



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

ENRIQUE MURIAS FERNÁNDEZ

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

**DESARROLLO Y SEGURIDAD SCADA DE UNA  
PLANTA CERÁMICA**

(JULIO 2019)

# ÍNDICE

<b>1. MEMORIA</b> .....	<b>4</b>
1.1 OBJETO .....	4
1.2 ALCANCE .....	4
1.3 ENTORNO TECNOLÓGICO.....	4
1.3.1. <i>Ciberseguridad industrial.</i> .....	4
1.3.2. <i>Sistema SCADA:</i> .....	4
1.4 SEGURIDAD SCADA Y REFORMA DE RED. ....	5
1.4.1. <i>Detección de los problemas iniciales.</i> .....	5
1.4.1.1. Seguridad Modbus/TCP.....	5
1.4.1.2. Estructura inicial de la red.....	6
1.4.1.3. Sistema operativo del SCADA actual. ....	6
1.4.1.4. Resultados del escaneo de puertos con Nmap .....	7
1.4.1.5. Resultados de la identificación de las vulnerabilidades.....	9
1.4.2. <i>Soluciones adoptadas.</i> .....	10
1.4.2.1. Sistemas operativos de los equipos SCADA. ....	10
1.4.2.2. Diseño de la red según estándar ISA99/IEC6443. ....	10
1.4.3. <i>Equipos instalados y reforma de la arquitectura de red.</i> .....	12
1.4.3.1. Distribución de los equipos en el armario CPD .....	14
1.4.4. <i>Configuración de los equipos instalados</i> .....	14
1.4.4.1. Configuración de las VLANs.....	15
1.4.4.2. Configuración de las políticas de seguridad .....	18
1.5 DESCRIPCIÓN DEL PROCESO PRODUCTIVO Y LA PLANTA DE PRODUCCIÓN.....	19
1.6 DESARROLLO DEL SISTEMA SCADA.....	21
1.6.1. <i>Requisitos de diseño</i> .....	21
1.6.2. <i>Arquitectura SCADA desarrollada.</i> .....	22
1.6.2.1. Hipervisor VMware ESXi.....	24
1.6.2.2. Diagrama de la arquitectura SCADA.....	25
1.6.2.3. Clientes ligeros o <i>Thin Clients</i> .....	26
1.6.3. <i>Programas utilizados</i> .....	27
1.7 NORMATIVA Y REFERENCIAS .....	28
1.7.1. <i>Normas y disposiciones legales aplicadas</i> .....	28
1.7.2. <i>Referencias y bibliografía</i> .....	29
<b>2. ANEXOS</b> .....	<b>30</b>
2.1 ANEXO I - MANUAL DEL PROGRAMADOR.....	30
2.1.1. <i>Versión de WinCC</i> .....	30
2.1.2. <i>Consideraciones de rendimiento desarrollo SCADA en WinCC</i> .....	30
2.1.2.1. Empleo óptimo de faceplates .....	30
2.1.2.2. Dinamización de objetos .....	31
2.1.2.3. Configuración de los disparadores .....	33
2.1.2.4. Inicialización de variables.....	34
2.1.2.5. Ocultar/mostrar valor de variable en Runtime .....	34
2.1.2.6. Uso del atributo “Adaptar borde” .....	35
2.1.2.7. Optimizar lectura y/o escritura de variables en Scripting VBS.....	36
2.1.2.8. Lectura o escritura directa de variables .....	36
2.1.2.9. Fondo de imagen estático como archivo EMF .....	37
2.1.3. <i>Creación del Proyecto con WinCC Explorer</i> .....	37
2.1.3.1. Árbol del proyecto.....	38
2.1.3.2. PLCs del proyecto .....	39

2.1.3.3.	Gestión de las variables.....	40
2.1.4.	<i>Desarrollo de la programación</i> .....	42
2.1.4.1.	Desarrollo de librerías de funciones.....	42
2.1.4.2.	Acciones globales implementada.....	67
2.1.4.3.	Programación de objetos.....	69
2.1.4.4.	Programación de las pantallas de mando.....	86
2.1.5.	<i>Estándar decluttering y niveles de zoom en sinóptico PLANTA TIERRAS</i> .....	102
2.1.6.	<i>Archivos de alarmas</i> .....	103
2.1.7.	<i>Archivo de variables</i> .....	104
2.1.8.	<i>Administración de permisos y usuarios</i> .....	107
2.1.9.	<i>Configuración de permisos en imágenes de proceso</i> .....	109
2.1.10.	<i>Scripts generados en VBA para automatizar y agilizar la creación del proyecto</i> 110	
2.2	ANEXO II - MANUAL DEL USUARIO SCADA.....	114
2.2.1.	<i>Pantalla de inicio</i> .....	114
2.2.2.	<i>Pantalla base</i> .....	114
2.2.3.	<i>Pantalla de gráficas</i> .....	115
2.2.4.	<i>Pantalla de actuaciones de operador</i> .....	117
2.2.5.	<i>Pantalla de alarmas</i> .....	118
2.2.6.	<i>Pantalla planta tierras</i> .....	119
2.2.7.	<i>Pantalla destinos</i> .....	119
2.2.8.	<i>Pantalla palista</i> .....	120
2.2.9.	<i>Pantalla silos</i> .....	120
2.2.10.	<i>Pantalla molino y laminadores</i> .....	121
2.2.11.	<i>Pantalla pudridero</i> .....	121
2.2.12.	<i>Pantalla destino fábricas</i> .....	122
2.2.13.	<i>Pantalla selección fábricas</i> .....	123
2.2.14.	<i>Pantalla básculas</i> .....	123
2.2.15.	<i>Pantalla configuración de parámetros</i> .....	125
2.2.16.	<i>Pantalla fábrica 2</i> .....	126
2.2.17.	<i>Pantalla fábrica 3</i> .....	126
2.2.18.	<i>Pantalla moldeo F3</i> .....	127
2.2.19.	<i>Pantalla apilado F3</i> .....	127
2.2.20.	<i>Pantalla secadero F3</i> .....	128
2.2.21.	<i>Pantalla caja de mezclas zona 1 F3</i> .....	129
2.2.22.	<i>Pantalla caja de mezclas zona 2 F3</i> .....	129
2.2.23.	<i>Pantalla caja de mezclas zona 3 F3</i> .....	130
2.2.24.	<i>Pantalla selección secadero F3</i> .....	130
2.2.25.	<i>Pantalla sala térmica</i> .....	131
2.2.26.	<i>Pantalla gráfica presión y humedad sala térmica 1</i> .....	131
2.2.27.	<i>Pantalla gráfica sala térmica 2 F3</i> .....	132
2.2.28.	<i>Pantalla gráfica temperaturas sala térmica 1 F3</i> .....	133
2.2.29.	<i>Pantalla horno-prehorno F3</i> .....	135
2.2.30.	<i>Pantalla prehorno F3</i> .....	135
2.2.31.	<i>Pantalla selección destinos horno F3</i> .....	136
2.2.32.	<i>Pantalla zona 1 horno F3</i> .....	137
2.2.33.	<i>Pantalla zona 2 horno F3</i> .....	138
2.2.34.	<i>Pantalla gráfica presión horno F3</i> .....	139
2.2.35.	<i>Pantalla gráfica temperatura horno F3</i> .....	139
2.2.36.	<i>Pantalla notas F3</i> .....	140
2.2.37.	<i>Pantalla parámetros F3</i> .....	140
2.2.38.	<i>Pantallas de consumos energéticos</i> .....	140

2.2.38.1.	Pantalla interruptores de linea .....	141
2.2.38.2.	Pantalla interruptores molienda .....	142
2.2.38.3.	Pantalla interruptores fabrica 3-1 .....	142
2.2.38.4.	Pantalla interruptores fabrica 3-2 .....	143
2.2.39.	<i>Pantallas red de comunicaciones</i> .....	144
2.2.39.1.	Pantalla red general comunicaciones .....	144
2.2.40.	<i>Pantalla servidores</i> .....	145
2.2.41.	<i>Pantalla red general de tierras</i> .....	146
2.2.41.1.	Pantalla estado MB+ salida tierras y distribución fábricas .....	147
2.2.41.2.	Pantalla estado MB+ molienda .....	148
2.2.41.3.	Pantalla red general fábrica 3 .....	148
2.2.41.4.	Pantalla estado MB+ moldeo 3 .....	149
2.2.41.5.	Pantalla red general horno F3.....	149
2.3	<b>ANEXO III - MANUAL DE CONFIGURACIÓN</b> .....	150
2.3.1.	<i>Procedimiento de instalación y configuración del software WinCC en los equipos.</i> 150	
2.3.1.1.	Particionado del disco duro.....	150
2.3.1.2.	Activar características en Windows .....	154
2.3.1.3.	Instalación de Simatic NET .....	162
2.3.1.4.	Instalación de WinCC 7.4 Service Pack 1 .....	167
2.3.1.5.	Instalación Update 10 para WinCC 7.4 SP1 .....	173
2.3.1.6.	Instalación licencias WinCC .....	179
2.3.2.	<i>Configurar Máquina Server Vereá</i> .....	181
2.3.2.1.	Activar el acceso remoto a los equipos .....	181
2.3.2.2.	Configuración de las máquinas virtuales de los servidores SCADA. ....	181
2.3.2.3.	Configuración de la resolución de hosts del sistema. ....	185
2.3.2.4.	Especificación de la carpeta compartida en el Servidor. ....	186
2.3.3.	<i>Configuración de la Estación de Ingeniería</i> .....	188
2.3.4.	<i>Configuración de los Equipos Clientes</i> .....	193
2.3.4.1.	Configuración de la caché de imágenes .....	198
2.3.5.	<i>Configuración WebNavigator</i> .....	201
2.3.5.1.	Configurar WebNavigator en el servidor .....	201
2.3.5.2.	Configurar cliente WebNavigator.....	206
2.3.6.	<i>Configuración cliente RDP en dispositivos móviles y tabletas</i> .....	216
2.3.7.	<i>Configuración de los clientes ligeros de los puestos de operación</i> .....	220
2.3.7.1.	Configuración con el Quick Set-Up.....	220
2.3.7.2.	Configuración desde el menú.....	222
<b>3.</b>	<b>PLANOS</b> .....	<b>227</b>
<b>4.</b>	<b>PRESUPUESTO</b> .....	<b>234</b>
<b>5.</b>	<b>CRONOGRAMA DEL TRABAJO</b> .....	<b>237</b>

# 1. Memoria

## 1.1 Objeto

El presente trabajo fin de máster tiene por objeto la digitalización y securización de la infraestructura sobre la que se desplegará un sistema SCADA de alta disponibilidad que permita la supervisión y control del proceso de producción de una planta de producción de cerámica, así como también el desarrollo del sistema SCADA redundante propiamente dicho según las necesidades especificadas por el cliente.

## 1.2 Alcance

En este trabajo se pretende reflejar desde el análisis de los problemas iniciales por los cuales se ha decidido llevar a cabo la reforma de la topología de red, la infraestructura y el sistema de supervisión de la planta, pasando por el diseño e implementación real de las soluciones adoptadas.

## 1.3 Entorno tecnológico

### 1.3.1. Ciberseguridad industrial.

La ciberseguridad en los entornos industriales ha cobrado importancia en la actualidad debido al proceso de convergencia de la tecnología de la información (IT) con la tecnología de operación (OT).

La tecnología IT está destinada a tratar datos mediante el uso de equipos informáticos, sin embargo el ámbito OT se caracteriza por la detección y el cambio de los procesos físicos empleando dispositivos cuya finalidad es el control de un proceso de producción.

Esta convergencia conlleva la convivencia en un mismo entorno de tecnologías y protocolos en los que prima la disponibilidad de la información ante la seguridad o se rigen por el principio de “seguridad por oscuridad” en el caso de los protocolos propietarios, con tecnologías del ámbito IT para procesar y tratar los datos, en las que a menudo se emplea conexión a internet con el riesgo que ello supone. Es por ello que surge la necesidad de integrar soluciones que aporten seguridad a estos entornos.

Actualmente la planta de producción dispone de una red totalmente plana configurada con varios switches industriales distribuidos por los armarios en distintos puntos de la fábrica. No se dispone de un firewall ni algún medio que controle el tráfico de la red.

### 1.3.2. Sistema SCADA:

Un sistema SCADA (Supervisory Control And Data Acquisition) es un tipo de aplicación encargada de supervisar, controlar y adquirir los datos operativos de la planta, permitiendo al operario tomar decisiones en su manejo.

Existen multitud de fabricantes de SCADAs comerciales que han desarrollado sus propios sistemas que facilitan la creación de aplicaciones mediante un entorno de desarrollo, además de ofrecer soporte y documentación para obtener una implementación relativamente rápida.

Los factores decisivos para elegir una solución comercial son los siguientes:

- Requisitos de integración de los equipos de la planta (PLCs y Tarjetas de comunicación).
- Aspectos económicos (sistema de desarrollo, licencias, soporte,...).
- Soporte (respuesta profesional ante errores inesperados y casos de uso particulares).

- Documentación.
- Facilidad de desarrollo.
- Actualizaciones del SCADA y proyección hacia el futuro.

Actualmente la fábrica dispone de un sistema SCADA parcialmente desarrollado con FactoryLink pero presenta problemas en la visualización de las gráficas de proceso (tarda mucho tiempo en cargar la base de datos y en ocasiones tampoco registra los datos adecuadamente con la consiguiente pérdida de información del proceso). Además está prevista la ampliación de la fábrica con más líneas de producción, por lo que es inviable seguir desarrollando sobre esta plataforma que ya no tiene soporte.

Por ello, se decide realizar un SCADA nuevo utilizando SIMATIC WinCC que sea de fácil manejo, ampliable y reutilizable en la medida de lo posible para facilitar el desarrollo y mantenibilidad del mismo.

## 1.4 Seguridad SCADA y reforma de red.

### 1.4.1. Detección de los problemas iniciales.

Las instalaciones de red de la planta que se han encontrado durante el proceso de valoración se consideran inseguras ya que esta no cuenta con los mecanismos de protección necesarios para que una persona ajena a las instalaciones pueda conectarse a un PLC, forzar una variable e inferir en el sistema productivo si esa fuese su intención o que mediante un vector de ataque se pueda infectar alguno de los sistemas para obtener información o cual quiera que sea la intención del atacante.

Se han detectado los siguientes problemas que afectan a la seguridad:

- Ausencia de un firewall que controle el acceso a la red.
- No hay segmentación en la red que permita establecer distintas zonas según el tráfico que circule por ellas.
- Empleo del protocolo Modbus/TCP en la comunicación de los equipos de campo.
- Sistema operativo del SCADA Windows NT, desactualizado y con graves problemas de seguridad.

#### 1.4.1.1. Seguridad Modbus/TCP.

Modbus es un protocolo de comunicaciones industriales situado en la capa de aplicación que permite el empleo de distintos medios físicos para su transporte. Permite realizar comunicaciones en modo cliente/servidor entre distintos equipos de diferentes tecnologías de capas inferiores como TCP/IP entre otras.

En la planta de producción se emplea Modbus TCP/IP que encapsula la trama Modbus serie sobre Ethernet.

Modbus no tiene ninguna característica de protección, por tanto las debilidades son intrínsecas a las especificaciones del protocolo, lo que hace necesario emplear elementos de seguridad adicionales para mitigar estos fallos de seguridad en la implementación del protocolo.

La solución más efectiva sería el cifrado de las comunicaciones para impedir el análisis de la información en caso de ser capturado el tráfico, pero es de difícil aplicación debido a que los equipos no tienen la capacidad suficiente para cifrar las comunicaciones y sería necesario contar con herramientas externas para realizar el cifrado por cada uno de los equipos que usen el protocolo Modbus en la red de control.

Por tanto, se considera adecuado en este caso el empleo de un firewall que controle el tráfico y el acceso a la red. Es una contramedida que requiere menor inversión que la implementación del cifrado y puede combinarse en un futuro con soluciones IDS e IPS si desea obtenerse mayor nivel de seguridad además de que con este mismo dispositivo se puede realizar la segmentación de la red permitiendo separar los dispositivos que se comunican por Modbus de los otros tipos de redes que se usan en la planta.

#### **1.4.1.2. Estructura inicial de la red.**

Inicialmente se ha encontrado una red totalmente plana en la que todos los equipos están conectados entre sí a través de los switches de la red de control sin ningún tipo de control de acceso ni restricción.

Tal y como puede apreciarse en el plano de la red inicial existen tres conexiones a internet descentralizadas mediante módems 3G que presentan redes wifi con implementaciones de cifrado vulnerables y que con pocos recursos podrían extraerse las claves y acceder a la red, además estas zonas descentralizadas comunican libremente con internet y no presentan un firewall que controle el tráfico.

Las conexiones entre los distintos armarios que se pueden apreciar en el plano y que componen la red de automatización, están realizadas con fibra óptica y establecen una configuración con 3 anillos. Esto permite que en caso de ruptura de una de las fibras pueda seguir produciendo la fábrica mientras se repara la avería, lo que incrementa la disponibilidad de la producción, además en cada cuadro general de cada una de las zonas de disponen switches redundantes para eliminar los puntos únicos de fallo.

#### **1.4.1.3. Sistema operativo del SCADA actual.**

El SCADA actual está funcionando sobre el sistema operativo Windows NT que ya no tiene soporte de actualizaciones y presenta graves problemas de seguridad que se han detectado durante la detección de las vulnerabilidades realizada en un entorno controlado con máquinas virtuales.



Versión de Windows del SCADA en FactoryLink

La prueba de detección de vulnerabilidades ha consistido en la identificación de puertos y servicios que presenta expuestos el servidor.

Se ha realizado con las herramientas Nmap para identificar los puertos y Nessus para detectar los servicios vulnerables.

#### 1.4.1.4. Resultados del escaneo de puertos con Nmap

```
root@kali:~# nmap -sV -A 192.168.10.2
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-18 12:30 EDT
Nmap scan report for 192.168.10.2
Host is up (0.50s latency).
Not shown: 985 closed ports
PORT      STATE SERVICE      VERSION
7/tcp    open  echo
9/tcp    open  discard?
13/tcp   open  daytime     Microsoft Windows International daytime
17/tcp   open  qotd        Windows qotd (Spanish)
19/tcp   open  chargen
135/tcp  open  msrpc       Microsoft Windows RPC
445/tcp  open  microsoft-ds Windows 2000 microsoft-ds
514/tcp  filtered shell
1025/tcp open  msrpc       Microsoft Windows RPC
1433/tcp open  ms-sql-s    Microsoft SQL Server 2000 8.00.534.00; SP2
5631/tcp open  pcanywheredata Symantec pcAnywhere
6100/tcp open  synchronet-db?
6101/tcp open  backupexec?
```



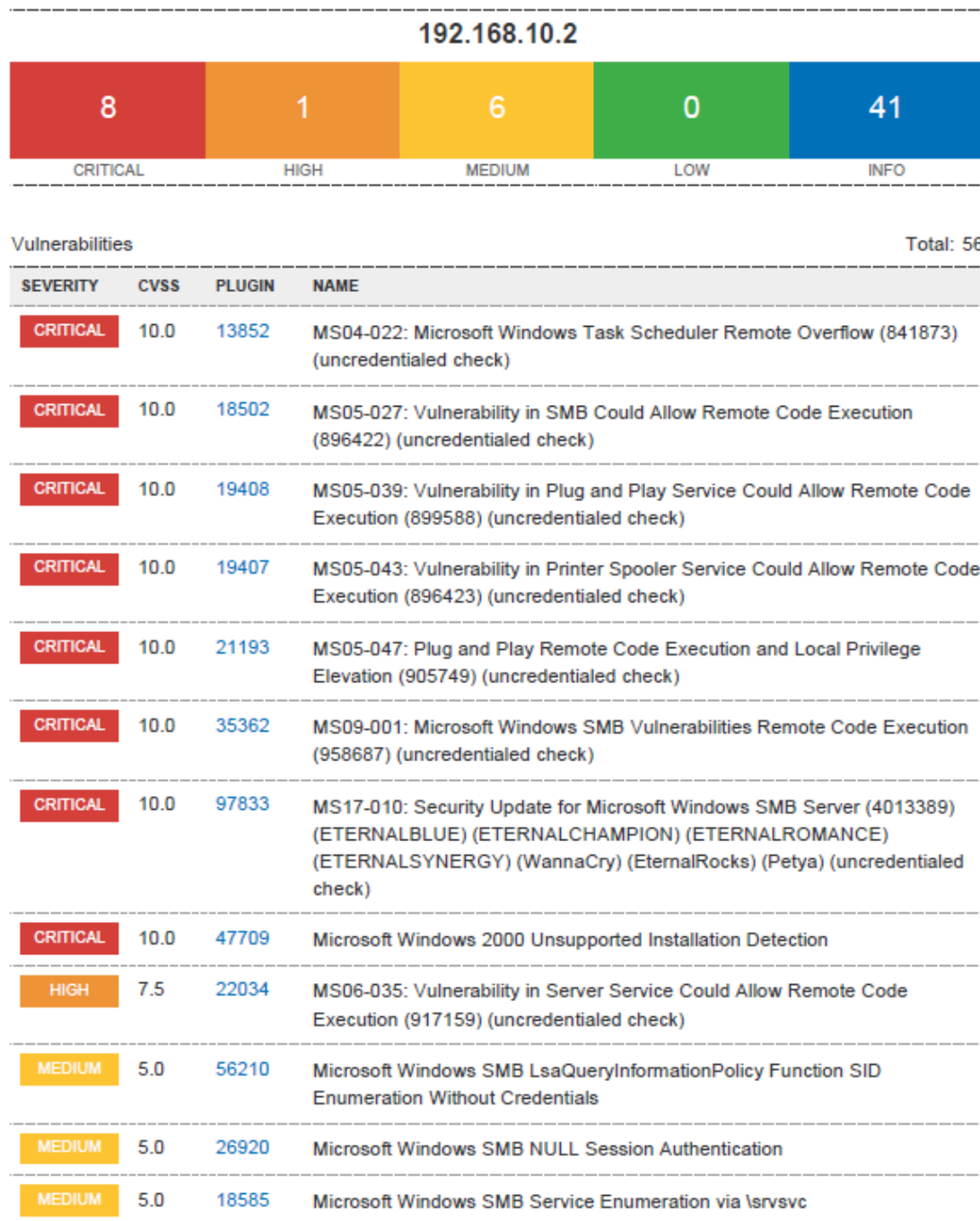
6106/tcp open isdninfo?  
8000/tcp open http-alt?  
Running: Microsoft Windows XP|7|2012  
OS CPE: cpe:/o:microsoft:windows\_xp::sp3 cpe:/o:microsoft:windows\_7  
cpe:/o:microsoft:windows\_server\_2012  
OS details: Microsoft Windows XP SP3, Microsoft Windows XP SP3 or Windows 7 or  
Windows Server 2012  
Network Distance: 2 hops  
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows,  
cpe:/o:microsoft:windows\_xp  
TRACEROUTE (using port 554/tcp)  
HOP RTT ADDRESS  
1 0.87 ms 192.168.44.2  
2 1003.83 ms 192.168.10.2

OS and Service detection performed. Please report any incorrect results at  
<https://nmap.org/submit/>.

Nmap done: 1 IP address (1 host up) scanned in 520.17 seconds

---

### 1.4.1.5. Resultados de la identificación de las vulnerabilidades.



Usando la herramienta Nessus se han detectado 9 vulnerabilidades que permiten la ejecución remota de código lo que posibilitaría a un atacante explotar la vulnerabilidad y mediante un payload ganar acceso a la máquina o elevar privilegios para el fin que se considere oportuno.

Mediante estos vectores de ataque se podría comprometer el sistema e inferir en el sistema productivo empleando herramientas públicas que están al alcance de cualquiera persona.

### 1.4.2. Soluciones adoptadas.

Por una parte es necesario mitigar las vulnerabilidades de los sistemas operativos que se emplean en la supervisión y control de la planta y por otra reformar la red siguiendo el estándar ISA99/IEC6443 para poder cumplir con los requisitos de seguridad que demanda la industria en la actualidad.

#### 1.4.2.1. Sistemas operativos de los equipos SCADA.

Se substituyen los servidores SCADA con el sistema operativo Windows NT por máquinas virtuales con Windows Server 2016 que tiene un soporte de actualizaciones hasta el 12/01/2027



De igual forma las máquinas virtuales de los clientes del SCADA tendrán instalado el sistema operativo Windows 10.

#### 1.4.2.2. Diseño de la red según estándar ISA99/IEC6443.

El estándar ISA99/IEC6443 es el primer marco de referencia internacional sobre ciberseguridad industrial en el cual la disponibilidad e integridad de los datos son los factores más importantes para establecer las medidas preventivas contra las amenazas, así como también contra los incidentes tecnológicos no intencionados.

Este estándar trata de definir distintas zonas y conductos que nos permitan aplicar distintos niveles de seguridad a cada uno de sus componentes.

- **Zonas:** A grandes rasgos la norma define las zonas como una agrupación física o lógica de aquellos componentes industriales independientes (tanto físicos como software o información) que compartan los mismos requisitos de seguridad.

Cabe destacar que la definición de las zonas puede realizarse usando la herencia y formar un árbol de zonas padres e hijos en caso de ser necesario.

Cada una de las zonas tiene unas características y requisitos de seguridad determinados que constituyen sus atributos:

- Políticas y niveles de seguridad
- Inventario de activos
- Requisitos de acceso y controles
- Amenazas y vulnerabilidades
- Consecuencias de una brecha de seguridad
- Tecnología autorizada
- Procesos de gestión de cambios.

**Conductos:** Los conductos son un tipo particular de zona en la que se aplican las comunicaciones que permiten la transmisión de información entre distintas zonas proporcionando funciones de seguridad para la transmisión segura de la información entre dos zonas distintas.

En los sistemas industriales los conductos suelen estar formados por los equipos que constituyen la arquitectura de la red (routers, switches, firewalls, gateways, etc...) que son los encargados de gestionar la transmisión de la información entre las distintas zonas y la conversión de los protocolos.

Para ayudar en la definición inicial de las zonas y conductos de la planta de producción existen distintos modelos de referencia propuestos por distintos fabricantes como: Rockwell, Honeywell, Siemens y por organizaciones como: DuPont, Purdue, Tofino, etc..., aunque cada uno de ellos debe adaptarse a las particularidades de cada instalación.

Para la instalación objeto de este trabajo se ha desarrollado un diagrama de segmentación de zonas y conductos.

- **Identificación de las Zonas:**

Zona	VLAN
Oficina	VLAN 3
Impresoras	VLAN 4
Zona desmilitarizada, servidores DMZ	VLAN 10
SAI	VLAN 13
Administración	VLAN 20
Estación de ingeniería (EI)	VLAN 21
Acceso Total (Mantenimiento)	VLAN 100

Cada una de las zonas identificadas se asigna a una única VLAN (red de área local virtual).

Una VLAN es un método de creación de redes lógicas independientes dentro de una misma red física, de esta forma se crean subredes independientes.

- **Identificación de los Conductos:**

Los conductos están formados por los equipos que conforman la arquitectura de la red. Se detallan en profundidad en el punto 1.4.3.

- **Centralización del acceso a Internet:**

Otra consideración a tener en cuenta es la distribución de los distintos puntos de acceso a Internet. En la anterior arquitectura se disponían de distintos puntos de acceso distribuidos

por los armarios de las plantas. Estos módems se han retirado y se han centralizado las conexiones a internet desde el armario del CPD, de esta forma reducimos tenemos controlados los puntos de acceso, con lo que conseguimos un mayor nivel de seguridad además de permitir configurarse desde una misma interfaz.

Recogiendo todos estos conceptos y consideraciones, se diseña un plano de la red reformado con los equipos necesarios para gestionar la red y la seguridad de la misma, siguiendo las especificaciones del estándar ISA99/IEC6443 y los requisitos de seguridad y operación que se han detectado.

### 1.4.3. Equipos instalados y reforma de la arquitectura de red.

Para llevar a cabo la reforma de la red se han instalado los equipos que se detallan a continuación y que están ubicados según el plano de la red reformado.

#### - Armario de servidores CPD:

En el armario de servidores marcado en el plano como “Armario rack CPD” se incluirán los siguientes equipos:

- 1 x Firewall FortiGate 100E
- 2 x SFP Transceivers de fibra óptica
- 3 x Switch 24 bocas RJ45 + 4 bocas slots SFP
- 7 x SFP Transceivers de fibra óptica
- 30 x latiguillos de cobre RJ45 Cat 6A
- 8 x latiguillos de fibra óptica



#### - Armarios de planta:

En cada uno de los armarios principales de planta existentes donde se encuentran los switches industriales actuales, se instalará un switch Fortinet.

A este nuevo switch se llevará todo el tráfico que no precise tiempo real (PC's cliente del SCADA, PC's ofimáticos, impresoras, cámaras IP).

El tráfico Ofimático estará desplegado por VLAN's separadas de la red de supervisión.

Los equipos incluidos en esta partida son:

- 8 x Switch 24 bocas RJ45 + 4 bocas slots SFP
- 18 x SFP Transceivers de fibra óptica
- 30 x latiguillos de cobre RJ45 Cat 6A
- 19 x latiguillos de fibra óptica
- Suministro de Armario RACK para pared para colocación del Switch de las oficinas comerciales



#### - Puntos de acceso Wifi:

Se incluyen puntos de acceso Wifi de la gama profesional de Fortinet para una densidad de tráfico media, incluyendo ya el nuevo estándar Wifi con doble radio 802.11ac.

Incluye punto de acceso de wifi de Invitados.

El que se instalará en la zona comercial tiene una forma similar a un detector de incendios, para poder montarlo en el techo y que pase desapercibido.

Los equipos que se instalan son:

- 1 x Punto de acceso Indoor Wireless AP con formato de detector de incendios para montaje en techo.



- 1 x Punto de acceso Indoor Wireless AP con 4 antenas de alta ganancia.



### 1.4.3.1. Distribución de los equipos en el armario CPD

Se realiza una distribución del armario principal CPD tal y como se muestra a continuación:

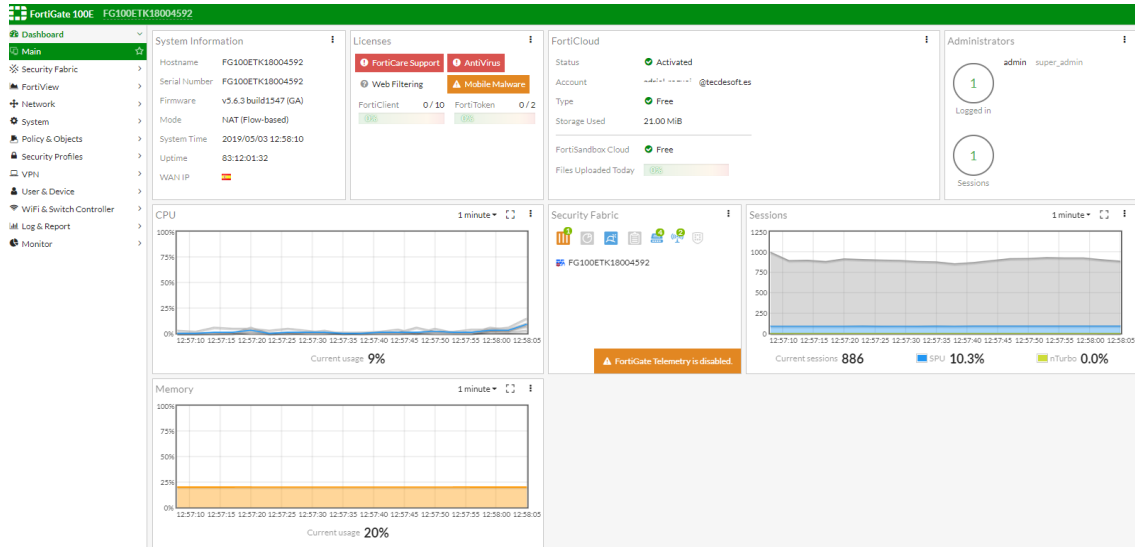
FRONTAL DE ARMARIO		TRASERA DE ARMARIO	
1	PASAJILLOS	1	
2	PATCH PANEL	2	
3	PATCH PANEL	3	
4	PASAJILLOS	4	BANDEJA PARA ROUTERS
5	PATCH PANEL	5	
6	PATCH PANEL RESERVA	6	
7	PASAJILLOS	7	PASAJILLOS
8	FIREWALL FORTINET	8	PANEL TERMICOS 230Vac
9	SWITCH RESERVA	9	
10	PASAJILLOS	10	PASAJILLOS
11	SWITCH FORTINET IT 1	11	
12	SWITCH FORTINET IT 2	12	
13	PASAJILLOS	13	
14	SWITCH RESERVA	14	PLACA DE MONTAJE PARA SWITCHES CARRIL DIN
15	SWITCH RESERVA	15	
16	PASAJILLOS	16	
17	BANDEJA F.OPTICA SC	17	
18	BANDEJA F.OPTICA SC	18	
19	PASAJILLOS	19	PASAJILLOS
20	BANDEJA F.OPTICA SC	20	
21	BANDEJA F.OPTICA SC RESERVA	21	
22	PASAJILLOS	22	
23	RESERVA	23	
24	RESERVA	24	
25	RESERVA	25	
26	RESERVA	26	
27	RESERVA	27	
28	SERVIDOR MAQUINAS	28	
29		29	
30	SERVIDOR REDUNDANTE 1	30	
31		31	
32	SERVIDOR REDUNDANTE 2	32	
33		33	
34	SERVIDOR RESERVA	34	
35		35	
36	HUECO AIREACION SAI	36	
37	SAI ONLINE	37	
38		38	
39	HUECO AIREACION SAI	39	TOMA 8 ENCHUFES 16A
40		40	TOMA 8 ENCHUFES 16A
41	BATERIAS SAI	41	TOMA 8 ENCHUFES 16A
42		42	

### 1.4.4. Configuración de los equipos instalados

Los equipos instalados se configuran desde el Firewall Fortigate 100E que es el encargado de gestionar las VLANs, establecer las políticas de seguridad, monitorizar la red y demás funcionalidades.

Para realizar la configuración del mismo se debe de acceder al portal web desde un navegador.

Una vez logueados en el panel general podemos ver el estado del sistema, licencias, uso de CPU, Memoria, Sesiones activas, etc...



#### 1.4.4.1. Configuración de las VLANs

Para configurar las VLANs se debe acceder al panel Network -> Interfaces, se selecciona el puerto físico de conexión y a continuación se pulsa en Create New.



FortiGate 100E FG100ETK18004592

Dashboard > Security Fabric > FortiView > **Network** > **+** Create New > Edit > Delete

**Interfaces**

	Status	Name
<b>Hardware Switch (1)</b>		
		lan (Red Control)
<b>Physical (22)</b>		
	+	dmz
	+	ha1
	+	ha2
	+	mgmt
	+	<b>port15 (10 Connected FortiSwitch(s))</b>
		VLAN3 (Oficina Técnica)
		VLAN4 (Impresoras)
		VLAN5 (ThinClient)
		VLAN10 (Servidores DMZ)
		VLAN13 (Gestión)
		VLAN14 (Cámaras)
		VLAN20 (Administración)
		VLAN21 (Ei)
		VLAN25 (PLC2)
		VLAN100 (Acceso total)
		VLAN200 (Neutral)
		vsw.port15 (FGT)
	+	port16 (Acceso total)
	+	wan1 (Intermax)
	+	wan2 (4G)
<b>WiFi (4)</b>		
		SSID Admin invi (6) SSID: Veree Admin Invitados

Aparece una ventana de configuración donde se le da nombre a la VLAN, se configura la IP y se ajusta según los accesos que deseamos tener disponibles.

**FortiGate 100E FG100ETK18004592**

- Dashboard
- Security Fabric
- FortiView
- Network**
  - Interfaces**
  - DNS
  - DNS Servers
  - SD-WAN
  - SD-WAN Status Check
  - SD-WAN Rules
  - Static Routes
  - Policy Routes
  - RIP
  - OSPF
  - BGP
  - Multicast
- System
- Policy & Objects
- Security Profiles
- VPN
- User & Device
- WiFi & Switch Controller
- Log & Report
- Monitor

### New Interface

Interface Name:

Alias:

Type:

Interface:

VLAN ID:

Color:

Role:

#### Address

Addressing mode:  Manual  DHCP  PPPoE

IP/Network Mask:

#### Administrative Access

IPv4  HTTPS  SSH  FortiTelemetry  HTTP  SNMP  PING  FTM  FMG-Access  CAPWAP  RADIUS Accounting

#### DHCP Server

DHCP Server

#### Networked Devices

Device Detection:

IGMP Snooping:

DHCP Snooping:

#### Admission Control

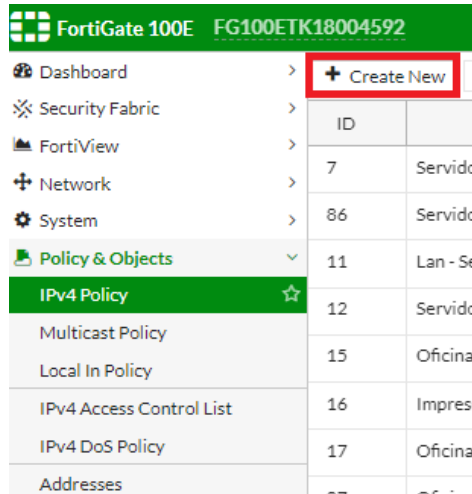
Security Mode:

Una vez configurada en el panel de Interfaces se aprecia un resumen donde se puede ver a simple vista las VLANs configuradas y el tipo de acceso permitido en cada una de ellas, así como también sus direcciones IP.

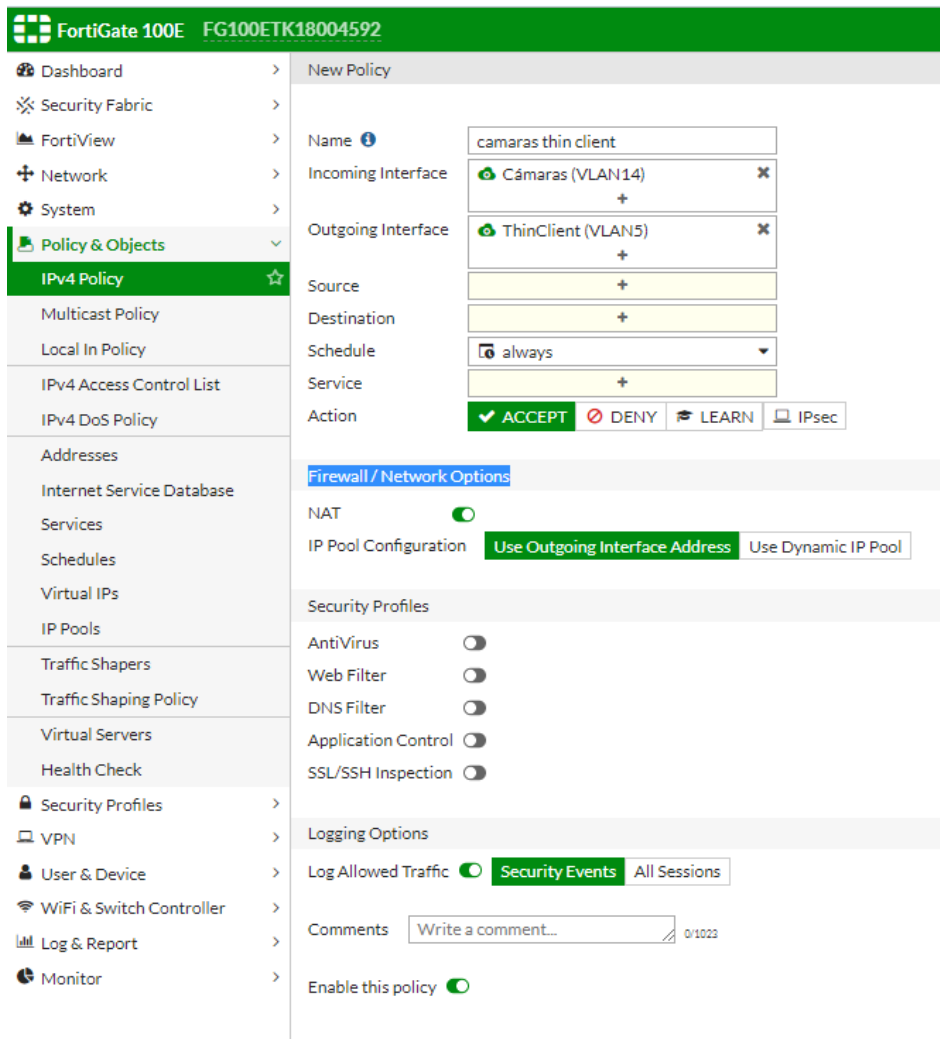
Status	Name	Members	IP/Netmask	Type	Access
<b>Hardware Switch (1)</b>					
	lan (Red Control)		192.168.1.1 / 255.255.240.0	Hardware Switch (14)	PING HTTPS SSH HTTP FMG-Access CAPWAP
<b>Physical (22)</b>					
	dmz		10.10.1.1 / 255.255.255.0	Physical Interface	PING HTTPS HTTP FMG-Access CAPWAP
	ha1		0.0.0.0 / 0.0.0.0	Physical Interface	
	ha2		0.0.0.0 / 0.0.0.0	Physical Interface	
	mgmt		192.168.1.1 / 255.255.255.0	Physical Interface	PING HTTPS SSH HTTP FMG-Access
	port15 (10 Connected FortiSwitch(s))		Dedicated to FortiSwitch	Physical Interface	PING CAPWAP
	VLAN3 (Oficina Técnica)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN4 (Impresoras)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN5 (ThinClient)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN10 (Servidores DMZ)		192.168.1.1 / 255.255.255.0	VLAN	PING HTTPS SSH SNMP HTTP
	VLAN13 (Gestión)		192.168.1.1 / 255.255.255.0	VLAN	PING CAPWAP
	VLAN14 (Cámaras)		192.168.1.1 / 255.255.255.0	VLAN	SSH
	VLAN20 (Administración)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN21 (Ei)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN25 (PLC2)		192.168.1.1 / 255.255.255.0	VLAN	PING
	VLAN100 (Acceso total)		192.168.1.1 / 255.255.255.0	VLAN	PING HTTPS SSH SNMP HTTP CAPWAP
	VLAN200 (Neutral)		0.0.0.0 / 0.0.0.0	VLAN	
	vsw.port15 (FGT)		192.168.1.1 / 255.255.255.0	VLAN	PING CAPWAP
	port16 (Acceso total)		192.168.1.1 / 255.255.255.0	Physical Interface	PING HTTPS SSH SNMP HTTP CAPWAP
	wan1 (Intermax)		178.60.1.1 / 255.255.255.255	Physical Interface	
	wan2 (4G)		192.168.1.1 / 255.255.255.0	Physical Interface	PING FMG-Access
<b>WiFi (4)</b>					
	SSID Admin Invi (SSID: Veree Admin Invitados)		192.168.1.1 / 255.255.255.0	WiFi SSID	PING
	SSID Invitados (SSID: Veree Invitados)		192.168.1.1 / 255.255.255.0	WiFi SSID	

### 1.4.4.2. Configuración de las políticas de seguridad

Para configurar las políticas de seguridad del firewall se debe acceder al panel Policy & Objects -> IPv4 Policy y pulsar sobre Create New



En el panel que se despliega para configurar la nueva política configuramos un nombre, la interfaz de entrada y salida, la planificación de la política y el servicio que queremos establecer acompañado de la acción de permiso o denegación.



Si no se especifica un servicio en concreto por defecto se asume que son todos.

Por último en el Panel IPv4 Policy aparece un resumen con todas las políticas establecidas en el firewall.

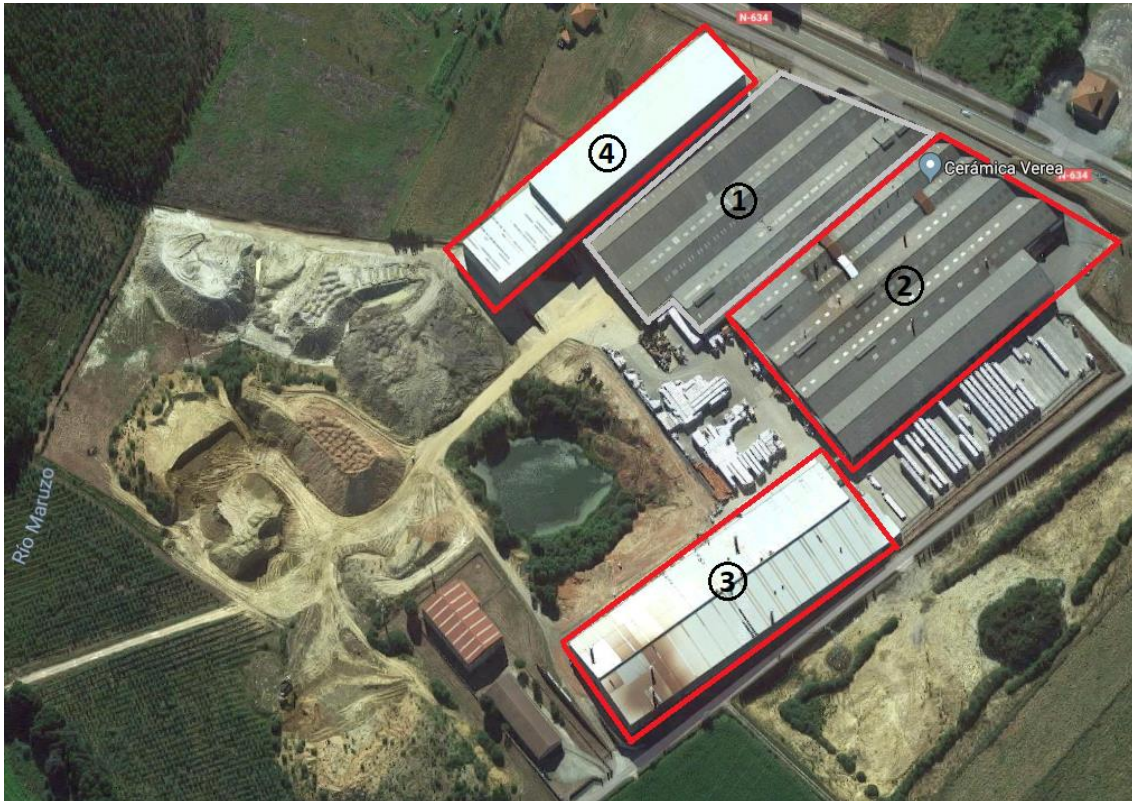
ID	Name	From	To	Source	Destination	Schedule	Service	Action
7	Servidores DMZ- Internet	Servidores DMZ (VLAN10)	Intermax (wan1)	all	all	always	ALL	ACCEPT
86	Servidores DMZ-Internet2	Servidores DMZ (VLAN10)	4G (wan2)	all	all	always	ALL	ACCEPT
11	Lan - Servidores	Red Control (lan)	Servidores DMZ (VLAN10)	all	all	always	ALL	ACCEPT
12	Servidores - Lan	Servidores DMZ (VLAN10)	Red Control (lan)	all	all	always	ALL	ACCEPT
15	Oficina - Impresoras	Oficina Técnica (VLAN3)	Impresoras (VLAN4)	all	all	always	ALL	ACCEPT
16	Impresoras - Oficina	Impresoras (VLAN4)	Oficina Técnica (VLAN3)	all	all	always	ALL	ACCEPT
17	Oficina - Internet	Oficina Técnica (VLAN3)	Intermax (wan1)	all	all	always	ALL	ACCEPT
87	Oficina-Internet2	Oficina Técnica (VLAN3)	4G (wan2)	all	all	always	ALL	ACCEPT
21	SSID Oficina - Internet	Verea (SSID PC Oficina)	Intermax (wan1)	all	all	always	ALL	ACCEPT
88	SSID Oficina-Internet2	Verea (SSID PC Oficina)	4G (wan2)	all	all	always	ALL	ACCEPT
25	SSID Oficina - Impresoras	Verea (SSID PC Oficina)	Impresoras (VLAN4)	all	all	always	ALL	ACCEPT
27	SSID Invitados - Internet	Verea Invitados (SSID Invitados)	Intermax (wan1)	all	all	always	ALL	ACCEPT
89	SSID Invitados-Internet2	Verea Invitados (SSID Invitados)	4G (wan2)	all	all	always	ALL	ACCEPT
33	Acceso - Control	Acceso total (port16)	Red Control (lan)	all	all	always	ALL	ACCEPT
34	Acceso - DMZ	Acceso total (port16)	Servidores DMZ (VLAN10)	all	all	always	ALL	ACCEPT
35	Acceso-Gestion	Acceso total (port16)	Gestión (VLAN13)	all	all	always	ALL	ACCEPT
36	DMZ-Gestión	Servidores DMZ (VLAN10)	Gestión (VLAN13)	all	all	always	ALL	ACCEPT
37	VLAN100-DMZ	Acceso total (VLAN100)	Servidores DMZ (VLAN10)	all	all	always	ALL	ACCEPT
38	VLAN100-Gestión	Acceso total (VLAN100)	Gestión (VLAN13)	all	all	always	ALL	ACCEPT
39	VLAN100-Red Control	Acceso total (VLAN100)	Red Control (lan)	all	all	always	ALL	ACCEPT
40	DMZ-Acceso	Servidores DMZ (VLAN10)	Acceso total (port16)	all	all	always	ALL	ACCEPT
41	Temp1	Acceso total (port16)	Oficina Técnica (VLAN3)	all	all	always	ALL	ACCEPT
44	Acceso - Internet	Acceso total (port16)	Intermax (wan1)	all	all	always	ALL	ACCEPT
90	Acceso-Internet2	Acceso total (port16)	4G (wan2)	all	all	always	ALL	ACCEPT
45	Oficina - Control	Oficina Técnica (VLAN3)	Red Control (lan)	all	all	always	ALL	ACCEPT

Si se desea modificar alguna de ellas basta con seleccionarla y pulsar Edit para que se abra el panel de edición.

## 1.5 Descripción del proceso productivo y la planta de producción.

La fábrica objeto de este proyecto produce distintos tipos de tejas cerámicas para cubiertas y el SCADA que se pretende desarrollar como parte de este TFM contempla todo el proceso de producción, desde el tratamiento y molienda de los distintos tipos de tierras de las que se compone la cerámica hasta el proceso de conformado de las piezas, secado, horneado, hasta llegar a tener los productos embalados y almacenados, listos para su distribución.

La fábrica se divide en 4 plantas claramente diferenciadas por su proceso que se detallan a continuación con la finalidad de facilitar la comprensión del flujo del producto y por consiguiente para el desarrollo del SCADA.



Distribución en planta de la fábrica

- **Planta de tierras (4):** Es el lugar destinado al almacenamiento y procesado de la arcilla que se empleará para conformar las tejas. Está compuesta por una red de cintas transportadoras que distribuyen la tierra desde la entrada del material que se realiza vertiendo la tierra con una retroexcavadora en los alimentadores, hasta las tolvas o el pudridero que almacena distintos tipos de arcilla, pasando por los molinos y los laminadores (que reducen la granulometría de la tierra). El pudridero está compuesto por un carro y una draga para mover la tierra a donde corresponda en cada momento.

Desde la planta de tierras se distribuye el material a las tolvas de las fábricas 1, 2 y 3 para continuar con el proceso de fabricación.

En la siguiente Figura puede apreciarse los distintos caminos que puede tomar el material en la planta de tierras.

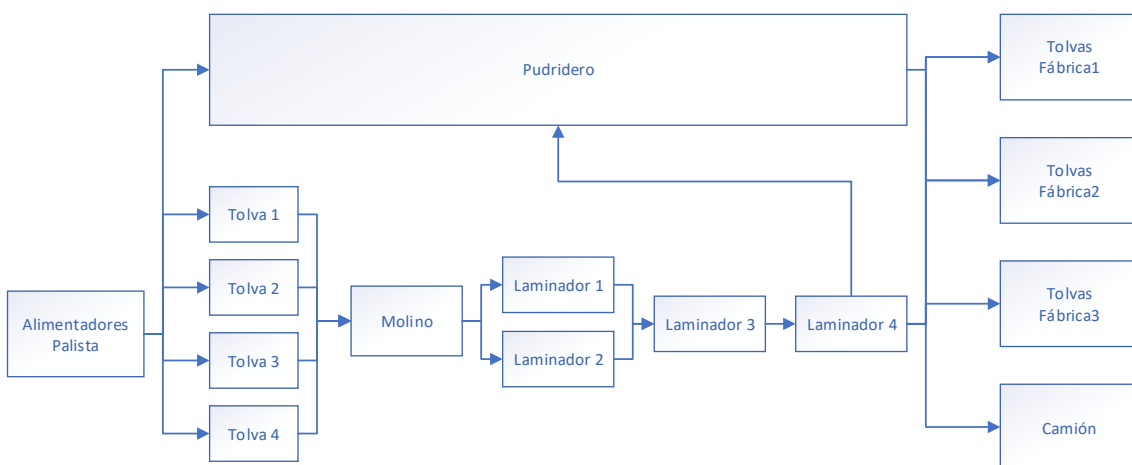


Diagrama de flujo Planta de tierras

- **Fábrica (3):** Se transporta el material desde las tolvas hasta la amasadora donde se le añade agua principalmente, entre otros químicos para conseguir una masa que posteriormente pasará a través de una extrusora donde se conforma el producto. Una vez conformado el producto final, este es apilado por robots en bandejas que se introducirán en el secadero para reducir la humedad de la teja, posteriormente se apilan en bandejas y se introducen en el horno para realizar el tratamiento térmico del producto. El proceso continúa con el desapilado de las bandejas mediante robots a la salida del horno y su disposición para proceder al flejado de las tejas en grupos, después se colocan en palets de madera y se procede a su embalaje y etiquetado, consiguiendo de esta forma el producto terminado que se dispondrá en el almacén.

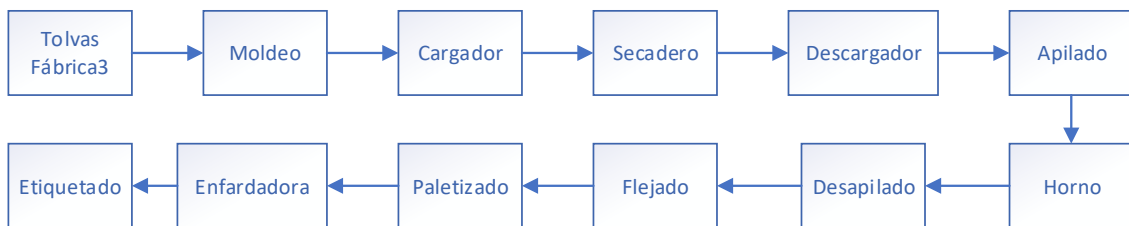


Diagrama de flujo Fábrica 3

- **Fábrica (2):** El proceso y flujo del producto es similar al de la fábrica 3 aunque esta línea de fabricación todavía no está prevista para realizar el SCADA.

-**Fábrica (1):** La fábrica 1 es un espacio reservado que se utiliza actualmente como un almacén provisional y taller para la maquinaria. Está previsto la instalación en un futuro de una línea de producción de similares características a las de la fábrica 3 pero se escapa del objetivo de este proyecto.

## 1.6 Desarrollo del sistema SCADA

### 1.6.1. Requisitos de diseño

El sistema SCADA debe cumplir con los siguientes requisitos especificados por el cliente de forma explícita:

- Es necesario que el sistema que se despliegue tenga alta disponibilidad de forma que si el servidor principal no pudiese dar servicio, sincronice los datos con otro servidor de forma automática de manera que se pueda seguir produciendo normalmente.
- Los equipos (ordenadores y paneles de operación) de los puestos de trabajo de los trabajadores de la planta no pueden contener información sensible o que dependan del hardware instalado en ellos ya que en caso de accidente podría ponerse en riesgo la operatividad de la planta si no se cuenta con una copia de seguridad del equipo.
- Deberá tenerse en cuenta que el sistema SCADA debe tener la capacidad de manejar hasta 1000 productos diferentes, actualmente no se cuenta con tal cantidad de productos, pero debe dejarse previsto para ello.
- Las pantallas de mando de los objetos tales como motores, válvulas, registros, etc... deberán de funcionar de la siguiente forma. Al hacer click para abrir el mando, este no debe tapar el objeto en cuestión. Cuando se abra el mando solo

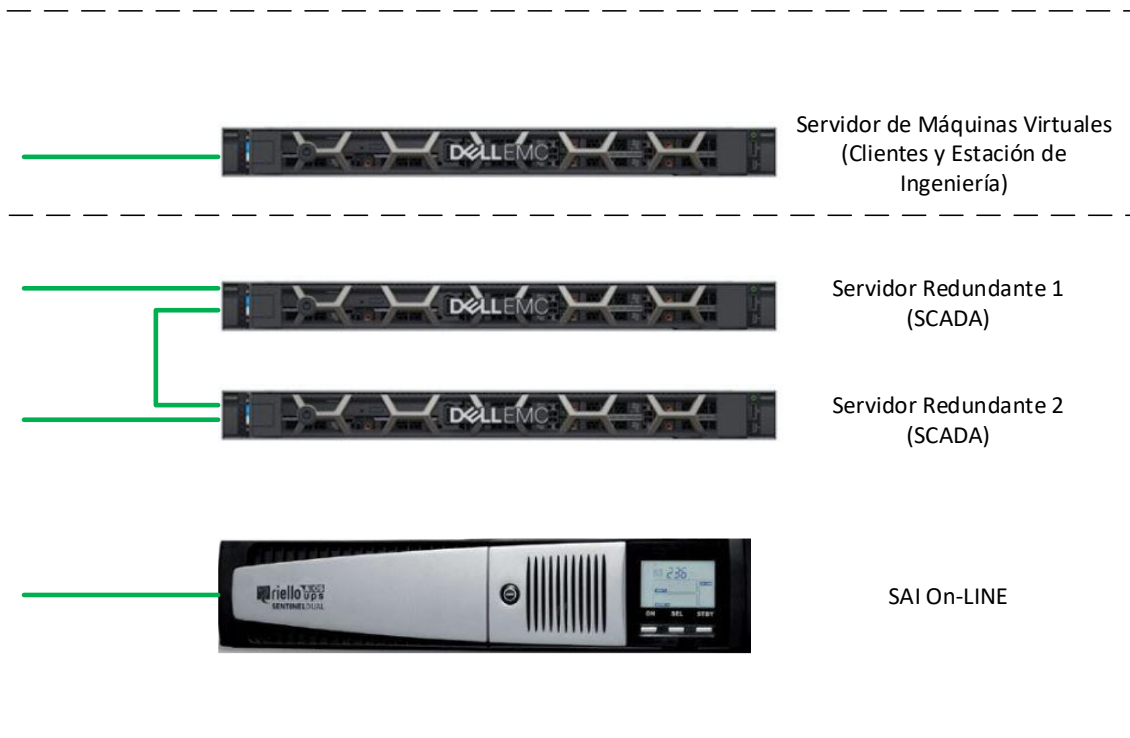
deberá mostrar el estado del elemento, los botones para su manejo, y en la esquina superior derecha un botón que permita ampliar la ventana de mando para ver a su izquierda la lista de posibles alarmas para cada uno de ellos, a la derecha las informaciones de estado, y en la parte inferior la lista de las alarmas correspondientes al objeto. Una vez abierta la pantalla de mando deberá poder ampliarse y reducirse indefinidamente.

- Al pulsar en el fondo de los sinópticos generales deberán cerrarse las pantallas de mando o ventanas que estén abiertas. Esto facilita mucho la operación desde los dispositivos móviles.
- El SCADA deberá poder funcionar tanto en dos monitores independientes si se requiere, aunque en la mayoría de los puestos de operación funcionará solo con un monitor o panel táctil.
- El sistema SCADA tendrá una resolución de 1920x1080p y podrá manejarse desde dispositivos móviles compatibles con esa resolución, tales como teléfonos móviles y tabletas que usa el personal de mantenimiento y supervisión de la planta.
- Se requiere que se pueda acceder al SCADA desde una ubicación remota de forma segura y manejarse de igual forma, como si se estuviese en un puesto de operación en planta.
- Se establece el siguiente código de colores que indican el estado de funcionamiento de los objetos:
  - **Rojo:** Indica el estado de funcionamiento apagado
  - **Verde:** Indica que el elemento está encendido
  - **Amarillo:** Indica que el elemento está en alarma porque ha ocurrido un fallo.

### 1.6.2. Arquitectura SCADA desarrollada.

La solución de arquitectura propuesta consiste en desplegar un sistema SCADA redundante empleando máquinas virtuales a las que se accederá remotamente desde los puestos de operación. De esta forma conseguimos tener la información centralizada y que los equipos de los puestos de operación no contengan ningún tipo de información más que la configuración necesaria para conectarse remotamente al cliente desde el cual manejará el SCADA.

Se plantea la siguiente solución:



- Servidor de Máquinas Virtuales (3 Clientes y Estación de Ingeniería): Es aquel servidor dedicado para instalar las siguientes máquinas:
  - o **Estación de Ingeniería:** Es la máquina virtual con el sistema operativo Windows 10 que contiene el software necesario para desarrollar el sistema SCADA en WinCC y cargarlo en los servidores SCADA.
  - o **Cliente Molienda:** Es una máquina virtual con Windows 10 configurada con el software de WinCC Runtime en modo cliente sin proyecto propio, de forma que se pueda ejecutar el SCADA en modo cliente. Licencia individual de Cliente Simatic WinCC.
  - o **Cliente Horno3:** Al igual que Cliente Molienda esta máquina contiene el mismo sistema operativo y el mismo software, solo cambia la dirección de acceso a la misma. Licencia individual de Cliente Simatic WinCC.
  - o **Cliente Laboratorio:** Al igual que los otros 2 clientes, tiene el mismo sistema operativo y el mismo software, solo cambia la dirección de acceso a la misma y la licencia individual de Cliente Simatic WinCC.
  - o **Cliente WebNavigator:** Es una máquina virtual con Windows 10 y licencias de escritorio remoto de Windows que permiten la conexión de hasta 13 equipos simultáneamente. Esta máquina contiene el navegador Internet Explorer configurado para que se pueda visualizar el SCADA desde el. Está pensada para conectarse remotamente desde los paneles de operador distribuidos por toda la planta de producción.
  
- Servidor Redundante 1 (SCADA-A): Es un servidor dedicado que contiene únicamente una máquina virtual con Windows Server 2016 y el software WinCC Runtime configurado para ejecutar la aplicación del SCADA en modo servidor. Licencia individual de Servidor Simatic WinCC.
  
- Servidor Redundante 2 (SCADA-B): Es un servidor dedicado que contiene únicamente una máquina virtual con Windows Server 2016 y el software WinCC



Runtime configurado para ejecutar la aplicación del SCADA en modo servidor.  
Licencia individual de Servidor Simatic WinCC.

- SAI On-LINE: En caso de un corte de la red eléctrica, el SAI será el encargado de proporcionar la energía necesaria durante un período de tiempo determinado para alimentar los servidores.

### 1.6.2.1. Hipervisor VMware ESXi

Para la virtualización de todas las máquinas virtuales que se instalan en el servidor se emplea el hipervisor de VMware ESXi cuyas características y prestación se detallan a continuación.



Un hipervisor es una plataforma que permite la virtualización de diferentes sistemas operativos en un mismo hardware consiguiendo un mejor aprovechamiento de los recursos, así como también gestionar las copias de seguridad de los sistemas operativos de una forma fácil y segura.

VMware ESXi es un hipervisor nativo o *baremetal* que se ejecuta directamente sobre el hardware y se instala directamente sobre el servidor físico y permite dividirlo en varios servidores lógicos denominados máquinas virtuales.

#### Características:

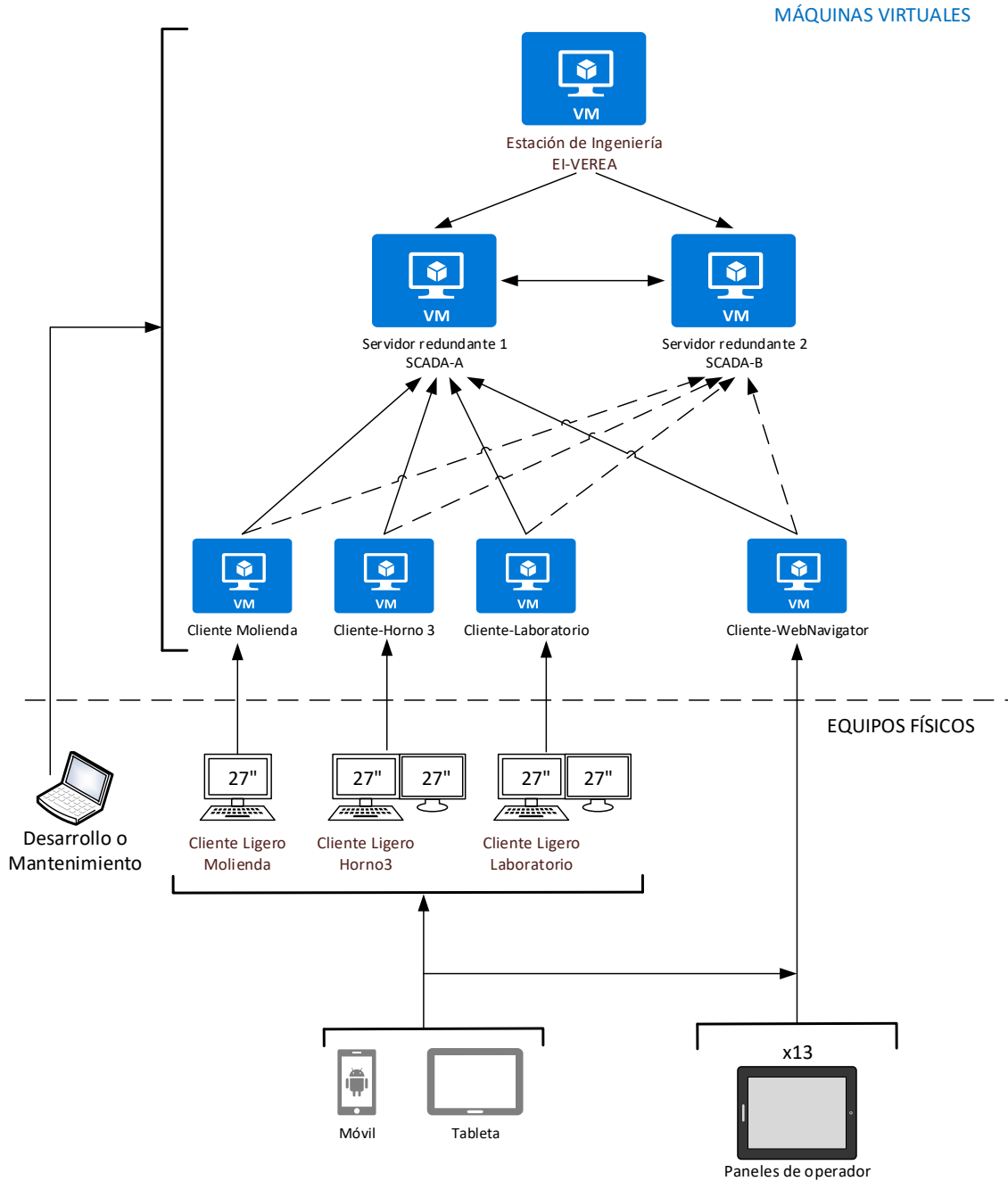
- *Implementación y configuración optimizadas*: La configuración e implementación de VMware ESXi es sencilla con lo que se consigue una infraestructura virtual homogénea que facilita el mantenimiento de las máquinas virtuales.
- *Fiabilidad y seguridad*: el hipervisor ESXi está integrado en VMkernel que tiene un tamaño muy reducido (150Mb) lo que ofrece un impacto de ataque reducido para los programas maliciosos y amenazas de la red. También permite crear y configurar permisos de administración a usuarios y grupos de usuarios.
- *Registros y auditorías*: tiene la opción de registrar toda la actividad que llevan a cabo los usuarios, tanto desde la Shell como de la interfaz de usuario. De esta forma se permite gestionar la responsabilidad y auditar las acciones por parte de los usuarios.

#### Prestaciones que admiten las máquinas virtuales que se ejecuten sobre VMware ESXi:

- Hasta 128 CPU virtuales
- Hasta 4 TB de RAM
- Controladora xHCI admite dispositivos USB 3.0
- Advanced Host Controller Interface (AHCI) admite hasta 120 dispositivos por máquina virtual.
- Tamaño máximo de los archivos VMDK es de 62 TB.

### 1.6.2.2. Diagrama de la arquitectura SCADA.

El diagrama de la arquitectura desarrollada se representa en la siguiente figura.



La estación de ingeniería (EI-VEREA) está destinada al desarrollo de la aplicación SCADA y se permite realizar cambios y ampliaciones en el proyecto empleando WinCC Explorer y también cargarlo en los servidores SERVER-A y SERVER-B desde Step7.

Los servidores redundantes SCADA-A y SCADA-B contienen la aplicación del proyecto ejecutándose en modo servicio ya que en los servidores no es necesario la visualización de la aplicación y de esta forma se consigue incrementar un rendimiento en la ejecución del mismo del 30% según Siemens, de todas formas si fuese necesario activar la visualización

puntualmente podría realizarse activando WinCC Graphics Runtime y abrir manualmente la aplicación gráfica para manejar el SCADA.

Si uno de los servidores redundantes deja de estar en servicio, se ha configurado el sistema para que automáticamente se lleve a cabo la sincronización de los datos del SCADA y posteriormente entra en servicio el servidor que se encontraba en standby y los clientes pueden seguir en producción mientras se repara el servidor averiado.

Las máquinas virtuales Clientes son las encargadas de conectarse al proyecto del servidor que esté activo, 3 de ellas lo hacen mediante WinCC Runtime y otra de ellas mediante WebNavigator, debido a que las licencias de WebNavigator son más baratas que las de cliente de WinCC Runtime.

Con esta arquitectura se consigue que toda la información que genera el sistema SCADA se almacene en los servidores y que los equipos desde los que se conectan a los clientes no estén expuestos a posibles pérdidas de información por averías o posibles accidentes en la planta de producción.

Los equipos físicos que se conectan a las máquinas virtuales cliente con WinCC Runtime se denominan Clientes Ligeros o *Thin Clients* y desde ellos se establece una sesión de escritorio remoto para poder manejar la máquina virtual cliente a la que se conecta en la cual se ejecuta el SCADA.

A las máquinas virtuales cliente también pueden conectarse dispositivos móviles y tabletas cuando sea necesario.

A la máquina virtual cliente con WebNavigator se conectan los paneles de operador distribuidos por distintos puntos de operación.

### 1.6.2.3. Clientes ligeros o *Thin Clients*

Los Clientes Ligeros o *Thin Clients* son ordenadores de reducidas dimensiones y con unos recursos mínimos para configurar la conexión de red y establecer una conexión por escritorio remoto con la máquina virtual que tiene la aplicación del SCADA ejecutándose. Se ha decidido emplear clientes ligeros en los puestos de operación porque de esta forma la información solo se encuentra en las máquinas virtuales y no en los equipos de campo que pueden sufrir accidentes con la consiguiente pérdida de información.

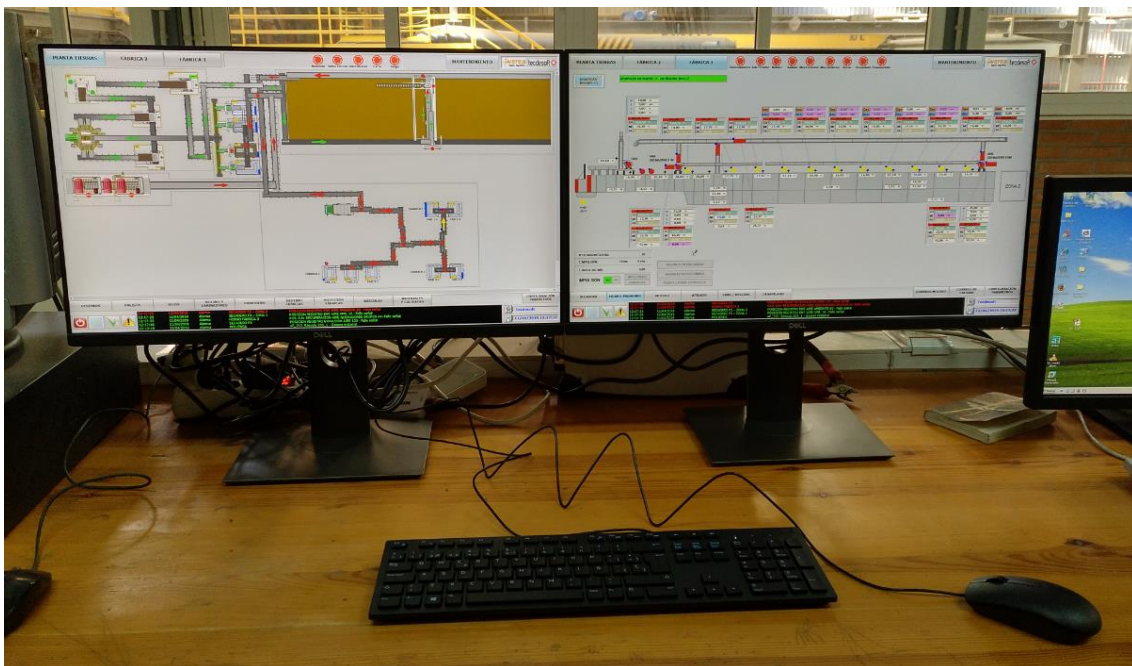
En la planta hay 3 puestos de operación donde se instalarán los clientes ligeros:

- **Puesto de operación de Molienda:**  
Se instalará una pantalla de 27" y un cliente ligero AXEL AX90 junto con un teclado y ratón Logitech.



Equipos del puesto de operación de Molienda

- **Puestos de operación de Laboratorio y Horno3:**  
En cada uno de estos puestos de operación se instalarán dos pantallas de 27" y un cliente ligero AXEL AX90 junto con un teclado y ratón Logitech.



Instalación real en el puesto de operación de Horno3

### 1.6.3. Programas utilizados

Para el desarrollo de la programación y simulación se han usado el siguiente software:

- **WinCC v7.4 SP1 + Upd10:** empleado para programar el SCADA
- **ModScan:** permite visualizar y forzar los valores de las direcciones Modbus para realizar simulaciones y verificar la correcta programación.
- **Concept v2.6:** empleado para abrir y cargar los programas de los PLCs Modicon en los equipos reales para verificar el correcto funcionamiento del SCADA con el PLC correspondiente.

## 1.7 Normativa y referencias

### 1.7.1. Normas y disposiciones legales aplicadas

Se han empleado las normas y disposiciones legales que se citan a continuación para la elaboración de la documentación técnica en cuanto a principios generales de representación, ejecución de planos, indicaciones, escritura, rotulación, acotación, símbolos gráficos, plegado, listas de elementos gráficos y gestión de la información técnica asistida por ordenador.

- **UNE 1027.** Dibujo Técnico. Plegado de planos.
- **UNE 1032.** Dibujos Técnicos. Principios generales de representación.
- **UNE 1039.** Dibujos Técnicos. Acotación. Principios generales, definiciones, métodos de ejecución e indicaciones especiales.
- **UNE 1135.** Dibujos Técnicos. Lista de elementos.
- **UNE 1166-1.** Documentación técnica de productos. Vocabulario. Parte 1: Términos relativos a los dibujos técnicos: generalidades y tipos de dibujo.
- **UNE-EN-ISO 3098-0.** Documentación técnica de productos. Escritura. Requisitos generales. (ISO 3098-0:1997).
- **UNE-EN-ISO 3098-2.** Documentación técnica de productos. Escritura. Parte 2: Alfabeto latino, números y signos. (ISO 3098-2:2000).
- **UNE-EN-ISO 3098-3.** Documentación técnica de producto. Escritura. Parte 3: Alfabeto griego. (ISO 3098-3:2000).
- **UNE-EN-ISO 3098-4.** Documentación técnica de producto. Escritura. Parte 4: Signos diacríticos y particulares del alfabeto latino. (ISO 3098-4:2000).
- **UNE-EN-ISO 3098-5.** Documentación técnica de productos. Escritura. Parte 5: Escritura en diseño asistido por ordenador (DAO), del alfabeto latino, las cifras y los signos. (ISO 3098-5:1997).
- **UNE-EN-ISO 3098-6.** Documentación técnica de producto. Escritura. Parte 6: Alfabeto cirílico. (ISO 3098-6:2000).
- **UNE-EN-ISO 5455.** Dibujos Técnicos. Escalas. (ISO 5455:1979).
- **UNE-EN-ISO 5456-1.** Dibujos Técnicos. Métodos de Proyección. Parte 1: Sinopsis. (ISO 5456-1:1996).
- **UNE-EN-ISO 5456-2.** Dibujos técnicos. Métodos de proyección. Parte 2: Representaciones ortográficas. (ISO 5456-2:1996).
- **UNE-EN-ISO 5456-3.** Dibujos técnicos. Métodos de proyección. Parte 3: Representaciones axonométricas. (ISO 5456-3:1996).
- **UNE-EN-ISO 5457.** Documentación técnica de producto. Formatos y presentación de los elementos gráficos de las hojas de dibujo. (ISO 5457:1999).
- **UNE-EN ISO 6433.** Dibujos técnicos. Referencia de los elementos. (ISO 6433:1981).
- **UNE EN ISO 7200.** Documentación técnica de productos. Campos de datos en bloques de títulos y en cabeceras de documentos.
- **UNE-EN 80416-1:2011.** Principios básicos para los símbolos gráficos utilizables en los equipos. Parte 1: Creación de símbolos gráficos para registro.
- **UNE-EN 80416-2:2003.** Principios básicos para los símbolos gráficos utilizables en los equipos. Parte 2: Formas y utilización de las flechas. (ISO 80416-2:2001)
- **UNE-EN 80416-3:2003.** Principios básicos para los símbolos gráficos utilizables en los equipos. Parte 3: Guía para la aplicación de los símbolos gráficos.
- **UNE-EN-ISO 10209-2.** Documentación técnica de producto. Vocabulario. Parte 2: Términos relacionados con los métodos de proyección. (ISO 10209-2:1993).

- **UNE-EN ISO 11442:2006.** Documentación técnica de productos. Gestión de documentos (ISO 11442:2006).
- **UNE-EN ISO 81714-1:2010.** Diseño de símbolos gráficos utilizables en la documentación técnica de productos. Parte 1: Reglas fundamentales. (ISO 81714-1:2010)

## 1.7.2. Referencias y bibliografía

Se han empleado las siguientes referencias bibliográficas como material de consulta para la documentación del trabajo.

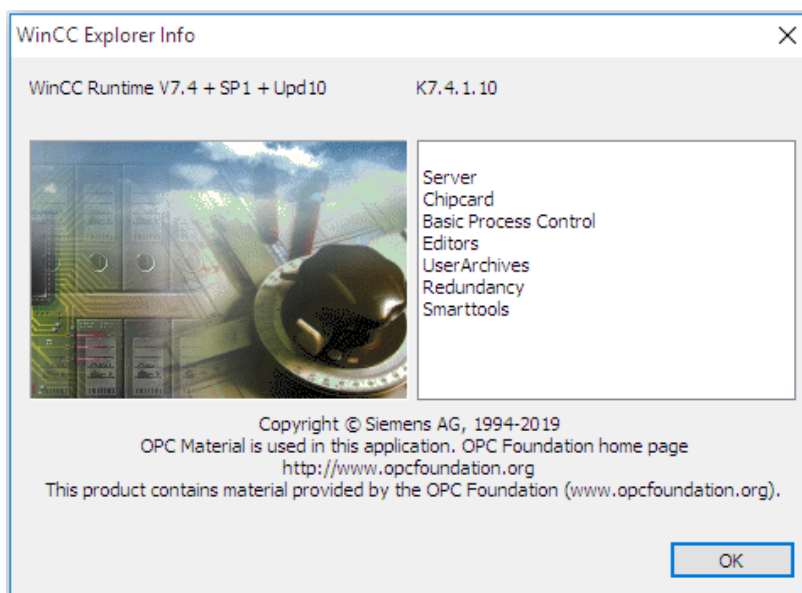
- “Características de vSphere ESXi Hypervisor” – VMware, <https://www.vmware.com/es/products/esxi-and-esx.html>
- “Estableciendo zonas y conductos según el estándar ISA99/IEC6443” – Centro de Ciberseguridad Industrial, <https://www.cci-es.org/documents/10694/613683/Establecimientos+zonas+y+conductos.pdf/a479e3db-81f4-43c1-b5d1-f9e5f7754bcf>
- Manual de sistema: Trabajar con WinCC, SIEMENS
- Manual de sistema: Scripting(VBS, ANSI-C, VBA), SIEMENS
- Manual de sistema: Configuraciones, SIEMENS
- Manual de sistema: Comunicación, SIEMENS
- Manual de sistema: Basic Options, SIEMENS
- System manual: Options for Process Control, SIEMENS
- Limitaciones de rendimiento WinCC, <https://support.industry.siemens.com/cs/mdm/109736216?c=84024098699&lc=es-WW>

## 2. ANEXOS

### 2.1 ANEXO I - Manual del programador

#### 2.1.1. Versión de WinCC

Para realizar el desarrollo del SCADA objeto de este trabajo se emplea WinCC v7.4+SP1+Upd10



En concreto dentro de las herramientas de SIMATIC WinCC se emplea el WinCC Explorer para la creación del proyecto.

#### 2.1.2. Consideraciones de rendimiento desarrollo SCADA en WinCC

En este trabajo se ha procurado realizar un desarrollo modular de forma que se pueda reaprovechar el código en la medida de lo posible para acelerar el tiempo de desarrollo y poder aprovechar el código dentro del mismo y en futuros proyectos, pero debido a la amplitud y carga de objetos en algunas de las pantallas que se han desarrollado deben tenerse en cuenta las siguientes consideraciones para conseguir un rendimiento alto en la ejecución y manejo del SCADA.

Las consideraciones que se explican a continuación surgen de la multitud de pruebas que se han realizado durante la resolución de los problemas que han surgido en la puesta en marcha y desarrollo del programa.

##### 2.1.2.1. Empleo óptimo de faceplates

Los faceplates son un tipo de objeto de imagen estandarizado que permite que se utilicen varias veces en el proyecto, lo que permite la reducción del trabajo de configuración al crear y modificar objetos.

Este objeto permite su dinamización creando variables internas y accediendo a ellas con el componente "SmartTags" en el código en VBS, de esta forma se consigue aprovechar el mismo

código para cada una de las instancias desde la cual se asignan las variables para las que se desea realizar el cálculo de la dinamización.

Se ha comprobado que si se instancian más de 31 faceplates con scripts internos en una imagen de proceso, el rendimiento del SCADA disminuye considerablemente introduciendo retrasos de varios segundos, según la cantidad de instancias de faceplates, hasta la completa actualización del estado de los objetos de la pantalla.

El rendimiento al abrir un sinóptico de proceso en runtime está afectado por los siguientes factores:

- Número de instancias de faceplate en el sinóptico de proceso.
- Utilización de scripts internos en los faceplates y sus instancias.
- El número de variables internas del faceplate.

Teniendo en cuenta estos factores, en este desarrollo se han desarrollado los sinópticos de proceso empleando faceplates para facilitar la reconfiguración de los objetos si fuese necesario, pero su dinamización se ha programado desde la instancia del mismo en vez de emplear scripts y variables en el faceplate.

### 2.1.2.2. Dinamización de objetos

La dinamización de los objetos del sinóptico de proceso supone la creación de la lógica de decisión para determinar el estado de uno o varios de sus atributos, como puede ser el color de un motor, el parpadeo del mismo si está en alarma, el texto indicativo del valor analógico de una sonda, etc...

Para generar la lógica que determina el valor o estado de un atributo puede realizarse mediante los siguientes métodos:

- Conexión directa
- Diálogo de dinamización
- Acciones en C
- Acciones en VBS

Atendiendo a su rendimiento en Runtime se tienen en cuenta las siguientes consideraciones para cada uno de los métodos para elegir cuál de ellos utilizar en las dinamizaciones de los objetos instanciados en los sinópticos de proceso.

- **Mediante conexión directa:** Ofrece el mayor rendimiento en tiempo de ejecución y la dinámica más rápida en la imagen entre todos los métodos de dinamización pero en contrapartida solo puede crearse una conexión por propiedad. Sus ventajas son la sencillez de configuración y la rapidez en Runtime.

Atributo	Estático	Dinamización	Actualiz.	Indir.
VALOR_REAL	0,000000e+000	SP_M1_10_1.ValorReal	Al cambiar	<input type="checkbox"/>
UNIDAD	Und	SP_M1_		<input type="checkbox"/>
ESTADO	0	SP_M1_		<input type="checkbox"/>

- Diálogo de dinamización...
- Acción C...
- Acción VBS...
- Variable...
- Borrar

- **Mediante el diálogo de dinamización:** Tiene un elevado rendimiento en tiempo de ejecución aunque este desciende en cuanto se emplean varias variables u operadores.



Internamente el diálogo de dinamización genera una acción en C que si se desea puede ampliarse posteriormente perdiendo la ventaja del alto rendimiento de este método.

Atributo	Estático	Dinamización	Actualiz.	Indir.
COLOR_NOMBRE				
UNIDAD_SA	Und			<input type="checkbox"/>
VALOR_REAL_SA	0,000000e+000			<input type="checkbox"/>
COLOR_FONDO_CO				<input type="checkbox"/>
COLOR_FONDO_SA				<input type="checkbox"/>
ENCLIB_Visible	No			<input type="checkbox"/>
VOLUM_Visible	No			<input type="checkbox"/>

Este método ofrece un diálogo de configuración muy intuitivo desde el que se pueden realizar las operaciones que se consideren necesarias.

The dialog box 'Rango de valores' contains the following fields and options:

- Idioma utilizado: Español (España, internacional)
- Nombre del evento: Variable
- Expresión / Fórmula: 'REG\_AV1.Info\_Auto'
- Resultado de la expresión/fórmula: Preview showing 'Sí / TRUE' in green and 'No / FALSE' in red.
- Tipo de datos: Bool (selected), Analóg., Bit, Direct.
- Options: No evaluar estado de variables (selected), Evaluar estado de variables, Evaluar Quality Code.
- Buttons: Verificar, Agregar, Quitar, Aceptar, Cancelar.

Para emplear el diálogo de dinamización no es necesario tener amplios conocimientos de programación ya que solo debe de realizarse la expresión y formula correctamente y asignar los valores que se deseen al resultado de esta.

