaris...

- Manteniment: l'adquisició de dades materialitza la possibilitat d'obtenir dades d'un procés, emmagatzemar-los i presentar-los de manera intel·ligible per un usuari no especialitzat. La mateixa aplicació es pot programar de manera que ens avisi quan s'aproximin les dades de revisió o quan una màquina tingui més fallades de les considerades normals.
- Ergonomia: és la ciència que procura fer que la relació entre l'usuari i el procés sigui el menys tirant possible.
- Gestió: totes les dades recopilades poden ser valorades de múltiples maneres mitjançant eines estadístiques, gràfiques, valors tabulats... que permetin explotar el sistema amb el major rendiment possible.
- Flexibilitat: qualsevol modificació d'alguna de les característiques del sistema de visualització no significa una despesa en temps ni mitjans, ja que hi ha modificacions físiques que requereixen la instal·lació de cablejat o d'un controlador.
- Connectivitat: es busquen sistemes oberts. La documentació dels protocols de comunicació actuals permeten la interconnexió de sistemes de diferents proveïdors i evita l'existència de llacunes informatives que puguin causar fallades en el funcionament o en la seguretat.

Tots els sistemes, de major o menor complexitat, orientats al que anteriorment s'ha comentat, apareixen normalment com a:

- **MMI**: Man Machine Interface, Interfície home-màquina.
- **HMI**: Human Machine Interface, Interfície humà-màquina.

El sistema a controlar apareix davant l'usuari a partir d'un nombre més o menys elevat de pantalles amb major o menor informació. Podem trobar plànols, fotografies, esquemes, gràfics de tendències...

#### **4.1.- Prestacions d'un sistema SCADA.**

Un paquet SCADA, com a eina d'interfícies home-màquina, comprèn tota una sèrie de funcions i utilitats encaminades a establir una comunicació el més clara possible entre el procés i l'operador:

Entre les prestacions d'una eina d'aquest tipus destaquen:

La monitorització. Representació de dades en temps real als operadors de planta. Es llegeixen dades dels PLCs (temperatures, velocitats, detectors...), des d'una màquina simple a tota una línia d'extrusió.

La supervisió. Supervisió, comandament i adquisició de dades d'un procés i eines de gestió per la presa de decisions (manteniment predictiu, per exemple). A més permet la capacitat d'executar programes que poden supervisar i modificar el control establert i, segons certes condicions donades, anular o modificar tasques associades als autòmats. Evita una continua supervisió humana.

L'adquisició de dades dels processos en observació. Un sistema de captació solar, per exemple, es pot observar mitjançant eines registradores i obtenir així un valor mitjà de la irradiació en la zona, guardant els valors obtinguts i avaluant-los a posteriori.

La visualització de l'estat de les senyals del sistema (alarmes i events). Reconeixement d'events excepcionals ocorreguts a la planta i la seva immediats posada en coneixement als operaris per tal d'efectuar les accions correctores pertinents. A més, els panells d'alarma poden exigir alguna acció de reconeixement per part de l'operari, de manera que quedin registrades les incidències.

El comandament. Possibilitat de que els operadors puguin canviar consignes o altres dades claus del procés directament des de l'ordenador.

Gravació d'accions o receptes. En alguns processos s'utilitzen combinacions de variables que són sempre les mateixes. Un sistema de receptes permet configurar tota la planta de producció executant una sola ordre.

Garantir la seguretat de les dades. Tant l'enviament com la recepció de dades han d'estar suficientment protegits d'influències no desitjades, intencionades o no.

Garantir la seguretat en els accessos. Restringint zones de programa compromeses a usuaris no autoritzats, registrant tots els accessos i accions dutes a terme per qualsevol operari.

Possibilitat de programació numèrica. Permet realitzar càlculs aritmètics d'elevada resolució sobre la CPU de l'ordinador.

#### **4.2.- El sistema SCADA utilitzat en aquest projecte (VIJEO CITECT)**

Vijeo Citect és un dels paquets SCADA més difosos a nivell mundial. Les característiques i la potència d'aquest software fan que sigui adequat per qualsevol tipus d'aplicació. Amb una complexitat mitjana, les seves eines de configuració i les seves característiques de funcionament permeten gran flexibilitat en el seu desenvolupament, permetent qualsevol tipus de solució.

El software Vijeo Citect necessita una clau de hardware i la llicència per tal de poder activar el runtime per un temps indefinit. Tot i això, es pot baixar de la pàgina web del fabricant Schneider Electric una versió de lliure distribució amb un *runtime* de 15 minuts, passats els quals s'atura automàticament i cal tornar a activar-lo. A part d'aquesta limitació, no n'hi ha de cap altre tipus, de manera que el fa totalment adequat per la seva utilització docent.

Les llicències de *runtime* es compren segons el nombre de *tags* de variable de comunicació que es vol disposar; a més *tags* de comunicació més alt és el preu (un *tag* de variable és un element de comunicació entre l'SCADA i el procés). Per tant, és interessant que en la programació del sistema de control i monitorització s'ajustin els *tags* de comunicació als mínims indispensables.

Pel que fa a les variables del tipus analògic, cada variable serà un *tag* de comunicació. Ara bé, pel que fa a les variables del tipus digital, aquestes es poden agrupar en paraules de vuit bits cada una de manera que tota la paraula és un sol *tag* de comunicació. D'aquesta manera es pot disminuir substancialment el nombre d'aquests *tags* necessaris. La Universitat de Vic ha adquirit una llicència de 150 *tags* amb la perspectiva de poder programar el sistema SCADA pel control i monitorització de tota la cèl·lula flexible de fabricació.

El sistema SCADA Vijeo Citect permet fer la programació pel control i supervisió d'un procés d'una forma molt ordenada i estructurada.

**Possibilitats de programació que ofereix el Vijeo Citect:**

- Definició de clústers. Per cada projecte es poden definir diferents *clústers*. En l'àmbit de la informàtica, un *clúster* és un conjunt d'ordinadors que es comporten com si fossin un de sol.

Definir *clústers* permet agrupar diferents conjunts de components dins d'un únic projecte, el que permet múltiples sistemes independents per a ser monitoritzats i controlats. D'aquesta manera s'aconsegueix:

- alt rendiment (gran capacitat de càlcul)
- alta disponibilitat (sistemes redundants)
- equilibri de càrrega (el treball es reparteix entre diferents ordinadors)
- escalabilitat (facilitat de creixement)

Per a projectes de petita escala i poques variables de procés, només serà necessari definir un sol *clúster*.

- Definició de servidors. Per cada projecte cal definir un o més servidors en els quals es guardaran les variables utilitzades. Disposem de quatre tipus diferents de servidors:

- Servidors d'alarmes. Els servidors que monitoritzaran les alarmes.
- Servidors d'informes. Els servidors que controlen el processament d'informes.
- Servidors de tendències. Els servidors que controlen l'acumulació i el registre d'informació de tendències.
- Servidor d'E/S. Servidors de comunicacions que intercanvien dades entre els dispositius d'E/S i clients.

- Els privilegis. El Vijeo Citect proporciona vuit privilegis, numerats de l'1 al 8, que s'utilitzen per restringir l'accés a les diferents parts del projecte. Per posar en pràctica els privilegis en el projecte cal:

- Assignar un privilegi a l'element del sistema al qual es vol restringir l'accés (ordre, objecte, informe...)



- Assignar el privilegi o privilegis al rol o als rols que hauran de controlar aquests elements.

Si en un element no se li assigna cap restricció de privilegi, per defecte el programa li assigna el privilegi 0, i per tant és accessible per qualsevol usuari.

- Definició de rols. En crear un rol es permet establir diferents grups d'usuaris (a cada un d'ells se li assignarà un rol) per tal de poder establir quins seran els privilegis d'actuació dins els sistema que tindrà cada un. D'aquesta manera, es pot dur a terme un control exhaustiu dels accessos a modificació de paràmetres del sistema. Per tal de definir els rols, s'haurà de tenir en compte les tasques que els usuaris tindran assignades en el projecte i quins seran els elements als quals cada usuari haurà de tenir accés o no.

El Vijeo CitectSCADA proporciona 8 privilegis, numerats de l'1 al 8 que s'utilitzen per restringir l'accés a les diferents parts del projecte. D'aquesta manera, a cada rol se li assignarà un número de privilegi, i aquest número anirà associat als objectes de control del procés. Així, només l'usuari que disposi del rol amb un privilegi igual al que té assignat l'objecte el podrà manipular.

- Definició d'usuaris. A cada usuari se li assigna un rol depenent de les activitats i processos als que han d'operar i controlar. Si no s'assigna cap rol a l'usuari, aquest usuari no tindrà cap tipus de privilegi i tan sols podrà veure el procés i accedir als elements que no tinguin assignada cap restricció.

- Configuració de dispositius de comunicació d'E/S. El Vijeo CitectSCADA es pot configurar per comunicar-se amb els dispositius d'E/S de diferents fabricants. Per establir la comunicació amb un dispositiu d'E / S, es pot fer utilitzant els drivers de comunicació propietaris de cada equip de

treball amb una comunicació directa client/servidor o utilitzar un servidor OPC (OLE for Process Control).

Ens permet configurar tres tipus de dispositiu d'E/S diferents:

- Dispositiu d'E/S extern. Escollirem aquest tipus de dispositiu quan vulguem comunicar-nos amb un element extern o de camp utilitzant un servidor OPC o els *tags* de propietaris. Cada *tag* de comunicació que associem a aquest dispositiu serà un *tag* que descomptarà de la llicència de què disposem.
- Dispositius de memòria de E/S. Si escollim aquest tipus de dispositiu, els *tags* que hi associem utilitzaran una àrea de memòria del PC per tal de ser emmagatzemats. Aquests *tags* no gasten llicència i mantenen el valor de la variable en cas de pèrdua de corrent.
- Dispositius d'E/S de disc. Són dispositius iguals que els anteriors.

El fabricant del sistema SCADA Vijeo Citect recomana utilitzar els anteriors en front d'aquests ja que, en cas de pèrdua de corrent, si s'utilitzen diferents equips configurats com a client-servidor, en el cas dels dispositius anteriors, les variables mantindran el seu valor.

- Definició de *tags* de variables del projecte. Cal definir un *tag* per cada element de camp que es vol utilitzar en el mode *runtime*. Per tal de definir els *tags*, cal associar a cada un d'ells una etiqueta. D'aquesta manera:
  - no cal recordar l'adreça física de l'element cada vegada que es vulgui utilitzar la variable
  - s'utilitza un nom pel *tag* que sigui lògic i descriptiu
  - l'adreça física en la que es troba aquesta variable en el dispositiu de camp només es defineix una vegada. Si es canvia aquesta direcció, només cal actualitzar-la una vegada.

Es pot treballar amb quatre tipus diferent de *Tags*:

- *Tags de variable*: són variables que permeten treballar directament amb punts reals de la instal·lació (PLC) o virtuals (RAM o disc dur).
- *Tags locals*: variables que no es retenen. No tenen una adreça física de memòria que permetin guardar-les.
- *Tags de tendència*: permeten gestionar els gràfics de tendències.
- *Tags SPC*: permeten el control estadístic de procés.
- Cal definir aquestes variables com un tipus de dades específic (veure figura 24). Les variables més comuns que s'utilitzen són del tipus digital i sencers.

Data Type	Variable	Size	Allowed Values
BCD	Binary- Coded Decimal	2 bytes	0 to 9,999
BYTE	Byte	1 byte	0 to 255
DIGITAL	Digital	1 bit or 1 byte	0 or 1
INT	Integer	2 bytes	-32,768 to 32,767
UINT	Unsigned Integer	2 bytes	0 to 65,535
LONG	Long Integer	4 bytes	- 2,147,483,648 to 2,147,483,647
ULONG	Unsigned Long Integer (Only for display on a screen. Arithmetic operations are not supported. )	4 bytes	0 to 4,294,967,295
LONGBCD	Long Binary- Coded Decimal	4 bytes	0 to 99,999,999
REAL	Floating Point	4 bytes	-3.4E38 to 3.4E38
STRING	String	256 bytes (maximum)	ASCII (null terminated)

Figura 24. Tipus de dades de *tags* de variables.

- Alarmes. Les alarmes es basen en la vigilància dels paràmetres de les variables del sistema. En definir qualsevol alarma s'haurà d'associar a un *tag* de variable definit prèviament. Es tracta de successos no desitjables i que la seva aparició dona lloc a problemes de funcionament.

CitectSCADA suporta set tipus d'alarma diferents:

- Digital. Es poden configurar les alarmes digitals per activar-les basant-se en l'estat d'una o dues variables digitals. L'alarma s'activa quan la condició d'activació té una durada d'un especificat període de retard.
  - Analògica. Les Alarmes analògiques s'activen quan es canvia una variable analògica més enllà d'un o més límits específics.
  - Amb marca de temps. Són similars a les alarmes digitals, excepte que s'utilitza un comptador per proporcionar un registre de l'hora exacta de quan es produeix una condició d'activació. Només es pot associar amb una única variable digital.
  - Alarma Avançada. Una alarma avançada es converteix en activa amb el resultat dels canvis en una expressió CICODE.
  - Multi digital. L'alarma Múltiple digital utilitza la sortida de tres variables digitals (per exemple: les etiquetes A, B i C) per definir vuit estats. Els Estats representen totes les combinacions possibles de veritable / fals que les variables poden tenir.
  - Amb marca de temps digital i analògic. Les alarmes analògiques i digitals amb marca de temps són diferents a altres tipus d'alarma ja que no es basen en el sondeig de variables per determinar les condicions d'alarma. Operen a través d'un procés en el qual es notifica al servidor d'alarma de qualsevol canvi de valor a una variable concreta utilitzant la funció de CICODE AlarmNotifyVarChange.
- Definir i crear pàgines gràfiques. En el mode *runtime* del sistema SCADA caldrà dissenyar una sèrie de pàgines gràfiques que ens mostrin a la pantalla una "finestra del procés" per tal de poder-hi interaccionar. En aquestes pàgines gràfiques es mostrarà l'estat en temps real dels diferents elements del procés. Aquests elements de la pantalla gràfica es coneixen com a objectes i n'hi pot haver del tipus digital, analògic, numèric, text...
- Cada un d'aquests objectes s'associa a un *tag* definit prèviament i canvia la seva forma física en funció del valor d'aquesta variable.

➤ Genies i Super Genies. CitectSCADA inclou biblioteques de genis i Super Genis que es poden utilitzar en qualsevol projecte. També se'n poden definir de propis i agregar-los a una biblioteca específica per ser reutilitzats de nou dins del projecte.

- Genie - Normalment els objectes gràfics que s'insereixen a les pàgines gràfiques es configuren de manera independent per cada una, i cal anar entrant els *tags* en cada un d'ells. Pot ser que en més d'una pàgina gràfica es vulgui monitoritzar un objecte de les mateixes característiques (ex: un tanc amb el seu motor, agitador, nivell...), llavors, mitjançant un Genie, es poden agrupar una col·lecció d'objectes relacionats, que posteriorment s'afegiran a les pàgines de gràfics com un conjunt. Es pot enganxar qualsevol nombre de genis en una pàgina de gràfics (per exemple, diversos tancs a la mateixa pàgina).

Quan inserim un Genie en una pàgina gràfica tan sols se'ns obrirà una pàgina de diàleg que ens sol·licitarà els *tags* de variables que li vulguem associar i tots els altres aspectes ja es troben configurats.

- Súper Genie - Els Super Genies (també coneguts com a pàgines emergents) són pàgines dinàmiques que permeten transferir informació entre pàgines durant l'execució del programa.

Es pot definir, per exemple, un supergenie que siguin els comandaments d'un agitador. Llavors, aquests comandaments s'associen al Genie definit amb els controls del tanc i en activar-lo se'ns obrirà una pantalla emergent amb els comandaments. A més, també podem fer servir la finestra emergent amb els comandaments per associar-los a un altre Genie que en necessiti uns d'iguals.

Una pantalla emergent es pot associar a un Genie o ser cridada directament des d'un símbol (per exemple un botó).

- Tendències. Els gràfics de tendència poden ajudar a entendre millora una planta i el rendiment dels diferents equips. Es poden utilitzar per l'anàlisi visual dinàmica, per dur a terme registres de producció o per registrar periòdicament l'estat dels equips per a l'eficiència i el manteniment preventiu. Les tendències es poden registrar a intervals regulars (tendència periòdica) o només quan es produeix un esdeveniment (tendència d'esdeveniment).
  
- Informes. Es poden sol·licitar informes periòdics sobre l'estat de la planta que ens proporcionin informació sobre condicions especials de la instal·lació. Els informes es poden executar en una sol·licitud base o a una hora determinada, quan es produeixin uns determinats esdeveniments. Un informe es pot imprimir o guardar-lo al disc per ser imprès posteriorment.
  
- Editor de funcions CICODE. El Vijeo Citect ofereix dos llenguatges de programació amb els quals l'usuari pot controlar i manipular els diferents components d'un projecte:
  - Cicode. Un llenguatge de programació estructurat dissenyat per utilitzar en el sistema SCADA Vijeo Citect per monitoritzar i controlar objectes d'un projecte.
  - Citect VBA. Un llenguatge Visual Basic per aplicacions (VBA ) i VBScript compatible amb llenguatge Basic.
  
- Comunicació Vijeo Citect amb altres aplicacions. El software SCADA Vijeo Citect permet la comunicació i intercanvi de dades amb altres aplicacions pel seu post tractament, emmagatzematge, anàlisi, gestió del control de planta...

El Citect SCADA utilitza diferents metodologies per l'intercanvi de dades:

- Dynamic data exchange (DDE), on el Citect SCADA pot actuar com a DDE server aportant els valors dels tags requerits pels clients o de DDE client, requerint dades d'altres aplicacions.
- Open database connectivity (ODBC), on el Citect SCADA actua com a ODBC Server, permetent a altres aplicacions llegir les variables directament.
- Utilitzant una base de dades externa, on el Citect SCADA i altres aplicacions utilitzen la mateixa base de dades per emmagatzemar i llegir informació.

**Atès que l'objectiu principal d'aquest projecte és fer-lo servir per la docència del programa Vijeo Citect, en els punts següents es passa a descriure molt detalladament com s'ha elaborat la programació pel control i supervisió de l'estació magatzem utilitzant aquest software.**

#### **4.2.1.- Procés de creació d'aquest projecte amb el programa Vijeo Citect**

En aquest apartat es descriu d'una manera molt acurada tot el procés de programació que s'ha seguit del sistema SCADA Vijeo Citect per tal de dur a terme la monitorització i control de l'estació magatzem de la cèl·lula flexible de fabricació. Per poder fer un fàcil seguiment del mètode seguit, es va complementant la descripció de la programació amb les pantalles capturades del mateix programa.

En executar el programa Vijeo Citect se'ns obren tres finestres:

- CitectSCADA Explorer. Aplicació que s'utilitza per crear i gestionar els projectes. Es mostra una llista dels projectes i proporciona accés directe als components de cada un. Es pot utilitzar l'Explorador per canviar el nom, còpia de seguretat, restaurar o eliminar un projecte...
- CitectSCADA Editor de projectes. Aplicació utilitzada per crear i gestionar la informació de configuració d'un projecte; incloent etiquetes, alarmes, components del sistema, components de comunicacions...

- CitectSCADA Gràfics Builder. Aplicació que s'utilitza per dissenyar, crear i editar els components gràfics d'un projecte; incloent plantilles, objectes gràfics, símbols, genis i supergenis...

Per crear aquest projecte nou, es fa des de la pantalla CitectSCADA Explorer. El programa ofereix en aquesta pantalla uns projectes tipus, que inclouen plantilles i exemples de programació i poden ser utilitzats i inclosos en el nostre propi. Per crear aquest projecte, anirem al menú "Archivo-nuevo proyecto" i obrirem la finestra per tal de definir-ne el nom i les propietats (veure figura 25).