- **Mediante acciones en C:** el lenguaje de programación ANSI-C permite realizar dinamizaciones complejas y se recomienda su empleo cuando las posibilidades de conexión directa o diálogo de dinamización no son suficientes ya que es el siguiente método con mejor rendimiento, aunque dependerá también de la amplitud y complejidad de las operaciones a realizar en el código que se genere.

- **Mediante acciones en VBS:** el lenguaje de scripting visual basic script VBS ofrece la posibilidad de realizar dinimizaciones complejas cuando los métodos de conexión directa y diálogo de dinamización no sean suficientes. Es el método dinamización con peor rendimiento aunque este también dependerá de la amplitud y complejidad de las operaciones que presente el código, además de la correcta selección de los disparadores.

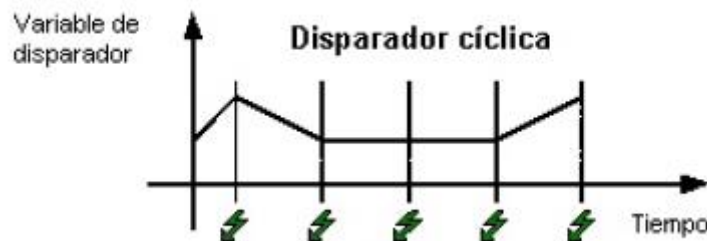
### 2.1.2.3. Configuración de los disparadores

Los disparadores o “Triggers” son necesarios para decidir cuándo se ejecutan las acciones que determinan el valor de los atributos en tiempo de ejecución.

La correcta configuración de los disparadores en cada uno de los casos de uso afectará notablemente al rendimiento del SCADA.

Se disponen de los siguientes disparadores para la dinamización de objetos:

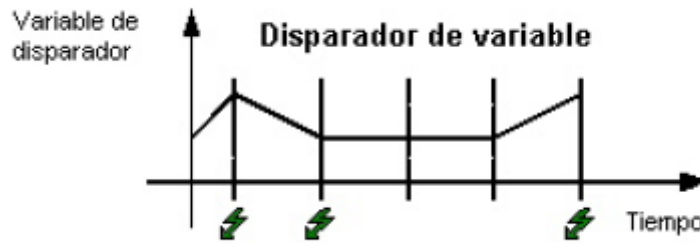
- **Disparadores controlados por eventos:** son aquellos que disparan la acción vinculada a un evento como pueden ser: cambios en alguna propiedad del objeto, clic con el ratón, operaciones con el teclado, modificación del foco, etc...  
Este tipo de disparadores no suponen una carga alta para el sistema ya que ocurren eventualmente.
- **Disparadores cíclicos:** los disparadores cíclicos son aquellos capaces de ejecutar acciones periódicamente, por ejemplo cada 10 segundos.



Empleando este tipo de disparadores puede conseguirse una mayor reutilización del código ya que el evento que dispara la acción no requiere de una configuración específica de los disparadores para cada objeto, pero por contrapartida supone una carga alta para el sistema si la pantalla está compuesta por gran cantidad de objetos ya que se realizarán consultas a las variables independientemente de que el valor de estas haya cambiado.

No se recomienda su uso en sinópticos de proceso con grandes cantidades de objetos porque supone una alta carga para el sistema y mermar el rendimiento en Runtime.

- **Disparadores de variables:** los disparadores de variables llevarán a cabo una acción solo cuando se detecte un cambio en el valor de las variables que se configuren, por lo tanto se consigue un alto rendimiento en Runtime reduciendo la carga del sistema.



Con este método también se consigue optimizar el tráfico de las comunicaciones ya si las variables del disparador no cambian no se consulta su estado desde el código de la acción cíclica.

Con el fin de conseguir un elevado rendimiento en Runtime se recomienda el uso de los disparadores de variables y eventos ante los disparadores cíclicos.

#### 2.1.2.4. Inicialización de variables

Cuando se declaran las variables es recomendable su inicialización porque de esta forma en el primer ciclo de lectura de una variable de proceso se retorna su valor inicial aunque ya exista un valor de proceso disponible.

Variables [ FIFO4_SEC ]		
	Nombre	Valor inicial
1	via4_1.ciclo	0
2	via4_1.estado	0
3	via4_1.estanteria	0
4	via4_1.producto	0

Al asignarle a una variable un valor inicial se consigue acelerar el despliegue de imágenes en Runtime y al realizar un cambio de imagen.

#### 2.1.2.5. Ocultar/mostrar valor de variable en Runtime

En tiempo de ejecución se actualizan cíclicamente los valores de las variables y mostrar continuamente el valor de estas variables puede suponer una gran carga para el sistema si la cantidad de variables es grande.

Variables [ FIFO_SEC_VIA2_F2 ]			
	Nombre	Valor	Quality Code
1	f2via2_1.ciclo	0	Orden ascendente
2	f2via2_1.estado	0	Orden descendente
3	f2via2_1.estanteria	0	Quitar orden
4	f2via2_1.producto	0	Filtrar
5	f2via2_2.ciclo	0	Cortar
6	f2via2_2.estado	0	Copiar
7	f2via2_2.estanteria	86	Pegar
8	f2via2_2.producto	98	Buscar y reemplazar
9	f2via2_3.ciclo	0	Borrar
10	f2via2_3.estado	0	Ocultar
11	f2via2_3.estanteria	39	
12	f2via2_3.producto	98	
13	f2via2_4.ciclo	0	
14	f2via2_4.estado	0	
15	f2via2_4.estanteria	37	

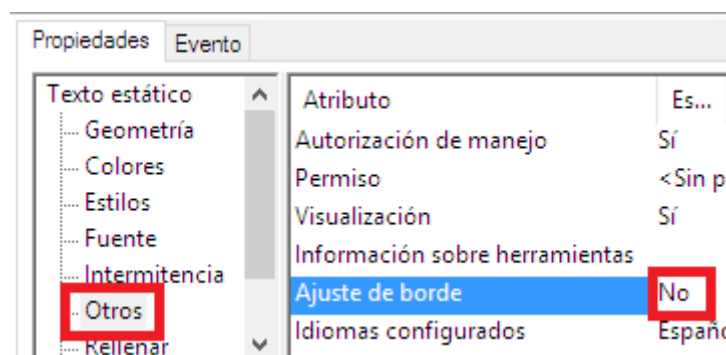
La visualización del valor de las variables puede configurarse desde el área de las variables y se recomienda ocultarlo seleccionando la columna "Valor" y haciendo clic en Ocultar.

Si es necesario volver a ver el valor en Runtime por motivos de depuración puede mostrarse de nuevo haciendo clic derecho sobre cualquier columna del área de variables y seleccionando "Valor" dentro del desplegable "Mostrar".

### 2.1.2.6. Uso del atributo "Adaptar borde"

Con el atributo "Adaptar borde" de los campos de texto puede configurarse que el campo de texto adapte sus dimensiones al tamaño del texto que contiene de forma dinámica.

Si este atributo está configurado para que se adapte de forma dinámica en una gran cantidad de campos de texto dentro de una misma imagen puede suponer una carga alta en tiempo de ejecución y que el sistema pierda rendimiento.



Es por ello que a no ser que la aplicación lo requiera en ciertos casos específicos, se recomienda tener este parámetro configurado a "No" de forma general para que las dimensiones de los campos de texto sean estáticas y no suponga un coste computacional para el motor de ejecución de WinCC.

### 2.1.2.7. Optimizar lectura y/o escritura de variables en Scripting VBS

En ocasiones es necesario el empleo de scripts en lenguaje VBS cuando los métodos de dinamización que presentan mejor rendimiento, descritos anteriormente, no son suficientes para realizar la dinamización de los atributos del objeto. Estos casos suelen surgir cuando es necesario leer y/o escribir gran cantidad de variables en un único script y el tiempo de lectura y/o escritura de las variables puede suponer una sobrecarga para el sistema si el número de variables es elevado.

Esto puede optimizarse empleando el objeto de tipo "TagSet" para realizar una lectura conjunta de variables como se muestra en el siguiente código de ejemplo.

```
Dim group, i

Set group = HMIRuntime.Tags.CreateTagSet 'Se crea un grupo de variables

For i=0 To 6 'Se añaden las variables de alarmas al grupo
    group.Add Item.ObjectName&"."&"alar"&i
Next

group.Read 'Se leen las variables conjuntamente

'Se accede al valor leído en cualquier punto del programa
For i=0 To 6
    If group(Item.ObjectName&"."&"alar"&i).Value = 1 Then
        Item.BackFlashColorOn = RGB(252,255,0) 'amarillo alarma
        Item.BackFlashColorOff = RGB(218,218,218) 'gris fondo
        Item.FlashRateBackColor = 2 'Intermitencia rápida
        Item.FlashBackColor = 1 'Activa intermitencia
    End If
Next
```

De esta forma se pueden procesar varias órdenes de lectura en una misma ejecución del script, consiguiendo mejorar el rendimiento en comparación con varias lecturas individuales por ejecución.

### 2.1.2.8. Lectura o escritura directa de variables

En ocasiones cuando es necesario representar en el SCADA secciones críticas del proceso, se necesita leer directamente el valor real del PLC en vez de obtener el valor de la imagen de variable que se actualiza cíclicamente y no nos asegura que la lectura corresponda con ese mismo instante.

En VBS se realiza una lectura directa de la siguiente forma:

```
Dim valor, objTag
Set objTag = HMIRuntime.Tags("Variable1")

valor = objTag.Read(1) 'Lectura directa de valor de "Variable1" del PLC
```

En C se realiza una lectura directa de la siguiente forma:

```
int valor;

valor = GetTagBitWait("Variable1");//Retorna el valor directamente del PLC
```

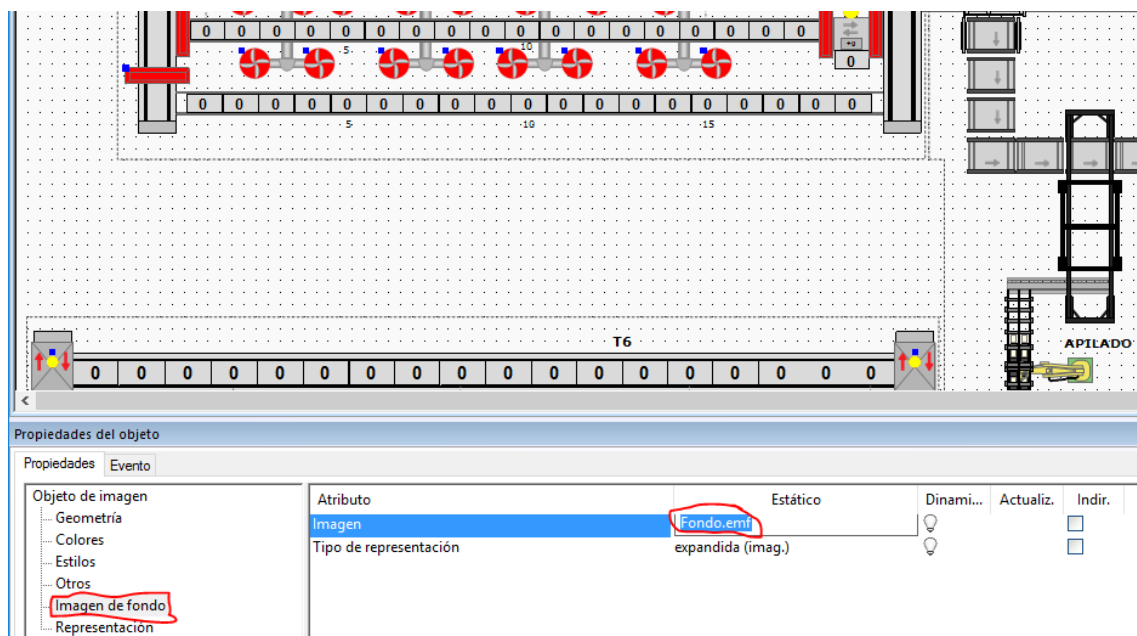
Este tipo de llamadas para leer y/o escribir directamente del PLC duran más tiempo y son más costosas para el sistema que el empleo de las llamadas estándar, aunque ello también depende del canal y el PLC.

De igual forma ocurre con “SetTagWait” en C y “.Write(1)” en VBS para el caso de la escritura.

Su empleo está orientado al uso a eventos de lectura/escritura para sincronizar procesos rápidos y deberán evitarse en acciones cíclicas en la medida de lo posible ya que supone un coste computacional muy elevado para el sistema y causa problemas de rendimiento.

### 2.1.2.9. Fondo de imagen estático como archivo EMF

En las imágenes de proceso si se desea cambiar el fondo estándar la forma más eficiente de gestionarlo para el sistema es insertando una imagen en formato .emf en el propiedad “Imagen de fondo” atributo “Imagen”.



Para crear el archivo .emf con el color de fondo se puede hacer desde Graphics Designer en Archivo > Nuevo (CTRL+N) y se le configura el color de fondo deseado.

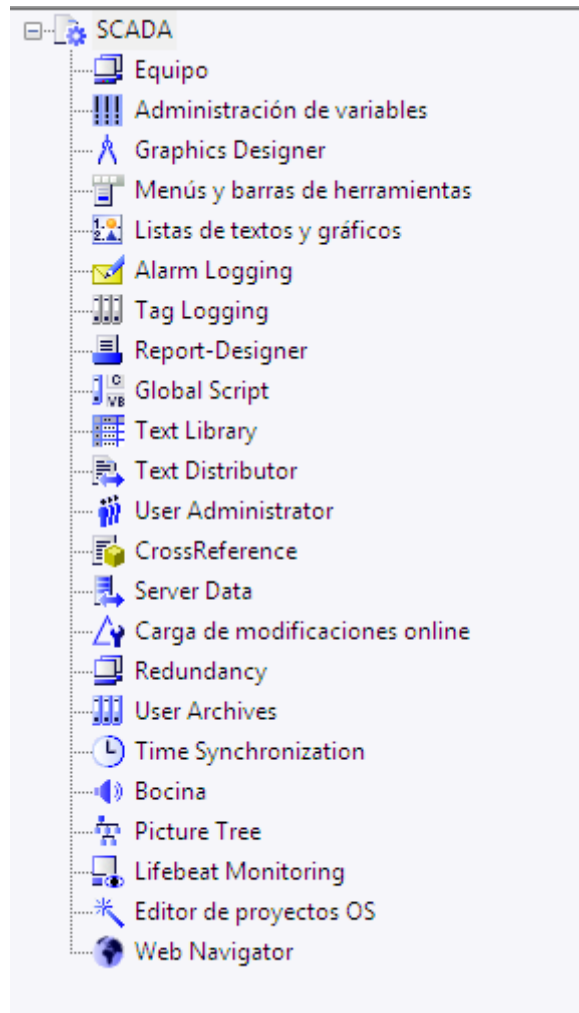
Después en Archivo > Exportar, se guarda como “Fondo.emf” y ya puede asignarse donde se requiera.

### 2.1.3. Creación del Proyecto con WinCC Explorer

Desde el software WinCC Explorer (instalado en la estación de ingeniería) se gestiona la creación y programación del proyecto.

### 2.1.3.1. Árbol del proyecto

Cuando se crea un proyecto en WinCC aparece un árbol con los distintos módulos de programa que pueden usarse para el desarrollo del mismo.



A continuación se explican las funcionalidades de cada uno de los módulos que se han usado para desarrollar el trabajo.

- **Equipo:** Permite listar y configurar todos los equipos asignados al proyecto.
- **Administrador de variables:** Permite gestionar las variables y las comunicaciones con los PLCs empleados en el proyecto.
- **Graphics Designer:** Con esta herramienta se diseñan las pantallas que actuarán como interface de cara al usuario del SCADA y se programan los objetos que se usarán en su desarrollo.
- **Listas de textos y gráficos:** Permite crear listas de textos y gráficos para poder usarse después en visores de estado.
- **Alarm Logging:** El alarm logging se emplea para registrar y gestionar los avisos (alarmas).

- **Tag Logging:** Se emplea para gestionar el almacenamiento de los valores del proceso y posteriormente poder usar los archivos para realizar gráficas o consultar valores.
- **Global Script:** Permite programar funciones para poder emplearlas en cualquier punto del proyecto y scripts que se ejecutarán de forma global.
- **User Administrator:** se emplea para crear y administrar los usuarios y permisos del proyecto.
- **Redundancy:** permite activar la redundancia y configurar la sincronización de archivos entre los servidores
- **Web Navigator:** permite procesar el SCADA desarrollado y configurarlo para que pueda ser manejado desde internet o una red intranet sin la necesidad de realizar cambios en el proyecto.

### 2.1.3.2. PLCs del proyecto

El proyecto consta de los siguientes PLCs configurados en el administrador de variables con el driver Modbus TCP/IP:

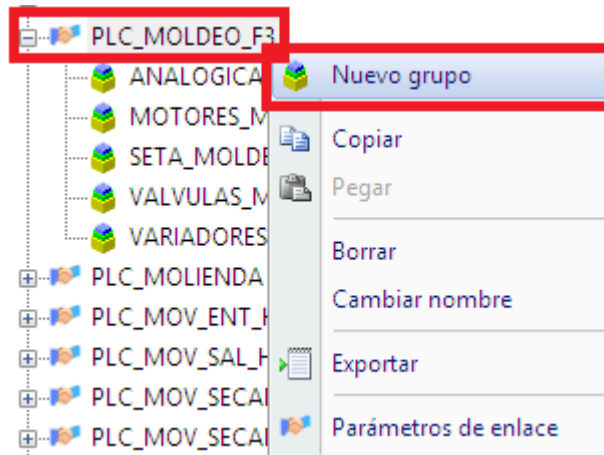
PLC	Dirección IP
PLC_ALMACÉN_BANDEJAS	192.168.3.6
PLC_APILADO_F3	192.168.3.5
PLC_CARGADOR_SECADERO_F3	192.168.3.2
PLC_CARRO_ALIMENTACIÓN	192.168.4.34
PLC_COMUNICACIONES	192.168.4.36
PLC_CONTADOR_GAS	192.168.0.31
PLC_DESAPILADO_F3	192.168.3.9
PLC_DESCARGADOR_SECADERO_F3	192.168.3.4
PLC_DISTRIBUCIÓN_FÁBRICAS	192.168.4.33
PLC_DRAGA_EXTRACCIÓN	192.168.4.35
PLC_EMPAQUETADO_F3	192.168.3.17
PLC_ETIQUETADO_F3	192.168.3.35
PLC_FLEJADORAS	192.168.3.33
PLC_HORNO_F3	192.168.3.12
PLC_MOLDEO_F3	192.168.3.1
PLC_MOLIENDA	192.168.4.31
PLC_MOV_ENT_HORNO	192.168.3.10
PLC_MOV_SAL_HORNO	192.168.3.3
PLC_MOV_SECADERO_F3	192.168.3.7
PLC_MÓVIL	192.168.3.37
PLC_PUPITRE	192.168.3.36
PLC_RECIRCULADORES_F3	192.168.3.11
PLC_SALA_TÉRMICA	192.168.3.8
PLC_SALIDA_TIERRAS	192.168.4.32
PLC_TRAZABILIDAD	192.168.3.50
PLC_VISIÓN	192.168.3.28



### 2.1.3.3. Gestión de las variables

En WinCC pueden crearse variables empleando dos tipos de objetos:

- **Grupos de variables:** permiten organizar las variables internas en grupos en la conexión de cada PLC.



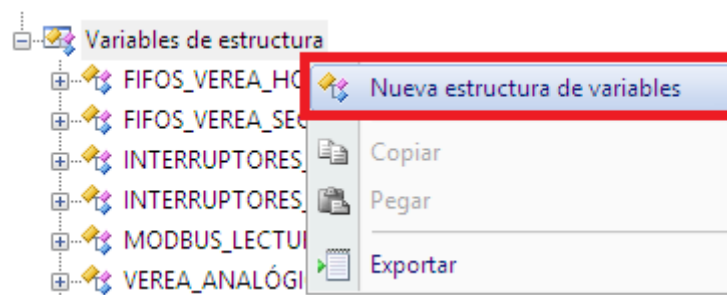
Para crear un nuevo grupo de variables en un PLC se hace clic derecho encima de su conexión y se selecciona “Nuevo grupo”, posteriormente se le debe dar un nombre que sea unívoco en todo el proyecto.

Dentro de los grupos pueden declararse variables de forma independiente o empleando estructuras de variables.

- **Estructuras de variables:** las estructuras de variables permiten la creación de variables internas o de proceso de forma simultánea como elementos de variables de forma que obtiene una estructura similar para cada variable de estructura.

Esta forma de crear variables permite crear una estructura para cada tipo de objeto, de forma que se acelera el proceso de creación de las mismas y se consigue que el código que se desarrolle en la programación para dinamización de atributos y diseño de las pantallas de mando sea reciclable para objetos del mismo tipo.

Para crear una nueva estructura de variables se debe hacer clic derecho encima de Variables de estructura desde el Administrador de variables y se selecciona “Nueva estructura de variables” a la que se le asignará un nombre unívoco en todo el proyecto.



Automáticamente dentro de cada estructura que se cree aparecerán las siguientes pestañas en la parte inferior del Administrador de variables que nos permitirá la configuración de la estructura:

- **Elementos de estructura:** hacen referencia a aquellas propiedades que pueda tener cada una de las variables de estructura. Pueden definirse como variable de proceso (si se marcan como Externo) o variables internas.

*Por ejemplo en caso de un motor: alarma1, alarma2, Marcha, Paro, Manual, Automatico, Pantalla\_mando, etc...*

- **Variables de estructura:** las variables de estructura son aquellos objetos del mismo tipo que poseen los mismos elementos de estructura que definen sus propiedades. Para cada una de las variables de estructura puede configurarse con que PLC está conectado para direccionar la variable de proceso y a que grupo de variables pertenece dentro de cada PLC.

*Por ejemplo en caso de motores podrían definirse: MOTOR1, MOTOR2, etc...*

- **Elementos de variables de estructura:** es el resultado de la concatenación de las variables de estructura con un punto y los elementos de estructura y define el conjunto de variables que se genera para cada una de los objetos de mismo tipo.

*Por ejemplo para el caso de los motores dado se obtendría:*

*MOTOR1.alarma1*

*MOTOR1.alarma2*

*MOTOR1.Marcha*

*MOTOR1.Paro*

*MOTOR1.Manual*

*MOTOR1.Automatico*

*MOTOR1.Pantalla\_mando*

*MOTOR2.alarma1*

*MOTOR2.alarma2*

*MOTOR2.Marcha*

*MOTOR2.Paro*

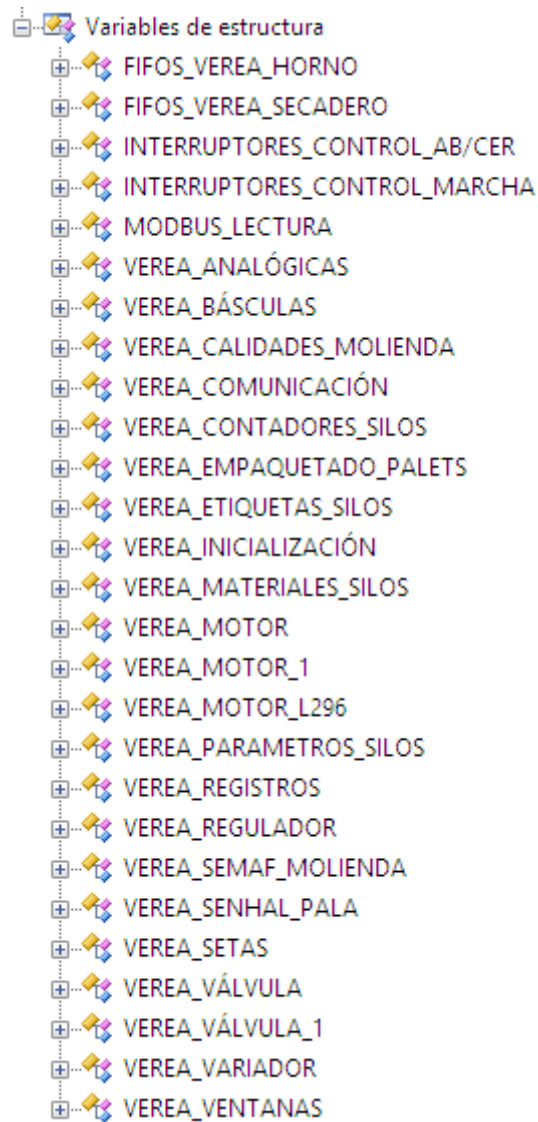
*MOTOR2.Manual*

*MOTOR2.Automatico*

*MOTOR2.Pantalla\_mando*

Desde los elementos de variables de estructura se le asignan los valores iniciales a las variables tanto internas como de proceso y se direccionan las variables de proceso.

En este proyecto se ha tratado de agrupar la mayor cantidad posible de variables creándolas como estructuras de variables para poder reaprovechar pantallas de mando y el código de dinamización de atributos de objetos. Para ello se han creado las siguientes estructuras de variables:



## 2.1.4. Desarrollo de la programación

En esta sección del anexo se detalla la programación desarrollada en el proyecto

### 2.1.4.1. Desarrollo de librerías de funciones

Una de las formas de reciclar el código es el empleo de funciones que pueden ser llamadas en cualquier punto del proyecto: en eventos, acciones cíclicas, scripts globales, así como también dentro de otras funciones.

Desde el Global Script se han desarrollado las siguientes funciones de proyecto que se detallan a continuación.

## Librería de funciones en C

### Función *cursorCoord*:

```
void cursorCoord(int* posX, int* posY) //Devuelve las coordenadas del
cursor
{
typedef struct {
    LONG x;
    LONG y;
} tPOINT;

#pragma code ("user32.dll"); //Usa la libreria de la API de Windows
    BOOL GetCursorPos(tPOINT* lpPoint);
#pragma code ();

tPOINT pt;
GetCursorPos(&pt);

//printf("CoordX: %i, CoordY: %i\n", pt.x, pt.y);

*posx = pt.x;
*posy =pt.y;
}
```

En WinCC pueden obtenerse la coordenadas de un objeto respecto a su instancia en el sinóptico de proceso empleando las funciones estándar que se incluyen por defecto en la librería de WinCC, pero si se usa el zoom y se desplaza la imagen puede que se de en muchas ocasiones que la posición del objeto que se muestra en pantalla no coincida con su posición real en pantalla, es por ello que se desarrolla esta función que emplea la librería del sistema operativo "user32.dll" para obtener las cordenadas del puntero del ratón.

WinCC ofrece la posibilidad de emplear DLLs (Dynamic Link Libraries) propias haciendo uso de "#pragma code" para usar funciones externas al compilador de C de WinCC.

La función *cursorCoord()* devuelve las coordenadas del puntero en sus parámetros, que serán usadas para realizar los cálculos para determinar el punto de apertura de las pantallas de mando de los objetos y lograr que estas no tapen los objetos a los que hacen referencia.

### Función *devuelveCoord*:

```
void devuelveCoord(int* posX, int* posY) //Devuelve las coordenadas de
apertura para el faceplate
{
int Xpos, Ypos; //posiciones de apertura de ventana
//char* cuadr= (char*)SysMalloc(80);

//Calcula las coordenadas del cursor
typedef struct {
LONG x;
LONG y;
} tPOINT;

#pragma code ("user32.dll"); //Usa la libreria de la API de Windows
BOOL GetCursorPos(tPOINT* lpPoint);
#pragma code ();
tPOINT pt;
GetCursorPos(&pt);
}
```

```

//Se mapea el area de la pantalla para calcular el punto de apertura
de una ventana o mando
//Colocación Ventana Base

/* Se divide la pantalla en 12 regiones (Resolucion 1920*1080)

      C1  C2  C3  C4      C5  C6  C7  C8
-----
F1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
----- 360
F2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
----- 720
F3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
-----
      480  960  1440  1920  2400  2880  3360  3840

      MONITOR 1      MONITOR 2
*/

//Columna1
if((pt.x<=480)&&(pt.y<=360)){ //C1_F1
    //strcpy(cuadr,"C1_F1");
    Xpos = pt.x+30;
    Ypos = pt.y+30;
}
if((pt.x<=480)&&(pt.y>360)&&(pt.y<=720)){ //C1_F2
    //strcpy(cuadr,"C1_F2");
    Xpos = pt.x+70;
    Ypos = pt.y-350;
}
if((pt.x<=480)&&(pt.y>720)){ //C1_F3
    //strcpy(cuadr,"C1_F3");
    Xpos = pt.x+70;
    Ypos = pt.y-600;
}

//Columna2
if((pt.x>480)&&(pt.x<=960)&&(pt.y<=360)){ //C2_F1
    //strcpy(cuadr,"C2_F1");
    Xpos = pt.x+50;
    Ypos = pt.y;
}
if((pt.x>480)&&(pt.x<=960)&&(pt.y>360)&&(pt.y<=720)){ //C2_F2
    //strcpy(cuadr,"C2_F2");
    Xpos = pt.x+70;
    Ypos = pt.y-250;
}
if((pt.x>480)&&(pt.x<=960)&&(pt.y>720)){ //C2_F3
    //strcpy(cuadr,"C2_F3");
    Xpos = pt.x+70;
    Ypos = pt.y-600;
}

//Columna3
if((pt.x>960)&&(pt.x<=1440)&&(pt.y<=360)){ //C3_F1
    //strcpy(cuadr,"C3_F1");
    Xpos = pt.x-800;

```

```

        Ypos = pt.y+30;
    }
    if ((pt.x>960) && (pt.x<=1440) && (pt.y>360) && (pt.y<=720)) { //C3_F2
        //strcpy(cuadr, "C3_F2");
        Xpos = pt.x-800;
        Ypos = pt.y-350;
    }
    if ((pt.x>960) && (pt.x<=1440) && (pt.y>720)) { //C3_F3
        //strcpy(cuadr, "C3_F3");
        Xpos = pt.x-800;
        Ypos = pt.y-600;
    }

    //Columna4
    if ((pt.x>1440) && (pt.x<=1920) && (pt.y<=360)) { //C4_F1
        //strcpy(cuadr, "C4_F1");
        Xpos = pt.x-750;
        Ypos = pt.y+30;
    }
    if ((pt.x>1440) && (pt.x<=1920) && (pt.y>360) && (pt.y<=720)) { //C4_F2
        //strcpy(cuadr, "C4_F2");
        Xpos = pt.x-750;
        Ypos = pt.y-350;
    }
    if ((pt.x>1440) && (pt.x<=1920) && (pt.y>720)) { //C4_F3
        //strcpy(cuadr, "C4_F3");
        Xpos = pt.x-750;
        Ypos = pt.y-590;
    }

    //Columna5
    if ((pt.x>1920) && (pt.x<=2400) && (pt.y<=360)) { //C4_F1
        //strcpy(cuadr, "C5_F1");
        Xpos = pt.x-1920+30;
        Ypos = pt.y+30;
    }
    if ((pt.x>1920) && (pt.x<=2400) && (pt.y>360) && (pt.y<=720)) { //C4_F2
        //strcpy(cuadr, "C5_F2");
        Xpos = pt.x-1920+70;
        Ypos = pt.y-350;
    }
    if ((pt.x>1920) && (pt.x<=2400) && (pt.y>720)) { //C4_F3
        //strcpy(cuadr, "C5_F3");
        Xpos = pt.x-1920+70;
        Ypos = pt.y-600;
    }

    //Columna6
    if ((pt.x>2400) && (pt.x<=2880) && (pt.y<=360)) { //C2_F1
        //strcpy(cuadr, "C6_F1");
        Xpos = pt.x-1920+50;
        Ypos = pt.y;
    }
    if ((pt.x>2400) && (pt.x<=2880) && (pt.y>360) && (pt.y<=720)) { //C2_F2
        //strcpy(cuadr, "C6_F2");
        Xpos = pt.x-1920+70;
        Ypos = pt.y-250;
    }
    if ((pt.x>2400) && (pt.x<=2880) && (pt.y>720)) { //C2_F3
        //strcpy(cuadr, "C6_F3");
        Xpos = pt.x-1920+70;
    }

```

```

        Ypos = pt.y-600;
    }

    //Columna7
    if((pt.x>2880) && (pt.x<=3360) && (pt.y<=360)) { //C3_F1
        //strcpy(cuadr, "C7_F1");
        Xpos = pt.x-1920-800;
        Ypos = pt.y+30;
    }
    if((pt.x>2880) && (pt.x<=3360) && (pt.y>360) && (pt.y<=720)) { //C3_F2
        //strcpy(cuadr, "C7_F2");
        Xpos = pt.x-1920-800;
        Ypos = pt.y-350;
    }
    if((pt.x>2880) && (pt.x<=3360) && (pt.y>720)) { //C3_F3
        //strcpy(cuadr, "C7_F3");
        Xpos = pt.x-1920-800;
        Ypos = pt.y-600;
    }

    //Columna8
    if((pt.x>3360) && (pt.x<=3840) && (pt.y<=360)) { //C4_F1
        //strcpy(cuadr, "C8_F1");
        Xpos = pt.x-1920-750;
        Ypos = pt.y+30;
    }
    if((pt.x>3360) && (pt.x<=3840) && (pt.y>360) && (pt.y<=720)) { //C4_F2
        //strcpy(cuadr, "C8_F2");
        Xpos = pt.x-1920-750;
        Ypos = pt.y-350;
    }
    if((pt.x>3360) && (pt.x<=3840) && (pt.y>720)) { //C4_F3
        //strcpy(cuadr, "C8_F3");
        Xpos = pt.x-1920-750;
        Ypos = pt.y-590;
    }

    //Debug
    //printf("x=%d,y=%d\n",pt.x,pt.y); //Coordenadas puntero
    //printf("apX=%d,apY=%d\n",Xpos,Ypos); //Punto de apertura ventana
    //printf("Cuadrante: %s\n", cuadr); //Cuadrante en el que se pulsa

    //Devuelve los valores de posición de apertura de ventana por
    referencia
    *posx= Xpos;
    *posy= Ypos;

    //Liberar memoria
    //SysFree(cuadr);
    }}

```

Con esta función se realiza un mapa de regiones de los 2 monitores del SCADA dividiendo cada uno de ellos en 12 áreas para las cuales se calcula el punto de apertura de las ventanas de mando. Esta función se emplea dentro de las funciones que gestionan la apertura de las ventanas de mando.

### Función devuelveTagTDS:

```
char * devuelveTagTDS(char sstr[], char cch)
{
    /***** FUNCIÓN devuelveTagsTDS *****/
    ver 0.11: 06/08/2018
    *****/

    //sstr[]: string en formato prefijoDELIMITADORtag
    //cch: char que contiene el delimitador

    char *sTagExtraido;
    char *sRetornado;
    sTagExtraido = SysMalloc ( sizeof( sstr ) );
    sRetornado = SysMalloc ( sizeof( sstr ) );

    if ( sstr[0] != NULL )
    {
        sTagExtraido = strchr(sstr, cch);

        if ( sTagExtraido != NULL )
        {
            //printf("String despues de %c es: %s\n", cch,
sTagExtraido);
            //printf("Se devuelve: %s \n", sTagExtraido+1);
            sRetornado = sTagExtraido+1; //retorno el caracter
siguiente al asterisco.
        } else
        {
            //printf("-->Funciones de Proyecto.\"devuelveTagsTDS\":
no se ha encontrado asterisco \n");
            sRetornado = &sstr[0] ;
        }
    } else
    {
        printf("-->Funciones de Proyecto.\"devuelveTagsTDS\": se ha
recibido un valor NULL. \n");
        sRetornado = NULL ;
    }

    return sRetornado ;
}
```

WinCC no permite realizar varias instancias de objetos con el mismo nombre dentro de la misma imagen de proceso, pero en ocasiones es necesario realizarlo, por ejemplo cuando se usan niveles con los mismos objetos o simplemente se requiere representar el mismo objeto en varios lugares dentro de la misma imagen. Es por ello que se desarrolla esta función que permite extraer el nombre de un objeto delimitado por un carácter y un asterisco (\*).

De esta forma se puede tener instanciado un mismo objeto dentro de la misma imagen de proceso con nombres distintos y que desde los dos se abra la misma ventana de mando. Por ejemplo, se puede tener instanciado dos objetos con nombres **p\*M1\_10** y **q\*M1\_10** respectivamente y al llamar a esta función se consigue que los dos hagan referencia al nombre de objeto **M1\_10** que corresponde con su declaración en las variables de estructura.

Esta función se emplea dentro de las funciones que gestionan la apertura de las ventanas de mando que se especifican a continuación.



### ***Función ventanaAnalogicasTDS:***

```
void ventanaAnalogicasTDS (char* lpszPictureName, char*
lpszObjectName)
{
    // #include "AP_PBIB.H"
    #include "apdefap.h"
    /***** 30/11/2018 *****/
    /*****/

    char* ventanaBase;
    char* ventanaImagen;

    char* var                = ( char * ) SysMalloc( 80 );
    char* tag_prefix         = ( char * ) SysMalloc( 80 );
    char* tag_descripcion    = ( char * ) SysMalloc( 80 );
    char* tag_pantalla       = ( char * ) SysMalloc( 80 );
    char* NombreTag          = ( char * ) SysMalloc( 120 );
    char* p;
    int monitor;

    const char ch = '*'; //delimitador
    DWORD dwState =0;

    int posX, posY;

    ventanaBase= GetParentPicture(lpszPictureName);
    ventanaImagen= GetParentPictureWindow(lpszPictureName);

    // En WebNavigator no se puede utilizar la propiedad
    WindowMonitorNumber de las ventanas de imagen porque siempre vale
    null, en vez de eso utilizamos el nombre de la ventana de imagen
    // buscamos en el nombre que contenga el numero de monitor (1 o 2)
    para saber en que monitor abrimos la ventana de faceplate

    p = strchr(ventanaImagen, '1');
    if (p) {
        monitor = 1;
    }
    else monitor = 2;

    //Concateno la ventana de imagen con el numero de monitor
    sprintf(ventanaImagen, "Analogicas%d", monitor);

    // Colocación Ventana Base
    //Gestion de la coordenada donde se abre la
    ventana_____

    //Obtiene el punto de apertura de la ventana de mando de forma dinámica
    devuelveCoord(&posx, &posy);
    //printf("CoordX: %d ,CoordY: %d\n", posX, posY);
    SetLeft(ventanaBase ,ventanaImagen, (monitor-1)*1920 + posX);
    SetTop(ventanaBase ,ventanaImagen, posY);

    //_____

    if ( devuelveTagTDS(lpszObjectName, ch) != NULL ){
        //printf ( "Tag que llega a la función \n\r: %s ",
        devuelveTagTDS( lpszObjectName, ch ) );
    }
}
```

```

        strcpy(NombreTag, devuelveTagTDS(lpszObjectName, ch));
        //printf ( "Nombre tag leido de la funcion:  \n\r %s ",
NombreTag );

        // A continuación se comprueba la existencia del tag. Como es
una estructura comprobamos uno de sus elementos, por ejemplo:
NombreTag.Nombre:
        strcpy( var, NombreTag );
        strcat( var, ".Nombre" );
        GetTagCharState( var, &dwState );

        if (dwState == 0 || dwState != 0x800 ){ // Se analiza: si no
hay error (0) o bien el error es diferente de 800h (sin conexión).

        //printf(" Tag existente.Valor DWState: %d \r\n",dwState);

        //Se concatena "." a continuación del lpszObjectName para
apuntar a la variable de la estructura ANALOGICAS
        strcpy(tag_prefix,NombreTag);
        strcat(tag_prefix, ".");

        //Se fija el Tag_Prefix de la ventana de Analógicas
SetPropChar(ventanaBase ,ventanaImagen, "TagPrefix", tag_prefix);

        //En la estructura la variable interna Descripcion contiene la
descripcion de la analógica para el CaptionText de la pantalla.
        strcpy( tag_descripcion, tag_prefix );
        strcat( tag_descripcion, "Descripcion" );
        SetPropChar ( ventanaBase ,ventanaImagen, "CaptionText",
GetTagChar( tag_descripcion ) );

        //En la estructura la variable interna Faceplate contiene el
nombre de la pantalla de faceplate que queremos usar.
        strcpy(tag_pantalla, tag_prefix );
        strcat(tag_pantalla, "Faceplate" );

        // Se obtiene el nombre de la imagen a mostrar
SetPropChar ( ventanaBase ,ventanaImagen, "PictureName",
GetTagChar( tag_pantalla ) );
        // Se visibiliza la Ventana
SetPropBOOL ( ventanaBase ,ventanaImagen, "Visible", 1 );

    }
    else{
        printf("-->%s . %s : Problema con el TAG: %s
\r\n", lpszPictureName, lpszObjectName, NombreTag);
        printf("Código de Error (hex): %X \r\n", dwState );
    }
}
else{
    printf("-->%s . %s : La función ha devuelto un valor NULL:
\r\n", lpszPictureName, lpszObjectName);
}

//Liberacion de memoria
SysFree( var );
SysFree( tag_prefix );
SysFree( tag_descripcion );
SysFree( tag_pantalla );
SysFree( NombreTag );
}

```

Esta función se usa en el evento "Hacer clic con el ratón" de las instancias de faceplates de las analógicas, para abrir la ventana de mando donde se visualiza la gráfica histórica de su valor.

Al igual que en el resto de funciones para abrir ventanas de mando, se emplea el nombre del objeto que corresponde con el declarado en las variables de estructura como TagPrefix de la ventana de imagen en la que se abrirá la imagen de proceso del mando. De esta forma se consigue reciclar la misma imagen para objetos con la misma estructura de variables.

**Función ventanaEsterasTDS:**

```
void ventanaEsterasTDS(char* lpszPictureName, char* lpszObjectName)
{
/***** 26/12/2018*****/
/*****/
#include "apdefap.h"

char* ventanaBase;
char* ventanaImagen;

char* tag_prefix= (char *)SysMalloc(80);
char* tag_pantalla= (char *)SysMalloc(80);
char *nombreTag= (char *)SysMalloc(120);

char *p;
int monitor;
const char ch= '*'; //delimitador
DWORD dwState= 0;

int posx, posy;

ventanaBase= GetParentPicture(lpszPictureName);
ventanaImagen= GetParentPictureWindow(lpszPictureName);

// monitor =
GetPropWord(ventanaBase ,ventanaImagen,"WindowMonitorNumber");
// En WebNavigator no se puede utilizar la propiedad WindowMonitorNumber de las ventanas de imagen porque siempre vale null, en vez de eso utilizamos el nombre de la ventana de imagen
//buscamos en el nombre que contenga el numero de moniro (1 o 2) para saber en que monitor abrimos la ventana de faceplace

p = strchr(ventanaImagen, '*');
if ( p ) {
monitor = 1;
}
else monitor = 2;

//Se concatena la ventana de imagen con el numero de monitor
sprintf(ventanaImagen,"Esteras_sec%d",monitor);

//Gestion de la coordenada donde se abre la
ventana_____

//Obtiene el punto de apertura de la ventana de mando de forma dinámica
devuelveCoord(&posx, &posy);
//printf("CoordX: %d ,CoordY: %d\n", posx, posy);
```

```

SetLeft(ventanaBase ,ventanaImagen, (monitor-1)*1920 + posx);
//Return-Type: BOOL
SetTop(ventanaBase ,ventanaImagen, posy); //Return-Type: BOOL

//
-----

if (devuelveTagTDS(lpszObjectName,ch) != NULL) //se comprueba que el
tag no sea nulo
{
    //printf("Tag que llega a la función \n\r: %s ", tempTag);
    strcpy(nombreTag, devuelveTagTDS(lpszObjectName,ch));
    //printf("Nombre tag leído de la función: \n\r %s ", nombreTag);

    //Cocatenamos "." a continuación del lpszObjectName para apuntar
a la variable de la estructura.
    strcpy(tag_prefix, nombreTag);
    strcat(tag_prefix, ".");
    SetPropChar
(ventanaBase ,ventanaImagen, "TagPrefix", tag_prefix);

    //En la estructura, la variable interna Faceplate contiene el
nombre de la pantalla de faceplate que queremos usar para este motor.
    strcpy(tag_pantalla, tag_prefix);
    strcat(tag_pantalla, "Faceplate");
    SetPropChar(ventanaBase ,ventanaImagen, "PictureName",
GetTagChar(tag_pantalla));
    SetPropBOOL(ventanaBase ,ventanaImagen, "Visible", 1 );
    //Tipo de retorno :BOOL
} else
{
    printf("-->%s . %s : La función ha devuelto un valor NULL: \r\n"
, lpszPictureName, lpszObjectName);
}

//se libera explícitamente la memoria de las zonas de memorias que se
reservan
SysFree(tag_prefix);
SysFree(tag_pantalla);
SysFree(nombreTag);
}

```

Esta función se usa en el evento "Hacer clic con el ratón" de las instancias de faceplates de las estanterías y vagonetas, para abrir la ventana de mando donde se visualizan las descripciones de los productos que contienen y los detalles correspondientes a cada una de ellas.

Al igual que en el resto de funciones para abrir ventanas de mando, se emplea el nombre del objeto que corresponde con el declarado en las variables de estructura como TagPrefix de la ventana de imagen en la que se abrirá la imagen de proceso del mando. De esta forma se consigue reciclar la misma imagen para objetos con la misma estructura de variables.

#### **Función ventanaMotoresTDS:**

```

void ventanaMotoresTDS(char* lpszPictureName, char* lpszObjectName)
{
/*****
**** 28/11/2018****
*****/

```

```

#include "apdefap.h"

char* ventanaBase;
char* ventanaImagen;
char* ventanaVariadores= (char *)SysMalloc(80);

char* tag_prefix= (char *)SysMalloc(80);
char* tag_descripcion= (char *)SysMalloc(80);
char* tag_pantalla= (char *)SysMalloc(80);
char* var= (char *)SysMalloc(80);
char *nombreTag= (char *)SysMalloc(120);

char *p;
int monitor;
const char ch= '*'; //delimitador
DWORD dwState= 0;

int posx, posy;

ventanaBase= GetParentPicture(lpszPictureName);
ventanaImagen= GetParentPictureWindow(lpszPictureName);

// monitor =
GetPropWord(ventanaBase ,ventanaImagen,"WindowMonitorNumber");
// En WebNavigator no se puede utilizar la propiedad
WindowMonitorNumber de las ventanas de imagen porque siempre vale
null, en vez de eso utilizamos el nombre de la ventana de imagen
//buscamos en el nombre que contenga el numero de moniro (1 o 2) para
saber en que monitor abrimos la ventana de faceplace

p = strchr(ventanaImagen, '1');
if ( p ) {
    monitor = 1;
}
else monitor = 2;

//Se concatena la ventana de imagen con el numero de monitor
sprintf(ventanaImagen, "VentanaFaceplate%d",monitor);

//Se concatena la ventana de imagen de los variadores con el numero de
monitor
sprintf(ventanaVariadores, "Variadores%d",monitor);

//Gestion de la coordenada donde se abre la
ventana_____

//Obtiene el punto de apertura de la ventana de mando de forma dinámica
devuelveCoord(&posx, &posy);
//printf("CoordX: %d ,CoordY: %d\n", posx, posy);

SetLeft(ventanaBase ,ventanaImagen, (monitor-1)*1920 + posx);
//Return-Type: BOOL
SetTop(ventanaBase ,ventanaImagen, posy); //Return-Type: BOOL

//_____

if (devuelveTagTDS(lpszObjectName, ch) != NULL ) //se comprueba que el
tag no sea nulo
{

```

```

//printf("Tag que llega a la función \n\r: %s ", tempTag);
strcpy(nombreTag, devuelveTagTDS(lpszObjectName, ch));
//printf("Nombre tag leído de la función: \n\r %s ", nombreTag);

// A continuación se comprueba la existencia del tag. Como es
una estructura comprobamos uno de sus elementos, por ejemplo:
nombreTag.QIntEstado:
strcpy          ( var, nombreTag );
strcat         ( var, ".QIntEstado" );
GetTagCharState ( var, &dwState );

if (dwState == 0 || dwState != 0x800 ) // Se analiza: si no
hay error (0) o bien el error es diferente de 800h (sin conexión).
{
    //Cocatenamos "." a continuación del lpszObjectName para
apuntar a la variable de la estructura MOTOR
    strcpy      (tag_prefix, nombreTag);
    strcat     (tag_prefix, ".");
    SetPropChar
(ventanaBase , ventanaImagen, "TagPrefix", tag_prefix);

    //En la estructura MOTOR, la variable interna Descripcion
contiene la descripcion del motor para el CaptionText de la pantalla.
    strcpy      (tag_descripcion, tag_prefix);
    strcat     (tag_descripcion, "Descripcion");
    SetPropChar (ventanaBase      , ventanaImagen, "CaptionText",
GetTagChar(tag_descripcion));

    //OFFSET IMAGEN y tamaño ajustable
    SetOffsetLeft (ventanaBase, ventanaImagen, 260);
    SetPropWord (ventanaBase, ventanaImagen, "Width", 350);
//Altura
    SetPropWord (ventanaBase, ventanaImagen, "Height", 440);
//Anchura

    //En la estructura MOTOR, la variable interna Faceplate
contiene el nombre de la pantalla de faceplate que queremos usar para
este motor.
    strcpy      (tag_pantalla, tag_prefix );
    strcat     (tag_pantalla, "Faceplate" );
    SetPropBOOL (ventanaBase      , ventanaVariadores, "Visible", 0);
//Se cierra la ventana de variadores
    SetPropChar (ventanaBase      , ventanaImagen, "PictureName",
GetTagChar(tag_pantalla)); //Se selecciona la imagen del mando
    SetPropBOOL (ventanaBase      , ventanaImagen, "Visible", 1 );
    //Tipo de retorno :BOOL
}
else
{
    printf("-->%s . %s : Problema con el TAG: %s \r\n",
lpszPictureName, lpszObjectName, nombreTag);
    printf("Código de Error (hex): %X \r\n", dwState);
}
} else
{
    printf("-->%s . %s : La función ha devuelto un valor NULL: \r\n",
lpszPictureName, lpszObjectName);
}

//se libera explícitamente la memoria de las zonas de memorias que se
reservan

```

```

SysFree(ventanaVariadores);
SysFree(tag_prefix);
SysFree(tag_descripcion);
SysFree(tag_pantalla);
SysFree(var);
SysFree(nombreTag);
}

```

Esta función se usa en el evento “Hacer clic con el ratón” de las instancias de faceplates de los motores, para abrir la ventana de mando donde se visualizan las informaciones de estado del motor, las alarmas y los botones de mando si es que lo presenta.

Al igual que en el resto de funciones para abrir ventanas de mando, se emplea el nombre del objeto que corresponde con el declarado en las variables de estructura como TagPrefix de la ventana de imagen en la que se abrirá la imagen de proceso del mando. De esta forma se consigue reciclar la misma imagen para objetos con la misma estructura de variables.

#### *Función ventanaPuertasTDS:*

```

void ventanaPuertasTDS(char* lpszPictureName, char* lpszObjectName)
{
/*****
**** 07/12/2018****
*****/
#include "apdefap.h"

char* ventanaBase;
char* ventanaImagen;

char* tag_prefix= (char *)SysMalloc(80);
char* tag_descripcion= (char *)SysMalloc(80);
char* tag_pantalla= (char *)SysMalloc(80);
char* var= (char *)SysMalloc(80);
char *nombreTag= (char *)SysMalloc(120);

char *p;
int monitor;
const char ch= '*'; //delimitador
DWORD dwState= 0;

int posx, posy;

ventanaBase= GetParentPicture(lpszPictureName);
ventanaImagen= GetParentPictureWindow(lpszPictureName);

// monitor =
GetPropWord(ventanaBase ,ventanaImagen,"WindowMonitorNumber");
// En WebNavigator no se puede utilizar la propiedad
WindowMonitorNumber de las ventanas de imagen porque siempre vale
null, en vez de eso utilizamos el nombre de la ventana de imagen
//buscamos en el nombre que contenga el numero de moniro (1 o 2) para
saber en que monitor abrimos la ventana de faceplace

p = strchr(ventanaImagen,'1');
if ( p ) {
monitor = 1;
}
else monitor = 2;

//Se concatena la ventana de imagen con el numero de monitor

```

```

sprintf(ventanaImagen, "VentanaPuertas%d", monitor);

//Gestion de la coordenada donde se abre la
ventana_____

//Obtiene el punto de apertura de la ventana de mando de forma dinámica
devuelveCoord(&posx, &posy);
//printf("CoordX: %d ,CoordY: %d\n", posx, posy);

SetLeft(ventanaBase ,ventanaImagen, (monitor-1)*1920 + posx);
//Return-Type: BOOL
SetTop(ventanaBase ,ventanaImagen, posy); //Return-Type: BOOL

//_____

if (devuelveTagTDS(lpszObjectName, ch) != NULL ) //se comprueba que el
tag no sea nulo
{
    //printf("Tag que llega a la función \n\r: %s ", tempTag);
    strcpy(nombreTag, devuelveTagTDS(lpszObjectName, ch));
    //printf("Nombre tag leído de la función: \n\r %s ", nombreTag);

    // A continuación se comprueba la existencia del tag. Como es
    una estructura comprobamos uno de sus elementos, por ejemplo:
    nombreTag.QIntEstado:
    strcpy ( var, nombreTag );
    strcat ( var, ".info0" );
    GetTagCharState ( var, &dwState );

    if (dwState == 0 || dwState != 0x800 ) // Se analiza: si no
    hay error (0) o bien el error es diferente de 800h (sin conexión).
    {
        //Cocatenamos "." a continuación del lpszObjectName para
        apuntar a la variable de la estructura MOTOR
        strcpy (tag_prefix, nombreTag);
        strcat (tag_prefix, ".");
        SetPropChar
        (ventanaBase ,ventanaImagen, "TagPrefix", tag_prefix);

        //En la estructura MOTOR, la variable interna Descripcion
        contiene la descripcion del motor para el CaptionText de la pantalla.
        strcpy (tag_descripcion, tag_prefix);
        strcat (tag_descripcion, "Descripcion");
        SetPropChar (ventanaBase ,ventanaImagen, "CaptionText",
        GetTagChar(tag_descripcion));

        //En la estructura MOTOR, la variable interna Faceplate
        contiene el nombre de la pantalla de faceplate que queremos usar para
        este motor.
        strcpy (tag_pantalla, tag_prefix );
        strcat (tag_pantalla, "Faceplate" );
        SetPropChar (ventanaBase ,ventanaImagen, "PictureName",
        GetTagChar(tag_pantalla)); //Se selecciona la imagen de mando
        SetPropBOOL (ventanaBase ,ventanaImagen, "Visible", 1 );
        //Tipo de retorno :BOOL
    }
    else
    {

```



```

        printf("-->%s . %s : Problema con el TAG: %s \r\n",
lpszPictureName, lpszObjectName, nombreTag);
        printf("Código de Error (hex): %X \r\n", dwState);
    }
} else
{
    printf("-->%s . %s : La función ha devuelto un valor NULL: \r\n",
lpszPictureName, lpszObjectName);
}

//se libera explícitamente la memoria de las zonas de memorias que se
reservan
SysFree(tag_prefix);
SysFree(tag_descripcion);
SysFree(tag_pantalla);
SysFree(var);
SysFree(nombreTag);
}

```

Esta función se usa en el evento "Hacer clic con el ratón" de las instancias de faceplates de las puertas, para abrir la ventana de mando donde se visualizan las informaciones de estado de la puerta, las alarmas y los botones de mando si es que lo presenta.

Al igual que en el resto de funciones para abrir ventanas de mando, se emplea el nombre del objeto que corresponde con el declarado en las variables de estructura como TagPrefix de la ventana de imagen en la que se abrirá la imagen de proceso del mando. De esta forma se consigue reciclar la misma imagen para objetos con la misma estructura de variables.

#### *Función ventanaVariadoresTDS:*

```

void        ventanaVariadoresTDS(char*        lpszPictureName,        char*
lpszObjectName)
{
/*****
**** 28/11/2018****
*****/
#include "apdefap.h"

char* ventanaBase;
char* ventanaImagen;
char* ventanaFaceplate= (char *)SysMalloc(80);

char* tag_descripcion1= (char *)SysMalloc(80);
char* tag_pantalla1= (char *)SysMalloc(80);
char* tag_tipo= (char *)SysMalloc(80);
char* tag_variador= (char *)SysMalloc(80);
char* var= (char *)SysMalloc(80);
char *nombreTag= (char *)SysMalloc(120);

char *p;
int monitor;
const char ch= '*'; //delimitador
DWORD dwState= 0;

int posx, posy;

ventanaBase= GetParentPicture(lpszPictureName);
ventanaImagen= GetParentPictureWindow(lpszPictureName);

```

```

// monitor =
GetPropWord(ventanaBase ,ventanaImagen,"WindowMonitorNumber");
// En WebNavigator no se puede utilizar la propiedad
WindowMonitorNumber de las ventanas de imagen porque siempre vale
null, en vez de eso utilizamos el nombre de la ventana de imagen
//buscamos en el nombre que contenga el numero de moniro (1 o 2) para
saber en que monitor abrimos la ventana de faceplate

p = strchr(ventanaImagen,'1');
if ( p ) {
    monitor = 1;
}
else monitor = 2;

//Se concatena la ventana de imagen con el numero de monitor
sprintf(ventanaImagen,"Variadores%d",monitor);

//Se concatena la ventana de imagen del FP con el numero de monitor
sprintf(ventanaFaceplate,"VentanaFaceplate%d",monitor);

//Gestion de la coordenada donde se abre la
ventana_____

cursorCoord(&posx, &posy); //Obtiene las coordenadas del cursor
//printf("CoordX: %d ,CoordY: %d\n", posx, posy);

if(GetOffsetLeft(ventanaBase, ventanaFaceplate)==0){ //Cuando la
ventana del FP del motor extá maximizada (Sin OFFSet)
    if(((posx<=1920)&&(posx<=1113))||((posx>1920)&&(posx<=3033))){
        SetLeft(ventanaBase ,ventanaImagen,posx+276);
        //Return-Type: BOOL
        SetTop(ventanaBase ,ventanaImagen,posy-90);
        //Return-Type: BOOL
    }else{
        SetLeft(ventanaBase ,ventanaImagen,posx-1115);
        //Return-Type: BOOL
        SetTop(ventanaBase ,ventanaImagen,posy-90);
        //Return-Type: BOOL
    }
}else{ //Cuando la ventana del FP del motor extá minimizada (Con
OFFSet)
    if(((posx<=1920)&&(posx<=1375))||((posx>1920)&&(posx<=3295))){
        SetLeft(ventanaBase ,ventanaImagen,posx+22);
        //Return-Type: BOOL
        SetTop(ventanaBase ,ventanaImagen,posy-90);
        //Return-Type: BOOL
    }else{
        SetLeft(ventanaBase ,ventanaImagen,posx-854);
        //Return-Type: BOOL
        SetTop(ventanaBase ,ventanaImagen,posy-90);
        //Return-Type: BOOL
    }
}

//_____

if (devuelveTagTDS(lpszObjectName,ch) != NULL ) //se comprueba que el
tag no sea nulo
{

```

```

//printf("Tag que llega a la función \n\r: %s ", tempTag);
strcpy(nombreTag, devuelveTagTDS (lpszObjectName, ch));
//printf("Nombre tag leído de la función: \n\r %s ", nombreTag);

// A continuación se comprueba la existencia del tag. Como es
una estructura comprobamos uno de sus elementos, por ejemplo:
nombreTag.QIntEstado:
strcpy(var, nombreTag);
strcat(var, "@NOTP::OPAuto");
GetTagCharState(var, &dwState);

if(dwState == 0 || dwState != 0x800) // Se analiza: si no hay
error (0) o bien el error es diferente de 800h (sin conexión).
{
//En la estructura MOTOR, la variable interna Tipo contiene el
tipo de motor que queremos usar para visualizar la pantalla de variador
//TIPO = 1 --> Estandar / TIPO = 2 --> Inversor / TIPO = 3 -->
Variador
strcpy      (tag_tipo, nombreTag );
strcat      (tag_tipo, "@NOTP::Tipo" );
if ( GetTagByte(tag_tipo)== 3 )
{
//Cocatenamos "_VAR." a continuación del objeto
referenciado por NombreTAg para apuntar a la variable de la estructura
VARIADOR

strcpy(tag_variador, nombreTag);
strcat(tag_variador, "_VAR.");
SetPropChar(ventanaBase , ventanaImagen, "TagPrefix",
tag_variador);

//En la estructura VARIADOR, la variable interna
Tipo_Variador contiene la descripción del tipo de variador para el
CaptionText de la pantalla.
strcpy(tag_descripcion1, tag_variador);
strcat(tag_descripcion1, "@NOTP::Tipo_Variador");

//En la estructura VARIADOR, la variable interna
Faceplate_VAR contiene el nombre de la pantalla de faceplate que
queremos usar para este variador
strcpy(tag_pantallal, tag_variador);
strcat(tag_pantallal, "@NOTP::Faceplate_VAR");

SetPropChar(ventanaBase , ventanaImagen, "CaptionText",
GetTagChar(tag_descripcion1));

SetPropChar(ventanaBase , ventanaImagen, "PictureName",
GetTagChar(tag_pantallal));

if(GetPropBOOL(ventanaBase , ventanaImagen, "Visible")==0) {

SetPropBOOL(ventanaBase , ventanaImagen, "Visible", 1);
}
else{

SetPropBOOL(ventanaBase , ventanaImagen, "Visible", 0);
}
}
else{

```

```

        SetPropBOOL(ventanaBase ,ventanaImagen,"Visible",0 );
    }

}
else{
    printf("-->%s . %s : Problema con el TAG: %s \r\n",
lpszPictureName, lpszObjectName, nombreTag);
    printf("Código de Error (hex): %X \r\n", dwState);
}
}
else{
    printf("-->%s . %s : La función ha devuelto un valor NULL: \r\n",
lpszPictureName, lpszObjectName);
}

//se libera explícitamente la memoria de las zonas de memorias que se
reservan
SysFree(ventanaFaceplate);
SysFree(tag_descripcion1);
SysFree(tag_pantalla1);
SysFree(tag_tipo);
SysFree(tag_variador);
SysFree(var);
SysFree(nombreTag);
}

```

Esta función se usa en el evento "Hacer clic con el ratón" del botón de los variadores de la imagen de mando de los motores para abrir la ventana donde se muestra la información del variador del motor.

Al igual que en el resto de funciones para abrir ventanas de mando, se emplea el nombre del objeto que corresponde con el declarado en las variables de estructura como TagPrefix de la ventana de imagen en la que se abrirá la imagen de proceso del mando. De esta forma se consigue reciclar la misma imagen para objetos con la misma estructura de variables.

### Librería de funciones en VBS

```

'Funciones      para      manejo      de      BBDD
SQL_____

Function CompruebaEntrada_DB(conStr, consultaSQL) 'Funcion para comprobar si
existe La entrada de un dato en La Base de Datos
    Dim objConnection
    Dim objCommand
    Dim objRecordset
    Dim objTag
    Dim strSQL
    Dim lngValue
    Dim lngCount

    strSQL = consultaSQL 'Consulta SQL

    Set objConnection = CreateObject("ADODB.Connection")
    objConnection.ConnectionString = conStr 'Connection String

```

```

objConnection.Open 'Abre La conexión con La DB
Set objRecordset = CreateObject("ADODB.Recordset")
Set objCommand = CreateObject("ADODB.Command")
objCommand.ActiveConnection = objConnection
objCommand.CommandText = strSQL
Set objRecordset = objCommand.Execute 'Ejecuta La consulta SQL
lngCount = objRecordset.Fields.Count
If (lngCount>0) Then
    objRecordset.movefirst
    CompruebaEntrada_DB = objRecordset.Fields(0).Value
Else
    HMIRuntime.Trace "Selection returned no fields" & vbNewLine
End If
Set objCommand = Nothing
objConnection.Close 'Cierra La conexión con La DB
Set objRecordset = Nothing
Set objConnection = Nothing
End Function

Function Escribe_DB(conStr, consultaSQL) 'Funcion de escritura en DB
    Dim objConnection
    Dim objCommand
    Dim objRecordset
    Dim objTag
    Dim lngValue

    Set objConnection = CreateObject("ADODB.Connection")
    objConnection.ConnectionString = conStr 'Cadena de conexion con el
servidor de La DB
    objConnection.Open 'Abre La conexión con La DB
    Set objRecordset = CreateObject("ADODB.Recordset")
    Set objCommand = CreateObject("ADODB.Command")
    objCommand.ActiveConnection = objConnection
    objCommand.CommandText = consultaSQL 'Consulta SQL
    Set objRecordset = objCommand.Execute 'Ejecuta La consulta SQL
    Set objCommand = Nothing
    objConnection.Close 'Cierra La conexión con La DB
    Set objRecordset = Nothing
    Set objConnection = Nothing
End Function

Function Leer_DB(conStr, Tabla_rd, columna, Campo, Valor) 'Funcion Lectura
en DB
    Dim objConnection
    Dim objCommand
    Dim objRecordset
    Dim objTag
    Dim conSQL
    Dim lngValue
    Dim lngCount

    conSQL = "SELECT ["&columna&"] FROM [BaseDatos].[dbo].["&Tabla_rd&"]
WHERE ["&Campo&"]="&Valor 'Consulta SQL para Leer La descripción según el
codigo del producto
    Set objConnection = CreateObject("ADODB.Connection")
    objConnection.ConnectionString = conStr 'Connection String con el
servidor donde esta La DB
    objConnection.Open
    Set objRecordset = CreateObject("ADODB.Recordset")

```

```

Set objCommand = CreateObject("ADODB.Command")
objCommand.ActiveConnection = objConnection
objCommand.CommandText = strSQL
Set objRecordset = objCommand.Execute
lngCount = objRecordset.Fields.Count
If (lngCount>0) Then
    objRecordset.movefirst
    Leer_DB = objRecordset.Fields(0).Value
Else
    HMIRuntime.Trace "Selection returned no fields" & vbNewLine
End If
Set objCommand = Nothing
objConnection.Close
Set objRecordset = Nothing
Set objConnection = Nothing
End Function

'Función para Leer descripcion de Los productos de La base de datos
Function Leer_desc(code_prod)

    Dim objConnection
    Dim objCommand
    Dim objRecordset
    Dim strConnectionString
    Dim strSQL
    Dim lngValue
    Dim lngCount
    Set strConnectionString = HMIRuntime.Tags("@NOTP::DB_Conn_StringEI")
    'Conexion con La variable que tiene el token para acceder a La DB
    strConnectionString.Read
    strSQL = "SELECT [Prod_Desc] FROM [BaseDatos].[dbo].[PRODUCTOS] WHERE
[Prod_Codigo]=" &code_prod 'Consulta para Leer La descripcion segun el codigo
del producto
    Set objConnection = CreateObject("ADODB.Connection")
    objConnection.ConnectionString = strConnectionString.Value
    objConnection.Open
    Set objRecordset = CreateObject("ADODB.Recordset")
    Set objCommand = CreateObject("ADODB.Command")
    objCommand.ActiveConnection = objConnection
    objCommand.CommandText = strSQL
    Set objRecordset = objCommand.Execute
    lngCount = objRecordset.Fields.Count
    If (lngCount>0) Then
        objRecordset.movefirst
        lngValue = objRecordset.Fields(0).Value
        Leer_desc = lngValue
    Else
        HMIRuntime.Trace "Selection returned no fields" & vbNewLine
    End If
    Set objCommand = Nothing
    objConnection.Close
    Set objRecordset = Nothing
    Set objConnection = Nothing

End Function

'Función para Leer el valor del color de La base de datos
Public Function Leer_color(code_prod, nombre_objeto)

```

```

Dim objConnection
Dim objCommand
Dim objRecordset
Dim objTag
Dim strConnectionString
Dim strSQL
Dim lngValue
Dim lngCount
Set strConnectionString = HMIRuntime.Tags("@NOTP::DB_Conn_StringEI")
'Conexion con la variable que tiene el token para acceder a la DB
strConnectionString.Read
strSQL = "SELECT [Prod_Color] FROM [BaseDatos].[dbo].[PRODUCTOS] WHERE
[Prod_Codigo]=" &code_prod 'Consulta para leer la descripción según el código
del producto
Set objConnection = CreateObject("ADODB.Connection")
objConnection.ConnectionString = strConnectionString.Value
objConnection.Open
Set objRecordset = CreateObject("ADODB.Recordset")
Set objCommand = CreateObject("ADODB.Command")
objCommand.ActiveConnection = objConnection
objCommand.CommandText = strSQL
Set objRecordset = objCommand.Execute
lngCount = objRecordset.Fields.Count
If (lngCount>0) Then
    objRecordset.movefirst
    lngValue = objRecordset.Fields(0).Value
    Set objTag = HMIRuntime.Tags(nombre_objeto & "." & "color")
    objTag.Write lngValue
Else
    HMIRuntime.Trace "Selection returned no fields" & vbNewLine
End If
Set objCommand = Nothing
objConnection.Close
Set objRecordset = Nothing
Set objConnection = Nothing
End Function
'
-----
-

Sub encuadrePantalla(Item, oPbase, ovImagen, sImagen, iZoom, iSup, iIzq)

'Version 0.12 03/08/2018
Dim objGraphicView
Set objGraphicView = HMIRuntime.Screens(oPbase).ScreenItems(ovImagen)

' En cualquier caso, se invisibilizan las ventanas abiertas. De ser
necesario, abrirlo posteriormente.
Call cerrarVentanas (oPbase,Item.Parent.ObjectName)
' CONFIGURACIÓN DE LA PANTALLA DE PROCESO VISIBLE EN ESTE MOMENTO:

'Comprobar que la ventana de imagen contiene la imagen correcta indicada en
la función llamada por el botón.
'Si no es la correcta, se carga la imagen correcta
If objGraphicView.PictureName <> sImagen Then
    objGraphicView.PictureName = sImagen

```

```

End If

'Zoom:
objGraphicView.Zoom = iZoom

'Si La ventana está configurada para que existan scrollbars -> se centra La
ventana donde se desee
If ( objGraphicView.ScrollBars = vbTrue) Then
    objGraphicView.scrollPositionx = iIzq
    objGraphicView.scrollPositionY = iSup
End If
' Revisar si usar ObjectSizeDeclutteringMin / Max /Enable para dejar de
visualizar objetos cuando su tamaño de pixel al escalarse sea menor que uno.
End Sub

Sub encuadreDetalle(oPbase, oVentanaImagen, oImagenCargar, iPosSup, iPosIzq,
iAlto, iAncho )

'Version 0.1 - 26/07/2018
Dim objGraphicView
Dim coleccion
Dim elemento

On Error Resume Next 'iniciamos La gestión de errores

'Selección de La ventana de imagen de detalle. Si no existe se registra un
mensaje de debug y no se continúa La ejecución.
'Declarar La ventana
Set objGraphicView =
    HMIRuntime.Screens(oPbase).ScreenItems(oVentanaImagen)
'Control de errores
If 0 <> Err.Number Then
    HMIRuntime.Trace "Error Capturado: GlobalScript->Sub
encuadreDetalle: no existe el objeto " & oPbase & ">" & oVentanaImagen &
vbCrLf
    'Delete error message
    Err.Clear
Else ' No hay errores
    'HMIRuntime.Trace "Se ha encontrado: " & oVentanaImagen &
vbNewLine
    'Configurar La ventana
    objGraphicView.PictureName = oImagenCargar
    objGraphicView.Top = iPosSup
    objGraphicView.Left = iPosIzq
    objGraphicView.Width = iAncho
    objGraphicView.Height = iAlto
    objGraphicView.Visible = vbTrue

End If

On Error Goto 0 ' Desactivar control de errores.
End Sub

Sub cerrarDetalle(oPbase,oVentanaImagen)

```



```

'Version 0.1 - 26/07/2018

Dim objGraphicView
On Error Resume Next ' Activar control de errores

'Declarar La ventana
Set objGraphicView =
HMIRuntime.Screens(oPbase).ScreenItems(oVentanaImagen)

'Control de errores
If 0 <> Err.Number Then
    HMIRuntime.Trace "Error Capturado: GlobalScript->Sub cerrarDetalle:
no existe el objeto " & oPbase & ">" & oVentanaImagen & vbCrLf
    'Delete error message
    Err.Clear
Else
    objGraphicView.Visible = vbFalse

End If

On Error Goto 0 ' Desactivar control de errores.
End Sub

Sub cerrarVentanas(oPbase,objParent)
'Version 0.1 - 02/08/2018

Dim objGraphicView
Dim arrayVentanas(13)
Dim i
Dim monitor
On Error Resume Next ' Activar control de errores

If InStr(objParent,"1")<>0 Then
    monitor="1"
Else
    monitor="2"
End If

arrayVentanas(1) = "Analogicas"&monitor
arrayVentanas(2) = "VentanaFaceplate"&monitor
arrayVentanas(3) = "Variadores"&monitor
arrayVentanas(4) = "Valvulas"&monitor
arrayVentanas(5) = "VentanaDetalleZona"&monitor
arrayVentanas(6) = "VentanaSeleccion"&monitor
arrayVentanas(7) = "VentanaAjustable"&monitor
arrayVentanas(8) = "VentanaBotoneraUp"&monitor
arrayVentanas(9) = "VentanaMenuDropUp"&monitor
arrayVentanas(10) = "VentanaInterruptores"&monitor
arrayVentanas(11) = "VentanaTrafoGen"&monitor
arrayVentanas(12) = "Estanterias_sec"&monitor

For i = 1 To 12
    'Declarar La ventana
    Set objGraphicView =
HMIRuntime.Screens( oPbase ).ScreenItems( arrayVentanas(i) )
    If 0 <> Err.Number Then 'Control de errores

```

```

HMIRuntime.Trace "Error Capturado: GlobalScript->Sub
cerrarVentanas: no existe el objeto " & oPbase & ">" & arrayVentanas(i) &
vbCrLf
    'Delete error message
    Err.Clear
Else
    objGraphicView.Visible = vbFalse
End If
Next

On Error Goto 0 ' Desactivar control de errores.
End Sub

Sub logeoActuacionOperador(almState, sAviso, sResultado, sNombreElto,
sDescElto )
'Version 0.1 - 31/07/2018

'FUNCIÓN PARA LA ESCRITURA DE ACTUACIONES DE OPERADOR TALES COMO CAMBIO DE
MODO O ESCRITURAS SQL

'almState = Por defecto 5
'sDescripción = Texto a incluir en el Logueo de la actuación
'sResultado = [opcional] Texto que indica resultado de la operación SQL
'sNombreElto= [opcional] De tratarse de un motor

' EJEMPLO INSERCIÓN válvula:
'
' Dim Motor,Descripcion
' Set Descripcion = HMIRuntime.Tags ("Descripcion")
' Set Motor = HMIRuntime.Tags ("NombreMotor")
' Call logeoActuacionOperador( 5, "Pulsador Mando Automático Valvula", ,
Motor.Value, Descripcion.Value )
' Pulsador Mando Automático Valvula V1_230 / E.V. Embrague molino

Dim MyAlarm, User

Set User =
HMIRuntime.Tags("@NOTP::@CurrentUserName") 'Nombre de Usuario actual
Set MyAlarm = HMIRuntime.Alarms(999999)
User.Read

HMIRuntime.Trace "almState" & almState & vbNewLine

MyAlarm.State = almState
MyAlarm.UserName = User.Value

If sNombreElto = Empty Then
    ' Para logueos de SQL, etc
    MyAlarm.ProcessValues(10) = sAviso & " " & sResultado
Else
    ' Para logueos de elementos tipo motor, válvula
    MyAlarm.ProcessValues(10) = sAviso & " " & sResultado & " " &
sNombreElto & " / " & sDescElto
End If

```

```
MyAlarm.Create "MyApplication"
```

```
End Sub
```

```
Function colorProducto(nombreObjeto) 'Funcion de para dinamizar Los colores de Las estanterias y vagonetas CON SCRIPT GLOBAL  
'Se hace uso de la variable local 'prodAux' para que en la primera ejecución de la pantalla no se realicen todas las consultas a la BD  
'de las estanterias y se ralentice el sistema.
```

```
Dim groupProd
```

```
Set groupProd= HMIRuntime.Tags.CreateTagSet  
groupProd.Add nombreObjeto & ".producto"  
groupProd.Add nombreObjeto & ".prodAux"
```

```
groupProd.Read
```

```
If groupProd(nombreObjeto & ".prodAux").Value <> groupProd(nombreObjeto & ".producto").Value Then
```

```
    'HMIRuntime.Trace "dentro!!" & vbNewLine 'debug
```

```
    'Se comprueba que dicho producto existe en la base de datos
```

```
    Dim numeroEntradas, conStr, qerySQL
```

```
    conStr = HMIRuntime.Tags("@NOTP::DB_Conn_StringEI").Read 'Conexion con la variable que tiene el token para acceder a la DB
```

```
    qerySQL = "SELECT count(Prod_Color) FROM [BaseDatos].[dbo].[PRODUCTOS] WHERE [Prod_Codigo]=" &groupProd(nombreObjeto & ".producto").Value 'Consulta para Leer la descripcion segun el codigo del producto
```

```
    numeroEntradas = CompruebaEntrada_DB(conStr, qerySQL) 'Devuelve el numero de productos definidos en la Tabla
```

```
    'Si hay entrada del producto en la base de datos se lee su color
```

```
    If numeroEntradas>0 Then
```

```
        Call Leer_color(groupProd(nombreObjeto&".producto").Value, nombreObjeto) 'funcion para Leer el color de la base de datos (escribe en *.color)
```

```
        groupProd(nombreObjeto & ".prodAux").Write groupProd(nombreObjeto & ".producto").Value 'escribe el codigo del producto en la variable Local
```

```
    End If
```

```
End If
```

```
End Function
```

```
Function tdsLOGDATE 'Formato de fecha y hora(yyyymmddhhmmss)
```

```
    tdsLOGDATE= Year(Date)& Right(String(2,"0")&Month(Date),2) & Right(String(2,"0")&Day(Date),2) & Right(String(2,"0")& Hour(Time),2) & Right(String(2,"0")& Minute(Time),2) & Right(String(2,"0")& Second(Time),2)  
End Function
```

## 2.1.4.2. Acciones globales implementadas

### Acciones globales en C

#### AlarmasComunicacionPLCs.pas

```
#include "apdefap.h"

int gscAction( void )
{
    char* var=(char*)SysMalloc(80);
    char* alarFCOM=(char*)SysMalloc(80);
    char* wdog=(char*)SysMalloc(80);
    char* wdog_old=(char*)SysMalloc(80);
    char* alarPLCStp=(char*)SysMalloc(80);

    DWORD dwState= 0;
    int i;

    for (i=1; i<26; i++){
        //Concatenación de las estaciones con las variables
        sprintf(var, "Estacion%d", i);
        strcpy(alarFCOM, var);
        strcpy(wdog, var);
        strcpy(wdog_old, var);
        strcpy(alarPLCStp, var);
        strcat(var, ".qCode"); //En var tenemos el nombre de la variable
        a la que queremos acceder para leer el QualityCode de la señal
        strcat(alarFCOM, ".FalloCOM"); //Cadena de la variable de alarma
        strcat(wdog, ".wdog"); //En wdog tenemos el valor de un contador
        interno del PLC
        strcat(wdog_old, ".old_wdog"); //En old_wdog registrado
        localmente el valor anterior del contador interno del PLC
        strcat(alarPLCStp, ".PLCStop"); //En old_wdog registrado
        localmente el valor del contador interno del PLC

        GetTagBitState(var, &dwState); //Obtiene el Quality Code de la
        variable binaria
        if (dwState == 0){ //Ningún error (Estados de variable en Manual
        WinCC)
            SetTagBit(alarFCOM, 0);
            //Comprobar si el PLC está en Stop
            if(GetTagWord(wdog)== GetTagWord(wdog_old)){
                SetTagBit(alarPLCStp, 1); //activa la variable de
                alarma de PLC en STOP
            }else{
                SetTagBit(alarPLCStp, 0); //PLC en RUN
            }
        }
        else{ //Hay algún error con el dispositivo
            SetTagBit(alarFCOM, 1); //activa la variable de alarma de
            fallo de comunicación del PLC
            //printf("Código de Calidad (dec): %d \r\n", dwState);
        }
        SetTagWord(wdog_old, GetTagWord(wdog)); //Guardo el valor del
        watchdog en la variable local
    }

    //Liberacion de memoria reservada
    SysFree(var);
}
```

```

SysFree (alarFCOM) ;
SysFree (wdog) ;
SysFree (wdog_old) ;
SysFree (alarPLCStp) ;

return 0;

}

```

Este script se ejecuta cíclicamente cada 10 segundos para comprobar el estado de las comunicaciones con los PLCs. Si alguno de ellos está en STOP se genera una alarma de “Parado” y si no hay comunicación con él se genera también la alarma de “Fallo comunicación”

### Acciones globales en VBS

Se ha desarrollado el siguiente Script en VBS para calcular los colores de las vagonetas y estanterías.

coloresEstanteriasVagonetas.bac

```

Option Explicit
Function action
Dim i, k

'COLORES FÁBRICA 3
For i = 1 To 13
'Cálculo de Los colores SECADERO F3
If (i=1 Or i=2) Then 'Vía 1 o 2
For k=1 To 29
colorProducto("via"&i&"_"&k)
Next
End If
If (i=3 Or i=4) Then 'Vía 3 o 4
For k=1 To 19
colorProducto("via"&i&"_"&k)
Next
End If
If i=5 Then 'Vía 5
For k=1 To 20
colorProducto("via"&i&"_"&k)
Next
End If
If i=6 Then 'Vía 6
For k=1 To 10
colorProducto("via"&i&"_"&k)
Next
End If
'Cálculo de Los colores HORNO F3
If (i=7 Or i=8) Then 'Vía 7 o 8
For k=1 To 18
colorProducto("via"&i&"_"&k)
Next
End If
If i=9 Then 'Vía 9
For k=1 To 38
colorProducto("via"&i&"_"&k)
Next
End If
If (i=10 Or i=11) Then 'Vía 10 o 11

```

```

        For k=1 To 11
            colorProducto("via"&i&"_"&k)
        Next
    End If
    If i=12 Then 'Vía 12
        For k=1 To 5
            colorProducto("via"&i&"_"&k)
        Next
    End If
    If i=13 Then 'Vía 13
        For k=1 To 25
            colorProducto("via"&i&"_"&k)
        Next
    End If
Next
End Function

```

Este script se ejecuta cíclicamente cada 500ms y escribe el valor del color del producto o vagoneta en su variable “.color” correspondiente a cada uno de los objetos.

### 2.1.4.3. Programación de objetos

En los sinópticos de proceso de la planta de producción están reflejados los objetos que representan los dispositivos físicos tales como: motores, válvulas, cintas y todos aquellos accionamientos automáticos de los cuales se necesita saber su estado real en todo momento para conseguir controlar el proceso productivo.

Para ello es necesario programar la lógica de dinamización de los distintos atributos que se desean ver reflejados en el SCADA.

Los objetos que se desarrollan son de tipo faceplate para reducir el tiempo de desarrollo y obtener mayor flexibilidad si es necesario realizar un cambio en la representación, pero no se usa la programación empleando SmartTags debido a su ineficiencia al tener gran cantidad de instancias dentro de un mismo sinóptico de proceso.

A continuación se detalla la programación de cada tipo de instancia de faceplate que se realiza desde el editor Graphics Designer.

#### Programación de Motores

Se han desarrollado varios faceplates para motores según su orientación y representación gráfica, ya que los faceplates no se pueden girar, pero todos ellos tienen los mismos atributos y su dinamización se programa de la misma forma.



Instancia de faceplate – Motor

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

*Atributo Información sobre herramientas:*

```
Function ToolTipText_Trigger(ByVal Item)
'Dinamización del ToolTip en función del nombre del objeto.
Dim TagName, TextPos
Dim Name, Desc

'Formato del nombre del objeto: PrefijodePosicion*NombreTag
TextPos = InStr( 1, Item.ObjectName, "*",vbBinaryCompare )
TagName = Mid ( Item.ObjectName,TextPos+1, Len(Item.ObjectName) )

Name = HMIRuntime.Tags(TagName & ".NombreMotor").Read
Desc = HMIRuntime.Tags(TagName & ".Descripcion").Read

ToolTipText_Trigger = CStr(Name&" "&Desc)
End Function
```

**Triggers de variable:**

NombreObjeto.NombreMotor	Al cambiar
NombreObjeto.Descripcion	Al cambiar

Se dinamiza la información sobre herramientas para que cuando el operario sitúe el cursor del ratón encima del objeto aparezca un tooltip donde se refleja el nombre del motor y su descripción.

*Atributo QIntState:*

```
Function QIntState_Trigger(ByVal Item)
Dim TagName, TextPos
Dim Marcha(1)
Dim Alarma(21)
Dim objGroup, i

'Formato del nombre del objeto: PrefijodePosicion*NombreTag
TextPos = InStr(1, Item.ObjectName, "*",vbBinaryCompare )
TagName = Mid(Item.ObjectName,TextPos+1,Len(Item.ObjectName))

Set objGroup= HMIRuntime.Tags.CreateTagSet

'se anaden las variables de informacion al grupo
objGroup.Add TagName & "." &"Info2_Informacion1"

For i=0 To 21 'se anaden las variables de alarmas al grupo
    objGroup.Add TagName & "." &"Info1_Alarma"&i
Next

objGroup.Read

Marcha(0)=objGroup(TagName & "." &"Info2_Informacion1").Value

'Estado Marcha/Paro/Alarma
For i=0 To 21
    Alarma(i)=objGroup(TagName & "." &"Info1_Alarma"&i).Value
    If Alarma(i)=1 Then
        QIntState_Trigger = 2 'Alarma
    End If
Next
```

```

If Alarma(0)=0 And Alarma(1)=0 And Alarma(2)=0 And Alarma(3)=0 And
Alarma(4)=0 And Alarma(5)=0 And Alarma(6)=0 And Alarma(7)=0 And Alarma(8)=0
And Alarma(9)=0 And Alarma(10)=0 And Alarma(11)=0 And Alarma(12)=0 And
Alarma(13)=0 And Alarma(14)=0 And Alarma(15)=0 And Alarma(16)=0 And
Alarma(17)=0 And Alarma(18)=0 And Alarma(19)=0 And Alarma(20)=0 And
Alarma(21)=0 Then
    If Marcha(0)=1 Or Marcha(1)=1 Then 'bits que indican motor funcionando
        QIntState_Trigger = 4 'En marcha
    Else
        QIntState_Trigger = 0 'Parado
    End If
End If

End Function

```

**Triggers de variable:**

NombreObjeto.Info2_Informacion0	Al cambiar
NombreObjeto.Info2_Informacion1	Al cambiar
NombreObjeto.Info2_Informacion2	Al cambiar
NombreObjeto.Info2_Informacion3	Al cambiar
NombreObjeto.Info2_Informacion4	Al cambiar
NombreObjeto.Info2_Informacion5	Al cambiar
NombreObjeto.Info2_Informacion6	Al cambiar
NombreObjeto.Info2_Informacion7	Al cambiar
NombreObjeto.Info2_Informacion8	Al cambiar
NombreObjeto.Info2_Informacion9	Al cambiar
NombreObjeto.Info2_Informacion10	Al cambiar
NombreObjeto.Info2_Informacion11	Al cambiar
NombreObjeto.Info2_Informacion12	Al cambiar
NombreObjeto.Info2_Informacion13	Al cambiar
NombreObjeto.Info2_Informacion14	Al cambiar
NombreObjeto.Info2_Informacion15	Al cambiar
NombreObjeto.Info1_Alarma0	Al cambiar
NombreObjeto.Info1_Alarma1	Al cambiar
NombreObjeto.Info1_Alarma2	Al cambiar
NombreObjeto.Info1_Alarma3	Al cambiar
NombreObjeto.Info1_Alarma4	Al cambiar
NombreObjeto.Info1_Alarma5	Al cambiar
NombreObjeto.Info1_Alarma6	Al cambiar
NombreObjeto.Info1_Alarma7	Al cambiar
NombreObjeto.Info1_Alarma8	Al cambiar
NombreObjeto.Info1_Alarma9	Al cambiar
NombreObjeto.Info1_Alarma10	Al cambiar
NombreObjeto.Info1_Alarma11	Al cambiar
NombreObjeto.Info1_Alarma12	Al cambiar
NombreObjeto.Info1_Alarma13	Al cambiar
NombreObjeto.Info1_Alarma14	Al cambiar
NombreObjeto.Info1_Alarma15	Al cambiar
NombreObjeto.Info1_Alarma16	Al cambiar
NombreObjeto.Info1_Alarma17	Al cambiar
NombreObjeto.Info1_Alarma18	Al cambiar
NombreObjeto.Info1_Alarma19	Al cambiar
NombreObjeto.Info1_Alarma20	Al cambiar
NombreObjeto.Info1_Alarma21	Al cambiar



Con este script se calcula el valor entero que determina el estado del motor para representar cuando está en marcha, parado o en estado de alarma.

Como puede observarse se configuran todos los triggers de variables de estado y alarmas independientemente de la cantidad real que presente el motor por si en un futuro se decide añadir más informaciones o alarmas solo sea necesario añadirlas en la lista de alarmas y no sea necesario añadir los triggers en cada una de las imágenes donde este instanciado el objeto que representa el motor.

Atributo *QIntControl\_M\_A*:

The screenshot shows a dialog box titled "Rango de valores" for the attribute "QIntControl\_M\_A". The dialog is in Spanish. It contains the following fields and options:

- Idioma utilizado: Español (España, internacional)
- Nombre del evento: Variable
- Expresión / Fórmula: 'M1\_150.Info2\_Informacion0'
- Resultado de la expresión/fórmula: A table with two columns: "Ámbito de validez" and "QInt...". The rows are "Sí / TRUE" with value 3, and "No / FALSE" with value 0.
- Tipo de datos: Radio buttons for "Analog.", "Bool." (selected), "Bit", and "Direct.".
- Buttons: "Agregar", "Quitar", "Aceptar", "Cancelar", "Verificar", and "...".
- Options: Radio buttons for "No evaluar estado de variables" (selected), "Evaluar estado de variables", and "Evaluar Quality Code".

El atributo *QIntControl\_M\_A* hace referencia al estado automático o manual del motor y se programa desde el diálogo de dinamización de forma que si la variable binaria "NombreObjeto.Info2\_Informacion0" vale 1 el motor se encuentra en modo automático y se le asigna un 3 al atributo, en caso contrario se encuentra en modo manual y se le asigna un 0. Cuando el motor se encuentra en modo manual se representa un cuadrado azul en la esquina superior izquierda y cuando está en modo automático se oculta de forma que de un vistazo los operarios saben que elementos se encuentran en modo manual.

Por último se programa el evento “Hacer clic con el ratón” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaMotoresTDS( lpszPictureName, lpszObjectName );
}
```

Como parámetros de la función se le pasa el nombre de la ventana de imagen donde se encuentra instanciado el objeto y el nombre del propio objeto.

### Programación de Válvulas

Se han desarrollado varios faceplates para válvulas según su orientación, ya que los faceplates no se pueden girar, pero todos ellos tienen los mismos atributos y su dinamización se programa de la misma forma.



Instancia de faceplate – Válvula

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

#### Atributo Información sobre herramientas:

```
Function ToolTipText_Trigger(ByVal Item)
'Dinamización del Tooltip en función del nombre del objeto.
Dim TagName, TextPos
Dim Name, Desc

'Formato del nombre del objeto: PrefijodePosicion*NombreTag
TextPos = InStr( 1, Item.ObjectName, "*",vbBinaryCompare )
TagName = Mid ( Item.ObjectName,TextPos+1, Len(Item.ObjectName) )

Name = HMIRuntime.Tags(TagName & ".Nombre").Read
Desc = HMIRuntime.Tags(TagName & ".Descripcion").Read

ToolTipText_Trigger = CStr(Name&" "&Desc)
End Function
```

#### Triggers de variable:

NombreObjeto.Nombre	Al cambiar
NombreObjeto.Descripcion	Al cambiar

Se dinamiza la información sobre herramientas para que cuando el operario sitúe el cursor del ratón encima del objeto aparezca un tooltip donde se refleja el nombre del motor y su descripción.

#### Atributo QIntState:

```
Function QIntState_Trigger(ByVal Item)
Dim TagName, TextPos
Dim Marcha(1)
Dim Alarma(2)
```

```

Dim objGroup, i

'Formato del nombre del objeto: PrefijodePosicion*NombreTag
TextPos      = InStr(1, Item.ObjectName, "*",vbBinaryCompare )
TagName      = Mid(Item.ObjectName,TextPos+1,Len(Item.ObjectName))

Set objGroup= HMIRuntime.Tags.CreateTagSet

'se anaden las variables de informacion al grupo
objGroup.Add TagName & "." &"Info2_Informacion1"

For i=0 To 1 'se anaden las variables de alarmas al grupo
  objGroup.Add TagName & "." &"Info1_Alarma"&i
Next

objGroup.Read

Marcha(0)=objGroup(TagName & "." &"Info2_Informacion1").Value

'Estado Marcha/Paro/Alarma
For i=0 To 1
  Alarma(i)=objGroup(TagName & "." &"Info1_Alarma"&i).Value
  If Alarma(i)=1 Then
    QIntState_Trigger = 0 'Alarma
  End If
Next

If Alarma(0)=0 And Alarma(1)=0 Then
  If Marcha(0)=1 Then 'bits que indican valvula funcionando
    QIntState_Trigger = 8 'Abierta
  Else
    QIntState_Trigger = 4 'Cerrada
  End If
End If
End Function

```

**Triggers de variable:**

NombreObjeto.Info2_Informacion1	Al cambiar
NombreObjeto.Info1_Alarma0	Al cambiar
NombreObjeto.Info1_Alarma1	Al cambiar

Con este script se calcula el valor entero que determina el estado de la valvula para representar cuando está en abierta, cerrada o en estado de alarma.

Como las válvulas son un objeto muy simple y con pocas informaciones solo se añade en los triggers la variable binaria “NombreObjeto.Info2\_Informacion1” que indica el estado abierta o cerrada según su valor sea 1 o 0 respectivamente.

Por último se programa el evento “Hacer clic con el ratón” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
char* ventanaBase;
char* ventanaImagen;
char* tag_prefix=(char*)SysMalloc(80);
char* tag_descripcion=(char*)SysMalloc(80);

```

```

char* tag_pantalla=(char*)SysMalloc(80);

char *p;
int monitor;
int posx, posy;

ventanaBase = GetParentPicture(lpszPictureName); //Return-Type: char*
ventanaImagen= GetParentPictureWindow(lpszPictureName); //Return-Type: char*

// En WebNavigator no se puede utilizar la propiedad WindowMonitorNumber de las ventanas de
imagen porque siempre vale null, en vez de eso utilizamos el nombre de la ventana de imagen
//buscamos en el nombre que contenga el numero de moniro (1 o 2) para saber en que monitor
abrimos la ventana de faceplate

p= strchr(ventanaImagen,'1');
if ( p ) {
monitor =1;
}
else monitor =2;

sprintf(ventanaImagen,"Valvulas%d",monitor); //concatena el numero de monitor con el nombre de
la ventana

devuelveCoord(&posx,&posy); //Obtiene las coordenadas donde se pulsa

SetLeft(ventanaBase ,ventanaImagen,(monitor-1)*1920 + posx); //Return-Type: BOOL
SetTop(ventanaBase ,ventanaImagen,posy); //Return-Type: BOOL

//OFFSET IMAGEN y tamaño ajustable
SetOffsetLeft(ventanaBase,ventanaImagen, 260);
SetPropWord(ventanaBase,ventanaImagen,"Width",350); //Altura
SetPropWord(ventanaBase,ventanaImagen,"Height",440); //Anchura

//Cocatenamos "." a continuación del lpszObjectName para apuntar a la variable de la estructura
MOTOR
strcpy(tag_prefix,lpszObjectName);
strcat(tag_prefix,".");
SetPropChar(ventanaBase ,ventanaImagen,"TagPrefix",tag_prefix);

//En la estructura MOTOR, la variable interna Descripcion contiene la descripcion del motor
para el CaptionText de la pantalla.
strcpy(tag_descripcion,tag_prefix);
strcat(tag_descripcion,"Descripcion");
SetPropChar(ventanaBase ,ventanaImagen,"CaptionText", GetTagChar(tag_descripcion));

//En la estructura MOTOR, la variable interna Faceplate contiene el nombre de la pantalla de
faceplate que queremos usar para este motor.
strcpy(tag_pantalla,tag_prefix);
strcat(tag_pantalla,"Faceplate");
SetPropChar(ventanaBase ,ventanaImagen,"PictureName", GetTagChar(tag_pantalla));
SetPropBOOL(ventanaBase ,ventanaImagen,"Visible",1); //Tipo de retorno :BOOL

//Liberacion de memoria
SysFree(tag_prefix);
SysFree(tag_descripcion);
SysFree(tag_pantalla);
}

```

Como el número de objetos de tipo válvula no es muy elevado no se ha desarrollado una función propia para abrir las ventanas de mando de las válvulas.

#### *Atributo QIntControl\_M\_A:*

El atributo *QIntControl\_M\_A* hace referencia al estado automático o manual de la válvula y se programa desde el diálogo de dinamización de igual forma que en el caso del motor.

## Programación de Puertas y registros

Se han desarrollado dos faceplates para puertas y registros según su orientación, ya que los faceplates no se pueden girar, pero ambos tienen los mismos atributos y su dinamización se programa de la misma forma.



Instancia de faceplate – Puerta o registro

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

### *Atributo Información sobre herramientas:*

El atributo “*Información sobre herramientas*” tiene el mismo código que en el caso de los motores ya que la estructura de variables de las puertas y registros es similar.

### *Atributo BackColor:*

```
Function BackColor_Trigger(ByVal Item)

Dim group
Dim varInfo(4) 'array de variables de informacion
Dim varAlarm(7) 'array de variables de alarmas
Dim i

Set group = HMIRuntime.Tags.CreateTagSet 'se crea un grupo de variables

For i = 1 To 4 'anhado variables de informaciones al grupo
    group.Add Item.ObjectName&".Info2_Informacion"&i
Next

For i=0 To 6 'anhado variables de alarmas al grupo
    group.Add Item.ObjectName&".Info1_Alarma"&i
Next

group.Read 'se leen las variables conjuntamente

'Se leen individualmente las variables
For i = 1 To 4
    varInfo(i)=group(Item.ObjectName&".Info2_Informacion"&i).Value
Next

For i=0 To 6
    varAlarm(i) = group(Item.ObjectName&".Info1_Alarma"&i).Value
    If varAlarm(i)=1 Then
        Item.BackFlashColorOn = RGB(255,255,0) 'amarillo alarma
        Item.BackFlashColorOff = RGB(218,218,218) 'gris fondo
        Item.FlashRateBackColor = 2 'Intermitencia rápida
        Item.FlashBackColor = 1
    End If
Next
```

```

If (varAlarm(0)=0) And (varAlarm(1)=0) And (varAlarm(2)=0) And
(varAlarm(3)=0) And (varAlarm(4)=0) And (varAlarm(5)=0) And (varAlarm(6)=0)
Then
  If varInfo(1)=1 Then 'Abierta
    Item.FlashBackColor = 0
    Item.BackColor = RGB(0,255,0) 'verde
  End If
  If varInfo(2)=1 Then 'Abriendo
    Item.BackFlashColorOn = RGB(0,255,0) 'verde
    Item.BackFlashColorOff = RGB(218,218,218) 'gris fondo
    Item.FlashRateBackColor = 2 'Intermitencia rápida
    Item.FlashBackColor = 1
  End If
  If varInfo(3)=1 Then 'Cerrando
    Item.BackFlashColorOn = RGB(255,0,0) 'rojo
    Item.BackFlashColorOff = RGB(218,218,218) 'gris fondo
    Item.FlashRateBackColor = 2 'Intermitencia rápida
    Item.FlashBackColor = 1
  End If
  If varInfo(4)=1 Then 'Cerrada
    Item.FlashBackColor = 0
    Item.BackColor = RGB(255,0,0) 'rojo
  End If
  If (varInfo(1)=0) And (varInfo(2)=0) And (varInfo(3)=0) And
(varInfo(4)=0) Then
    Item.BackFlashColorOn = RGB(252,255,0) 'amarillo alarma
    Item.BackFlashColorOff = RGB(218,218,218) 'gris fondo
    Item.FlashRateBackColor = 2 'Intermitencia rápida
    Item.FlashBackColor = 1
  End If
End If
End Function

```

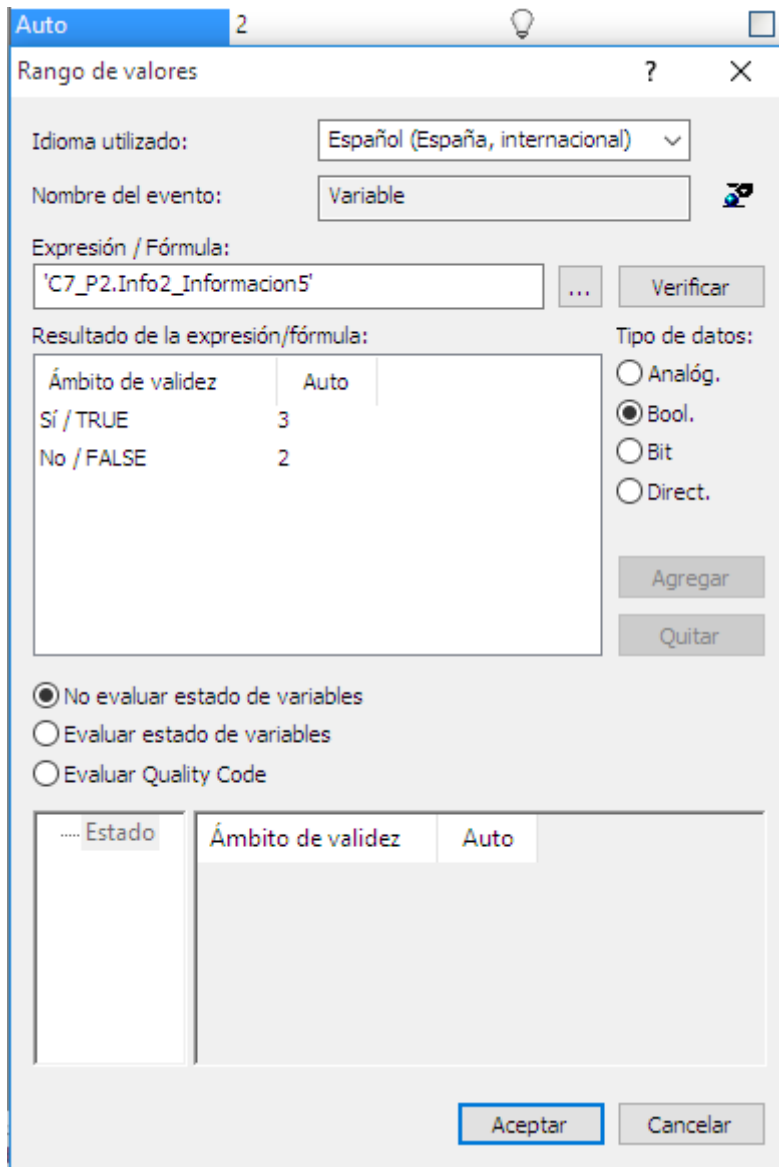
**Triggers de variable:**

NombreObjeto.Info2_Informacion0	Al cambiar
NombreObjeto.Info2_Informacion1	Al cambiar
NombreObjeto.Info2_Informacion2	Al cambiar
NombreObjeto.Info2_Informacion3	Al cambiar
NombreObjeto.Info2_Informacion4	Al cambiar
NombreObjeto.Info1_Alarma0	Al cambiar
NombreObjeto.Info1_Alarma1	Al cambiar
NombreObjeto.Info1_Alarma2	Al cambiar
NombreObjeto.Info1_Alarma3	Al cambiar
NombreObjeto.Info1_Alarma4	Al cambiar
NombreObjeto.Info1_Alarma5	Al cambiar
NombreObjeto.Info1_Alarma6	Al cambiar

Con este script se calcula el color y la intermitencia según los siguientes estados:

- **Abierta:** Color verde sin parpadeo.
- **Abriendo:** Color verde con parpadeo rápido.
- **Cerrando:** Color rojo con parpadeo rápido.
- **Cerrada:** Color rojo sin parpadeo.
- **Alarma:** Color amarillo con parpadeo rápido.

*Atributo Auto:*



El atributo “Auto” hace referencia al estado automático o manual de la puerta o registro y se programa desde el diálogo de dinamización de forma que si la variable binaria “NombreObjeto.Info2\_ Informacion5” vale 1 el motor se encuentra en modo automático y se le asigna un 3 al atributo, en caso contrario se encuentra en modo manual y se le asigna un 2. Cuando la puerta o registro se encuentra en modo manual se representa un cuadrado azul en la esquina superior izquierda y cuando está en modo automático se oculta.

Por último se programa el evento “Hacer clic con el ratón” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaPuertasTDS( lpszPictureName, lpszObjectName );
}
```

Como parámetros de la función se le pasa el nombre de la ventana de imagen donde se encuentra instanciado el objeto y el nombre del propio objeto.

## Programación de Recirculadores

Un recirculador es un tipo de ventilador que insufla y distribuye de forma uniforme el aire caliente en el interior del secadero. Para representar este objeto se ha desarrollado un único faceplate.



Instancia de faceplate – Recirculador

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

*Atributo Información sobre herramientas:*

El atributo “*Información sobre herramientas*” tiene el mismo código que en el caso de los motores ya que la estructura de los recirculadores se declaran en la misma estructura de variables que un motor.

*Atributo BackColor:*

BackColor

Rango de valores

Idioma utilizado: Español (España, internacional)

Nombre del evento: Variable

Expresión / Fórmula: 'M43\_C11.Info2\_Informacion1' Verificar

Resultado de la expresión/fórmula:

Ámbito de validez	Back...
Sí / TRUE	
No / FALSE	

Tipo de datos:

Analóg.  
 Bool.  
 Bit  
 Direct.

No evaluar estado de variables  
 Evaluar estado de variables  
 Evaluar Quality Code

..... Estado

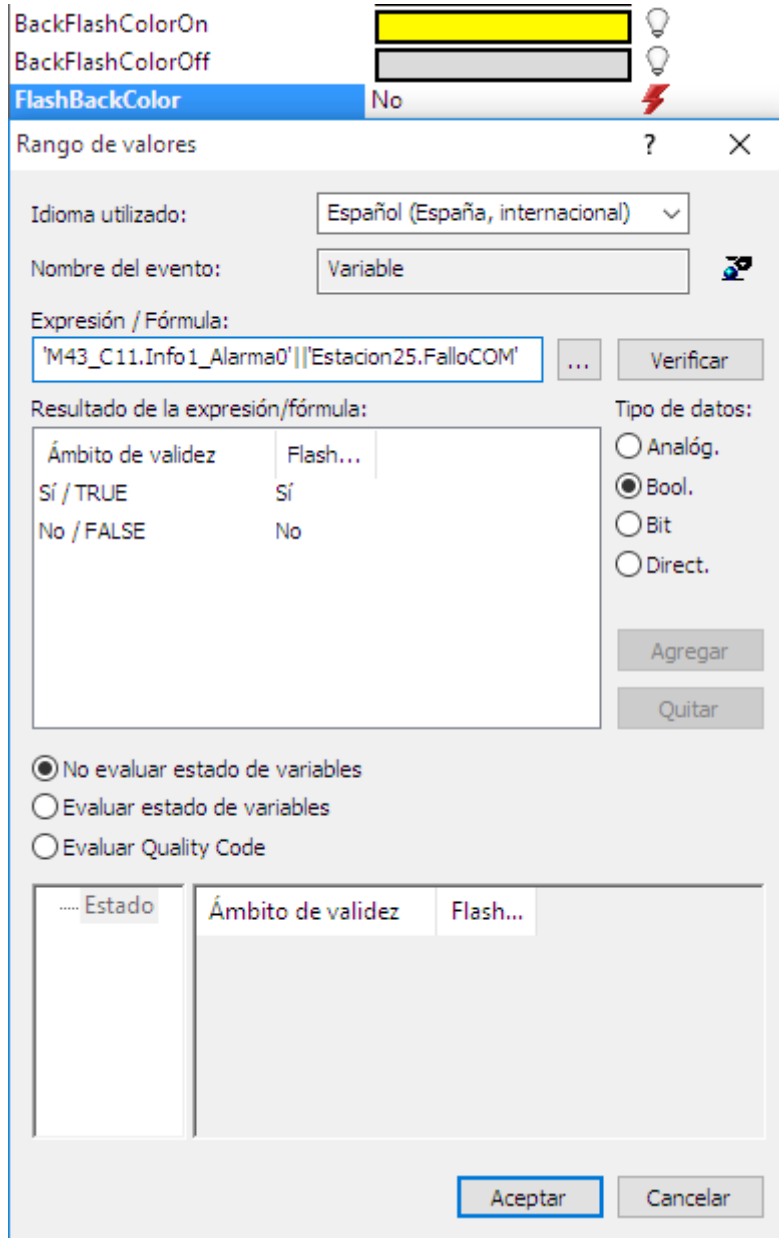
Ámbito de validez	Back...
-------------------	---------

Aceptar Cancelar



El atributo BackColor se programa desde el diálogo de dinamización de forma que si la variable “NombreObjeto.Info2\_Informacion1” que corresponde al estado de Marcha vale 1 el color de fondo será verde, en caso contrario el color de fondo será rojo.

Atributo FlashBackColor:



El atributo “FlashBackColor” toma el valor Sí cuando el recirculador da la alarma de fallo térmico o cuando hay un fallo de comunicación con el PLC, es decir se pone en estado de alarma parpadeando.

Para que parpadee con color amarillo debe seleccionarse de forma estática en los atributos “BackFlashColorOn” y “BackFlashColorOff” los colores amarillo y gris respectivamente.

Atributo Manual\_Auto:

El atributo Manual\_Auto hace referencia al estado automático o manual de la válvula y se programa desde el diálogo de dinamización de igual forma que en el caso del motor.

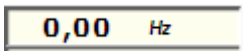
Por último se programa el evento “Hacer clic con el ratón” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaMotoresTDS( lpszPictureName, lpszObjectName ) ;
}
```

Como parámetros de la función se le pasa el nombre de la ventana de imagen donde se encuentra instanciado el objeto y el nombre del propio objeto.

### Programación de Analógicas

Se ha desarrollado un faceplate para representar el valor de una variable analógica junto con su unidad de forma que se anime si se superan los límites establecidos para ella o hay un fallo en la señal.




Instancia de faceplate – Analógica

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

#### Atributo Información sobre herramientas:




Se realiza una asignación directa de la variable de la siguiente forma:

Información sobre herramientas  SP\_M2\_520.Descripcion Al cambiar

El usuario podrá obtener la descripción de la variable analógica cuando sitúe el cursor encima del objeto.

#### Atributos VALOR\_REAL, UNIDAD, ESTADO:

Se realiza una asignación directa de las variables de la siguiente forma:

Atributo	Estático	Dinamización	Actualiz.
VALOR_REAL	0,000000e+000	 SP_M2_520.ValorReal	Al cambiar
UNIDAD	Und	 SP_M2_520.Unidad	Al cambiar
ESTADO	0	 SP_M2_520.Estado	Al cambiar

Dentro del faceplate se dispone de un campo de E/S que actúa como trigger para dinamizar los colores del fondo y el texto. Para ello se dinamiza el evento Valor de salida con el siguiente código:

```
Sub OutputValue_OnPropertyChanged(ByVal Item, ByVal value)

'ALARMAS Analogica
'Bit 0: Alarma L
'Bit 1: Alarma H
'Bit 2: Reserva
'Bit 3: Reserva
'Bit 4: Reserva
'Bit 5: Fallo Señal
'Bit 6: Reserva
'Bit 7: Reserva
```

```

'Bit 8: Reserva
'Bit 9: Reserva
'Bit 10: Reserva
'Bit 11: Reserva
'Bit 12: Reserva
'Bit 13: Reserva
'Bit 14: Reserva
'Bit 15: Reserva

Dim Alm_H, Alm_L, Alm_F
Dim objFondo, objValor, objUnidad
Dim objTexto
Dim colAmaClaro, colVioClaro, colRojo, colAzul,
colOscuro, colGrisOscuro, colGrisClaro

colAmaClaro = RGB (254,254,239)' Color Crema Claro
colVioClaro = RGB (255,153,255) ' Violeta claro
colRojo = RGB (240,0,0)
colAzul = RGB (0,0,240)
colOscuro = RGB (0,0,0) 'Negro
colGrisOscuro = RGB (20,20,20)
colGrisClaro = RGB (100,100,100)

Alm_H = CBool (value And &H00000002)
Alm_L = CBool (value And &H00000001)
Alm_F = CBool (value And &H00000020)

Set ObjFondo = ScreenItems("Fondo")
Set objValor = ScreenItems("Valor")
Set objUnidad = ScreenItems("Unidad")

'Si se dan alarma H y L a la vez se considera fallo:
If ( Alm_H And Alm_L ) Or ( Alm_F )Then
    objValor.ForeFlashColorOn = colOscuro
    objValor.ForeFlashColorOff = colGrisClaro
    objValor.FlashForeColor = True

    objUnidad.ForeFlashColorOn = colOscuro
    objUnidad.ForeFlashColorOff = colGrisClaro
    objUnidad.FlashForeColor = True

    ObjFondo.BackColor = colVioClaro

Elseif ( Alm_H ) Then
    objValor.ForeFlashColorOn = colRojo
    objValor.ForeFlashColorOff = colGrisOscuro
    objValor.FlashForeColor = True

    objUnidad.ForeFlashColorOn = colRojo
    objUnidad.ForeFlashColorOff = colGrisOscuro
    objUnidad.FlashForeColor = True

    ObjFondo.BackColor = colAmaClaro

Elseif ( Alm_L ) Then
    objValor.ForeFlashColorOn = colAzul
    objValor.ForeFlashColorOff = colGrisOscuro
    objValor.FlashForeColor = True

```

```

objUnidad.ForeFlashColorOn = colAzul
objUnidad.ForeFlashColorOff = colGrisOscuro
objUnidad.FlashForeColor = True

ObjFondo.BackColor = colAmaClaro

Else ' estado normal
objUnidad.ForeColor = colOscuro
objValor.ForeColor = colOscuro
objValor.FlashForeColor = False

objUnidad.ForeFlashColorOn = colOscuro
objUnidad.ForeFlashColorOff = colOscuro
objUnidad.FlashForeColor = False

ObjFondo.BackColor = colAmaClaro

End If

End Sub

```

El valor del estado de la analógica se lee en el parámetro *value* de la función del evento, del cual se extraen las alarmas realizando una máscara de bits. En función de las alarmas de estado alto, bajo y fallo señal se calculan los colores para el fondo y el texto como para los estados que se especifican a continuación:

- **Alarma H (alto):** Fondo crema y texto en rojo parpadeando sobre color gris
- **Alarma L (bajo):** Fondo crema y texto en azul parpadeando sobre color gris
- **Fallo señal:** Fondo violeta y texto color negro parpadeando sobre color gris
- **Sin alarma:** Fondo crema y texto en negro.

Por último se programa el evento “Hacer clic con el ratón” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```

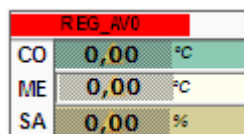
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaAnalogicasTDS( lpszPictureName, lpszObjectName );
}

```

Como parámetros de la función se le pasa el nombre de la ventana de imagen donde se encuentra instanciado el objeto y el nombre del propio objeto.

### Programación de Reguladores

Se ha desarrollado un faceplate para representar los valores de la consigna y salida de los reguladores.



Instancia de faceplate – Regulador

Primero se debe configurar el nombre de objeto para que corresponda con la variable de estructura con la que se ha declarado en la estructura de variables.

Posteriormente se dinamizan los siguientes atributos:

*Atributo Información sobre herramientas:*

```
Function ToolTipText_Trigger(ByVal Item)
'Dinamización del ToolTip en función del nombre del objeto.
Dim TagName, TextPos
Dim Name, Desc

'Formato del nombre del objeto: PrefijodePosicion*NombreTag
TextPos = InStr( 1, Item.ObjectName, "*",vbBinaryCompare )
TagName = Mid ( Item.ObjectName,TextPos+1, Len(Item.ObjectName) )

Name = HMIRuntime.Tags(TagName & ".Nombre").Read
Desc = HMIRuntime.Tags(TagName & ".Descripcion").Read


ToolTipText_Trigger = CStr(Name&" "&Desc)
End Function
```

**Triggers de variable:**

NombreObjeto.Nombre	Al cambiar
NombreObjeto.Descripcion	Al cambiar

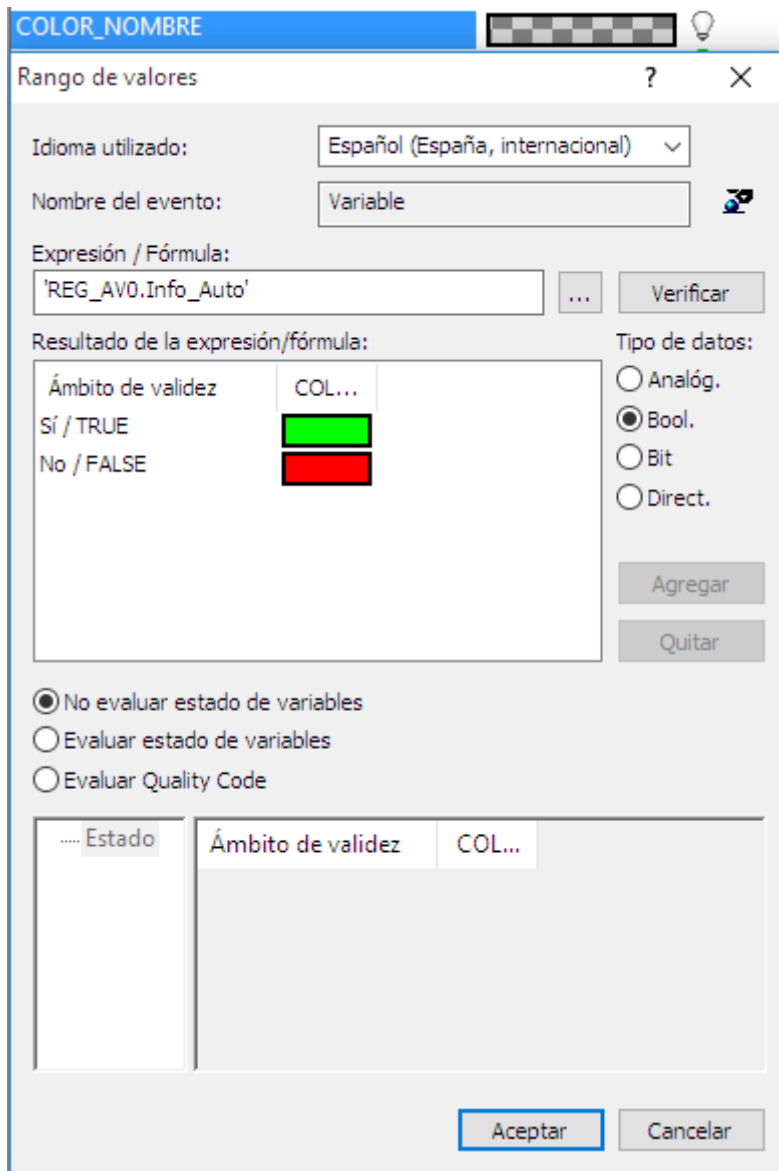
Se dinamiza la información sobre herramientas para que cuando el operario sitúe el cursor del ratón encima del objeto aparezca un tooltip donde se refleja el nombre del regulador y su descripción.

*Atributos NOMBRE, VALOR\_REAL\_CO, UNIDAD\_CO, COLOR\_NOMBRE, UNIDAD\_SA, VALOR\_REAL\_SA:*

Atributo	Estático	Dinamización
<b>NOMBRE</b>	NOMBRE	💡 REG_AV0.Nombre
<b>VALOR_REAL_CO</b>	0,000000e+000	💡 REG_AV0.Valor_con
<b>UNIDAD_CO</b>	Und	💡 REG_AV0.Unidad_con
<b>COLOR_NOMBRE</b>		⚡
<b>UNIDAD_SA</b>	Und	💡 REG_AV0.Unidad_sal
<b>VALOR_REAL_SA</b>	0,000000e+000	💡 REG_AV0.Valor_sal

Los atributos *NOMBRE, VALOR\_REAL\_CO, UNIDAD\_CO, COLOR\_NOMBRE, UNIDAD\_SA, VALOR\_REAL\_SA* se dinamizan mediante la asignación directa de variable para que se muestra el nombre del regulador en la parte superior y se muestren los valores de la consigna y la salida junto con sus unidades en los campos correspondientes a CO y SA respectivamente.

*Atributo COLOR\_NOMBRE:*



El atributo *COLOR\_NOMBRE* se programa desde el diálogo de dinamización de forma que se le asigna el color de fondo verde si el regulador se encuentra en modo automático y rojo si está en modo manual según sea el valor de la variable binaria “*NombreObjeto.Info\_Auto*”.

Por último se programa el evento “*Hacer clic con el ratón*” con el siguiente código para abrir la pantalla de mando correspondiente al objeto:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaAnalogicasTDS( lpszPictureName, lpszObjectName );
}
```

Como parámetros de la función se le pasa el nombre de la ventana de imagen donde se encuentra instanciado el objeto y el nombre del propio objeto.

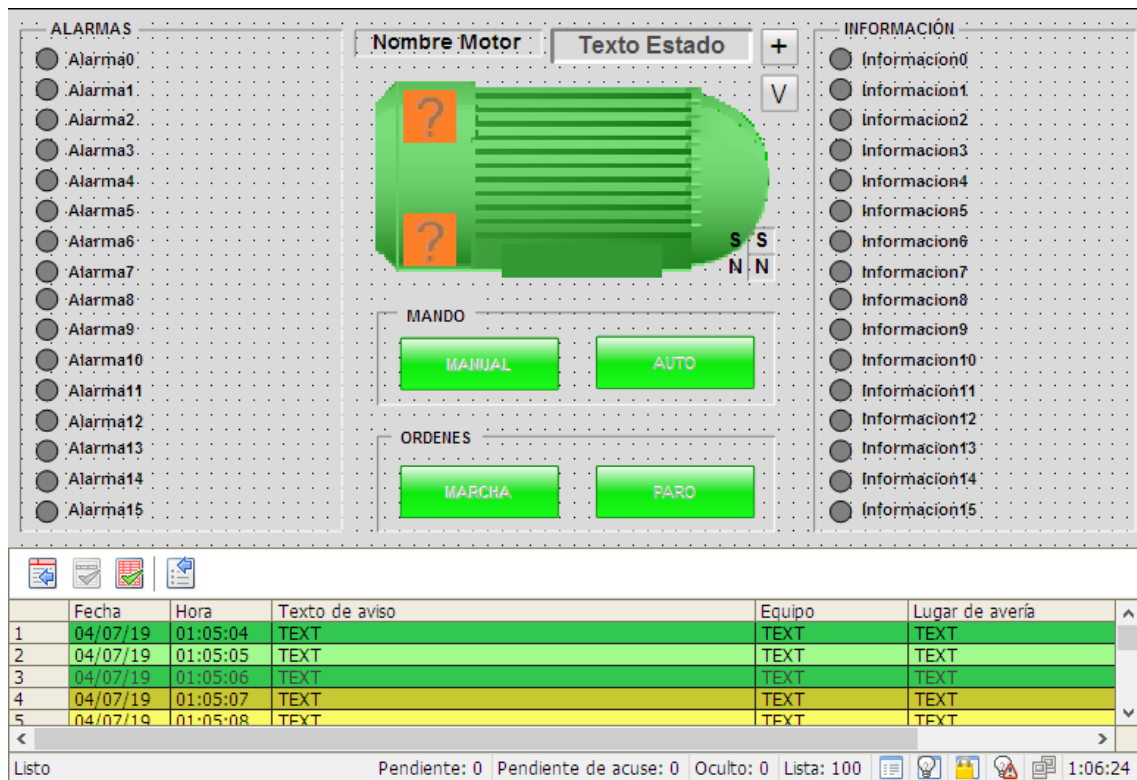
### 2.1.4.4. Programación de las pantallas de mando

Las funciones que se han programado para abrir las pantallas de mando dinamizan el atributo TagPrefix o prefijo de variable de la ventana de imagen con el nombre del objeto, de forma que cada una de las imágenes .PDL de mando se programa usando los elementos de estructura como si fuese una variable, ya que automáticamente cuando produzca una lectura o escritura de una variable se estará concatenando el TagPrefix con el elemento de estructura formando el elemento de variable de estructura para cada objeto.

El uso del TagPrefix permite reciclar tanto el código desarrollado en las pantallas de mando como las propias imágenes de una forma muy eficiente, consiguiendo un sistema muy flexible sobre el que se pueden realizar cambios de forma muy rápido en caso de que el proceso lo requiera.

#### Pantalla de mando de motores

Las pantallas de mando de los motores permiten la visualización de las alamas que puede tener y las informaciones de estado asi como también manejar el modo automático y manual y arranque y paro en caso de que el motor presente mando.



- En la parte izquierda (área de ALARMAS) de la ventana de mando aparecen los textos de las alarmas que se dinamizan de la siguiente forma:

Atributo Texto:

Atributo	Estático	Dinamización
Texto	Alarma0	Alarma0

El atributo Texto se dinamiza mediante asignación directa de la variable correspondiente.

Atributo Visualización:

```
Function Visible_Trigger(ByVal Item)
Dim objText, objTag
```

```

Set objText = ScreenItems("szTagName10")
Set objTag = HMIRuntime.Tags("Alarma0")

```

```
objTag.Read
```

```

If objTag.Value = "Reserva" Then
    objText.Visible = False

```

```

Else
    objText.Visible = True

```

```

End If
End Function

```

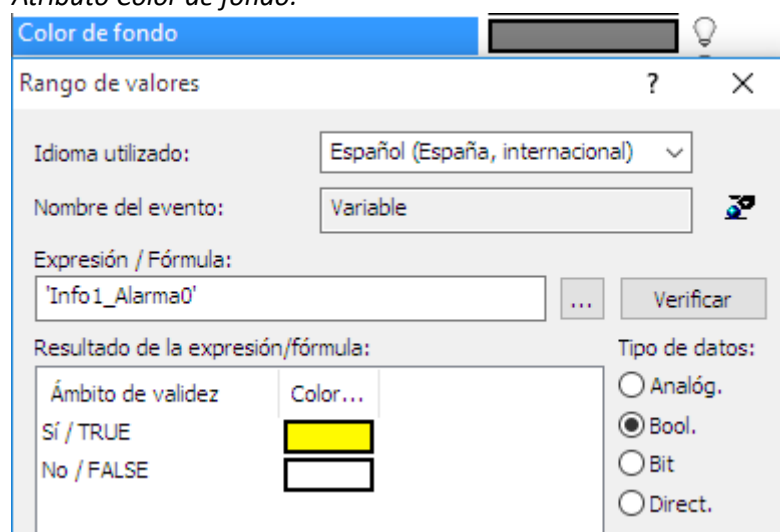
**Trigger de variable:**

Alarma0    Al cambiar

Si la variable Alarma0 contiene un texto distinto a “Reserva” se muestra y en caso contrario se oculta.

- En la parte izquierda (área de ALARMAS) también aparecen los círculos que representan el estado de cada una de las alarmas y se dinamizan de la siguiente forma:

*Atributo Color de fondo:*



El color del círculo se programa mediante el diálogo de dinamización de forma que si el valor de la variable de alarma binaria es 1 se asigna el color de fondo amarillo, en caso contrario el color de fondo será gris para representar el estado de alarma inactivo.

- De igual forma se programa el área de (INFORMACION) situada a la derecha de la pantalla teniendo en cuenta las variables “InformacionX” y “Info2\_Informacion0” que corresponden al texto de la información y al valor respectivamente.
- En la parte inferior de la pantalla se configura un visor de alarmas con nombre “Control1” para que se puedan gestionar y visualizar las alarmas del objeto y acusarlas en caso de ser necesario.



Para que en dicho visor solo aparezcan las alarmas correspondientes al objeto se realiza un filtrado que se ejecuta en el evento “*Seleccionar imagen*” de la propia imagen de mando, de esta forma se ejecutará el siguiente script cada vez que se abra la pantalla de mando y se filtrarán las alarmas para el objeto:

```
#include "apdefap.h"
void OnOpenPicture(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// Se modifica el filtro de las alarmas para que aparezcan solo las de cada motor
char *parentPicture;
char *tagName;
char szMsgFilterSQL[128] = "";

parentPicture = GetParentPicture(lpszPictureName);

//tagName = GetPropChar(parentPicture, "VentanaFaceplate", "TagPrefix");
tagName = GetPropChar(parentPicture, GetParentPictureWindow(lpszPictureName), "TagPrefix");


// printf ("PantallaPadre: %s\r\n", parentPicture );

sprintf (szMsgFilterSQL, "Text3 LIKE '%s'", tagName);
// printf ("SQL Filter: %s\r\n", szMsgFilterSQL);

SetPropChar(lpszPictureName, "Control1", "MsgFilterSQL", szMsgFilterSQL);
}
```

En la consulta SQL que se genera en este Script el parámetro “Text3” hace referencia al campo “Equipo” del AlarmLoggin.

- El campo de texto Nombre Motor que se visualiza en la parte superior se dinamiza mediante la asignación directa de la variable “NombreMotor”

Atributo	Estático	Dinamización
Texto	Nombre Motor	 NombreMotor

- El texto de estado que se visualiza en la parte superior se le dinamizan sus atributos de la siguiente forma

*Atributo Color de fondo:*

```
Function BackColor_Trigger(ByVal Item)
Dim Marcha(2)
Dim Alarma(21)
Dim objGroup, i

Set objGroup= HMIRuntime.Tags.CreateTagSet

For i=0 To 1 'se anaden las variables de informacion al grupo
objGroup.Add "Info2_Informacion"&i
Next

For i=0 To 21 'se anaden las variables de alarmas al grupo
objGroup.Add "Info1_Alarma"&i
Next

objGroup.Read

For i=0 To 1
Marcha(i)=objGroup("Info2_Informacion"&i).Value
Next

'Estado Marcha/Paro/Alarma
```

```

For i=0 To 21
    Alarma(i)=objGroup("Info1_Alarma"&i).Value
    If Alarma(i)=1 Then
        BackColor_Trigger = RGB(255, 255, 0) 'Alarma
    End If
Next

If Alarma(0)=0 And Alarma(1)=0 And Alarma(2)=0 And Alarma(3)=0 And
Alarma(4)=0 And Alarma(5)=0 And Alarma(6)=0 And Alarma(7)=0 And
Alarma(8)=0 And Alarma(9)=0 And Alarma(10)=0 And Alarma(11)=0 And
Alarma(12)=0 And Alarma(13)=0 And Alarma(14)=0 And Alarma(15)=0 And
Alarma(16)=0 And Alarma(17)=0 And Alarma(18)=0 And Alarma(19)=0 And
Alarma(20)=0 And Alarma(21)=0 Then
    If Marcha(1)=1 Then 'bit que indica motor funcionando
        BackColor_Trigger = RGB(0, 255, 0) 'En marcha
    Else
        BackColor_Trigger = RGB(255, 0, 0) 'Parado
    End If
End If
End Function

```

**Trigger de variable:**

Info2_Informacion0	Al cambiar
Info2_Informacion1	Al cambiar
Info2_Informacion2	Al cambiar
Info2_Informacion3	Al cambiar
Info2_Informacion4	Al cambiar
Info2_Informacion5	Al cambiar
Info2_Informacion6	Al cambiar
Info2_Informacion7	Al cambiar
Info2_Informacion8	Al cambiar
Info2_Informacion9	Al cambiar
Info2_Informacion10	Al cambiar
Info2_Informacion11	Al cambiar
Info2_Informacion12	Al cambiar
Info2_Informacion13	Al cambiar
Info2_Informacion14	Al cambiar
Info2_Informacion15	Al cambiar
Info1_Alarma0	Al cambiar
Info1_Alarma1	Al cambiar
Info1_Alarma2	Al cambiar
Info1_Alarma3	Al cambiar
Info1_Alarma4	Al cambiar
Info1_Alarma5	Al cambiar
Info1_Alarma6	Al cambiar
Info1_Alarma7	Al cambiar
Info1_Alarma8	Al cambiar
Info1_Alarma9	Al cambiar
Info1_Alarma10	Al cambiar
Info1_Alarma11	Al cambiar
Info1_Alarma12	Al cambiar
Info1_Alarma13	Al cambiar
Info1_Alarma14	Al cambiar
Info1_Alarma15	Al cambiar
Info1_Alarma16	Al cambiar
Info1_Alarma17	Al cambiar
Info1_Alarma18	Al cambiar
Info1_Alarma19	Al cambiar
Info1_Alarma20	Al cambiar

Info1\_Alarma21

Al cambiar

Con este script se comprueba si el motor tiene alguna de sus alarmas activas, en cuyo caso el color de fondo será amarillo, en caso contrario si el bit de marcha vale 1 el color de fondo será verde, sino significa que el motor está parado y el color de fondo será rojo.

*Atributo Texto:*

```
Function Text_Trigger(ByVal Item)
Dim Marcha(2)
Dim Alarma(21)
Dim objGroup, i

Set objGroup= HMIRuntime.Tags.CreateTagSet

For i=0 To 1 'se anaden las variables de informacion al grupo
    objGroup.Add "Info2_Informacion"&i
Next

For i=0 To 21 'se anaden las variables de alarmas al grupo
    objGroup.Add "Info1_Alarma"&i
Next

objGroup.Read

For i=0 To 1
    Marcha(i)=objGroup("Info2_Informacion"&i).Value
Next

'Estado Marcha/Paro/Alarma
For i=0 To 21
    Alarma(i)=objGroup("Info1_Alarma"&i).Value
    If Alarma(i)=1 Then
        Text_Trigger = "ALARMA"
    End If
Next

If Alarma(0)=0 And Alarma(1)=0 And Alarma(2)=0 And Alarma(3)=0 And
Alarma(4)=0 And Alarma(5)=0 And Alarma(6)=0 And Alarma(7)=0 And
Alarma(8)=0 And Alarma(9)=0 And Alarma(10)=0 And Alarma(11)=0 And
Alarma(12)=0 And Alarma(13)=0 And Alarma(14)=0 And Alarma(15)=0 And
Alarma(16)=0 And Alarma(17)=0 And Alarma(18)=0 And Alarma(19)=0 And
Alarma(20)=0 And Alarma(21)=0 Then
    If Marcha(1)=1 Then 'bit que indica motor funcionando
        Text_Trigger = "EN MARCHA"
    Else
        Text_Trigger = "PARADO"
    End If
End If

End Function
```

**Trigger de variable:**

Info2\_Informacion0

Al cambiar

Info2\_Informacion1

Al cambiar

Info1_Alarma0	Al cambiar
Info1_Alarma1	Al cambiar
Info1_Alarma2	Al cambiar
Info1_Alarma3	Al cambiar
Info1_Alarma4	Al cambiar
Info1_Alarma5	Al cambiar
Info1_Alarma6	Al cambiar
Info1_Alarma7	Al cambiar
Info1_Alarma8	Al cambiar
Info1_Alarma9	Al cambiar
Info1_Alarma10	Al cambiar
Info1_Alarma11	Al cambiar
Info1_Alarma12	Al cambiar
Info1_Alarma13	Al cambiar
Info1_Alarma14	Al cambiar
Info1_Alarma15	Al cambiar
Info1_Alarma16	Al cambiar
Info1_Alarma17	Al cambiar
Info1_Alarma18	Al cambiar
Info1_Alarma19	Al cambiar
Info1_Alarma20	Al cambiar
Info1_Alarma21	Al cambiar

La lógica de este script es similar al explicado en el punto anterior, pero en este caso se asigna el estado del motor al atributo de Texto.

*Atributo Intermitencia de fondo activa:*

```
Function FlashBackColor_Trigger(Byval Item)

Dim Alarma(21)
Dim objGroup, i

Set objGroup= HMIRuntime.Tags.CreateTagSet

For i=0 To 21 'se anhaden las variables de alarmas al grupo
    objGroup.Add "Info1_Alarma"&i
Next

objGroup.Read

'Intermitencia Alarma
For i=0 To 21
    Alarma(i)=objGroup("Info1_Alarma"&i).Value
Next

If Alarma(0)=1 Or Alarma(1)=1 Or Alarma(2)=1 Or Alarma(3)=1 Or
Alarma(4)=1 Or Alarma(5)=1 Or Alarma(6)=1 Or Alarma(7)=1 Or
Alarma(8)=1 Or Alarma(9)=1 Or Alarma(10)=1 Or Alarma(11)=1 Or
Alarma(12)=1 Or Alarma(13)=1 Or Alarma(14)=1 Or Alarma(15)=1 Or
Alarma(16)=1 Or Alarma(17)=1 Or Alarma(18)=1 Or Alarma(19)=1 Or
Alarma(20)=1 Or Alarma(21)=1 Then
    FlashBackColor_Trigger = True
Else
    FlashBackColor_Trigger = False
End If
```

## End Function

### Trigger de variable:

Info1_Alarma0	Al cambiar
Info1_Alarma1	Al cambiar
Info1_Alarma2	Al cambiar
Info1_Alarma3	Al cambiar
Info1_Alarma4	Al cambiar
Info1_Alarma5	Al cambiar
Info1_Alarma6	Al cambiar
Info1_Alarma7	Al cambiar
Info1_Alarma8	Al cambiar
Info1_Alarma9	Al cambiar
Info1_Alarma10	Al cambiar
Info1_Alarma11	Al cambiar
Info1_Alarma12	Al cambiar
Info1_Alarma13	Al cambiar
Info1_Alarma14	Al cambiar
Info1_Alarma15	Al cambiar
Info1_Alarma16	Al cambiar
Info1_Alarma17	Al cambiar
Info1_Alarma18	Al cambiar
Info1_Alarma19	Al cambiar
Info1_Alarma20	Al cambiar
Info1_Alarma21	Al cambiar

Este script comprueba si el motor tiene alguna de sus alarmas activa, en cuyo caso activa la intermitencia de fondo del campo de texto, en caso contrario la intermitencia de fondo permanecerá desactivada.

- Código evento "Hacer clic con el ratón" del botón ampliar/reducir pantalla:

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    long int offset;
    char* ventanaBase;
    char* ventanaImagen;
    double gris, azul;

    ventanaBase = GetParentPicture(lpszPictureName); //Return-Type: char*
    ventanaImagen= GetParentPictureWindow(lpszPictureName); //Return-Type: char*

    // Color azul RGB(150,200,255)
    azul=150 + (pow(2, 8)) * 200 + (pow(2, 16)) * 255 + (pow(2, 24)) * 1;
    // Color gris RGB(192,192,192)
    gris=192 + (pow(2, 8)) * 192 + (pow(2, 16)) * 192 + (pow(2, 24)) * 1;

    //obtenemos el offset de la imagen

    offset = GetOffsetLeft(ventanaBase, ventanaImagen);

    if(offset!=0){ //Si la imagen no está en 0
        SetOffsetLeft(ventanaBase,ventanaImagen, 0); //Offset
        SetPropWord(ventanaBase,ventanaImagen,"Width",869); //Ancho
        SetPropWord(ventanaBase,ventanaImagen,"Height",623); //Alto
        SetPropDouble(lpszPictureName,lpszObjectName,"BackColor",azul); //Return-
Type: BOOL
        SetPropChar(lpszPictureName ,lpszObjectName,"Text","-");
    }else{
        SetOffsetLeft(ventanaBase, ventanaImagen, 260);
        SetPropWord(ventanaBase,ventanaImagen,"Width",350); //Ancho
        SetPropWord(ventanaBase,ventanaImagen,"Height",440); //Alto
        SetPropDouble(lpszPictureName,lpszObjectName,"BackColor",gris); //Return-
Type: BOOL
    }
```

```

        SetPropChar(lpszPictureName ,lpszObjectName,"Text","+");
    }
}

```

Cuando se abre la pantalla de mando al hacer clic con el ratón encima de una instancia de faceplate de tipo motor, esta aparece en “modo reducido” de forma que solo se muestra la parte central donde se refleja el motor y los botones para su manejo, porque en la mayoría de ocasiones el operador no necesita que se le muestre más información. Pero en ocasiones se necesitan consultar las alarmas e informaciones del motor, para ello se dispone en la parte superior derecha un botón [+] que permite ampliar y reducir la ventana de mando para poder visualizar toda la información, para ello se usa el script que se refleja anteriormente en el evento “Hacer clic con el ratón” del botón.

Este script comprueba el offset de la ventana de imagen para saber si está en modo reducido o ampliado y en función de ello dinamiza los atributos de las dimensiones de la ventana de imagen.

- Código evento “Hacer clic con el ratón” del botón (V) variador:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    ventanaVariadoresTDS(lpszPictureName, GetTagChar("NombreMotor")); //Ventana de los
    variadores
}

```

En la parte superior derecha también se dispone de un botón que permite la abrir en una nueva ventana de imagen la imagen de proceso que permite visualizar la información del variador del motor en caso de tenerlo, sinó el botón permanecerá oculto lo que indica al usuario que dicho motor no dispone de variador de velocidad.

- Código evento “Hacer clic con el ratón” del botón de mando MANUAL:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    SetTagBit("OPManual",1); //Return-Type: BOOL
}

```

- Código evento “Hacer clic con el ratón” del botón de mando AUTO:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    SetTagBit("OPAuto",1); //Return-Type: BOOL
}

```

- Código evento “Hacer clic con el ratón” del botón de orden MARCHA:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    SetTagBit("OPMarcha",1); //Return-Type: BOOL
}

```

- Código evento “Hacer clic con el ratón” del botón de orden PARO:

```

#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
    SetTagBit("OPParo",1); //Return-Type: BOOL
}

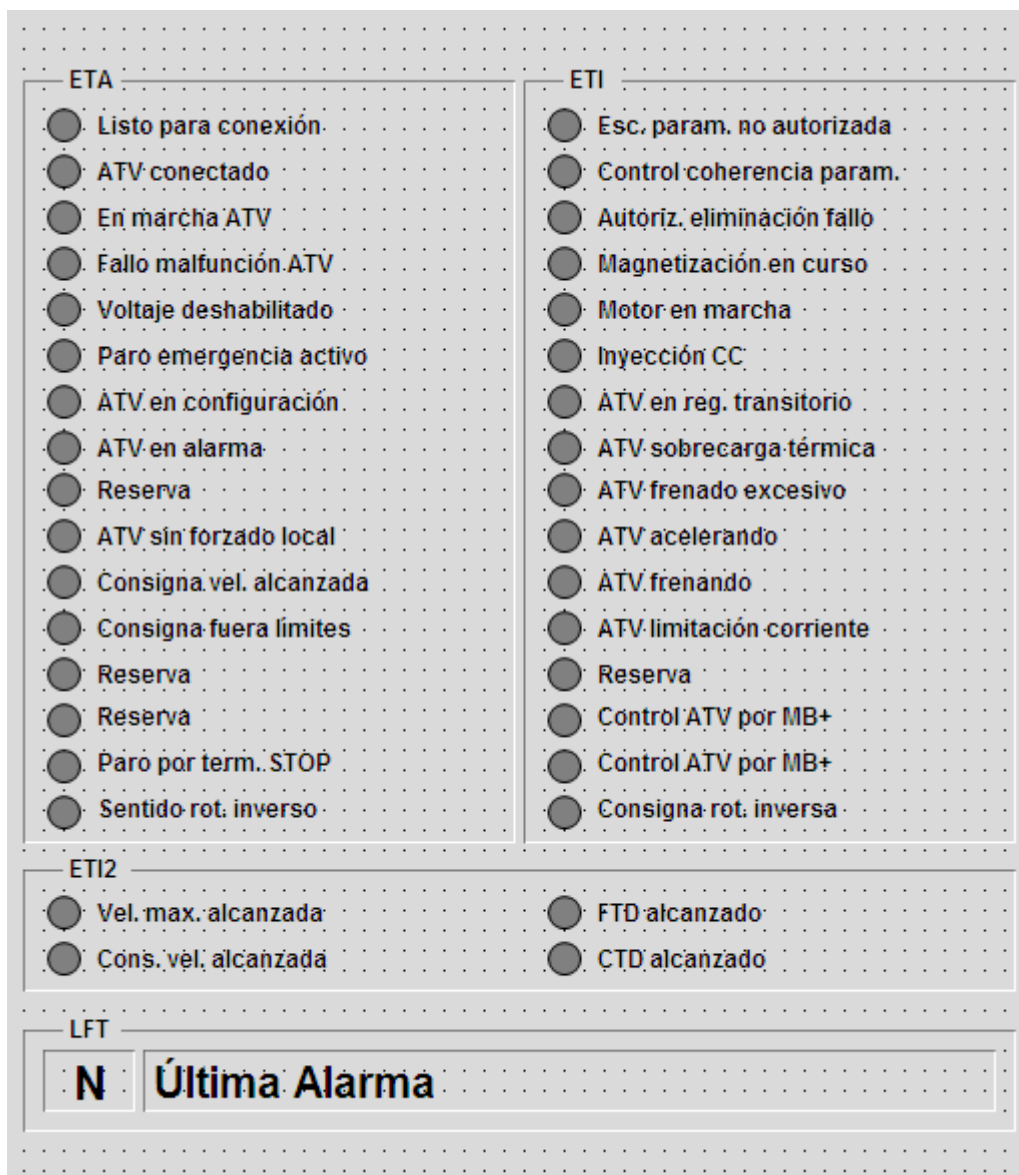
```

La función *SetTagBit()* escribe en la variable de operación del motor el valor booleano que se le configure.

## Pantalla de visualización de variadores

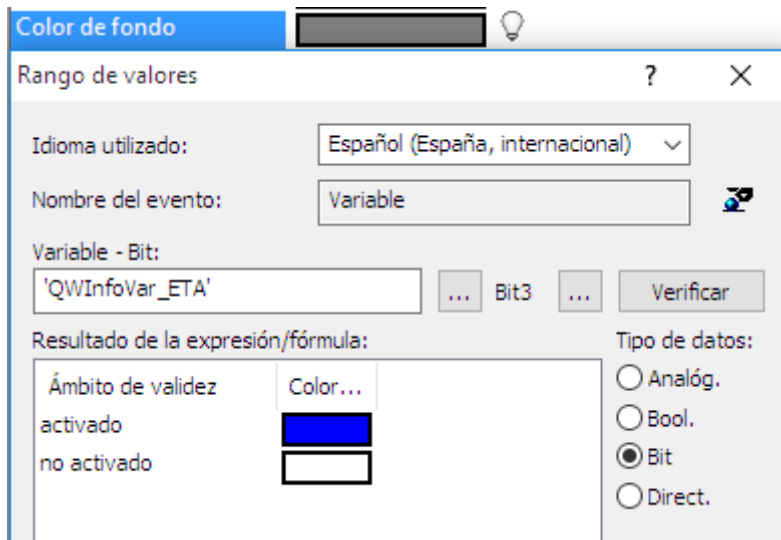
La pantalla de visualización de los variadores está dividida en cuatro áreas:

- **Área ETA:** se reflejan las alarmas.
- **Área ETI:** se reflejan las informaciones generales.
- **Área ETI2:** se reflejan las informaciones adicionales.
- **Área LFT:** se refleja la última alarma que ha tenido el variador junto con su código de error.



Como todos los variadores de la planta son iguales los textos de las alarmas e informaciones se configuran estáticamente y se dinamizan los círculos correspondientes a cada una de las alarmas e informaciones de la siguiente forma desde el diálogo de dinamización:

*Atributo Color de fondo:*



Si el bit correspondiente a la alarma está activado el color de fondo del círculo correspondiente es azul, en caso contrario será blanco para representar el estado inactivo.

La dinamización para los círculos de las áeras ET1 y ET12 se realiza de igual forma accediendo a los bits de las palabras "QWInfoVar\_ET1" y "QWInfoVar\_ET12" respectivamente.

En la parte inferior de la imagen se refleja el área LFT que está compuesta por el campo de texto correspondiente al número de error y el que corresponde a la descripción de la última alarma.

- La dinamización del campo de texto correspondiente al número de error se realiza mediante la asignación directa de la variable "QIntInfoVar\_LFT" del variador, tal y como se muestra a continuación.

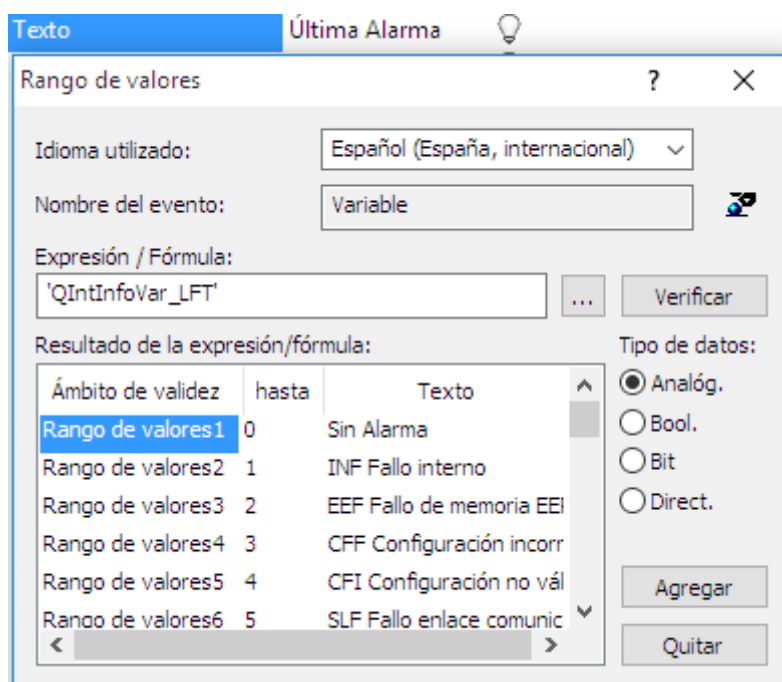
*Atributo Texto:*

Atributo	Estático	Dinamización
Texto	N	QIntInfoVar_LFT

- La dinamización del campo de texto correspondiente a la descripción de la alarma se realiza mediante el diálogo de dinamización tal y como se muestra a continuación.

*Atributo Texto:*

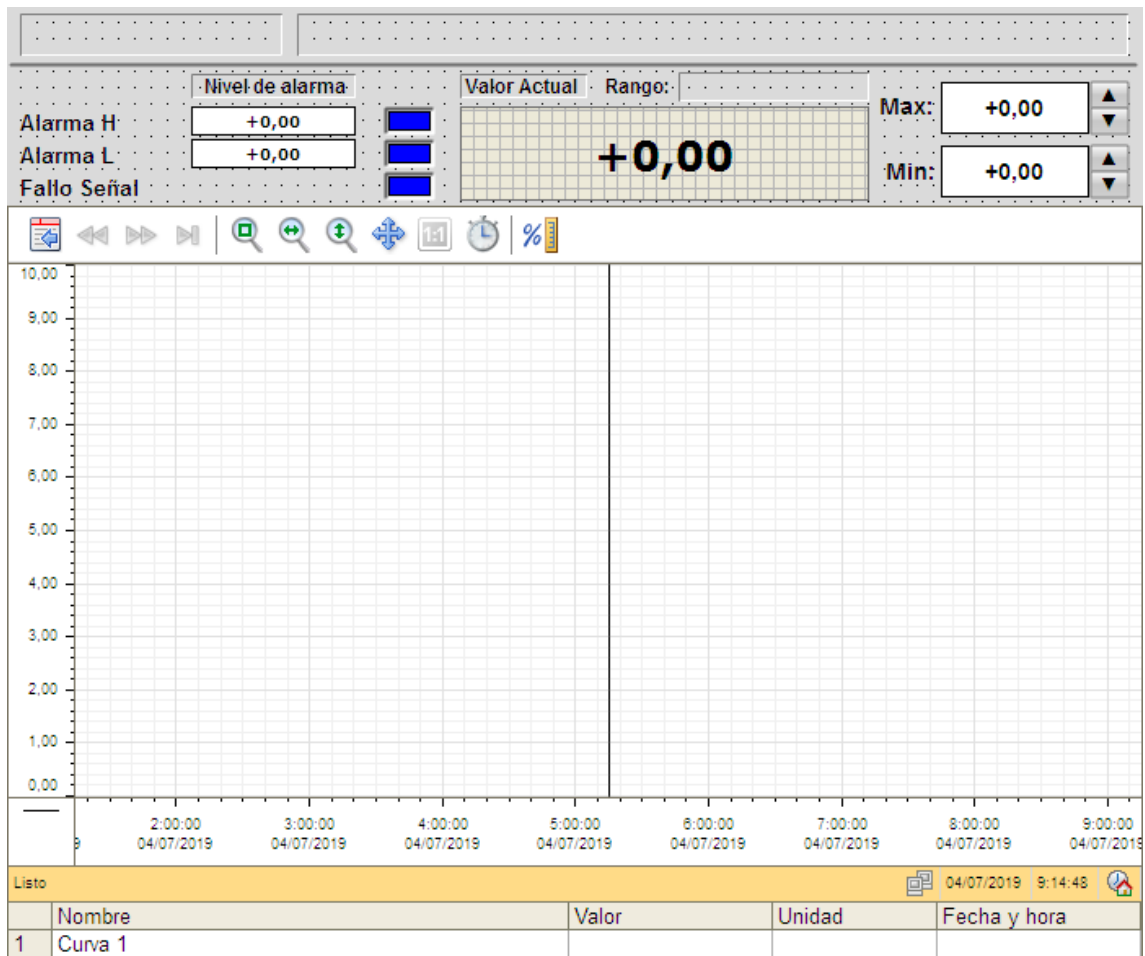




Se crea una lista de textos que corresponden con el código de error del variador que transmite por la palabra "QIntInfoVar\_LFT".

### Pantalla de mando de analógicas

La pantalla de mando de las analógicas permite la representación gráfica del valor histórico de la variable analógica, visualizar el valor actual que se está leyendo, visualizar el estado de las alarmas que presenta, establecer los límites para las alarmas y configurar el rango de visualización de la gráfica.



Esta pantalla presenta los siguientes campos de texto con dinamización:

- **NombreAnalogica**

Atributo texto:

Atributo	Estático	Dinamización	Actualiz.
Texto		Nombre	Al cambiar

Se dinamiza mediante la asignación directa de la variable "Nombre"

- **DescripciónAnalogica**

Atributo Texto:

Atributo	Estático	Dinamización	Actualiz.
Texto		Descripcion	Al cambiar

Se dinamiza mediante la asignación directa de la variable "Descripcion"

- **Texto\_rango**

Atributo Texto:

```
Function Text_Trigger(ByVal Item)
Text_Trigger = HMIRuntime.Tags("Anamin").Read & " a " &
HMIRuntime.Tags("Anamax").Read
End Function
```

**Trigger de variable:**

Anamax	Al cambiar
Anamin	Al cambiar

Con este script se concatena el valor definido en la variable de estructura “Anamin” con “Anamax” para formar el texto que representa el rango de medida de la sonda analógica.

- **Unidad\_rango**

*Atributo Texto:*

Atributo	Estático	Dinamización	Actualiz.
Texto		Unidad	Al cambiar

Se dinamiza mediante la asignación directa de la variable “Unidad”

La pantalla presenta los siguientes campos de entrada/salida (E/S) que permiten la visualización y manipulación de los valores configurables para la analógica:

- **CampoValorActual**

En un campo configurado como solo salida que representa el valor actual que está leyendo la sonda analógica. Su dinamización se realiza como se muestra a continuación:

*Atributo Valor de salida:*

Valor de salida	0,000000e+000	ValorReal	Al cambiar
-----------------	---------------	-----------	------------

Se dinamiza mediante la asignación directa de la variable “ValorReal”

- **Alarma H**

El campo de E/S correspondiente al nivel alto de alarma refleja el valor de la desviación que se permite para que se produzca la alarma de nivel alto. Este campo se dinamiza como se muestra a continuación:

*Atributo Valor de salida:*

Valor de salida	0,000000e+000	Ana_H	Al cambiar
-----------------	---------------	-------	------------

Se dinamiza mediante la asignación directa de la variable “Ana\_H”

- **Alarma L**

El campo de E/S correspondiente al nivel bajo de alarma refleja el valor de la desviación que se permite para que se produzca la alarma de nivel bajo. Este campo se dinamiza como se muestra a continuación:

*Atributo Valor de salida:*


Valor de salida	0,000000e+000	Ana_L	Al cambiar
-----------------	---------------	-------	------------

Se dinamiza mediante la asignación directa de la variable “Ana\_L”

- **CampoRgMax**

El campo de E/S correspondiente al rango máximo refleja el valor máximo del eje de valores de la gráfica de representación. Este campo se dinamiza como se muestra a continuación:

Atributo Valor de salida:

Valor de salida 0,000000e+000  Rangomax Al cambiar


Se dinamiza mediante la asignación directa de la variable "Rangomax"

Si el operador introduce un valor manualmente en este campo se modifica el valor máximo de representación del eje de valores de la gráfica.

- **CampoRgMin**

En el campo de E/S correspondiente al rango mínimo se visualiza el valor mínimo del eje de valores de la gráfica de representación. Este campo se dinamiza como se muestra a continuación:

Atributo Valor de salida:

Valor de salida 0,000000e+000  Rangomin Al cambiar

Se dinamiza mediante la asignación directa de la variable "Rangomin"

Si el operador introduce un valor manualmente en este campo se modifica el valor mínimo de representación del eje de valores de la gráfica.

Para modificar los rangos máximos y mínimos de representación también se dispone de unos botones situados en el margen derecho de cada campo donde se visualiza el valor, que permiten incrementar o decrementar dichos valor en intervalos de 10 unidades para ajustar la visualización de la gráfica de una forma sencilla.

A continuación se detalla la programación de los botones de modificación de rango:

- **BtnRGMaxUp**

Corresponde al botón para incrementar el valor máximo de representación de la gráfica. Se programa la siguiente acción que se ejecuta cuando ocurre el evento "Hacer clic con el ratón":

```
Sub OnClick(ByVal Item)
Dim aux
aux= HMIRuntime.Tags("Rangomax").Read

aux=aux+10

HMIRuntime.Tags("Rangomax").Write aux
End Sub
```

Cada vez que se pulsa el botón se incrementa el valor de la variable "Rangomax" en 10 unidades.

- **BtnRGMaxDown**

Corresponde al botón para decrementar el valor máximo de representación de la gráfica. Se programa la siguiente acción que se ejecuta cuando ocurre el evento "Hacer clic con el ratón":

```
Sub OnClick(ByVal Item)
Dim aux
aux= HMIRuntime.Tags("Rangomax").Read
```

```

aux=aux-10

HMIRuntime.Tags("Rangomax").Write aux
End Sub

```

Cada vez que se pulsa el botón se decrementa el valor de la variable "Rangomax" en 10 unidades.

- **BtnRGMinUp**

Corresponde al botón para incrementar el valor mínimo de representación de la gráfica. Se programa la siguiente acción que se ejecuta cuando ocurre el evento "Hacer clic con el ratón":

```

Sub OnClick(ByVal Item)
Dim aux
aux= HMIRuntime.Tags("Rangomin").Read

aux=aux+10

HMIRuntime.Tags("Rangomin").Write aux
End Sub

```

Cada vez que se pulsa el botón se incrementa el valor de la variable "Rangomin" en 10 unidades.

- **BtnRGMinDown**

Corresponde al botón para decrementar el valor mínimo de representación de la gráfica. Se programa la siguiente acción que se ejecuta cuando ocurre el evento "Hacer clic con el ratón":