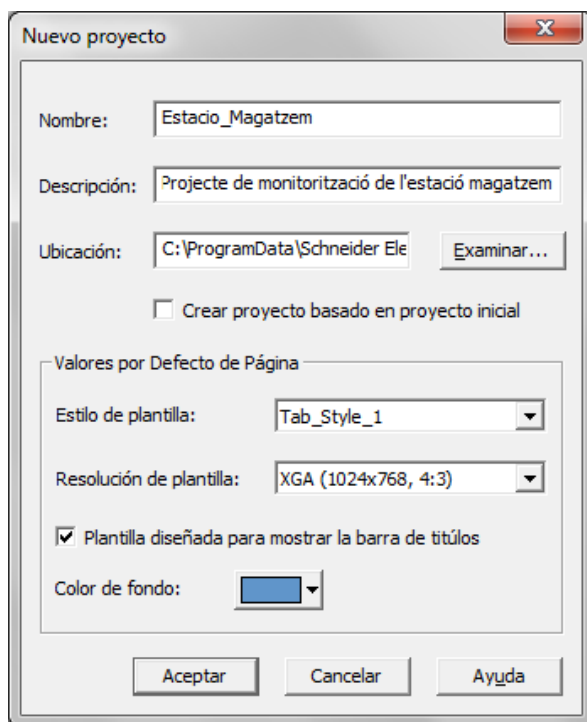


Figura 25. Pantalla de creació de projecte nou

Aquesta pantalla ens permet:

- Definir el nom del projecte
- Donar-li una descripció
- Definir-ne la ubicació
- Definir-ne les propietats bàsiques com el tipus de plantilla inicial, la resolució i el color de fons.

Nota: si escollim el tic "crear proyecto basado en proyecto inicial", el podem enllaçar a un altre projecte i automàticament tindrem totes les llibreries d'aquest.

#### 4.2.2.- Definició de *clústers* i servidors.

- Definició de *clústers*

Per tal de definir un *clúster*, anem al CitectSCADA Editor de projectes i seleccionem el menú "Servidores-clusters". Aquí se'ns obra una nova pantalla que ens permet definir els *clústers* del nostre projecte (veure figura 26).



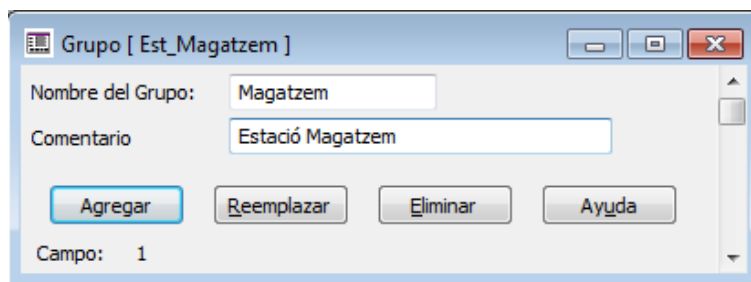


Figura 26. Pantalla d'agregació de clústers

És important tenir en compte que en totes les pantalles de configuració disposem d'un botó "Agregar" (l'utilitzarem sempre que vulguem afegir un element al projecte) i d'un botó "Reemplazar" (l'utilitzarem per canviar les propietats d'un element ja creat).

Per cada *clúster* que entrem, haurem de fer "Agregar". En aquest projecte només s'ha definit el clúster *Magatzem*.

➤ Definició de servidors

Una vegada definits els *clústers*, podem passar a definir els servidors possibles per l'aplicació.

Des del menú "Servidores" disposem de l'accés als quatre tipus diferents de servidors. Per aquest projecte s'han definit els quatre servidors mostrats en les figures següents:

- Servidors d'alarmes. Els servidors que monitoritzaran les alarmes (veure figura 27).

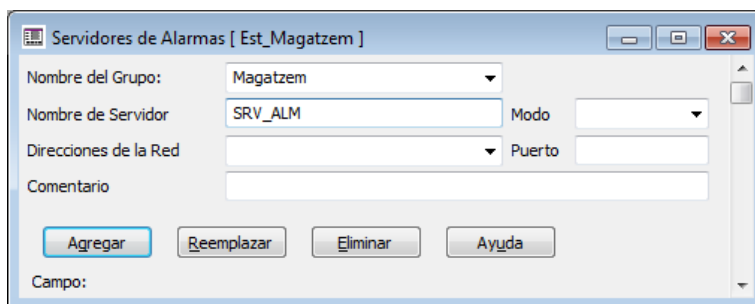


Figura 27. Pantalla d'agregació de servidors d'alarmes

- Servidors d'informes. Els servidors que controlen el processament d'informes (veure figura 28).

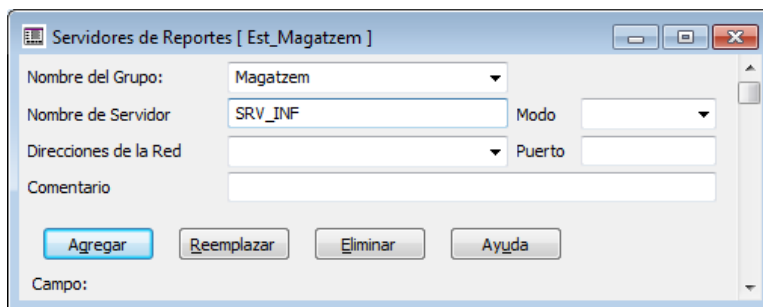


Figura 28. Pantalla d'agregació de servidors d'informes

- Servidors de tendències. Els servidors que controlen l'acumulació i el registre d'informació de tendències (veure figura 29).

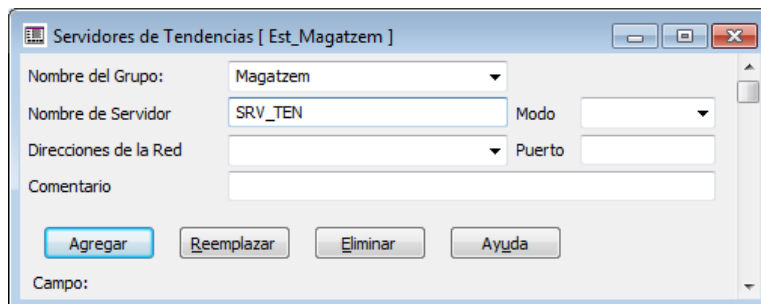


Figura 29. Pantalla d'agregació de servidors de tendències

- Servidor d'E/S. Servidors de comunicacions que intercanvien dades entre els dispositius d'E/S i clients (veure figura 30).

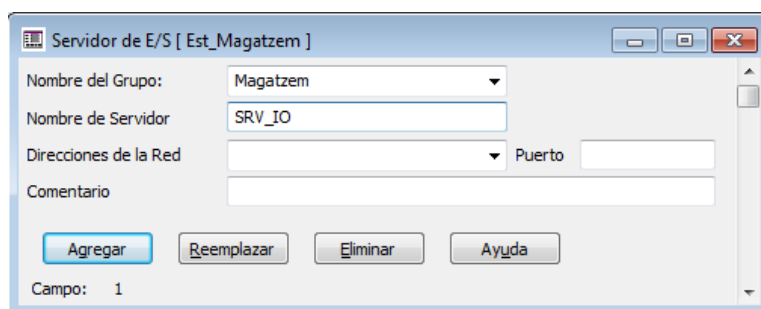


Figura 30. Pantalla d'agregació de servidors d'entrades i sortides

Per cada una d'aquestes pantalles, si a "Direcciones e la Red" no hi poso cap adreça, automàticament m'agafa la IP de l'equip on l'instal·lo.

**Sempre que es treballi amb registres en la pantalla CitectSCADA Editor de projectes, és important fer servir l'opció de menú: "Archivo-Empacar (pack)". Aquesta opció tanca de forma ordenada les taules on es guarden les dades que hem modificat.**

#### 4.2.3.- Definició de rols i usuaris

➤ Definició de rols

En aquest projecte s'han definit tres tipus de rols diferents (veure figures 31, 32 i 33):

- Administrador. Al rol d'administrador se li ha associat el màxim de privilegis (de 1 a 8) que li permeten accedir a qualsevol element del programa.
- Vendes. Al rol de vendes se li ha associat el privilegi 1.
- Producció. Al rol de producció se li ha associat el privilegi 2.

Per definir els rols, cal accedir a la pantalla corresponent des del menú "Sistema\_rols" de CitectSCADA Editor de projectes:

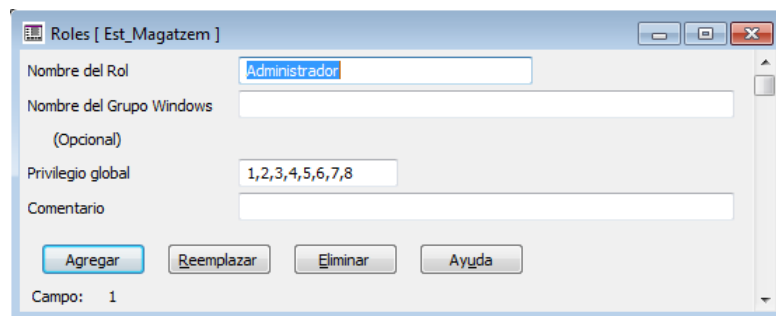


Figura 31. Pantalla d'agregació del rol *Administrador*

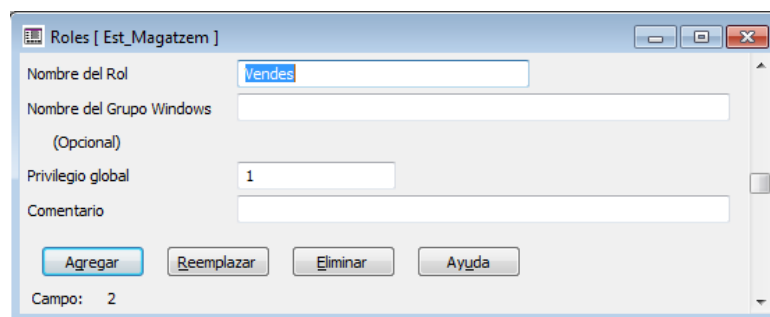


Figura 32. Pantalla d'agregació del rol *Vendes*

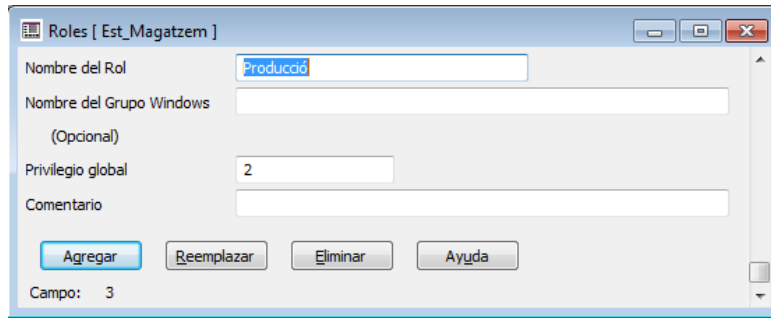


Figura 33. Pantalla d'agregació del rol *Producció*

Una vegada definits aquests rols, tots els elements d'aquest projecte seran accessibles entrant com a usuari amb rol d'*Administrador*. Els usuaris amb rol de *Vendes* només tindran accés a aquells elements en els que se li hagi associat un privilegi, 1 i els usuaris amb rol *Producció* només tindran accés a aquells elements en els que se li hagi associat un privilegi 2.

➤ Definició d'usuaris

Per cada projecte és necessari definir com a mínim un usuari.

En aquest projecte s'ha definit un usuari per cada un dels rols. Per aquest fet, des del menú "Sistema\_usuarios" de CitectSCADA Editor de projectes (veure figura 34, 35 i 36).

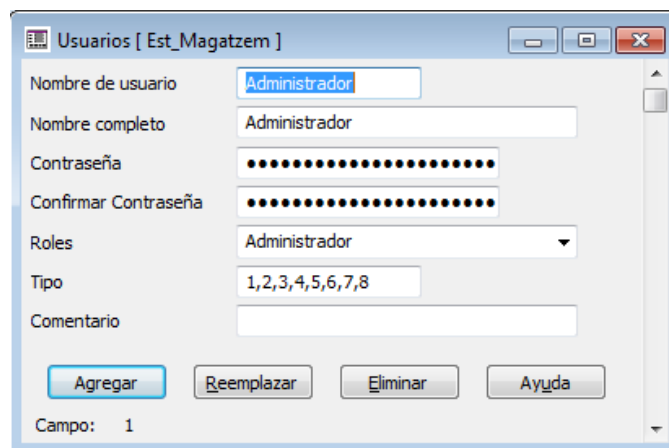


Figura 34. Pantalla d'agregació de l'usuari *Administrador*

En la pantalla que se'ns obre per tal de definir els usuaris, haurem d'introduir-hi:

- el nom de l'usuari (serà el nom que caldrà introduir en la pantalla de "log in" del *runtime* del CitectSCADA);
- el nom complet d'aquest usuari;
- una contrasenya d'accés;
- i els Rols que se li donen a aquest usuari.

Per defecte tots els usuaris tenen definit un rol amb el privilegi "zero". Això vol dir que tindran accés a tots aquells elements que no tinguin definida cap restricció per privilegis.

The screenshot shows a software window titled "Usuarios [ Est\_Magatzem ]". It contains several input fields and a dropdown menu. The "Nombre de usuario" field is filled with "Vendes". The "Nombre completo" field is filled with "Vendes". The "Contraseña" and "Confirmar Contraseña" fields are filled with 12 black dots each. The "Roles" dropdown menu is set to "Vendes". The "Tipo" field is filled with "1". The "Comentario" field is empty. At the bottom, there are four buttons: "Agregar", "Reemplazar", "Eliminar", and "Ayuda". The "Agregar" button is highlighted in blue. Below the buttons, it says "Campo: 2".

Figura 35. Pantalla d'agregació de l'usuari *Vendes*

The screenshot shows the same software window "Usuarios [ Est\_Magatzem ]". The "Nombre de usuario" field is filled with "Producció". The "Nombre completo" field is filled with "Producció". The "Contraseña" and "Confirmar Contraseña" fields are filled with 12 black dots each. The "Roles" dropdown menu is set to "Producció". The "Tipo" field is filled with "2". The "Comentario" field is empty. At the bottom, there are four buttons: "Agregar", "Reemplazar", "Eliminar", and "Ayuda". The "Agregar" button is highlighted in blue. Below the buttons, it says "Campo: 3".

Figura 36. Pantalla d'agregació de l'usuari *Producció*

#### 4.2.4.- Configuració del dispositiu de comunicació d'E/S

Per poder enllaçar les variables (*tags*) als seus corresponents objectes, hem de definir els anomenats *Dispositius d'Entrada/Sortida*.

Per la configuració dels dispositius de comunicació d'E/S el millor és fer-ho des de l'assistent que trobarem en la pantalla del CitectSCADA Explorer:

- seleccionem el nostre projecte;
- anem a la pestanya de "comunicaciones";
- fem doble clic en la icona de "configuración rápida de dispositivo E/S".

Per aquest projecte la comunicació entre el Vijeo Citect i el PLC SIEMENS S7-300 s'ha fet utilitzant un servidor OPC (veure apartat 6.- La comunicació entre PLC i PC via OPC). Per tant, la configuració del dispositiu extern de comunicació haurà d'anar en concordança amb el protocol corresponent (veure figura 37 i següents).



Figura 37. Primera pantalla assistent de creació dispositiu E/S

La figura 37 és la primera pantalla que se'ns obre de l'assistent de configuració. Aquí tan sols cal confirmar clicant a la pestanya "siguiete".

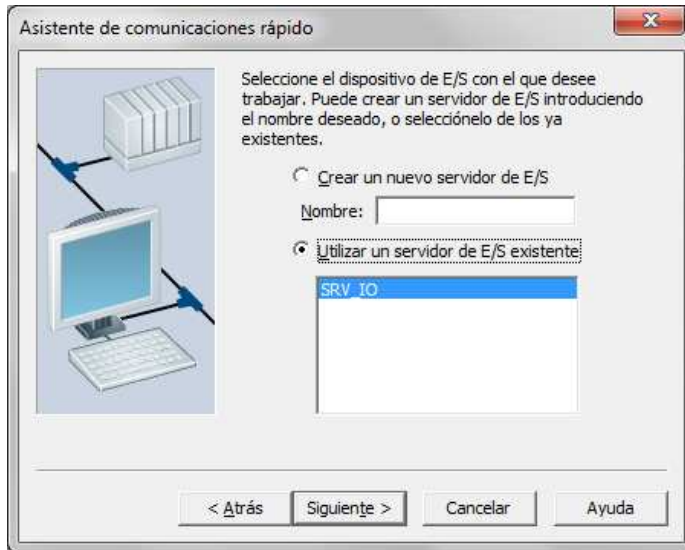


Figura 38. Segona pantalla assistent de creació dispositiu E/S

En la pantalla corresponent a la figura 38 ens permet seleccionar si volem crear un servidor nou d'E/S o si volem treballar amb un dels que hem definit prèviament i que ens surten llistats. Si no hem definit cap servidor, per defecte ens crearà un nou servidor que es dirà IOServer.



Figura 39. Tercera pantalla assistent de creació dispositiu E/S

En la següent pantalla de la figura 39 es crea pròpiament el dispositiu d'E/S amb el nom que li vulguem donar. Per defecte li dona el nom de IODev i els va numerant correlativament, començant sense cap número i seguint per l'1, 2...



Figura 40. Quarta pantalla assistent de creació dispositiu E/S

En la pantalla de la figura 40 ens permet seleccionar el tipus de dispositiu d'E/S que volem crear. En el nostre cas s'ha definit un dispositiu extern d'E/S per la comunicació amb el PLC via OPC. Cada *tag* de comunicació que associem a aquest dispositiu serà un *tag* que descomptarà de la llicència de què disposem.

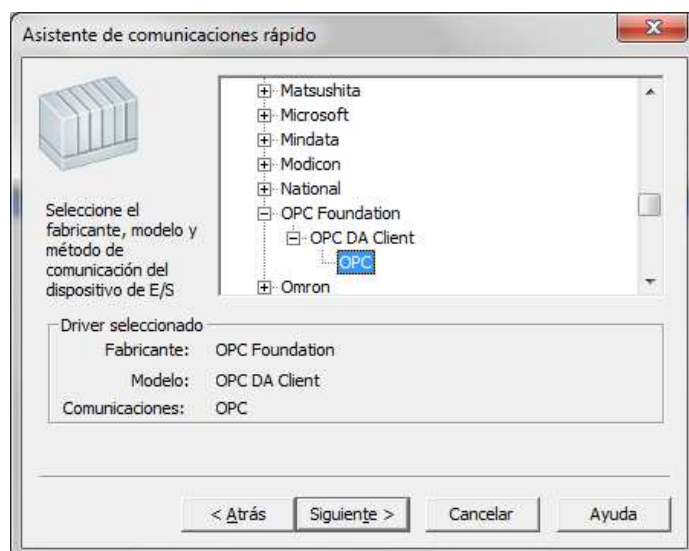


Figura 41. Cinquena pantalla assistent de creació dispositiu E/S

En la pantalla corresponent a la figura 41 haurem d'escollir el protocol de comunicació que caldrà utilitzar per comunicar el dispositiu amb l'element de



camp. En el nostre cas, escollim uns drivers OPC ja que ens comunicarem amb els elements de camp utilitzant aquesta via.



Figura 42. Sisena pantalla assistent de creació dispositiu E/S

En la pantalla de la figura 42 cal posar-hi l'adreça corresponent a l'OPC que utilitzem. En el nostra cas s'ha utilitzat l'OPC del fabricant IBH i l'adreça específica és *IBHSoftec.IBHOPC.DA* (veure l'apartat 6.- La comunicació entre PLC i PC via OPC).



Figura 43. Setena pantalla assistent de creació dispositiu E/S

La pantalla de la figura 43 la utilitzarem en cas de que vulguem vincular el dispositiu d'E/S amb una base de dades de *tags* externa. Pot ser d'utilitat en projectes amb gran quantitat de variables i que aquestes es trobin molt ben etiquetades i referenciades. Cal tenir en compte que si s'importen tots els *tags* d'una base de dades, gastarem tants *tags* de llicència com *tags* importem. En el nostre projecte deixem aquest tic sense activar.



Figura 44. Última pantalla assistent de creació dispositiu E/S

La última pantalla de l'assistent, corresponent a la figura 44, ens fa un resum del tipus de comunicació que acabem de crear i ens permet confirmar i crear el dispositiu clicant a "Finalizar".

#### 4.2.5.- Definició dels *tags* de variable d'aquest projecte

En aquest projecte s'han definit un total de 49 *tags* de variable del tipus enter, i un total de 10 *tags* de variable del tipus digital, tots ells associats al dispositiu extern. Per tant, en aquest projecte s'utilitzen 59 *tags* de comunicació entre el Vijeo Citect i el PLC que necessiten de llicència per tal d'executar el runtime indefinidament. Per altra banda, s'han definit un total de 20 variables locals per tal de dur a terme els càlculs i els processos necessaris pel bon funcionament del programa (veure Annex I).

Per tal de poder establir comunicació amb el PLC s'ha utilitzat un servidor OPC. En la definició del dispositiu de comunicació d'E/S, definit en l'apartat anterior, li hem donat l'adreça corresponent al servidor OPC. Per tant, ja li hem indicat al programa Vijeo Citect a quin servidor ha d'anar a buscar les dades. En la

definició dels tags de variable li hem de donar el nom de l'adreça definida en el servidor que comunica amb el PLC, que en aquest cas és "EstMagatzem" (veure apartat 6.- La comunicació entre PLC i PC via OPC). A partir d'aquí, hi introduïm directament l'àrea de memòria del PLC amb la que desitgem establir comunicació.

Per definir aquests *tags* utilitzats en aquest projecte es fa des del menú "Tags\_Tags de variable" (veure figura 45) o "Tags\_Variables locales"(veure figura 46) de CitectSCADA Editor de projectes.

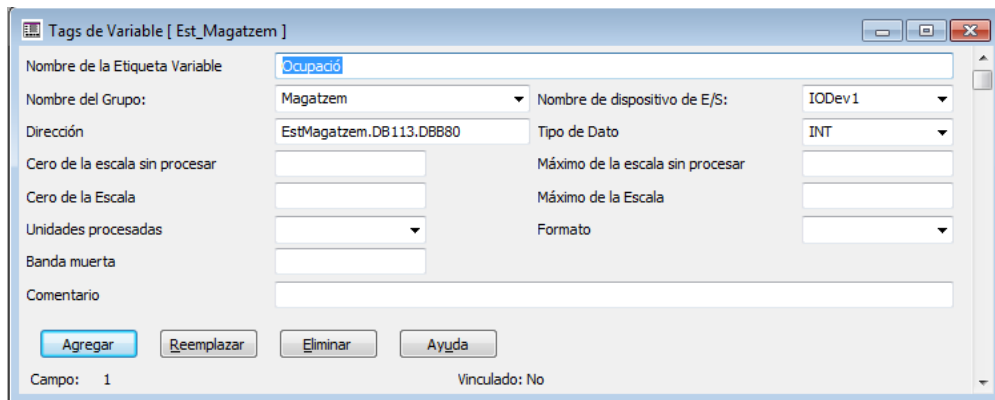


Figura 45. Exemple de definició d'un *Tag de variable* de comunicació

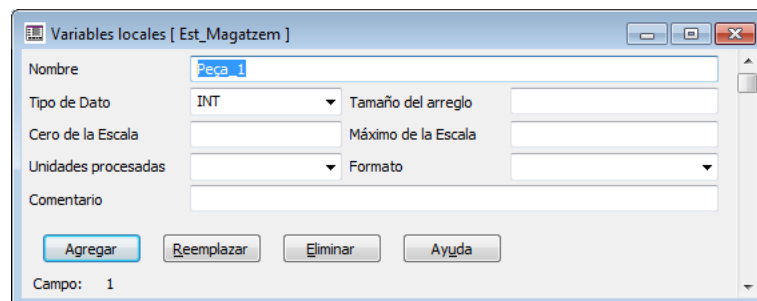


Figura 46. Exemple de definició d'una *Variable local*

#### 4.2.6.- La programació de les pantalles del procés

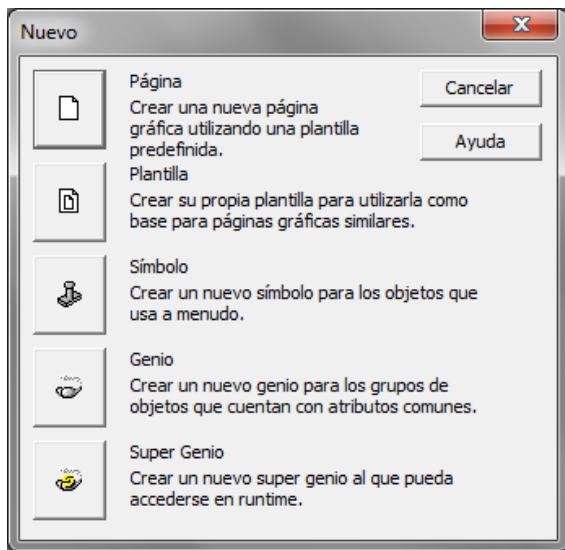
En el mode *runtime* del sistema caldrà dissenyar una sèrie de pàgines de gràfics que ens mostrin a la pantalla una "finestra del procés". En aquestes pàgines gràfiques es mostrarà l'estat en temps real dels diferents elements del procés. Aquests elements de la pantalla gràfica es coneixen com a objectes.

L'estructura de l'aplicació creada està formada per tres pantalles principals:

- Pantalla Magatzem. Es mostra l'esquema complet del magatzem així com la seva ocupació.
- Pantalla de comandes en procés. Es mostra la taula de la base de dades amb les comandes que cal servir i la taula amb les comandes ja servides.
- Pantalla d'entrada de comandes. Es mostra la taula de la base de dades amb les comandes a servir i permet l'entrada de noves comandes.

Per tal de crear una pàgina nova, anem al CitectSCADA Grafic Builder, i cliquem al menú "Archivo\_Nuevo".

Ens apareix una pantalla en la que tenim diferents opcions (veure figura 47).



- Crear una pàgina nova
- Crear una plantilla nova
- Crear un símbol nou
- Crear un Genie nou
- Crear un Super Genie nou

Figura 47. Pantalla per crear un nou element

En la pantalla que ens apareix, veure figura 48, ens permet escollir:

- L'estil de la plantilla que volem
- La pròpia plantilla
- La resolució de la pantalla

Si volem que la plantilla estigui vinculada a l'estil i amb la barra de títol o no.

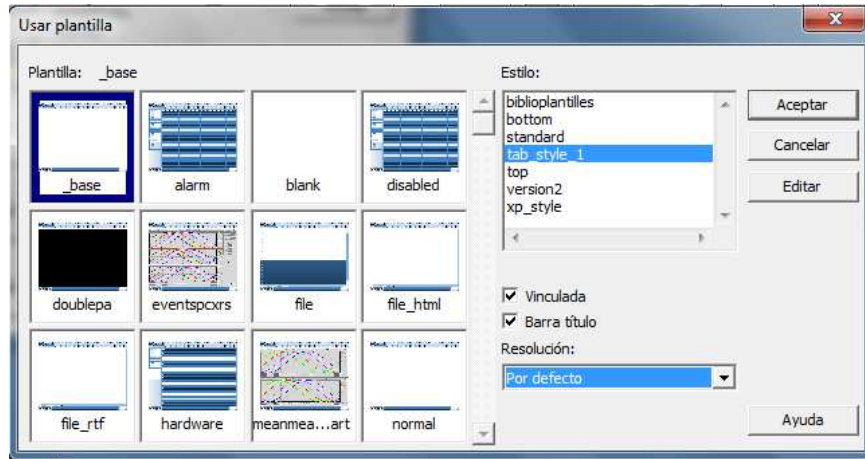


Figura 48. Pantalla per crear una nova pàgina

Una vegada hem escollit la plantilla per la nostra pàgina, anem al menú "Archivo\_Guardar como" i ens apareix el quadre de diàleg de la figura 49.

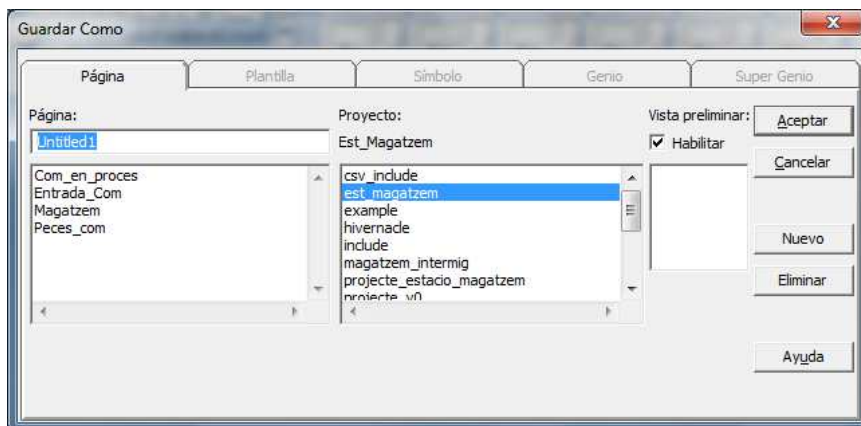


Figura 49. Pantalla per guardar la nova pàgina

En aquest quadre hem d'escollir el projecte dins el qual volem guardar la nostra pàgina i donar-li el nom corresponent.

Un cop fet això, ja podem passar a inserir objectes gràfics a la nostra pàgina.

#### 4.2.6.1.- Pantalla Magatzem

En aquesta pantalla es mostren les 40 posicions del magatzem distribuïdes tal i com estan configurades físicament en el procés i es mostra en tot moment la

seva ocupació. Cada una de les posicions del magatzem està associada a la seva corresponent àrea de memòria del PLC.

En aquesta pantalla també es pot veure l'estat en què es troba el procés (Manual/Automàtic):

- Si el procés es troba en estat manual, apareix un polsador que ens permet extreure manualment peces del magatzem per cada pulsació. L'ordre en què s'extrauran aquestes peces segueix la numeració de les posicions del magatzem, de manera que comença a extreure peces a partir de la posició 1, continua per la posició 2 i així va avançant (veure figura 51).
- Si el procés es troba en estat automàtic, es van extraient i paletitzant conjunts de tres peces seguint l'ordre de les comandes que es van llegint de la taula corresponent de la base de dades. En aquesta pantalla es pot observar en tot moment quina és la configuració del palet que s'està extraient i la conformació d'aquest en temps real (veure figura 50).

Una altra informació donada en aquesta pantalla principal és el contingut global del magatzem. Es mostra en una taula el total de peces de cada color de què es disposa en el magatzem així com la suma total de totes elles i el tant per cent d'ocupació del magatzem que això representa.

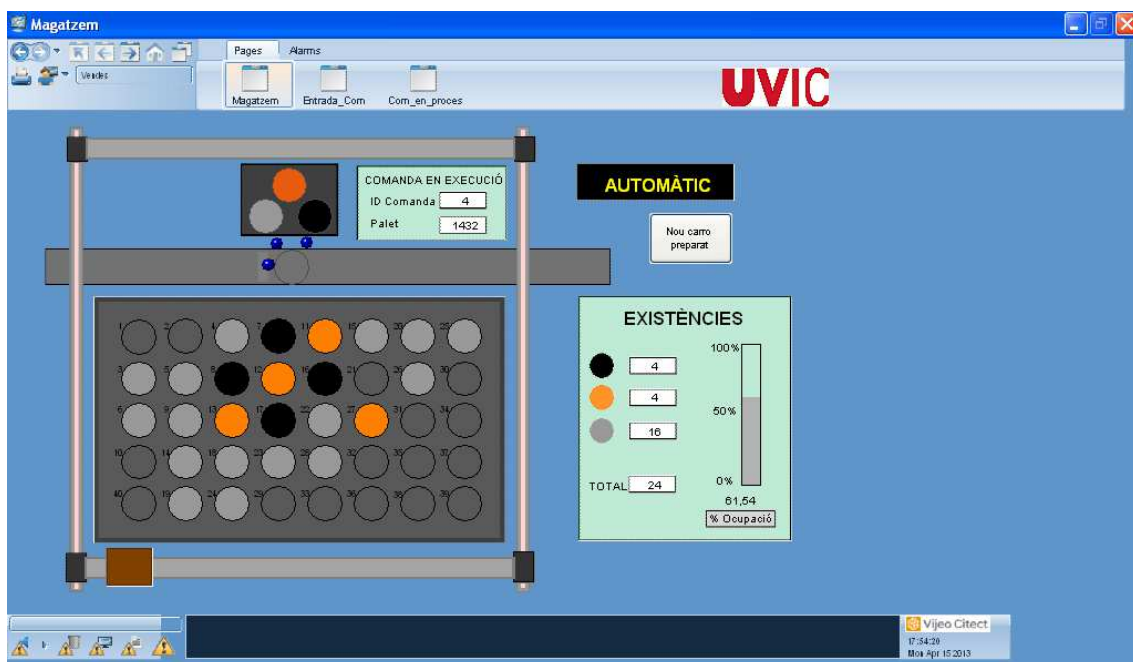


Figura 50. Pantalla gràfica de monitorització del magatzem en automàtic

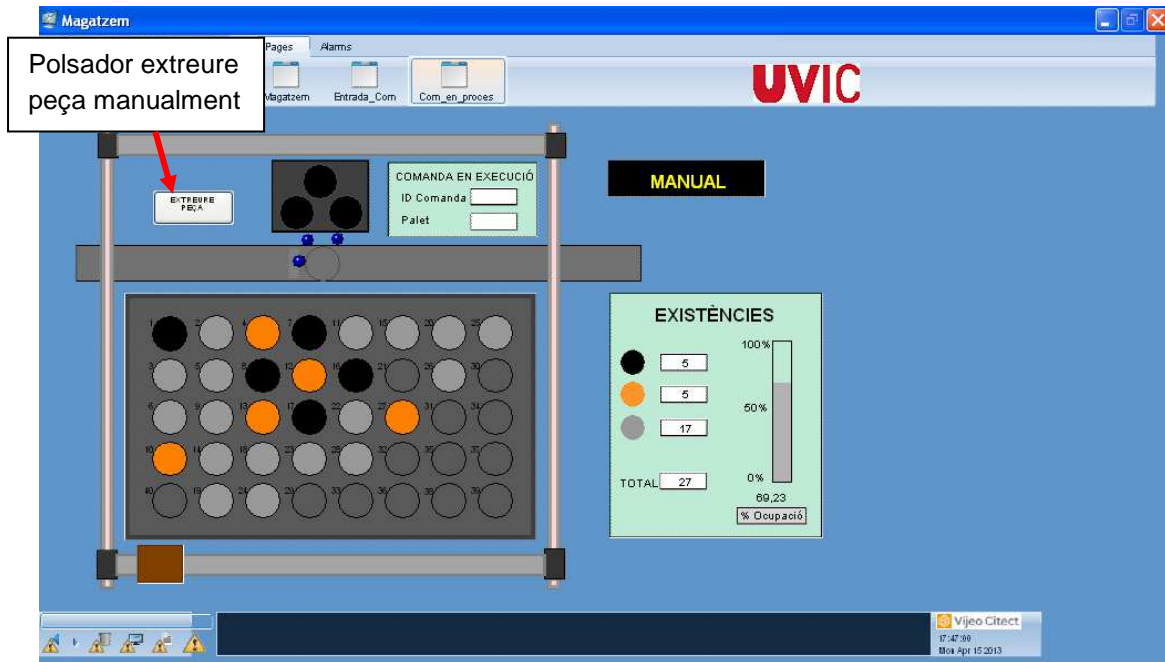


Figura 51. Pantalla gràfica de monitorització del magatzem en manual

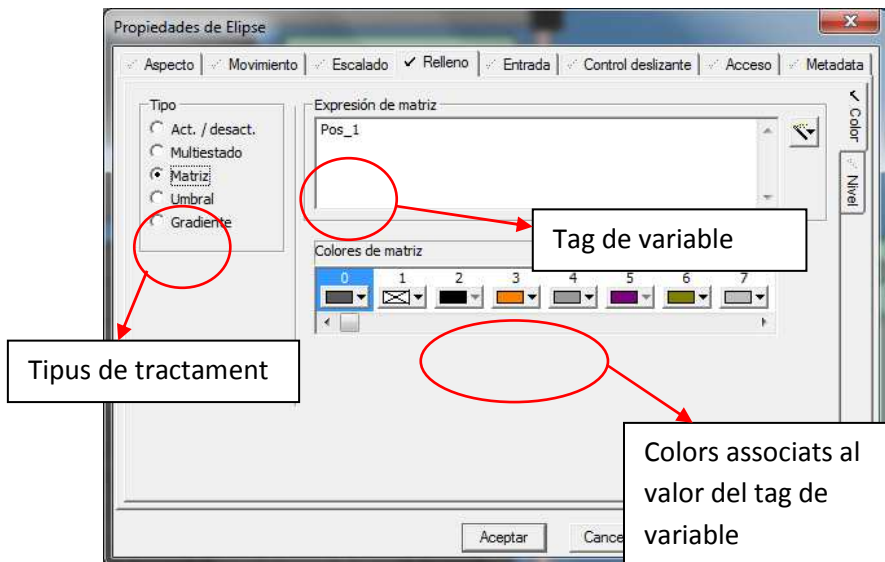


Figura 52. Pantalla de programació dels objectes circulars que figuren les peces del magatzem

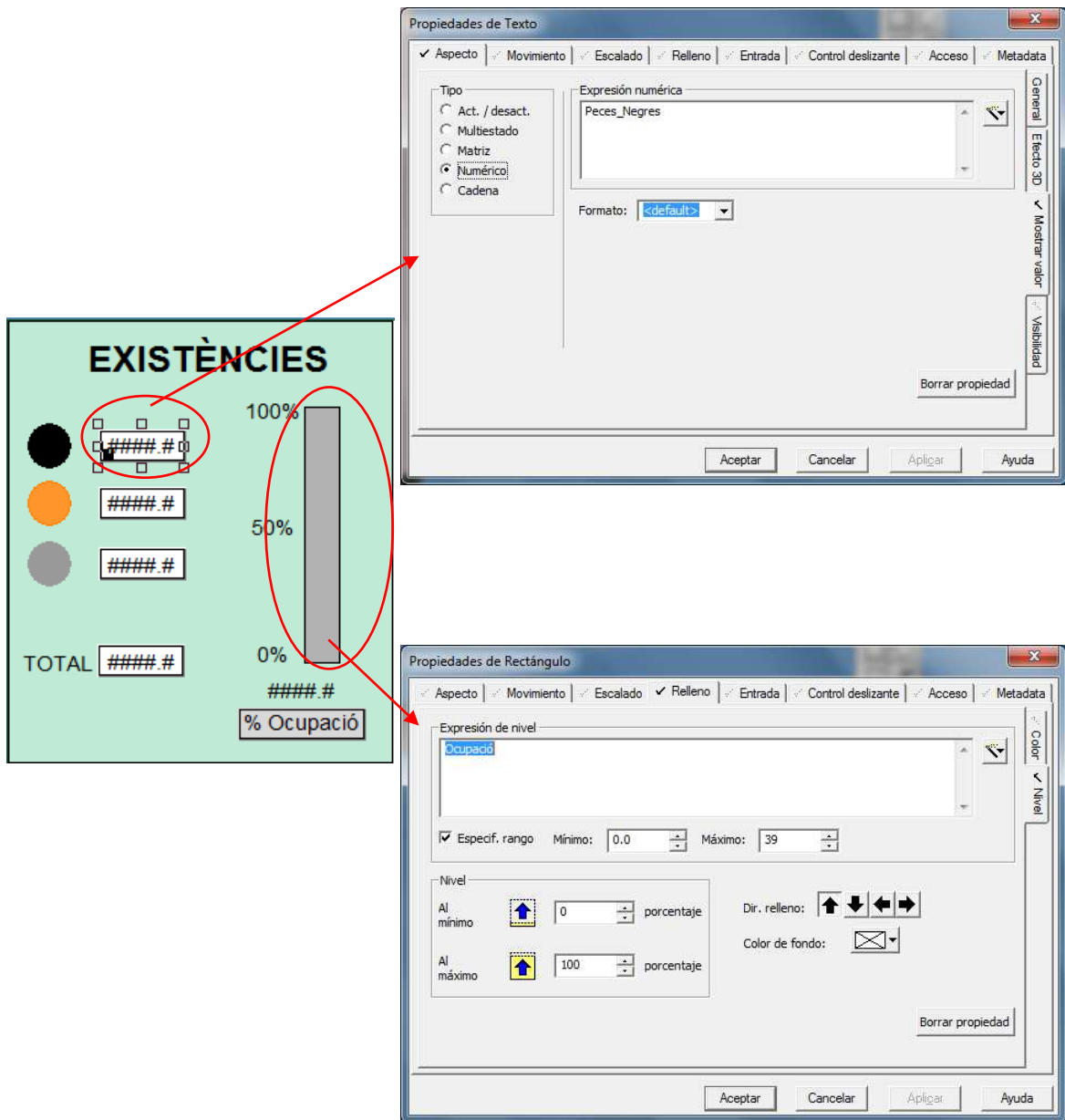


Figura 53. Pantalles de programació de l'indicador de nombre de peques negres i de la barra indicadora de l'ocupació del magatzem

En les figures 52, 53, 54 i 55 es mostren les pantalles de programació dels diferents objectes de la pantalla gràfica de monitorització del magatzem.