```

Sub OnClick(ByVal Item)
Dim aux
aux= HMIRuntime.Tags("Rangomin").Read

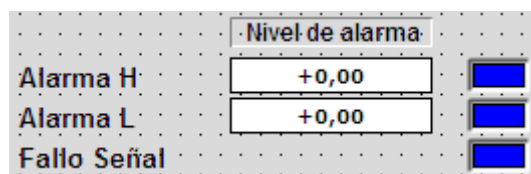
aux=aux-10

HMIRuntime.Tags("Rangomin").Write aux
End Sub

```

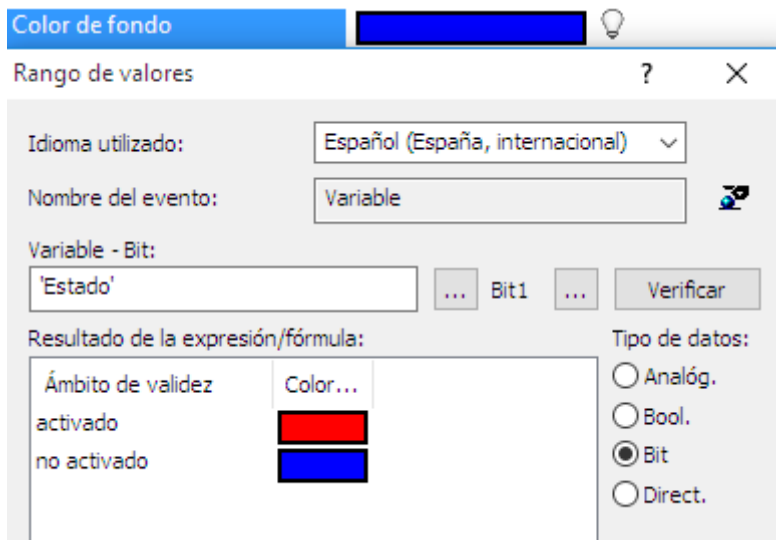
Cada vez que se pulsa el botón se decrementa el valor de la variable "Rangomin" en 10 unidades.

En esta pantalla también se dispone de unos cuadros situados en el área de alarma que indican cuando ocurre una alarma en la analógica para saber si se encuentra fuera del rango que se define o ocurre un fallo de señal.



El cuadro correspondiente a la "Alarma H" se programa con el diálogo de dinamización tal y como se explica a continuación:

*Atributo Color de fondo:*



Se comprueba si el bit 1 de la palabra de estado de la analógica que corresponde a la variable “Estado” se encuentra activo, en cuyo caso se asignará el color rojo como color de fondo del cuadro para indicar que la “Alarma H” está activa, en caso contrario el color de fondo será azul para indicar que no hay alarma de estado alto.

De igual forma se programan los cuadros correspondientes a la “Alarma L” y fallo señal teniendo en cuenta la correspondencia de los bits de la palabra con su alarma, tal y como se especifica a continuación:

Tipo de Alarma o Fallo	Bit de la palabra “Estado”
Alarma H	1
Alarma L	0
Fallo señal	5

Por último, en esta pantalla se dispone de un control de gráficas que permite visualizar la traza de la analógica formada por sus valores históricos de proceso. Se programan los siguientes atributos mediante la asignación directa de variables tal y como se muestra a continuación:

ValueAxisBeginValue	0,000000e+000	💡 Rangomin	Al cambiar
ValueAxisEndValue	10,000000	💡 Rangomax	Al cambiar
TrendLowerLimitColor		💡	
TrendLowerLimit	0,000000e+000	💡 Ana_L	Al cambiar
TrendUpperLimitColor		💡	
TrendUpperLimit	1,000000	💡 Ana_H	Al cambiar
TrendUncertainColor		💡	
TrendTagName		💡 Archivo_TagLogging	Al cambiar
TrendRename	Curva 1	💡 Descripcion	Al cambiar

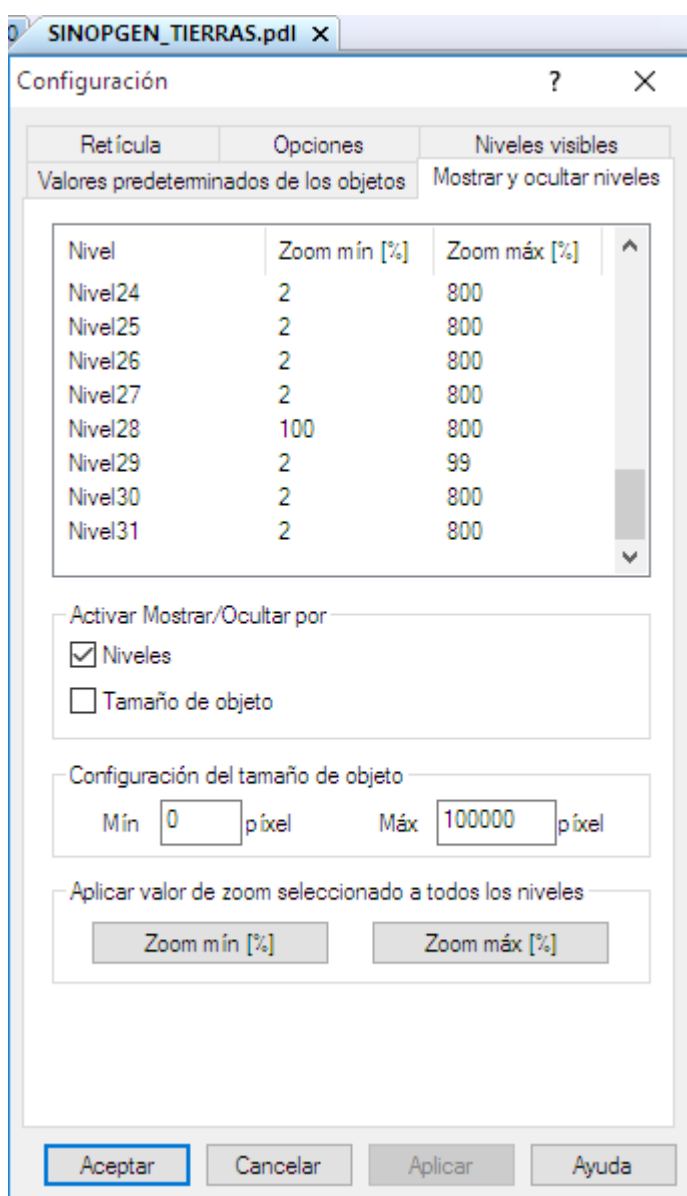
Además de establecer el rango y los límites de representación en los atributos correspondientes se configura de forma estática el color azul en el atributo “TrendLowerLimitColor” para que la traza se represente en dicho color cuando el valor de proceso sea inferior al valor mínimo establecido en el atributo “TrendLowerLimit” que tomará el valor de la variable “Ana\_L”. De forma equivalente se realiza la dinamización para cuando el valor de proceso sobrepasa el límite máximo establecido asignando el color rojo a la traza, y en caso de que el valor de proceso sea incierto se representará en color violeta para indicar que ocurre un fallo al obtener la señal.

Esta forma de representar la traza es de gran utilidad para el operador, ya que de un vistazo puede saber si el valor de proceso se ha desviado de sus límites en el intervalo de tiempo que desee consultar.

### 2.1.5. Estándar decluttering y niveles de zoom en sinóptico PLANTA TIERRAS

El decluttering es una técnica de zoom que permite mostrar y ocultar niveles y los objetos que estos contienen. Los valores límite para cada uno de los niveles pueden finirse para cada una de las ventanas de imagen en “Herramientas -> Preferencias” y la ficha “Mostrar/Ocultar” de Graphics Designer.

En este proyecto se ha empleado esta técnica de zoom en el sinóptico de proceso SINOPGEN\_TIERRAS.Pdl, que corresponde a la planta de tierras donde se realiza la molienda, para el cual se configuran los siguientes valores límite:

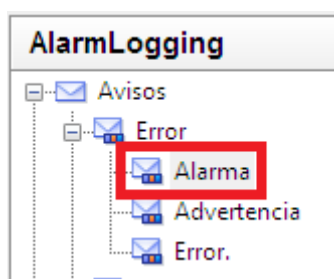


Todos los niveles se visualizan en todo el rango de 2% de zoom hasta el máximo que es 800% excepto el nivel 28 que solo se visualiza cuando el nivel de zoom está entre el 100% y el 800% y el nivel 29 que solo se visualiza cuando el nivel de se encuentra entre 0 y 99%.

Con esta técnica se logra generar distintos niveles de detalle en la visualización de la planta empleando solo un sinóptico de proceso de forma que en la vista general de la planta de tierras, donde el zoom será menor del 99%, se sitúan los objetos que se consideren más importantes en el nivel 29 para que cuando se amplie el zoom hacia una de las zonas estos objetos se oculten y se muestren los que están en el nivel 28 que representan los detalles de las máquinas de la planta tales como motores más pequeños, consumos de motores, analógicas, etc...

## 2.1.6. Archivos de alarmas

Las alarmas del sistema se configuran en el Alarm Logging categoría Error y tipo de aviso Alarma.



Avisos [Alarma]								
	Número	Variable de aviso	Bit de aviso	Grupo de avisos	Texto de aviso	Lugar de avería (ESP)	Equipo	Planta
1	25001	M1_290.QWInfo1	0	MOTORES	M1_290 - Fallo motor	MOLIENDA	M1_290.	PLANTA DE TIERRAS
2	25002	M1_290.QWInfo1	1	MOTORES	M1_290 - Fallo térmico	MOLIENDA	M1_290.	PLANTA DE TIERRAS
3	25003	M1_290.QWInfo1	2	MOTORES	M1_290 - Fallo CDR	MOLIENDA	M1_290.	PLANTA DE TIERRAS
4	25004	M1_290.QWInfo1	3	MOTORES	M1_290 - Fallo potencia	MOLIENDA	M1_290.	PLANTA DE TIERRAS
5	25005	M1_290.QWInfo1	4	MOTORES	M1_290 - Fallo variador	MOLIENDA	M1_290.	PLANTA DE TIERRAS

Para cada una de las variables de Alarma se le configuran los siguientes campos para que posteriormente pueda filtrarse correctamente y aparezca reflejada en los alarmeros individuales para cada objeto:

**Variable de aviso:** Corresponde a la variable que contiene la información de la alarma activa o inactiva.

**Bit de aviso:** En caso de ser una palabra se hace referencia al bit correspondiente a la alarma en cuestión, si es un valor booleano el Bit de aviso es 0.

**Grupo de avisos:** hace referencia a la categoría a la que pertenece el objeto.

**Texto de aviso:** Se configura el texto que queremos que se refleje en el visor de alarmas.

**Lugar de avería:** Corresponde a la zona donde se encuentra situado el objeto real físicamente dentro de las instalaciones. Se requiere para poder filtrar posteriormente por zonas dentro de la planta

**Equipo:** Hace referencia a la designación del equipo y es necesaria su configuración para que se produzca correctamente el filtrado automático dentro de cada una de las pantallas de mando.

**Planta:** Corresponde a cada línea de producción donde se encuentra instalado el dispositivo físicamente (PLANTA DE TIERRAS, FÁBRICA 2 o FÁBRICA 3).

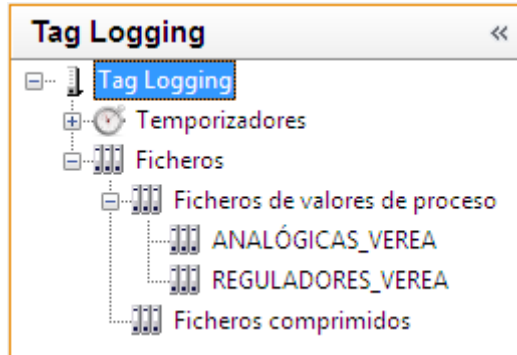


## 2.1.7. Archivo de variables

Los archivos de variables se crean en el TagLogging de la siguiente forma:

En el árbol del TagLogging, dentro de Ficheros -> Ficheros de valores de proceso se crean las dos categorías que corresponden con los valores de proceso para cada uno de los distintos elementos de los que se dispone en el proyecto:

- **ANALÓGICAS\_VEREA:** corresponde a los valores de las señales analógicas que se representan en los sinópticos de proceso.
- **REGULADORES\_VEREA:** como su propio nombre indica corresponde a los valores de consigna y salida de los reguladores que se representan en las pantallas de los sinópticos de proceso.



Dentro de cada una de las secciones se añaden las variables de fichero correspondientes de la siguiente forma:

Ficheros [ ANALÓGICAS_VEREA ]										Buscar
	Variable de proceso	Tipo de variable	Nombre de variable	Nombre del fichero	Comentario	Tipo de adquisición	Suministro de variables	Archivación bloqueada	Además en variable	Ciclo de adquisición
1	34.ValorReal	Analógico	34.ValorReal	ANALÓGICAS_VEREA		cíclico-continuo	Sistema	<input type="checkbox"/>		1 minute
2	35.ValorReal	Analógico	35.ValorReal	ANALÓGICAS_VEREA		cíclico-continuo	Sistema	<input type="checkbox"/>		1 minute
3	36.ValorReal	Analógico	36.ValorReal	ANALÓGICAS_VEREA		cíclico-continuo	Sistema	<input type="checkbox"/>		1 minute
4	60.ValorReal	Analógico	60.ValorReal	ANALÓGICAS_VEREA		cíclico-continuo	Sistema	<input type="checkbox"/>		1 minute
5	61.ValorReal	Analógico	61.ValorReal	ANALÓGICAS_VEREA		cíclico-continuo	Sistema	<input type="checkbox"/>		1 minute

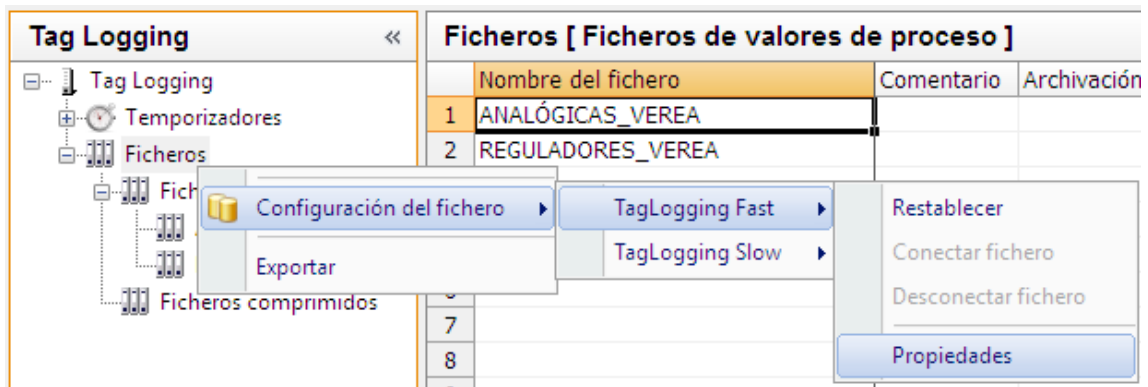
Se configura un ciclo de adquisición de 1 minuto para todas las variables de proceso.

### Configuración del fichero de valores de proceso

Se disponen de los dos tipos de fichero que se especifican a continuación para la configuración de fichero:

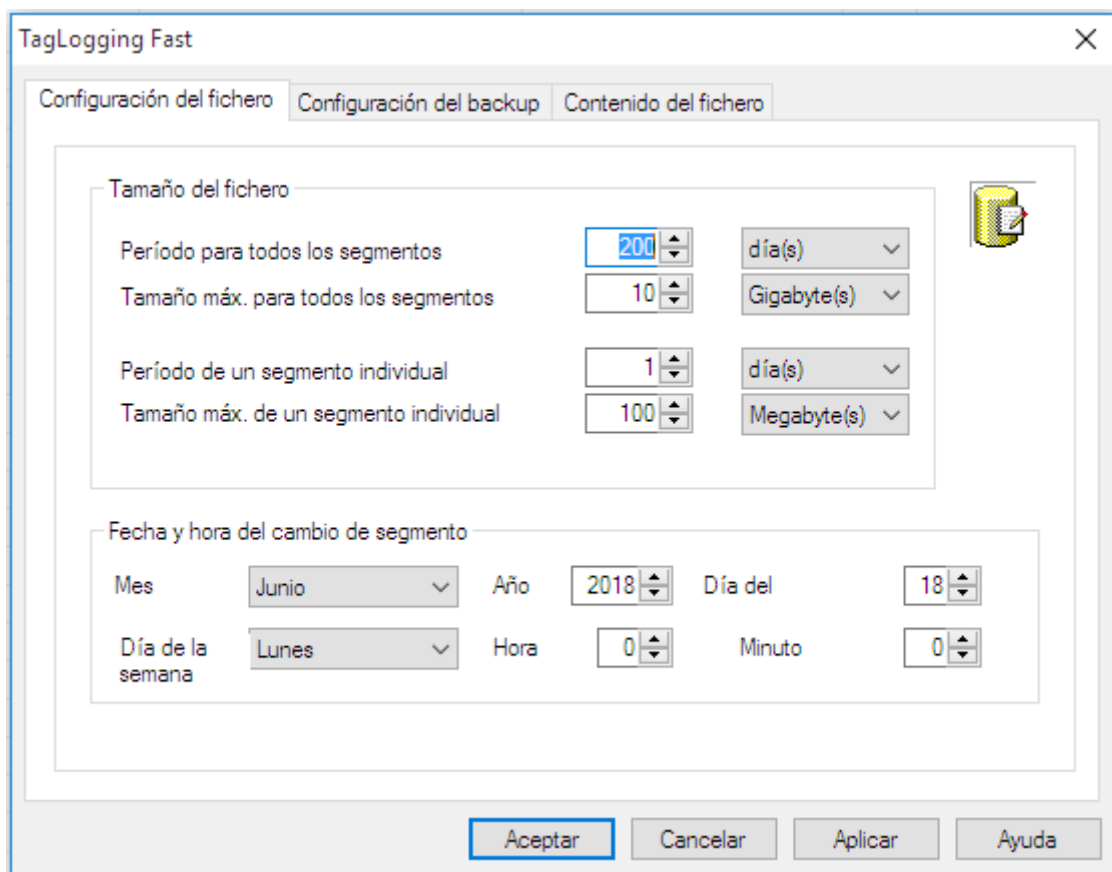
- **TagLogging Fast:** las variables que tienen un tiempo de ciclo de adquisición inferior o igual al que se configure se archivan en esta área. Por defecto el tiempo de ciclo de adquisición configurado es de un minuto.
- **TagLogging Slow:** es el área donde se archivan las variables que tienen un tiempo de ciclo de adquisición superior al que se configure en el TagLogging Fast.

La configuración del fichero es necesaria para gestionar el traslado y almacenamiento de los datos de los valores de proceso de forma correcta y consistente. Para ello se hace clic derecho encima de "Fichero" y se selecciona Configuración del Fichero -> TagLogging Fast o Slow -> Propiedades.



### Configuración TagLogging Fast

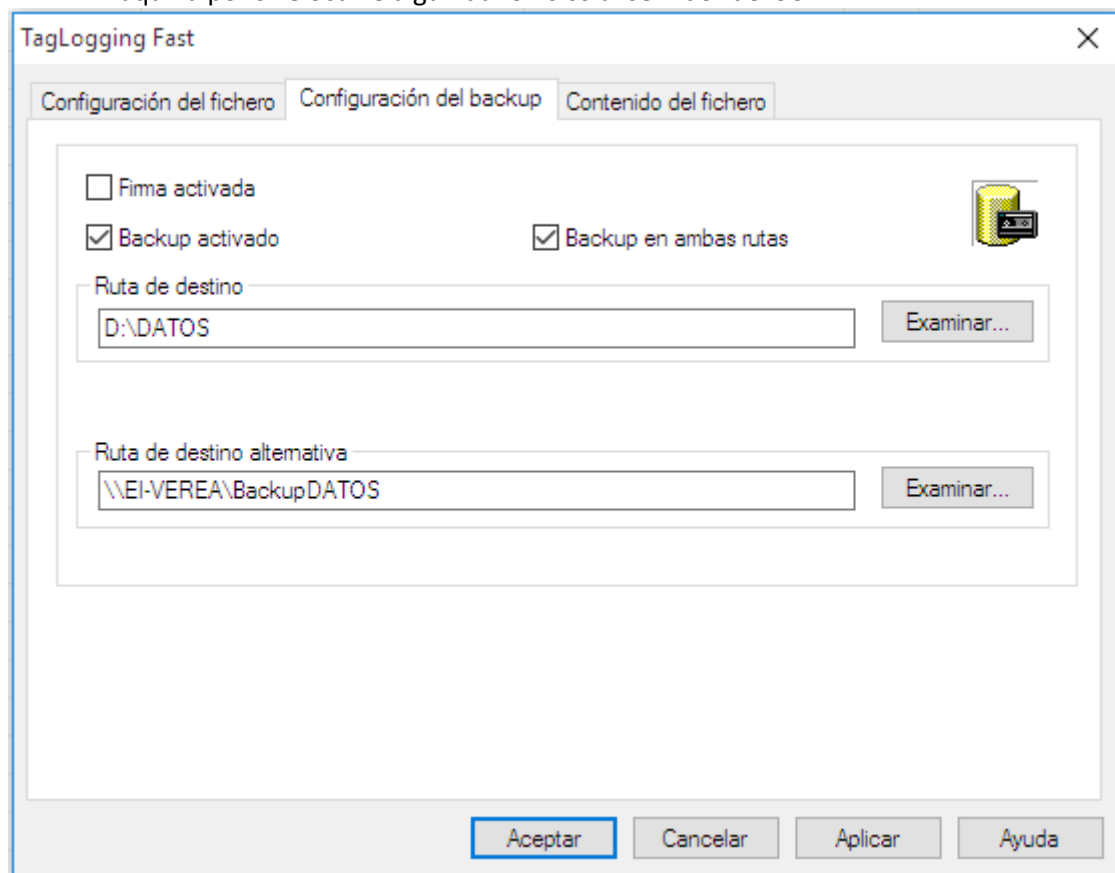
En la ficha Configuración del fichero se ajusta el tamaño máximo de los búferes de datos que compondrán la base de datos de ficheros y se configura el intervalo de tiempo del fichero, así como también la Fecha y hora del primer cambio de segmento.



Una vez superados los parámetros configurados para todos los segmentos se produce de forma automática un cambio de fichero y de igual forma cuando se superan los parámetros configurados para un segmento individual se produce automáticamente un cambio de segmento.

En la ficha Configuración del backup se dispone de los dos campos:

- **Ruta de destino:** para almacenar una copia de seguridad de los ficheros en el propio servidor.
- **Ruta de destino alternativa:** se recomienda que corresponda al directorio de otra máquina por si le ocurre algún daño físico al servidor del SCADA.

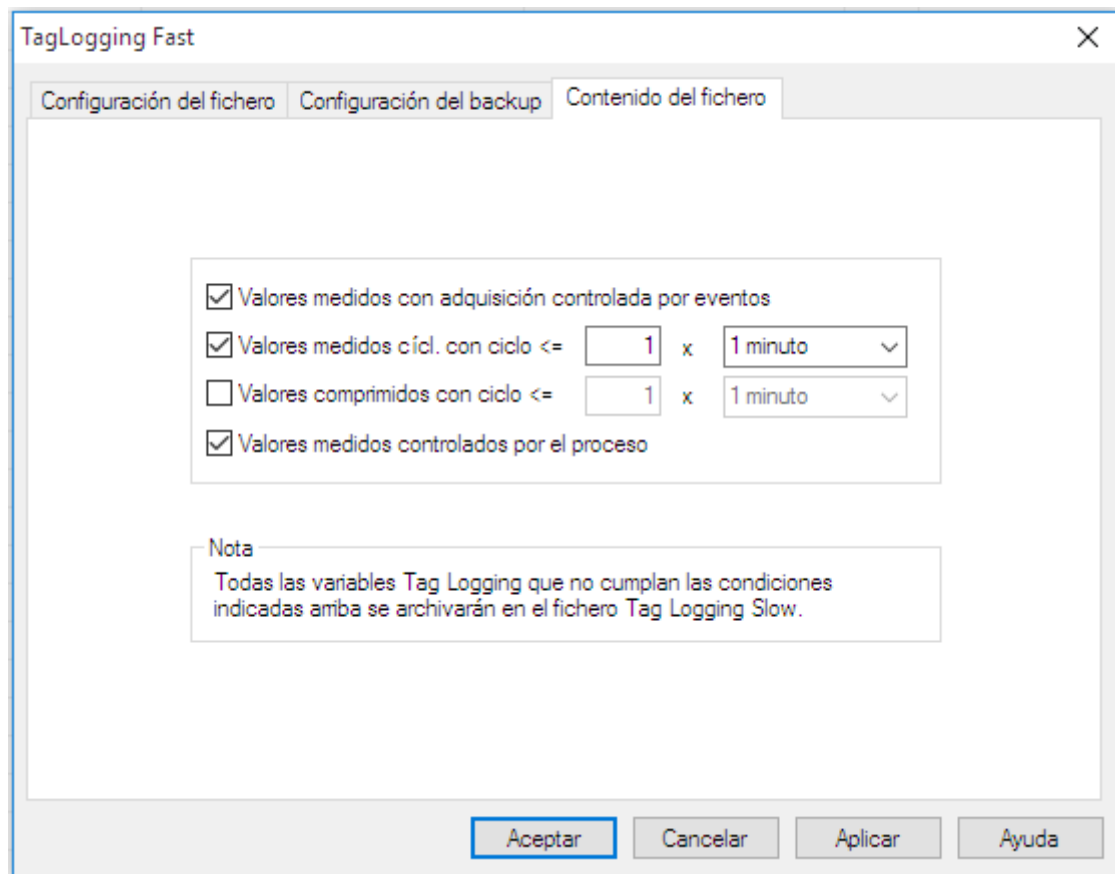


En Ruta de destino se selecciona la carpeta “DATOS” del disco D: del servidor del SCADA que está reservada específicamente para el almacenamiento de datos.

Como ruta de destino alternativa se configura la carpeta de red “BackupDATOS” que está situada en la estación de ingeniería.

Las casillas Backup activado y Backup en ambas rutas deben de estar activadas para que la copia de seguridad se lleve a cabo en los dos directorios.

En la ficha Contenido del fichero se configuran los tipos de valores de proceso que se desean almacenar en el área TagLogging Fast. En caso contrario los ficheros se almacenarán en el área TagLogging Slow.

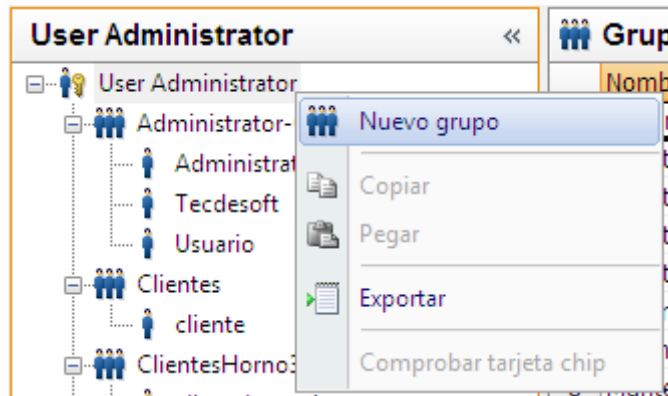


La configuración del contenido del fichero se deja como aparece por defecto con un tiempo de valores medidos cíclicamente de un minuto.

Como el tiempo del ciclo de adquisición que se configura para todas las variables de fichero es de 1 minuto solo se configura el TagLogging Fast.

### 2.1.8. Administración de permisos y usuarios

La herramienta User Administrator permite gestionar y realizar la configuración de los permisos que controlan los derechos de acceso para cada uno de los usuarios que manejarán el proyecto. De esta forma cuando un usuario inicia sesión, se comprueba si el usuario introducido está dado de alta como usuario del proyecto y los niveles de autorización de los que dispone para poder manejarlo.



Haciendo clic derecho sobre User Administrator se crean los grupos de usuario seleccionando “Nuevo grupo”, y se le asigna un nombre al grupo. Posteriormente se pueden crear los usuarios que se consideren oportunos dentro de cada uno de los grupos. De esta forma se permite gestionar los permisos de forma conjunta para cada uno de los grupos.

Se crean los usuarios y grupos de usuarios que se listan a continuación:

Usuarios [ todos ]								Buscar
Nombre de usuario	Nombre de grupo	Contraseña	Inicio de sesión	Tipo de cierre de sesión automático	Intervalo	WebNavigator	Imagen inicial de WebNavigator	
1 Administrator	Administrator-Group	●●●●●●	<input type="checkbox"/>	Ninguna		<input checked="" type="checkbox"/>	INICIO.Pd_	
2 angel	Encargados Turno	●●●●●●	<input type="checkbox"/>	Absoluto	5	<input type="checkbox"/>		
3 carlos	CientesMolienda	●●●●●●	<input type="checkbox"/>	Absoluto	500	<input type="checkbox"/>		
4 cliente	Cientes	●●●●●●	<input type="checkbox"/>	Ninguna		<input type="checkbox"/>		
5 clientehorno1	CientesHorno3	●●●●●●	<input type="checkbox"/>	Ninguna		<input type="checkbox"/>		
6 clientelab1	CientesLaboratorio	●●●●●●	<input type="checkbox"/>	Ninguna		<input type="checkbox"/>		
7 clientemol1	CientesMolienda	●●●●●●	<input type="checkbox"/>	Absoluto	500	<input type="checkbox"/>		
8 emilio	Encargados Turno	●●●●●●	<input type="checkbox"/>	Absoluto	5	<input type="checkbox"/>		
9 jlcastro	Mantenimiento	●●●●●●	<input type="checkbox"/>	Inactivo	10	<input type="checkbox"/>		
10 jose	Encargados Turno	●●●●●●	<input type="checkbox"/>	Absoluto	5	<input type="checkbox"/>		
11 lois	Encargados Turno	●●●●●●	<input type="checkbox"/>	Absoluto	5	<input type="checkbox"/>		
12 luciano	CientesMolienda	●●●●●●	<input type="checkbox"/>	Absoluto	500	<input type="checkbox"/>		
13 marcos	CientesMolienda	●●●●●●	<input type="checkbox"/>	Absoluto	500	<input type="checkbox"/>		
14 Miguel	Ingeniería	●●●●●●	<input type="checkbox"/>	Ninguna		<input type="checkbox"/>		
15 sergio	Encargados Turno	●●●●●●	<input type="checkbox"/>	Absoluto	5	<input type="checkbox"/>		
16 Tecdesoft	Administrator-Group	●●●●●●	<input type="checkbox"/>	Ninguna		<input checked="" type="checkbox"/>	INICIO.Pd_	
17 Usuario	Administrator-Group	●●●●●●	<input type="checkbox"/>	Ninguna		<input checked="" type="checkbox"/>	INICIO.Pd_	

Para cada uno de los usuarios, además de asignarles los permisos correspondientes, que se especificará como realizarlo a continuación, se le configuran los siguientes parámetros:

- **Contraseña:** es la cadena de caracteres alfanuméricos que deberá ingresar el usuario para poder iniciar sesión. Esta deberá tener una longitud entre 6 y 24 caracteres Unicode.
- **Tipo de cierre de sesión automático:** permite gestionar el cierre de sesión de forma automática estableciendo un tiempo relativo (tiempo de inactividad del usuario), absoluto (desde el inicio de sesión), o ninguno si se desea que el usuario tenga la sesión iniciada hasta que sea el mismo u outro usuario quien cierre la sesión.
- **Intervalo:** es el tiempo en segundos que se le asigna al tipo de cierre de sesión automático seleccionado.
- **WebNavigator:** permite seleccionar si el usuario puede acceder al SCADA desde el navegador web.
- **Imagen inicial de WebNavigator:** es la imagen de proceso que se encontrará el usuario al iniciar sesión desde el navegador web.

Al seleccionar un usuario o grupo de usuarios en el árbol del “User Administrator” se despliega a la derecha la lista de los permisos y se configuran marcando la casilla de la columna “Habilitación” para cada uno de los permisos.

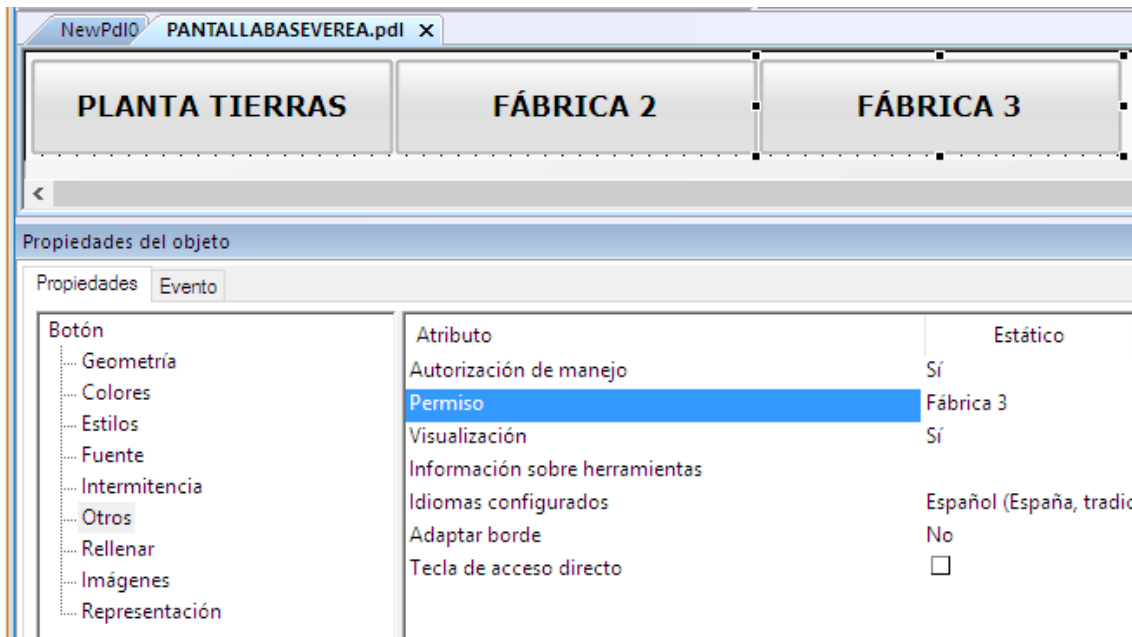
The screenshot shows the 'User Administrator' interface. On the left is a tree view of users and groups. The 'Tecdesoft' user is selected. On the right is a table titled 'Permisos [ Tecdesoft ]' with two columns: 'Función' and 'Habilitación'.

	Función	Habilitación
1	Administración de usuarios	<input checked="" type="checkbox"/>
2	Entrada de valores	<input checked="" type="checkbox"/>
3	Operaciones de bajo nivel	<input checked="" type="checkbox"/>
4	Tratamiento de imágenes	<input checked="" type="checkbox"/>
5	Cambiar imagen	<input checked="" type="checkbox"/>
6	Selección de ventanas	<input checked="" type="checkbox"/>
7	Copia de pantalla	<input checked="" type="checkbox"/>
8	Acusar avisos	<input checked="" type="checkbox"/>
9	Bloquear avisos	<input checked="" type="checkbox"/>
10	Liberar avisos	<input checked="" type="checkbox"/>
11	Edición de avisos	<input checked="" type="checkbox"/>
12	Iniciar fichero	<input checked="" type="checkbox"/>
13	Detener fichero	<input checked="" type="checkbox"/>
14	Editar valores de fichero	<input checked="" type="checkbox"/>
15	Configuración de ficheros	<input checked="" type="checkbox"/>
16	Edición de acciones	<input checked="" type="checkbox"/>
17	Administración de proyectos	<input checked="" type="checkbox"/>
18	Molienda	<input checked="" type="checkbox"/>
19	Fábrica 2	<input checked="" type="checkbox"/>
20	Fábrica 3	<input checked="" type="checkbox"/>
21	Mantenimiento	<input checked="" type="checkbox"/>
22	Alarmas	<input checked="" type="checkbox"/>
23	Actuaciones Operador	<input checked="" type="checkbox"/>
24	Comparación gráficas	<input checked="" type="checkbox"/>
25	Ajuste Básculas	<input checked="" type="checkbox"/>
26	Capturas de pantalla	<input checked="" type="checkbox"/>
27	Mando Setas	<input checked="" type="checkbox"/>
28	Mando	<input checked="" type="checkbox"/>
29	Mando FC205	<input checked="" type="checkbox"/>
30	Mando Draga	<input checked="" type="checkbox"/>
31	Activar remoto	<input checked="" type="checkbox"/>
32	Configurar remoto	<input checked="" type="checkbox"/>
33	Acceso web - Sólo observar	<input type="checkbox"/>

Además de los permisos estándar aparecen por defecto al crear un proyecto de WinCC, se configuran a mayores los permisos, desde el 18 al 30, necesarios para gestionar el acceso a las distintas áreas del proyecto y realizar operaciones de mando remoto desde las pantallas de mando de los motores, actuación sobre setas de emergencia, ajustar las básculas, etc...

### 2.1.9. Configuración de permisos en imágenes de proceso

Para configurar los permisos en los objetos de las imágenes de proceso puede realizarse desde el Graphics Designer seleccionando el objeto que se desee y asignando al atributo “Permiso” dentro de Otros el permiso que se desee.



Si el atributo “Permiso” se deja por defecto <Sin configuración de acceso> cualquier usuario, incluso si no tiene sesión iniciada en el sistema podría actuar sobre el objeto.

### 2.1.10. Scripts generados en VBA para automatizar y agilizar la creación del proyecto

Los scripts en VBA sirven para ejecutar macros que ayuden a automatizar al desarrollo en tiempo de diseño con lo que se consigue acelerar el desarrollo cuando se desea crear y/o configurar gran cantidad de objetos del mismo tipo.

Para ejecutar un script en VBA sobre una imagen .PDL determinada debe estar abierta en el Graphics Designer y ejecutarse dicho desde el editor de macros que puede abrirse pulsando Alt+F11.

#### Script que lista los objetos en pantalla y genera copias

A continuación se muestra el script desarrollado para generar copias a partir de objetos que ya estén instanciados en una imagen.PDL.

```
'BUSCA LOS NOMBRES DE LOS OBJETOS EN PANTALLA Y PERMITE GENERAR TANTAS COPIAS
DE ELLOS COMO SE PRECISE
Sub buscaCopiaObjetos()
    Dim resultadosBusqueda As HMICollection
    Dim objMember As HMIObject
    Dim iResult As Integer
    Dim j
    Dim myArray(20) As String 'OJO, si se produce un error '9' en tiempo de
    ejecucion, subindice fuera del intervalo, debemos aumentar la dimension del
    array
    Dim txt As String
    Dim i As Long
    Dim objNumber
    Dim numObj
    Dim nuevoObjeto
```

```

Dim k

Set resultadosBusqueda = ActiveDocument.HMIObjects.Find() 'Aquí se puede
implementar un filtro por tipo de objeto etc, p.ej: ObjectName:="*Faceplate*'
iResult = resultadosBusqueda.Count

j = 0
For Each objMember In resultadosBusqueda
If j <= iResult Then
    myArray(j) = objMember.ObjectName
End If
j = j + 1
Next objMember

For i = LBound(myArray) To UBound(myArray)
    txt = txt & "Objeto " & i & ": " & myArray(i) & vbCrLf
Next i

objNumber = InputBox("Ingrese el número de objeto que desea copiar: " &
vbCrLf & vbCrLf & txt, "Selector objetos") 'muestra y seleccion de los objetos
encontrados

If (objNumber >= 0) And (objNumber <> "") Then
    numObj = InputBox("Ingrese cantidad de objetos a generar: ",
"Generador de objetos") 'Introducir la cantidad de copias que queremos

    Set nuevoObjeto = ActiveDocument.HMIObjects(myArray(objNumber)) 'Se
selecciona el objeto por su nombre
    nuevoObjeto.Selected = True
    Application.ActiveDocument.CopySelection 'copia el objeto al
clipboard

    For k = 1 To numObj
        Application.ActiveDocument.PasteClipboard 'pega el objeto del
portapapeles
    Next
ElseIf (objNumber < 0) Then
    MsgBox ("Error!! numero introducido fuera de rango")
End If
End Sub

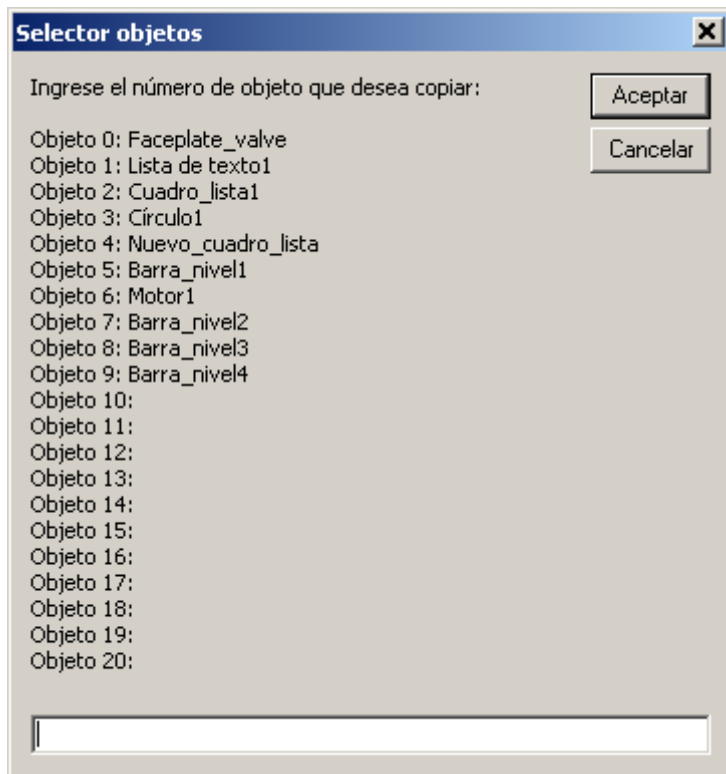
```

Debe configurarse el límite máximo de objetos en pantalla para que se cree la lista correctamente modificando la dimensión de "myArray" para que no de error cuando se ejecute si el número de objetos excede la dimensión del array.

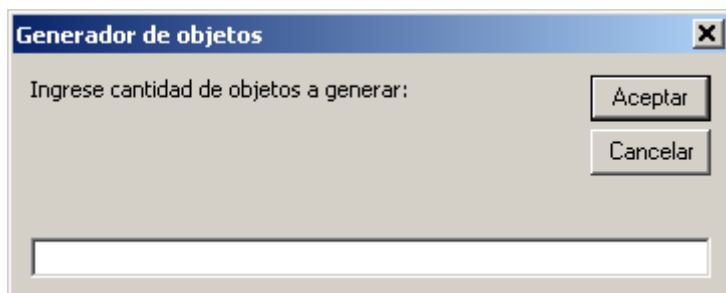
Por otra parte el objeto "resultadosBusqueda" queda preparado para poder implementar filtros para seleccionar objetos por tipo dentro de una imagen con distintos tipos de objeto. Esto es muy útil cuando ya se dispone de una pantalla desarrollada con muchos tipos de objetos y facilita la selección al usuario.

Cuando se ejecuta el script aparece una interfaz donde aparecen los objetos listados y un campo para que el usuario pueda introducir el valor numérico del objeto del cual desea realizar las copias.





Se introduce el número de objeto que se desea y se pulsa Aceptar.



Por último se ingresa el valor número de la cantidad de objetos que se desea generar y se pulsa Aceptar. Automáticamente aparecerán tantos objetos instanciados en la imagen de proceso como se hayan ingresado.

### Script configurador de Triggers en atributos de objetos

A continuación se muestra

*'Script que selecciona un objeto y dinamiza los atributos de forma automática con código vbs y triggers configurables*

```

Sub EditorTriggers ()
    Dim nombreObjeto As String
    Dim objVBScript As HMIInfo
    Dim objVarTrigger As HMIVariableTrigger

    Dim resultadosBusqueda As HMICollection
    Dim objMember As HMIObject
    Dim iResult As Integer

    Dim objNumber
    Dim numObj

```

```

Dim objSelected
Dim objName

Dim i As Integer
Dim strVBCode As String

'CONFIGURACION DEL NOMBRE DE OBJETO QUE SE DESEA SELECCIONAR
':.....:
nombreObjeto = "M2_LF1"
':.....:

Set                                resultadosBusqueda                                =
ActiveDocument.HMIObjects.Find(ObjectName:=nombreObjeto) 'Aqui se puede
implementar un filtro por tipo de objeto etc, p.ej: ObjectName:="*Faceplate*"
iResult = resultadosBusqueda.Count

For Each objMember In resultadosBusqueda
objName = objMember.ObjectName
Set objSelected = ActiveDocument.HMIObjects(objName) 'Se selecciona
el objeto por su nombre
objSelected.Selected = True
Next objMember

'CONFIGURACION DE LOS TRIGGERS QUE SE QUIEREN CREAR

'INFORMACION SOBRE HERRAMIENTAS
Set                                objVBScript                                =
objSelected.ToolTipText.CreateDynamic(hmiDynamicCreationTypeVBScript)
'objSelected.Propiedad...
With objVBScript
Nombre del trigger y tipo de ciclo definido con el metodo Add
Set objVarTrigger = .Trigger.VariableTriggers.Add(nombreObjeto &
".NombreMotor", hmiVariableCycleType_0) 'trigger al cambiar variable
(tiempo: hmiVariableCycleType_10ms)

'Codigo que lleva el script en VB (CONFIGURABLE)
strVBCode = "Dim TagName, TextPos" & vbCrLf & "Dim Name" & vbCrLf &
"' Dinamización del ToolTip en función del nombre del objeto." & vbCrLf &
vbCrLf & _
"' Formato del nombre del objeto: PrefijodePosicion*NombreTag" &
vbCrLf & "TextPos = InStr(1, Item.ObjectName, "" * "" , vbBinaryCompare)" &
vbCrLf & _
"TagName = Mid(Item.ObjectName, TextPos + 1, Len(Item.ObjectName))"
& vbCrLf & _
"Set Name = HMIRuntime.Tags(TagName & "" .NombreMotor"" )" & vbCrLf &
-
"Name.Read" & vbCrLf & _
"ToolTipText_Trigger = CStr(Name.value)"
.SourceCode = strVBCode
End With