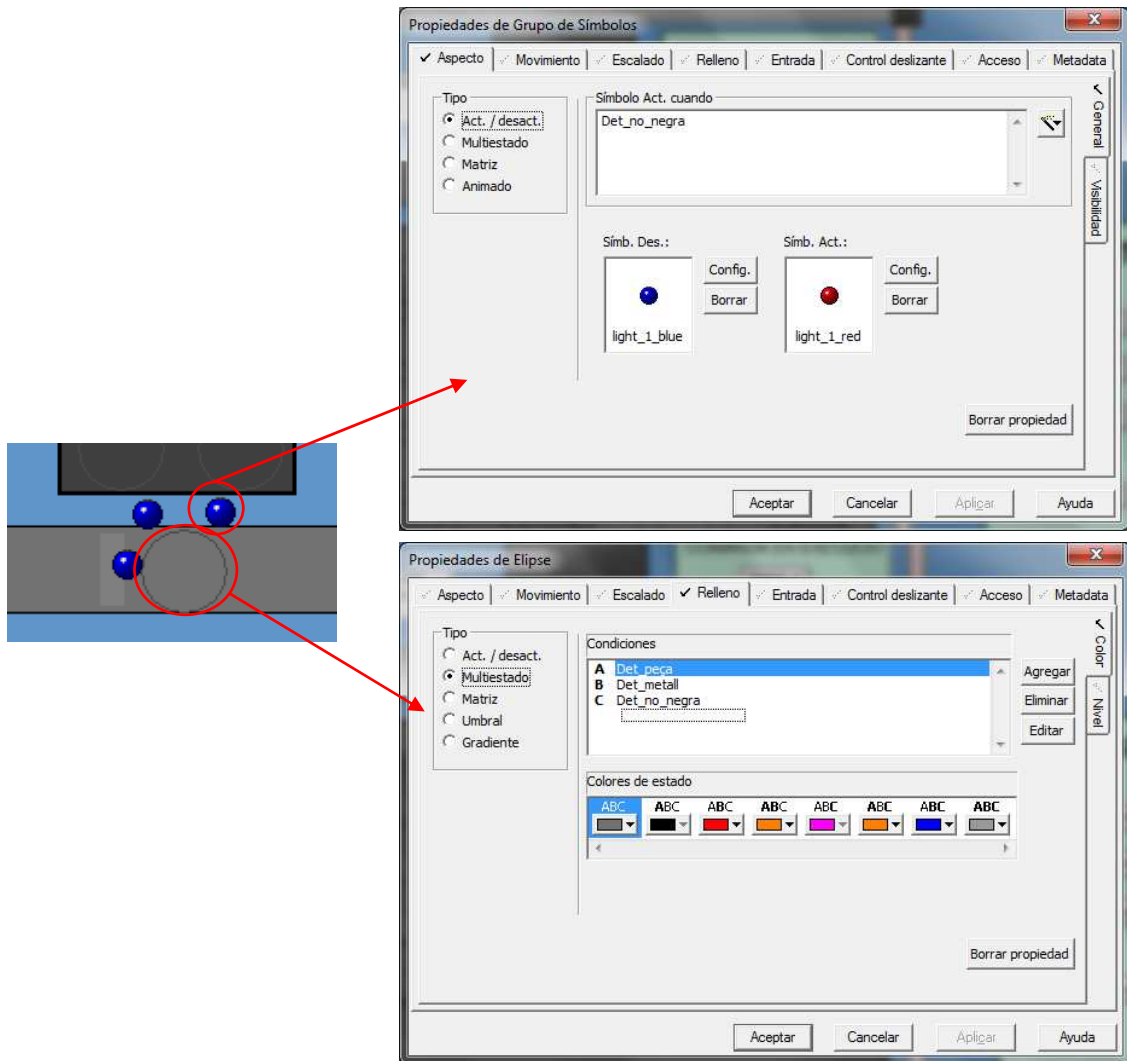


Figura 54. Pantalla de programació dels símbols que monitoritzen els detectors d'entrada de peça i de la determinació del color en funció de l'estat dels detectors.

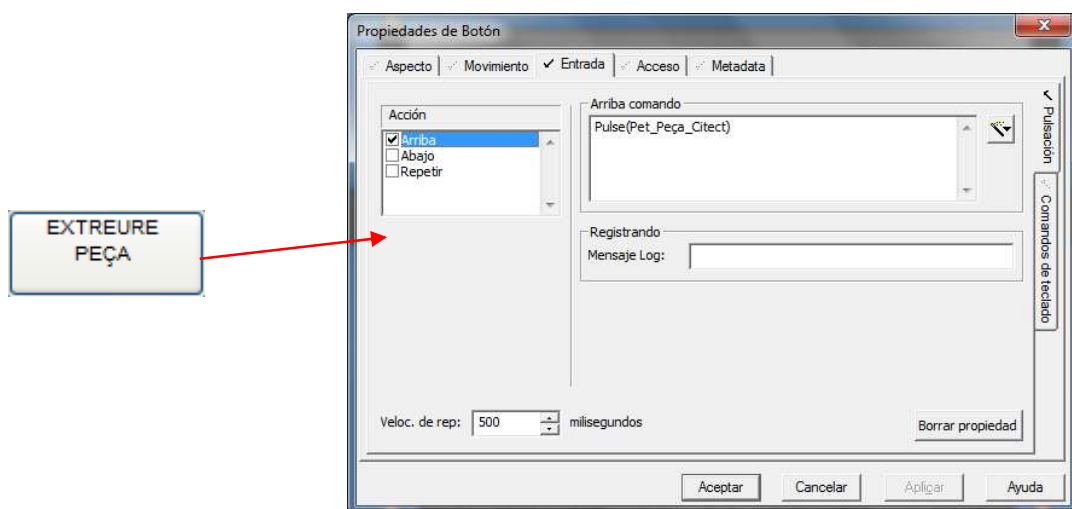


Figura 55. Pantalla de programació d'un boto que dóna un flanc de pujada

#### 4.2.6.2.- Pantalla de comandes en procés

En aquesta pantalla hi ha una primera capçalera en la que es mostra quina és la comanda que s'està executant en cas de que el procés estigui en mode automàtic (veure figura 56). Sota aquesta capçalera hi ha:

- una primera taula en la que es mostra la llista de comandes entrades de la base de dades d'Access i que caldrà anar executant seqüencialment,
- una segona taula, també de la base de dades Access, on es van registrant les comandes que ja s'ha executat. La lectura i escriptura d'aquestes dades es fa via ODBC amb la base de dades d'Access (veure apartat 5.2.- L'intercanvi de dades entre el sistema SCADA i l'Access).

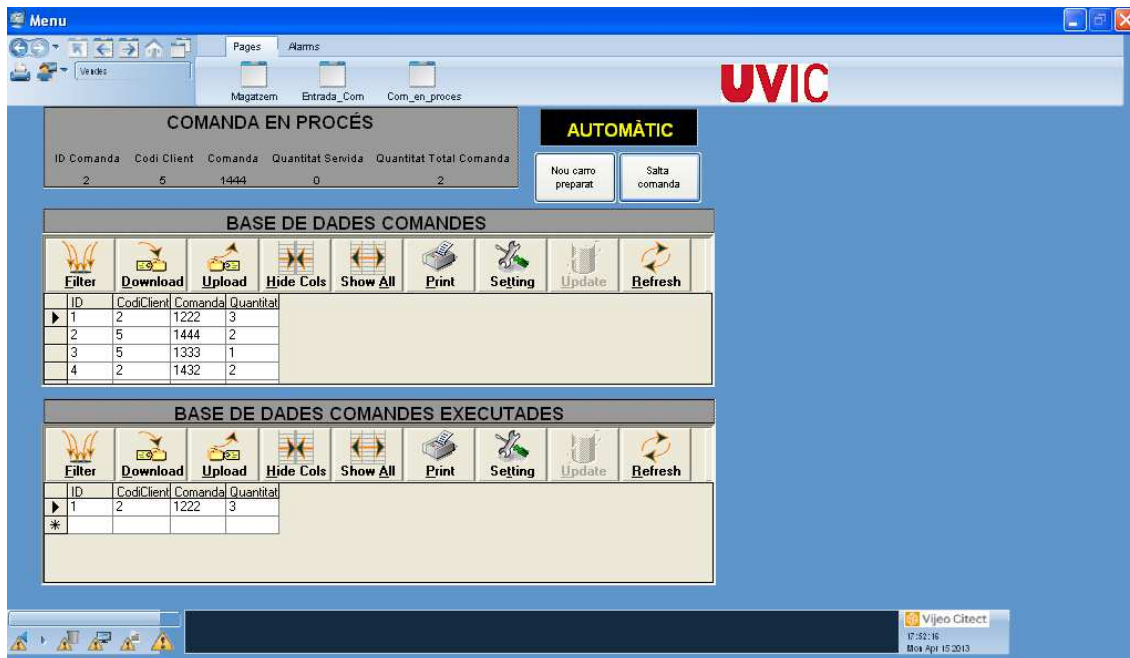


Figura 56. Pantalla gràfica de monitorització de comanda en procés

En aquesta pantalla també hi ha dos botons: un botó permet saltar la comanda que s'està executant (p.e. en cas de que hi hagi falta d'existències en el magatzem d'alguna de les peces que ha de formar del palet); i un altre botó que, una vegada executada una comanda, permet confirmar que el palet ha estat retirat i que podem executar la següent.

#### 4.2.6.3.- Pantalla d'entrada de comandes

En aquesta pantalla hi ha una primera capçalera en la que es permet a l'usuari entrar les comandes que passaran directament a la cua de la taula corresponent de la base de dades d'Access i, per tant, passaran també a la cua de producció (veure figura 57).

Per tal de poder entrar una comanda en la taula de la base de dades cal que l'usuari disposi de privilegis en el rol de "Vendes". D'aquesta manera, no qualsevol operari pot entrar comandes en la base de dades sinó que només aquells usuaris que estiguin donats d'alta en el sistema amb un rol de "Vendes" disposaran del nom d'usuari i contrasenya adequats per poder-les entrar.

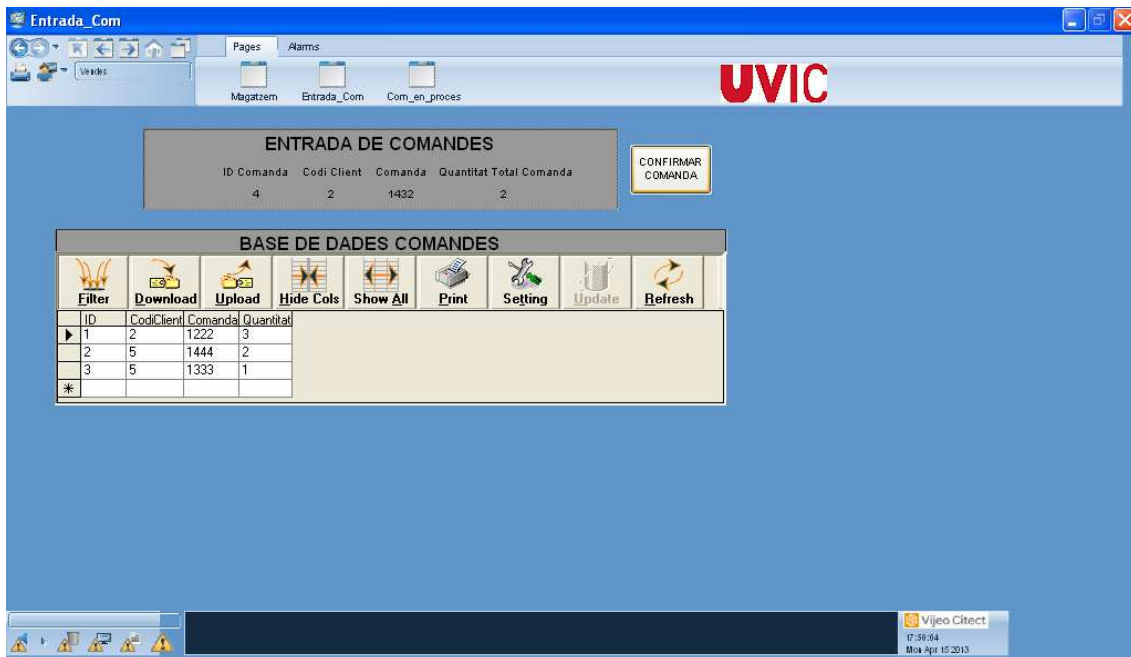


Figura 57. Pantalla d'entrada de comandes.

Cada una de les pàgines gràfiques editades amb el CitectSCADA Grafic Builder disposa d'unes propietats configurables que s'hi pot accedir amb el botó dret del mouse. En aquesta pantalla de propietats de pàgina es pot programar qualsevol acció que desitgem que s'executi en entrar-hi. En accedir a la pàgina d'entrada de comandes, s'ha programat que es mostri un missatge en el que s'indica a l'usuari que només es poden entrar comandes amb els privilegis de

"Vendes". Això es fa clicant amb el botó dret sobre la pantalla gràfica i accedint al menú "Propiedades de la Pàgina" (veure figures 58 i 59).

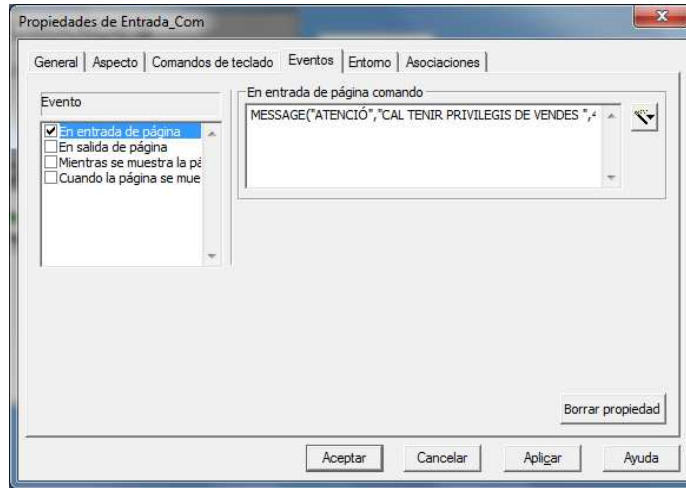


Figura 58. Pantalla de configuració d'un event en entrar a la pàgina d'entrada de comandes.



Figura 59. Pantalla d'avís que es mostra en entrar a la pàgina d'entrada de comandes.

Una vegada s'ha accedit a la pàgina amb el "login" corresponent, en primer lloc el programa ens proporciona automàticament un Identificador de comanda que correspon a la següent línia buida de la taula de comandes de la base de dades. A partir d'aquí, el sistema ens permet entrar el codi del client que demana la comanda, el tipus de comanda demanada i la quantitat desitjada.

Clicant sobre l'objecte numèric corresponent al codi del client, se'ns obrirà una pantalla emergent en la que hi podrem entrar el valor requerit (veure figura 60).

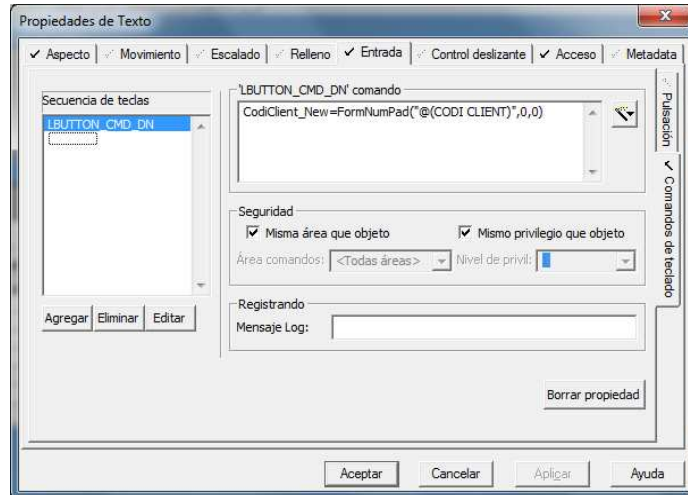


Figura 60. Programació de la pantalla emergent en clicar sobre l'objecte numèric d'entrada del CodClient.

En la pestanya "Acceso" de la mateixa pantalla és on restringim que només hi puguin escriure aquells usuaris amb un nivell de privilegis definit com a "Vendes" o "Administrador" (veure figura 61).

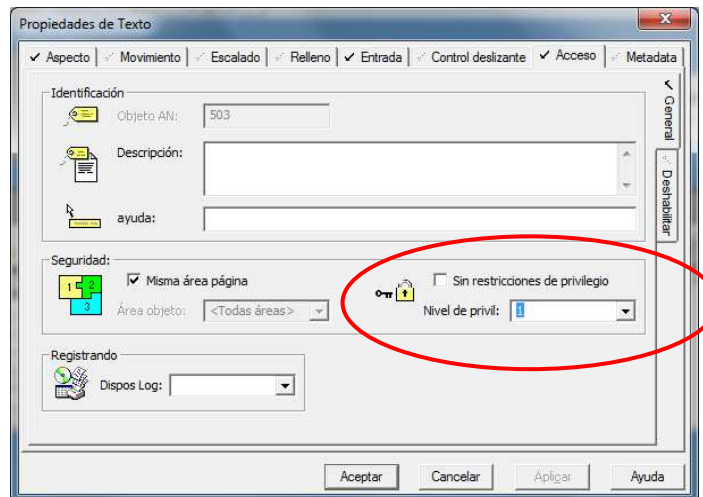


Figura 61. Programació de la restricció d'entrada segons privilegi.

Per tal de poder entrar el tipus de comanda, tal i com està establert, una peça negra correspon al codi 2, una peça taronja correspon al codi 3 i una peça metàl·lica correspon al codi 4. Per tal de poder agilitzar l'entrada del tipus de

comanda i fer-ho més senzill de cara a l'usuari, evitat falses entrades, s'ha programat una pantalla emergent en clicar sobre l'objecte numèric corresponent de manera que, en pulsar sobre l'objecte d'entrada de tipus de comanda, se'ns obre una pantalla en la que hi ha els tres tipus de peça que es pot demanar i tan sols cal fer doble clic sobre les tres peces desitjades que volem que formin part del palet (veure figura 62).