MsgBox ("Triggers creados en objeto " & objName)
End Sub

```

Este script requiere de una alta configuración inicial por tanto debe de valorarse su empleo dependiendo de la cantidad de objetos que se quieren configurar.

## 2.2 ANEXO II - Manual del usuario SCADA

En este anexo se explica el modo de navegar y usar el SCADA explicando cada una de las pantallas que se han desarrollado.

### 2.2.1. Pantalla de inicio



Esta es la primera pantalla que se muestra al poner en Runtime el SCADA.

En ella se muestra una ortofoto de la planta de producción y 3 botones claramente diferenciados:

- **Apagar PC:** Apaga el ordenador en el que se está ejecutando el programa.
- **Salir a Windows:** Sale de Runtime y se muestra el escritorio de Windows.
- **Iniciar:** Inicia el SCADA a la pantalla base de operación.

### 2.2.2. Pantalla base



La pantalla base dispone de un visor de alarmas en la parte central inferior que se mantendrá en todo el SCADA donde se muestran los 5 últimos eventos de alarmas.

El menú superior está formado por los siguientes botones:

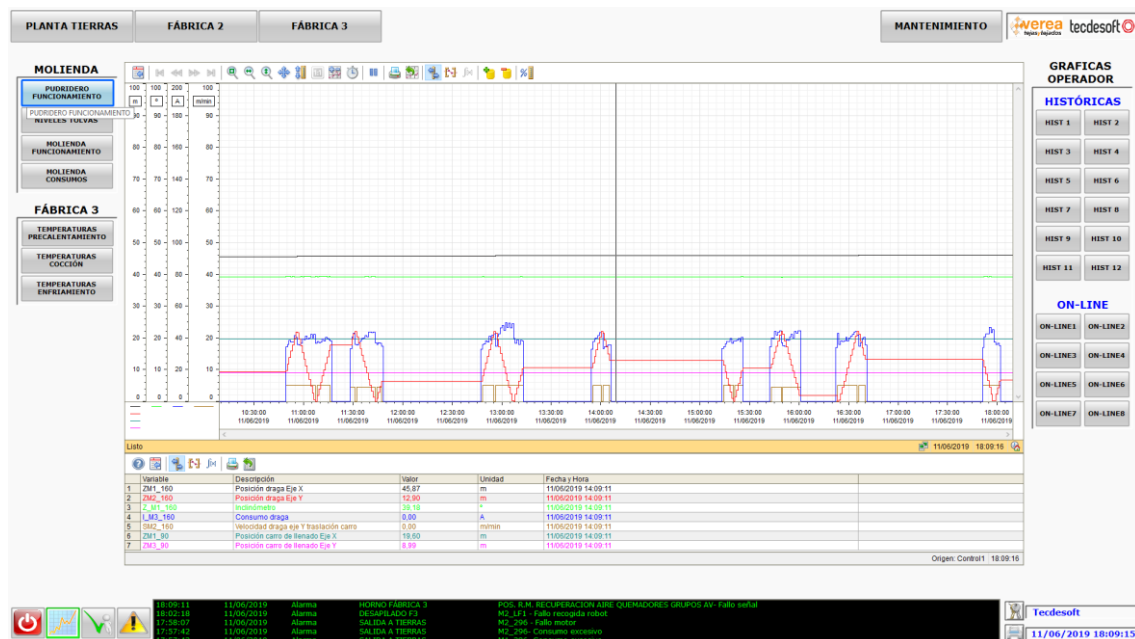
- 1- **Planta Tierras:** Accede al sinóptico general de la planta de tierras.
- 2- **Fábrica 2:** Accede al sinóptico general de la fábrica2.
- 3- **Fábrica 3:** Accede al sinóptico general de la fábrica3.
- 4- **Mantenimiento:** Accede a la pantalla de mantenimiento.

El menú inferior está formado por 6 botones:

- 5- **Salir a Inicio:** Vuelve a la pantalla de inicio explicada en el apartado anterior
- 6- **Gráficas:** Accede a la pantalla de las gráficas
- 7- **Actuaciones de operador:** Accede a la pantalla de actuaciones de operador
- 8- **Alarmas:** Accede a la pantalla donde se visualizan las alarmas
- 9- **Login:** Inicio o cierre de sesión de los usuarios.
- 10- **Imprimir:** Imprime lo que se ve en pantalla

### 2.2.3. Pantalla de gráficas

Si el operario hace clic en el botón 6 entra a la pantalla de gráficas que se muestra a continuación.



Esta pantalla tiene dos menús laterales con botones para poder seleccionar las gráficas que se deseen.

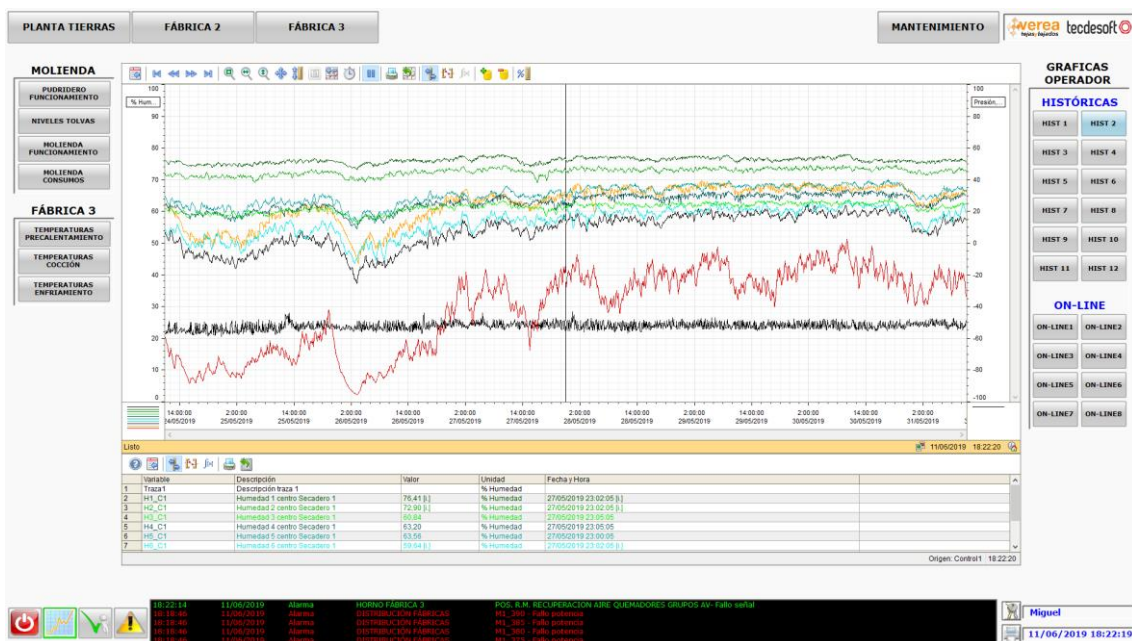
En el margen izquierdo se disponen las botoneras correspondientes a las gráficas fijas de la planta de Molienda (Planta de Tierras) y Fábrica 3.

Se han configurado las siguientes gráficas fijas:

- **Pudridero funcionamiento:** se representan las trazas de la posición de la draga y el carro, inclinómetro, consumo de la draga, velocidades.
- **Niveles tolvas:** se representan las trazas de los niveles de las 4 tolvas de la planta de tierras.

- **Molienda Funcionamiento:** se representan las trazas de caudales y consumos del desmenuzador y molino de rulos.
- **Molienda consumos:** se representan los consumos de los motores de los 4 laminadores de la panta de tierras.
- **Temperaturas precalentamiento:** se representan las temperaturas de la zona de precalentamiento del horno de la fábrica 3.
- **Temperaturas cocción:** se representan las temperaturas de la zona de cocción del horno de la fábrica 3.
- **Temperaturas enfriamiento:** se representan las temperaturas de la zona de enfriamiento del horno de la fábrica 3.

En el margen derecho se disponen las botoneras correspondientes a las gráficas históricas y on-line definidas por el operador, esto quiere decir que cada uno de los operarios puede definir en estos botones las gráficas que desee visualizar.



Tanto las gráficas históricas como las on-line se configuran en el botón de propiedades situado en la esquina izquierda de la gráfica. Al pulsarlo, se abrirá un menú de configuración que permitirá configurar las trazas, colores, tiempos, etc...

Hay que tener en cuenta que todos los cambios que realice el usuario en la configuración de las gráficas de operador se mantendrán siempre aunque se desactive el Runtime del SCADA, en su próxima ejecución los cambios y configuraciones que se hubiesen realizado se mantendrán.

PLANTA TIERRAS FABRICA 2 FABRICA 3 MANTENIMIENTO **WEREA** tecdesoft®

**MOLIENDA**

PROPIEDADES: WinCC OnlineTercerControl

Barra de herramientas Barra de estado Configuración online Espantación

Curvas General Fuente Ventana de curvas Ejes de tiempos Ejes de valores

Curvas: Traca1

Nombre de objeto: Traca1

ventana de curvas: Ventana de curvas 1

Eje de tiempos: Eje de tiempos 1

Eje de valores: Eje de valores 1

Esquema 1: Descripción: Descripción traca 1

Consolidar datos

Suministro de datos: Nombre de la variable:

2 - Variables online  Mostrar alarma

Representación

Tipo de curvas: 2 - Curva escalonada  Color de curva:  Relleno

Gras de línea: 1

0 - salida  Tipo de punto: 0 - rango  Avanzado  Valores límite

Libro

Variable	Descripción	Valor	Unidad	Fecha y Hora
1				

Origen: Control1 18:11:45

11/06/2019 18:11:45

10:58:31 11/06/2019 Alarma MOLIENDA FABRICA 3 POS: R.M. RECUPERACION AIRE QUIMICOS/GRUPOS AN- Falso nivel

10:58:27 11/06/2019 Alarma DESPLAZADO F3 M2-296 Falso potencia

10:58:40 11/06/2019 Alarma SALIDA A TIERRAS M2-296 Falso potencia

10:58:42 11/06/2019 Alarma SALIDA A TIERRAS M2-296 Corriente excesiva

10:58:42 11/06/2019 Alarma SALIDA A TIERRAS M2-296 Consumo excesivo

Tecdesoft 11/06/2019 18:11:45

Las gráficas on-line se representan sobre un fondo en negro para diferenciarlas de las históricas.

## 2.2.4. Pantalla de actuaciones de operador

PLANTA TIERRAS FABRICA 2 FABRICA 3 MANTENIMIENTO **WEREA** tecdesoft®

Actuaciones Operador

Fecha	Hora	Texto de aviso	Nombre de usuario	Nombre de equipo
97	11/06/19	11:27:17 Pulsador Mando Manual Motor M1_296 / Cilindro 1 - Laminador 296	icastro	
98	11/06/19	11:18:32 Pulsado Forzado Inmediato, Fabrica 2 - Tolva 2	icastro	
99	11/06/19	11:14:54 param_mol_65 - icastro nuevo-B antiguo-B	icastro	CLIENTE-MOLIENDA
100	11/06/19	11:14:54 param_mol_65 - icastro nuevo-B antiguo-B	icastro	CLIENTE-MOLIENDA
101	11/06/19	11:14:54 param_mol_65 - icastro nuevo-B antiguo-B	icastro	CLIENTE-MOLIENDA
102	11/06/19	11:12:01 USERT-CLIENTE-MOLIENDA:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
103	11/06/19	10:25:37 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
104	11/06/19	10:20:24 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
105	11/06/19	10:17:22 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
106	11/06/19	10:16:13 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
107	11/06/19	10:15:52 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
108	11/06/19	10:13:05 USERT-CLIENTE-MOLIEND:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
109	11/06/19	09:56:44 USERT-SERVER-A:Inicio de sesión manual	Tecdesoft	CLIENTE-MOLIEND
110	11/06/19	08:59:51 Pulsado Paro Inmediato, Origen Tolvas	marcos	CLIENTE-MOLIEND
111	11/06/19	08:57:14 Pulsador Orden Paro Motor M1_230 / Molino de rulos	marcos	CLIENTE-MOLIEND
112	11/06/19	08:57:11 Pulsador Orden Paro Motor M1_230 / Molino de rulos	marcos	CLIENTE-MOLIEND
113	11/06/19	08:57:00 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
114	11/06/19	08:57:00 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
115	11/06/19	08:57:00 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
116	11/06/19	08:56:59 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
117	11/06/19	08:56:59 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
118	11/06/19	08:56:59 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
119	11/06/19	08:56:58 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
120	11/06/19	08:56:58 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
121	11/06/19	08:56:58 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
122	11/06/19	08:56:58 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
123	11/06/19	08:56:58 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
124	11/06/19	08:56:57 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
125	11/06/19	08:56:57 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
126	11/06/19	08:56:57 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
127	11/06/19	08:56:56 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
128	11/06/19	08:56:56 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
129	11/06/19	08:56:56 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
130	11/06/19	08:56:56 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
131	11/06/19	08:56:56 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
132	11/06/19	08:56:55 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
133	11/06/19	08:56:55 Pulsado OFF, Origen Tolvas	marcos	CLIENTE-MOLIEND
134	11/06/19	08:56:28 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
135	11/06/19	08:56:28 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
136	11/06/19	08:56:28 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
137	11/06/19	08:56:27 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
138	11/06/19	08:56:27 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
139	11/06/19	08:56:27 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
140	11/06/19	08:56:27 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
141	11/06/19	08:56:27 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
142	11/06/19	08:56:26 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
143	11/06/19	08:56:26 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
144	11/06/19	08:56:26 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND
145	11/06/19	08:56:26 Pulsador Orden Paro Válvula V1_230 / E.V. Enbrague molino	marcos	CLIENTE-MOLIEND

Pendiente: 93 Pendiente de acuse: 93 Oculto: 0 Lista: 1000 18:34:33

10:58:31 11/06/2019 Alarma MOLIENDA FABRICA 3 POS: R.M. RECUPERACION AIRE QUIMICOS/GRUPOS AN- Falso nivel

10:58:40 11/06/2019 Alarma DISTRIBUCION FABRICA M1-388 Falso potencia

10:58:40 11/06/2019 Alarma DISTRIBUCION FABRICA M1-388 Falso potencia

10:58:40 11/06/2019 Alarma DISTRIBUCION FABRICA M1-388 Falso potencia

Tecdesoft 11/06/2019 18:34:33

En la pantalla de actuaciones de operador se visualiza el registro de las acciones que llevan los operadores del SCADA tales como, mando de accionamientos como pueden ser motores, válvulas, cintas, así como también cualquier tipo de actuación que afecte al proceso productivo como puede ser cambios en los parámetros, productos, etc...

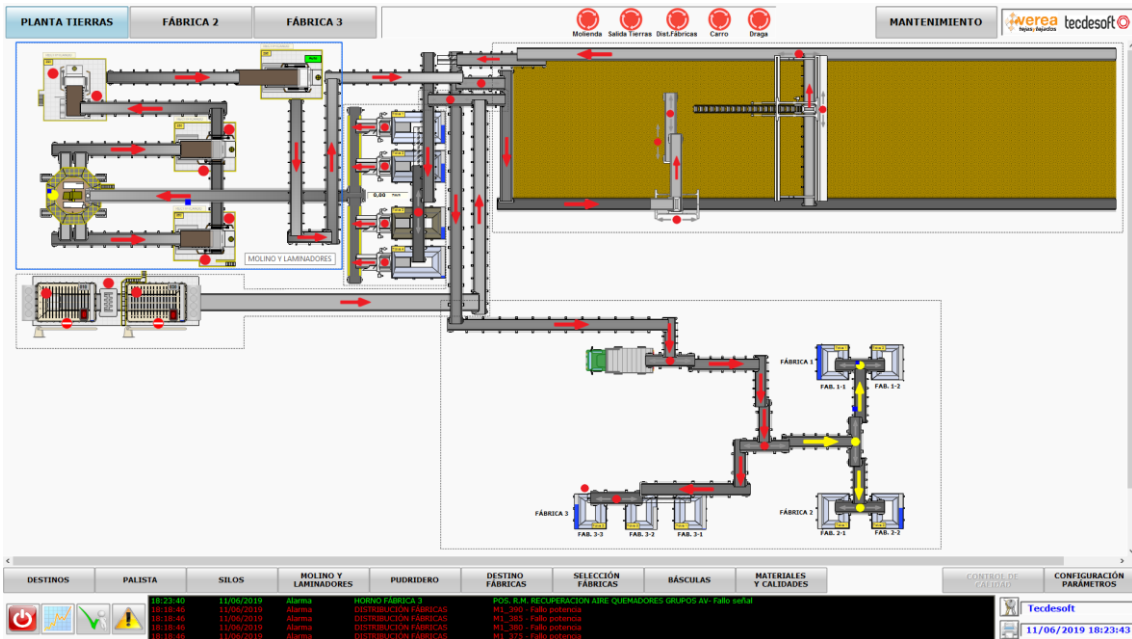
Cada operación se registra con una fecha, hora, texto de aviso, nombre de usuario y el equipo desde el que se ha realizado, lo que permite posteriormente realizar un filtrado para buscar algún evento específico.

## 2.2.5. Pantalla de alarmas

La pantalla de alarmas está compuesta por un menú lateral situado a la izquierda que nos permite establecer filtros sobre el visor de alarmas para seleccionar de forma rápida las alarmas correspondientes a cada una de las plantas.

Si se desea realizar un filtrado más específico se puede realizar desde el filtro del visor de alarmas seleccionando los campos y parámetros que se deseen.

## 2.2.6. Pantalla planta tierras

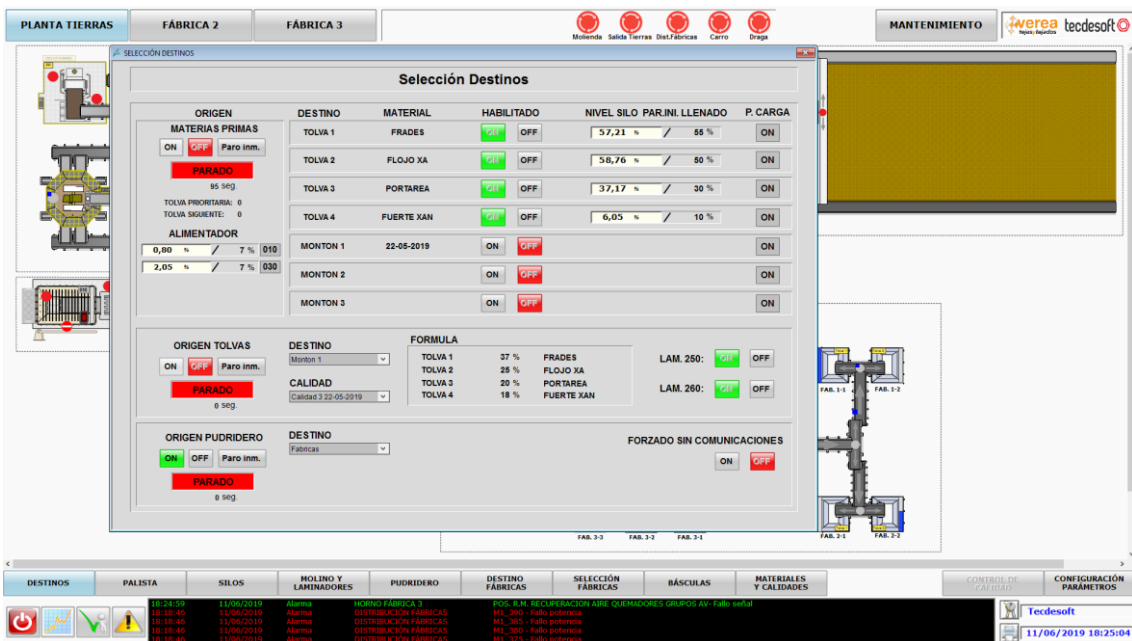


Al presionar el botón PLANTA TIERRAS se visualiza el sinóptico general de la planta de tierras donde se aprecian aquellos elementos más importantes como los motores, cintas principales, y niveles de las tolvas.

También se visualiza en el menú inferior los botones que permiten ir navegando por los detalles de cada una de las áreas de esta planta, aunque también se puede acceder del mismo modo presionando sobre el área recuadrada en el sinóptico general.

A continuación se describen en detalle las pantallas de las distintas zonas de la planta de tierras:

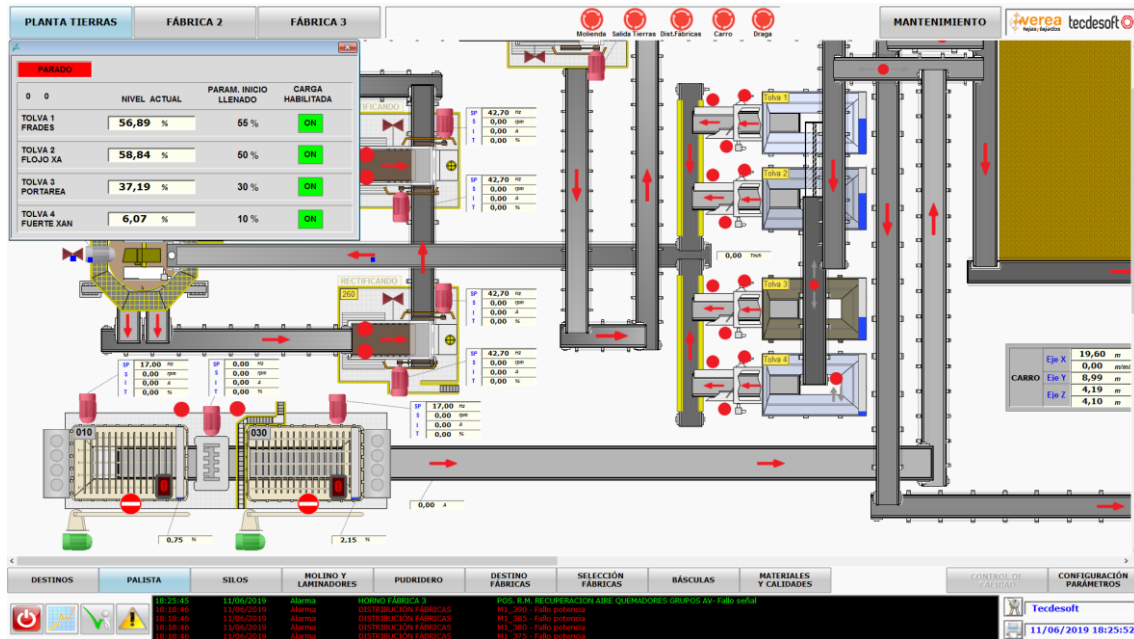
## 2.2.7. Pantalla destinos



En la pantalla de destinos se maneja el funcionamiento de la planta de tierras seleccionando el origen de las materias primas.

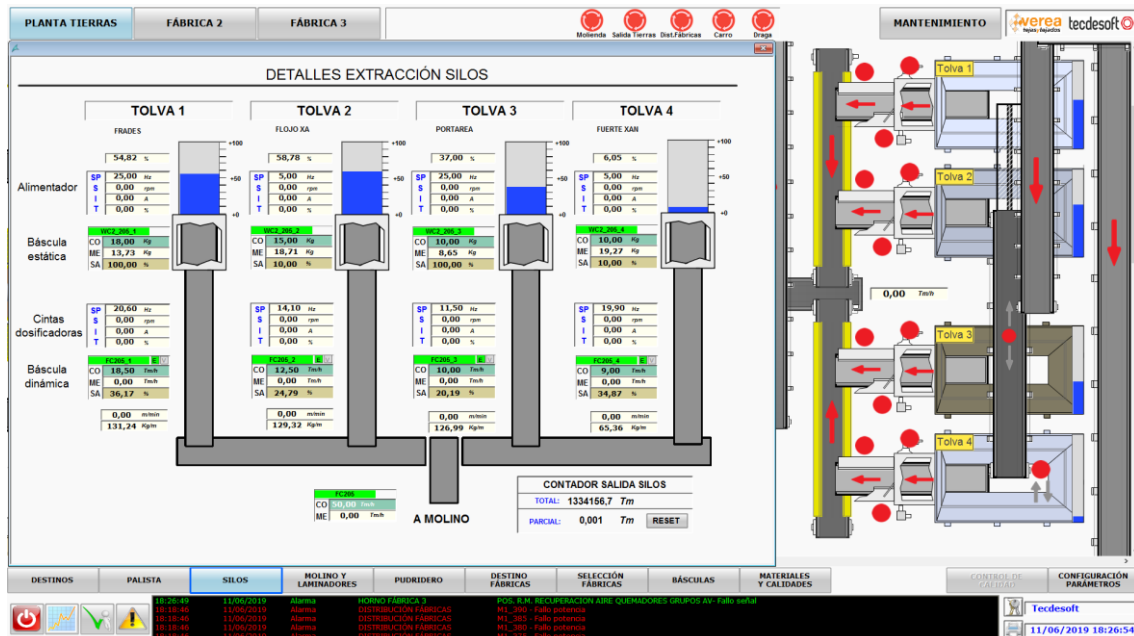


## 2.2.8. Pantalla palista



Esta pantalla está orientada al palista que añade tierra a los alimentadores que criban la tierra y la introducen a la planta por la cinta de alimentación principal. Desde ella pueden visualizarse tanto los niveles de los propios alimentadores, como el de las tolvas de tierras y actuar sobre los accionamientos que se consideren necesarios.

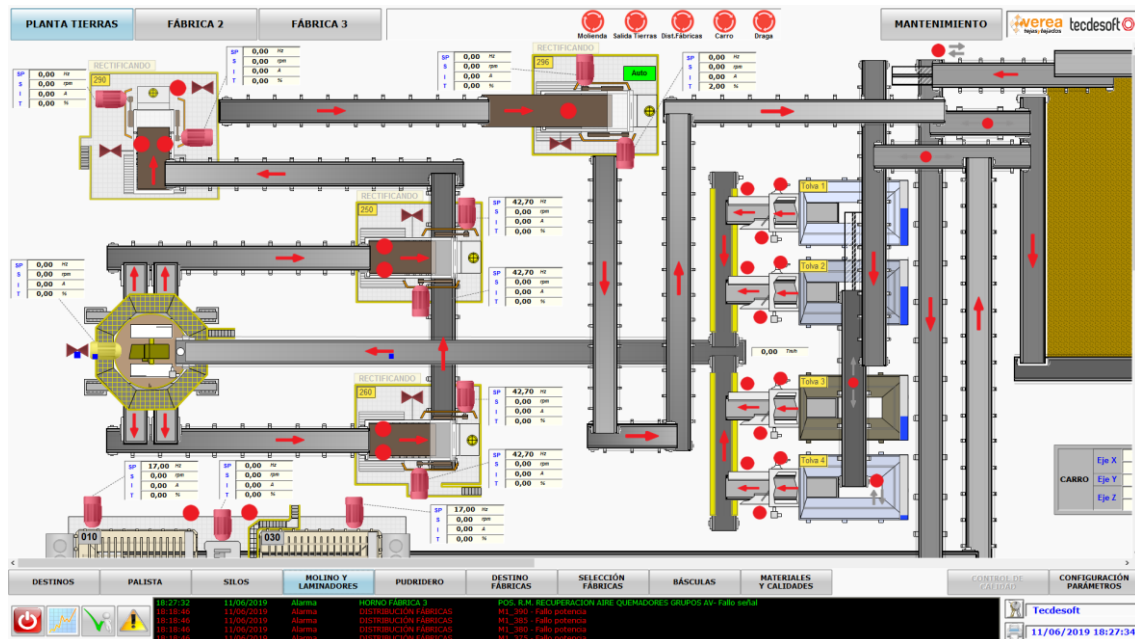
## 2.2.9. Pantalla silos



En esta pantalla se aprecia con más detalle la zona de los silos (o tolvas) que contiene las tierras, qué además de los niveles que también se disponen en otras visualizaciones se pueden apreciar los reguladores de las básculas estáticas y dinámicas así como también los alimentadores y cintas dosificadoras.

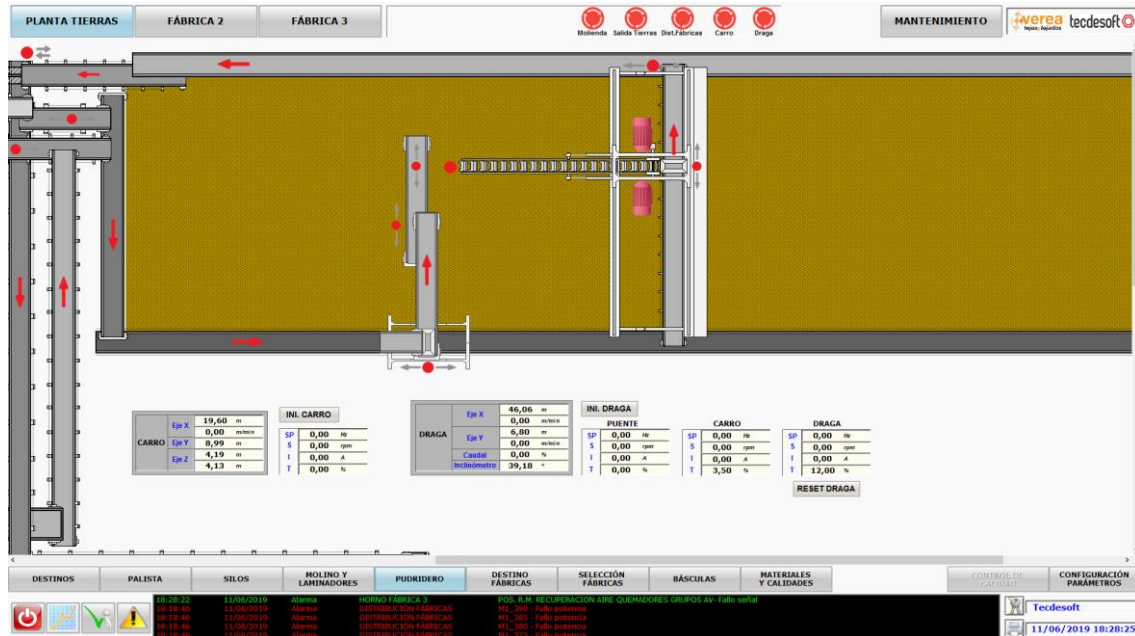
Si el operador desea poner a 0 el contador parcial debe de presionar el botón "RESET" situado en el contador de salida silos.

## 2.2.10. Pantalla molino y laminadores



En la pantalla de molino y laminadores pueden verse y comandarse todos los motores, raspadoras, cintas y valores analógicos de los motores correspondientes al molino de rulos y los laminadores encargados de reducir la granulometría de la tierra.

## 2.2.11. Pantalla pudridero

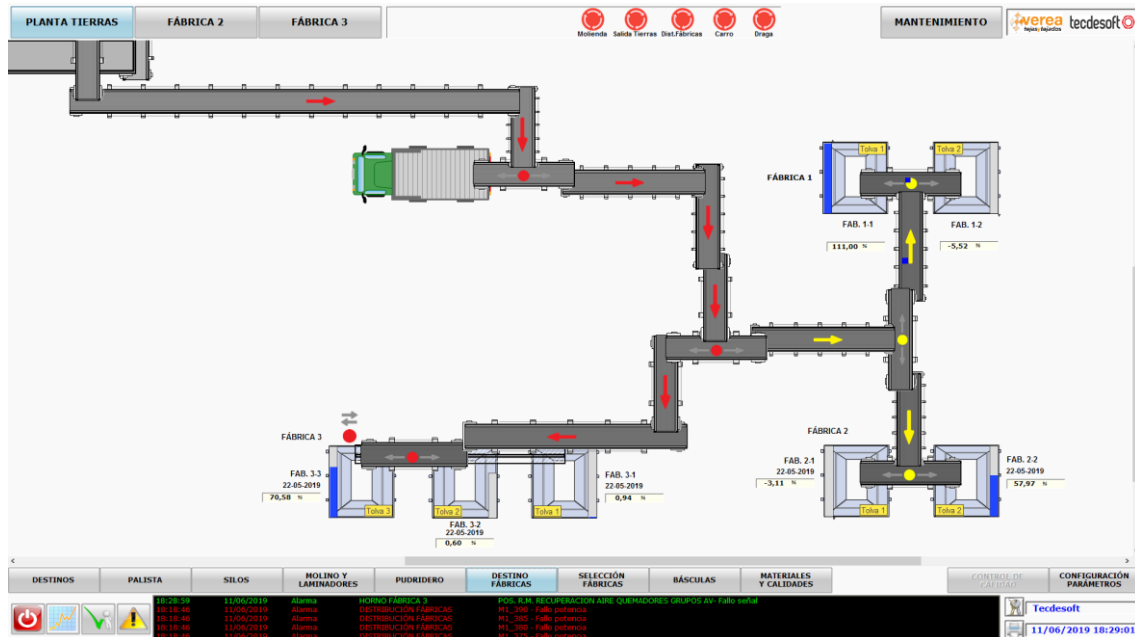


En esta pantalla puede apreciarse el pudridero que está constituido por un carro y una draga que distribuyen la tierra en los 3 montones que componen esta nave. La posición del carro y la draga es la que corresponde con la realidad. En esta pantalla existen los siguientes botones:

- **INI. CARRO:** Posiciona el carro en la posición inicial.
- **INI. DRAGA:** Posiciona la draga en la posición inicial.

- **RESET DRAGA:** Resetea los valores de la draga.

## 2.2.12. Pantalla destino fábricas



La pantalla de destino fábricas muestra las cintas que conforman los posibles caminos de salida de la tierra de la planta. Existen los siguientes destinos:

- **Camión:** Si es necesario transportar la tierra procesada en un camión, puede hacerse mediante la cinta que se dispone a la altura para echar la tierra en el camión.
- **Tolva 1 y 2 - Fábrica1:** Aunque la Fábrica 1 no está operativa se dispone de dos tolvas para almacenar la tierra procesada
- **Tolva 1 y 2 - Fábrica2:** Corresponde a las dos tolvas de la fábrica 2.
- **Tolva 1, 2 y 3 - Fábrica3:** Corresponden a las 3 tolvas que se disponen en la fábrica 3.

## 2.2.13. Pantalla selección fábricas

DESTINO	AUTO	NIV. INICIO / CAUDAL MAX	NIV. PARO / NIV. ACTUAL	BLOQUEO SCADA	CAMPO	FORZADO INMEDIATO
FAB 1: ON	FAB. 1-1	Aut 70% / 111,00 %	88% / 16%	OFF	BLQ	YA
ON	FAB. 1-2	Aut 70% / 111,00 %	88% / -5,42 %	OFF	BLQ	YA
FAB 2: ON	FAB. 2-1	Aut 40% / 70 %	18% / -2,42 %	OFF	BLQ	YA
ON	FAB. 2-2	Man 40% / 70 %	18% / 57,93 %	OFF	BLQ	YA
FAB 3: ON	FAB. 3-1	Aut 40% / 65 %	18% / 0,96 %	OFF	BLQ	YA
ON	FAB. 3-2	Aut 40% / 65 %	18% / 0,48 %	OFF	BLQ	YA
ON	FAB. 3-3	Man 40% / 60 %	18% / 69,91 %	OFF	BLQ	YA
CAMIÓN: ON	CAMION	18%		OFF	BLQ	Al seleccionar camión, para iniciar presione el botón verde en botonera debajo de la cinta, al parar presionar el botón rojo.

Esta pantalla permite seleccionar el destino de la salida de la tierra a las tolvas que se desee en cada momento y gestionar el direccionamiento de la misma en modo automático o manual.

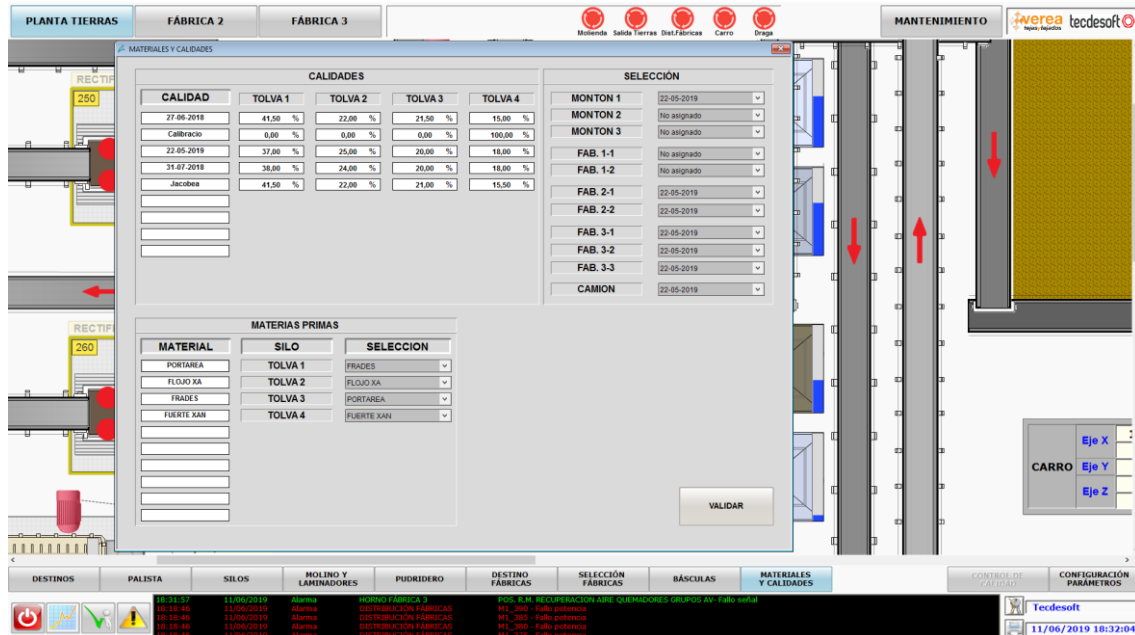
## 2.2.14. Pantalla básculas

REGULAD.	VELOCIDAD PESO	TOTALIZ.	CAUDAL NOM.
CO 38,50 Tmh ME 0,00 Tmh SA 36,16 %	0,00 m/min 131,26 kg/m	340402,5 0,001	38.998 Tmh 5.333 m³/min 125.000 Kg/m
CO 32,50 Tmh ME 0,00 Tmh SA 24,79 %	0,00 m/min 129,33 kg/m	297469,8 0	38.998 Tmh 5.333 m³/min 125.000 Kg/m
CO 10,00 Tmh ME 0,00 Tmh SA 20,19 %	0,00 m/min 126,97 kg/m	513416 0	38.998 Tmh 5.333 m³/min 125.000 Kg/m
CO 9,00 Tmh ME 0,00 Tmh SA 34,87 %	0,00 m/min 65,38 kg/m	182868,4 0	38.998 Tmh 5.333 m³/min 125.000 Kg/m

La pantalla de las básculas está compuesta por los reguladores y los valores analógicos correspondientes a cada uno de los parámetros: totalizador, caudal, peso, velocidad... además de disponer de los botones que permiten ajustarlas a cero, realizar un taraje

dinámico, ajustar el span, así como también ajustar los parámetros y temporizadores para cada una de ellas.

- **Pantalla materiales y calidades**



La pantalla de materiales y calidades se compone de 3 secciones claramente diferenciadas:

- **CALIDADES:** Permite definir hasta 5 calidades distintas y asignarle a cada una de ellas el porcentaje de los distintos tipos de tierras que se disponen en las tolvas de la planta.
- **MATERIAS PRIMAS:** Hace referencia al nombre que se le asigna a la tierra que contiene la tolva, según la procedencia de la mina de la que procede.
- **SELECCIÓN:** Se selecciona la calidad de la tierra que se desea obtener en cada uno de los destinos.

Si se realiza algún cambio en alguna de las áreas de esta pantalla y no se pulsa el botón VALIDAR, los cambios no tienen efecto. Para que se envíen al PLC y tengan efecto en el proceso productivo se deberá pulsar el botón VALIDAR.

## 2.2.15. Pantalla configuración de parámetros

**Parámetros Molenda**

**PARÁMETROS MOLENDA (1)**

Nivel muy alto tohva A - Paro carga materias primas	90,00	%
Nivel alto tohva A - Máximo admitido	85,00	%
Nivel bajo tohva A - Inicio carga materias primas	55,00	%
Nivel muy bajo tohva A - Paro basculas	2,00	%
Nivel muy alto tohva B - Paro carga materias primas	95,00	%
Nivel alto tohva B - Máximo admitido	85,00	%
Nivel bajo tohva B - Inicio carga materias primas	50,00	%
Nivel muy bajo tohva B - Paro basculas	2,00	%
Nivel muy alto tohva C - Paro carga materias primas	90,00	%
Nivel alto tohva C - Máximo admitido	55,00	%
Nivel bajo tohva C - Inicio carga materias primas	30,00	%
Nivel muy bajo tohva C - Paro basculas	2,00	%
Nivel muy alto tohva D - Paro carga materias primas	95,00	%
Nivel alto tohva D - Máximo admitido	35,00	%
Nivel bajo tohva D - Inicio carga materias primas	10,00	%
Nivel muy bajo tohva D - Paro basculas	1,00	%

**PARÁMETROS MOLENDA (2)**

Nivel bajo alimentador 010 - Detección material	7,00	%
Nivel bajo alimentador 030 - Detección material	7,00	%
Consumo elevado Laminador 250 - Mantenimiento	100,00	A
Consumo elevado Laminador 260 - Mantenimiento	100,00	A
Consumo elevado Laminador 290 - Mantenimiento	100,00	A
Velocidad trabajo M1_10 Alimentador lineal	34,00	%
Velocidad trabajo M1_30 Alimentador lineal	34,00	%
Velocidad trabajo M1_250 Cilindro 1 - laminador 1	85,00	%
Velocidad trabajo M2_250 Cilindro 2 - laminador 1	85,00	%
Velocidad trabajo M1_260 Cilindro 1 - laminador 2	85,00	%
Velocidad trabajo M2_260 Cilindro 2 - laminador 2	85,00	%
Velocidad trabajo M1_290 Cilindro 1 - laminador articulado	90,00	%
Velocidad trabajo M2_290 Cilindro 2 - laminador articulado	92,00	%
Velocidad Trabajo M1_20 - Desmenuzador	100,00	%
Velocidad Trabajo M1_230 - Molino de raños	8,00	%
Consumo alto en Molino de raños M1_230 (máximo seguridad)	200,00	A
Consumo bajo en Molino de raños M1_230 (mínimo seguridad)	65,00	A
Consumo alto en Molino de raños M1_230 (Paro alimentacion)	170,00	A

**PARÁMETROS MOLENDA (3)**

Alimentador Tohva A - Frecuencia Minima	3,00	Hz
Alimentador Tohva A - Frecuencia Maxima	25,00	Hz
Alimentador Tohva B - Frecuencia Minima	3,00	Hz
Alimentador Tohva B - Frecuencia Maxima	25,00	Hz
Alimentador Tohva C - Frecuencia Minima	3,00	Hz
Alimentador Tohva C - Frecuencia Maxima	25,00	Hz
Alimentador Tohva D - Frecuencia Minima	2,00	Hz
Alimentador Tohva D - Frecuencia Maxima	25,00	Hz

**PARÁMETROS MOLENDA (4)**

Velocidad rectificadora laminador M1_250	9,00	%
Velocidad rectificadora laminador M2_250	13,00	%
Velocidad rectificadora laminador M1_260	9,00	%
Velocidad rectificadora laminador M2_260	13,00	%
Velocidad rectificadora laminador M1_290	8,92	%
Velocidad rectificadora laminador M2_290	8,80	%
Velocidad rectificadora laminador M1_296	9,50	%
Velocidad rectificadora laminador M2_296	9,50	%
Diámetro camisa laminador M1_296	800,00	mm
Diámetro camisa laminador M2_296	800,00	mm
Consumo elevado laminador M1_296	216,00	A
Consumo elevado laminador M2_296	216,00	A
Referencia velocidad laminador M1_296 (Manual)	100,00	%
Referencia velocidad laminador M2_296 (Manual)	100,00	%

**PARÁMETROS MOLENDA (4) (Continuación)**

Nivel muy alto alimentador 010	100,00	%
Nivel muy alto alimentador 030	100,00	%
Factor equiancia nivel tohvas/alimentadores	+3,50	A
Valor alta intensidad M1.40	+22,50	A
Valor bajo intensidad M1.40	+10,00	A
Valor muy bajo intensidad M1.40, paro automático de la cinta	+5,00	A

**DESTINOS** | **PALISTA** | **SILOS** | **MOLINO Y LAMINADORES** | **PUDRIDERO** | **DESTINO FABRICAS** | **SELECCIÓN FABRICAS** | **BÁSCULAS** | **MATERIALES Y CALIDADES**

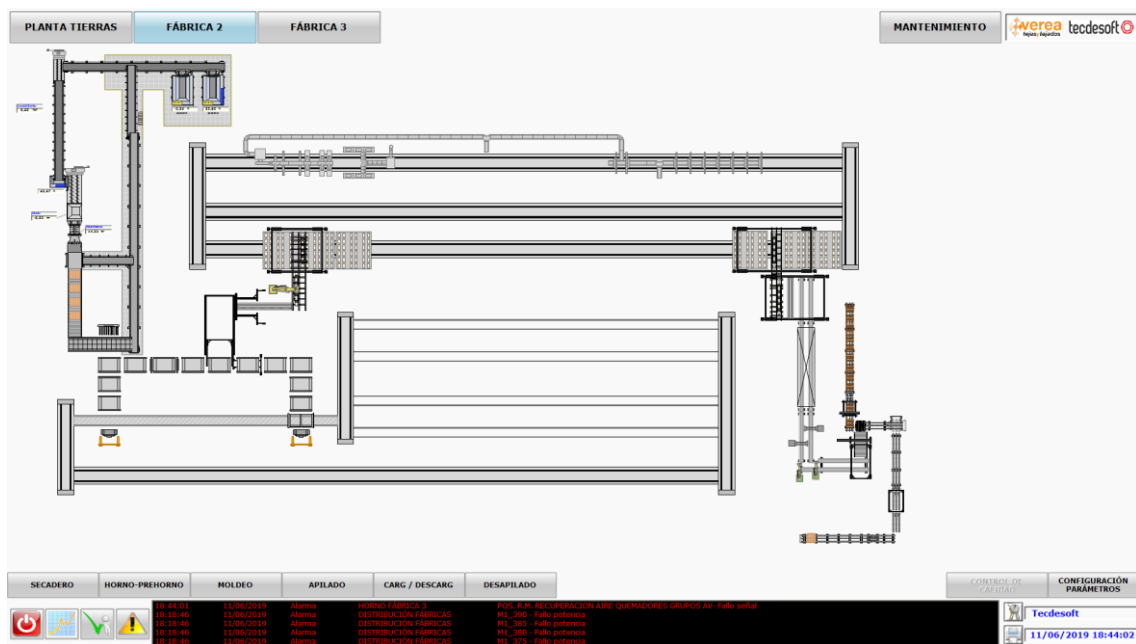
**CONFIGURACIÓN PARÁMETROS**

11/06/2019 18:33:15

Al presionar el botón “CONFIGURACIÓN DE PARÁMETROS” situado a la derecha del menú inferior se desplegará un menú emergente que permitirá seleccionar las pantallas de parámetros y temporizadores de cada una de las secciones de la planta de tierras.

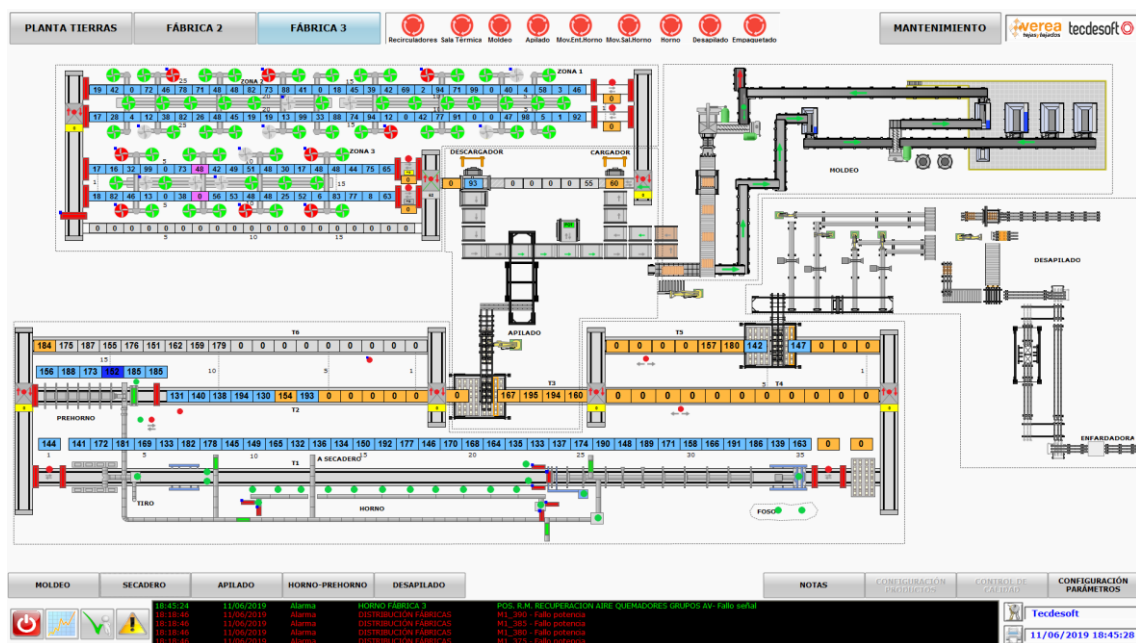
El procedimiento para realizar un cambio de parámetro consiste en hacer clic sobre el campo de entrada que deseemos, introducir el valor para ese parámetro y pulsar la tecla ENTER.

## 2.2.16. Pantalla fábrica 2



La pantalla de la fábrica 2 contiene el dibujo del sinóptico general y las analógicas de los niveles de las tolvas. Se disponen los botones del menú inferior pero no tienen código ya que la fábrica todavía está en proceso de montaje y no se dispone de la información necesaria para desarrollar esta parte del SCADA.

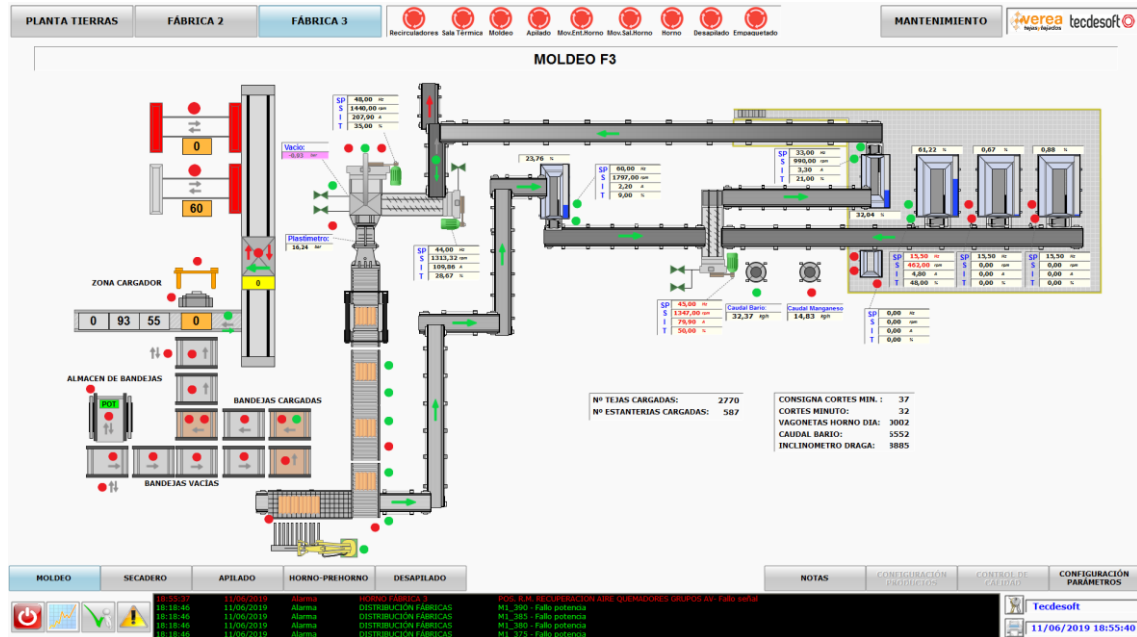
## 2.2.17. Pantalla fábrica 3



Al presionar el botón FÁBRICA 3 se visualiza el sinóptico general de la fábrica 3 donde se aprecian aquellos elementos más importantes como los motores, cintas principales, y niveles de las tolvas, estanterías, vagonetas, puertas y registros.

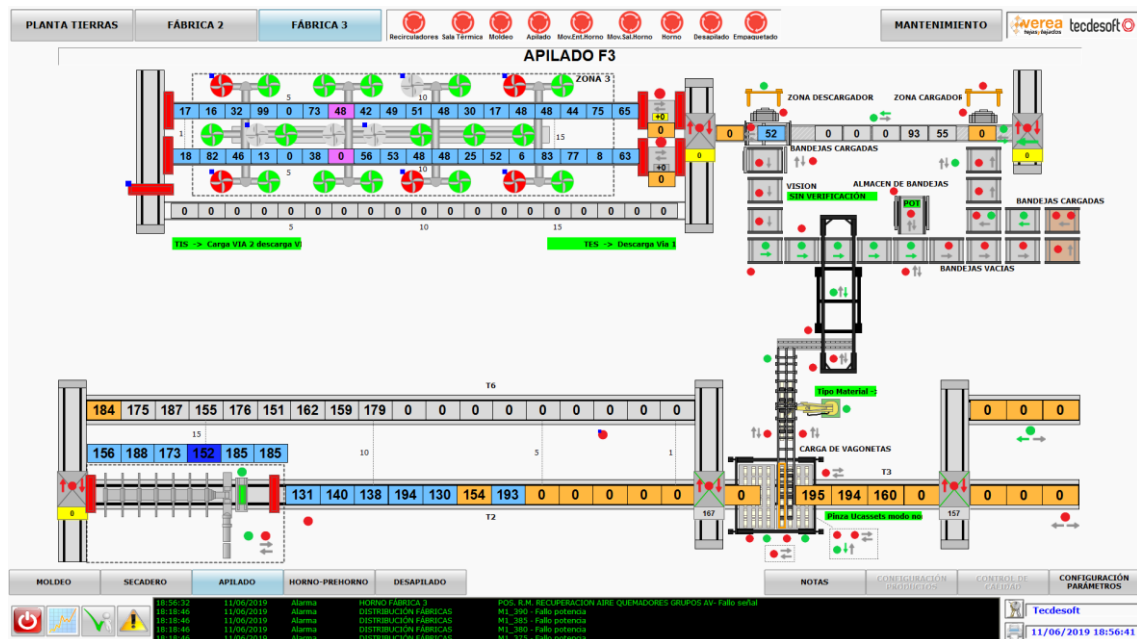
También se visualiza en el menú inferior los botones que permiten ir navegando por los detalles de cada una de las áreas de esta planta, aunque también se puede acceder del mismo modo presionando sobre el área recuadrada en el sinóptico general. A continuación se describen en detalle las pantallas de las distintas zonas de la fábrica 3:

### 2.2.18. Pantalla moldeo F3



En la pantalla de moldeo de la fábrica3 puede apreciarse con más detalle los elementos y las informaciones de la etapa de moldeo.

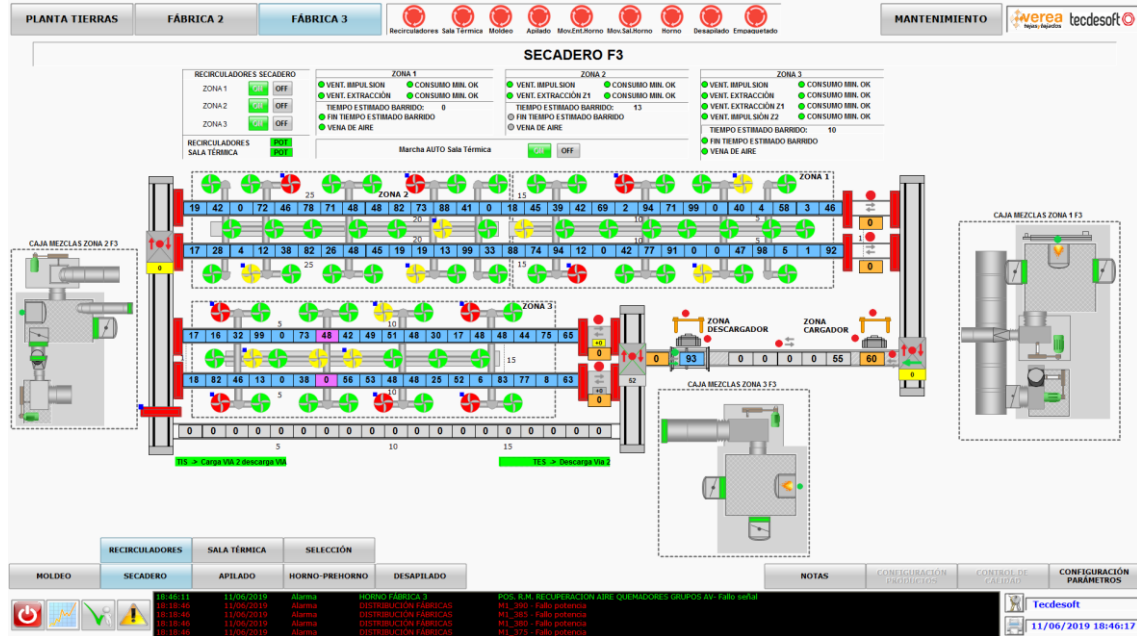
### 2.2.19. Pantalla apilado F3





En la pantalla de apilado se aprecian las bandejas y los motores que las mueven para realizar el apilado de las bandejas antes de ser introducidas al secadero. Como la pantalla es bastante ancha también se aprecian las zonas colindantes del secadero y el prehorno.

## 2.2.20. Pantalla secadero F3

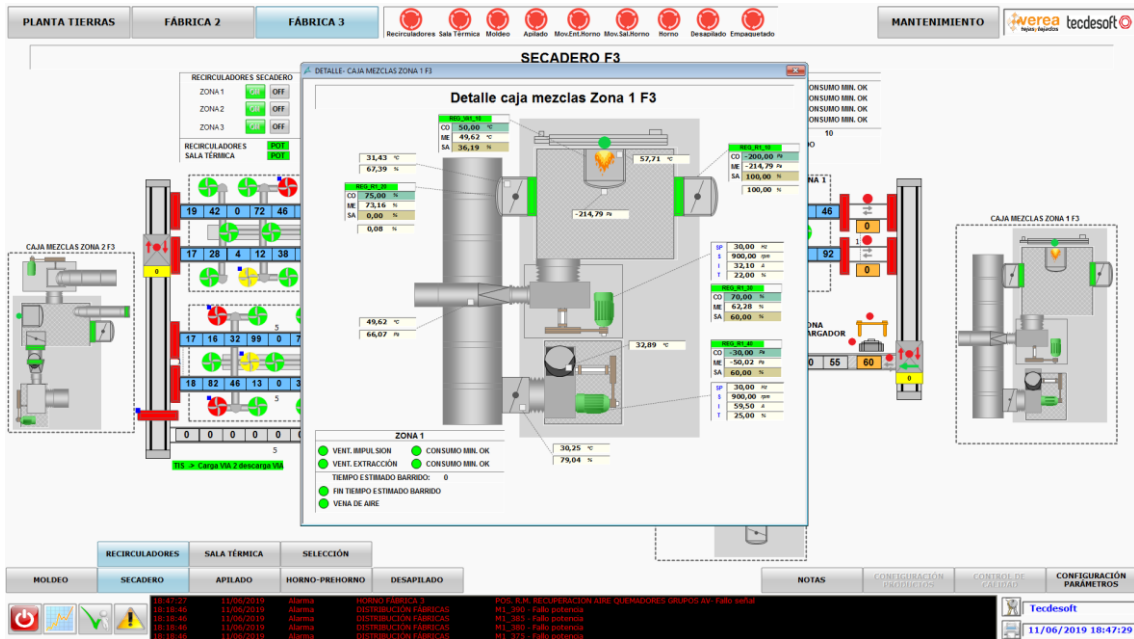


Al pulsar en “Secadero” se abre la pantalla del secadero de la fábrica 3 que está compuesta por los recirculadores, las puertas y los transbordadores de los carros, así como también por las cajas de mezclas. Además se abre un menú superior para poder navegar por el resto de pantallas del secadero.

En la parte superior pueden activarse o desactivarse los recirculadores de cada una de las zonas del secadero, ver el estado de estas y seleccionar la marcha automática de todo el secadero.

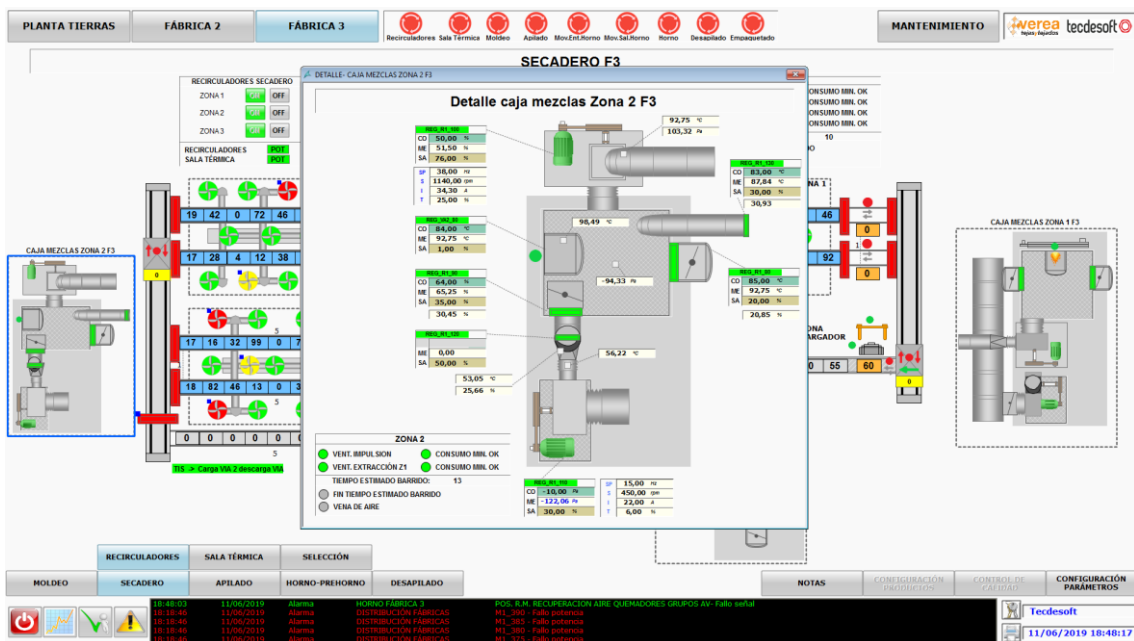
Si se pulsa sobre las cajas de mezclas se abrirá una pantalla donde se muestran todos sus elementos con más detalle, tal y como se muestra en las siguientes 3 pantallas.

## 2.2.21. Pantalla caja de mezclas zona 1 F3



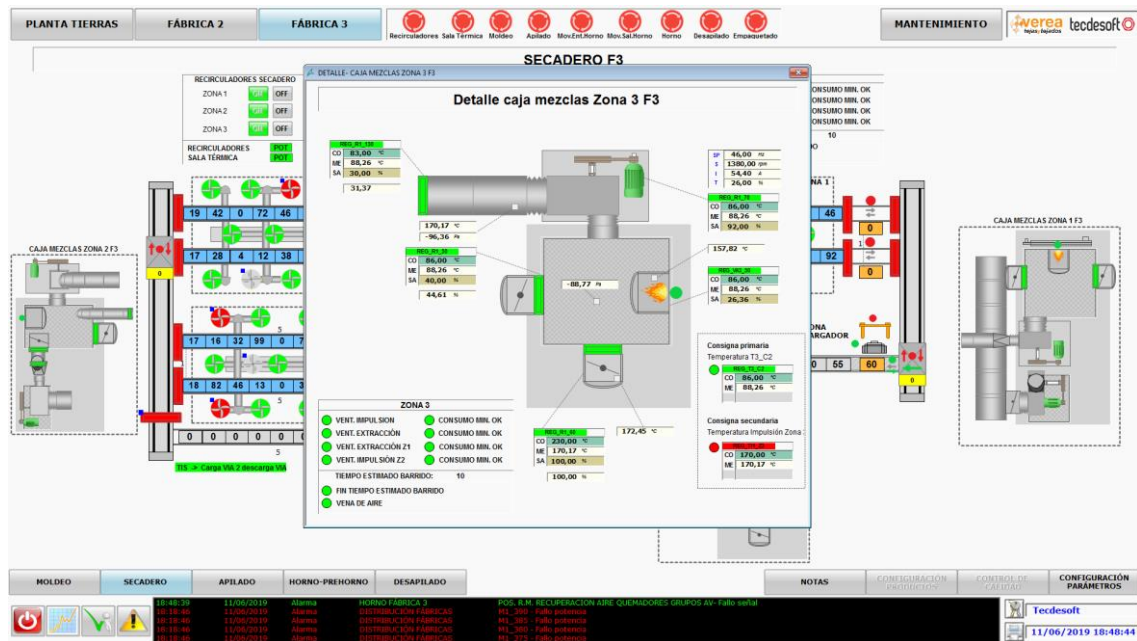
Al pulsar sobre la caja de mezclas de la Zona 1 se abre la pantalla de detalle de la caja de mezclas de esta zona en la que se pueden manejar los reguladores, registros y motores.

## 2.2.22. Pantalla caja de mezclas zona 2 F3



Al pulsar sobre la caja de mezclas de la Zona 2 se abre la pantalla de detalle de la caja de mezclas de esta zona en la que se pueden manejar los reguladores, registros y motores.

## 2.2.23. Pantalla caja de mezclas zona 3 F3



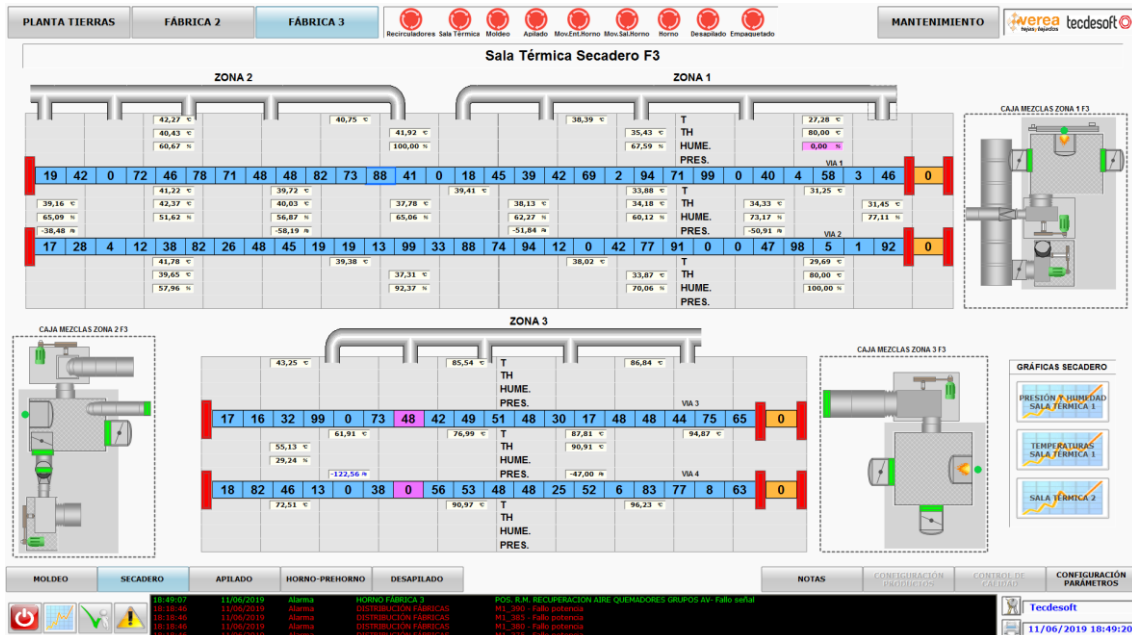
Al pulsar sobre la caja de mezclas de la Zona 3 se abre la pantalla de detalle de la caja de mezclas de esta zona en la que se pueden manejar los reguladores, registros y motores.

## 2.2.24. Pantalla selección secadero F3



Al pulsar sobre el botón “SELECCIÓN” se abre la pantalla de selección de los destinos del secadero mediante la cual se podrán seleccionar las vías, funcionamiento en paralelo, activar y desactivar la Bocina, etc...

## 2.2.25. Pantalla sala térmica



Al pulsar sobre el botón “SALA TÉRMICA” o sobre alguna de las zonas del secadero se abrirá la pantalla de la sala térmica, que representa la vista en perfil del secadero en la que se disponen los valores analógicos de las temperaturas y presiones, además de las estanterías que contienen los productos, las puertas y cajas de mezclas. En esta pantalla también se dispone en la parte derecha, de un menú de selección de gráficas del secadero.

- **PRESIÓN Y HUMEDAD SALA TÉRMICA 1**
- **TEMPERATURAS SALA TÉRMICA 1**
- **SALA TÉRMICA 2**

## 2.2.26. Pantalla gráfica presión y humedad sala térmica 1

Al pulsar sobre el botón “PRESIÓN Y HUMEDAD SALA TÉRMICA 1” desde la pantalla de la sala térmica se visualiza la pantalla de presión y humedad de la sala térmica 1.



En esta pantalla se reflejan los valores analógicos de la humedad y presiones correspondientes a las sondas que están distribuidas longitudinalmente por el secadero. En el eje de abscisas se representa la posición donde está colocada la sonda y en el eje de coordenadas su valor analógico.

En la gráfica se muestran las siguientes curvas:

- **H\_der** (Verde): Curva de humedad de la parte derecha del secadero
- **H\_cen** (Rojo): Curva de humedad de la parte central del secadero
- **P\_cen** (Naranja discontinuo): Curva de presión de la parte central del secadero
- **H\_izq** (Azul): Curva de humedad de la parte izquierda del secadero

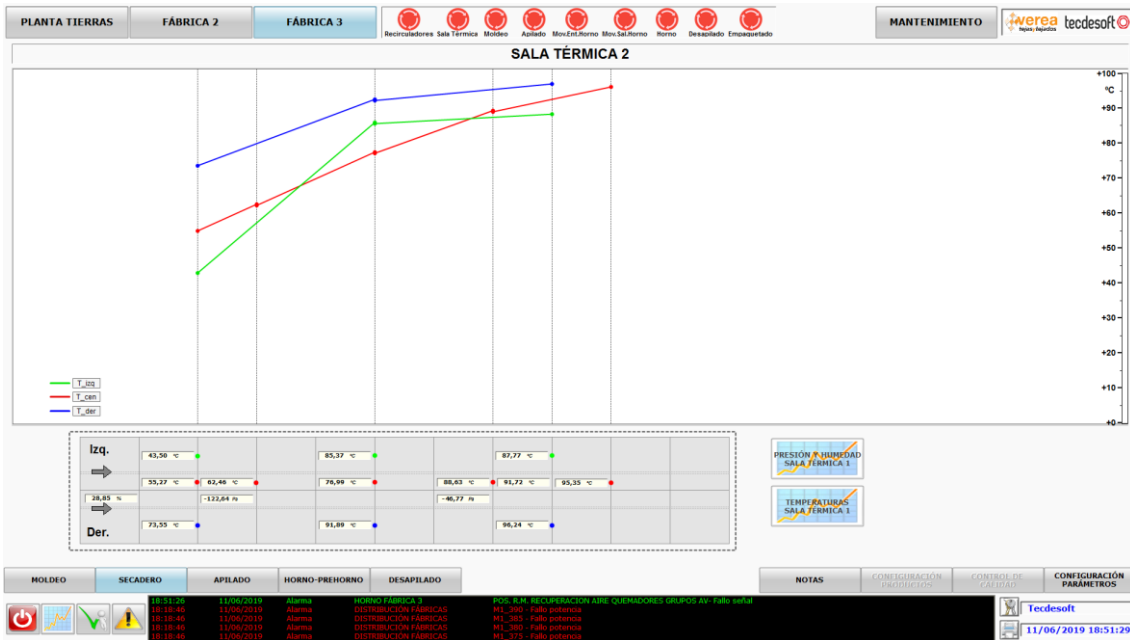
Si el operador posiciona el cursor encima de un segmento de la curva que une dos puntos se desplegará un cartel con el valor de los dos puntos.

En el margen inferior izquierdo aparecen dos botones para visualizar las gráficas de temperatura sala térmica 2 y sala térmica 2.

Si se pulsa sobre el perfil del secadero se vuelve a la pantalla de la sala térmica.

## 2.2.27. Pantalla gráfica sala térmica 2 F3

Al pulsar sobre el botón "SALA TÉRMICA 2" desde la pantalla de la sala térmica se visualiza la pantalla de la sala térmica 2.



En esta pantalla se reflejan los valores analógicos de las temperaturas correspondientes a las sondas que están distribuidas longitudinalmente por la zona 2 del secadero. En el eje de abscisas se representa la posición donde está colocada la sonda y en el eje de coordenadas su valor analógico.

En la gráfica se muestran las siguientes curvas:

- **T\_izq** (Verde): Curva de temperatura de la parte izquierda de la zona 2 del secadero.
- **T\_cen** (Rojo): Curva de temperatura de la parte central de la zona 2 del secadero.
- **T\_der** (Azul): Curva de temperatura de la parte derecha de la zona 2 del secadero.

Si el operador posiciona el cursor encima de un segmento de la curva que une dos puntos se desplegará un cartel con el valor de los dos puntos.

En el margen inferior derecho aparecen dos botones para visualizar las gráficas de la presión y humedad de la sala térmica 1 y temperaturas sala térmica 1.

Si se pulsa sobre el perfil del secadero se vuelve a la pantalla de la sala térmica.

## 2.2.28. Pantalla gráfica temperaturas sala térmica 1 F3

Al pulsar sobre el botón "TEMPERATURAS SALA TÉRMICA 1" desde la pantalla de la sala térmica se visualiza la pantalla de temperaturas de la sala térmica 1.



En esta pantalla se reflejan los valores analógicos de las temperaturas correspondientes a las sondas que están distribuidas longitudinalmente por la zona 1 del secadero. En el eje de abscisas se representa la posición donde está colocada la sonda y en el eje de coordenadas su valor analógico.

En la gráfica se muestran las siguientes curvas:

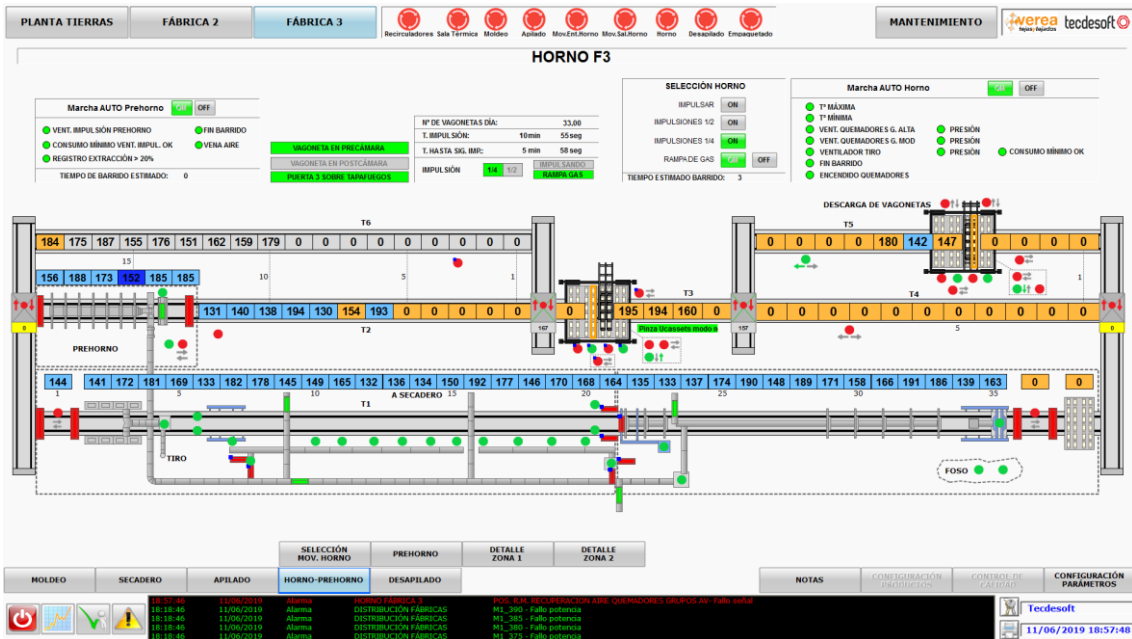
- **T\_der** (Turquesa discontinuo): Curva de temperatura de la parte derecha de la zona 1 del secadero.
- **TH\_der** (Verde): Curva de temperatura de la parte derecha de la zona 1 del secadero.
- **T\_cen** (Naranja discontinuo): Curva de temperatura de la parte central de la zona 1 del secadero.
- **TH\_cen** (Rojo): Curva de temperatura de la parte central de la zona 1 del secadero.
- **T\_izq** (Magenta discontinuo): Curva de temperatura de la parte izquierda de la zona 1 del secadero.
- **TH\_izq** (Azul): Curva de temperatura de la parte izquierda de la zona 1 del secadero.

Si el operador posiciona el cursor encima de un segmento de la curva que une dos puntos se desplegará un cartel con el valor de los dos puntos.

En el margen inferior izquierdo aparecen dos botones para visualizar las gráficas de la presión y humedad de la sala térmica 1 y sala térmica 2.

Si se pulsa sobre el perfil del secadero se vuelve a la pantalla de la sala térmica.

## 2.2.29. Pantalla horno-prehorno F3

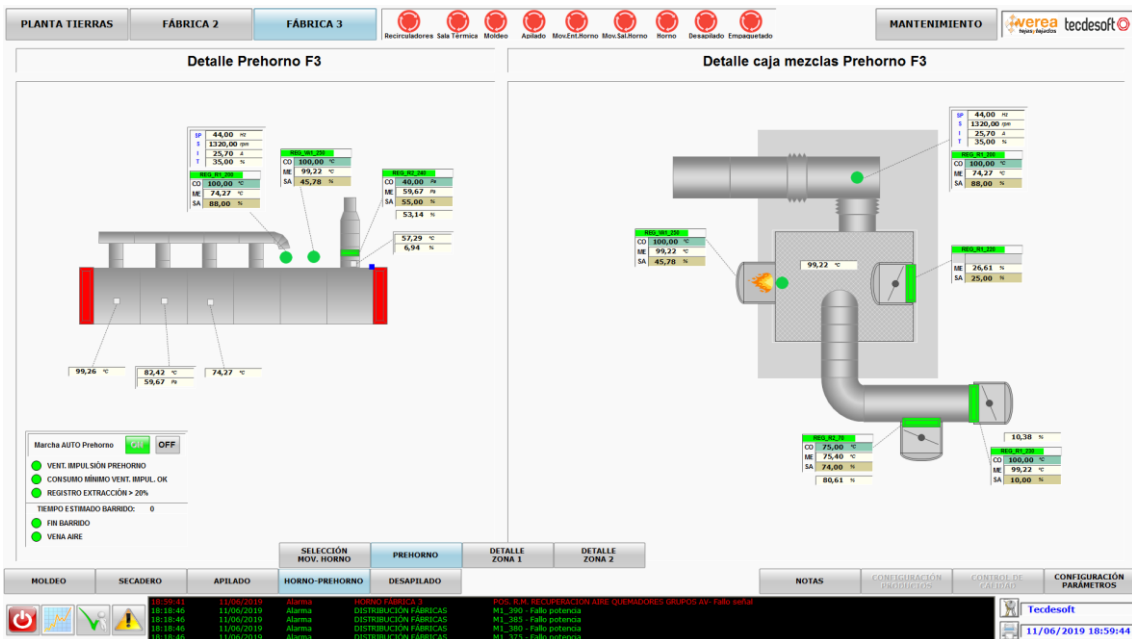


Para acceder a la pantalla de Horno-Prehorno F3 puede realizarse pulsando el botón “HORNO-PREHORNO” del el menú inferior de la fábrica 3 o pulsando en el área del horno desde el sinóptico general de la fábrica 3.

En esta pantalla pueden apreciarse y manipularse los distintos elementos que componen el horno como las puertas, registros, grupos de alta velocidad, grupos modulantes, transbordadores y motores.

En la parte superior se dispone de una botonera que permite realizar selecciones para gestionar la impulsión del horno y también controlarlo en modo Automático y Manual.

## 2.2.30. Pantalla prehorno F3





Para acceder a la pantalla del prehorno de la fábrica 3 puede realizarse pulsando el botón “PREHORNO” desde el menú desplegable que aparece al haber pulsado “HORNO-PREHORNO” o pulsando sobre el área del prehorno desde la pantalla de horno-prehornos F3.

Esta pantalla está dividida en dos zonas claramente diferenciadas:

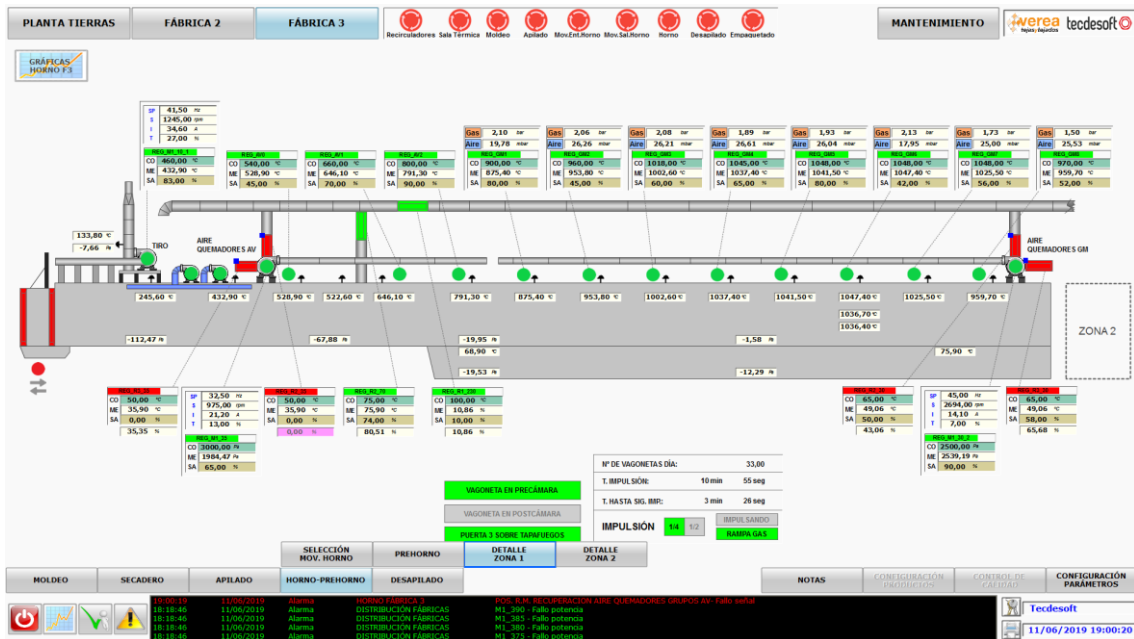
En la parte izquierda se dispone de una vista en perfil del prehorno donde se aprecian las puertas, registros, reguladores y motores que componen el prehorno. Dentro de esta zona hay dos botones que permiten seleccionar el modo de operación automático del prehorno. En la parte derecha se visualiza la caja de mezclas que alimenta el prehorno.

### 2.2.31. Pantalla selección destinos horno F3



Al seleccionar “SELECCIÓN MOV. HORNO” desde el menú desplegable que aparece al pulsar el botón “HORNO-PREHORNO” se visualiza esta ventana emergente desde la cual se gestiona la operación de la entrada y salida del horno.

## 2.2.32. Pantalla zona 1 horno F3



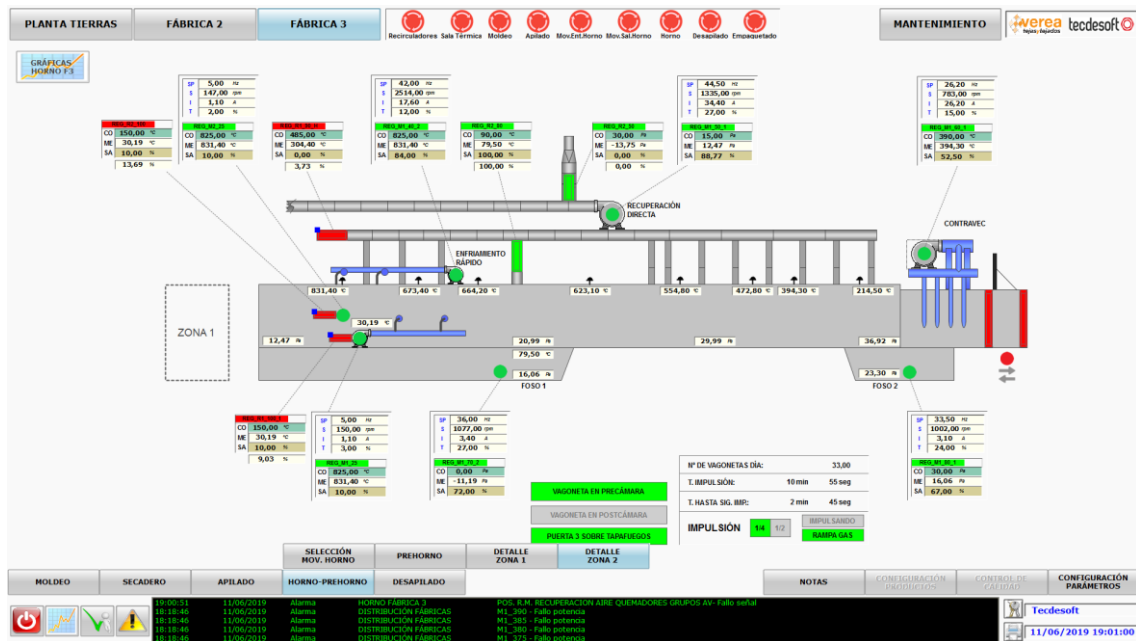
Para acceder a la pantalla de la zona 1 del horno F3 puede realizarse pulsando “DETALLE ZONA1” desde el menú desplegable que aparece al haber pulsado “HORNO-PREHORNO” o pulsando sobre el área de la zona 1 del horno desde la pantalla de horno-prehorno F3.

En esta pantalla se muestra parte del horno visto en perfil, en la cual se reflejan todos los componentes que lo conforman como: los quemadores, grupos de alta velocidad, grupos modulantes, motores, puertas, registros, reguladores y sondas analógicas.

En la parte derecha aparece un área etiquetada como Zona 2, que si pulsamos encima de ella podremos visualizar la parte restante del horno de la fábrica 3.

En la parte superior izquierda se dispone un botón para acceder a las gráficas instantáneas del horno.

## 2.2.33. Pantalla zona 2 horno F3



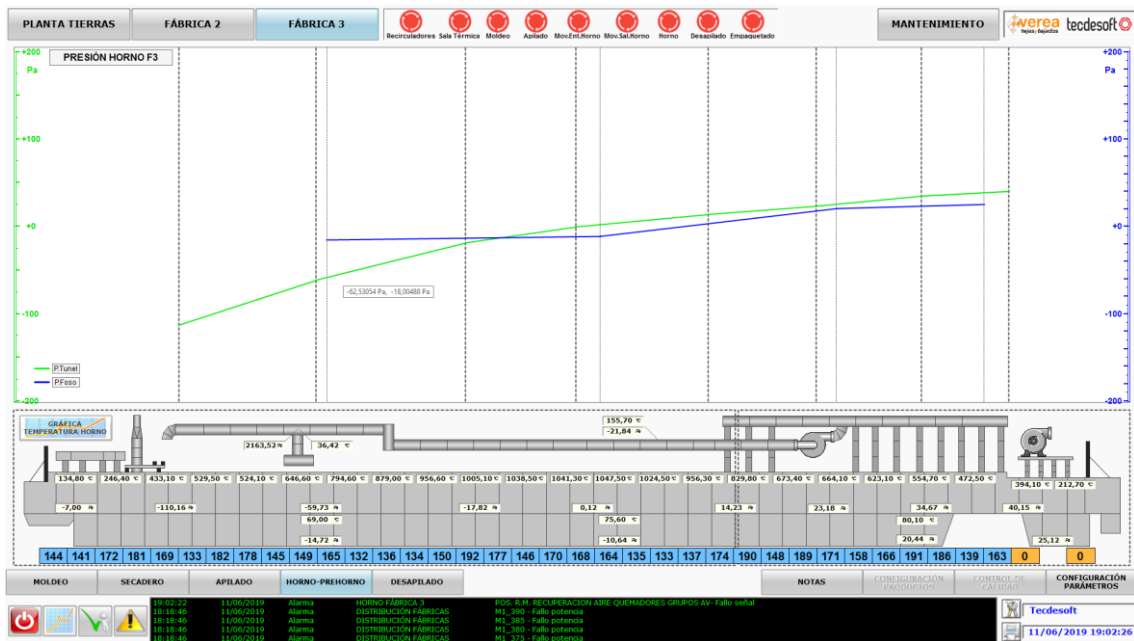
Para acceder a la pantalla de la zona 2 del horno F3 puede realizarse pulsando “DETALLE ZONA2” desde el menú desplegable que aparece al haber pulsado “HORNO-PREHORNO” o pulsando sobre el área de la zona 2 del horno desde la pantalla de horno-prehorno F3.

En esta pantalla se muestra parte del horno visto en perfil, en la cual se reflejan todos los componentes que lo conforman como: los quemadores, grupos de alta velocidad, grupos modulantes, motores, puertas, registros, reguladores y sondas analógicas.

En la parte izquierda aparece un área etiquetada como Zona 1, que si pulsamos encima de ella podremos visualizar la parte restante del horno de la fábrica 3.

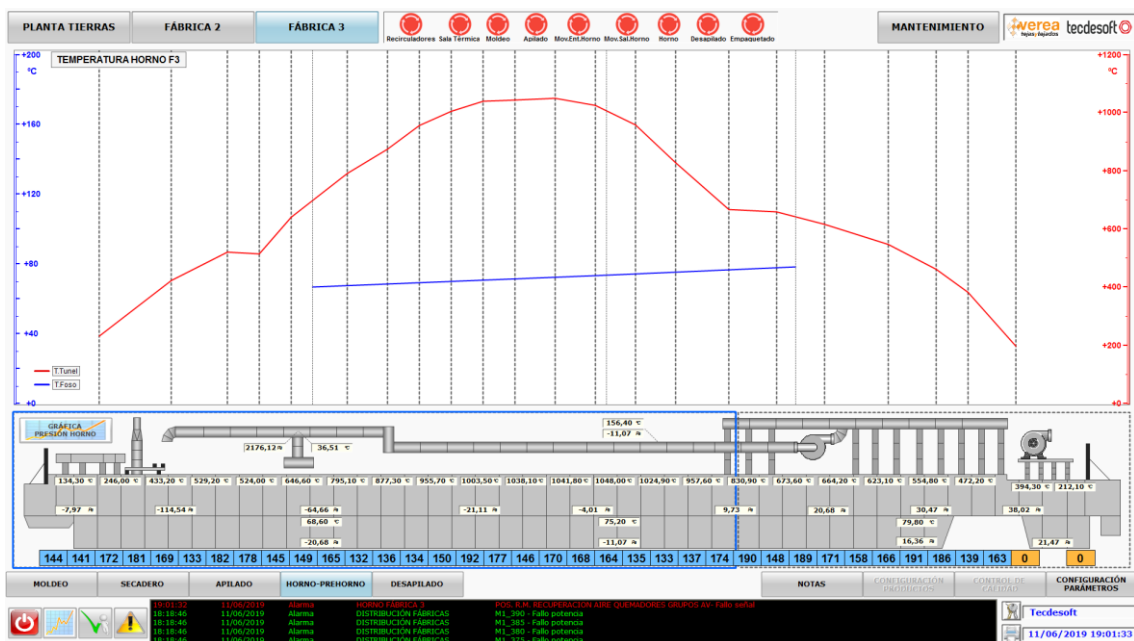
En la parte superior izquierda se dispone un botón para acceder a las gráficas instantáneas del horno.

## 2.2.34. Pantalla gráfica presión horno F3



Al pulsar sobre el botón de “GRAFICAS HORNO F3” desde las pantallas de las Zonas 1 o 2 del horno F3 se accede a la pantalla de la gráfica de presión del horno F3, la cual está compuesta por un área en la cual se representarán las curvas de presión del túnel y foso, en verde y azul respectivamente y en la parte inferior se dispone de una vista en perfil con los valores analógicos de presión y temperatura distribuidos por todo el horno, así como también las vagonetas que contiene en ese momento. Se dispone de un botón para intercambiar entre la gráfica de presiones y temperaturas.

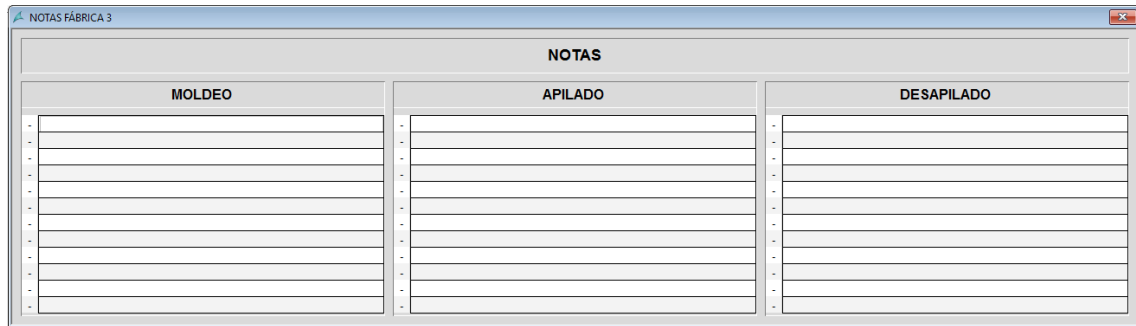
## 2.2.35. Pantalla gráfica temperatura horno F3



Si desde la pantalla de la gráfica presión horno f3 se pulsa el botón “GRÁFICA TEMPERATURAS HORNO” se representan en el área gráfica las dos curvas de temperatura del horno. Temperatura túnel (en rojo) y temperatura foso (en azul).

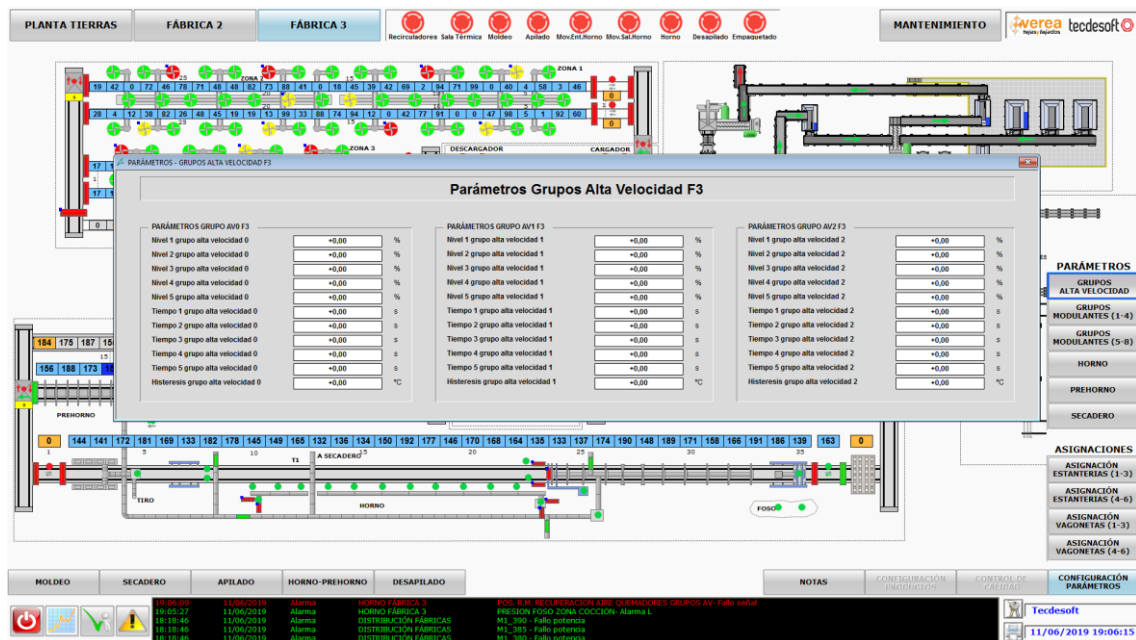
Se dispone de un botón para intercambiar entre la gráfica de presiones y temperaturas.

### 2.2.36. Pantalla notas F3



Desde el menú inferior de la fábrica 3 si se pulsa el botón “NOTAS” se abre una ventana emergente que contiene 12 campos de texto para poder fijar notas y que los operarios puedan intercambiar información.

### 2.2.37. Pantalla parámetros F3

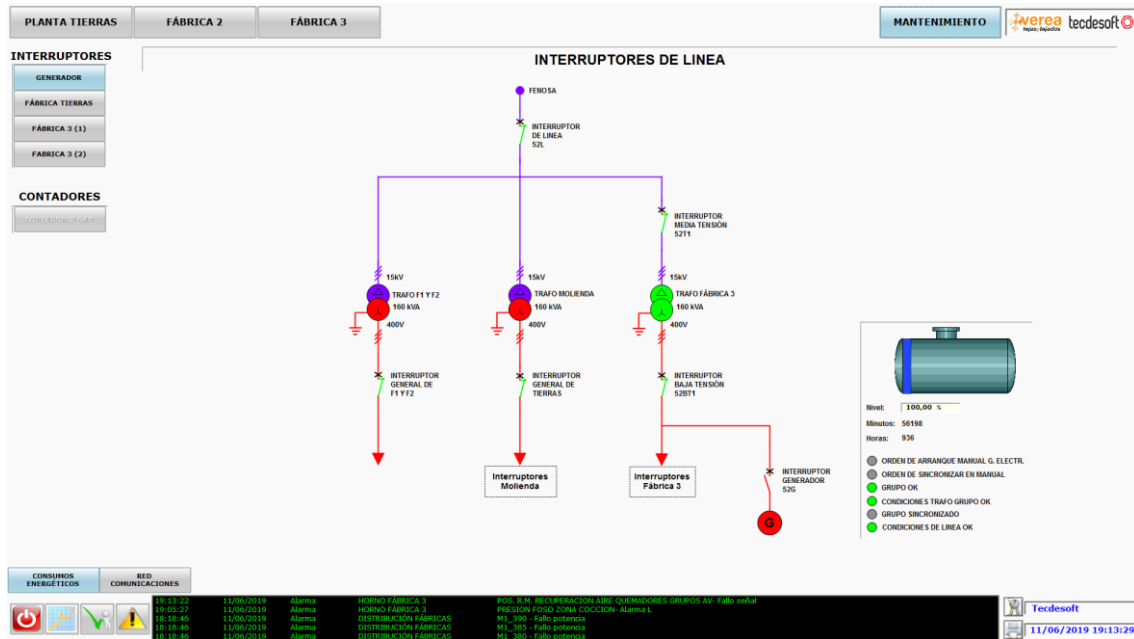


Al pulsar el botón de “CONFIGURACIÓN DE PARÁMETROS” situado en el menú inferior de la fábrica 3 se despliega un menú emergente desde el cual se podrá seleccionar la pantalla para cambiar los parámetros y asignaciones correspondientes a las distintas áreas de proceso de la fábrica 3.

### 2.2.38. Pantallas de consumos energéticos

Para acceder a las pantallas de mantenimiento hay que pulsar el botón “MANTENIMIENTO” situado en el menú superior. A continuación se explican detalladamente cada una de las pantallas.

## 2.2.38.1. Pantalla interruptores de línea



Al pulsar el botón “MANTENIMIENTO” situado en el menú superior se visualiza la pantalla de interruptores de línea que comienza con la acometida de energía eléctrica con la compañía y el interruptor general de línea.

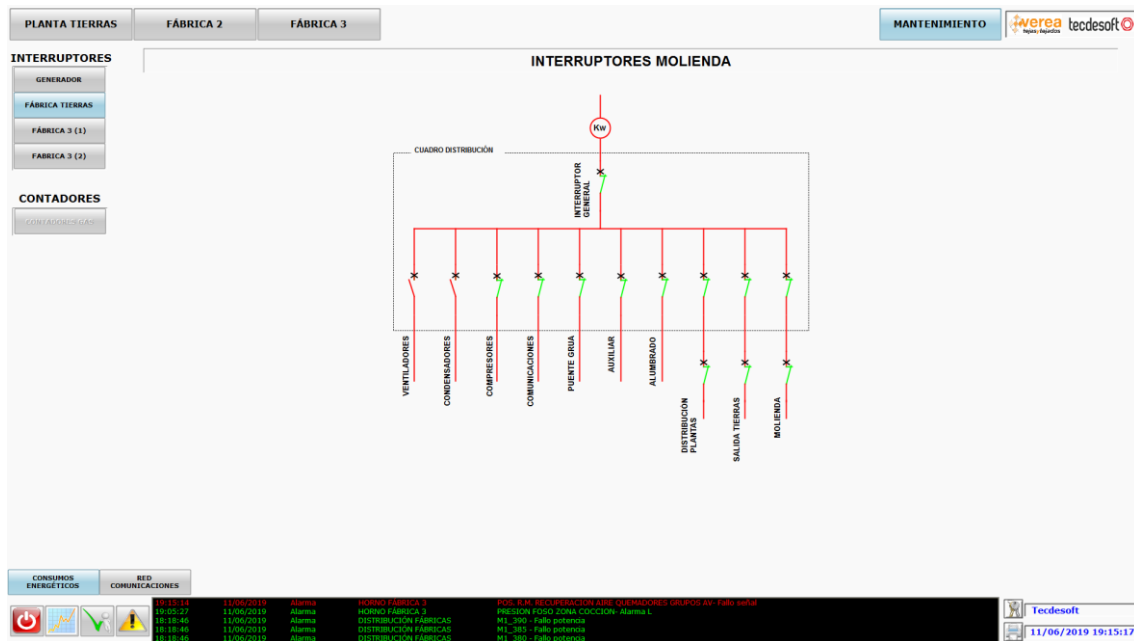
Aguas abajo del interruptor general de línea se divide el circuito en tres ramas que alimentan los transformadores de 15kV/400V y los respectivos circuitos:

- Fábrica 1 y 2
- Planta de tierras (Molienda)
- Fábrica 3

En este esquema general también se refleja el Generador con su interruptor y tanque de combustible por si fuese necesario aportar energía a las instalaciones cuando la compañía suministradora falle en el suministro.

Tanto los transformadores como el generador tienen una ventana donde se muestran las informaciones de su estado que aparecerá si pulsamos encima de cada elemento.

## 2.2.38.2. Pantalla interruptores molienda



En la pantalla de interruptores molienda se muestra el esquema de potencia que alimenta la planta de tierras.

Todos los interruptores muestran su estado: abierto (Rojo), cerrado (Verde).

Los interruptores que están dentro del área del cuadro de distribución no tienen mando pero los que se encuentran fuera si pueden manejarse a través del SCADA si pulsamos encima de ellos para que aparezca la ventana de mando.

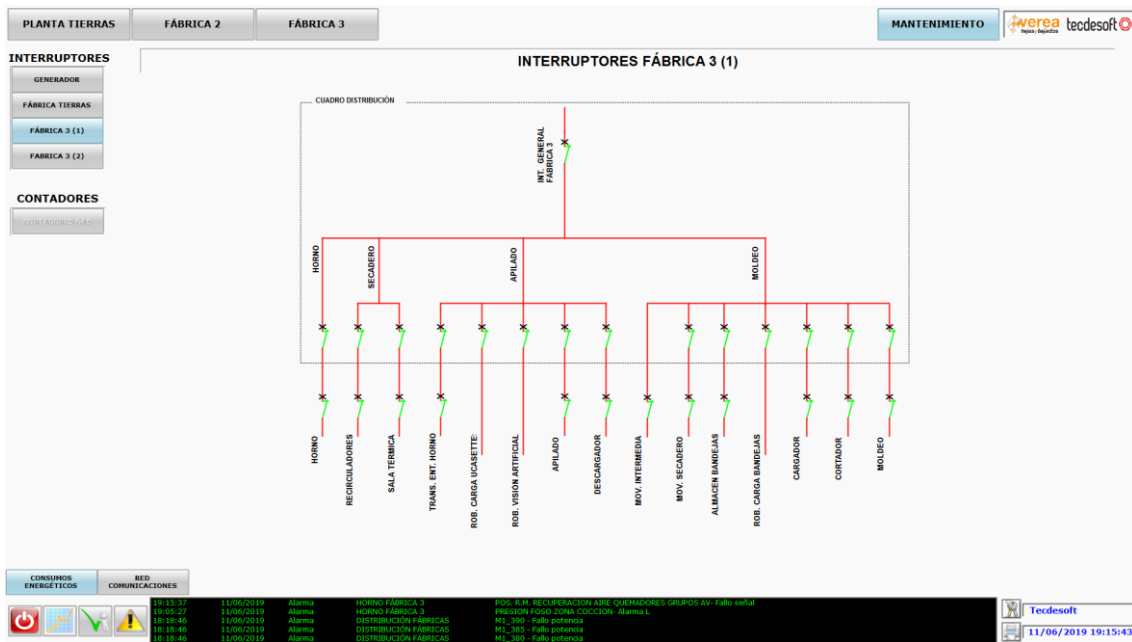
Desde esta pantalla se pueden manejar remotamente los siguientes interruptores:

- Distribución plantas
- Salida tierras
- Molienda

## 2.2.38.3. Pantalla interruptores fabrica 3-1

En la pantalla de interruptores fábrica 3-1 se muestra el esquema de potencia que alimenta las siguientes secciones de la fábrica 3:

- Horno
- Secadero
- Apilado
- Moldeo



Todos los interruptores muestran su estado: abierto (Rojo), cerrado (Verde).  
 Los interruptores que están dentro del área del cuadro de distribución no tienen mando pero los que se encuentran fuera si pueden manejarse a través del SCADA si pulsamos encima de ellos para que aparezca la ventana de mando.

Desde esta pantalla se pueden manejar remotamente los siguientes interruptores:

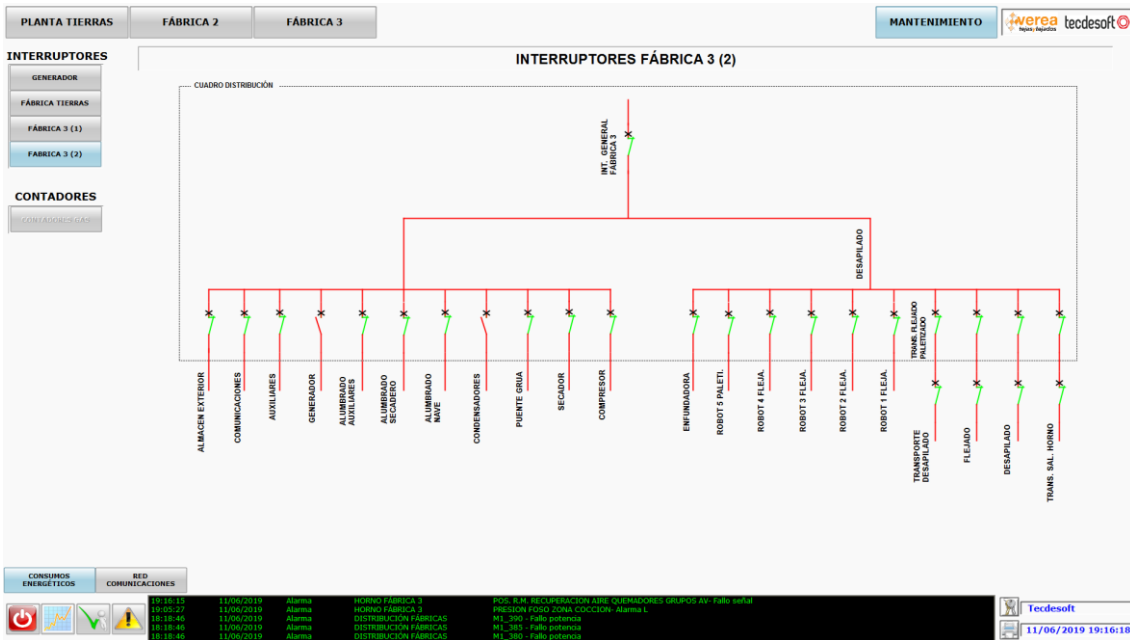
- Horno
- Recirculadores
- Sala térmica
- Transbordadores entrada horno
- Apilado
- Descargador
- Movimentación intermedia
- Movimentación secadero
- Almacén bandejas
- Cargador
- Cortador
- Moldeo

#### 2.2.38.4. Pantalla interruptores fabrica 3-2

En la pantalla de interruptores fábrica 3-2 se muestra el esquema de potencia que alimenta las siguientes secciones de la fábrica 3:

- Alumbrado
- Desapilado





Todos los interruptores muestran su estado: abierto (Rojo), cerrado (Verde). Los interruptores que están dentro del área del cuadro de distribución no tienen mando pero los que se encuentran fuera si pueden manejarse a través del SCADA si pulsamos encima de ellos para que aparezca la ventana de mando.

Desde esta pantalla se pueden manejar remotamente los siguientes interruptores:

- Transporte desapilado
- Flejado
- Desapilado
- Transbordadores salida Horno

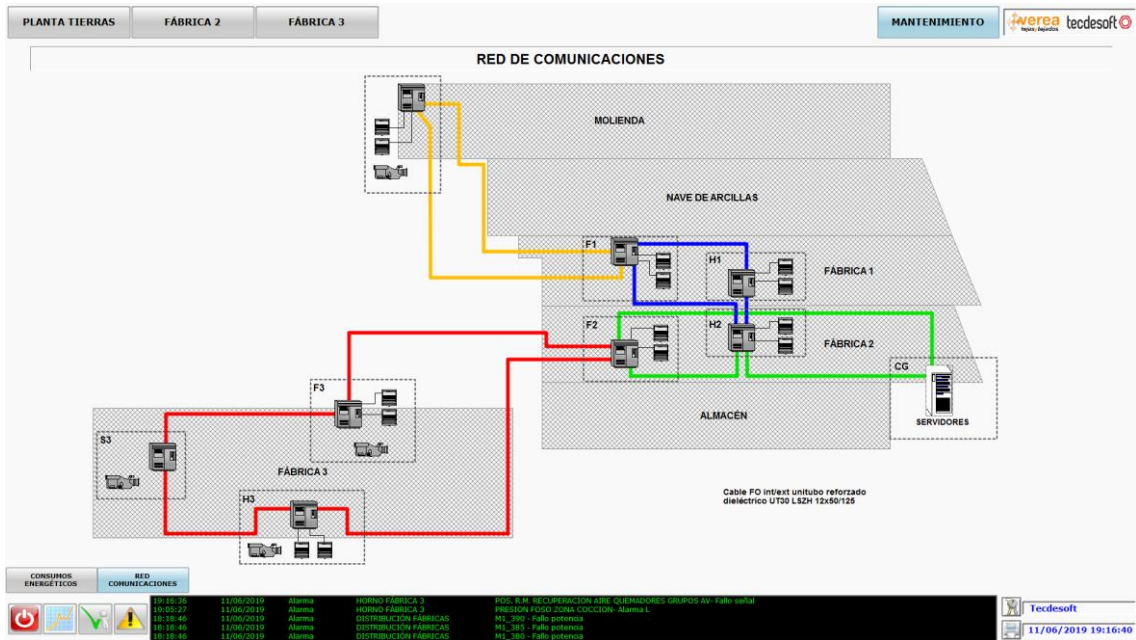
## 2.2.39. Pantallas red de comunicaciones

Para acceder a las pantallas de la red de comunicaciones hay que pulsar el botón “RED COMUNICACIONES” situado en el menú inferior.

A continuación se explican detalladamente cada una de las pantallas de la red de comunicaciones.

### 2.2.39.1. Pantalla red general comunicaciones

Es la pantalla que nos encontramos al haber pulsado el botón “RED COMUNICACIONES” situado en el menú inferior.



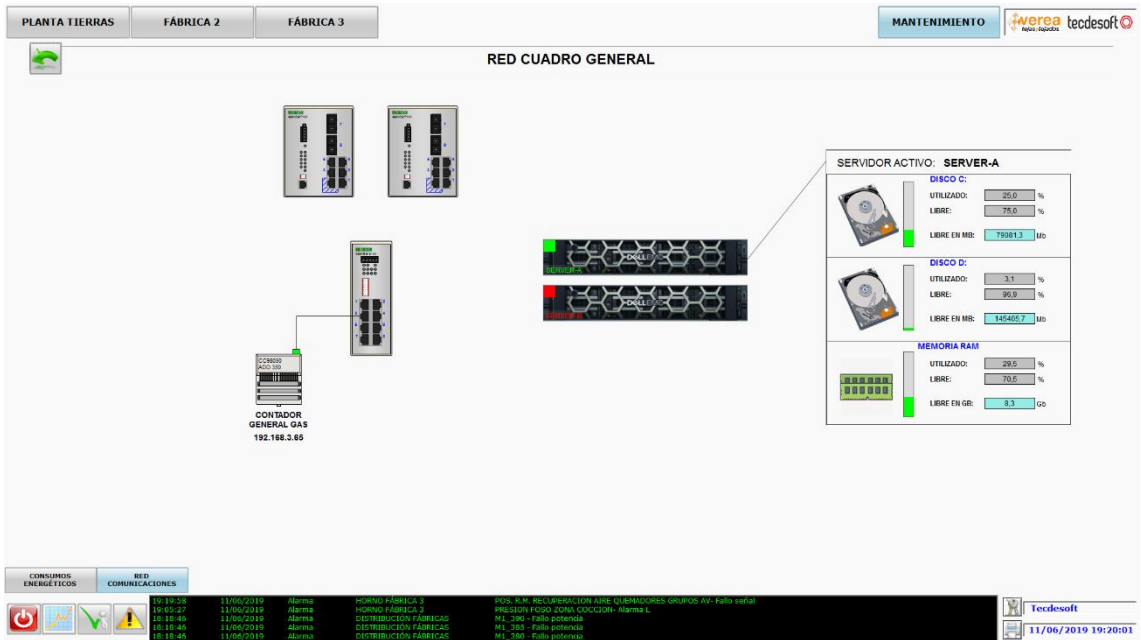
En esta pantalla se muestran como están distribuidos los anillos de la red de comunicación del proceso productivo.

Desde esta pantalla puede accederse a los detalles de la red de producción de cada una de las secciones de la fábrica pulsando sobre las áreas definidas alrededor de los cuadros de comunicaciones que se representan en la imagen.

## 2.2.40. Pantalla servidores

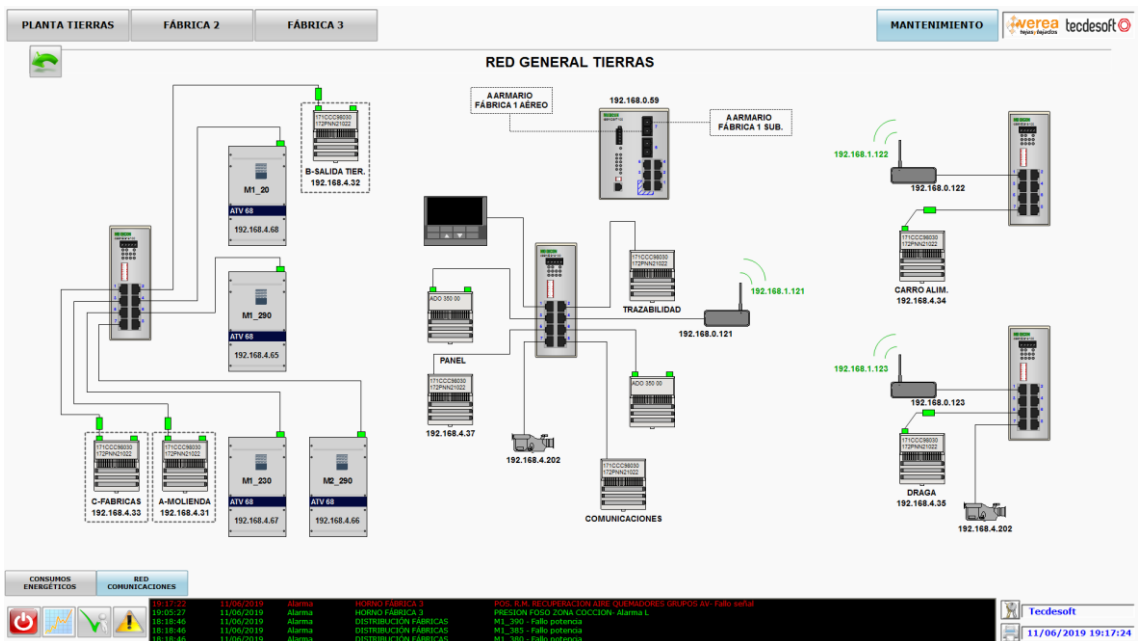
En la pantalla de servidores se muestran los componentes principales del rack CPD que está compuesto por:

- Contador general de GAS
- Server A
- Server B



En esta pantalla puede verse cuál de los dos servidores del SCADA está activo así como la información de espacio en los discos y uso de memoria RAM del servidor que se encuentre activo. Si se pulsa el botón con la flecha “back” situada en la esquina superior izquierda se vuelve a la pantalla anterior.

## 2.2.41. Pantalla red general de tierras



En la pantalla de la red general de tierras se la distribución los equipos instalados en el cuadro. Los PLCs tienen dos recuadros que muestran el estado de las comunicaciones. El recuadro izquierdo hace referencia a la comunicación por Modbus TCP/IP, mientras que el recuadro derecho hace referencia a la comunicación por Modbus Plus (MB+) con las tarjetas de comunicación y otros equipos.

Los variadores tienen solo un recuadro que hace referencia al estado de las comunicaciones por Modbus TCP/IP.

Si el recuadro está en verde, la comunicación se realiza correctamente, sin embargo si está rojo quiere decir que no se está llevando a cabo correctamente la transmisión de datos en ese nodo y hay un problema de comunicaciones.

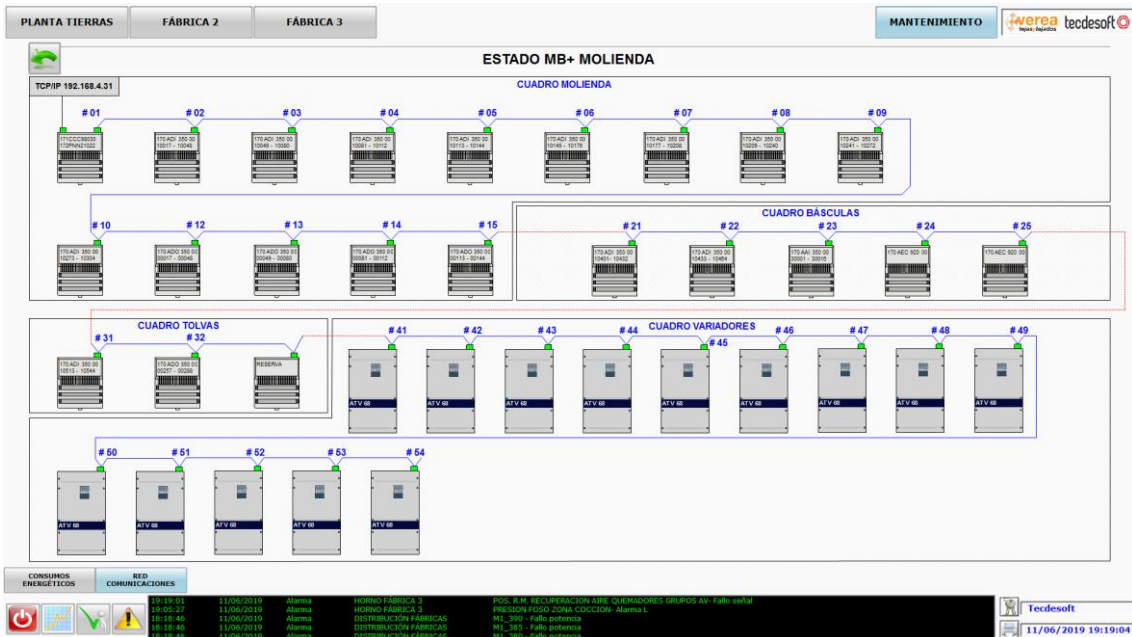
### 2.2.41.1. Pantalla estado MB+ salida tierras y distribución fábricas

Icono	Fecha	Estado	Descripción	Detalles
[Icono]	11/06/2019	Alarma	COMUNICACION MB+ SALIDA TIERRAS	Fallo potencia
[Icono]	11/06/2019	Alarma	COMUNICACION MB+ SALIDA TIERRAS	Fallo potencia
[Icono]	11/06/2019	Alarma	DISTRIBUCIÓN FÁBRICAS	Fallo potencia
[Icono]	11/06/2019	Alarma	DISTRIBUCIÓN FÁBRICAS	Fallo potencia
[Icono]	11/06/2019	Alarma	DISTRIBUCIÓN FÁBRICAS	Fallo potencia

En la pantalla de estado MB+ Salida tierras y distribución fábricas puede verse el estado de las comunicaciones por Modbus + entre los PLCs y las tarjetas de comunicaciones.

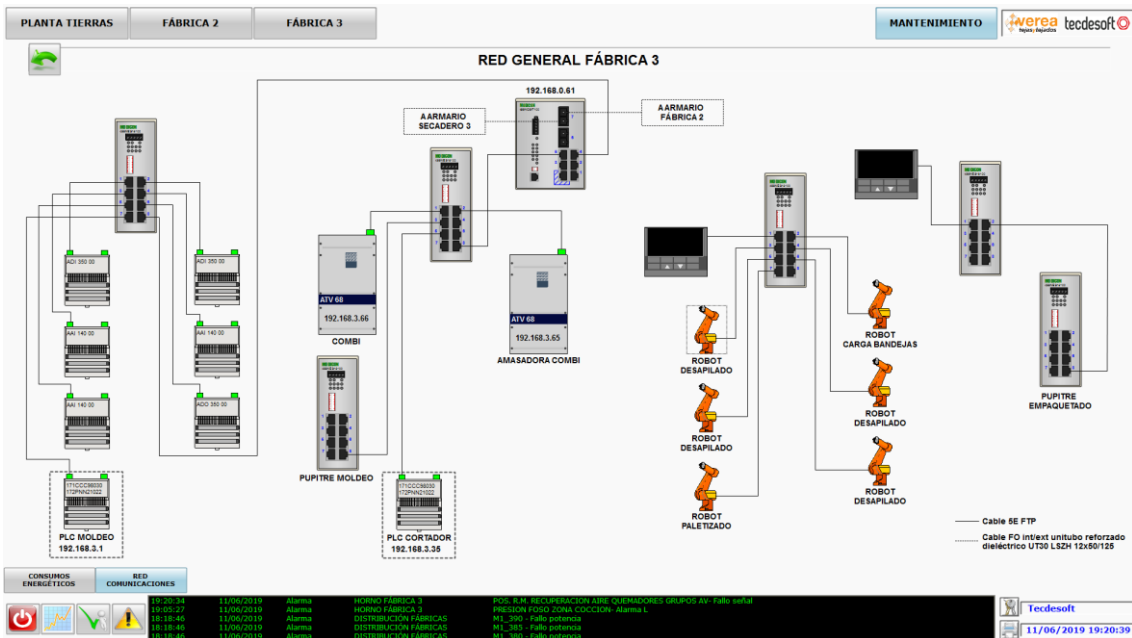
Además si falla la comunicación por TCP/IP en el PLC, el cartel que contiene la IP se pondrá en color Magenta.

## 2.2.41.2. Pantalla estado MB+ molienda



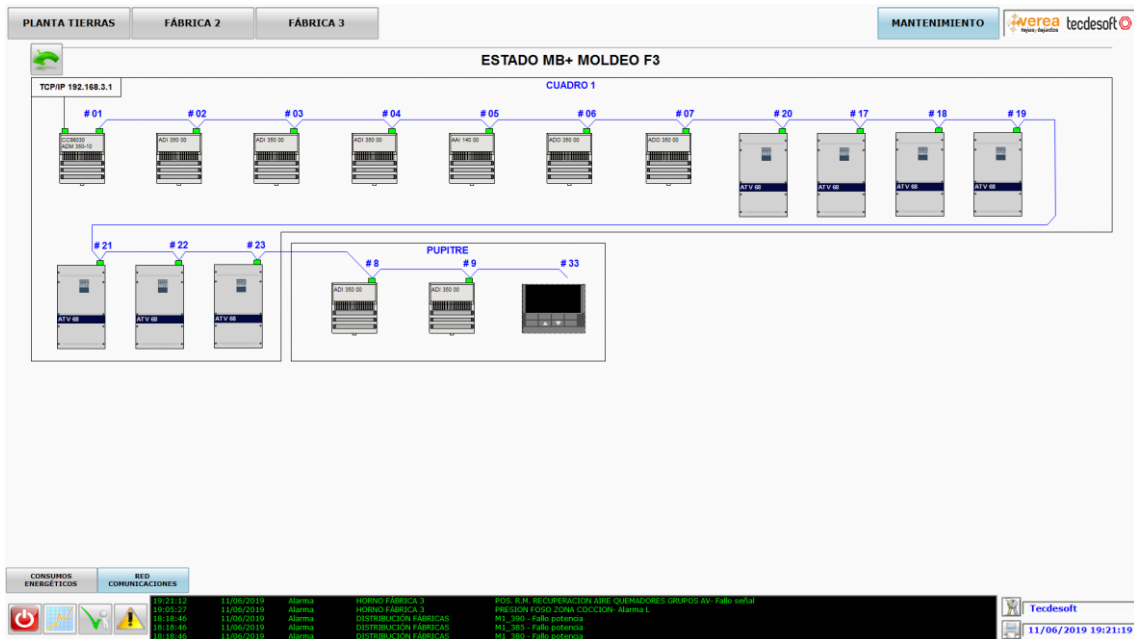
Al igual que la pantalla anterior se muestra el estado de la red Modbus Plus del PLC de molienda con las tarjetas de comunicación y los variadores. Además si falla la comunicación por TCP/IP en el PLC, el cartel que contiene la IP se pondrá en color Magenta.

## 2.2.41.3. Pantalla red general fábrica 3

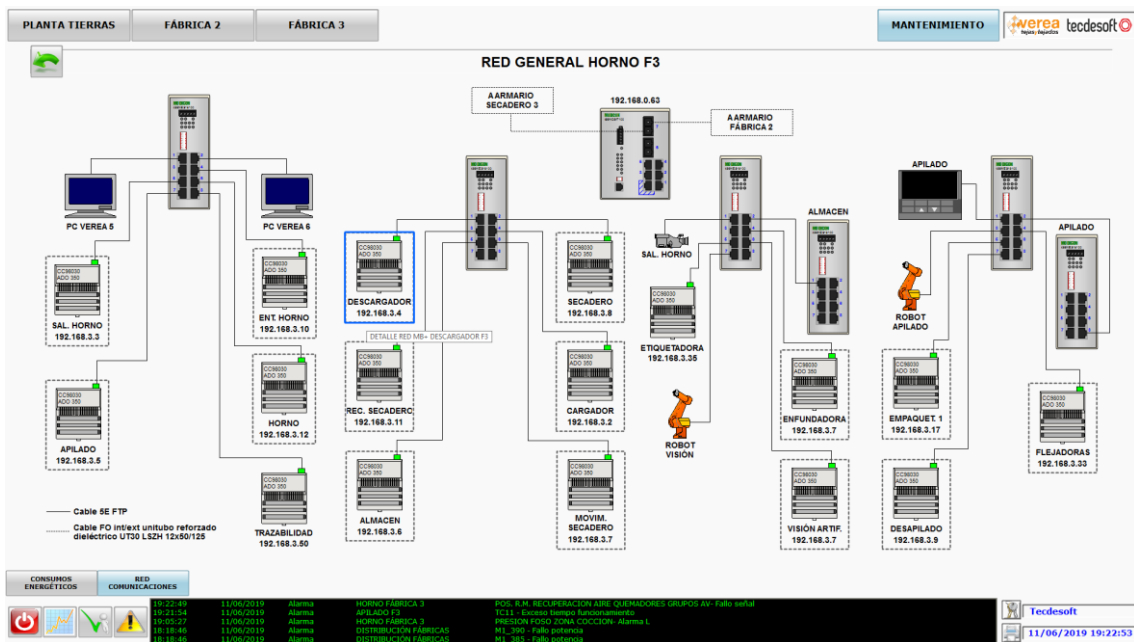


En esta pantalla se muestran los principales componentes de la red de comunicación de la fábrica 3.

## 2.2.41.4. Pantalla estado MB+ moldeo 3



## 2.2.41.5. Pantalla red general horno F3



Las pantallas de la red de comunicaciones se ampliará en un futuro con el resto de áreas de las fábricas.

## 2.3 ANEXO III - Manual de configuración

### 2.3.1. Procedimiento de instalación y configuración del software WinCC en los equipos.

En este apartado detallado a continuación se explica los pasos a seguir para instalar de forma correcta el software de Siemens WinCC versión 7.4 service pack 1 en los sistemas operativos de las máquinas virtuales, con la instalación del último update disponible para el software. Hay varios tipos de instalación: los SCADAs de los servidores, estación de ingeniería y la de los clientes.

Lista de ficheros necesarios para realizar la instalación:

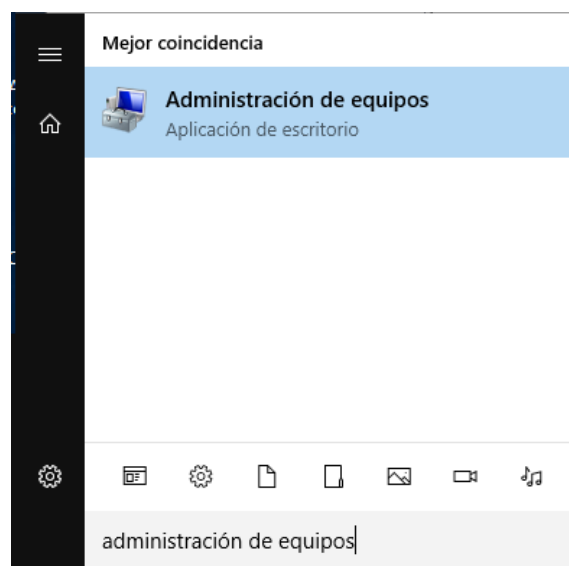
Nombre del archivo	Descripción	Tamaño
WinCC V7.4.iso	Archivos de la aplicación WinCC *	8 GB
SIMATIC NET.iso	Archivos de la aplicación SimaticNet *	1.3 GB
WinCC_V74_SP1_Upd10.001	Archivos comprimidos Update 10	929 MB
WinCC_V74_SP1_Upd10.exe	Ejecutable para la instalación del Update 10	2.73 MB

\* El software indicado se proporciona con un CD de instalación excepto el update 10 que se descargara de la página web de Siemens.