A partir d'aquí, el sistema tradueix el codi de colors escollit en el codi numèric necessari per transferir al PLC.

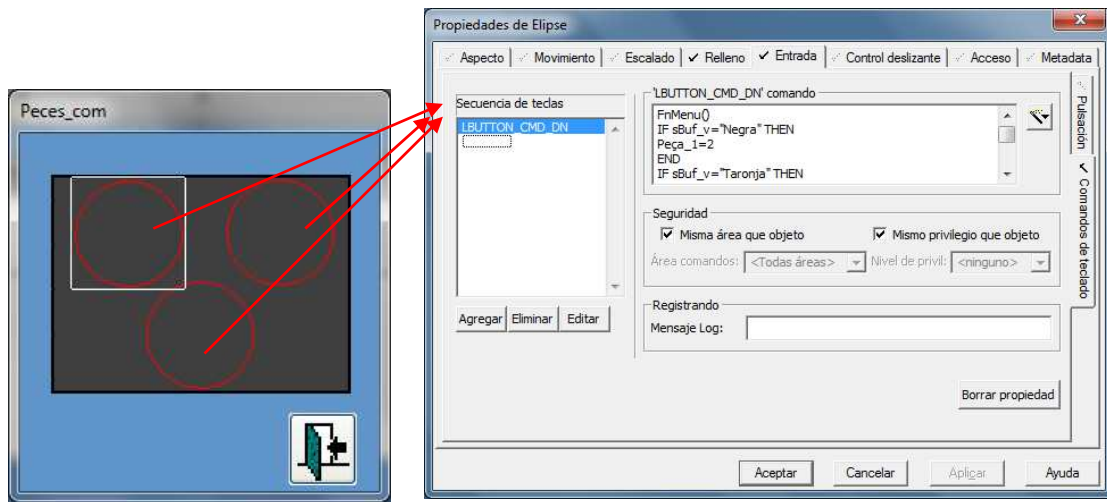


Figura 62. Pantalla emergent que permet escollir les peces de la comanda

En clicar sobre cada una de les peces en blanc es fa una crida a la funció CICODE "FnMenu()" (programada amb l'editor que ens ofereix el mateix software; veure l'apartat 4.3.7.- Les funcions CICODE). Aquesta funció mostra una pantalla amb les opcions dels tres colors de peça disponible i ens retorna el color que l'usuari ha escollit fent-t'hi doble clic a sobre. A partir d'aquí, i mitjançant el codi que es mostra a la figura 63, s'actualitza la variable corresponent traduint el color al valor numèric que té assignat.

```

IF sBuf_v="Negra" THEN
Peça_1=2
END
IF sBuf_v="Taronja" THEN
Peça_1=3
END
IF sBuf_v="Metall" THEN
Peça_1=4
END
    
```

Figura 63. Codi de programa per traduir el color seleccionat en el número corresponent.

Una vegada s'ha escollit el color de cada una de les peces, en sortir de la finestra emergent es construeix el codi de complet i ordenat per tal de poder-lo transferir al PLC (veure figura 64).

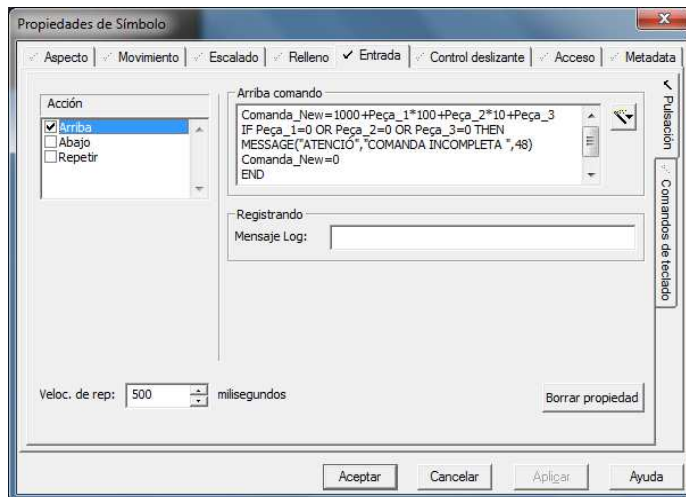


Figura 64. Pantalla de programació del botó per tancar la pantalla emergent

Com s'ha comentat, el codi a transferir està format per el nombre 1 seguit dels tres colors de les tres peces que han de formar part del palet. Per tant, el codi de programa per formar aquesta codificació és el que es mostra en la figura 65.

```

Comanda_New=1000+Peça_1*100+Peça_2*10+Peça_3
IF Peça_1=0 OR Peça_2=0 OR Peça_3=0 THEN
MESSAGE("ATENCIÓ", "COMANDA INCOMPLETA ",48)
Comanda_New=0
END
Winfree() // Tanca la finestra
    
```

Figura 65. Codi de programa per generar la codificació del palet

Finalment, l'objecte numèric corresponent a la quantitat que es vol de la comanda es programa igual que l'objecte numèric d'entrada de CodiClient.

Una vegada entrades totes les dades corresponents a la comanda, cal que s'activi el polsador de confirmació de comanda per tal que les dades introduïdes es registrin en la taula corresponent de la base de dades.

**L'intercanvi de dades entre el Citect SCADA i la base de dades Access es desenvolupa específicament en l'apartat 5 i següents.**

#### 4.2.7.- Les funcions CICODE

El llenguatge CICODE és un llenguatge de programació similar als llenguatges Visual Basic o C++, dissenyat específicament per la monitorització d'instal·lacions i el control d'aplicacions. El programa Vijeo Citect disposa d'un editor en llenguatge CICODE i permet generar funcions que llavors poden ser cridades i executades des de qualsevol pàgina gràfica.

En aquest programa s'han generat cinc fitxers CICODE, en els quals s'han definit les corresponents funcions en cada un d'ells:

- ID\_Peces. Fitxer on es genera la funció que permet construir la codificació de les peces que s'estan extraient en mode automàtic per tal de monitoritzar-les en la pantalla "*Magatzem*" (veure figura 66).

```

FUNCTION ID_Peces
IF Cnt_Sortides=1 THEN
Camisa_A_CITECT=Sortida_C;
Camisa_B_CITECT=0;
Camisa_C_CITECT=0;
END
IF Cnt_Sortides=2 THEN
Camisa_A_CITECT=Sortida_B;
Camisa_B_CITECT=Sortida_C;
Camisa_C_CITECT=0;
END
IF Cnt_Sortides=3 THEN
Camisa_A_CITECT=Sortida_A;
Camisa_B_CITECT=Sortida_B;
Camisa_C_CITECT=Sortida_C;
END
END
    
```

Figura 66. Fitxer que genera la funció ID\_Peces



La variable *Cnt\_Sortides* agafa els valors 1, 2 o 3 en funció del nombre de peces que ja s'han extret, en mode automàtic. Les variables *Sortida\_A*, *Sortida\_B* i *Sortida\_C* configuren un pila FIFO que es va actualitzant per cada peça que surt.

- *Triar\_Peça*. Fitxer on es genera la funció *FnMenu()*, que permet crear una llista amb els colors de les peces que l'usuari pot escollir a l'hora d'entrar una comanda (negre, vermell o metall) i retorna el color escollit (veure figura 67).

```
//Funció per crear un desplegable per escollir el tipus de peces
de la comanda
STRING FUNCTION FnMenu()
STRING sBuf;
//crea y configura la mida del combo
INT hForm = FormNew("ESCULL PECES", 20, 7, 1);
//crea la llista del combo
INT hField = FormListBox(2,0,15,4,sBuf);
//paràmetres a escollir
FormAddList("Negra");
FormAddList("Taronja");
FormAddList("Metall");
//espera a que l'usuari seleccioni una resposta amb doble clic
FormRead(0);
//guarda la selecció en una variable global
sBuf_v=sBuf;
RETURN sBuf_v;
END
```

Figura 67. Fitxer que genera la funció *ID\_Peces*

- *TaulaComandes* (veure figura 68). Fitxer on es genera la funció *GetComanda()* que permet llegir del dispositiu "Comandes" (Taula 1 de la base de dades d'Access) la nova comanda a executar per transferir-la al PLC i, a la vegada, permet escriure al dispositiu "SetComandes" (Taula 2 de la base de dades d'Access) l'última comanda executada

(veure apartat 8.- L'intercanvi de dades entre el sistema SCADA i l'Access (via ODBC)).

```

//Definició de variables
INT nnID;
INT hDev;
INT hDev1;
STRING nComanda;
STRING nCodiClient;
STRING nQuantitat;
INT NombreComand=0;
INT write;

//Funció per llegir dl dispositiu "Comandes" (Taula1) la nova comanda
a executar
//i escriure al dispositiu "SetComandes"(Taula2) la última comanda
executada
FUNCTION
GetComanda()

//Indexa el cursor a la primera posició de la Base de Dades
IF nID=0 THEN nID=1 END;

ComandaActual=nID;

//Obre les Bases de Dades "Comandes" i "SetComandes"
hDev=DevOpen("Comandes");
hDev1=DevOpen("SetComandes");

    IF hDev>= 0 THEN

        //Si el PLC està en mode automàtic i no s'està processant cap
        comanda, llegeix nova comanda
            IF Mode_Auto=-1 AND Sortida_auto=0 THEN

                //Escriu la última comanda executada en la taula de
                comandes executades "SetComandes"
                    IF nID> 0 AND write=1 THEN
                        IF hDev1>= 0 THEN

                            DevWrite(hDev1,nnID);
                            DevWrite(hDev1,nCodiClient);
                            DevWrite(hDev1,nComanda);
                            DevWrite(hDev1,nQuantitat);
                            DevClose (hDev1);
                            Cnt=StrToInt(nQuantitat)+1;
                            write=0;

                        END
                    END
            END
    END

```

```

//Si no s'ha arribat al final de la taula de comandes, llegeix la
següent comanda a executar
    IF DevFind(hDev, nID, "ID") = 0 THEN

        Cnt=0;

        //Llegeix la quantitat de la comanda i ho
transforma a enter per tal de tractar-ho com a tal
        nQuantitat=DevGetField(hDev,"Quantitat");
        NombreComand=StrToInt(nQuantitat);
        Quantitat=StrToInt(nQuantitat);

        //Llegeix el codi de la comanda a exectar i el
codi del client corresponent i ho transforma a enter
        nComanda = DevGetField(hDev,"Comanda");
        nCodiClient = DevGetField(hDev,"CodiClient");
        Exec_Comand=StrToInt(nComanda);
        Comanda_PLC=Exec_Comand;
        CodiClient=StrToInt(nCodiClient);

        IDS=nID;

        write=1;

        //Bucle per tal d'executar el nombre de palets
requerits d'una mateixa comanda

        WHILE NombreComand>=1 DO

            IF Sortida_auto=0 THEN

                Pulse(Processant_PLC);
                Sortida_auto=1;
                NombreComand=NombreComand-1;
                NombredeComanda=NombreComand;
                Cnt=Cnt+1;
                Sleep(2);

            END

        END

        nnID=nID;
        nID=nID+1;

        DevClose (hDev);

    END
END
END
END

```

Figura 68. Fitxer que genera la funció GetComanda()

- Nova\_Comanda (veure figura 69). Fitxer on es genera la funció *SetComanda()* que permet escriure al dispositiu "Comandes" (Taula 1 de la base de dades d'Access) la nova comanda entrada per l'usuari en la pantalla corresponent (veure apartat 8.- L'intercanvi de dades entre el sistema SCADA i l'Access (via ODBC)).

```

//Definició de variables
INT hDev2;
INT IDD_New;
INT CodiClient_New_Set;
INT Comanda_New_Set;
INT Quantitat_New_Set;

//Funció per escriure a la Base de Dades "Comandes"(Taula 1) la nova
comanda entrada per l'usuari
FUNCTION
SetComanda()

IDD_New=ID_CC;
CodiClient_New_Set=CodiClient_New;
Comanda_New_Set=Comanda_New;
Quantitat_New_Set=Quantitat_New;

//Obre el dispositiu Base de Dades "Comandes"
hDev2=DevOpen("Comandes");

    IF hDev2>= 0 THEN

        // Escriu les variables definides per l'usuari a la Base de
Dades "Comandes"
        DevWrite (hDev2,IDD_New);

        DevWrite (hDev2,CodiClient_New_Set);

        DevWrite (hDev2,Comanda_New_Set);

        DevWrite (hDev2,Quantitat_New_Set);

        DevClose (hDev2);

    END
END

```

Figura 69. Fitxer que genera la funció SetComanda()

- Copia\_comandes (veure figura 70). Fitxer on es genera la funció *CopyComanda()* que permet situar el cursor utilitzat en la funció *GetComanda()* en la última posició del dispositiu "Comandes" (Taula 1 de la base de dades d'Access).

```

//Definició de variables
INT ID_C;
INT hDev3;

//Funció per situar el cursor ID_C a la última posició de la taula
"Comandes" per permetre la següent entrada
FUNCTION
CopyComanda()

//Indexa el cursor a la primera posició de la Taula1 de la Base de
Dades
IF ID_C=0 THEN ID_C=1 END;

hDev3=DevOpen("Comandes");

    //Indexa el cursor fins a la última posició de la Taula 1 de la
Base de Dades
    IF hDev3>= 0 THEN
        IF DevFind(hDev3, ID_C, "ID") = 0 THEN

            ID_C=ID_C+1;
            ID_CC=ID_C;
            DevClose (hDev3);

        END
    END
END

```

Figura 70. Fitxer que genera la funció CopyComanda()

Les funcions de *CopyComanda()*, *GetComanda()* i *ID\_Peces()* cal que siguin executades permanentment mentre s'executa el programa en runtime. Per tal de definir aquestes accions s'ha programat un "Event" per l'execució de cada una d'elles. Un "Event" o succés del sistema permet activar una acció o executar una sèrie d'ordres (veure figura 71).

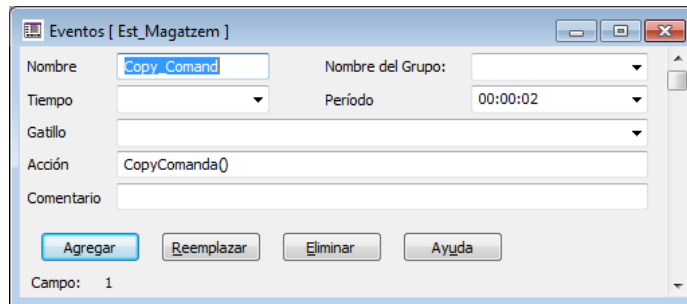



Figura 71. Pantalla de programació de l'event per executar *CopyComanda()*

Perquè els "events" funcionin han d'habilitar-se mitjançant l'assistent de configuració de l'ordinador (veure apartat 4.3.8.- Configuració de l'ordinador).

#### 4.2.8.- Configuració de l'ordinador

Abans d'executar el projecte en runtime s'haurà de configurar cada equip utilitzant l'assistent que ens proporciona el Vijeo CitectSCADA. La informació de configuració s'emmagatzema en cada màquina en un arxiu Citect.ini. Aquest assistent s'ha d'executar en cada equip per tal de configurar apropiadament el Vijeo CitectSCADA per cada màquina en particular. Cal executar aquesta configuració després de compilar el projecte i ha de ser l'últim pas abans d'executar el runtime.

Per tal d'executar l'assistent de configuració de l'ordinador, es pot fer des de la pantalla del Vijeo CitectSCADA Explorer, en el menú "*Herramientas-Asistente de configuración del ordenador*" o bé des de qualsevol de les tres pantalles del CitectSCADA amb la icona corresponent . Amb aquesta icona s'inicia un assistent de configuració. En les figures de la 72 a la 85 es mostren les diferents pantalles de l'assistent de configuració de l'ordinador.