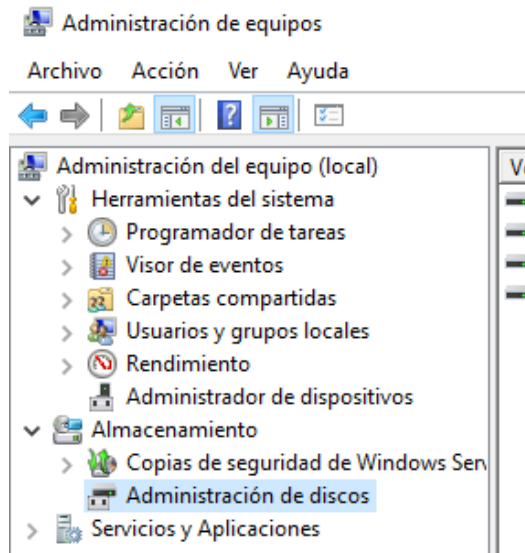
#### 2.3.1.1. Particionado del disco duro

El primer paso va a ser crear una partición del disco C:\ de aproximadamente la mitad de capacidad del mismo, a la que llamaremos D:\.

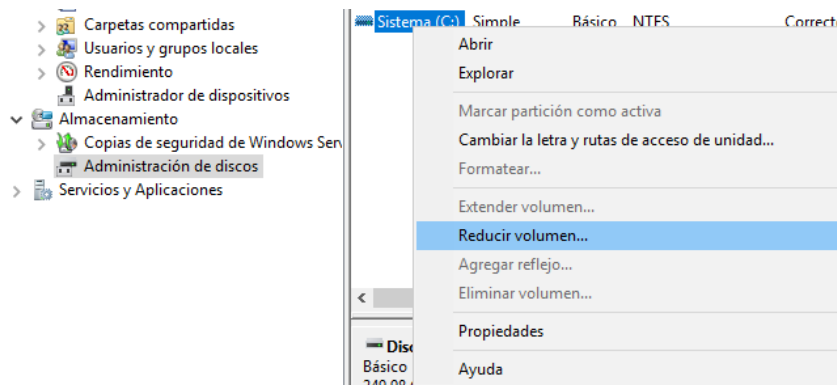
Se busca en el menú de inicio de Windows la aplicación “Administración de equipos”:



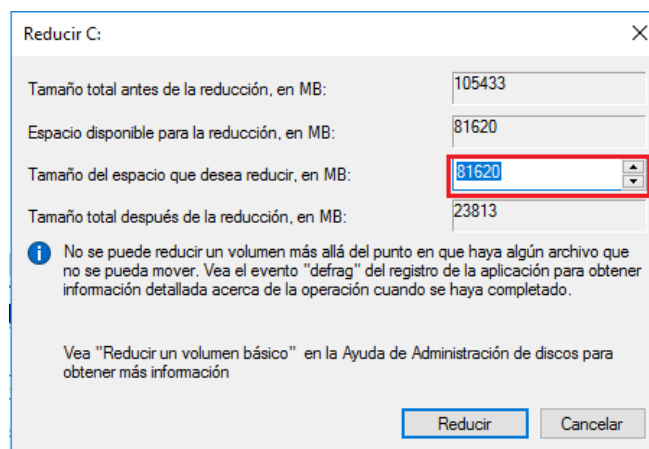
Una vez abierta la nueva ventana se despliega el menú lateral “Almacenamiento” y se hace clic en “Administración de discos”:



A continuación, aparecen los discos del sistema en la pantalla central. Encima del disco C:\ se pulsa botón derecho y se hace clic en la opción “Reducir Volumen”:



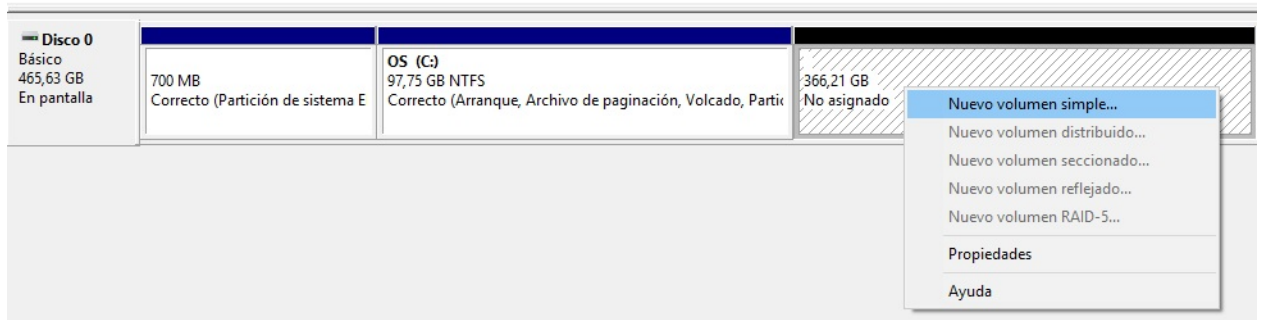
Cuando se abra la nueva ventana, se asigna el tamaño a reducir (que es el tamaño que se va a aplicar a la partición) en MB. Es recomendable dejar un espacio de aproximadamente 100GB al disco C:



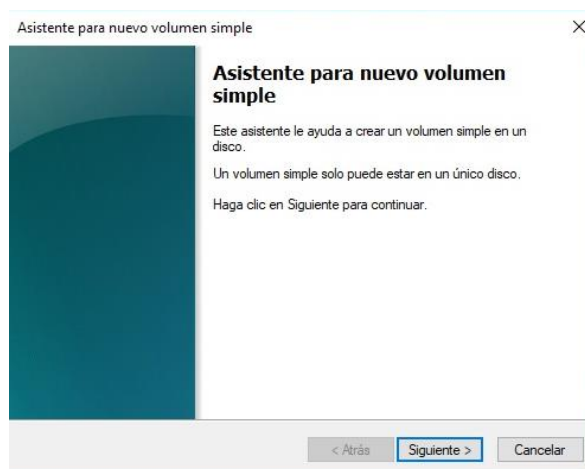
Para finalizar se hace clic en “Reducir” para que se apliquen los cambios. Una vez terminado el proceso, aparecerá en el menú del medio la nueva partición creada (sombreado en negro).



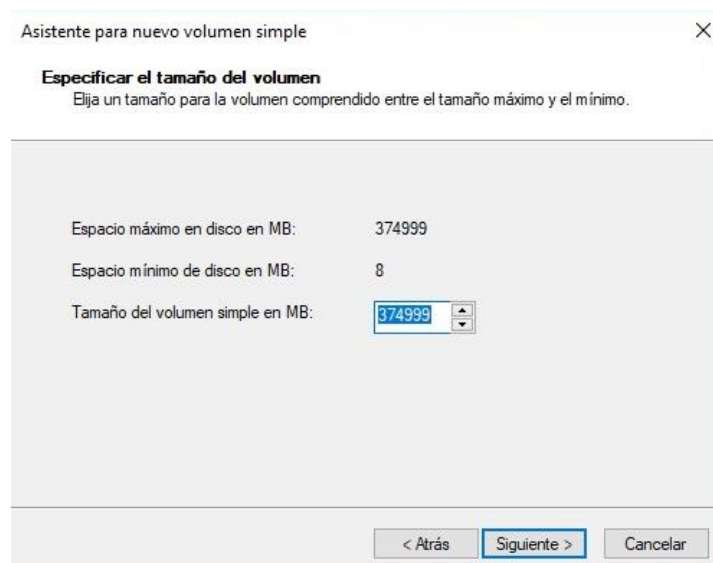
Es necesario dar formato a la partición, para lo que se pulsa botón derecho sobre la zona sombreada correspondiente a la nueva partición y se hace clic en “Nuevo volumen simple...”



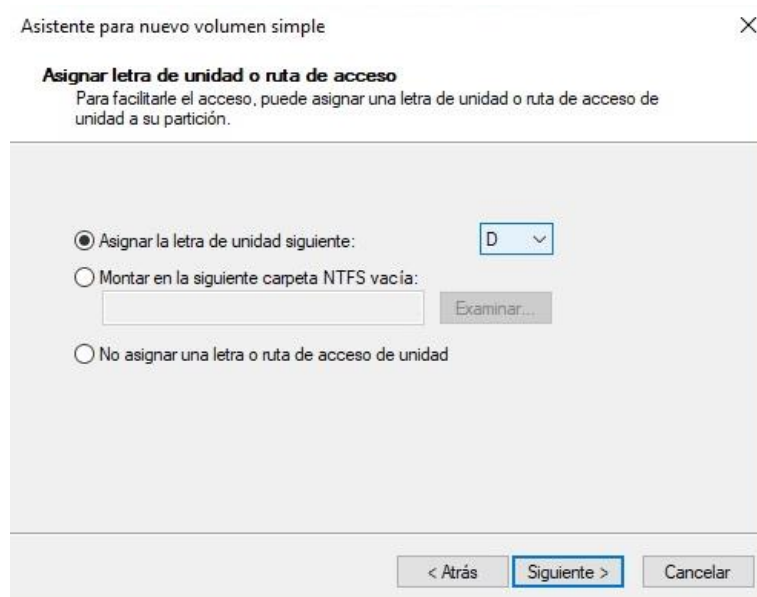
Se abre ahora un asistente para la creación del volumen. Se hace clic en “Siguiete”:



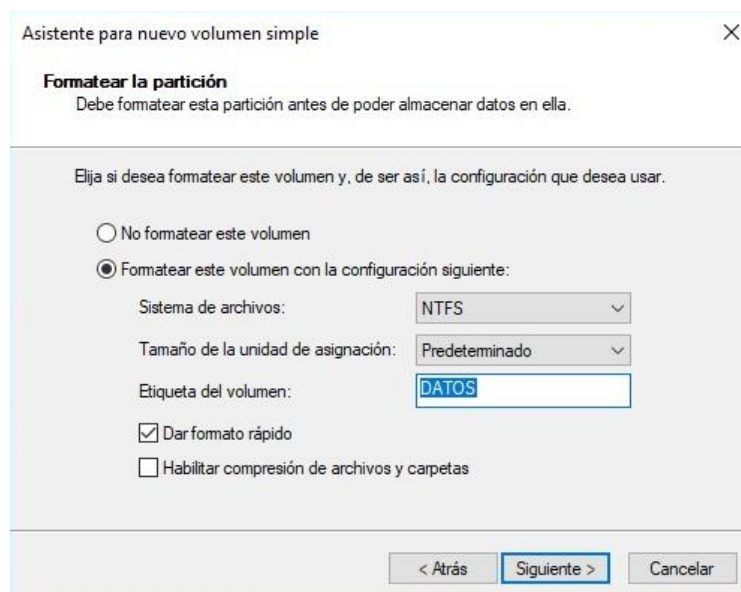
Para continuar, especificamos el tamaño que se dedica al nuevo volumen, que será el valor que aparezca por defecto, correspondiente al total de la partición. Una vez establecido se hace clic en “Siguiete”:



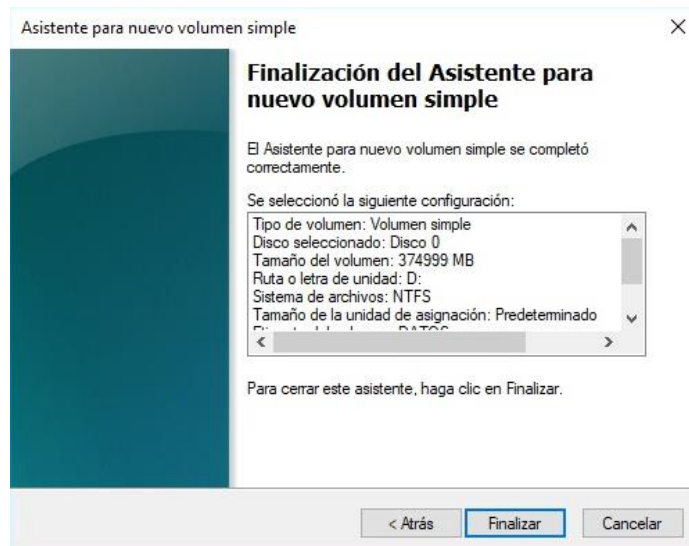
En la nueva pantalla, asignamos la letra que tendrá la unidad. Se selecciona “D” y se hace clic en “Siguiete”:



El siguiente paso es especificar el nombre “DATOS” para el nuevo volumen. Una vez completado ese campo se hace clic en “Siguiente”:



Para finalizar, aparece el resumen de las características que tendrá el volumen. Se pulsa “Finalizar”:



Una vez finalizados los cambios se reinicia el ordenador para que estos se apliquen correctamente.

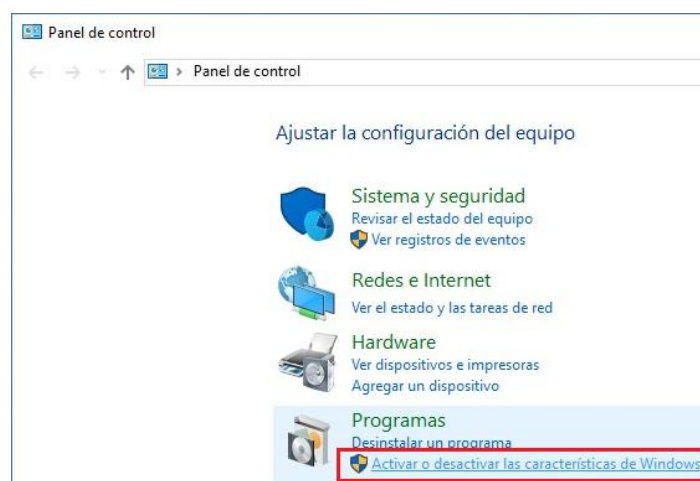
### 2.3.1.2. Activar características en Windows

Se distingue el caso de Windows Server 2016 y Windows 10

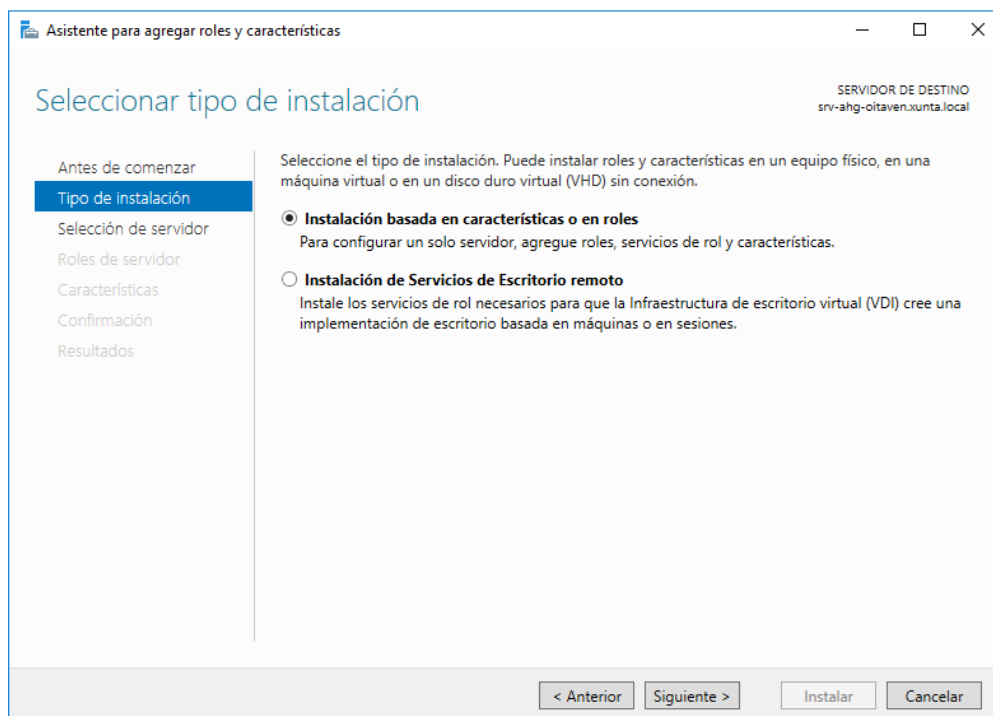
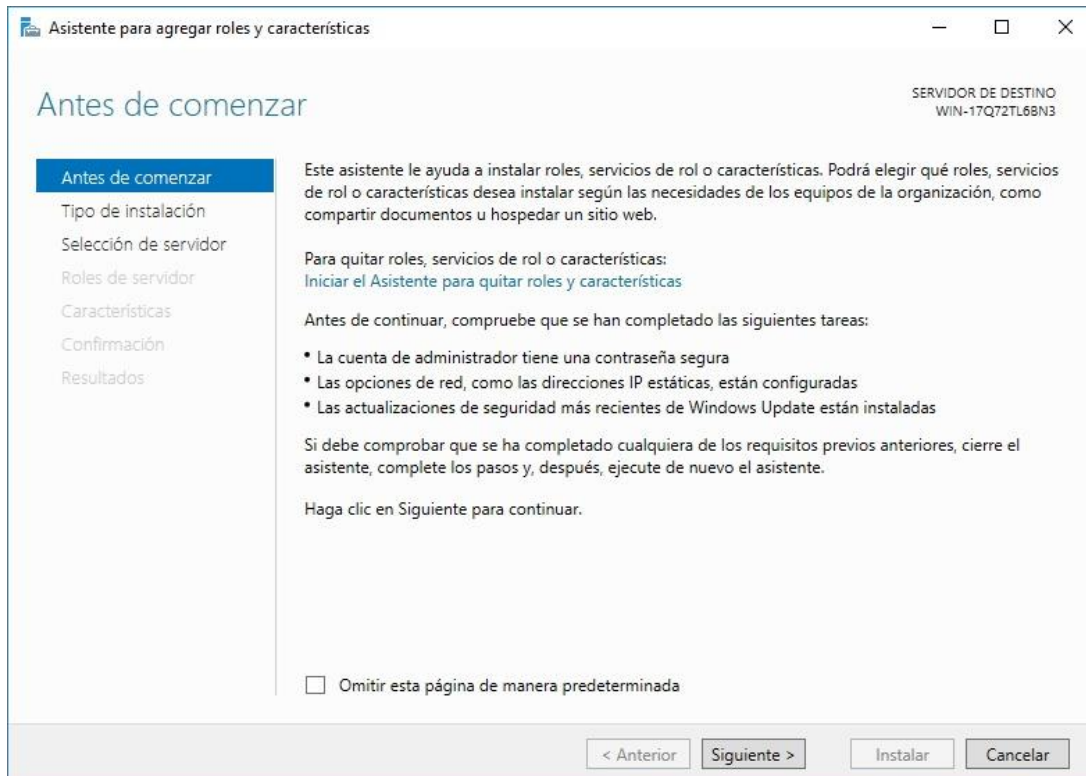
#### - Activar características en Windows Server 2016 (SERVIDORES)

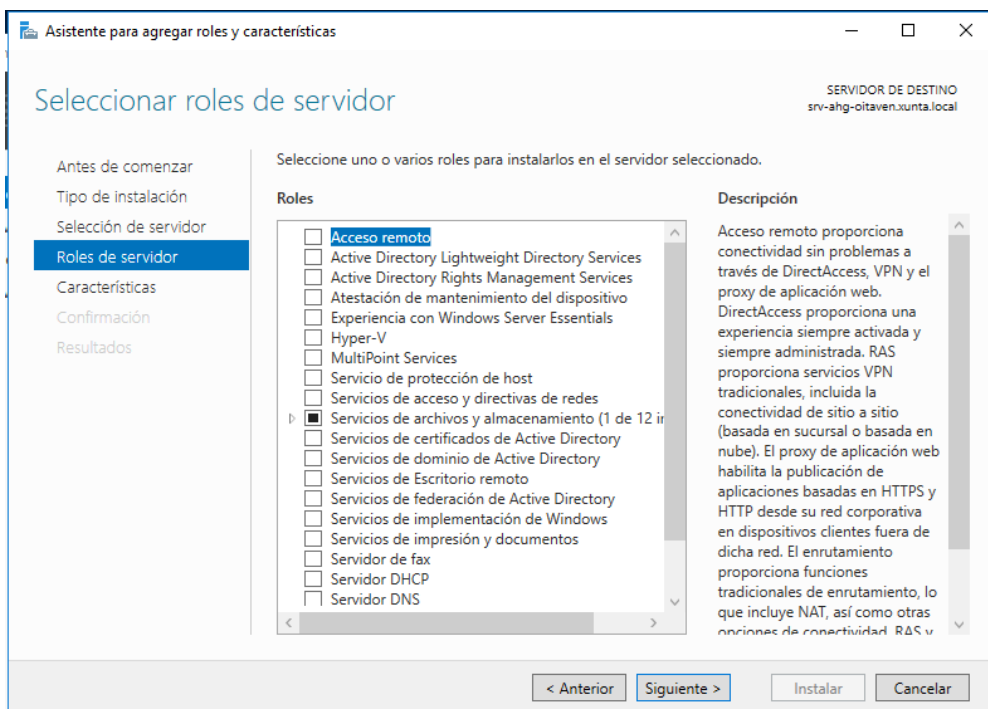
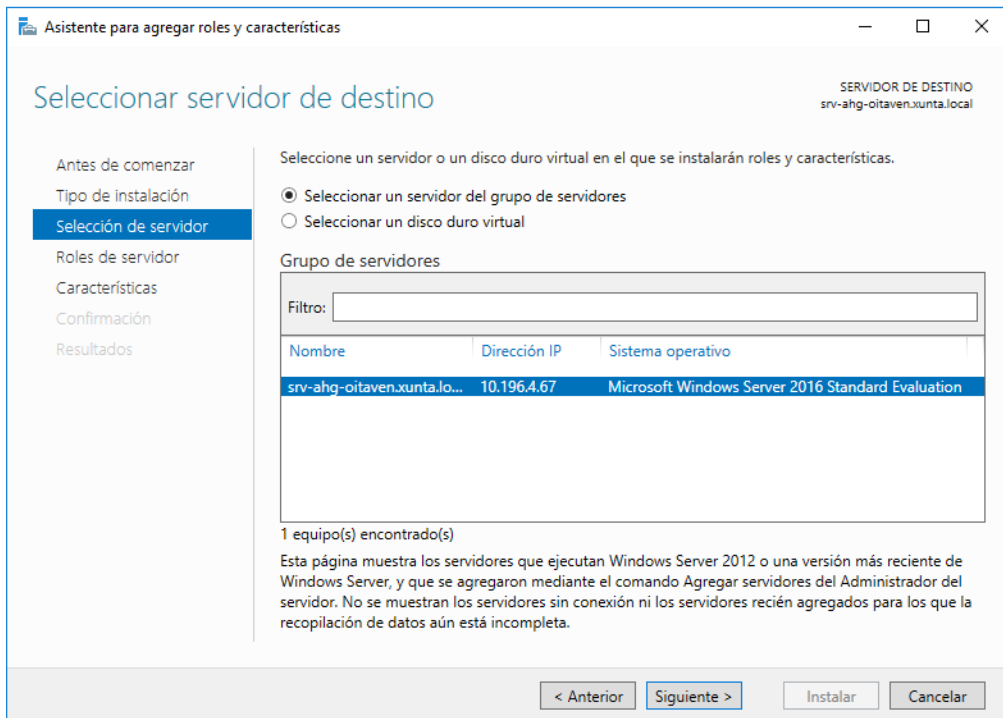
Una vez creada la partición, se procede a la habilitación de servicios necesarios para la instalación del software. Para ello se abre el panel de control, al que se puede acceder buscando en el menú de inicio. Se va a explicar el caso para un equipo con sistema operativo Windows Server y en los Anexos se incluye la explicación para el caso de que el sistema operativo sea Windows 10.

Una vez dentro del panel de control se hace clic en “Activar o desactivar las características de Windows”

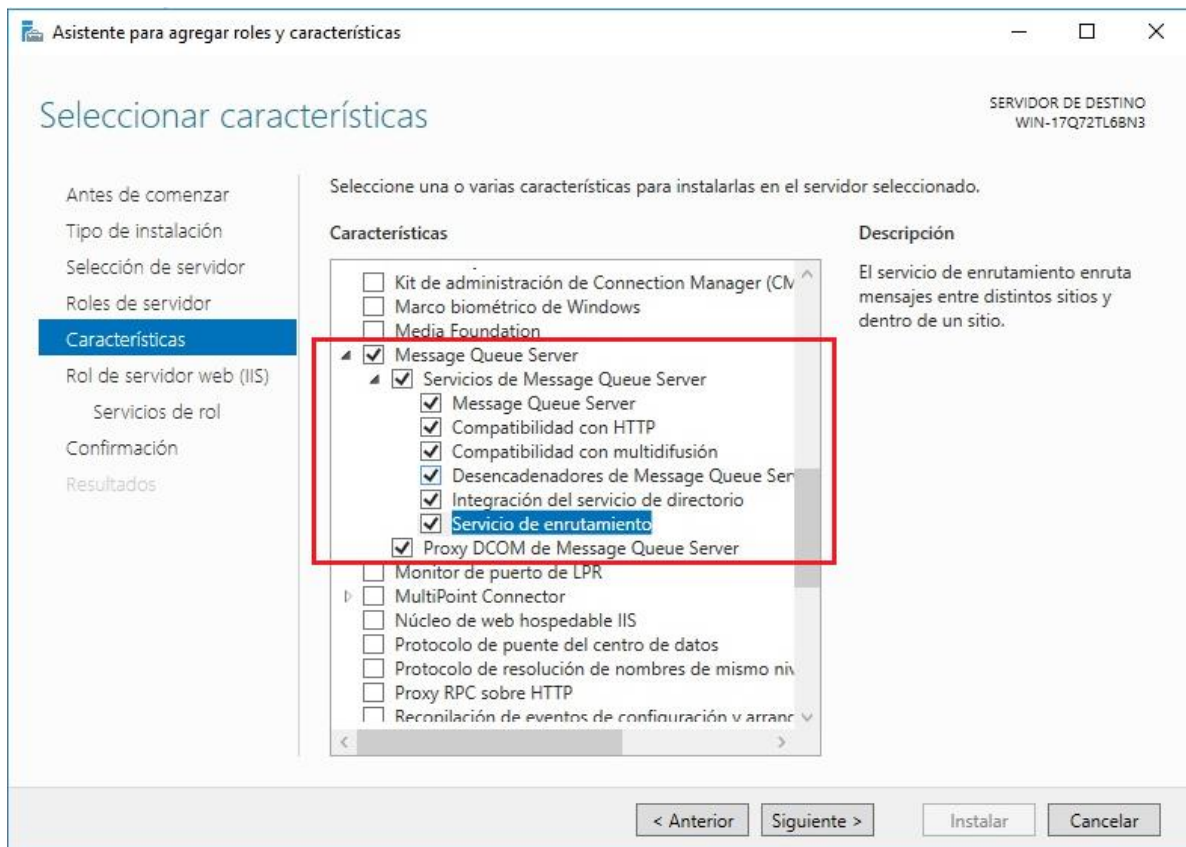


Aparece una nueva ventana como la siguiente:

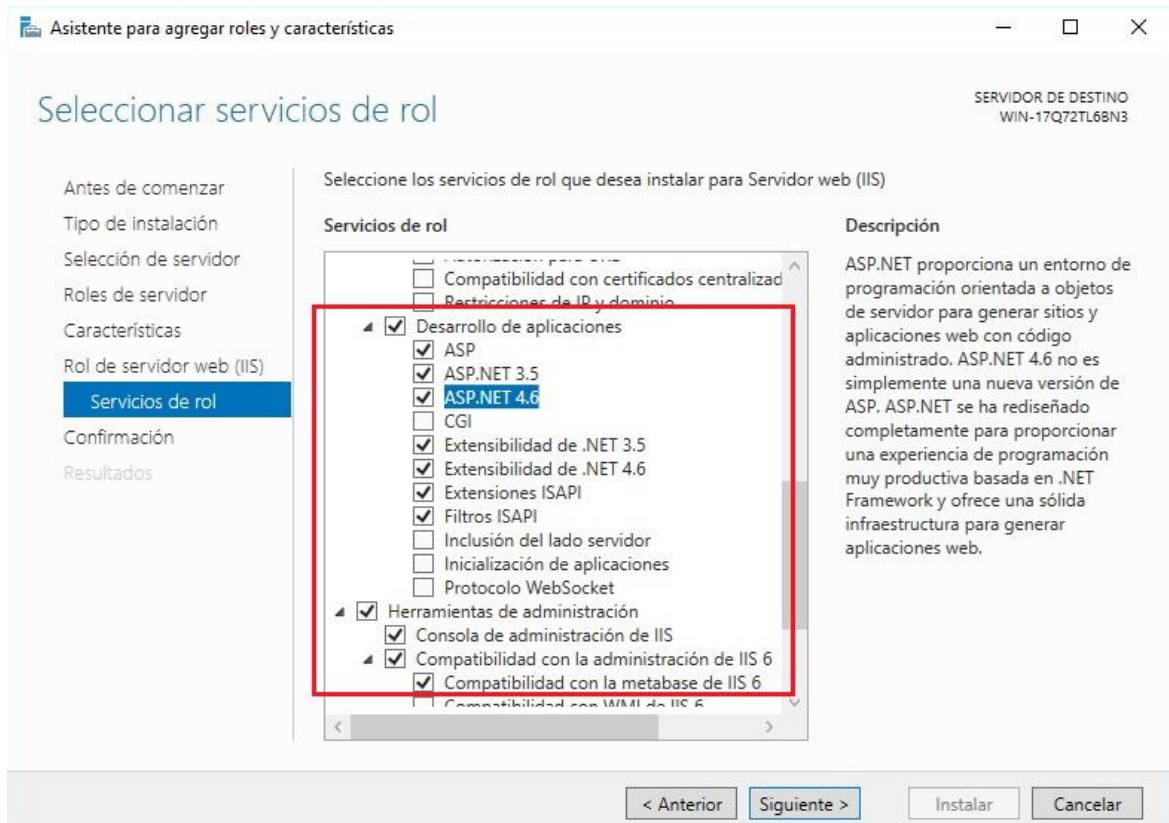




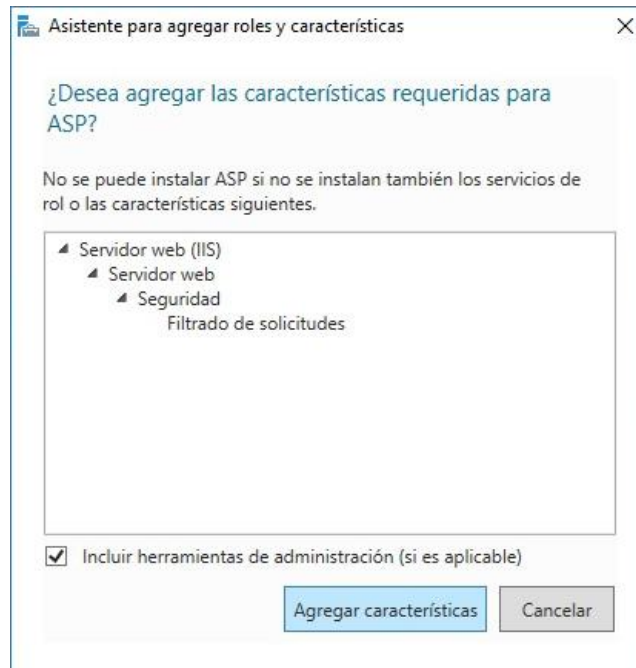
Una vez que llegamos a “Características” aparece el menú para habilitar las características de Windows. Para la instalación es necesario habilitar las que aparecen en la siguiente imagen:



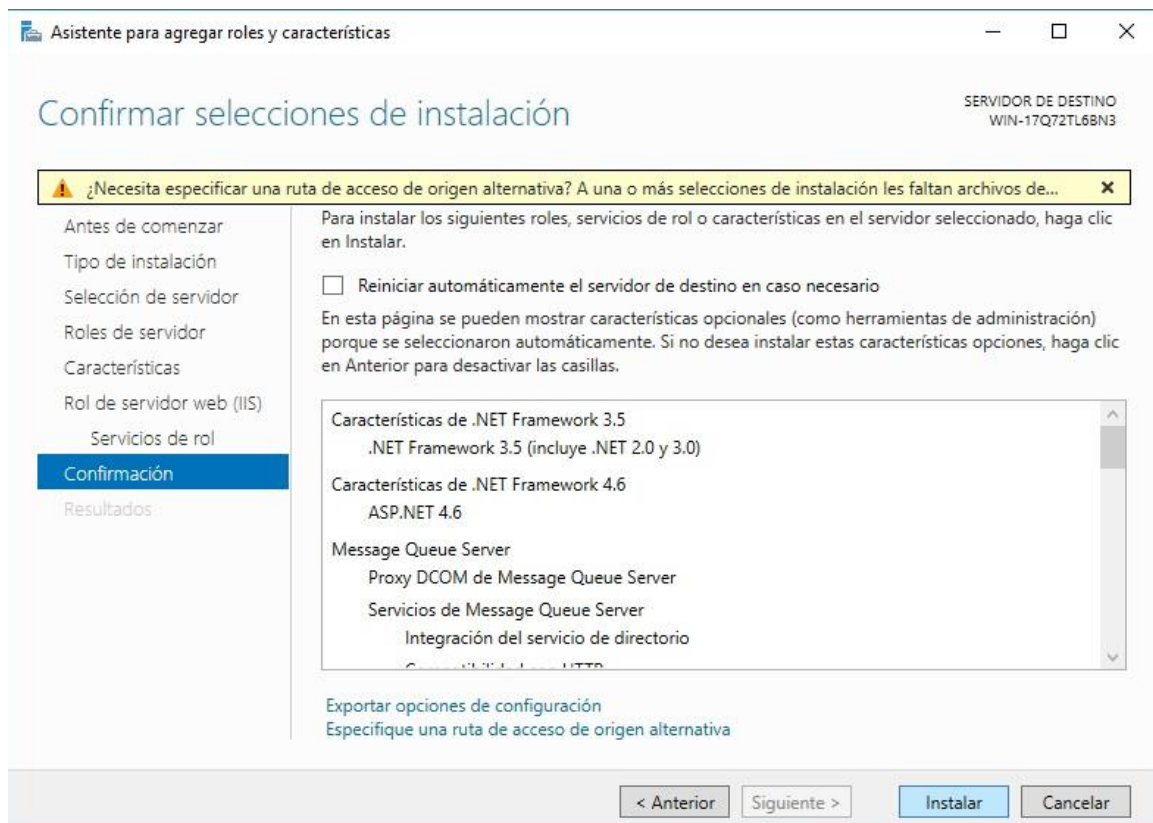
Una vez marcadas las opciones, es necesario pulsar en la pestaña del menú de la izquierda “Servicios de rol”. Al igual que en el paso anterior es necesario asegurarse de que se encuentran marcadas las opciones que aparecen en la siguiente imagen:



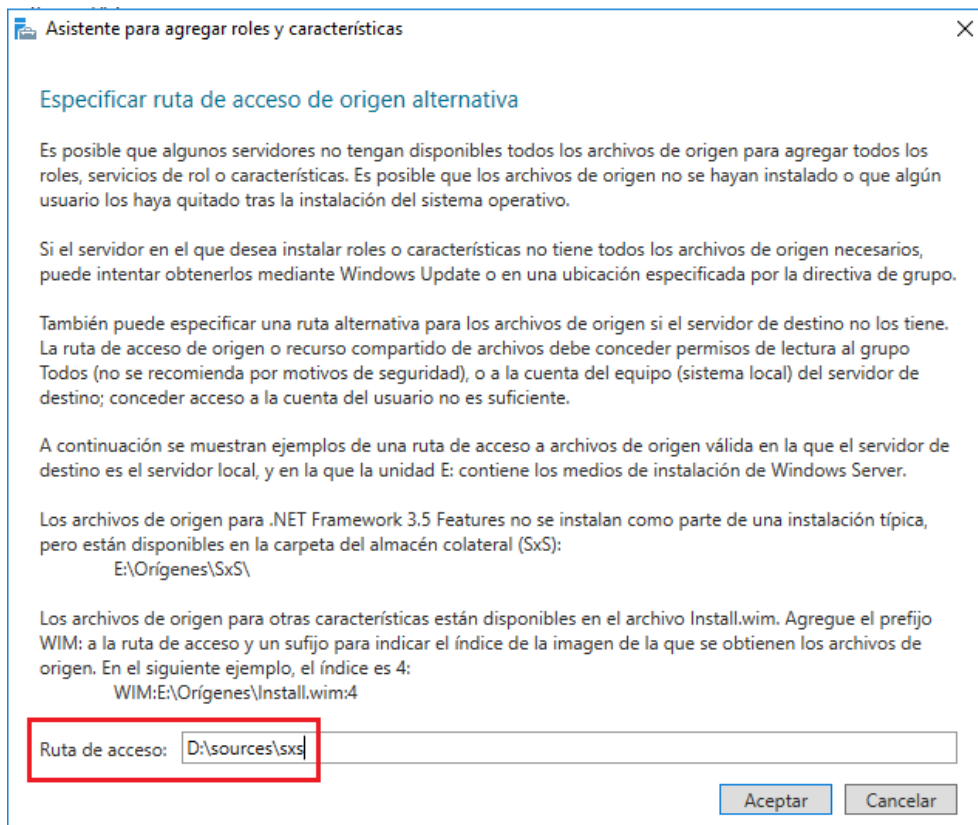
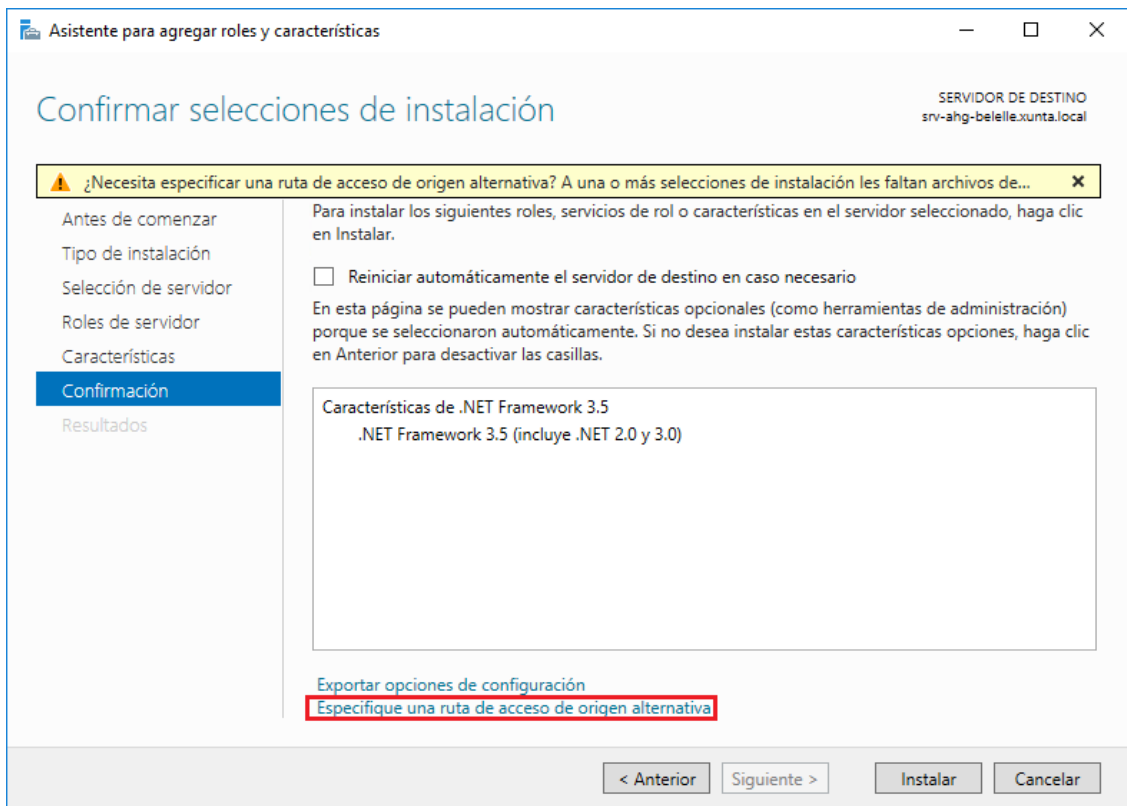
Cada vez que se marca una casilla nueva puede aparecer una ventana similar a la siguiente, en la que se debe de pulsar el botón “Agregar características”:



Una vez marcados todos los campos especificados en las imágenes anteriores, se pulsa en la pestaña del menú izquierdo “Confirmación”. En el menú del medio se puede ver el resumen de las características nuevas a instalar y también aparecerá en botón “Instalar” en el que se hace clic para proceder a instalar las características seleccionadas.

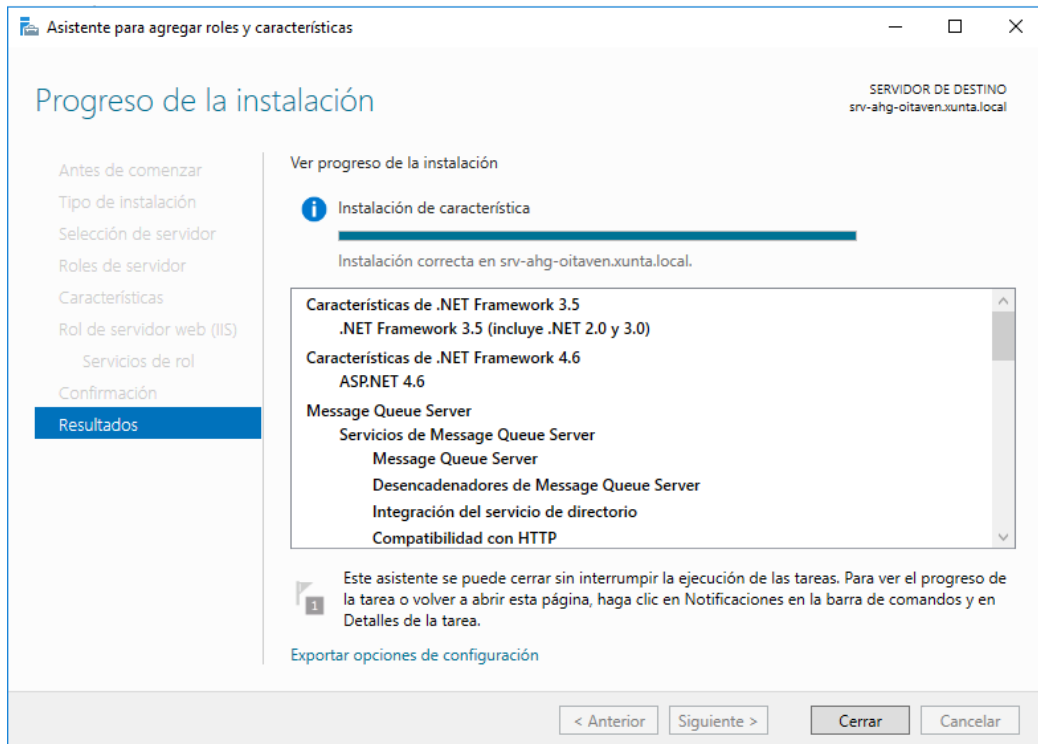


Si da un error es muy posible que sea porque no es capaz de instalar la característica de .NET Framework 3.5 ya que necesita los medios de instalación. Hay que tener el medio de instalación y especificar la ruta en la que se encuentra el .NET



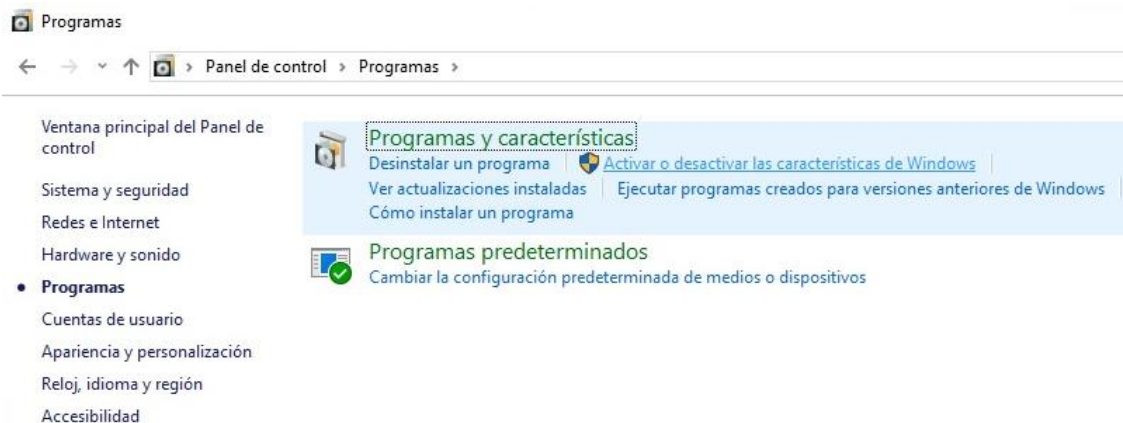


Una vez finalizado el proceso de instalación se pulsa en el botón “Cerrar” para abandonar el asistente y se procede a reiniciar el servidor para que se apliquen correctamente los cambios.



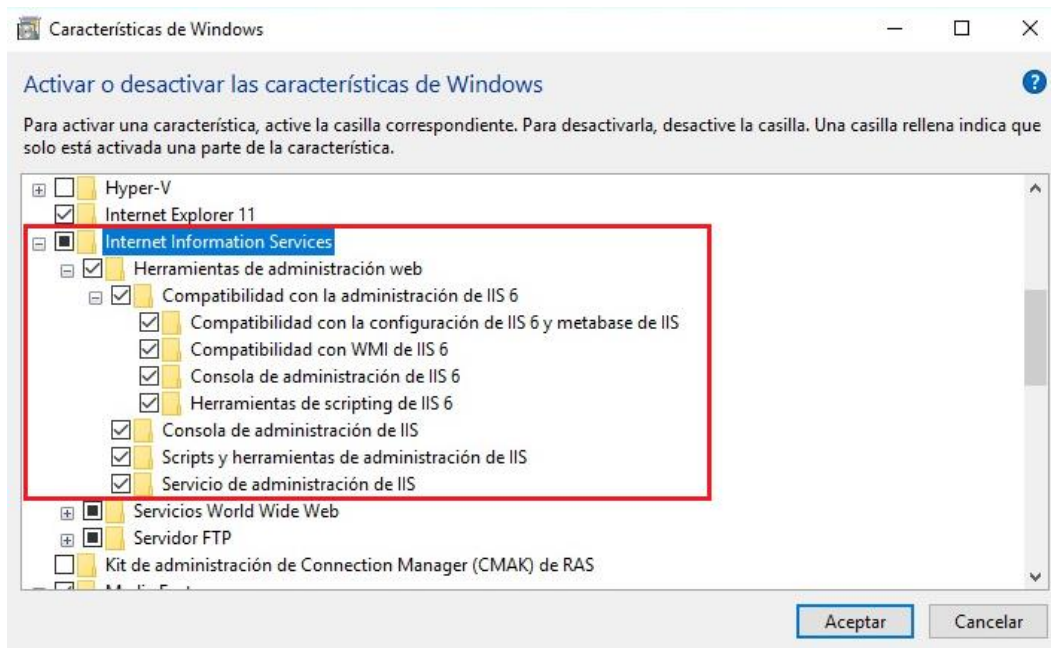
### - Activar características en Windows 10 (CLIENTES)

Una vez dentro del panel de control se hace clic en “Programas” y posteriormente en “Activar o desactivar las características de Windows”:

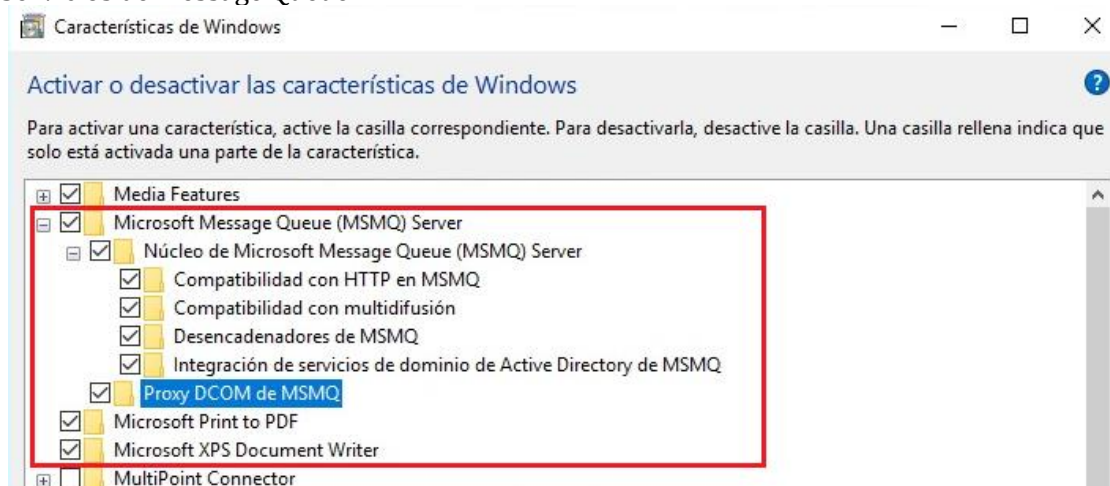


Se abre una nueva ventana como la siguiente, en la que se debe de marcar las mismas casillas que aparecen en las siguientes imágenes:

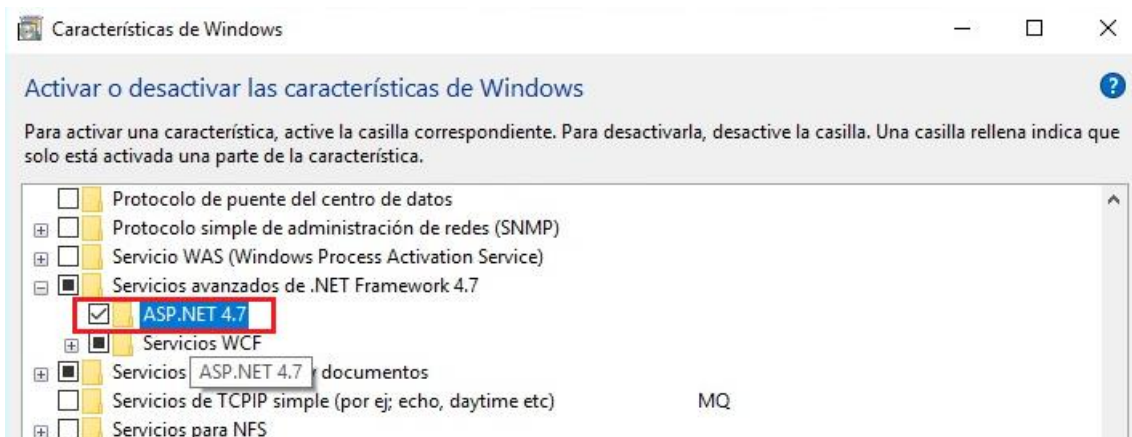
Habilitación de servicios IIS:



### Servicios de Message Queue:



### Servicios avanzados .NET:



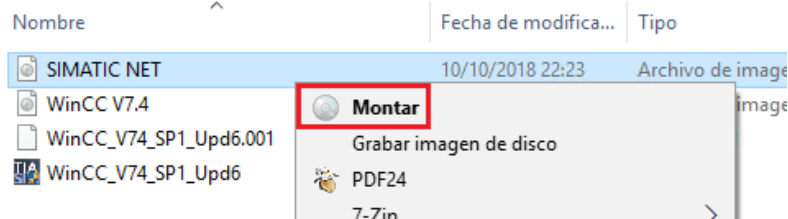
Una vez marcadas las opciones, es necesario pulsar en el botón "Aceptar" para aplicar los cambios (puede tardar varios minutos).

Ya finalizado el proceso de instalación, se pulsa en el botón “Cerrar” para abandonar el menú y se procede a reiniciar el equipo para que se apliquen correctamente los cambios.

### 2.3.1.3. Instalación de Simatic NET

Para la instalación del Simatic Net es necesario tener disponible el software en formato de imagen .ISO o en un disco si se dispone de un lector por USB. Para comenzar, si no se ejecuta de forma automática el contenido del CD, se abre el directorio y se hace doble clic sobre el archivo “setup” para que comience el asistente de instalación.

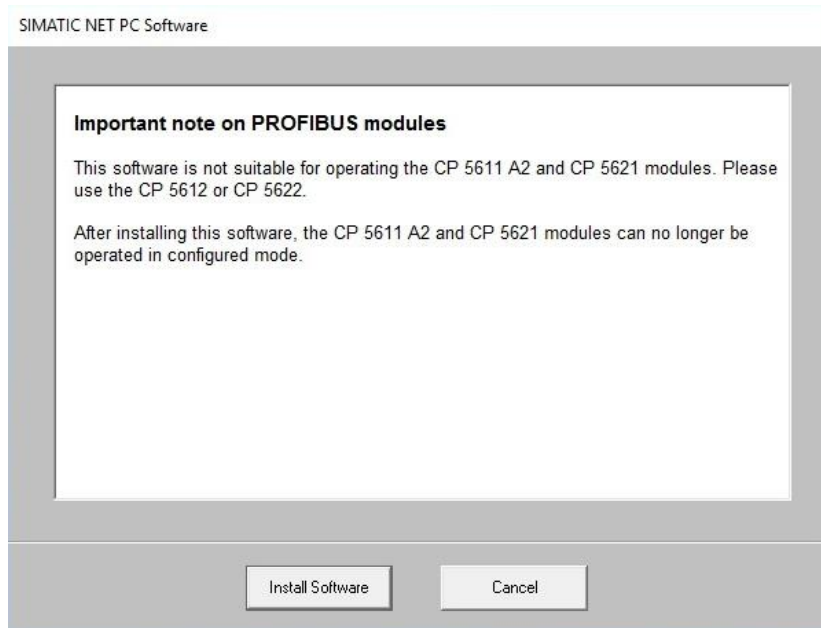
Nota: Windows 10 permite de forma nativa montar imágenes de disco en formato .iso por lo que no hace falta ningún tipo de software adicional para montar la imagen.



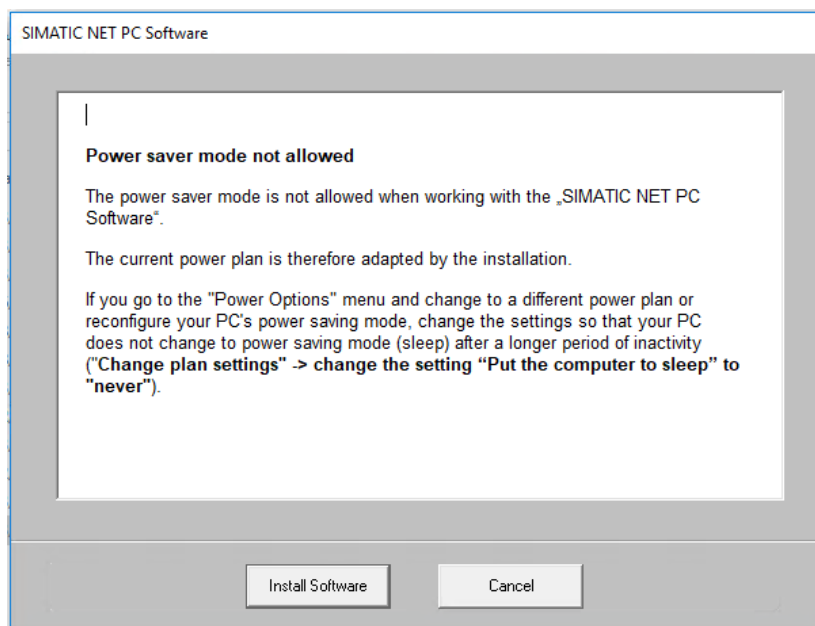
En el primer paso del asistente, se hace clic en el botón “Install Software” para avanzar de pantalla:



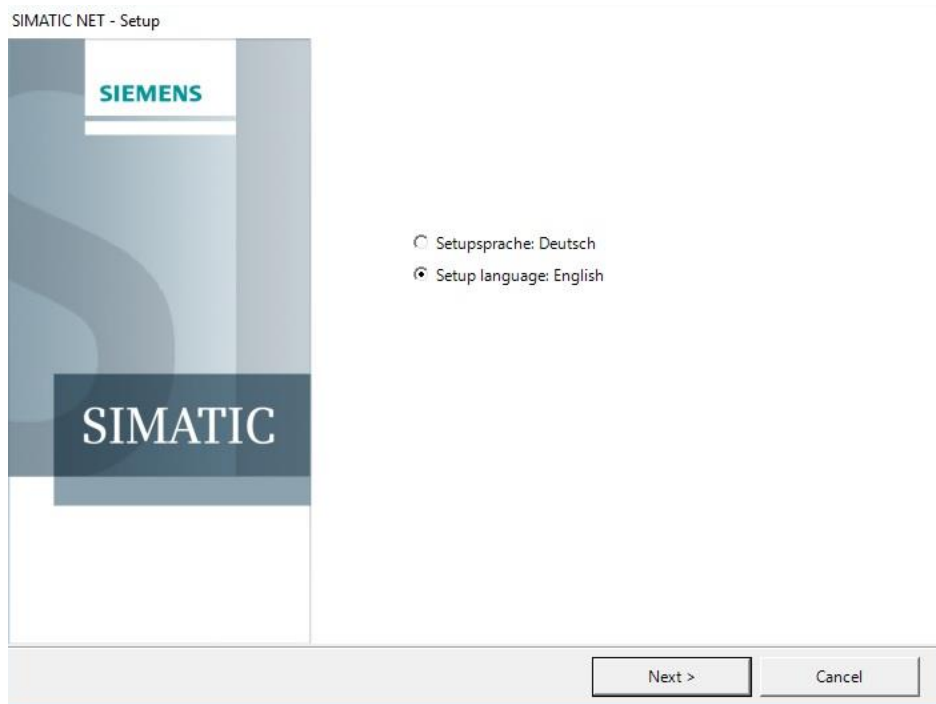
Aparecen a continuación dos avisos como el de la imagen siguiente, en el que para poder continuar, se debe de pulsar “Install Software” en ambos:



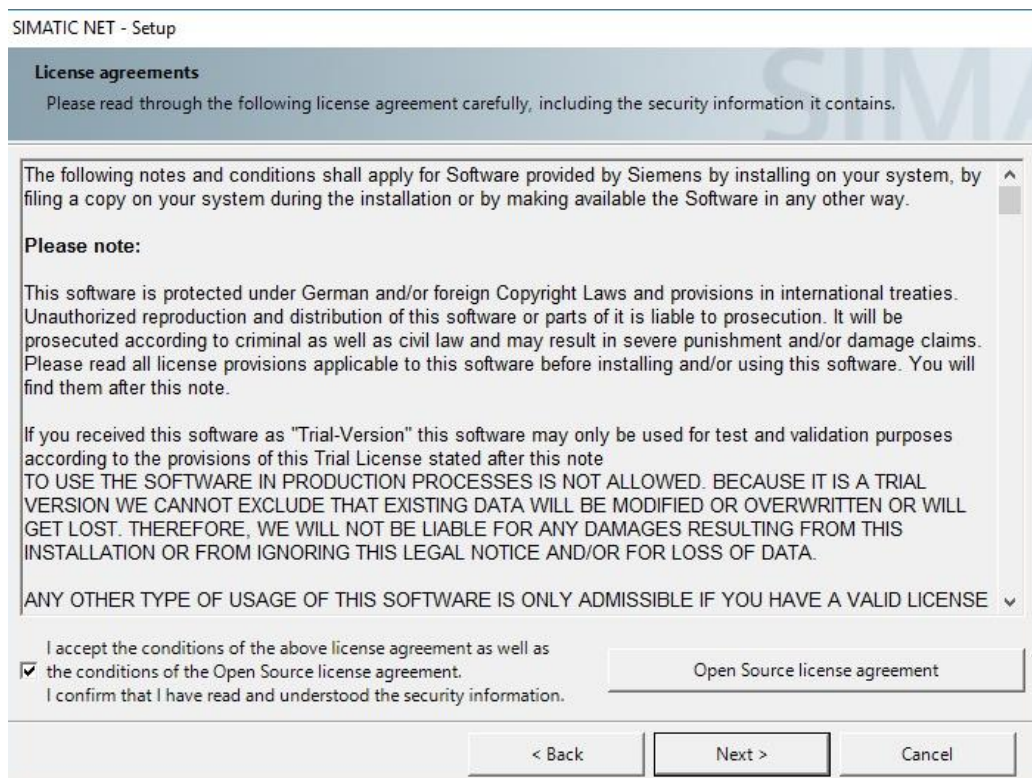
En el caso de que avise sobre el plan de gestión de energía, abortamos y lo configuramos para que el equipo nunca se suspenda, hiberne, apague discos duros... También se deshabilita en las interfaces de red que puedan pasar a modo de ahorro de energía.



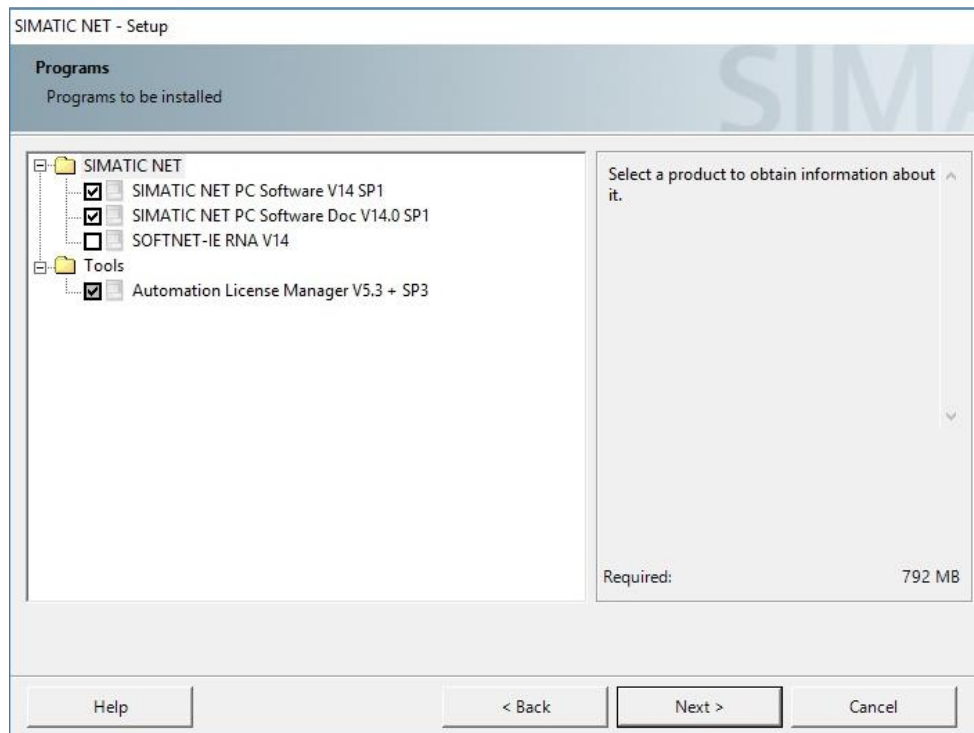
Una vez aceptados los avisos aparece la opción del idioma del asistente de instalación. Se marca la opción "English" y se hace clic en "Next":



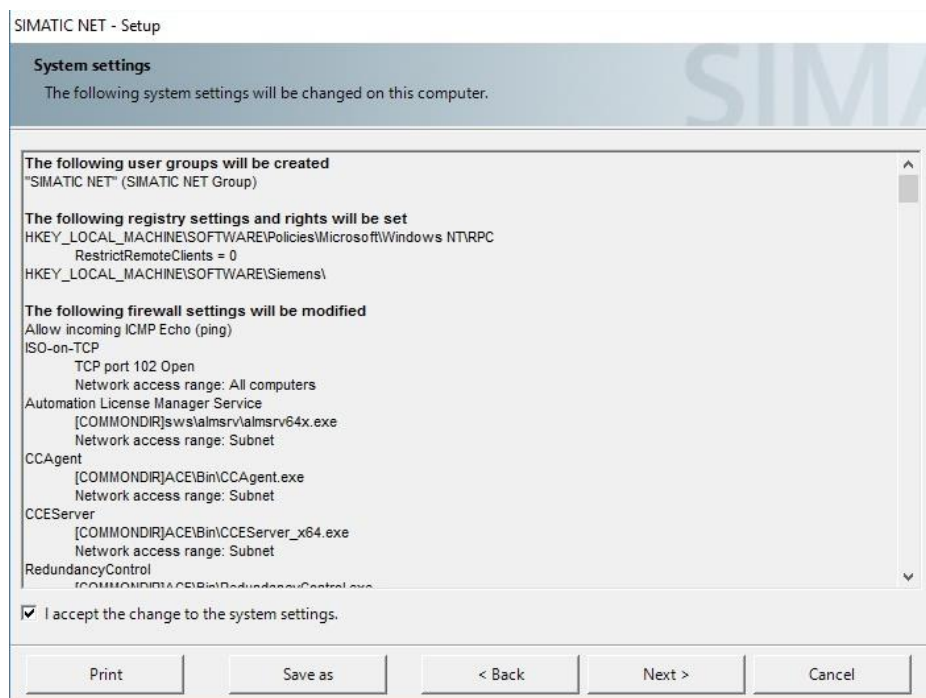
Para continuar, se aceptan las condiciones de licencia, marcando el check que aparece a la izquierda y pulsando de nuevo en “Next”:



La siguiente pantalla muestra el software que se va a instalar, por lo que se debe de verificar que se corresponde con el de la imagen siguiente y si es así hacemos clic en “Next”:



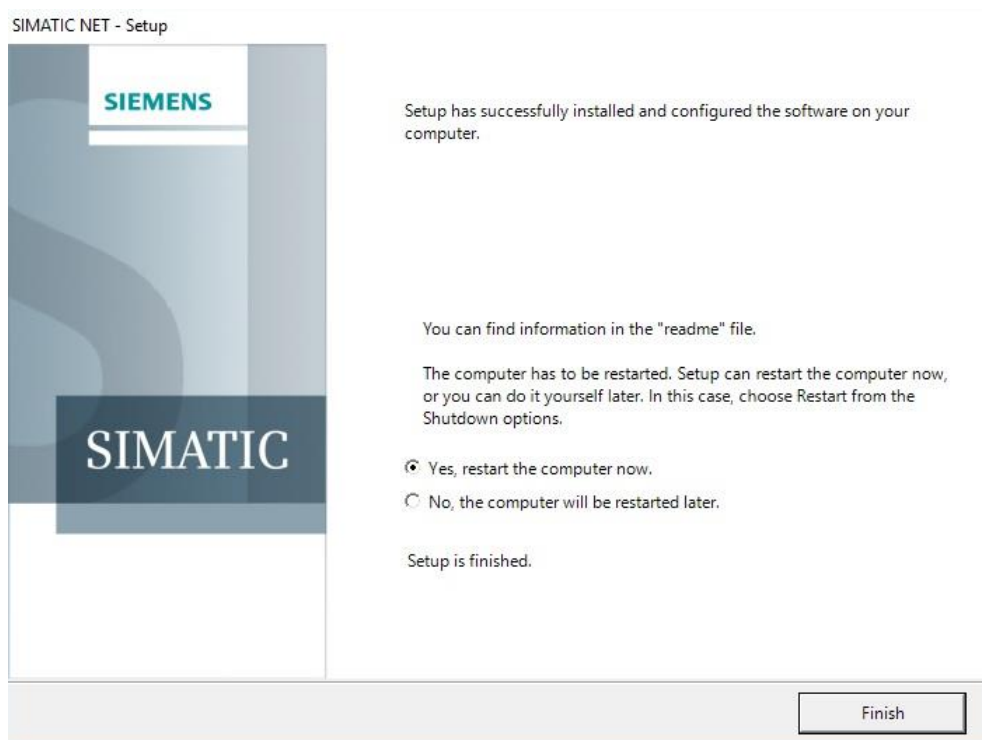
A continuación, aparece el resumen los cambios de configuración que se van a realizar sobre el equipo. Se marca el check aceptando los cambios y se pulsa “Next”. Una vez pulsado empezará la instalación, que se puede prolongar varios minutos:



Antes de completarse el proceso de instalación aparece una nueva ventana para licenciar el software instalado, tal y como la de la imagen siguiente. Se hace clic en “Next” para omitir ese paso:

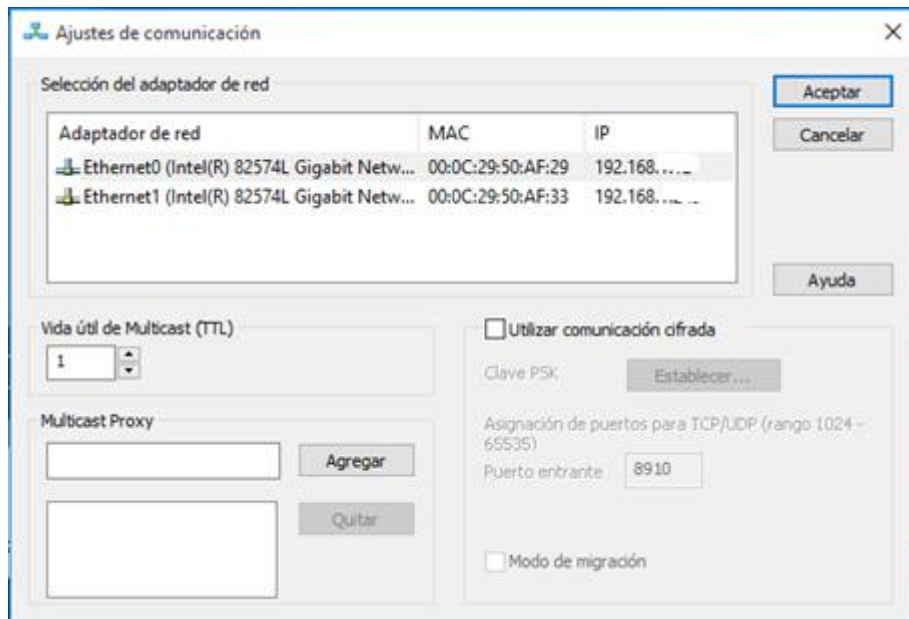


Una vez finalizado el proceso aparecerá la opción de reiniciar el equipo. Se marca ese punto y se hace clic en “Finish”.



Una vez reiniciado el equipo tendría correctamente instalado el software SimaticNet.

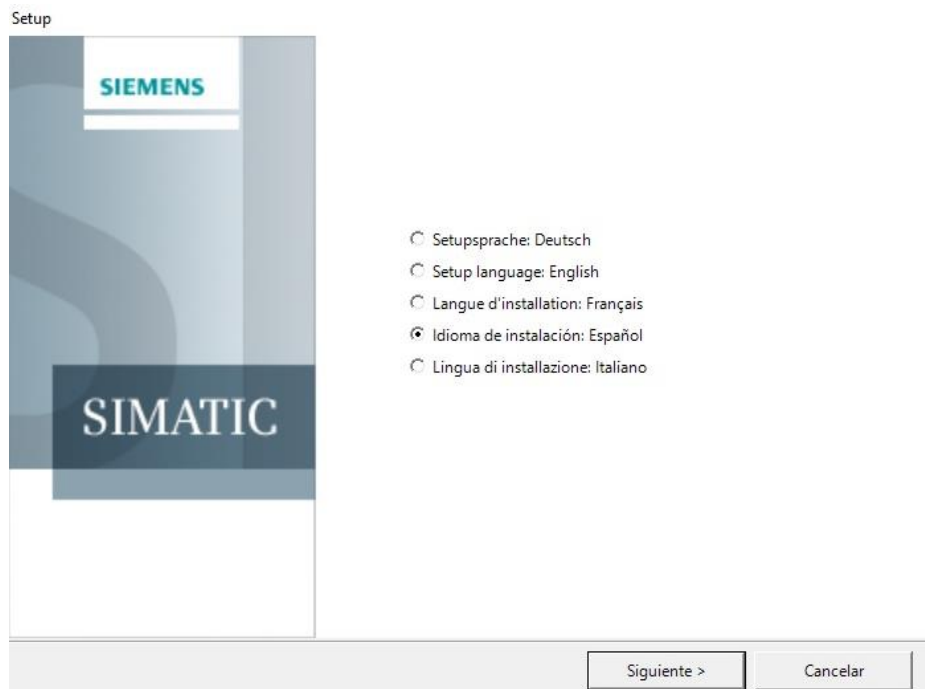
Al reiniciar puede que nos solicite la interfaz de red de la red de autómatas. La seleccionamos y Aceptamos. Nota: En el pantallazo la configuración TCP/IP del adaptador todavía no estaba definida.



#### 2.3.1.4. Instalación de WinCC 7.4 Service Pack 1

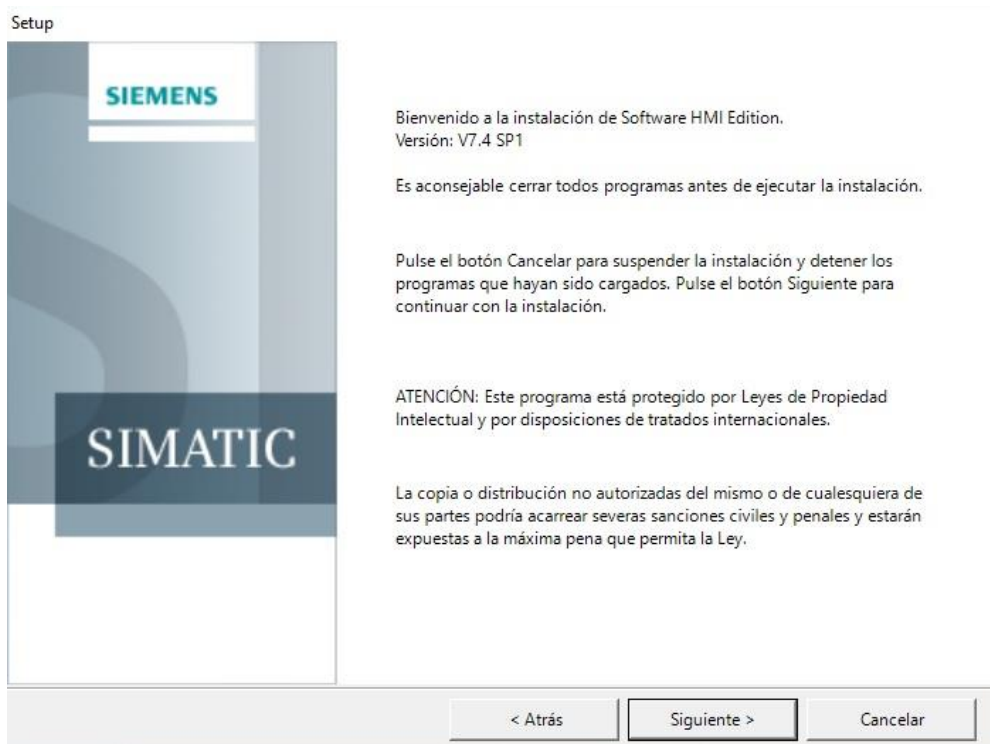
Al igual que para la instalación del SimaticNet es necesario tener disponible el software en formato de imagen .ISO o en un disco si se dispone de un lector externo. Para comenzar, si no se ejecuta de forma automática el contenido del CD, se abre el directorio y se hace doble clic sobre el archivo “setup” para que comience el asistente de instalación.

En el primer paso se selecciona el idioma de la instalación y se hace clic en “Siguiente” para continuar con el asistente:

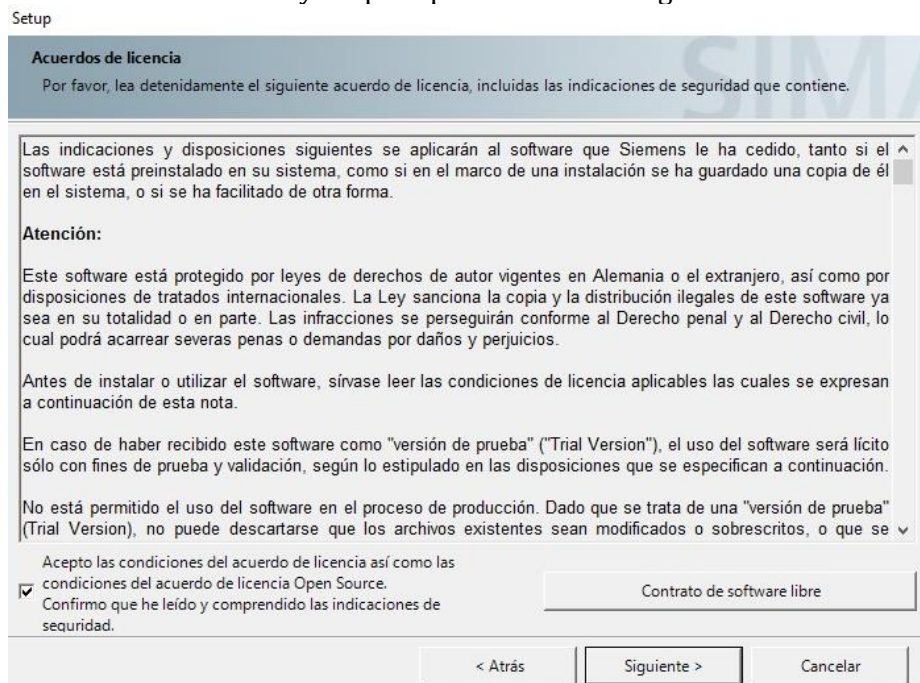


Antes de continuar es recomendable cerrar otros programas que se estén ejecutando (en caso de que los haya) y puedan interferir en el proceso de instalación. Una vez cerrados se hace clic en “Siguiente”:

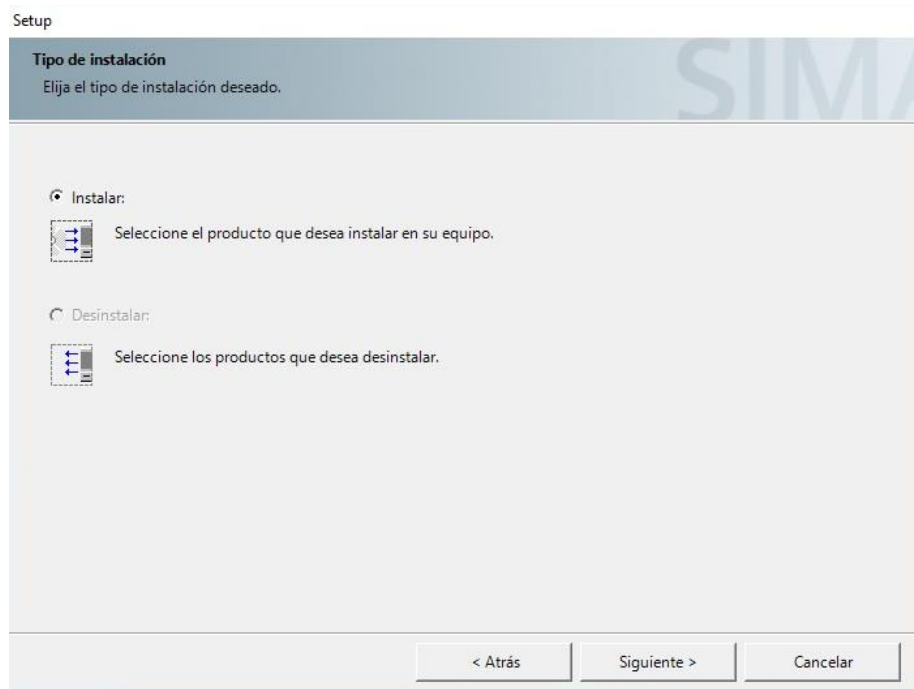




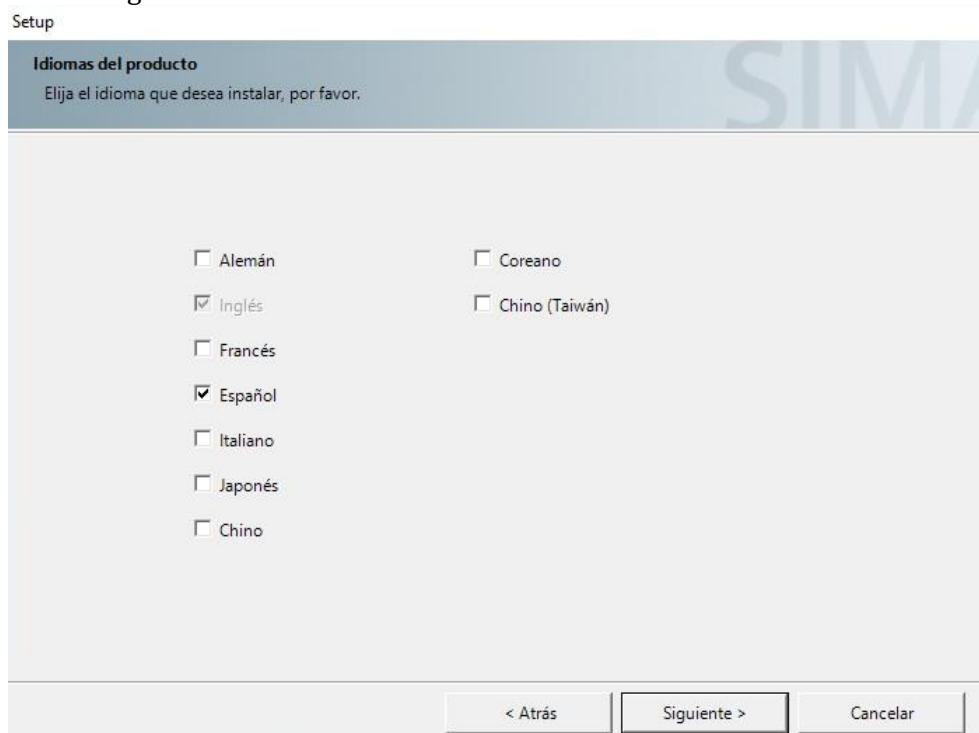
La siguiente ventana que aparece contiene los acuerdos de licencia. Es necesario aceptar las condiciones marcando el check y después pulsar el botón “Siguiente”:



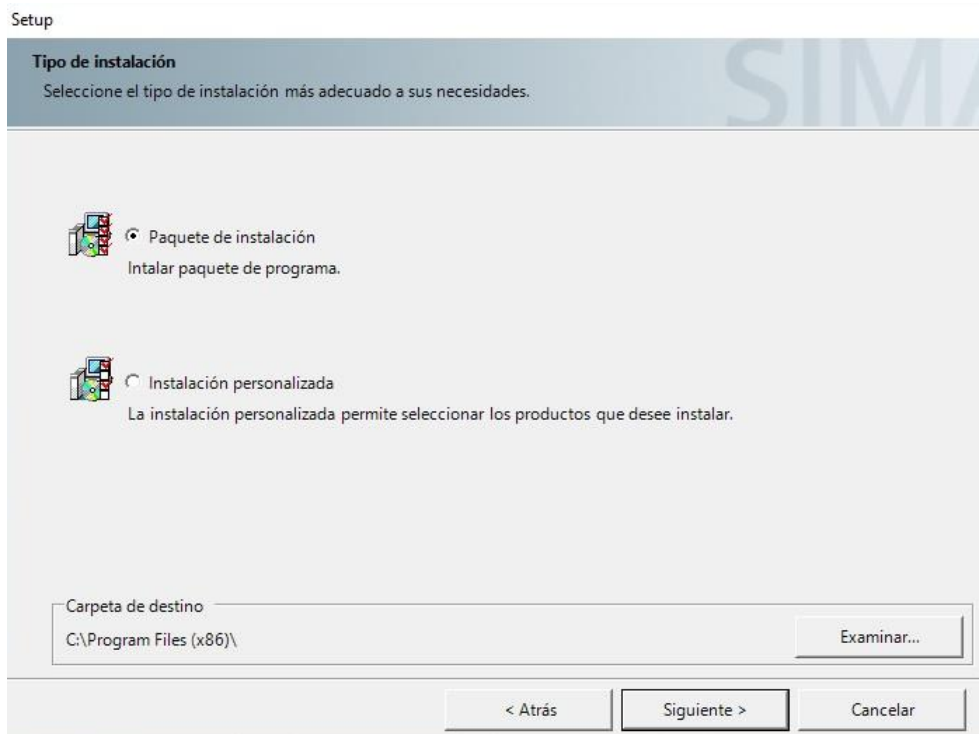
A continuación, en el tipo de instalación, se hace clic en “Siguiente”:



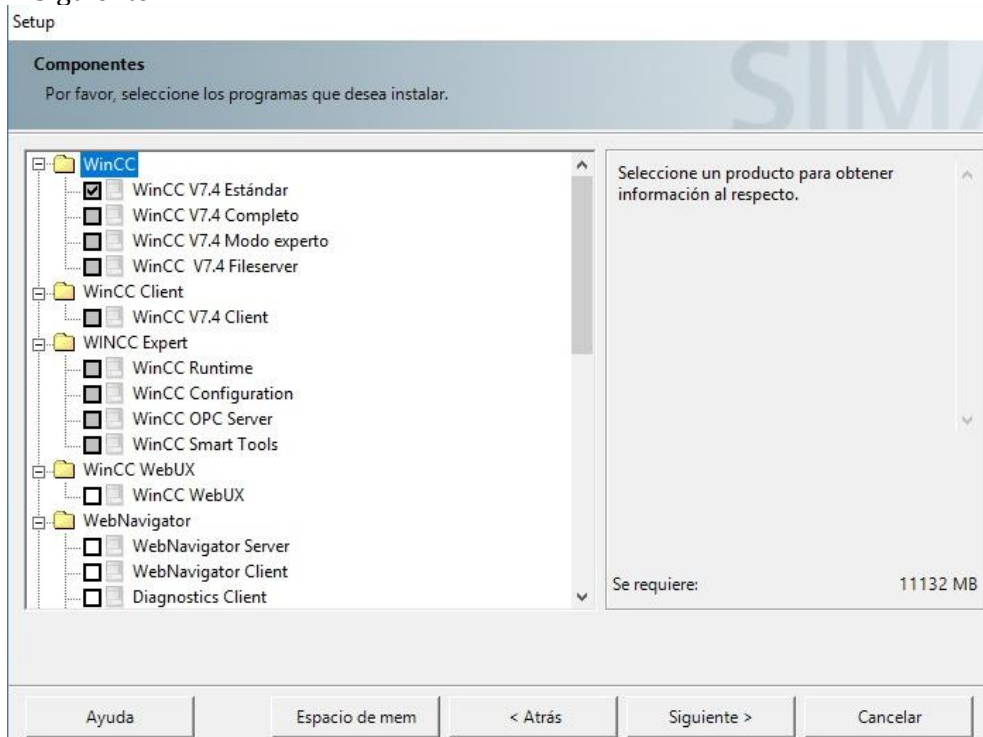
El siguiente paso es seleccionar el idioma del producto que se va a instalar. Por defecto aparece marcado “Español” e “Ingles”, por lo que se puede continuar con la instalación pulsando en “Siguiete”:



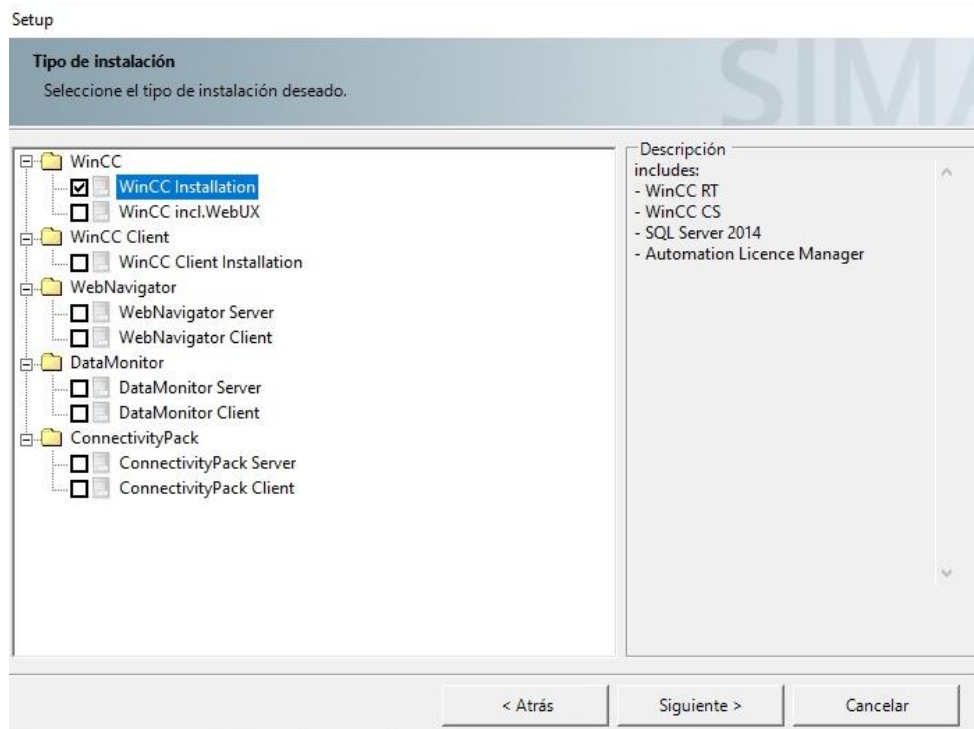
Se marca el tipo de instalación “Paquete de instalación” y se hace clic en “Siguiete”.



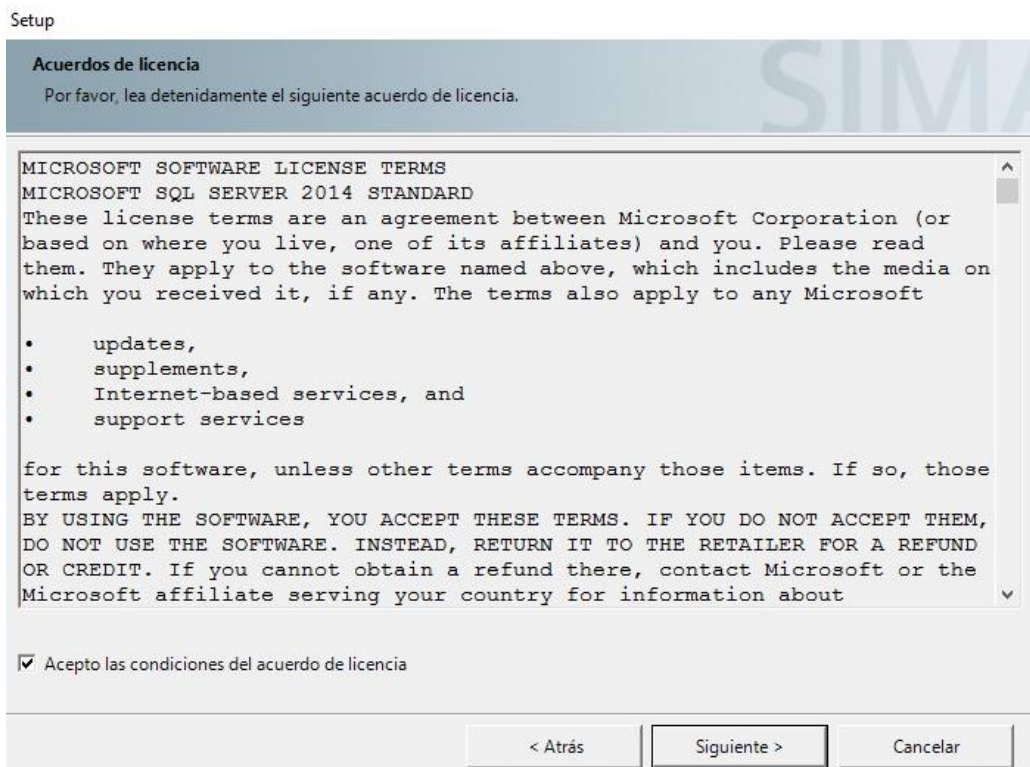
A continuación, se selecciona la casilla WinCC Installation, al igual que en la imagen siguiente, ya que incorpora todo lo necesario para este equipo. Una vez seleccionado se hace clic en "Siguiente":



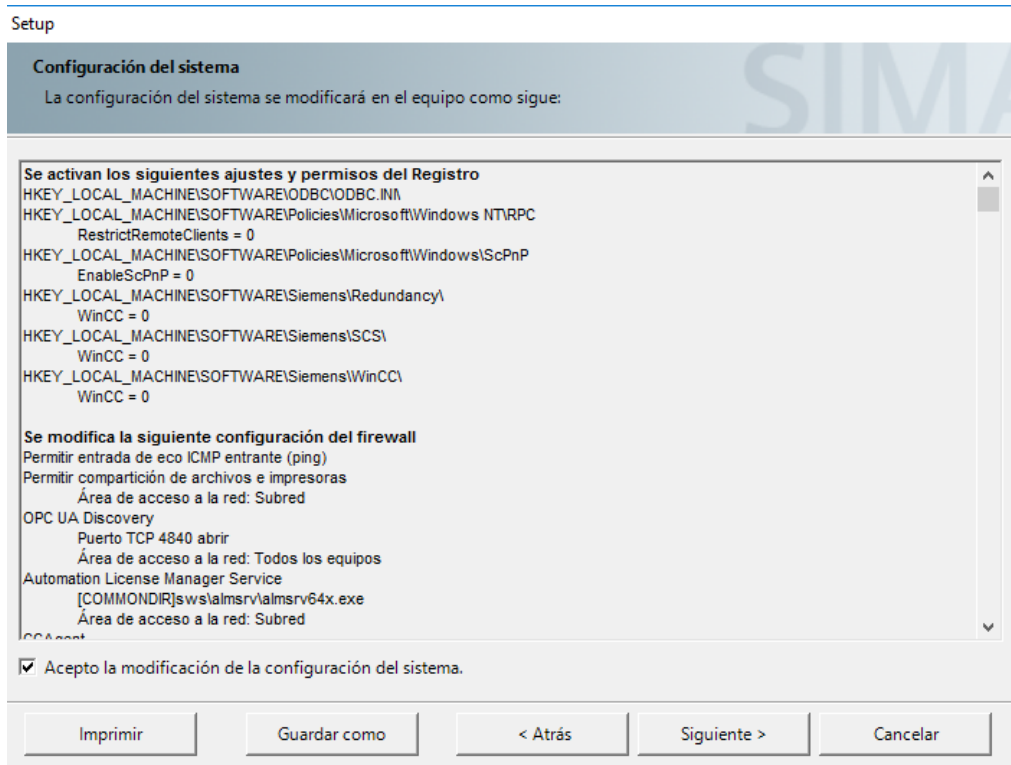
En la siguiente ventana que aparece, se verifica que corresponde la casilla marcada con la de la siguiente imagen y se hace clic en "Siguiente":



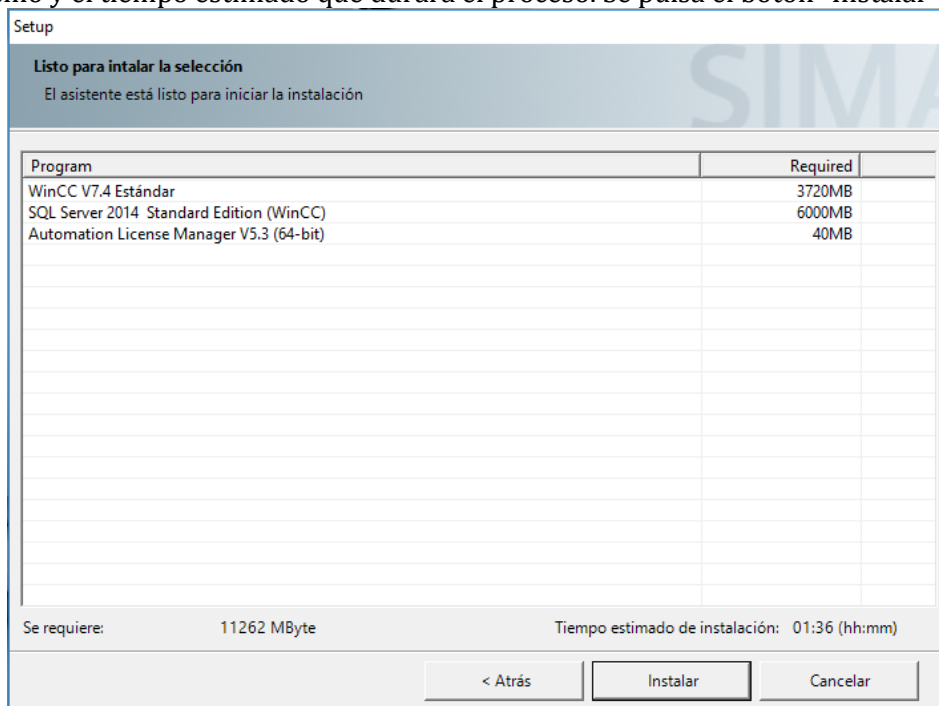
Para continuar, es necesario aceptar los nuevos términos y condiciones del software seleccionado anteriormente, por lo que se marca el check y se pulsa “Siguiete” para continuar con la instalación:



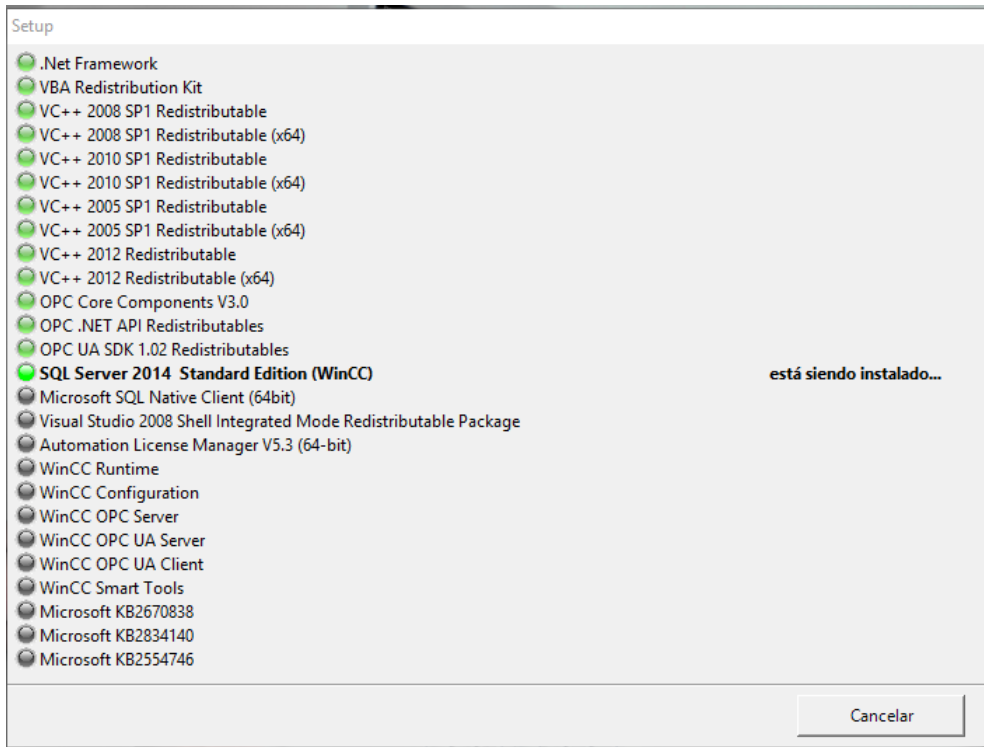
En la siguiente ventana aparece un resumen de la configuración del sistema que se va a cambiar, que también se debe de aceptar con el check y pulsar el botón “Siguiete”:



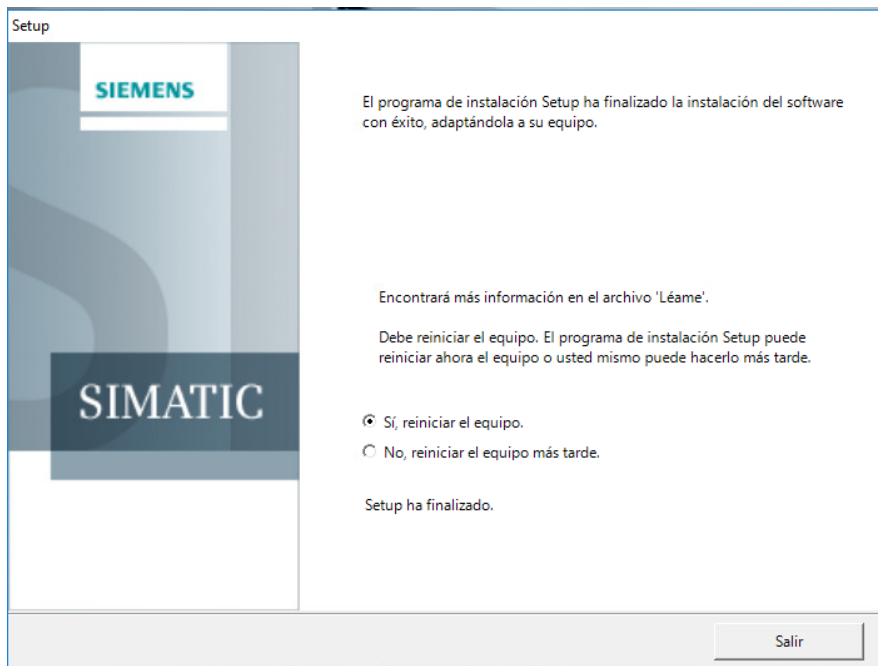
A continuación, aparece un resumen del software que se va a instalar junto con el tamaño del mismo y el tiempo estimado que durará el proceso. Se pulsa el botón "Instalar".



Mientras está instalando sale la pantalla siguiente indicando el progreso.



Para finalizar el proceso aparecerá la opción de reiniciar el equipo. Se marca ese punto y se hace clic en “Finish”.

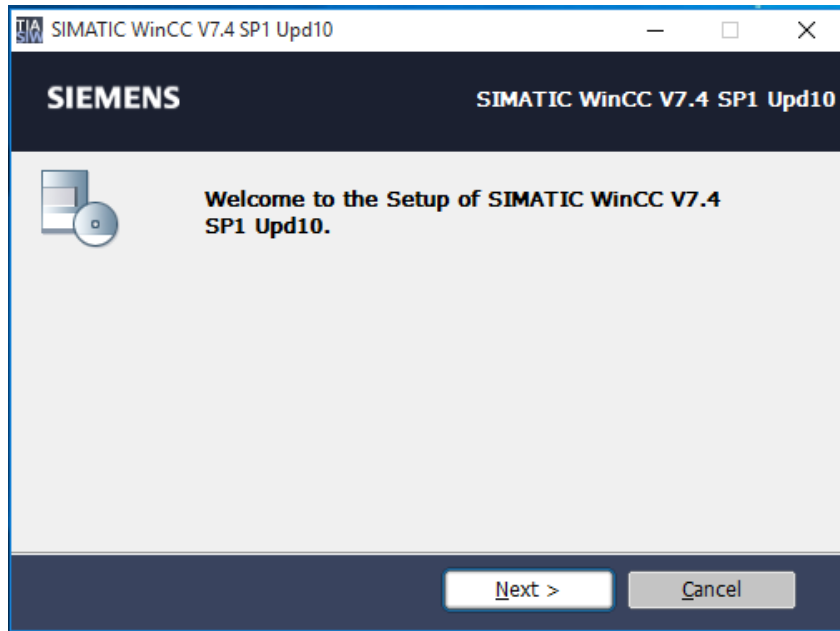


### 2.3.1.5. Instalación Update 10 para WinCC 7.4 SP1

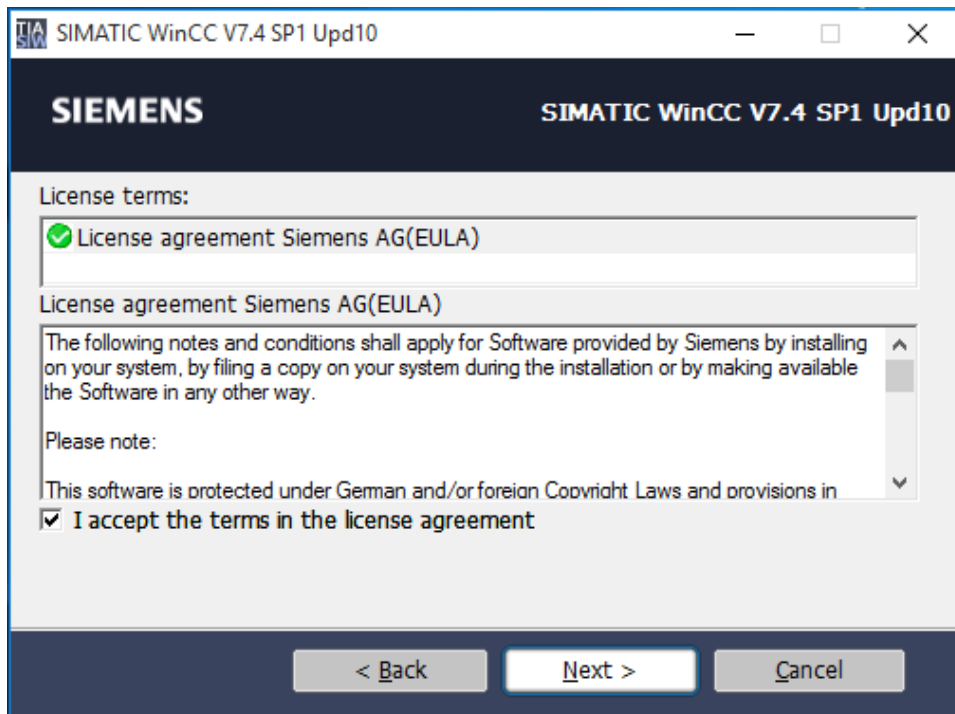
Para completar el proceso de instalación del equipo, es necesario instalar el último update disponible para WinCC 7.4 SP1, que en el momento del desarrollo de este proyecto era el Update 10. El update comprende dos archivos, un archivo comprimido (con extensión 001)

y un archivo ejecutable con extensión “.exe”. Es necesario depositar ambos en un mismo directorio.

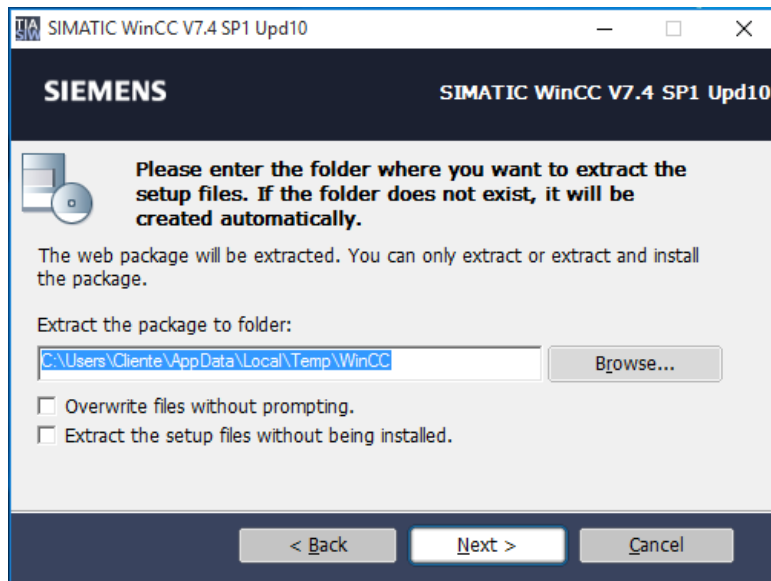
Para comenzar con la instalación se hace doble clic sobre el archivo de extensión “.exe”. Automáticamente se abre un asistente de instalación, en el que se debe de pulsar “Next” para continuar con el proceso:



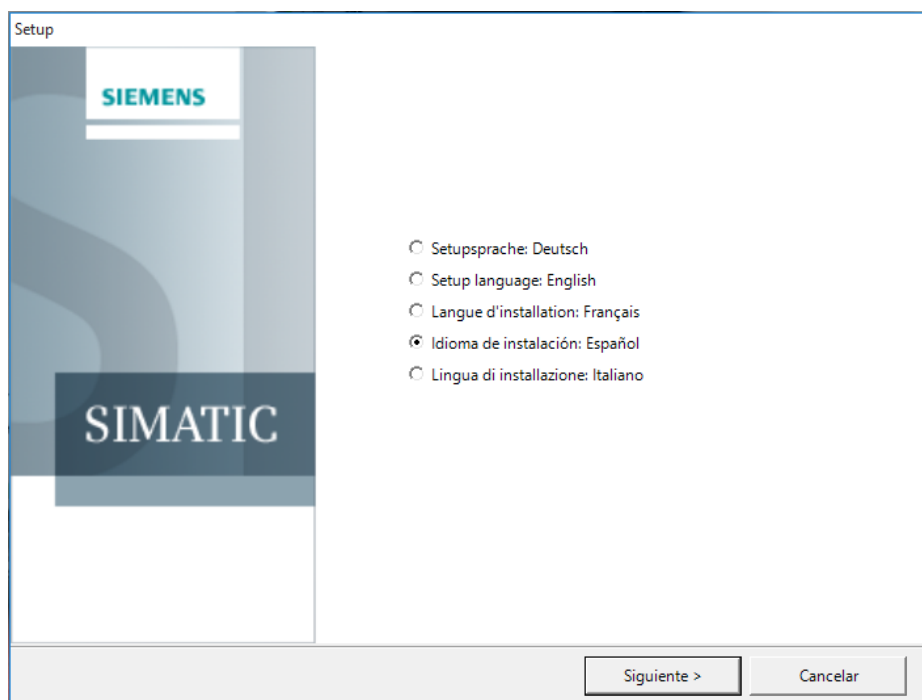
Se aceptan los términos de licencia marcando el check y se avanza pulsando en “Next”:



En la siguiente ventana hay que especificar el directorio donde se procede a descomprimir el contenido del Update. Una vez cubierta la ruta (se puede dejar la ruta que aparece por defecto) se hace clic en “Next”:

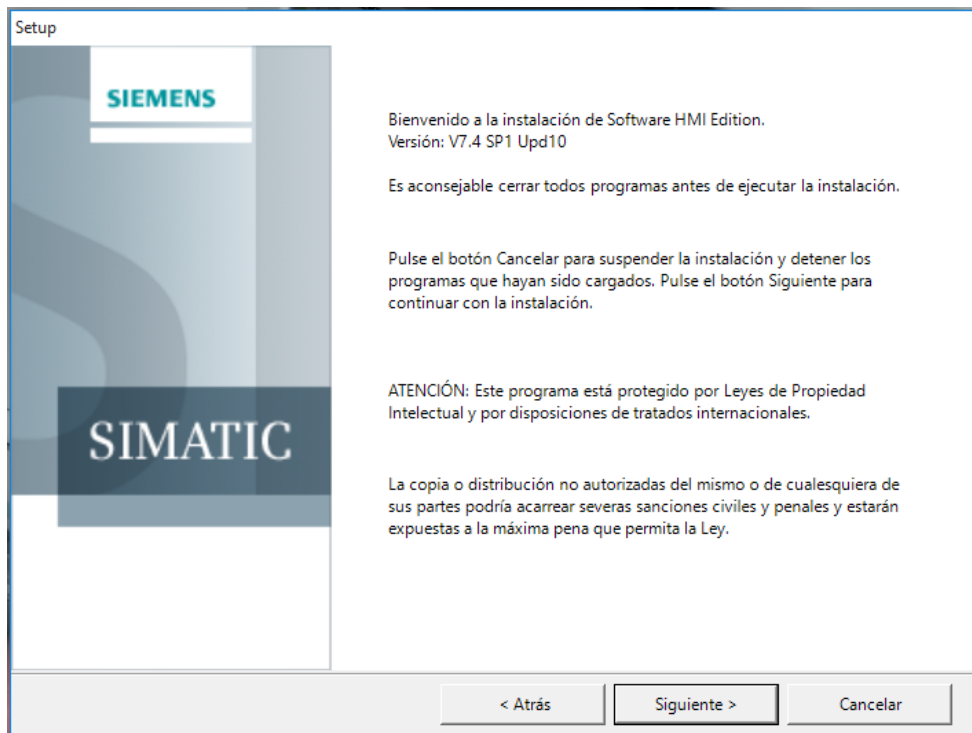


Una vez se extraen los archivos comenzará otro asistente. Se marca el idioma del asistente de instalación y se pulsa sobre “Siguiente”:

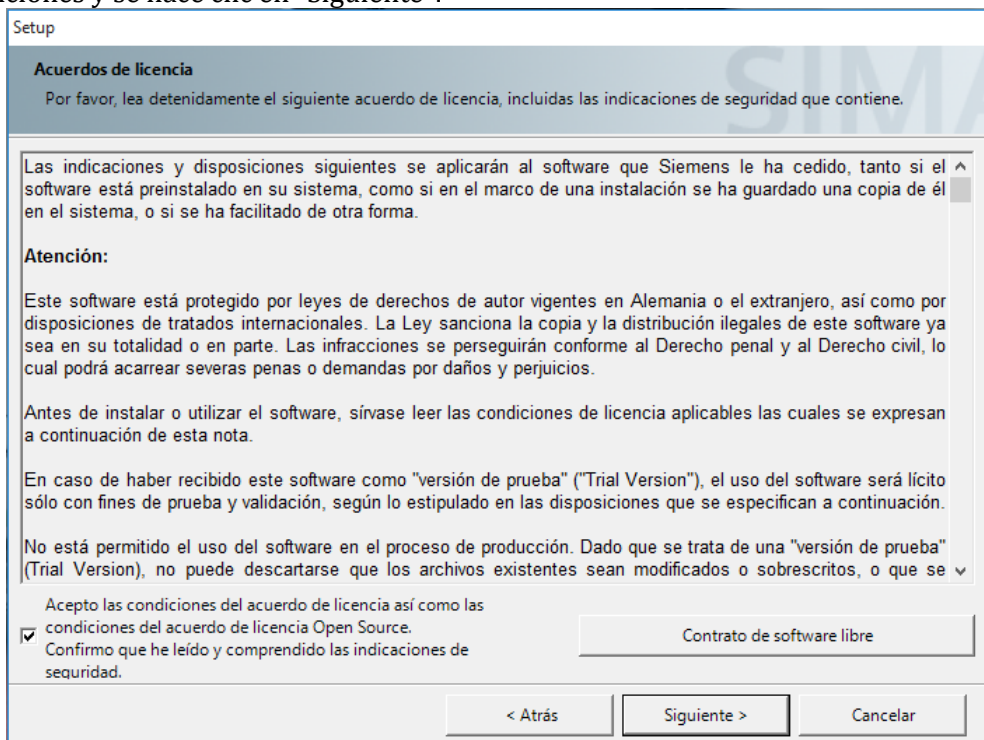


Al igual que durante las otras instalaciones, Siemens recomienda cerrar otras aplicaciones que puedan interferir en la instalación. Pulsamos en “Siguiente”:

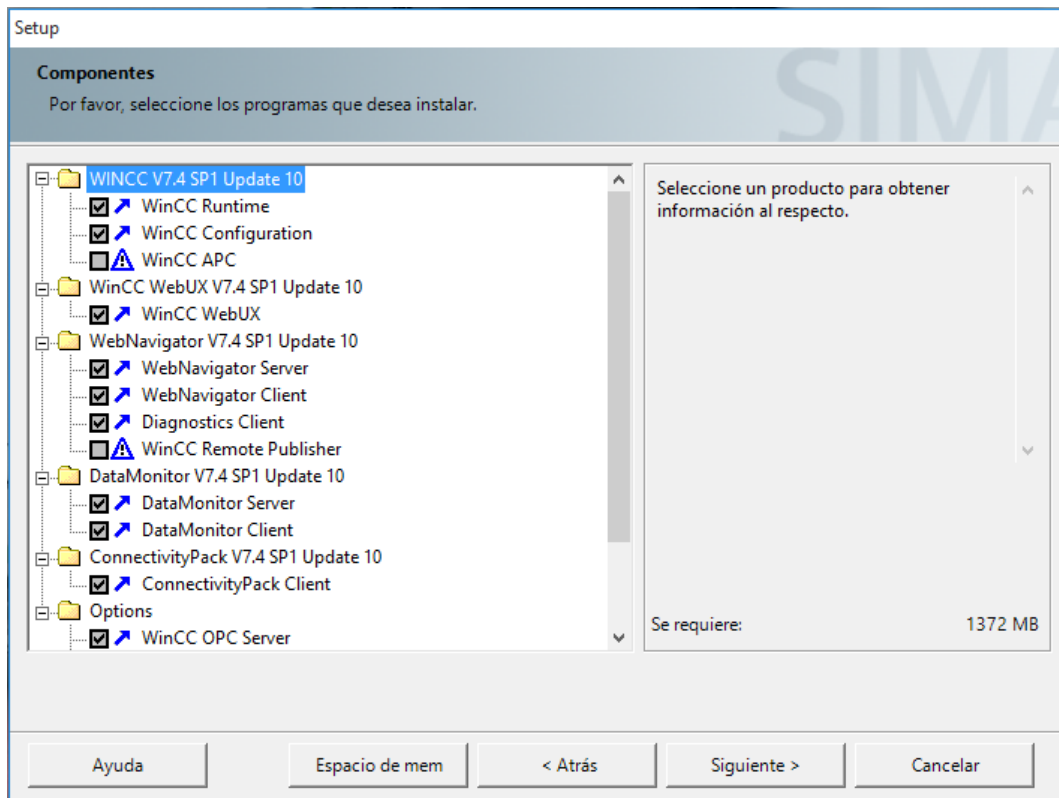




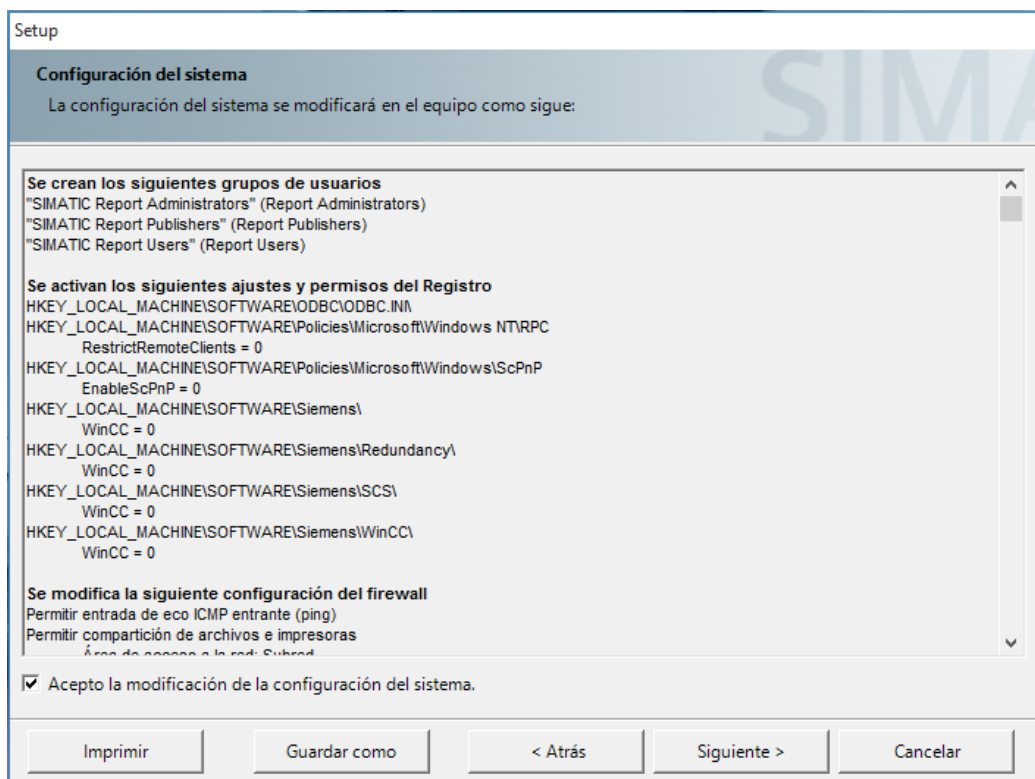
A continuación, aparecen los acuerdos de licencia. Se marca la casilla que acepta las condiciones y se hace clic en “Siguiente”:



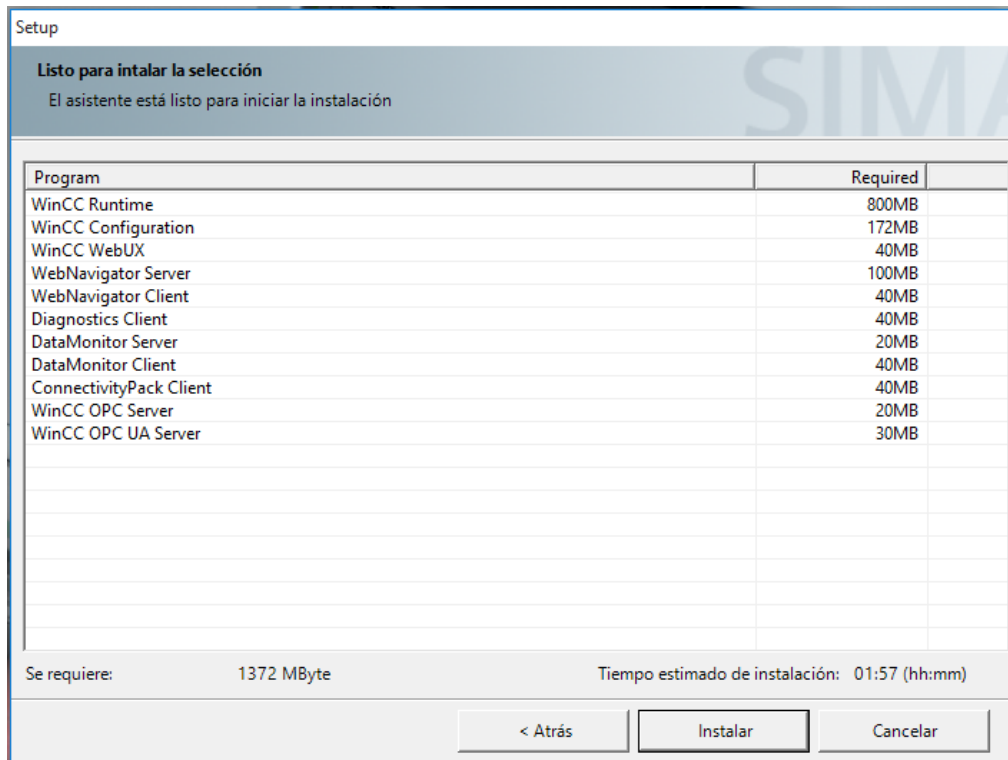
En la siguiente pantalla nos muestra otro software complementario que se puede instalar. En este caso es suficiente el que ya aparece marcado por lo que se hace clic en “Siguiente”:



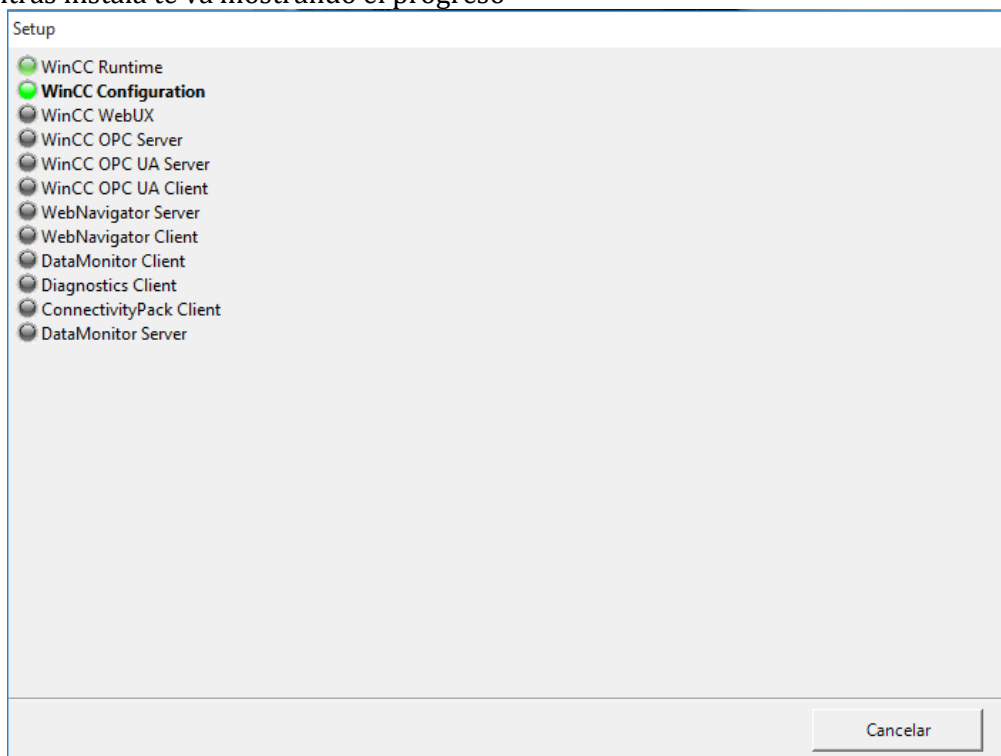
Para continuar, aparece un resumen de la configuración del sistema que se va a cambiar. Se debe de aceptar con el check y pulsar el botón “Siguiente”:



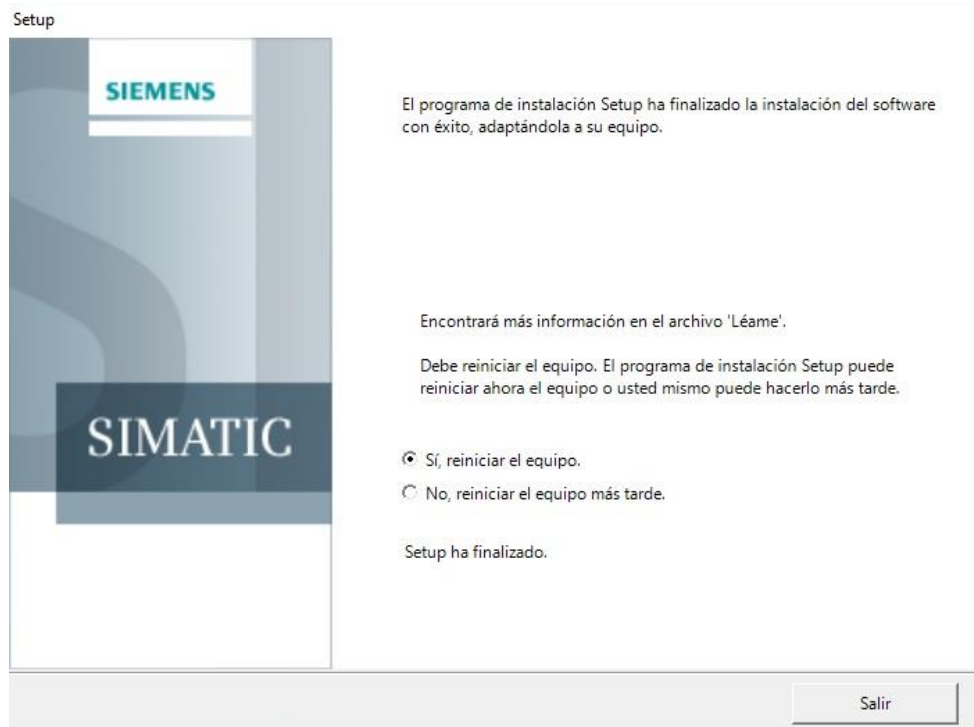
Antes de finalizar con el asistente, aparece un resumen del software que se va a instalar junto con el tamaño del mismo y el tiempo estimado que durará el proceso. Se pulsa el botón “Instalar”.



Mientras instala te va mostrando el progreso



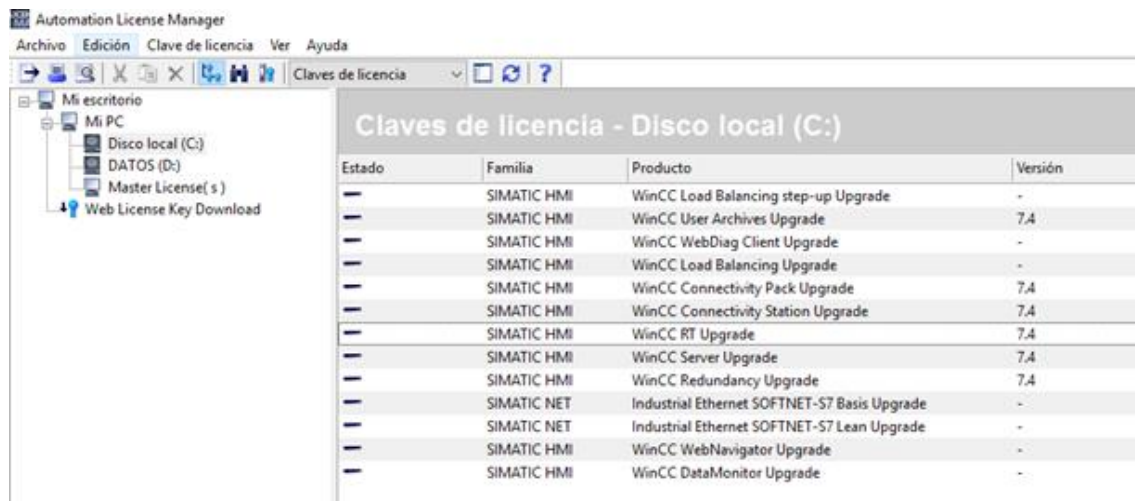
Una vez finalizado el proceso aparecerá la opción de reiniciar el equipo. Se marca ese punto y se hace clic en "Finish".



Una vez reiniciado, tendremos el equipo listo para su uso.

### 2.3.1.6. Instalación licencias WinCC

Las licencias de WinCC se administran con el Automation License Manager.



Con el Automation License Manager se transfieren licencias:

Para utilizar licencias en un equipo con WinCC.

Para eliminar licencias de un equipo, p. ej., para utilizarlas en otro equipo.

Los requisitos necesarios para la transferencia de las licencias son:

- Automation License Manager está instalado.

Solo es posible transferir licencias de WinCC o claves de licencia a otro software SIMATIC en lápices de memoria USB o unidades de disco duro no comprimidas.

La transferencia de las licencias a unidades RAM, unidades de disco duro comprimidas no es posible.

Procedimiento para desinstalar las licencias en el equipo antiguo:

Conecte un soporte para las licencias de WinCC al equipo (por ejemplo memoria USB).

Abra el Automation License Manager en el grupo de programas "Siemens Automation".

Seleccione en la ventana de navegación la unidad en la que existe la licencia que desea desinstalar. Se muestran las licencias de WinCC existentes en la unidad de disco.

Seleccione la licencia deseada en la tabla. También pueden seleccionarse varias licencias.

En el menú contextual de esta licencia, seleccione la entrada "Transferir...". Se abre el cuadro de diálogo "Transferir clave de licencia".

En la unidad de destino, seleccione el soporte de datos de licencia y confirme su selección haciendo clic en "Aceptar".

Se transfiere la licencia deseada y se escribe en la unidad de destino.

Procedimiento para la transferencia de licencias:

Conecte el soporte con las licencias de WinCC al equipo.

Abra el Automation License Manager en el grupo de programas "Siemens Automation". Seleccione en la ventana de navegación la unidad de disco. Aparecen las licencias disponibles para WinCC en el soporte de datos de licencia.

Seleccione una licencia en la tabla. También pueden seleccionarse varias licencias para transferir.

En el menú contextual de la licencia, seleccione la entrada "Transferir...". Se abre el cuadro de diálogo "Transferir clave de licencia".

Seleccione la unidad de destino C: y confirme haciendo clic en "Aceptar".

Se transfiere la licencia deseada y se escribe en la unidad de destino.

Para actualizar una nueva versión de software a partir de una clave de licencia anterior es necesario que tanto la "antigua" autorización o la "antigua" clave de licencia como la nueva actualización tienen que estar almacenadas en el disco duro del ordenador en el que deba llevarse a cabo la actualización.

Nota: Hay que tener en cuenta que las claves de licencia upgrade no se pueden instalar junto con la instalación del programa. Es necesario hacerlo a posteriori

## **Actualización de la licencia**

Para actualizar una clave de licencia proceda del siguiente modo:

1. Elija el comando de menú

Ver > Gestionar

2. En el área de navegación, seleccione la ubicación en la que se encuentra la clave de licencia que quiere actualizar.

3. Seleccione la clave de licencia que desea actualizar en el área del objeto.

4. Ejecute el comando

Clave de licencia > Actualizar

Si sólo existe una posibilidad de actualización, ésta se ejecutará sin más avisos ni consultas. Si se encuentran varias posibilidades de actualización, se abre el cuadro de diálogo "Actualizar clave de licencia".  
 Seleccione la actualización que desea ejecutar y haga clic en el botón "Actualizar".

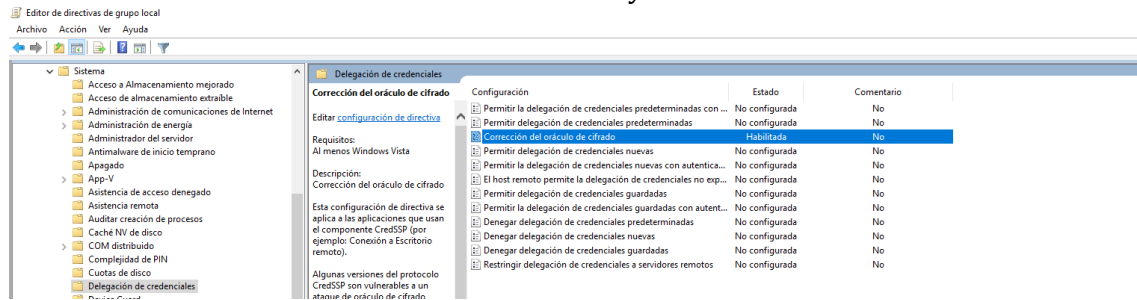
### 2.3.2. Configurar Máquina Server Verea

Una vez instalado el software de SIMATIC WinCC en cada una de las máquinas virtuales es necesario realizar configuraciones y ajustes tanto en los sistemas como en el software de WinCC.

#### 2.3.2.1. Activar el acceso remoto a los equipos

Para realizar el mantenimiento de los servidores y que los clientes ligeros se puedan conectar a las máquinas virtuales de los clientes por escritorio remoto hay que habilitar la directiva de seguridad "Corrección del oráculo de cifrado".

Desde el "Editor de directivas de grupo local" de Windows, en "Delegación de credenciales" se selecciona la Corrección del oráculo de cifrado y se Habilita.



#### 2.3.2.2. Configuración de las máquinas virtuales de los servidores SCADA.

Primero se debe asignar el nombre de la máquina SERVER-A o SERVER-B según corresponda.

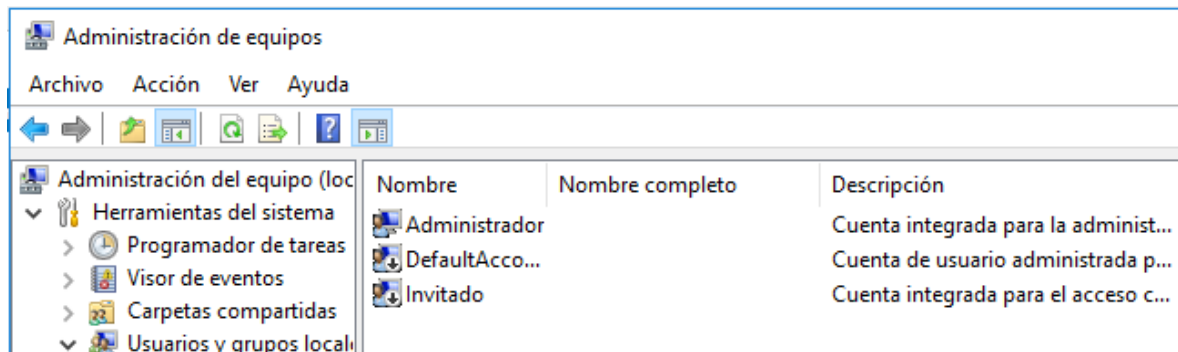
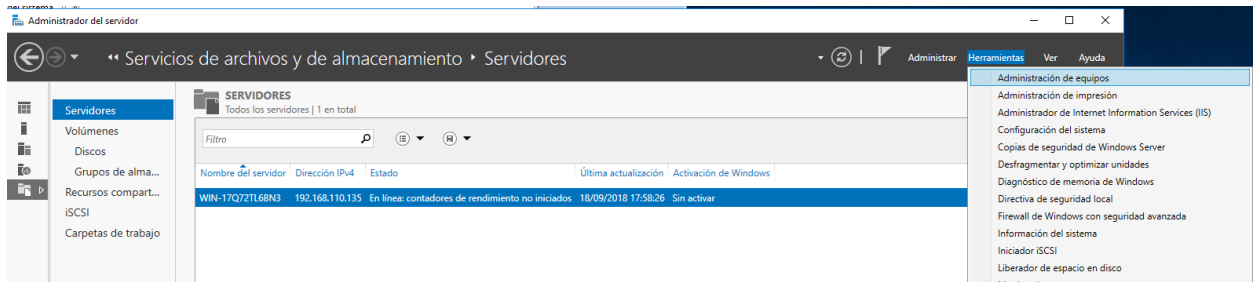
Configurar dirección IP, máscara de red y puerta de enlace de la interfaz de red.

SERVER -A	
IP	192.168.30.10
Máscara	255.255.255.0
Puerta:	192.168.30.254
SERVER -B	
IP	192.168.30.20
Máscara	255.255.255.0
Puerta:	192.168.30.254

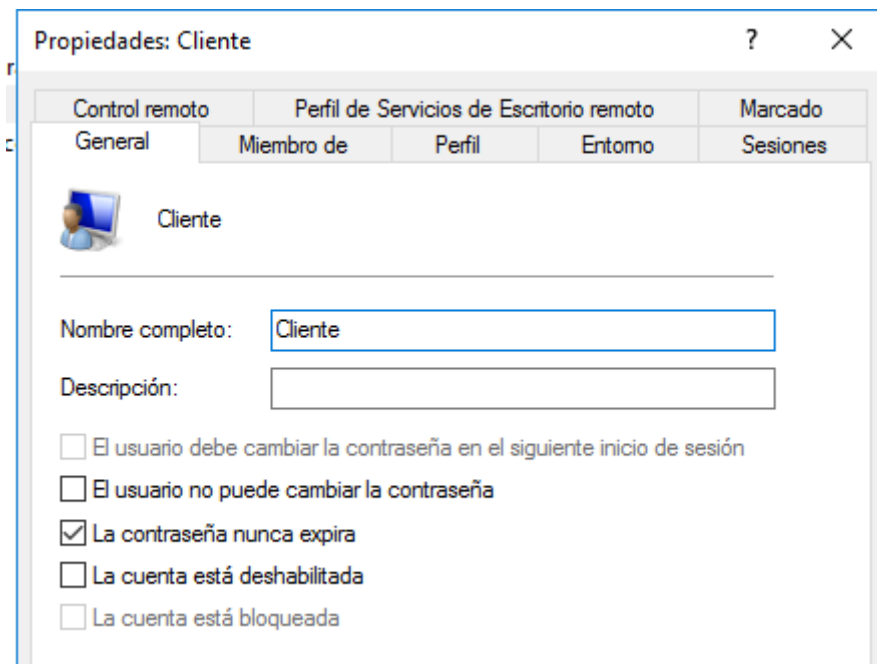
Para permitir copiar y pegar entre máquina y host -> Instalar VMWare Tools

Crear usuario "Cliente" con contraseña y permiso de administrador (En Windows Server: Ventana Administración de servidor ? Botón Administrar ? Administración de Equipos).

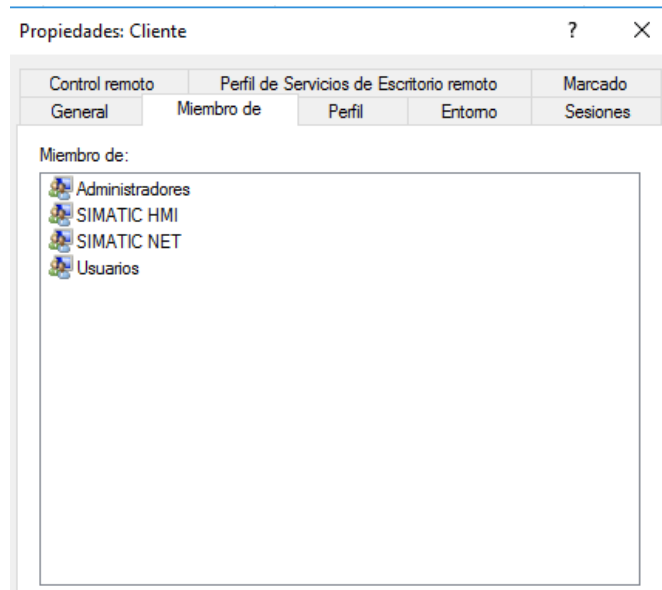
Después de la configuración, el usuario "Cliente" debe ser convertido a un usuario normal del Grupo Usuarios.



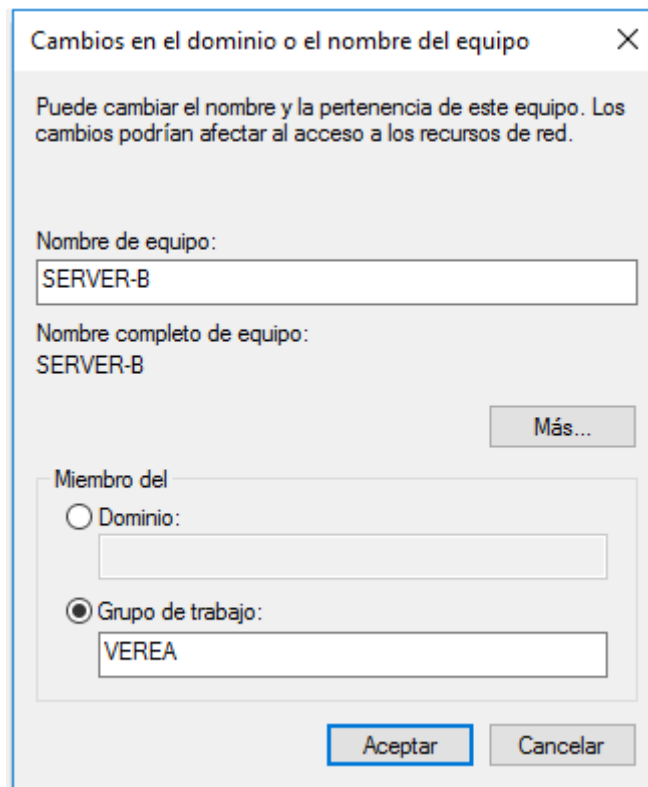
Propiedades Generales del Usuario:



Debe ser miembro de los grupos SIMATIC HMI y SIMATIC NET.

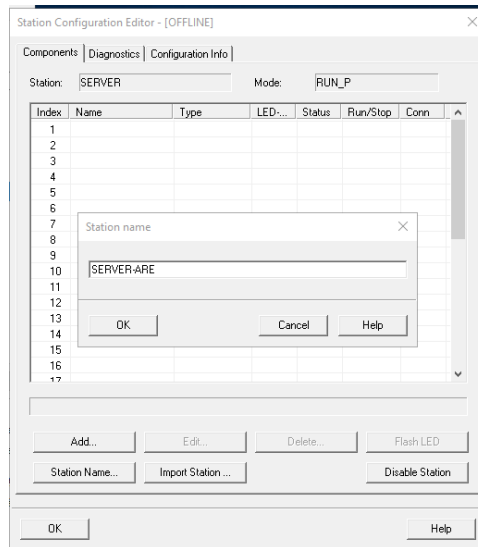


Asignar Nombre del Equipo:

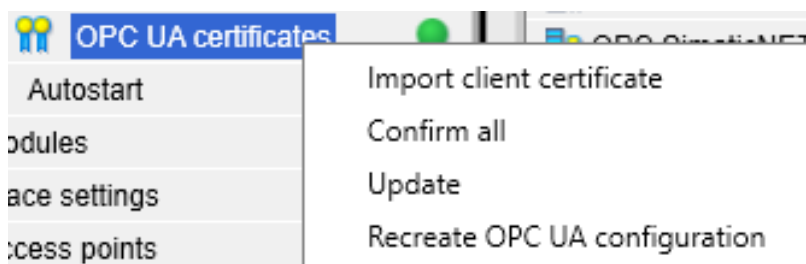
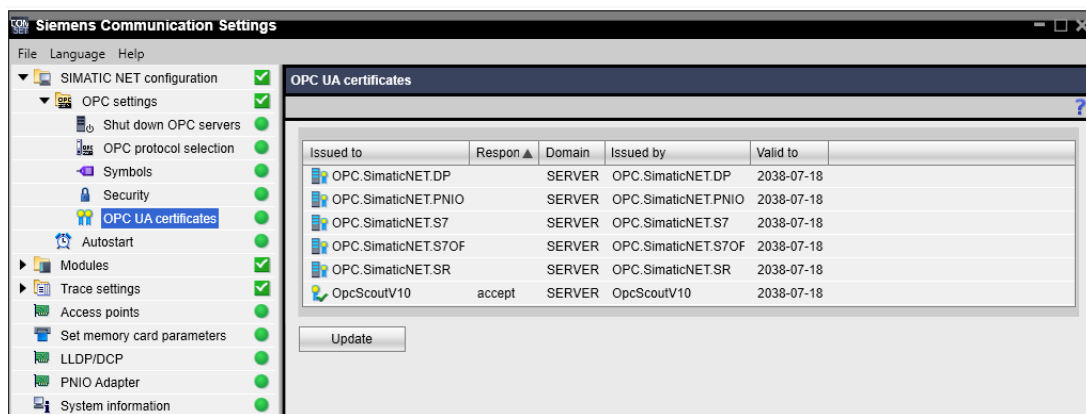
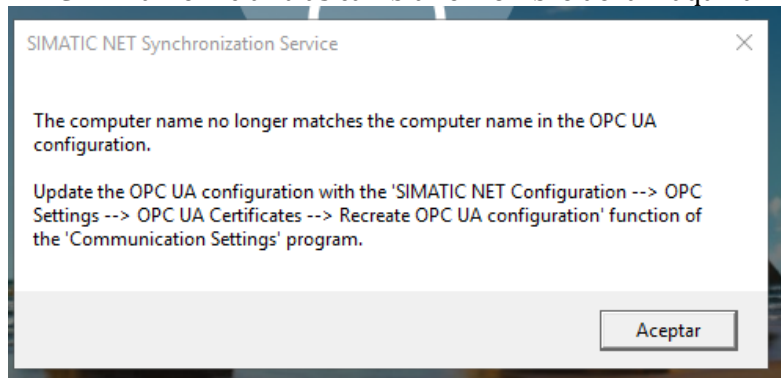


Si cambiamos el nombre después de instalar el WinCC (por ejemplo, por coger una máquina virtual ya configurada, etc) deberemos cambiar el nombre al Station Configuration Editor:



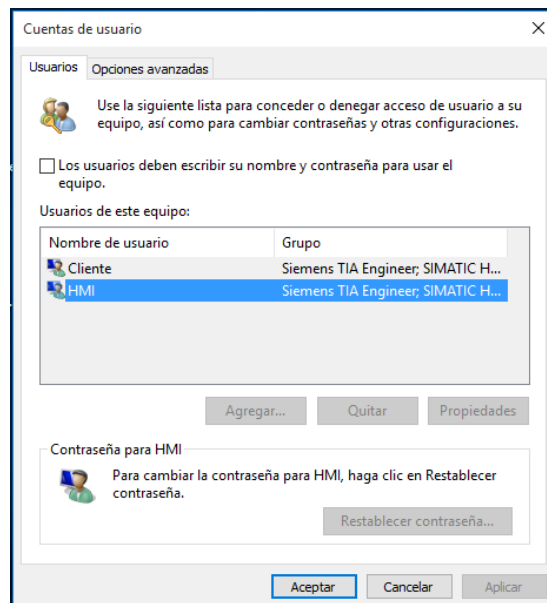


Si cambiamos el nombre tendremos que volver a crear los OPC UA certificates. Esto ya nos lo avisa el SIMATIC NET al reiniciar tras cambiar el nombre de la máquina:



Una vez tengamos el nombre y contraseña definitivos, configurar el **Autologon**.

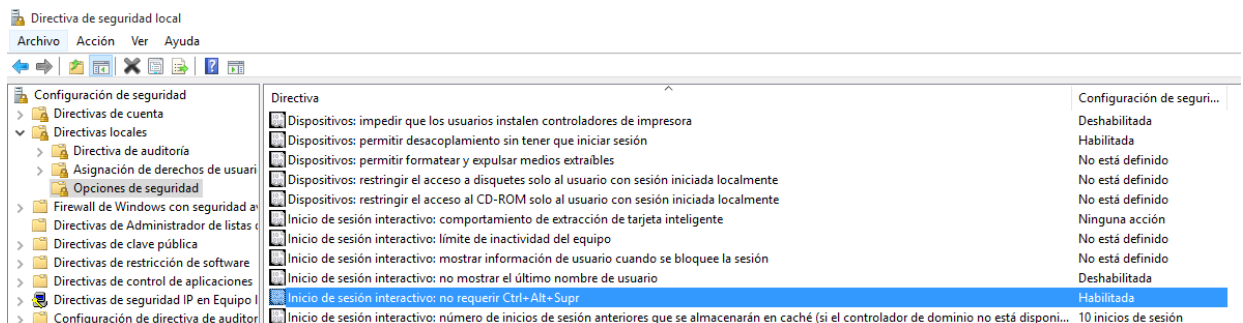
Para ello debemos de ejecutar **netplwiz**. Desactivar “Los usuarios deben escribir su nombre y contraseña para usar el equipo”.



Aceptar

A continuación, Windows pedirá un usuario y contraseña. Introducir usuario y contraseña que usaremos (Usuario: Cliente, Contraseña: ....) y aceptar.

Después desde Panel de Control > Herramientas Administrativas > Directiva de Seguridad Local > Directivas Locales > Opciones de seguridad > Inicio de Sesión interactivo no requerir Ctrl + Alt + Sup -> Habilitar



### 2.3.2.3. Configuración de la resolución de hosts del sistema.

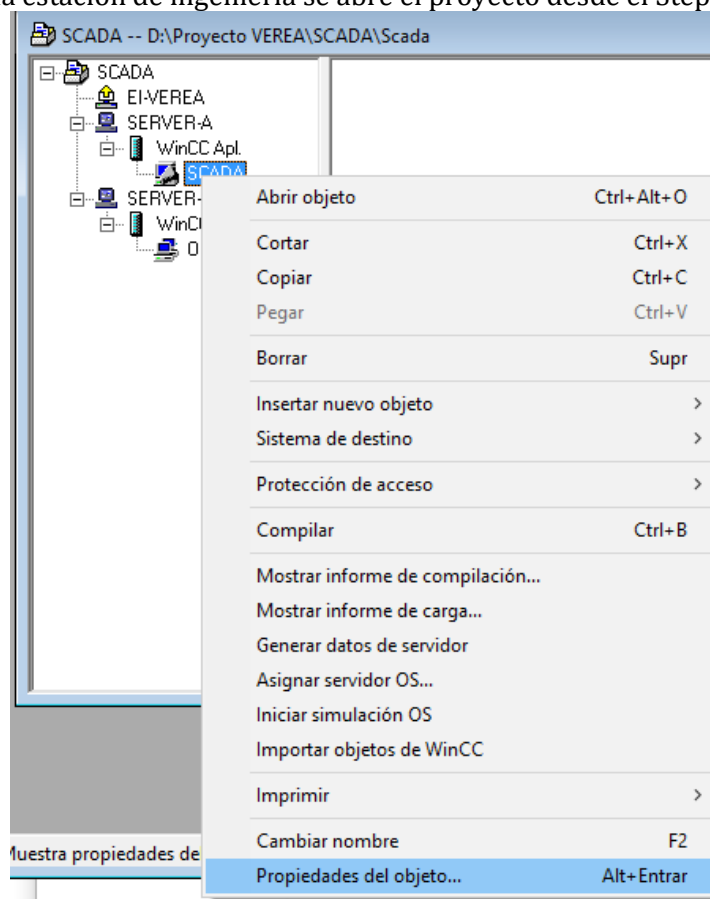
Por último deben de meterse todas las máquinas virtuales en el dominio modificando el archivo HOST situado en el directorio: C:\Windows\system32\drivers\etc\host  
Incluyendo IP y nombre de las máquinas a utilizar tal y como se especifica a continuación:

DIRECCIÓN IP	NOMBRE DE EQUIPO
192.168.30.10	SERVER-A
192.168.30.20	SERVER-B
192.168.30.30	EI-VEREA
192.168.30.40	CLIENTE-MOLIEND
192.168.30.50	CLIENTE-LABORAT
192.168.30.60	CLIENTE-HORNO3