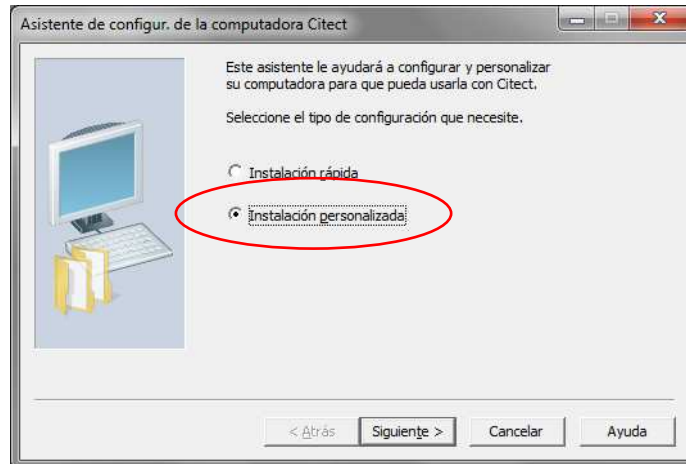


Figura 72. Primera pantalla de l'assistent

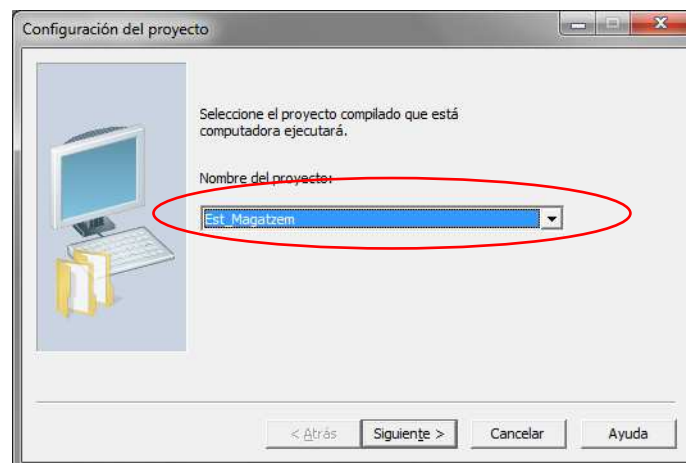


Figura 73. Segona pantalla de l'assistent

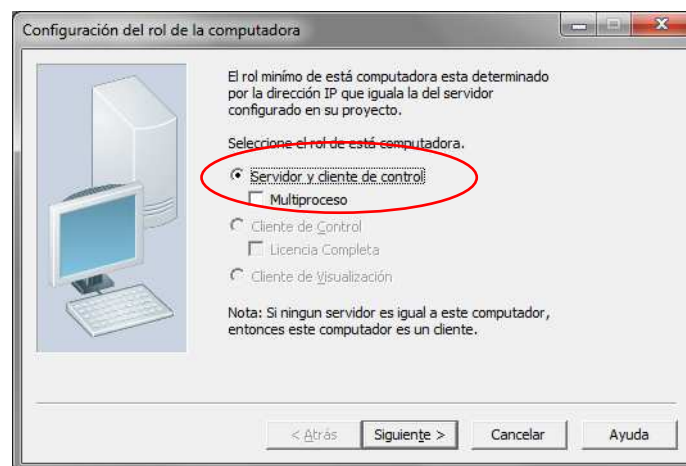


Figura 74. Tercera pantalla de l'assistent

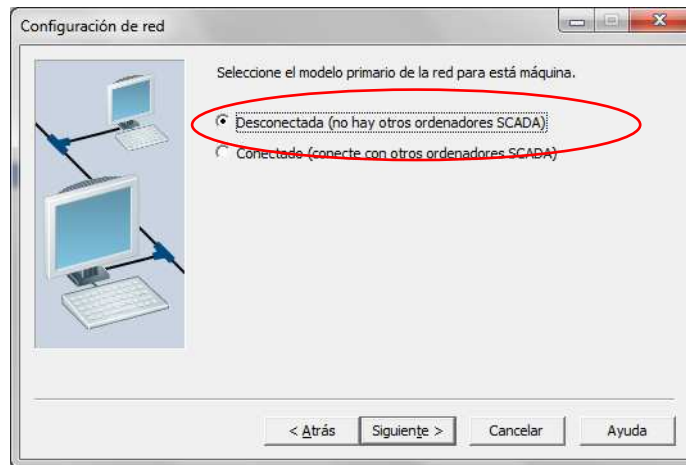


Figura 75. Cuarta pantalla de l'assistent

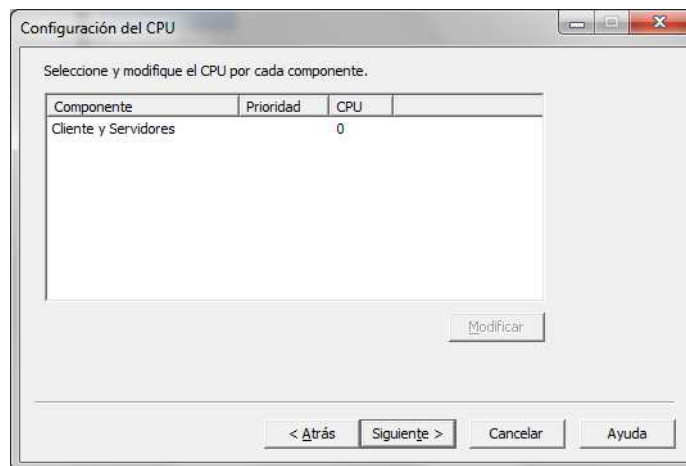


Figura 76. Cinquena pantalla de l'assistent

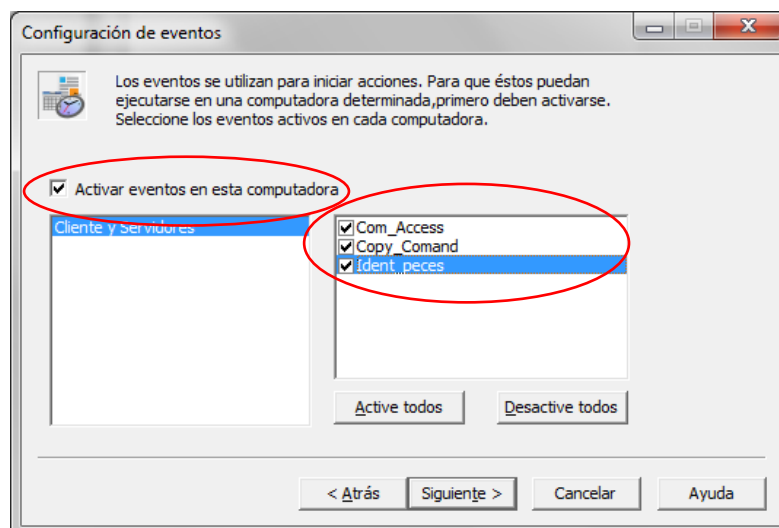


Figura 77. Sisena pantalla de l'assistent



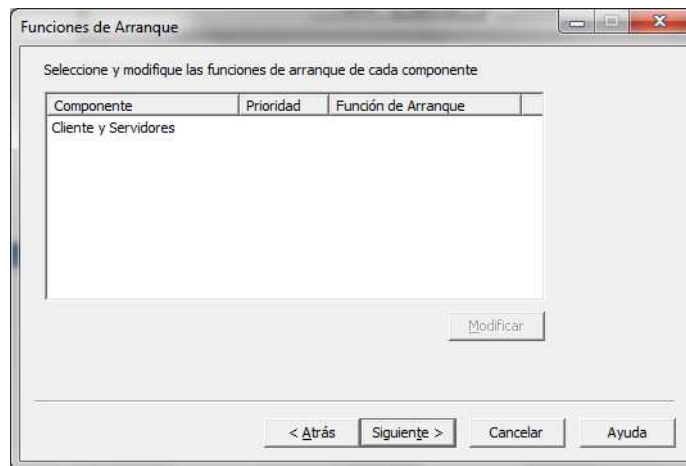


Figura 78. Setena pantalla de l'assistent

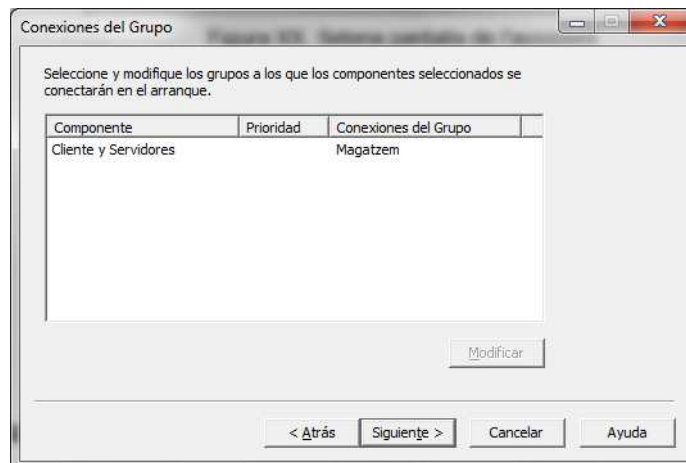


Figura 79. Vuitena pantalla de l'assistent

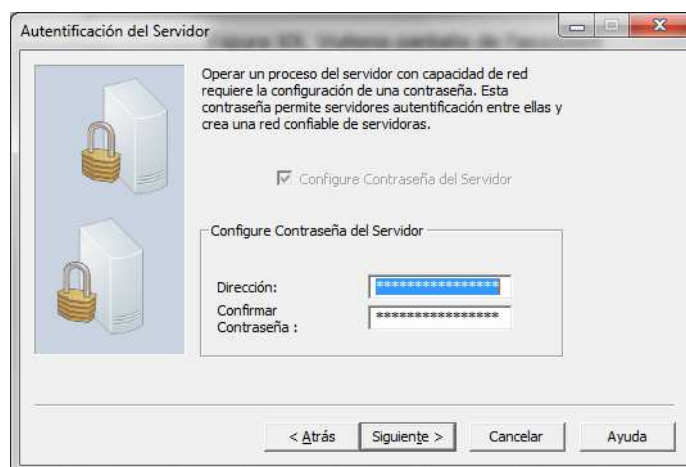


Figura 80. Novena pantalla de l'assistent



Figura 81. Desena pantalla de l'assistent



Figura 82. Onzena pantalla de l'assistent

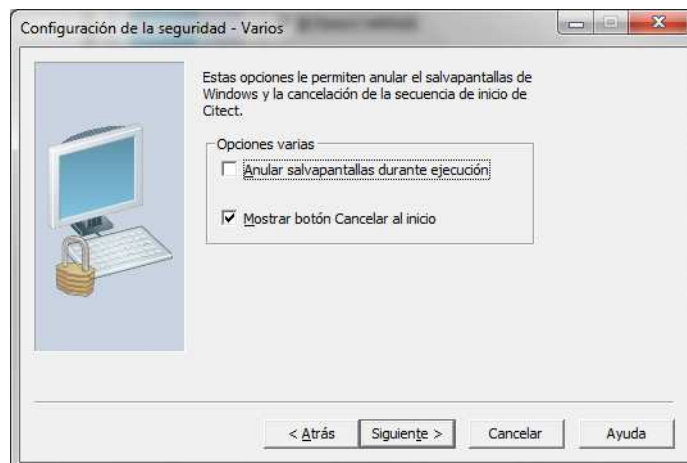


Figura 83. Dotzena pantalla de l'assistent

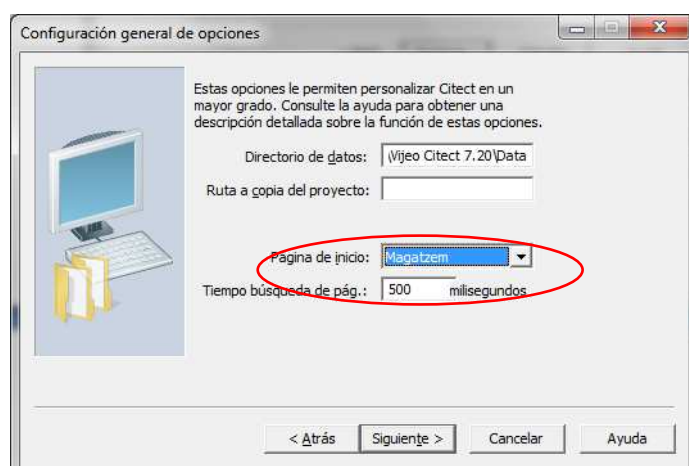


Figura 84. Tretzena pantalla de l'assistent



Figura 85. Última pantalla de l'assistent

## **5.- CREACIÓ DE LA BASE DE DADES DE COMANDES I INTERCANVI DE DADES ENTRE AQUESTA I EL VIJEEO CITECT**

En aquest apartat s'explica com s'ha creat la base de dades i quines han de ser les característiques de les taules d'aquesta per tal de poder intercanviar dades amb el sistema SCADA Vijeo Citect. També es descriu com cal configurar el PC i com cal configurar el sistema SCADA Vijeo Citect per tal d'establir aquest intercanvi de dades.

### **5.1.- Creació de la base de dades**

El sistema SCADA Vijeo Citect permet dos tipus de bases de dades per l'intercanvi d'informació:

- Bases de dades dBASE
- Bases de dades SQL

En aquest projecte s'ha utilitzat una base de dades del tipus dBASE. El format de fitxer de dBASE s'ha convertit en un estàndard de la indústria per a l'emmagatzematge de dades, i el Vijeo Citect és compatible amb aquest format dBASE estàndard. Es pot utilitzar qualsevol editor de base de dades per crear una base de dades i per llegir i escriure camps d'aquesta. La base de dades s'ha creat amb el programa Microsoft Access.

S'ha definit una base de dades (BD\_VijeoCitect.mdb) formada per dues taules diferents:

- Taula 1. Relació de totes les comandes entrades per l'usuari des del sistema SCADA (veure figura 86).
- Taula 2. Relació de totes les comandes ja servides per l'estació magatzem en estat automàtic.

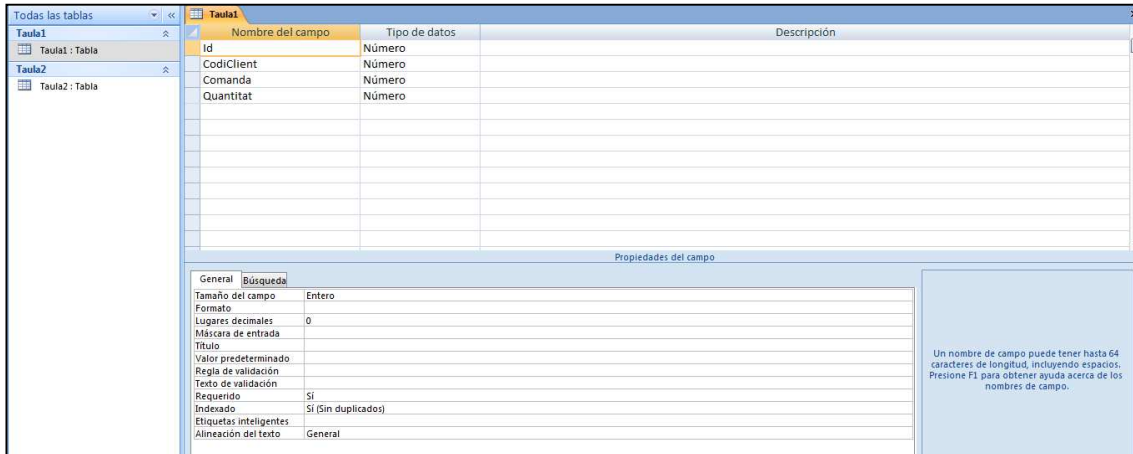


Figura 86. Detall de la vista de disseny de la taula 1

Per cada taula s'han definit quatre camps:

- Id. Identificador del número de comanda. Com es pot observar en la figura anterior, aquest camp s'ha definit com a enter i, com que serà l'índex de referència que s'utilitzarà per llegir les comandes, s'ha definit com a camp requerit, indexat i sense duplicats.
- CodiClient. Codi identificador del client. Aquest camp s'ha definit com a enter i requerit.
- Comanda. Peces que formen la comanda segons la codificació establerta. Aquest camp s'ha definit com a enter i requerit.
- Quantitat. Quantitat de comandes del tipus especificat a servir. Aquest camp s'ha definit com a enter i requerit.

## 5.2.- L'intercanvi de dades entre el Vijeo Citect i l'Access (via ODBC)

El software SCADA Vijeo Citect permet la comunicació i intercanvi de dades amb altres aplicacions pel seu post tractament, emmagatzematge, anàlisis, gestió del control de planta...

El Citect SCADA utilitza diferents metodologies per l'intercanvi de dades:

- Dynamic data exchange (DDE), on el Citect SCADA pot actuar com a **DDE server** aportant els valors dels tags requerits pels clients o de **DDE client**, requerint dades d'altres aplicacions.

- Open database connectivity (ODBC), on el Citect SCADA actua com a **ODBC Server**, permetent a altres aplicacions llegir les variables directament.
- Utilitzant una base de dades externa comuna, on el Citect SCADA i altres aplicacions utilitzen la mateixa base de dades per emmagatzemar i llegir informació.

En aquest projecte s'ha utilitzat la metodologia ODBC per establir l'intercanvi de dades entre la base de dades d'Access i el Vijeo Citect.

Totes les variables que s'utilitzen per aquest intercanvi de dades s'han definit com a variables locals (del tipus INT) de l'SCADA ("*CitectSCADA Editor de projectes - Tags - Variables Locales*"). D'aquesta manera no ocupem tags de llicència i tan sols transferim al PLC les variables una vegada recuperades de la base de dades.

### 5.2.1.- Configuració d'un nou origen de dades en el PC

- Des de l'administrador d'origen de dades del Windows creem un nou origen de dades. Per això cal anar al "*Panel de Control - Herramientas administrativas - Orígenes de datos ODBC*" (veure figura 87).

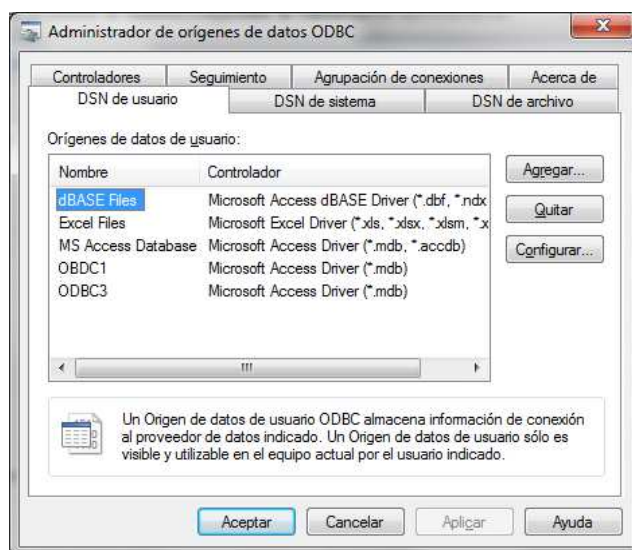


Figura 87. Administrador d'origen de dades de windows

- En aquesta pantalla escollim la pantalla "DSN usuario" i hem d'anar a agregar un nou origen de dades. En la pestanya d'agregació d'origen de dades hem d'escollir la opció "Microsoft Access Driver (\*.mdb)" (veure figura 88).

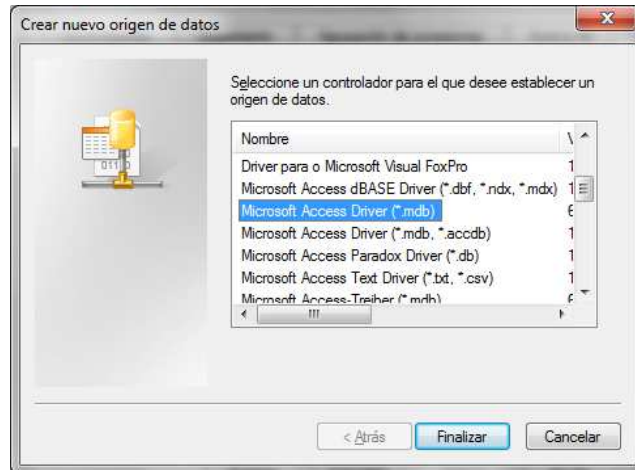


Figura 88. Pantalla de creació d'un nou origen de dades

- A partir d'aquí se'ns obrirà una nova pantalla que ens permet configurar aquest nou element amb el nom que li vulguem donar a l'origen de les dades (ODBC3 en aquest cas) i la ruta de les dades, que la seleccionarem des de la pestanya "Seleccionar" (base de dades a la que ens vulguem comunicar. Veure figura 89).

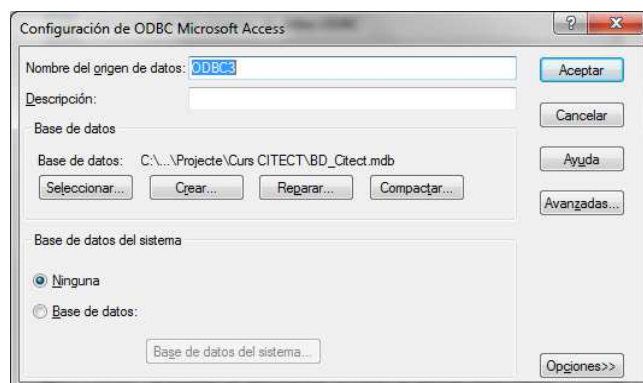


Figura 89. Configuració del nou ODBC

### 5.2.2.- Configuració de l'intercanvi de dades des del Vijeo Citect

Per poder establir la comunicació entre el Vijeo Citect i la base de dades d'Access, en primer lloc haurem de crear un nou dispositiu associat al fitxer de la base de dades. Un dispositiu permet transferir dades d'alt nivell entre el Citect SCADA i altres elements com impressores, bases de dades, fitxers RTF o fitxers ASCII. Aquests dispositius són similars als dispositius d'Entrada / Sortida en la manera que tots dos permeten l'intercanvi de dades amb altres components.

Es poden configurar diferents dispositius , per exemple es pot configurar un dispositiu que envia un informe de sortida a una impressora, o un dispositiu d'intercanvi amb una BD...

- Per definir un dispositiu anirem al CitectSCADA Editor de projectes i al menú "*Sistema - Dispositivos*" (veure figures 90 i 91).

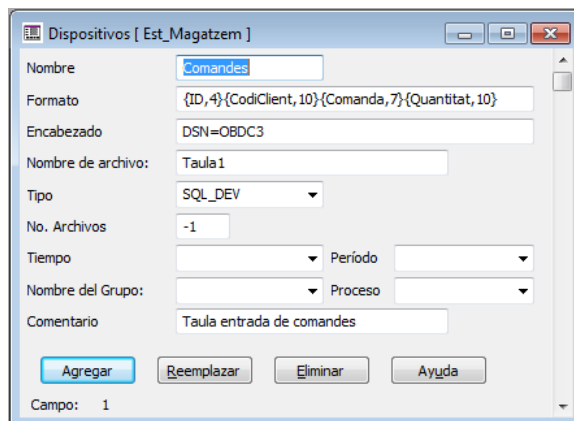


Figura 90. Pantalla de configuració de l'intercanvi amb la taula on es guarden les comandes a executar

- Al nou dispositiu li hem de donar un nom (en aquest cas se li ha donat el nom de "Comandes").
- Seguidament, al camp "*Formato*" hem de definir el mateix format de les columnes que hem entrat en el programa Access (mateix nom i mateix ordre), i li posem també l'amplada màxima que tindrà cada camp.
- A "*Encabezado*" li hem de definir quina és la font de les dades, per això li diem que DSN (Data Source Name) és igual al nom que li hem donat al origen de dades ODBC de Windows.



- A "*Nombre de archivo*" li indiquem la taula de la base de dades on hi ha guardada la informació; en aquest cas "*Taula1*".
- A "*tipo*" hem d'escollir "*SQL\_DEV*".
- I finalment a "*nº de archivos*" li indiquem "-1".

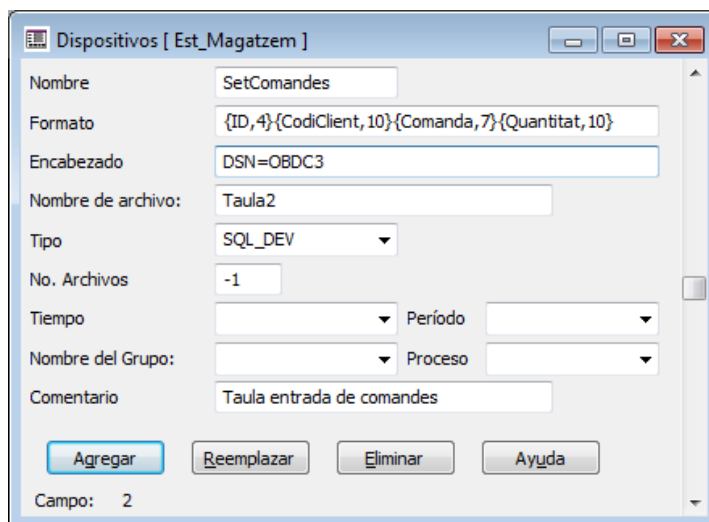


Figura 91. Pantalla de configuració de l'intercanvi amb la taula on s'escriuran les comandes executades

- Podem observar que les úniques diferències entre els dos dispositius són: el nom (cada un ha de tenir el seu propi); la taula a què fan referència (cada un intercanvia informació amb la seva taula corresponent del fitxer d'Access).

Per tal de poder establir l'intercanvi de dades s'han generat les funcions de lectura i escriptura amb l'editor CICODE de què disposa el Vijeo Citect (veure l'apartat 4.2.7.- Les funcions CIDODE). Amb aquest llenguatge de programació s'han creat les funcions que seran cridades des de les pàgines de l'SCADA per ser executades. En aquestes funcions es fa l'obertura, lectura, escriptura i tancament d'aquests dispositius d'intercanvi de dades.

## 6.- La comunicació entre PLC i PC via OPC

En aquest apartat s'explica quin tipus de comunicació s'ha establert entre el PLC SIEMENS S7-300 i el PC on hi ha instal·lat el sistema SCADA Vijeo Citect. S'explica què és i com funciona un OPC (OLE for Process Control), que es tracta del software que s'ha utilitzat per establir aquesta comunicació.

L'OPC (OLE for Process Control) és un estàndard de comunicació en el camp del control i supervisió de processos industrials, basat en una tecnologia Microsoft, que ofereix una interfície comú per la comunicació i permet que components software individuals interaccionin i comparteixin dades. La comunicació OPC es realitza a través d'una arquitectura client-servidor. El servidor OPC és la font de dades (com un dispositiu hardware a nivell de planta) i qualsevol aplicació basada en OPC pot accedir a aquest servidor per tal de llegir i escriure qualsevol variable que ofereixi aquest. És una solució oberta i flexible al clàssic problema dels drivers de propietaris (veure figura 92). Pràcticament tots els fabricants més importants de sistemes de control i instrumentació de processos han inclòs un OPC en els seus productes.

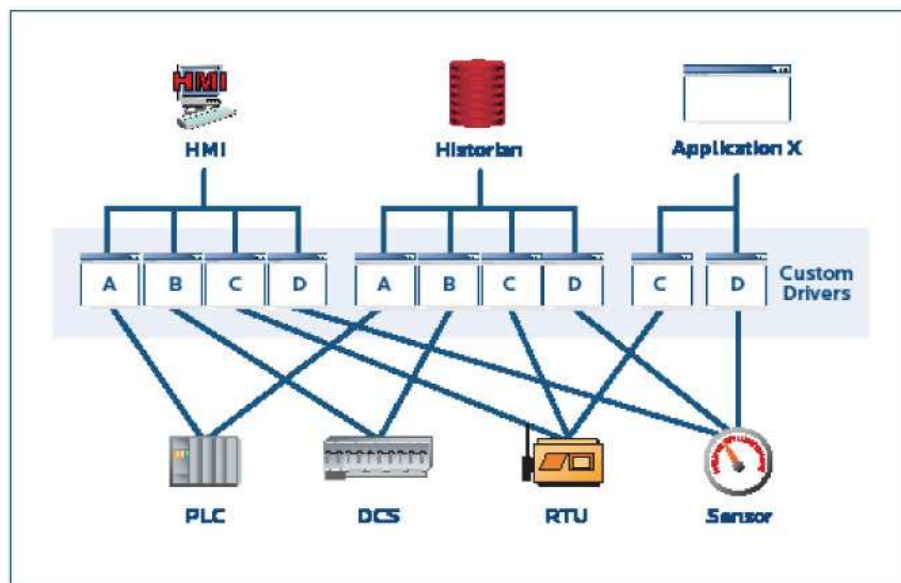


Figura 92. Problemàtica de comunicació entre dispositius

OPC DA prové de OPC Data Access. És una especificació de la fundació OPC que defineix la forma de comunicació i transferència de dades entre una Font de Dades i una Aplicació client (per exemple, com en aquest cas, entre un PLC

i un SCADA), sense necessitat de que cada un conegui el protocol nadiu de l'altre.

L'Arquitectura OPC Client / OPC Servidor va ser la primera arquitectura definida per la Fundació OPC. Abans de l'existència de l'especificació OPC DA, els fabricants de dispositius (PLC, RTU...) obligaven a qualsevol aplicació que necessités connectar-s'hi tenir un *Driver Propietari* capaç de traduir les dades entre l'aplicació i el dispositiu. La necessitat de *Drivers Propietaris* té associats varis problemes habituals en les comunicacions:

- Elevat cost de desenvolupament.
- Deixa "lligat" al client a un únic fabricant.
- Complexitat per configurar i mantenir ja que cada Driver té el seu propi funcionament particular diferent de qualsevol altre.
- Dificultat per mantenir-lo actualitzat davant nous llançaments de versions de hardware o sistema operatiu.

Pel contrari, OPC DA fa possible la connectivitat amb qualsevol dispositiu en temps real i sense la necessitat de *Drivers de Propietaris* (entre cada dispositiu). L'aplicació Client té la funcionalitat de lectura i escriptura sense haver de conèixer el protocol nadiu del dispositiu ni l'estructura de dades del dispositiu al que es vol connectar (això tan sols ho ha de conèixer l'OPC) (veure figura 93).

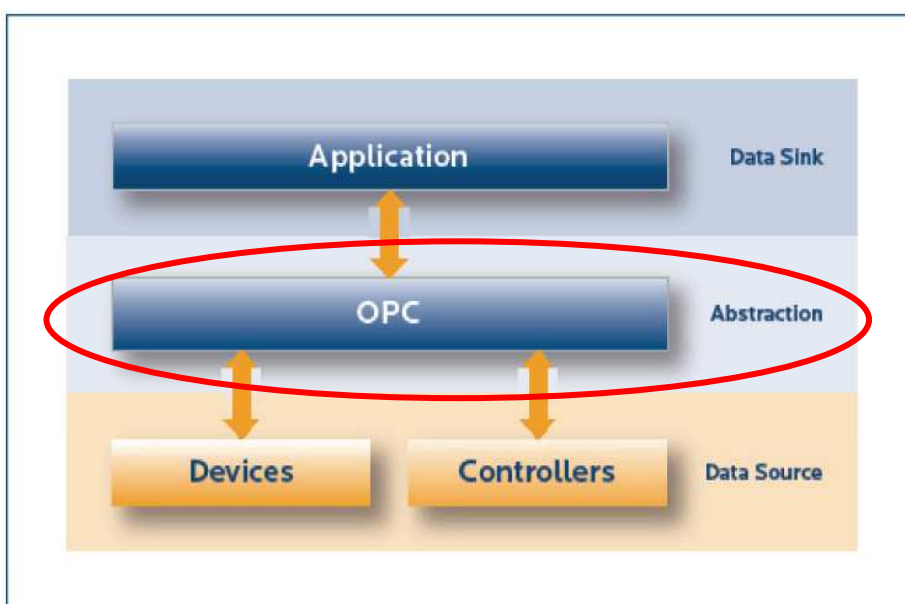


Figura 93. La solució que ofereix un software OPC

Els servidors OPC es mostren com el nivell entremig entre la font de dades i el client d'aquestes dades, habilitant la intercomunicació sense que cap dels extrems conegui el protocol nadiu de l'altre (veure figura 93).

Avantatges d'un OPC Server:

- ✓ Els servidors OPC són connectors que es poden assimilar a traductors entre el món OPC i els protocols nadius d'una Font de Dades.
- ✓ OPC és bidireccional: els servidors OPC poden llegir de i escriure en la Font de Dades.
- ✓ OPC elimina la necessitat de disposar de drivers específics entre cada aplicació i la Font de les Dades (OPC fa de pont).
- ✓ La utilització d'un OPC simplifica enormement la integració de dispositius ja que, una vegada es configura un Servidor OPC per a una Font de Dades, totes les aplicacions que utilitzen OPC poden començar a compartir dades amb aquesta Font de Dades, eliminant la necessitat de drivers addicionals.
- ✓ El trànsit i la càrrega de treball dels dispositius es redueix enormement utilitzant comunicacions OPC.
- ✓ OPC estén la vida útil de sistemes antics. OPC permet a les aplicacions més noves poder seguir-se comunicant amb les aplicacions més antigues.
- ✓ OPC fa possible de forma real que es puguin compartir dades provinents de l'automatització en tota la xarxa corporativa, eliminant la necessitat d'instal·lar nous drivers de comunicació; només es requereix d'un servidor OPC.

L'abstracció de dispositiu OPC s'aconsegueix utilitzant dos components OPC especialitzats: Client OPC i Servidor OPC (veure figura 94).

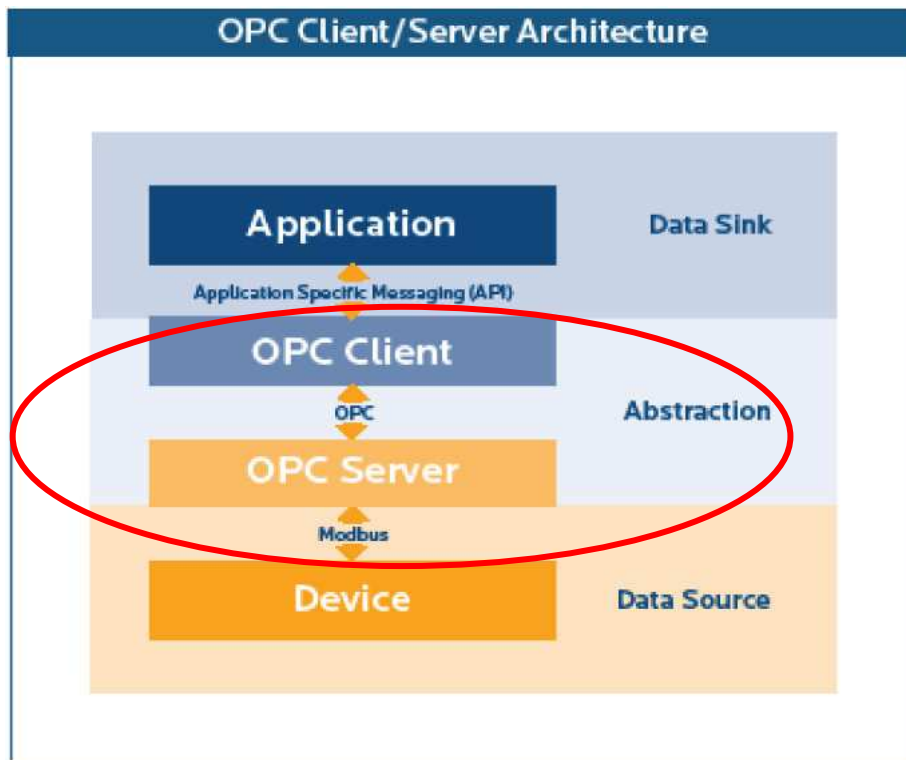


Figura 94. Funcionament d'un OPC

El fet de que la Font de les Dades i el Client de les dades puguin comunicar-se entre si mitjançant OPC no vol dir que els respectius protocols nadius deixin de ser necessaris. Els protocols segueixen existint però només comuniquen amb un dels dos components del software OPC. Llavors, són els components OPC els que intercanvien informació entre sí per tancar el cercle.

### 6.1.- La comunicació entre el PLC i el sistema SCADA (via OPC)

Fins fa poc, si es volia connectar un processador Simatic S7-300 a un PC a través d'una xarxa Ethernet calia instal·lar un mòdul CP del mateix fabricant SIEMENS. En aquests moments, una alternativa és el IBH Link S7++ (que és el dispositiu que s'ha utilitzat en aquest projecte). L'IBH Link S7++ és un convertidor Ethernet compacte per connexió directa a un switch, un hub o directament a la targeta de xarxa del PC. El protocol que utilitza és l'estàndard

TCP/IP. A partir d'aquí, amb l'IBH Link S7++ (veure figura 95) també és possible la comunicació amb Profibus DP.



Figura 95. Dispositiu IBH Link S7++

Tal i com s'ha comentat, el conjunt de la cèl·lula flexible de fabricació funciona en comunicació amb una xarxa industrial Profibus (veure figura 96). Per tal d'establir connexió entre el PC i el PLC s'ha utilitzat una xarxa Ethernet entre el PC i l'IBH Link S7++, que es troba connectat al PLC de control de l'estació de transport. A partir d'aquí, en la configuració del software de l'OPC se li ha dit quin és el node de la xarxa Profibus a la que ens volem connectar.

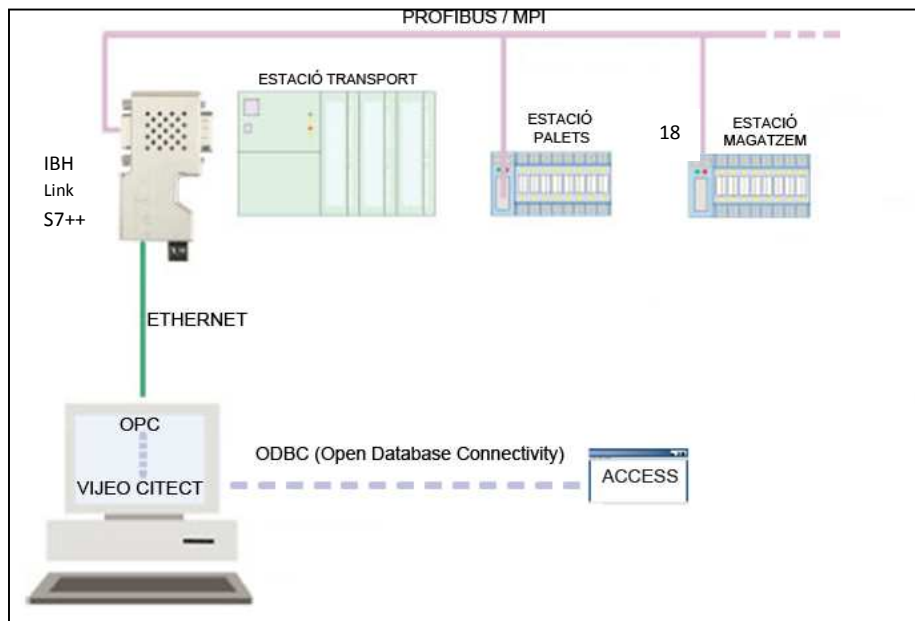


Figura 96. Configuració de comunicacions entre l'estació magatzem i el PC on hi ha instal·lat l'SCADA.

L'IBH Link S7++ necessita el software servidor OPC IBHsoftec, que permet a aplicacions del tipus client, com en aquest cas un sistema SCADA, connectar-se a un sistema de control de la sèrie SIMATIC S7.

### 6.1.1.- Configuració d'un servidor OPC amb l'OPCEditor de IBH OPC Server

- Crear una configuració nova mitjançant "*File - new*". En aquest cas li hem donat el nom de Prova OPC1.
- Seguidament anem a "*Edit - New PLC*" per tal d'introduir un nou PLC en el servidor per establir comunicació. En la pantalla que ens apareix introduïm el nom que li volem donar al PLC (en el nostre cas *EstMagatzem*) i en el desplegable inferior escollim el protocol de comunicació S7 IBHLink (veure figura 97).

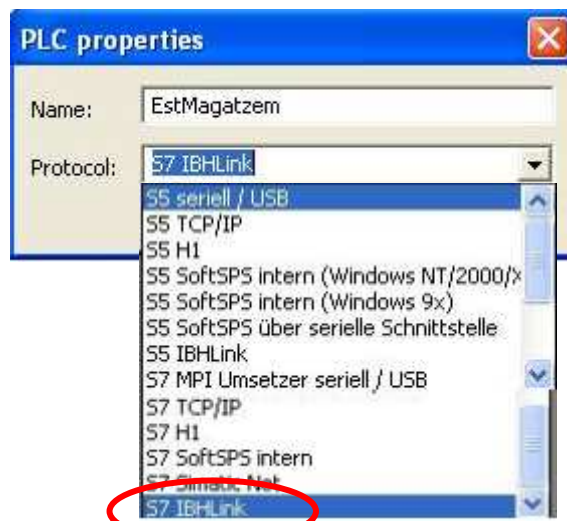


Figura 97. Pantalla d'introducció d'un nou PLC al servidor

- Una vegada escollit el protocol, cal introduir els ajustaments adequats. Per això, cal situar-se sobre el protocol escollit i amb el botó dret seleccionar "*Connection Settings*" (veure figura 98). En primer lloc fa una comprovació de connexió amb el PLC. Si aquesta surt OK, podem continuar endavant amb la configuració i ens apareix la pantalla de configuració on escollirem el node de connexió (en el nostre cas és el 18).

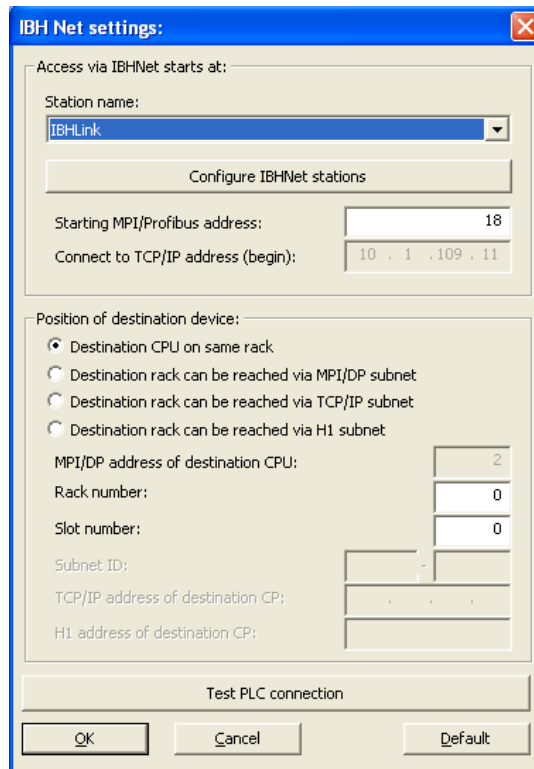


Figura 98. Pantalla de configuració de la comunicació.

- Després d'haver configurat la connexió cal guardar-la (símbol de disquet) i transferir-la al servidor. Per això, cliquem amb el botó dret del mouse sobre la connexió (Prova OPC1) i escollim "*Project Transfer*". En la pantalla següent, escollim l'OPC server IBHSoftec.IBHOPC.DA.1 i transferim (veure figura 99).

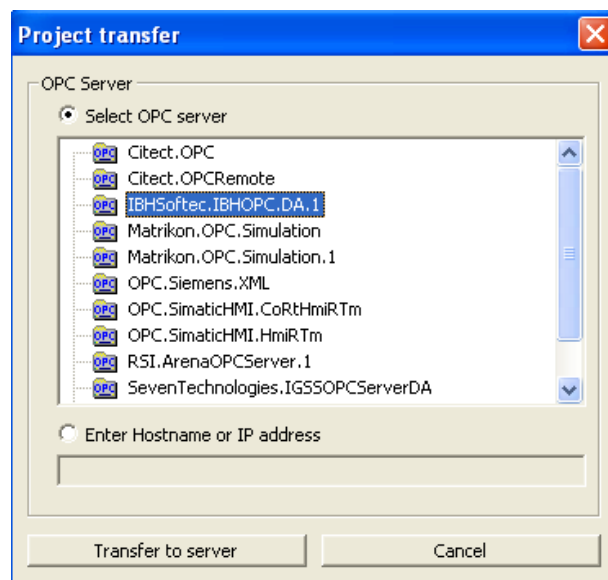


Figura 99. Pantalla de transferència de configuració al servidor



- A partir d'aquí, es pot recuperar el programa del PLC per tal de poder disposar de totes les variables configurades i establir-hi comunicació (veure figura 100). Per això, cliquem amb el botó dret sobre el PLC introduït (EstMagatzem) i anem a buscar el nostre programa del PLC. Ens apareixerà una taula amb totes les variables i podrem escollir les que volem que siguin visibles en l'OPC.

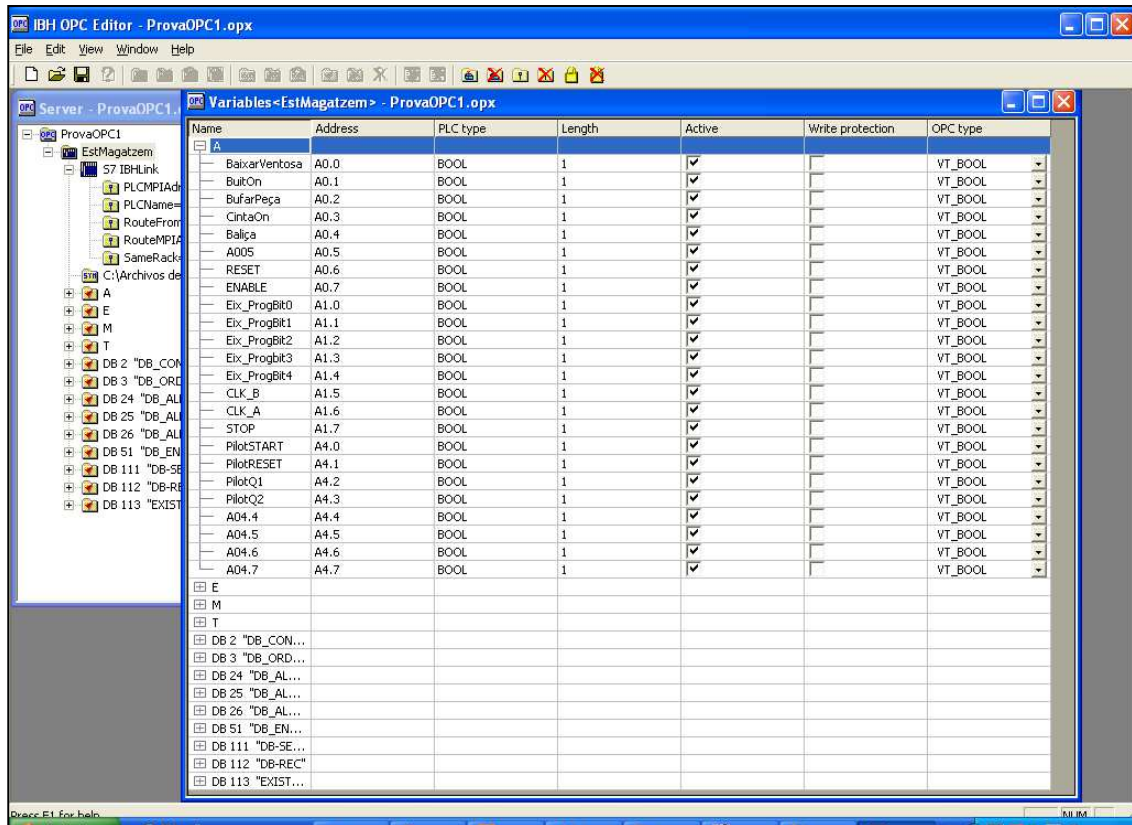


Figura 100. Configuració completa de l'OPC

## 7.- CONCLUSIONS

Aquest apartat es distribueix conceptualment sobre la base de les cinc fases en què s'ha desenvolupat aquest projecte:

- La primera fase s'ha dedicat a l'automatització pròpiament de l'estació magatzem a partir de l'autòmat programable Siemens S7-300 i de les necessitats plantejades. Pel desenvolupament d'aquesta fase s'ha comptat amb el programa d'origen de l'estació, elaborat per FESTO, i s'ha adaptat als requeriments d'aquest projecte. Ha estat un procés llarg i complex ja que l'estació magatzem, per les pròpies característiques del seu funcionament, no disposa d'un programa seqüencial i de fàcil seguiment, sinó que cal una estructuració més complexa en mòduls de programes i crides no seqüencials. En aquesta fase s'ha complert amb les expectatives d'aquest projecte de manera que s'ha automatitzat l'estació pel seu funcionament autònom segons requeriments previs.
- En la segona fase s'ha programat i establert la comunicació per l'intercanvi de dades (lectura i escriptura) entre el sistema SCADA Vijeo Citect i la base de dades de Microsoft Access. Per aconseguir l'intercanvi de dades entre el sistema SCADA Vijeo Citect i la base de dades de Microsoft Access ha calgut utilitzar l'entorn de programació de funcions en llenguatge CICODE que ofereix el mateix sistema SCADA. Ha estat una fase complicada que ha requerit la realització de moltes proves fins a aconseguir un correcte intercanvi de les dades via ODBC (Open Database Connectivity). En aquesta fase també s'ha complert amb les expectatives d'aquest projecte i podríem dir que ha estat la fase més important pel que fa al compliment dels requeriments inicials.
- En la tercera fase s'ha elaborat i programat l'entorn gràfic de supervisió i control del procés a partir del sistema SCADA Vijeo Citect. Aquesta ha estat la fase més entretinguda, ja que era necessari dissenyar un conjunt de pàgines gràfiques de manera que fossin de clara i ràpida interpretació, així com de fàcil utilització per part dels usuaris. Així, s'ha dissenyat un total de tres pàgines que, conceptualment, cada una d'elles

està destinada a una part específica del procés i ofereixen una interacció totalment intuïtiva. Val a dir que aquest sistema SCADA és de recent incorporació en el programa docent de la Universitat de Vic, per tant, en el desenvolupament d'aquest projecte hi ha hagut una tasca molt important d'aprenentatge en el disseny i programació del Vijeo Citect per la seva utilització en la formació de futurs enginyers. Per aprofitar encara millor l'aprenentatge que s'anava aconseguint en la utilització del sistema SCADA Vijeo Citect, paral·lelament al desenvolupament d'aquest projecte s'ha anat elaborant un manual/guia per la seva utilització en la docència del programa (veure ANNEX IV. Manual/guia per la docència del Vijeo Citect). Per tant, també s'ha complert amb els objectius requerits per aquest projecte.

- En la quarta fase s'ha instal·lat un OPC Server en el PC i s'ha establert la comunicació entre el PLC i el sistema SCADA. Aquesta també ha estat una fase que ha requerit molt temps i recerca d'informació. Primer de tot s'ha intentat establir una comunicació sèrie directament entre el PC i el PLC utilitzant el protocol SIEMENS de què disposa el mateix sistema SCADA, però no s'ha pogut establir aquesta comunicació de cap de les maneres. Després de dur a terme moltes proves, s'ha decidit establir la comunicació a través d'un OPC Server. D'aquesta manera s'ha instal·lat el software IBH OPC Server al PC i, per mitjà d'un dispositiu IBH Link S7++ connectat directament al port sèrie del PLC, s'ha establert comunicació via Ethernet. Una vegada establerta la connexió entre l'OPC i el PLC, ha calgut comunicar l'OPC amb el sistema SCADA. Per aquest fet s'ha definit un dispositiu extern d'entrades / sortides en el Vijeo Citect utilitzant el protocol genèric d'OPC Foundation. En aquesta fase també s'han complert els requeriments establerts com a objectius d'aquest projecte.
  
- Finalment s'ha anat revisant i depurant les diferents programacions i comunicacions per tal que el sistema funcioni com a un conjunt. Com qualsevol sistema automàtic, és molt difícil que funcioni tot a la primera, per tant, en aquesta última fase d'aquest projecte s'han anat perfilant

diferents aspectes per tal de depurar les petites fallades i errors que donava el conjunt, així com corregir i retocar també aspectes més estètics o per facilitar la seva utilització.

Com a possibles ampliacions d'aquest projecte es podria:

- Adequar el programa de les diferents estacions de la cèl·lula flexible de fabricació per tal de fer-la funcionar com un conjunt, utilitzant la xarxa industrial de què es disposa.
- Implantar el sistema SCADA al conjunt de la cèl·lula flexible de fabricació programant les pantalles gràfiques corresponents per cada estació.
- El potencial tant de la cèl·lula flexible de fabricació com del sistema SCADA Vijeo Citect són molt grans i permeten fer diferents configuracions: es poden introduir al sistema SCADA elements com tendències, històrics, alarmes, informes; es pot automatitzar les estacions individualment, automatitzar tot el conjunt...

Podem afirmar que aquest projecte ha complert amb tots els objectius proposats i ofereix un exemple que avarca un ventall molt ampli d'aplicació de la tecnologia en el camp de l'automatització, control, supervisió i intercanvi de dades en un procés industrial.

## 8.- BIBLIOGRAFIA

- SCHNEIDER ELECTRIC. *CitectSCADA 7.20 User Guide*. Austràlia: Schneider Electric, 2010.
- SCHNEIDER ELECTRIC. *Documentació Vijeo Citect*. (<http://www.schneider-electric.com/>)
- RODRÍGUEZ PENIN, Aquilino. *Sistemas SCADA*. Barcelona: Marcombo SA, 2012.
- SIEMENS. *Documentació SIEMENS S7-300*. ([www.automation.siemens.com](http://www.automation.siemens.com))
- SIEMENS. *Manual\_programacion\_simatic\_s7\_300*. Versió 2002.
- SIEMENS AG. *SIMATIC Lista de instrucciones (AWL) para S7-300 Y S7-400 Manual de referencia*. Alemanya, 2010.
- SIEMENS AG. *SIMATIC Programar con STEP 7 Manual*. Alemanya, 2006.
- AN CONSULT ESPAÑA SL. *Documentación técnica para IBH OPC Server Versión 4.00*. Madrid.
- AN CONSULT ESPAÑA SL. *IBH Link S7++ y HS*. Madrid.
- DAREJ KOMINEK, P. *Guía para entender la tecnología OPC*. Canadà, 2009. [www.infoPLC.net](http://www.infoPLC.net) (Matrikon OPC)
- BOSCH MARTÍN, Albert. *Simulació d'un magatzem automàtic amb Citect SCADA*. TFC Universitat de Girona, Escola Politècnica Superior. Girona, 2006.
- LLORENS MESTRE, Jordi. *Gestió de la planta FESTO amb el SCADA Citect*. TFC Universitat Rovira i Virgili, Escola Tècnica Superior Enginyeria. Barcelona, 2001.
- REDONDO SOL, Manel. *Diseño e implantació de un sistema SCADA para una planta de producción y envasado de líquidos*. TFC Universitat Autònoma de Barcelona, etse. Bellaterra, 2008.

**ANNEX I. *Tags* de variable definits en aquest projecte**

ANNEX I.1.- Tags de variable de comunicació amb el PLC.

En la taula següent es relacionen totes les variables de comunicació que s'han definit en el programa Vijeo Citect i que permeten una lectura/escriptura directa en el registre corresponent del PLC SIEMENS S7-300 de l'estació magatzem.

NAME	TYPE	UNIT	ADDR	COMMENT
Ocupació	INT	IODev1	EstMagatzem.DB113.DBB80	Ocupació total del magatzem
Peces_Metall	INT	IODev1	EstMagatzem.EXISTENCIES_ALM.P_Metall	Total peces metàl·liques
Peces_Negres	INT	IODev1	EstMagatzem.EXISTENCIES_ALM.P_Negres	Total peces negres
Peces_Taronges	INT	IODev1	EstMagatzem.EXISTENCIES_ALM.P_Verm	Total peces taronges
Pos_1	INT	IODev1	EstMagatzem.DB113.DBW0	Posició 1 del magatzem
Pos_2	INT	IODev1	EstMagatzem.DB113.DBW2	Posició 2 del magatzem
Pos_3	INT	IODev1	EstMagatzem.DB113.DBW4	Posició 3 del magatzem
Pos_4	INT	IODev1	EstMagatzem.DB113.DBW6	Posició 4 del magatzem
Pos_5	INT	IODev1	EstMagatzem.DB113.DBW8	Posició 5 del magatzem
Pos_6	INT	IODev1	EstMagatzem.DB113.DBW10	Posició 6 del magatzem
Pos_7	INT	IODev1	EstMagatzem.DB113.DBW12	Posició 7 del magatzem
Pos_8	INT	IODev1	EstMagatzem.DB113.DBW14	Posició 8 del magatzem
Pos_9	INT	IODev1	EstMagatzem.DB113.DBW16	Posició 9 del magatzem
Pos_10	INT	IODev1	EstMagatzem.DB113.DBW18	Posició 10 del magatzem
Pos_11	INT	IODev1	EstMagatzem.DB113.DBW20	Posició 11 del magatzem
Pos_12	INT	IODev1	EstMagatzem.DB113.DBW22	Posició 12 del magatzem
Pos_13	INT	IODev1	EstMagatzem.DB113.DBW24	Posició 13 del magatzem
Pos_14	INT	IODev1	EstMagatzem.DB113.DBW26	Posició 14 del magatzem
Pos_15	INT	IODev1	EstMagatzem.DB113.DBW28	Posició 15 del magatzem
Pos_16	INT	IODev1	EstMagatzem.DB113.DBW30	Posició 16 del magatzem

Pos_17	INT	IODev1	EstMagatzem.DB113.DBW32	Posició 17 del magatzem
Pos_18	INT	IODev1	EstMagatzem.DB113.DBW34	Posició 18 del magatzem
Pos_19	INT	IODev1	EstMagatzem.DB113.DBW36	Posició 19 del magatzem
Pos_20	INT	IODev1	EstMagatzem.DB113.DBW38	Posició 20 del magatzem
Pos_21	INT	IODev1	EstMagatzem.DB113.DBW40	Posició 21 del magatzem
Pos_22	INT	IODev1	EstMagatzem.DB113.DBW42	Posició 22 del magatzem
Pos_23	INT	IODev1	EstMagatzem.DB113.DBW44	Posició 23 del magatzem
Pos_24	INT	IODev1	EstMagatzem.DB113.DBW46	Posició 24 del magatzem
Pos_25	INT	IODev1	EstMagatzem.DB113.DBW48	Posició 25 del magatzem
Pos_26	INT	IODev1	EstMagatzem.DB113.DBW50	Posició 26 del magatzem
Pos_27	INT	IODev1	EstMagatzem.DB113.DBW52	Posició 27 del magatzem
Pos_28	INT	IODev1	EstMagatzem.DB113.DBW54	Posició 28 del magatzem
Pos_29	INT	IODev1	EstMagatzem.DB113.DBW56	Posició 29 del magatzem
Pos_30	INT	IODev1	EstMagatzem.DB113.DBW58	Posició 30 del magatzem
Pos_31	INT	IODev1	EstMagatzem.DB113.DBW60	Posició 31 del magatzem
Pos_32	INT	IODev1	EstMagatzem.DB113.DBW62	Posició 32 del magatzem
Pos_33	INT	IODev1	EstMagatzem.DB113.DBW64	Posició 33 del magatzem
Pos_34	INT	IODev1	EstMagatzem.DB113.DBW66	Posició 34 del magatzem
Pos_35	INT	IODev1	EstMagatzem.DB113.DBW68	Posició 35 del magatzem
Pos_36	INT	IODev1	EstMagatzem.DB113.DBW70	Posició 36 del magatzem
Pos_37	INT	IODev1	EstMagatzem.DB113.DBW72	Posició 37 del magatzem
Pos_38	INT	IODev1	EstMagatzem.DB113.DBW74	Posició 38 del magatzem
Pos_39	INT	IODev1	EstMagatzem.DB113.DBW76	Posició 39 del magatzem
Pos_40	INT	IODev1	EstMagatzem.DB113.DBW78	Posició 40 del magatzem
Det_peça	DIGIT AL	IODev1	EstMagatzem.E0.7	Detecció peça a l'entrada estació



Det_metall	DIGIT AL	IODev1	EstMagatzem.E1.1	Detecció peça metàl·lica
Det_no_negra	DIGIT AL	IODev1	EstMagatzem.E1.0	Detecció peça no negra
Sortida_A	INT	IODev1	EstMagatzem.MB28	Peça a la posició A del palet
Sortida_B	INT	IODev1	EstMagatzem.MB27	Peça a la posició B del palet
Sortida_C	INT	IODev1	EstMagatzem.MB29	Peça a la posició C del palet
Comanda_PLC	INT	IODev1	EstMagatzem.MW130	Comanda que s'envia al PLC
Processant_PLC	DIGIT AL	IODev1	EstMagatzem.M96.0	Bit activat mentre el PLC està processant comanda
Mode_Auto	DIGIT AL	IODev1	EstMagatzem.M54.3	Mode de funcionament de l'estació
Pet_Peça_Citect	DIGIT AL	IODev1	EstMagatzem.M200.0	Petició de peça en estat manual
Sortida_on	DIGIT AL	IODev1	EstMagatzem.M97.0	Activat mentre s'executa sortida de peça
Cnt_Sortides	INT	IODev1	EstMagatzem.MB34	Comptador de sortides executades
Sortida_auto	DIGIT AL	IODev1	EstMagatzem.M200.1	Ordre per confirmar nou carro preparat
Salta_comanda	DIGIT AL	IODev1	EstMagatzem.M200.3	Ordre de saltar la comanda actual
Salta_com_hide	DIGIT AL	IODev1	EstMagatzem.M200.2	Si no hi ha peça a servir, ens mostra pulsador salta com.

ANNEX I.2.- Tags de variables locals.

En la taula següent es relacionen les variables internes que s'han definit en el programa Vijeo Citect i que, per tant, no s'estableix cap comunicació entre elles i el PLC.

NAME	TYPE	COMMENT
Peça_1	INT	Peça 1 de la nova comanda que es vol entrar
Peça_2	INT	Peça 2 de la nova comanda que es vol entrar
Peça_3	INT	Peça 3 de la nova comanda que es vol entrar
CodiClient	INT	Codi del client pel que s'executa la comanda
Auto	DIGITAL	Mode de funcionament del procés
IDS	INT	Identificador de la BD de la comanda en procés
Quantitat	INT	Quantitat total de peces de la comanda en procés
Cnt	INT	Comptador de palets servits d'una comanda
ComandaActual	INT	Comanda que s'està servint
nID	INT	Variable utilitzada en funció Cicode
CodiClient_New	INT	Variable taula BD per entrada nova comanda
Comanda_New	INT	Variable taula BD per entrada nova comanda
Quantitat_New	INT	Variable taula BD per entrada nova comanda
ID_CC	INT	Variable utilitzada en funció Cicode
sBuf_v	STRING	Variable utilitzada en funció Cicode per desplegar
Exec_Comand	INT	Propera comanda a executar el procés
NombredeComanda	INT	Variable utilitzada en funció Cicode
Camisa_A_CITECT	INT	Variable per identificar primera peça extreta
Camisa_B_CITECT	INT	Variable per identificar segona peça extreta
Camisa_C_CITECT	INT	Variable per identificar tercera peça extreta

## **ANNEX II. Configuració IBH OPC Server**

### **ANNEX III. Programa del PLC S7-300 de l'Estació Magatzem**

**ANNEX IV. Manual/guia per la docència del Vijeo Citect**