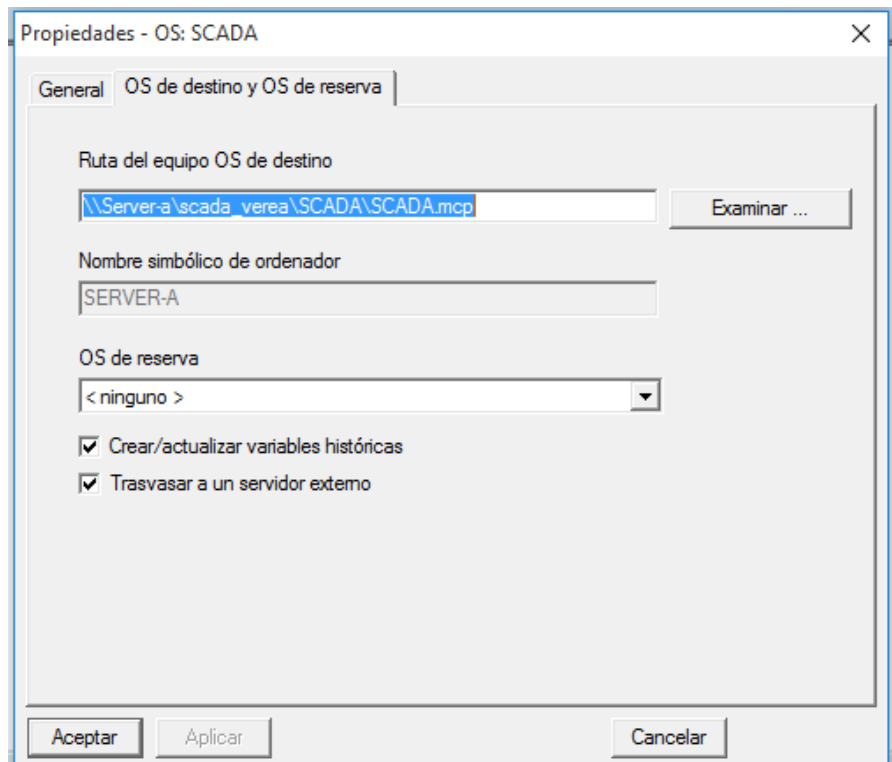
#### 2.3.2.4. Especificación de la carpeta compartida en el Servidor.

Es necesario que exista una carpeta compartida en el servidor donde se cargará el proyecto, que en este caso será "SCADA\_VEREA" en el SERVER-A

Para eso desde la estación de ingeniería se abre el proyecto desde el Step 7:



Indicar la carpeta de destino en el servidor. Si no existe crearla.



Se introduce la ruta del equipo de destino y Aceptar.

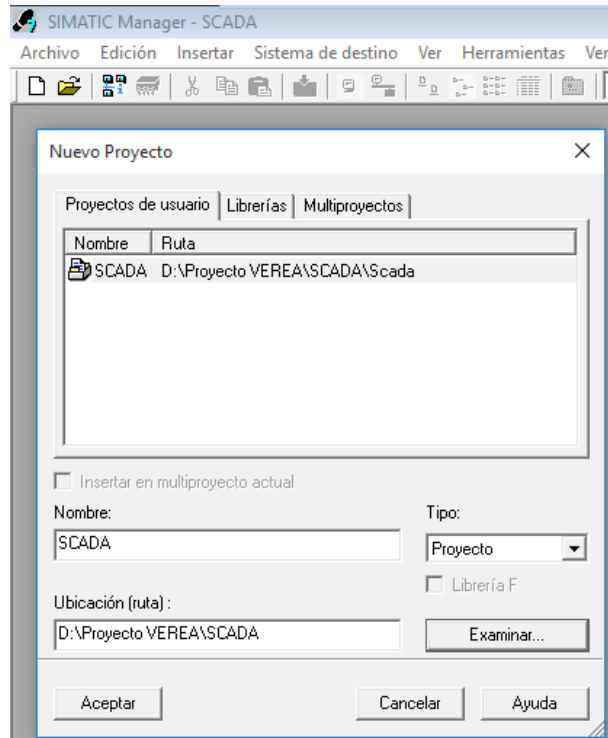
### 2.3.3. Configuración de la Estación de Ingeniería

En la estación de Ingeniería se instala SIMATIC Manager desde el cual se gestionará la carga del proyecto y se configurará la arquitectura de servidores redundantes y sus parámetros de comunicación.

La instalación de SIMATIC Manager es simple y bastará con aceptar las opciones que aparezcan por defecto en el instalador.

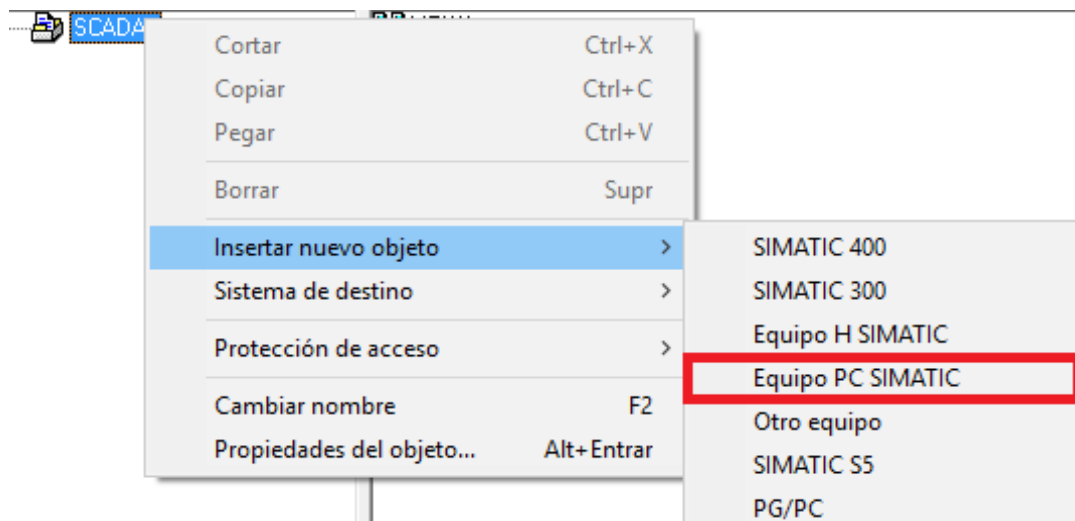
Una vez Instalado lo abrimos y generamos un proyecto de la siguiente forma:

Archivo -> Nuevo Proyecto.

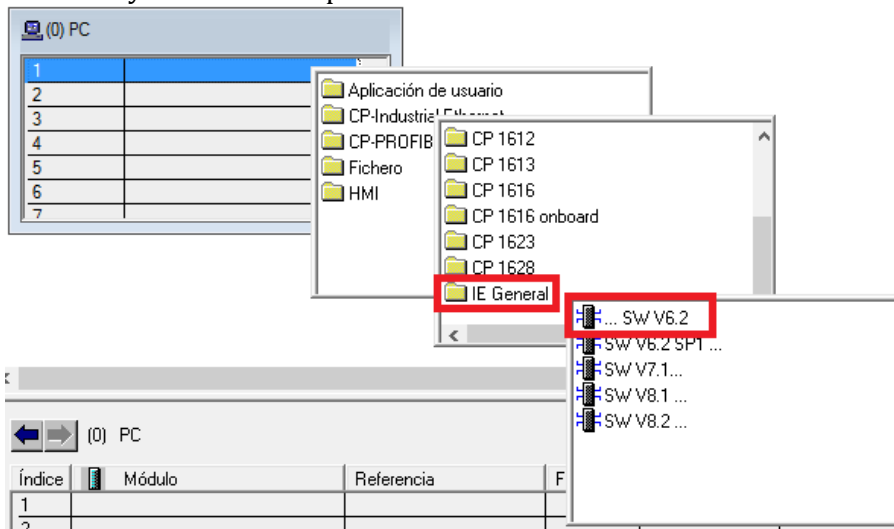


Se le asigna el nombre SCADA y la ubicación de la ruta en el disco D:\. Aceptar.

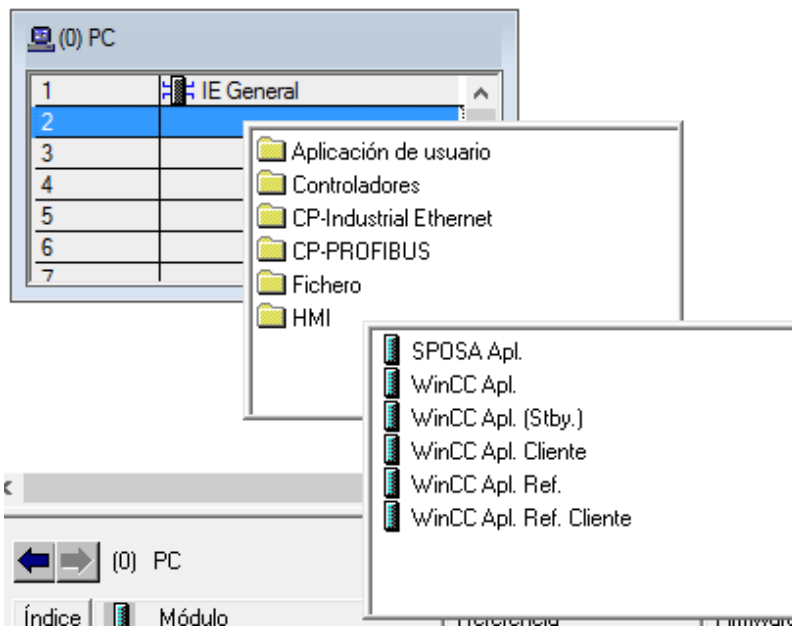
A continuación, en el árbol del proyecto se añaden los equipos haciendo click derecho sobre el nombre del proyecto.



Se selecciona Insertar nuevo objeto -> Equipo PC SIMATIC y se nombran los equipos: EL-VEREA, SERVER-A y SERVER-B respectivamente.

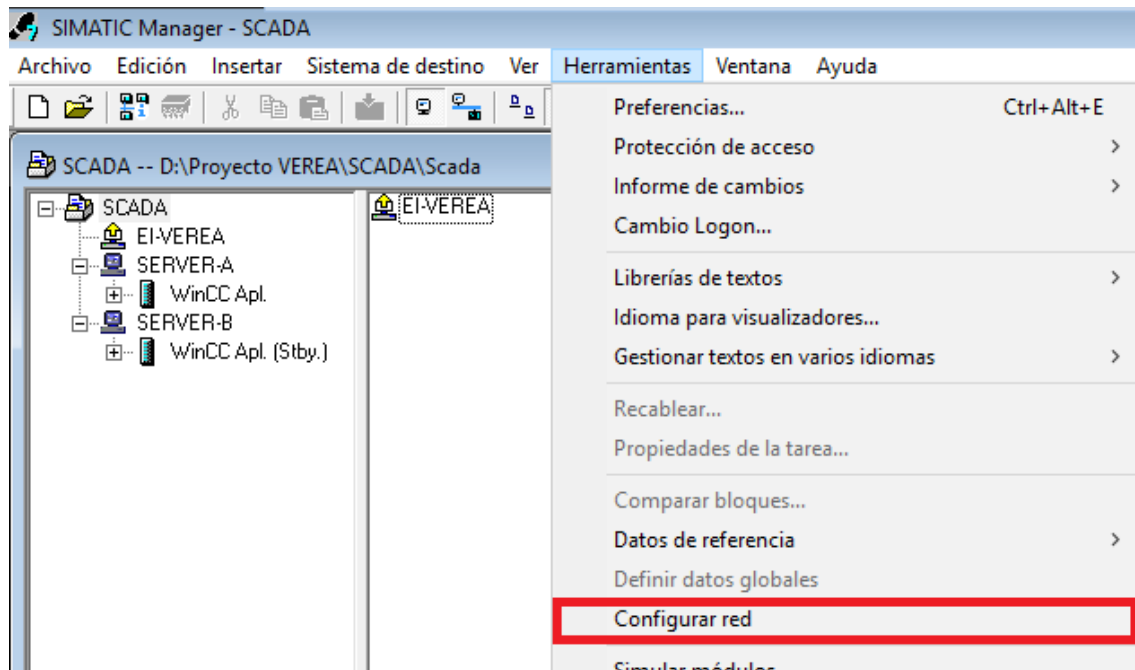


Una vez añadidos los equipos al proyecto se les configura la interfaz de red agregando en el slot 1 de cada uno de ellos un objeto de tipo Industrial Ethernet -> IE General -> SW V6.2

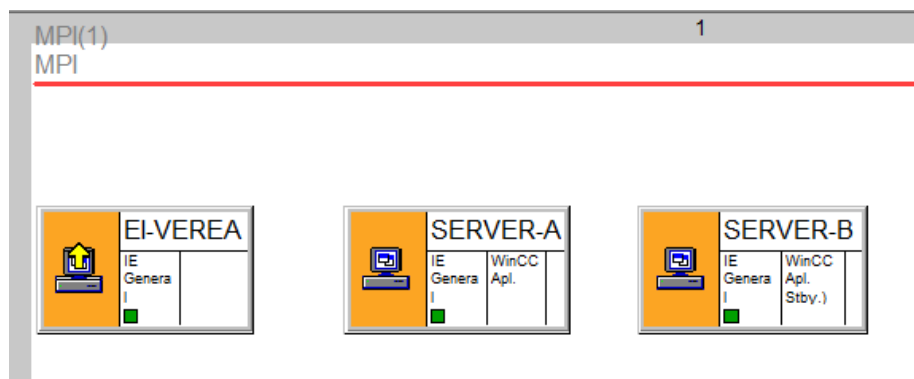


A continuación en la configuración de SERVER-A y SERVER-B se añaden en el slot 2 la aplicación de WinCC. Para ello se hace click derecho en el slot 1 y seleccionamos insertar objeto, se selecciona HMI y elegimos "WinCC Apl." para el SERVER-A y "WinCC Apl. (Stby.)" para el SERVER-B que será aquel que permanecerá en standby en cuanto el SERVER-A se encuentre activo.

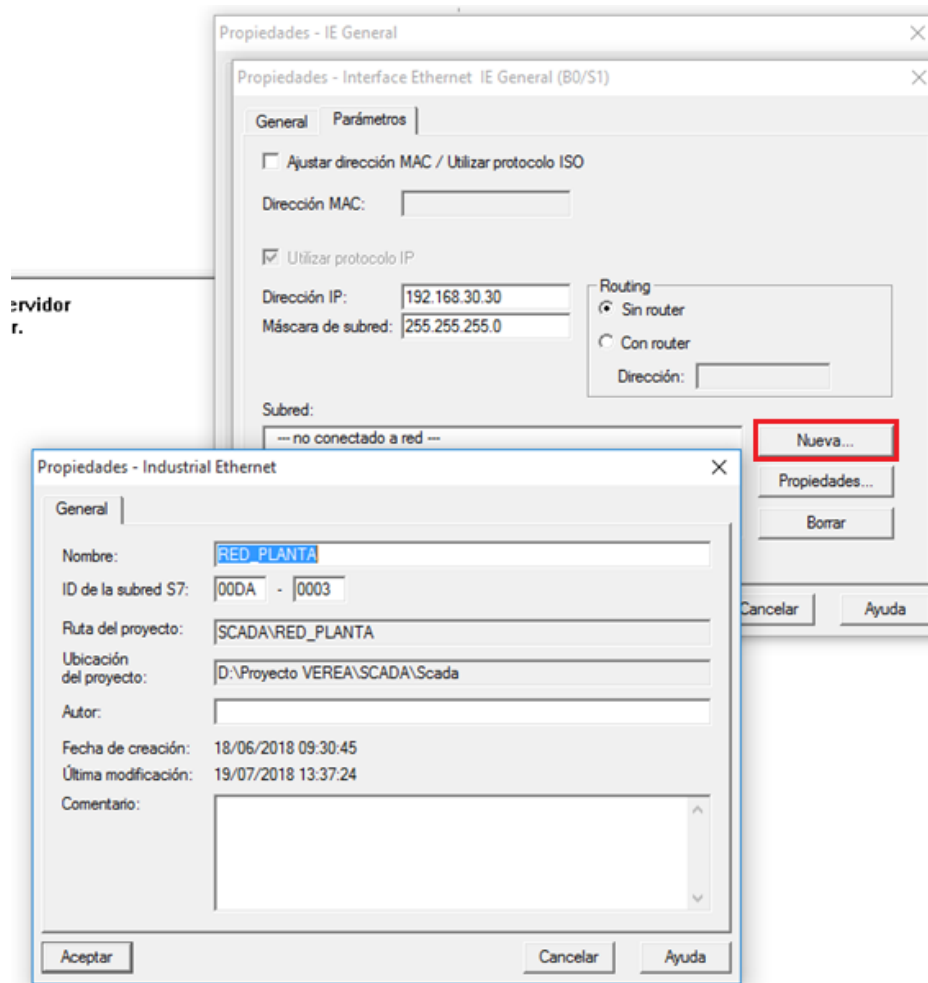
El siguiente paso es configurar la conexión entre las estaciones, para ello se configuran las interfaces con las direcciones IP de los servidores desde SIMATIC Manager, en Herramientas -> Configurar Red.



Se abre el NetPro donde se visualiza la estructura de los equipos creados.

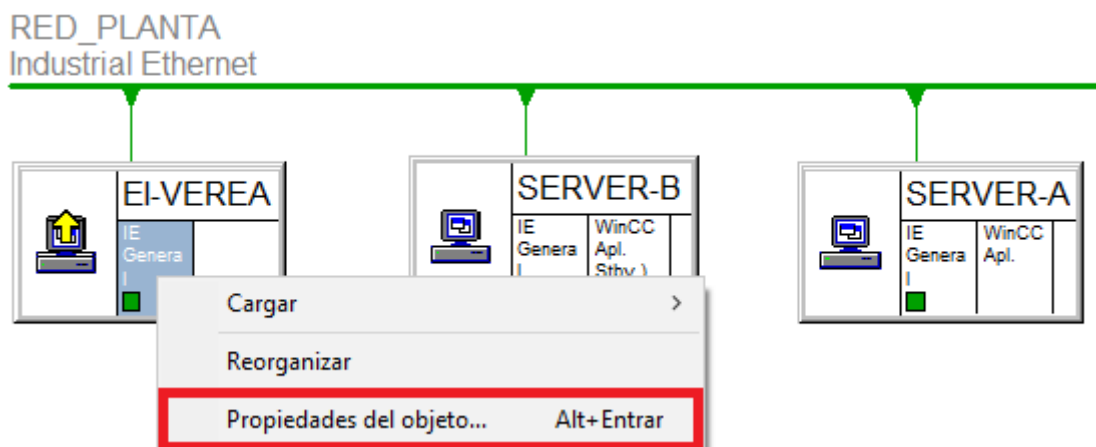


Haciendo doble click sobre las interfaces de Industrial Ethernet que se visualizan en color verde se configura dirección IP de cada equipo y se crea la red de planta a la que quieren conectarse haciendo click en Nueva...



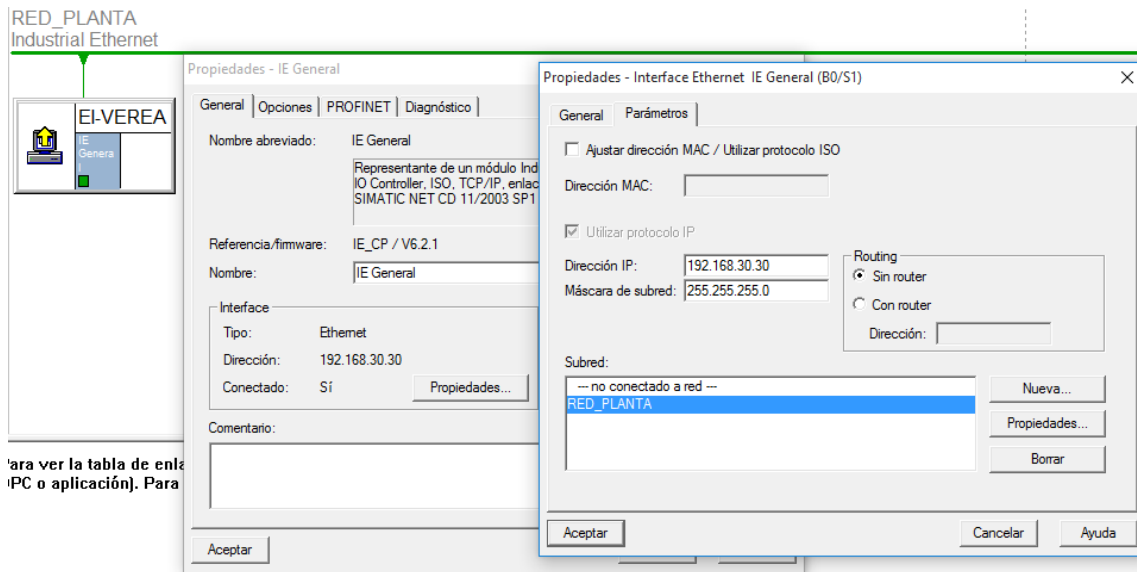
Una vez creada la red de planta, cuando se configuren las interfaces de red de las máquinas restantes esta red ya aparecerá listada en la lista de subredes y solo será necesario seleccionarla para conectar los equipos a ella.

Cuando se configuran todas las interfaces de red de los equipos el diagrama del hardware aparecerá tal y como se muestra en la siguiente imagen.



Al seleccionar la interfaz de red haciendo click con el ratón y posteriormente click derecho encima de ella se puede acceder a su configuración entrando en las propiedades del objeto.





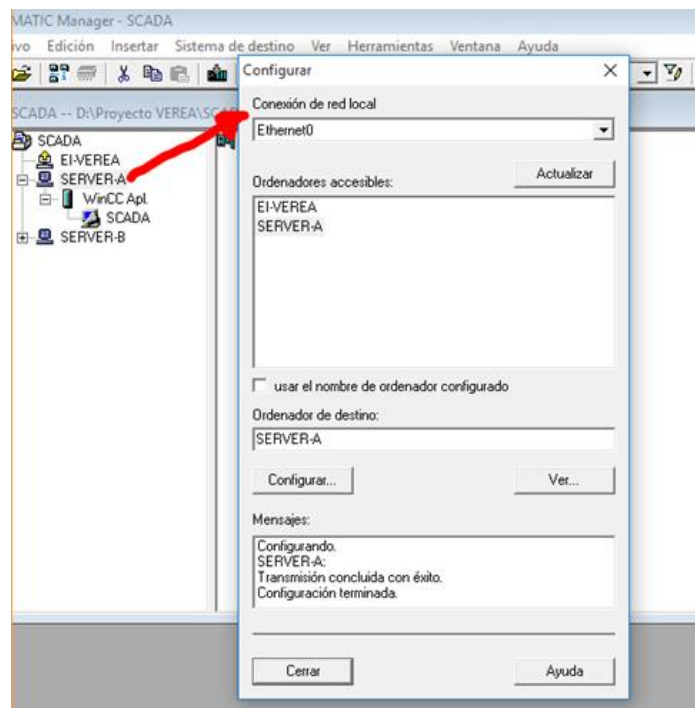
Para ver la tabla de enlaces (PC o aplicación). Para

Tanto los servidores como la estación de ingeniería se le configuran con las siguientes direcciones IP:

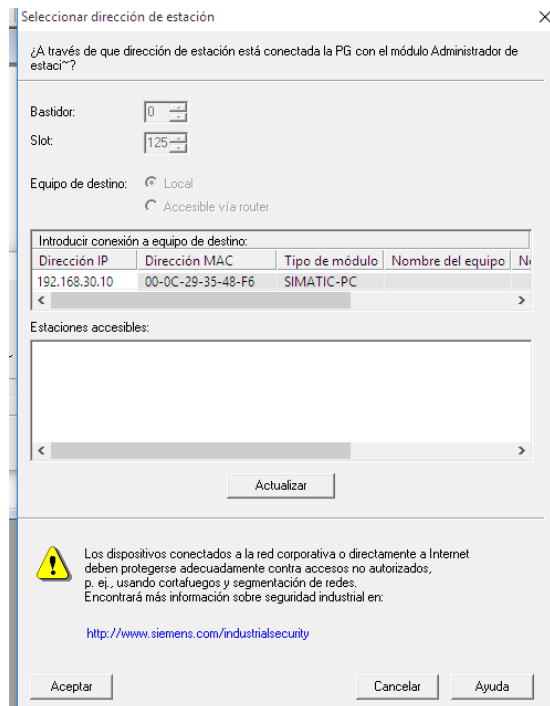
DIRECCIÓN IP	NOMBRE DE EQUIPO
<b>192.168.30.10</b>	SERVER-A
<b>192.168.30.20</b>	SERVER-B
<b>192.168.30.30</b>	EI-VEREA

Una vez configurado en el Server-A la tarjeta en ajustes Pc-PG, desde la máquina de ingeniería Step 7 -> Server-A (icono del ordenador) [ ] Sistema [ ] Configurar...

En el Server-A abrirá la configuración Estación Simatic (barra de tareas) y configurará el servidor a este nivel.



En la estación de destino de los servidores A y B, por defecto mete una dirección IP que no tiene que ser la adecuada y es necesario configurar las que corresponden a cada equipo.

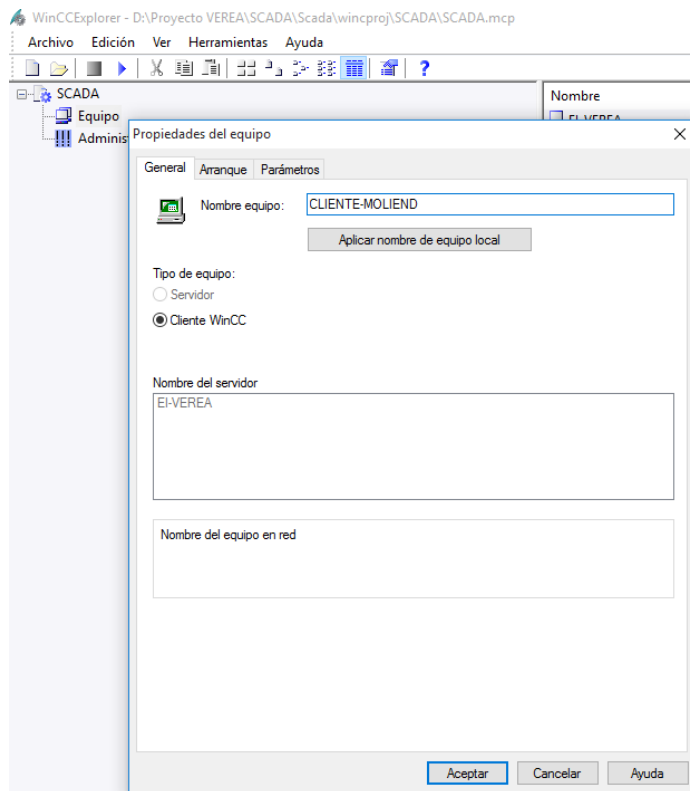


### 2.3.4. Configuración de los Equipos Clientes

El proyecto del servidor se configura en modo “Sin proyecto propio de WinCC”, de forma que todas las imágenes y datos se almacenan en el servidor y en los clientes se abrirá una instancia idéntica del proyecto servidor de WinCC.

Los pasos necesarios para configurar los clientes sin proyecto propio son:

1. Insertar los equipos clientes en la lista de equipos del proyecto del servidor.



Se insertan los siguientes equipos:

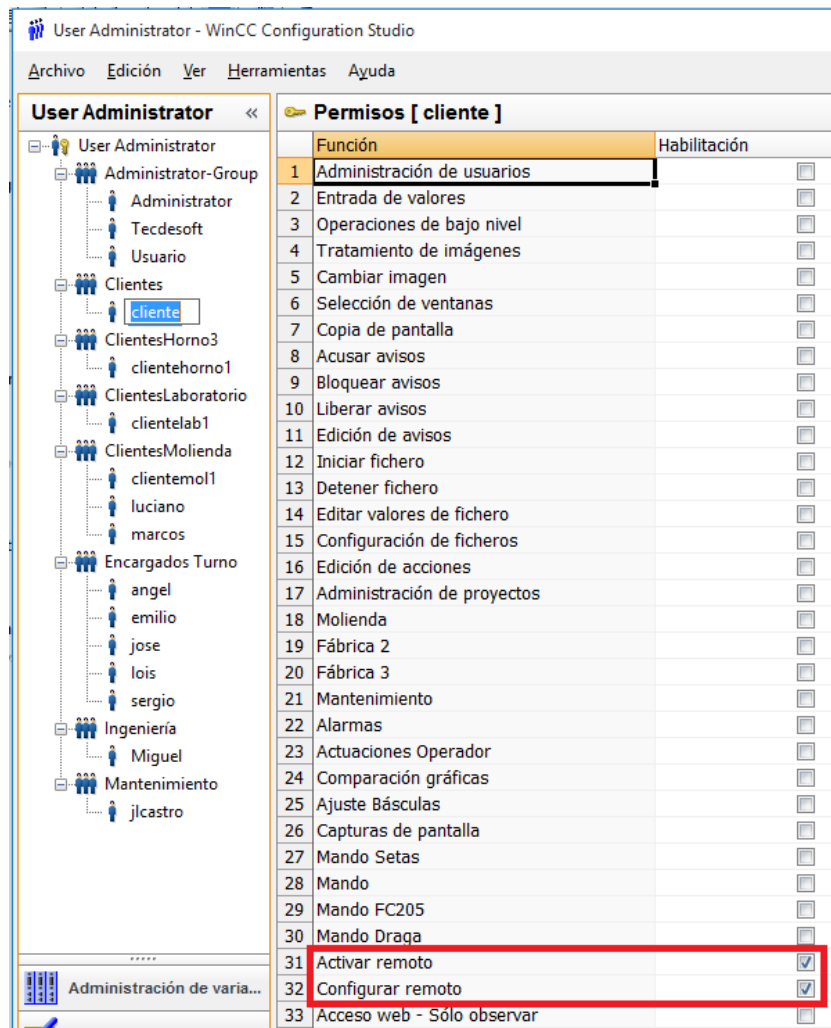
- EI-VEREA (Tipo de equipo: Servidor)
- CLIENTE-MOLIEND (Tipo de equipo: Cliente WinCC)
- CLIENTE-HORNO3 (Tipo de equipo: Cliente WinCC)
- CLIENTE-LABORAT (Tipo de equipo: Cliente WinCC)

2. Asignar permisos de acceso del cliente al servidor al usuario que se configure.

Para que un equipo cliente pueda abrir y editar remoto o en runtime un proyecto de servidor, es necesario asignar permisos de uso para el cliente.

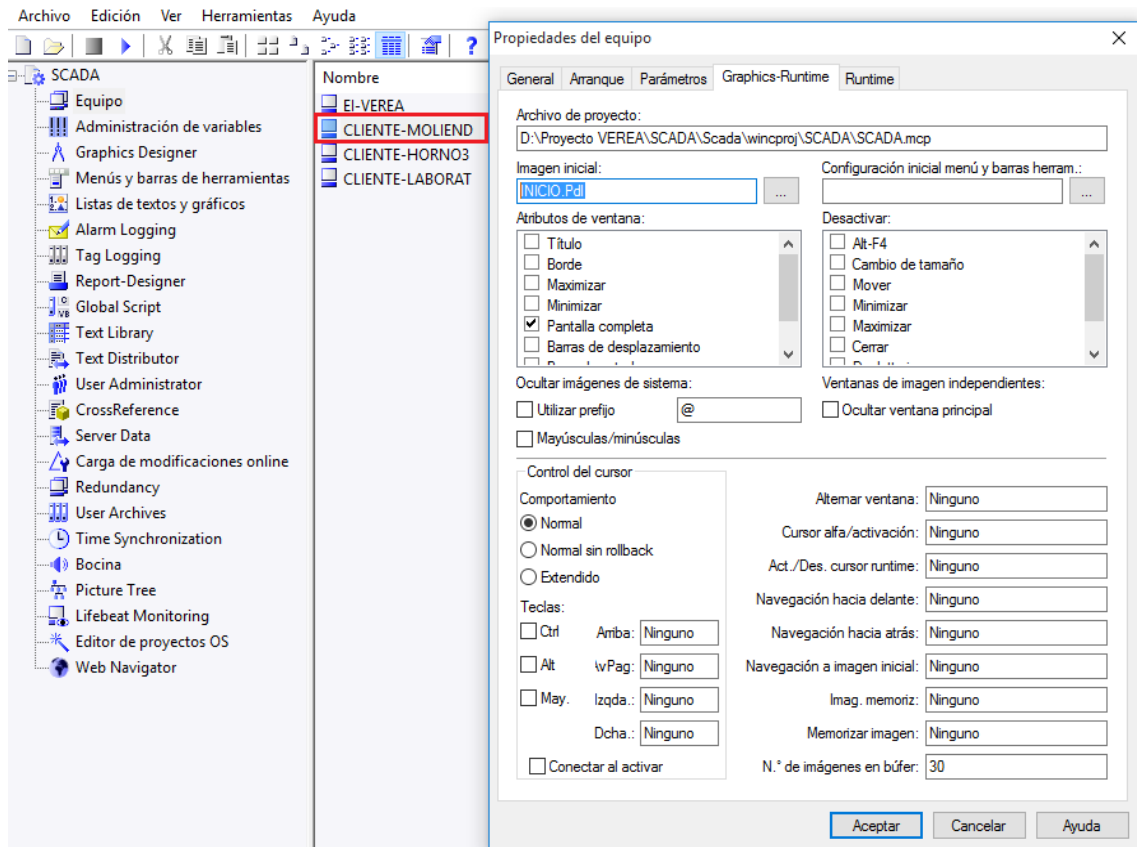
Asignar permisos de:

- **Activar remoto:** El cliente puede activar remoto un proyecto de servidor, es decir, ponerlo en runtime.
- **Configurar remoto:** El cliente puede abrir remoto un proyecto de servidor y tiene acceso completo al proyecto.

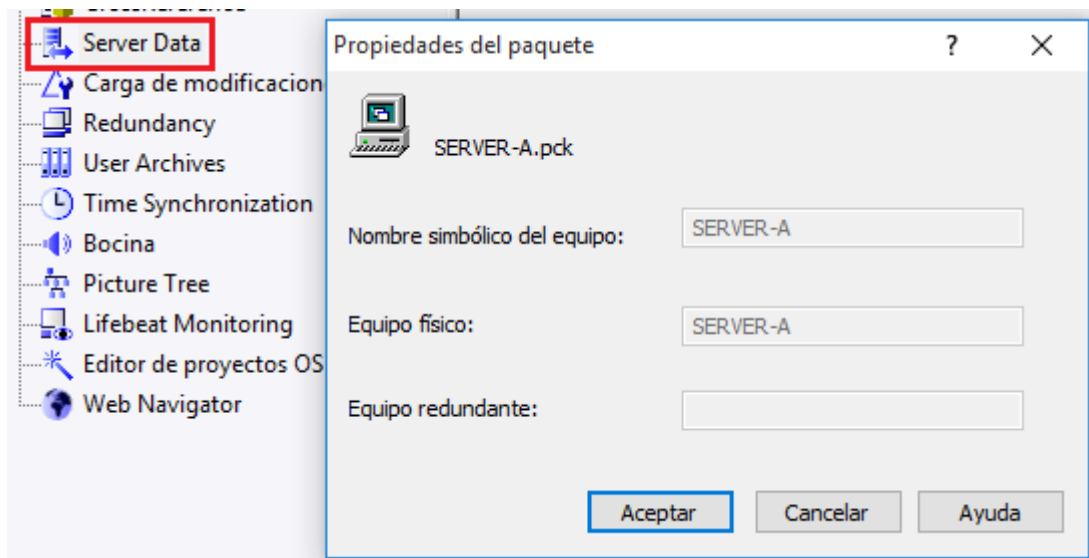


### 3. Configurar propiedades del equipo cliente (Imagen Inicial).

A cada uno de los equipos configurados como clientes se le asigna una imagen inicial entrando en las propiedades del equipo, pestaña Graphics-Runtime y se selecciona la imagen inicial.



4. Generar / Actualizar el paquete de datos.

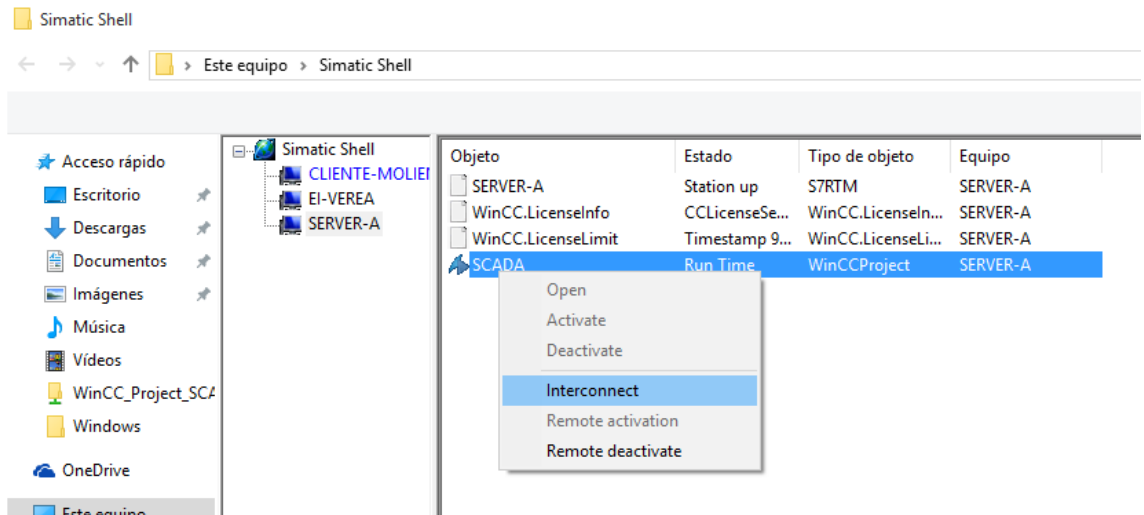


5. Conexión del cliente con el servidor

De igual forma que se realiza en el apartado de configuración de los servidores, hay que configurar el archivo hosts para los cliente (Windows\system32\drivers\etc\host ).

Posteriormente Abrir Simatic Shell, se selecciona el SERVER-A, click botón derecho encima de SCADA e **Interconnect**.

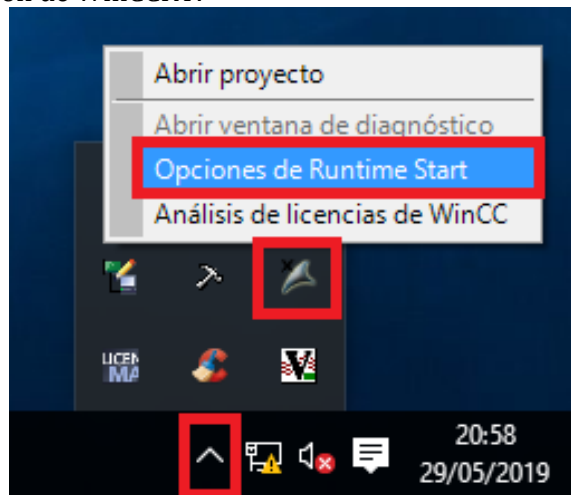
Nos pedirá un usuario y contraseña  
Quedará interconectado el proyecto.

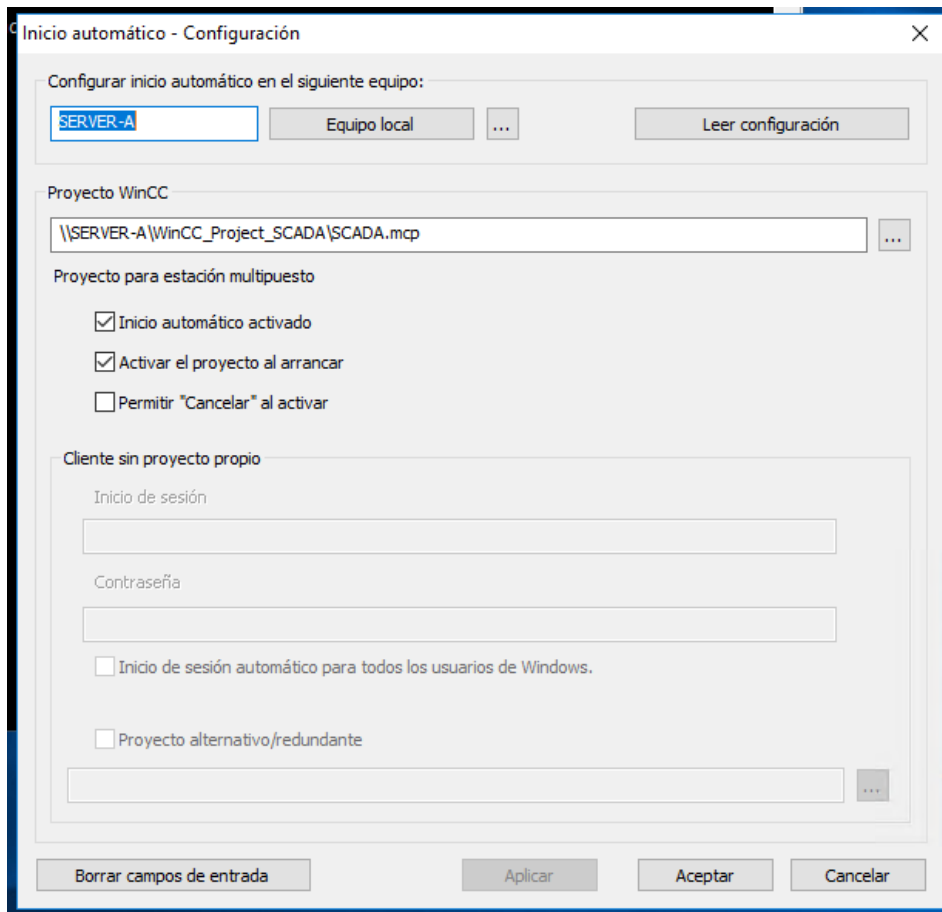


Comprobar accesos por IP, nombre, carpeta compartida `\\SERVER-A\...` Estén correctos.

6. Configurar el arranque automático.

Desde los equipos de los clientes, en “Opciones de Runtime Start” se abre el diálogo de configuración de WinCCRT.





En Proyecto WinCC: se indica la ruta donde está ubicado el proyecto en el SERVER-A. Se marcan las opciones “Inicio automático activado” y “Activar proyecto al arrancar”.

#### 2.3.4.1. Configuración de la caché de imágenes

En la configuración del proyecto en modo de “Sin proyecto propio de WinCC”, los clientes tienen que consultar al servidor las imágenes de los sinópticos y el estado de las variables para realizar las animaciones. Cuando las imágenes son de grandes dimensiones puede notarse un retraso en la carga de las mismas desde el cliente y en este caso es de gran utilidad crear una caché de imágenes (PDLCache) en cada uno de los clientes de WinCC para que no tenga que cargarlas desde el directorio del servidor de WinCC.

Hay tres opciones para configurar PDLCache. El caché se puede utilizar Siempre, Nunca o Preferentemente:

Siempre: indica que si la imagen existe en la carpeta correcta, siempre se utilizará; incluso si existe uno más nuevo en el servidor OS.

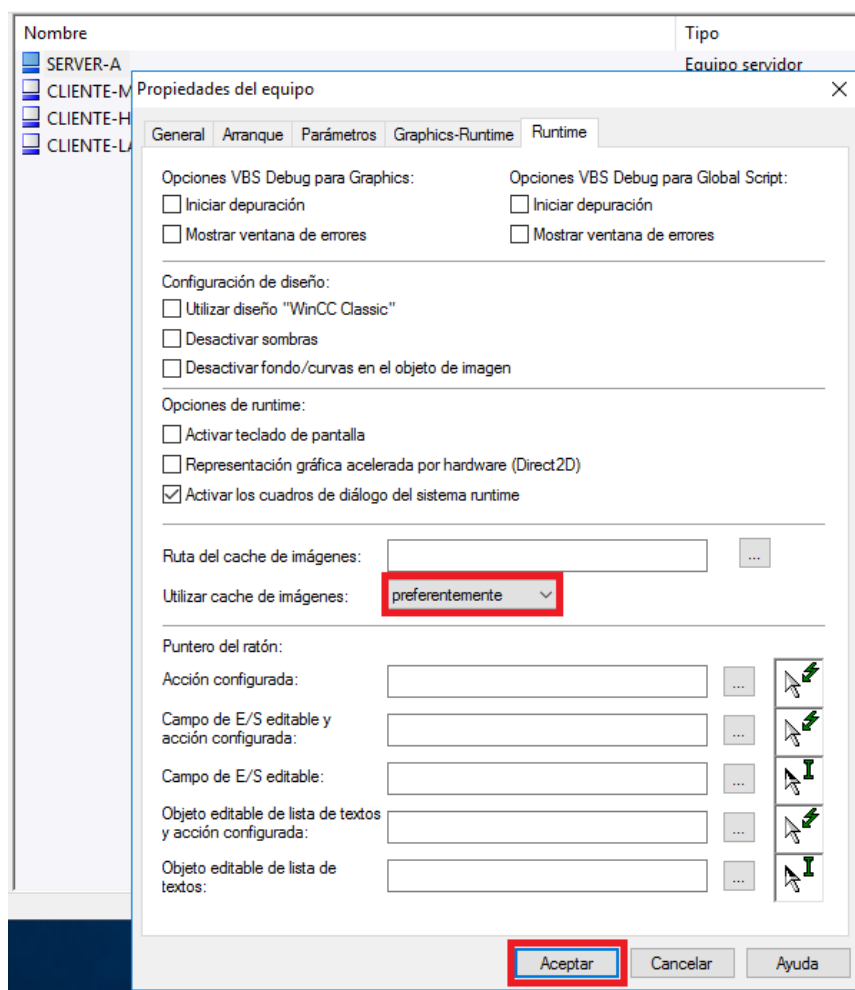
Nunca: no utiliza el PDLCache.

Preferentemente: es la configuración más complicada y predeterminada. Utilizará la imagen de la caché, si tiene una última modificación más actual o la misma última modificación que el servidor, sino la imagen será consultada al servidor.

Para utilizar el modo preferentemente, no se necesita ningún cambio en el proyecto WinCC. Para utilizar el modo Siempre, se debe cambiar la configuración en el proyecto del sistema operativo.

En este caso se configura en el modo preferentemente y aunque es el que tiene configurado por defecto nos aseguramos de que esta seleccionado en propiedades de equipo, pestaña Runtime, campo Usar caché de imágenes, se selecciona preferentemente en el desplegable.

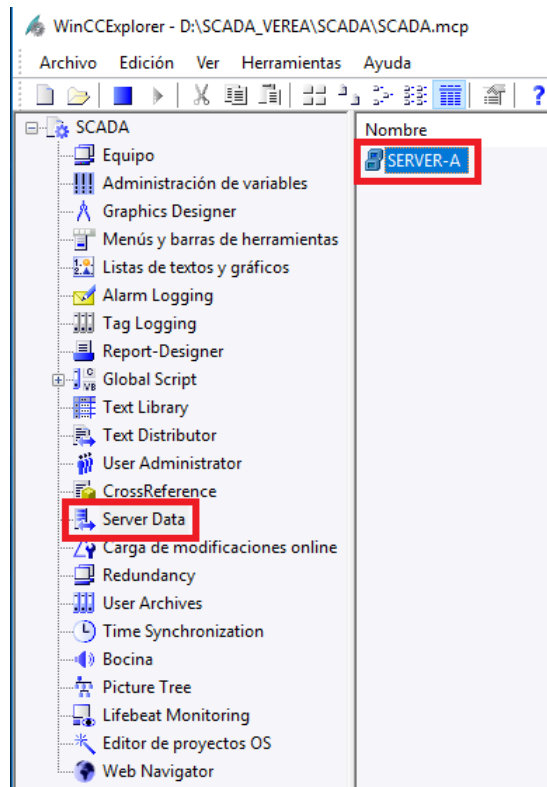
No hay que poner nada en la Ruta del caché de imágenes.



A continuación se configura en cada uno de los clientes la ruta donde estarán almacenadas las imágenes en caché.

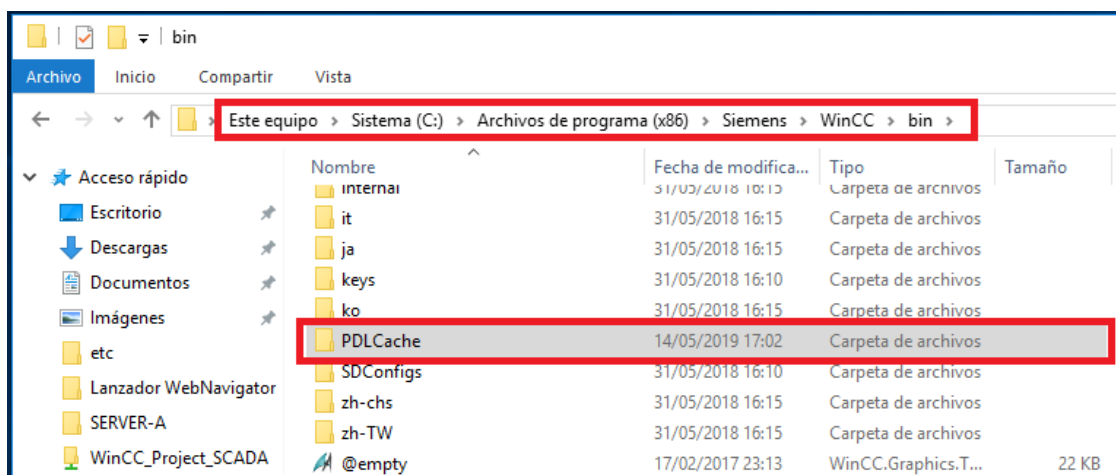
Para ello es necesario saber cual es el nombre del paquete de datos del proyecto.





El nombre del paquete de datos es SERVER-A.

Dentro del directorio de instalación de WinCC en los equipos clientes **C:\Program Files (x86)\Siemens\WinCC\bin** se crea la carpeta PDLCache



Dentro de la carpeta **PDLCache** se crea una carpeta que se nombra con el nombre del paquete de datos **SERVER-A** que contendrá los archivos de imagen .PDL que se necesiten. Si no existe una imagen .PDL en esta carpeta y necesita abrirse desde el cliente, irá al servidor para abrirla.

La carpeta **C:\Program Files (x86)\Siemens\WinCC\bin\PDLCache\SERVER-A** debe existir antes de ingresar el tiempo de ejecución con el cliente. Una vez en tiempo de ejecución, los archivos de imagen se pueden mover dentro y fuera de esta carpeta y el cambio surte efecto en el próximo cambio de pantalla.

Si los gráficos deben actualizarse, deberán copiarse nuevamente a las carpetas de los clientes.

## 2.3.5. Configuración WebNavigator

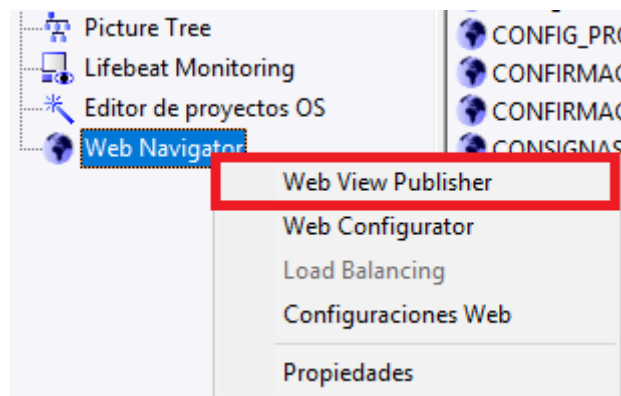
### 2.3.5.1. Configurar WebNavigator en el servidor

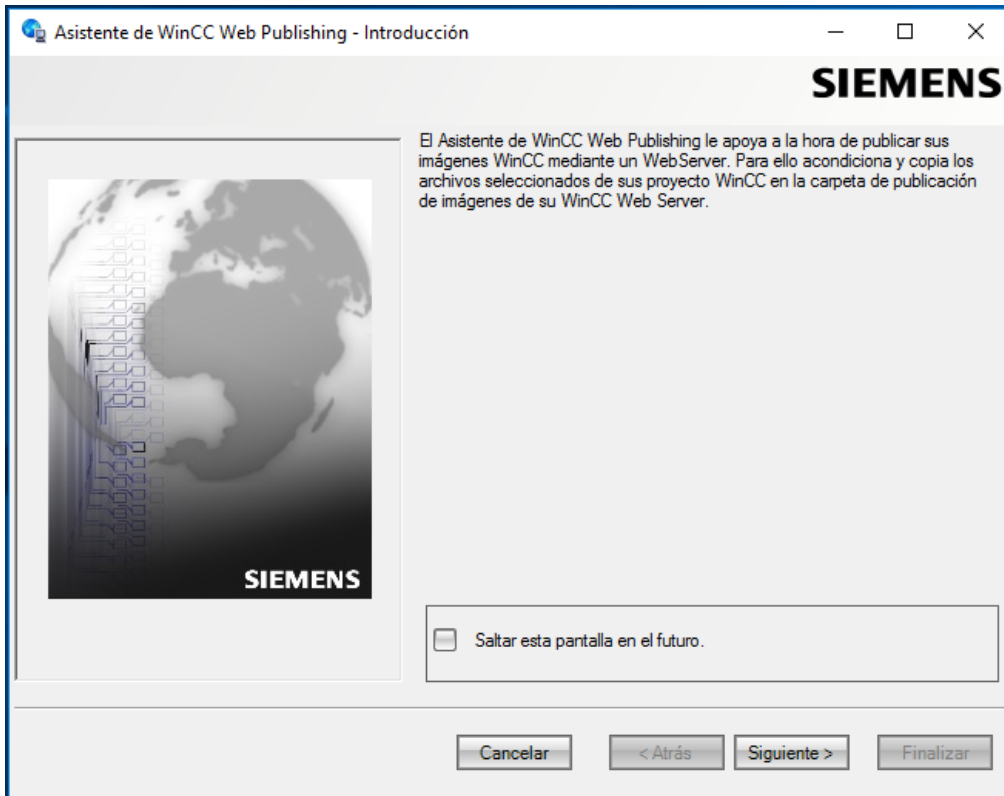
En la máquina Cliente-WebNavigator se abrirá el proyecto WinCC en WebNavigator del que se disponen 13 licencias en este proyecto que permiten se pueda controlar el SCADA desde los paneles de operador distribuidos por la planta al mismo tiempo.

Primero desde el SERVER-A ha de configurarse el servidor de WebNavigator de la siguiente forma:

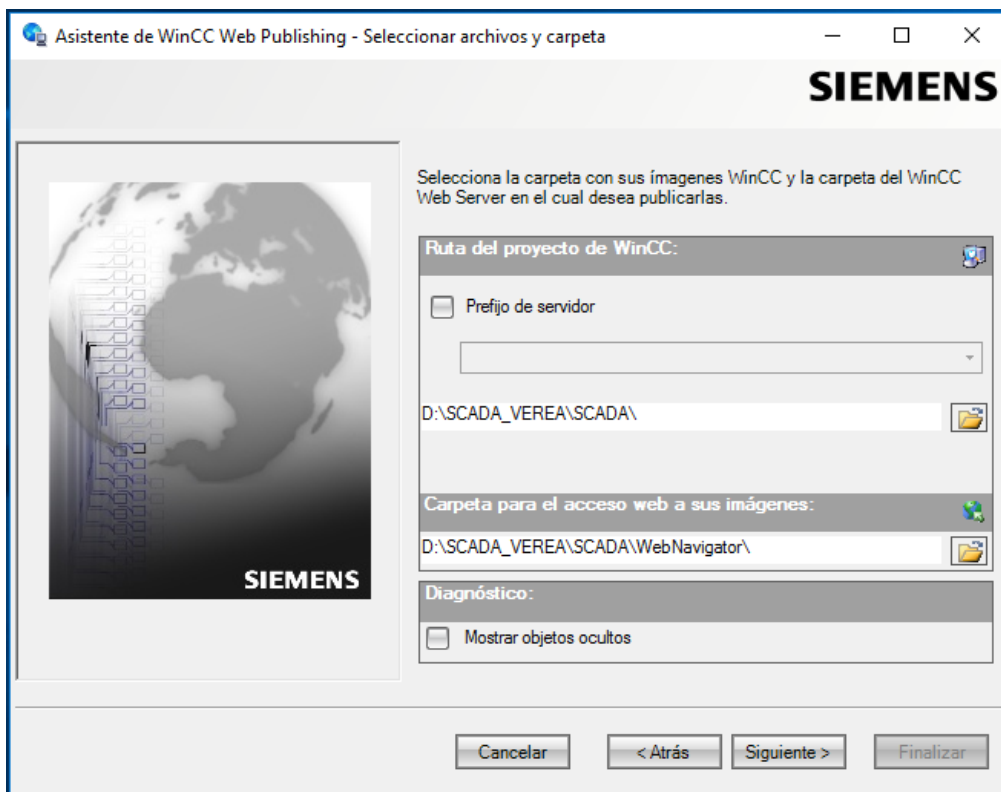
1. Primero desde el WinCC Explorer hay que publicar el proyecto en el servidor Webnavigator.

Para ello desde el arbol del proyecto se hace click derecho encima de Web Navigator y se abre el asistente Web View Publisher.

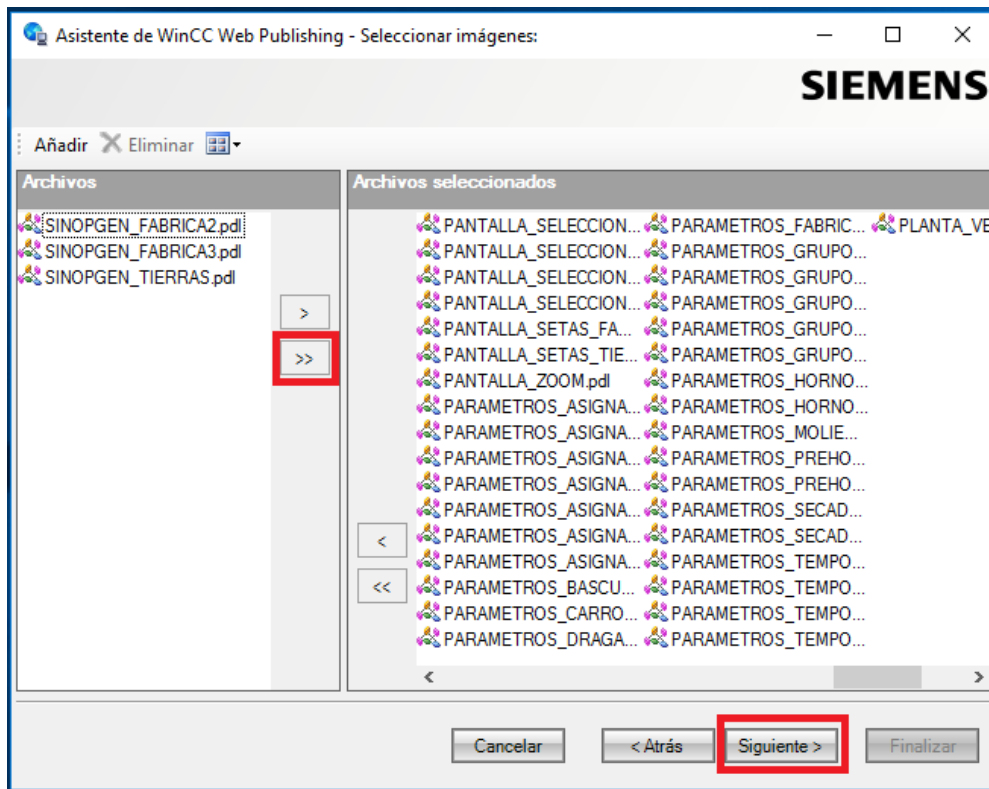




Siguiete.



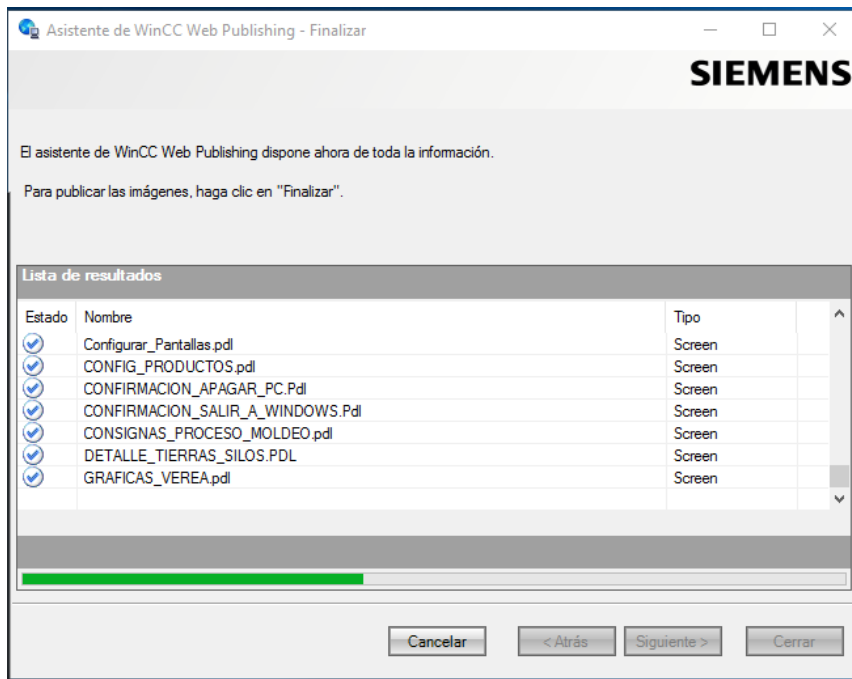
Dejar las rutas que salen por defecto y aceptamos con Siguiete.



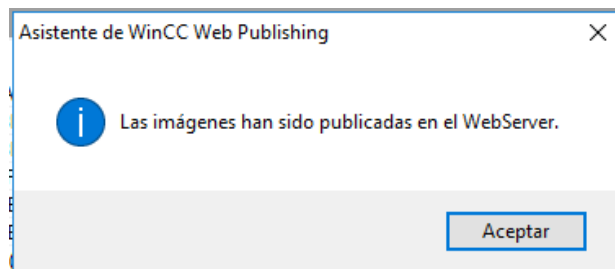
En el area de Archivos saldrán aquellas imágenes que se hayan modificado y no estén publicadas en el WebServer y en Archivos seleccionados aparecen aquellas que ya lo están. Se puede añadir los archivos que se deseen seleccionandolos individualmente y pasándolos a Archivos seleccionados con el botón > o si se desea seleccionarlos todos se pulsa el botón >>. Continuamos con Siguiente.

Este funcionamiento es el mismo para los selectores de funciones y gráficos.

Por último pulsamos Finalizar y comenzará el procesamiento para publicar el proyecto en el servidor WebNavigator.



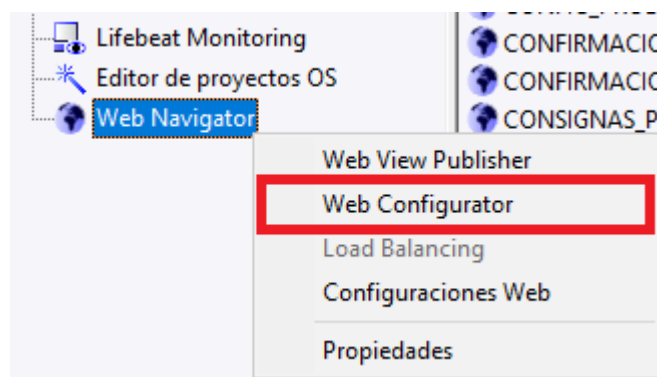
Al finalizar saldrá el siguiente mensaje:



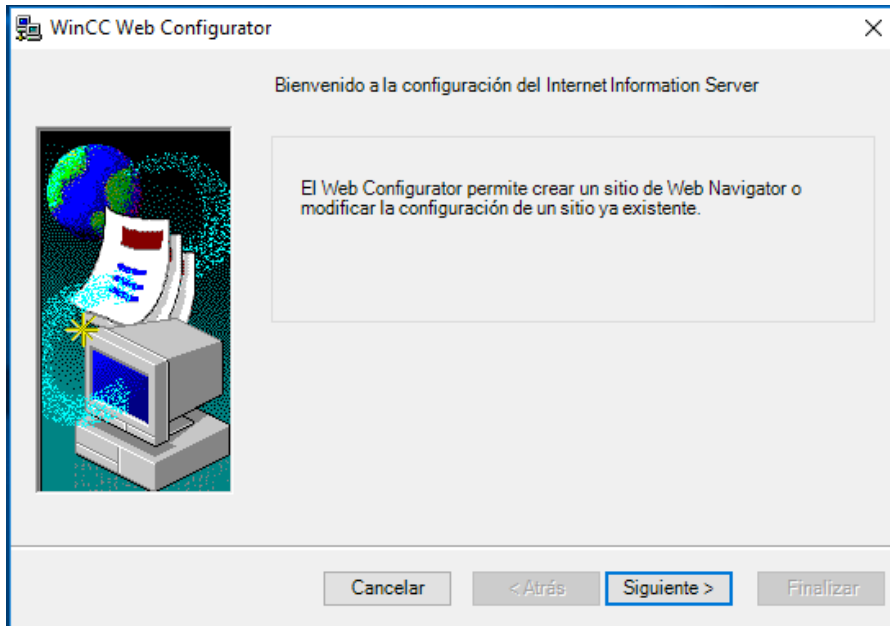
Se Acepta y las imágenes ya quedan publicadas en el directorio configurado para WebNavigator.

2. El siguiente paso consiste en configurar el servidor de WebNavigator.

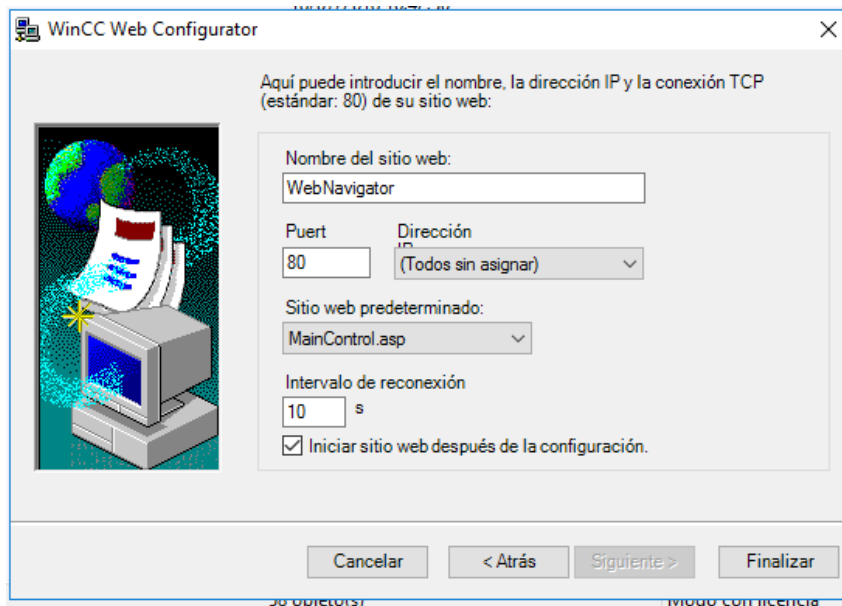
Para ello desde el árbol del proyecto hacemos click derecho sobre WebNavigator y abrimos Web Configurator.



La aplicación WinCC Web Configurator permite crear un sitio web con el SCADA desarrollado.



Pulsamos siguiente.



Configuramos los parámetros del sitio web.

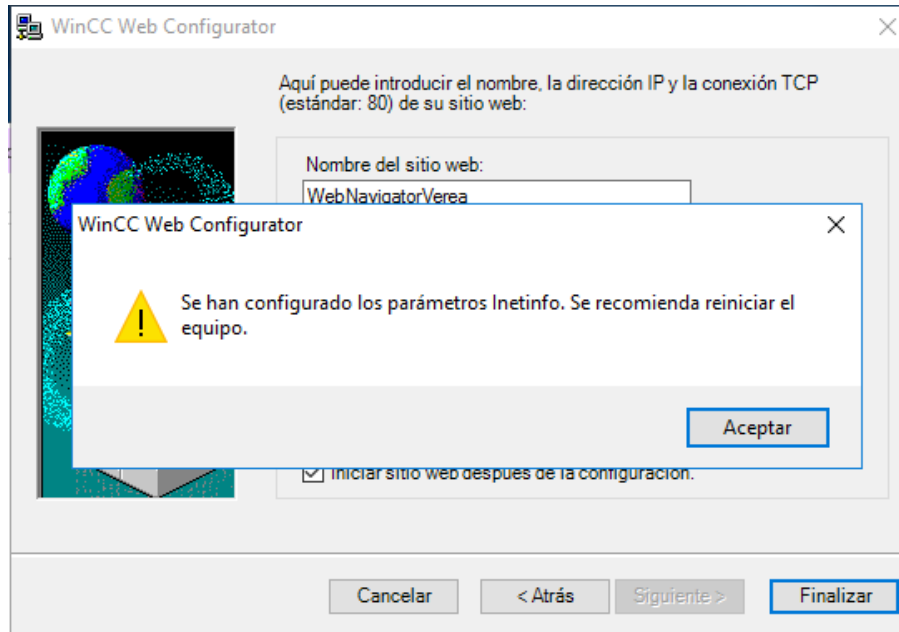
Puerto: 80

Dirección IP: 192.168.30.10

Intervalo de reconexión: 10 s

Se selecciona “Iniciar sitio web despues de la configuración”

Por ultimo se pulsa Finalizar y Aceptar.



Se reinicia el equipo.

Con esta configuración si ejecutamos el SCADA en el servidor tendremos el servidor web funcionando para que se puedan conectar los clientes.

### 2.3.5.2. Configurar cliente WebNavigator

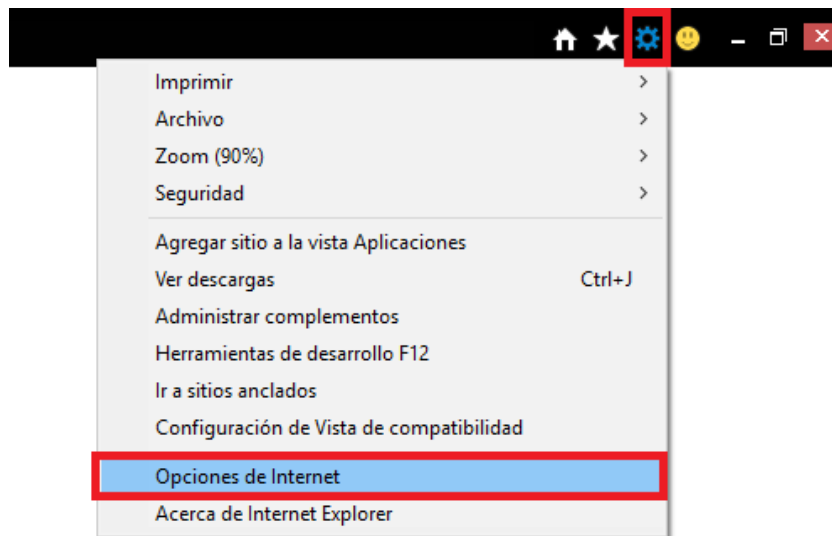
A continuación el servidor del equipo Cliente-WebNavigator ha de cofigurarse de la siguiente forma:

Primero tiene que estar instalado en esta máquina WinCC WebNavigator Client que nos permitira manejar los controles .NET que se hayan empleado en la aplicación.

Para conectarse al servidor de WebNavigator desde el cliente puede realizarse usando Internet Explorer (No vale Edge) o con la aplicación WinCCViewerRT.

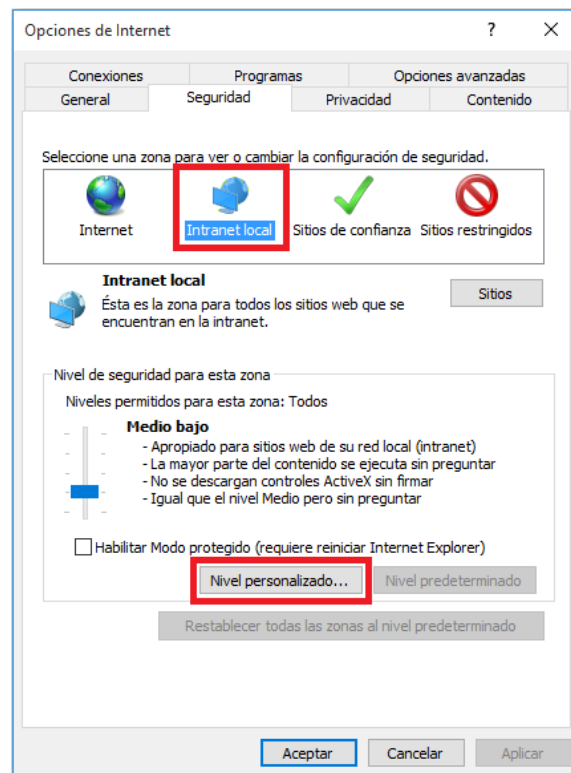
Para disponer de toda la funcionalidad en el cliente WebNavigator, la configuración de seguridad del Internet Explorer se debe adaptar convenientemente de la siguiente forma:

1. En el Internet Explorer, haga clic en "Herramientas" > "Opciones de Internet".



2. Hacer clic en la ficha "Seguridad".

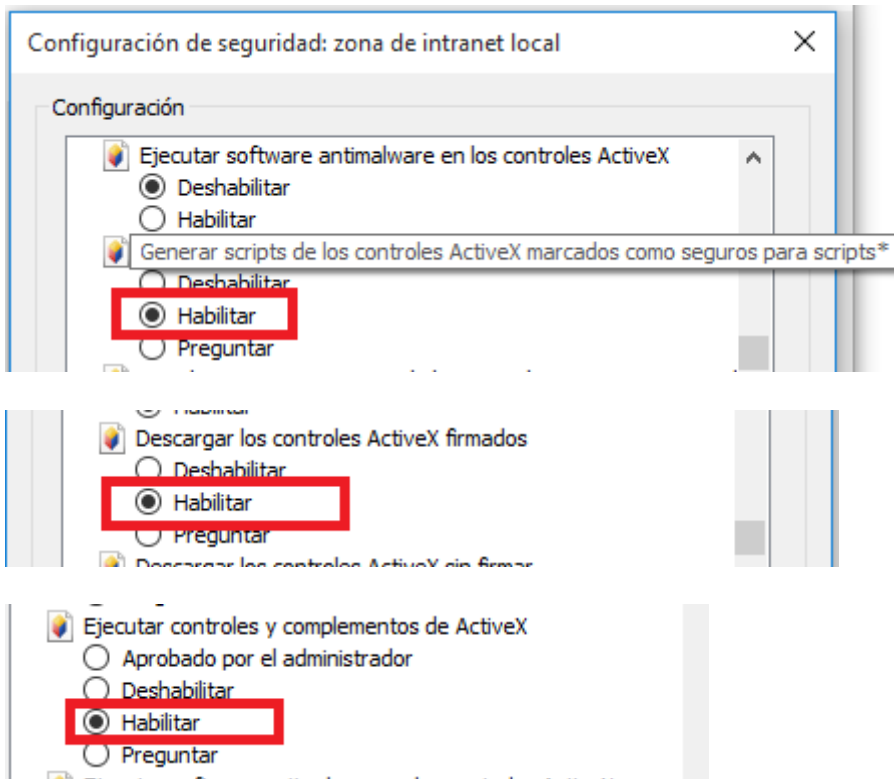
Seleccione la zona "Intranet local" ya que este proyecto va a usarse en la intranet de la fábrica.



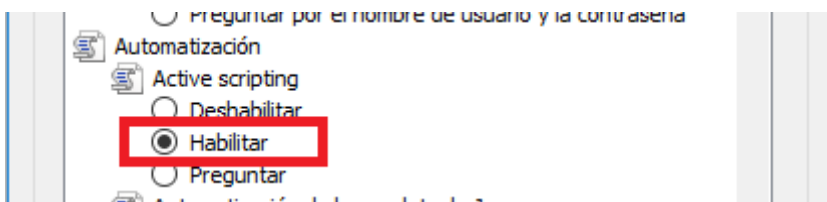
3. Allí haga clic en el botón de comando "Nivel personalizado".
4. Active las opciones "Generar scripts de los controles de ActiveX".



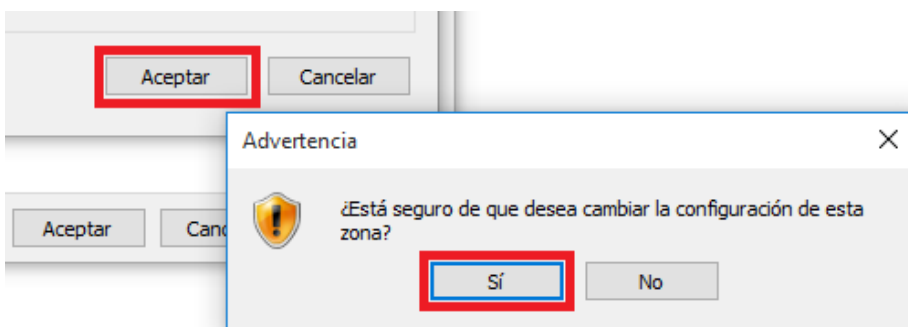
marcados como seguros para scripts”, “Descargar los controles ActiveX firmados” y “Ejecutar controles y complementos ActiveX”.



5. En "Scripting", active el "Active Scripting".

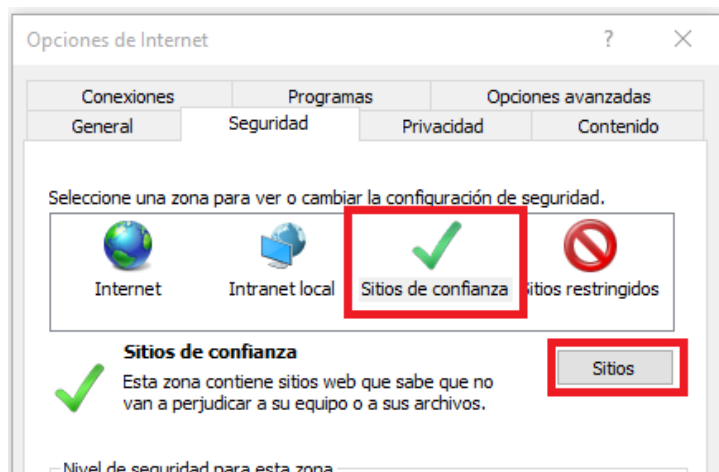


6. Haga clic en "Aceptar". En el siguiente diálogo confirme las modificaciones.

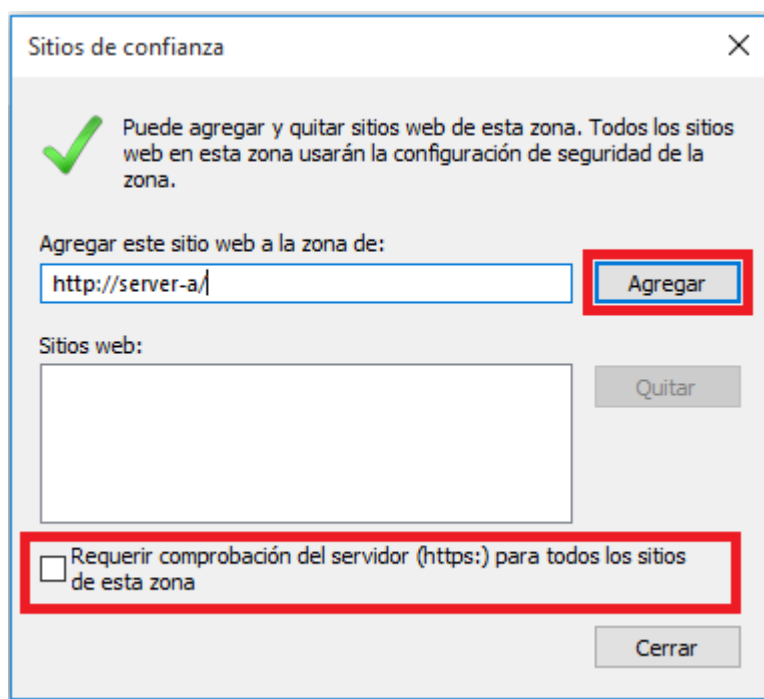


7. Haga clic sobre el icono "Sitios de confianza".

Mediante el botón "Sitios", abra el cuadro de diálogo "Sitios de confianza".



8. Introduzca en el campo "Agregar este sitio web a la zona" la dirección del servidor WebNavigator.



Desactive la opción "Requerir comprobación de servidor (https:) en todos los sitios de esta zona".

Haga clic en el botón "Agregar". Haga clic en "Aceptar".

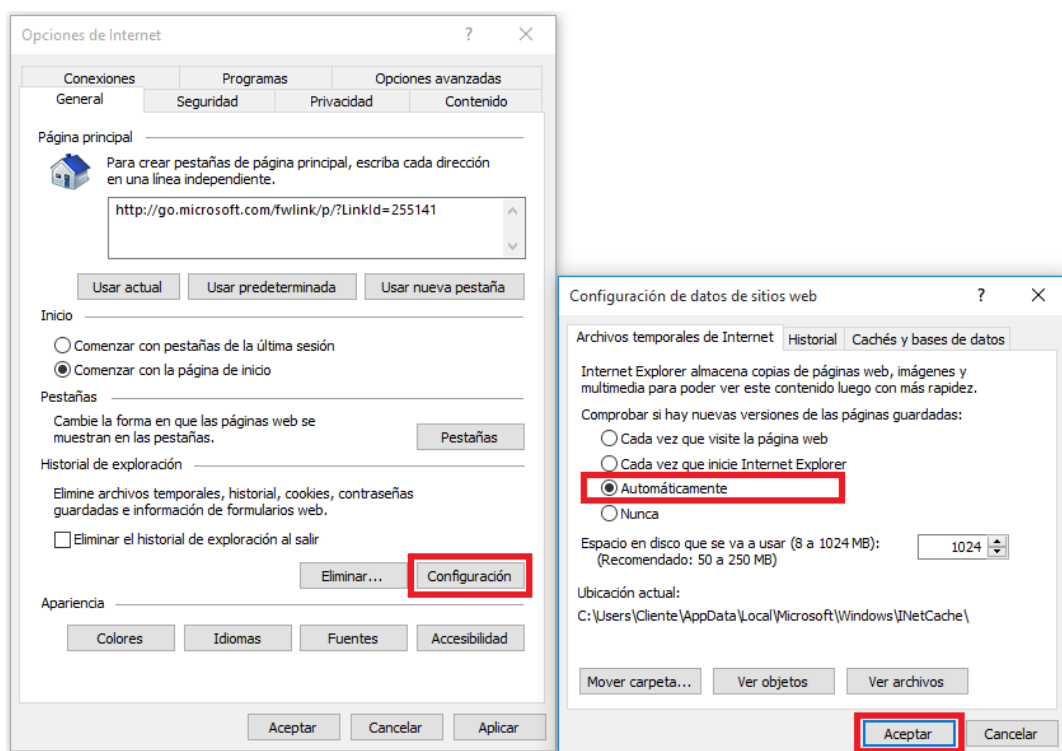
8. Haga clic sobre el icono "Sitios de confianza".

Haga clic en el botón de comando "Nivel estándar", y posteriormente en el botón "Nivel personalizado".

Active las opciones "Generar scripts de los controles de ActiveX marcados como seguros para scrips". Haga clic en "Aceptar".

9. Haga clic en la pestaña "General".

En el área "Historial de exploración" haga clic en el botón de comando "Configuración".



En "Comprobar si hay nuevas versiones de las páginas guardadas", active la opción "Automáticamente".

Haga clic en "Aceptar".

10. Cierre el cuadro de diálogo "Opciones de Internet" con "Aceptar".

A continuación si navegamos a la dirección <http://server-a> desde Internet Explorer pedirá instalar el siguiente software:

## SIMATIC WinCC/WebNavigator V7.4 + SP1

You first need to install the Web Navigator Client software

The installation will download approximately 62 MB to your computer.

[Click here to install WebNavigator Client](#)

Hacemos click en "Click here to install WebNavigator Client" y aparecerá el siguiente portal.



Se descargan e instalan los programas con permisos de administrador. Durante la instalación dejar todos los parámetros por defecto y pulsar Finalizar. Reiniciar el navegador.

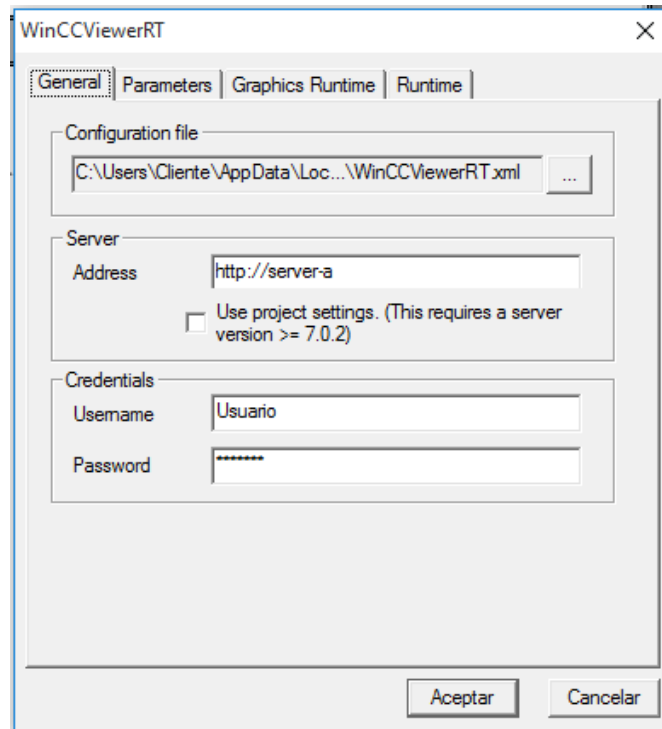
### Configuración de WinCCViewerRT:

1. En la carpeta de instalación, en "Webnavigator\Client\bin", haga clic en el la aplicación "WinCCViewerRT.exe".

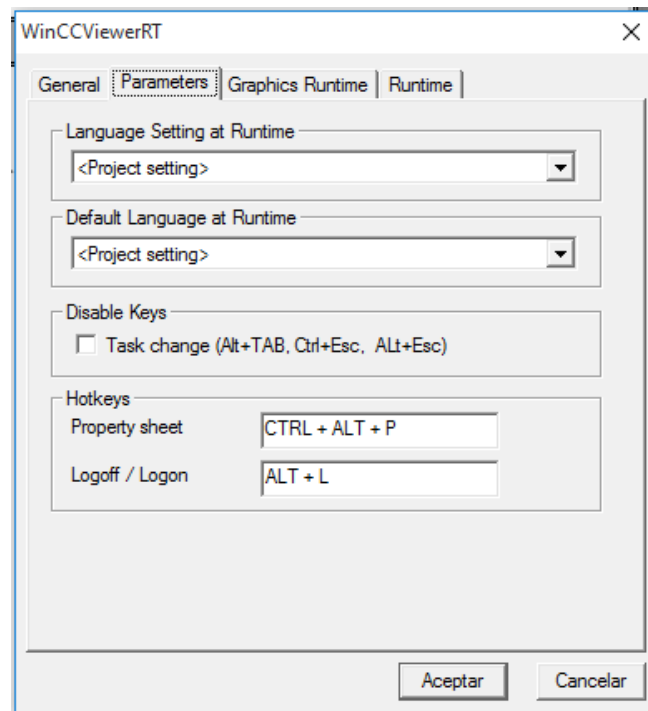
Al ejecutarla por primera vez se abre el diálogo de configuración.

Si el WinCCViewerRT ya se había configurado previamente, se abrirá el WinCCViewerRT

En tal caso, utilice la combinación de teclas <Ctrl+Alt+P> para abrir el cuadro de diálogo de configuración de WinCCViewerRT.

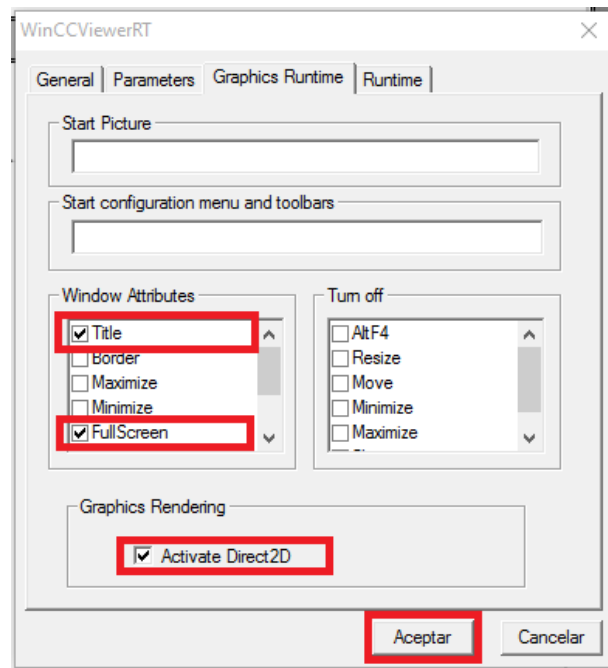


2. En la pestaña "General", introduzca los datos para el inicio de sesión:
  - Dirección del servidor: http://server-a
  - Nombre de usuario y contraseña, si desea predeterminar un usuario para el cuadro de diálogo de inicio de sesión.
  - Desactivar la opción "Use project settings"
  
3. En la pestaña "Parameters":



- Lenguaje Setting at Runtime: <Project setting>
- Default Lenguaje at Runtime: <Project setting>
- Se deshabilitan las combinaciones de teclas para cambiar de tarea.
- En Hotkeys se deja la configuración por defecto de la combinación de teclas <CTRL+ALT+P> para acceder al diálogo de configuración de WinCCViewerRT y se configura la combinación de teclas <ALT+L> para gestionar el login y logoff del usuario.

4. En la pestaña “Graphics Runtime” se configura:



En los campos de imagen inicial y configuración de menú se dejan vacíos para que cada usuario pille los que se le definan en la configuración del proyecto.

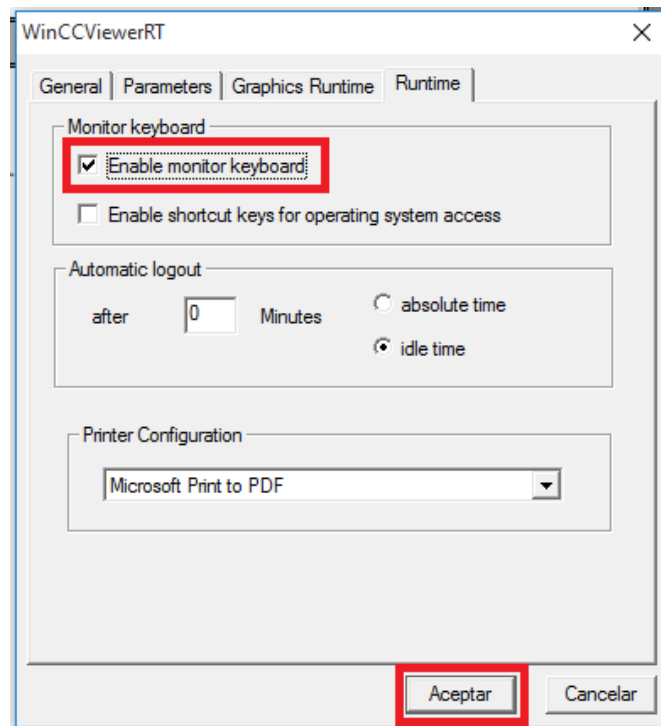
Se seleccionan los siguientes atributos de ventana en el área “Windows Attributes”:

- Title
- FullScreen

En “Graphics Rendering” se habilita “Activate Direct 2D”

5. En la pestaña “Runtime” se configura:

- Se selecciona habilitar teclado en pantalla.



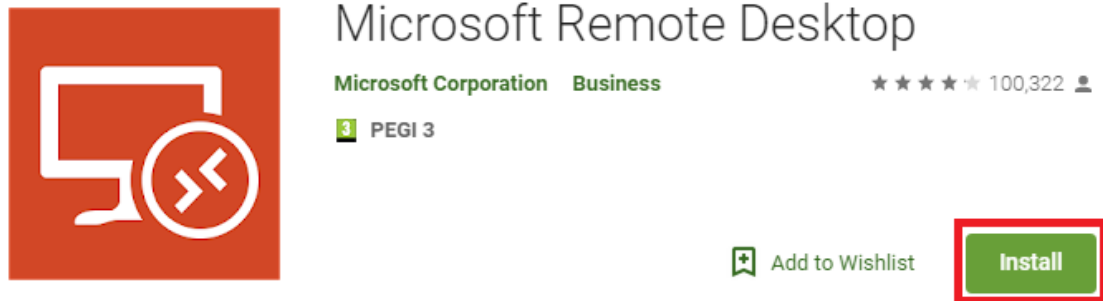
Finaliza la configuración haciendo click en Aceptar.



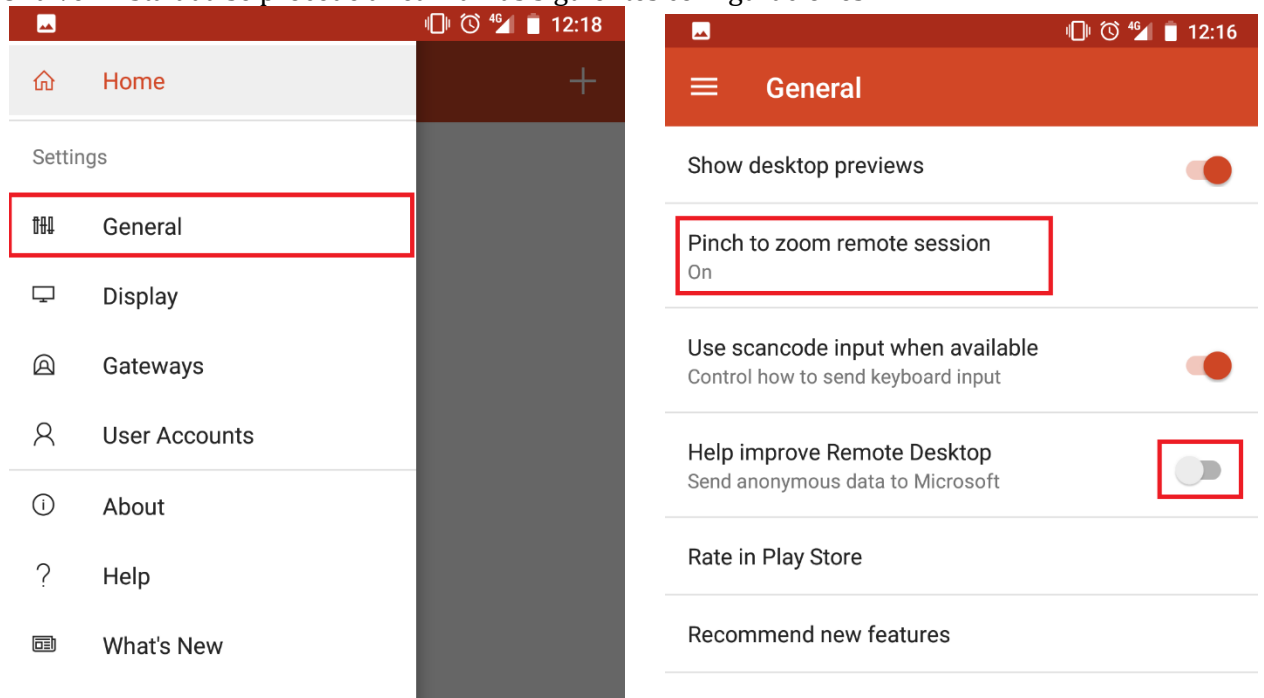
### 2.3.6. Configuración cliente RDP en dispositivos móviles y tabletas

Para conectarse al SCADA desde un dispositivo móvil se usa el cliente “Microsoft Remote Desktop” disponible para Android, iOS, y Windows Mobile. Puede descargarse desde la tienda correspondiente a cada uno de los sistemas operativos de los dispositivos portátiles.

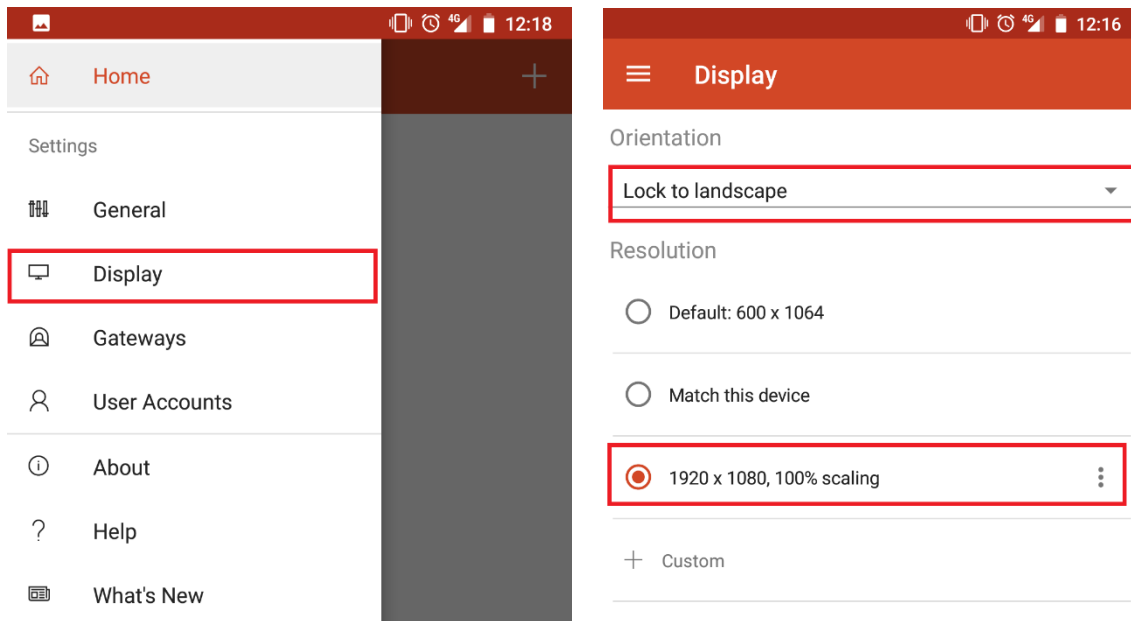
Buscamos la aplicación en la Play Store y se instala.



Una vez instalada se procede a realizar las siguientes configuraciones:



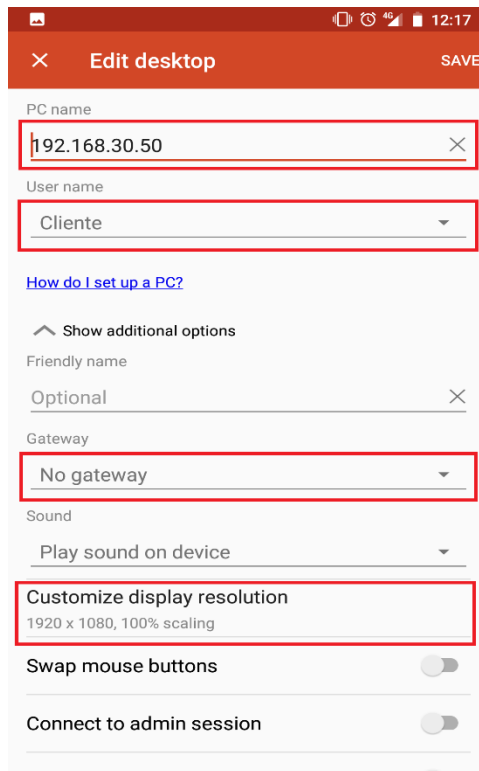
Desde los ajustes de la aplicación se selecciona General y se activa “Pinch to zoom remote session” si no está ya en On. Se desactiva el envío de datos a Microsoft.



Desde los ajustes de la aplicación se selecciona “Display” y en Orientación seleccionamos “Lock to landscape” para usarlo siempre en modo horizontal.

En “Resolución” añadimos la resolución “1920x1080” haciendo click en “Custom” para visualizar el SCADA con la resolución para la que se ha diseñado. Una vez definida dicha resolución debemos seleccionarla.

Añadir nueva conexión en Microsoft Remote Desktop:



En “PC name” se configura la dirección IP de la máquina remota a la que queremos conectarnos. Configuramos el Usuario y Contraseña con los que nos conectaremos a la máquina.

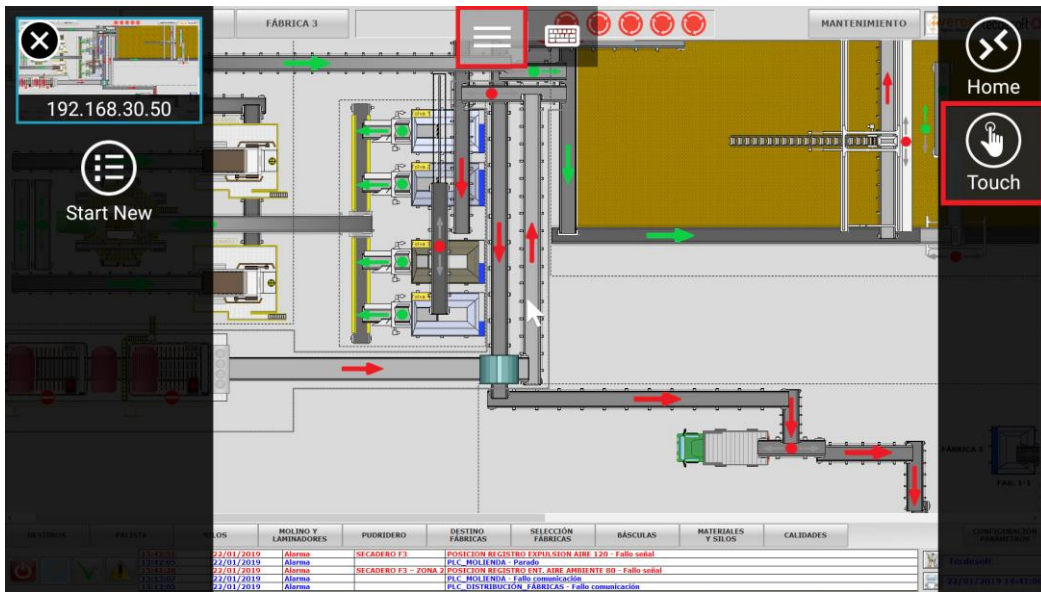
En la puerta de enlace se selecciona “No gateway”.

Por último nos aseguramos de que la resolución seleccionada es: 1920x1080p

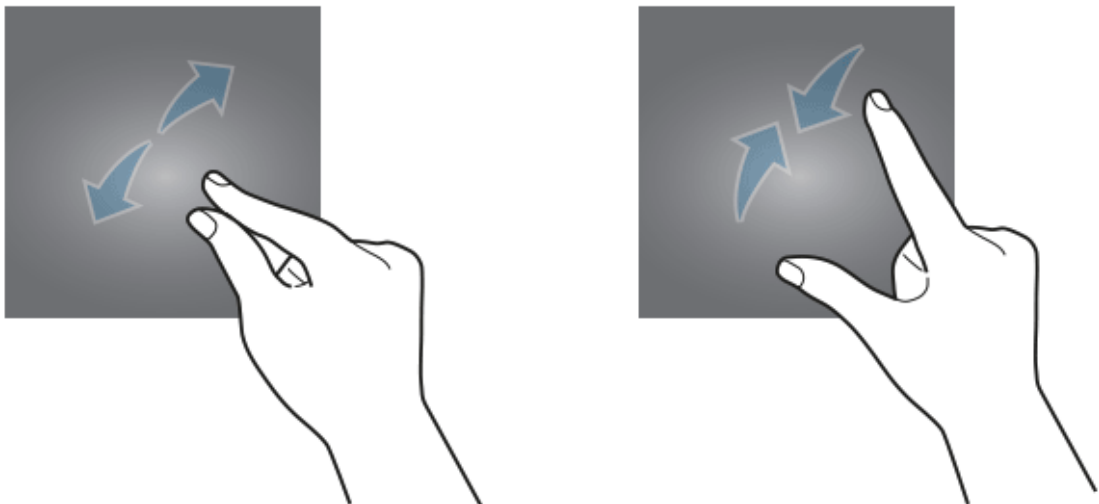
Manejo del SCADA a través del Cliente Remoto desde el dispositivo móvil.

Una vez que estamos conectados y visualizamos el SCADA en remoto podemos movernos a través de los siguientes gestos:

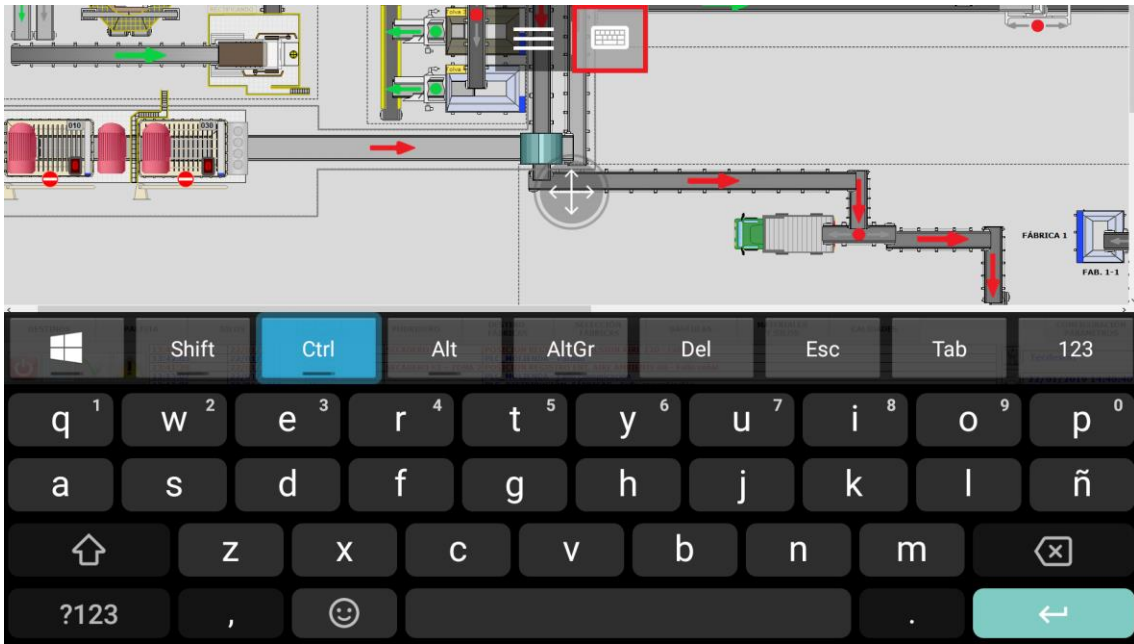
Antes de nada debe estar seleccionada la opción “Touch”.



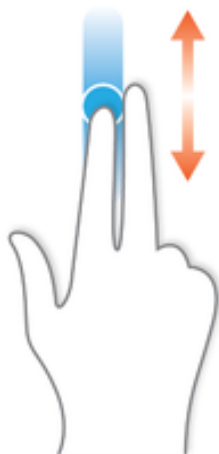
-Para hacer Zoom (Amplificar y reducir) en todo el escritorio remoto:



-Hacer Zoom solo en la ventana de imagen de los sinópticos generales para que se muestren u oculten las capas:



Con la tecla CTRL del teclado activada podemos hacer "Scroll" con dos dedos como si la rueda del ratón se tratase.



- Desplazar la ventana de imagen:

Con la tecla de CTRL activada se puede desplazar la imagen moviéndola con un solo dedo.



### 2.3.7. Configuración de los clientes ligeros de los puestos de operación

Para conectarse al SCADA desde los clientes ligeros instalados en los puestos de operación de la planta han de seguirse los siguientes pasos que se especifican a continuación: Primero debemos encender el cliente AXEL AX90 pulsando el botón central.

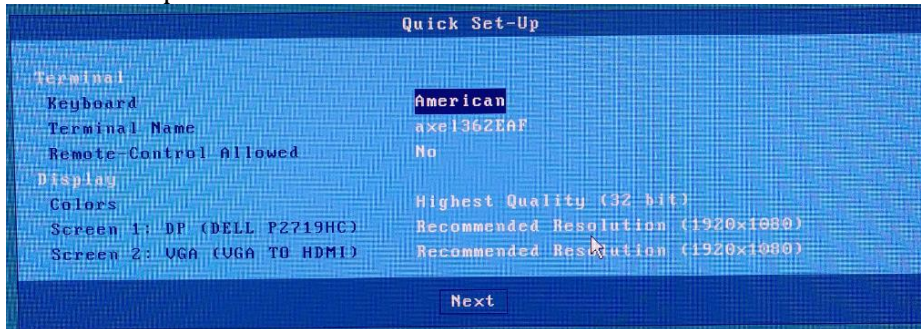


- Mantener pulsado para apagar
- Pulsar para encender

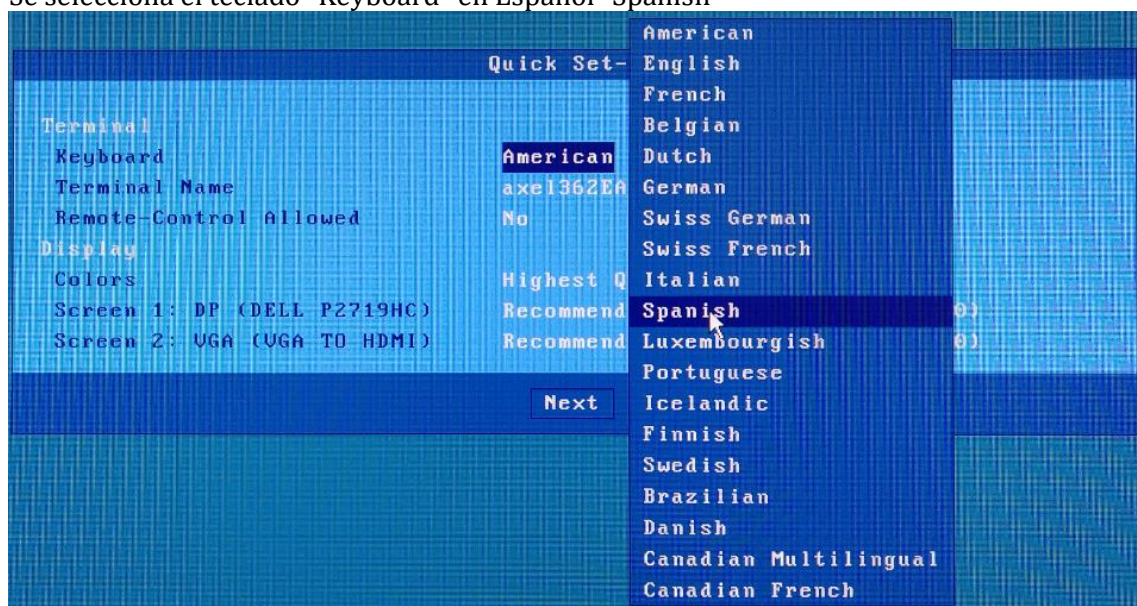
La configuración de la conexión por escritorio remoto con la máquina virtual puede realizarse empleando el Quick Set-Up y desde el menú de configuración.

#### 2.3.7.1. Configuración con el Quick Set-Up

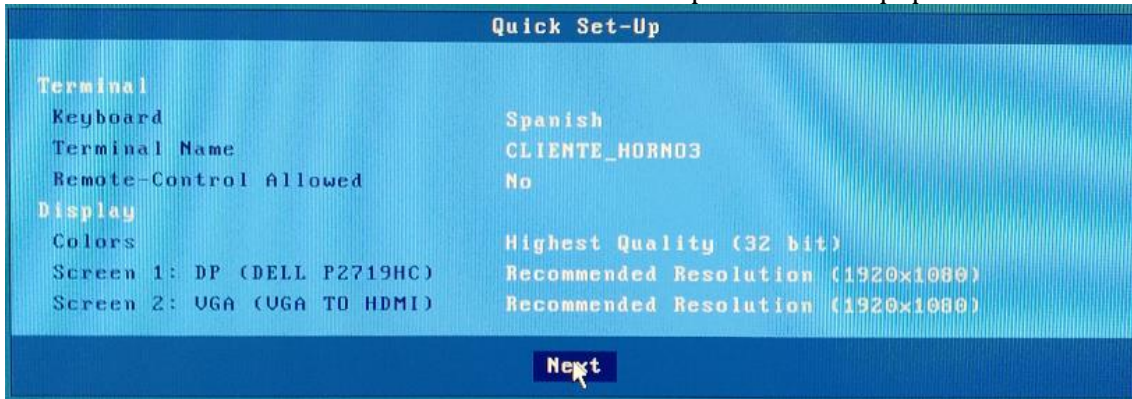
La primera vez que se enciende el AXEL AX90 aparece el Quick Set-Up que nos guiará en la configuración del dispositivo.



Se selecciona el teclado "Keyboard" en Español "Spanish"



En Terminal Name introducimos el nombre del cliente para tener el equipo identificado.

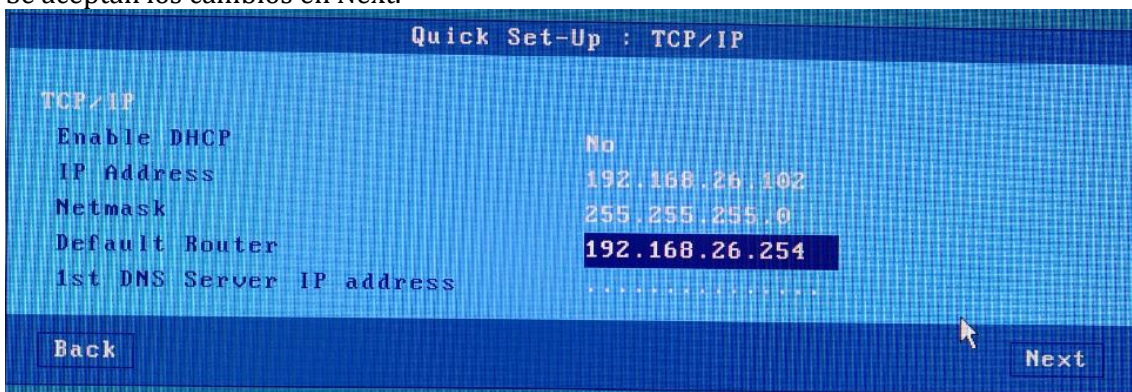


Por defecto si están las pantallas conectadas les asigna una resolución recomendada de 1920x1080p, si las pantallas no están conectadas se puede configurar la resolución manualmente.

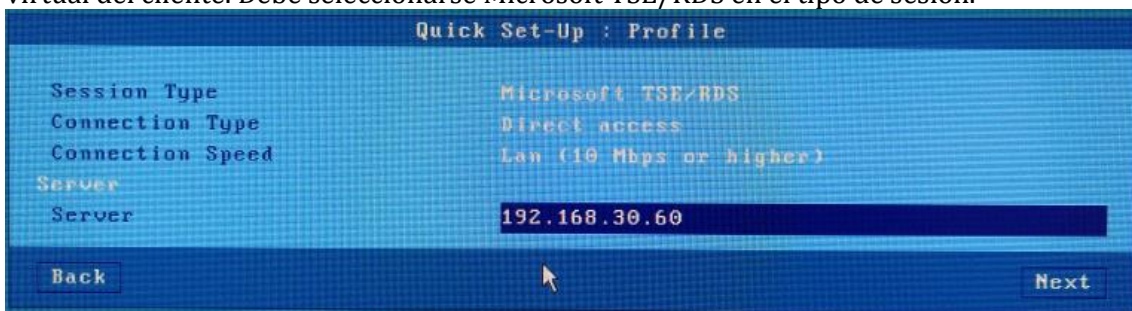
Se continúa con la configuración rápida aceptando en Next.

Se establece la configuración de la interfaz de red asignándole una dirección IP estática al cliente ligero en el campo IP Address, se configura una máscara de subred 255.255.255.0 y se le asigna la puerta de enlace 192.168.26.254.

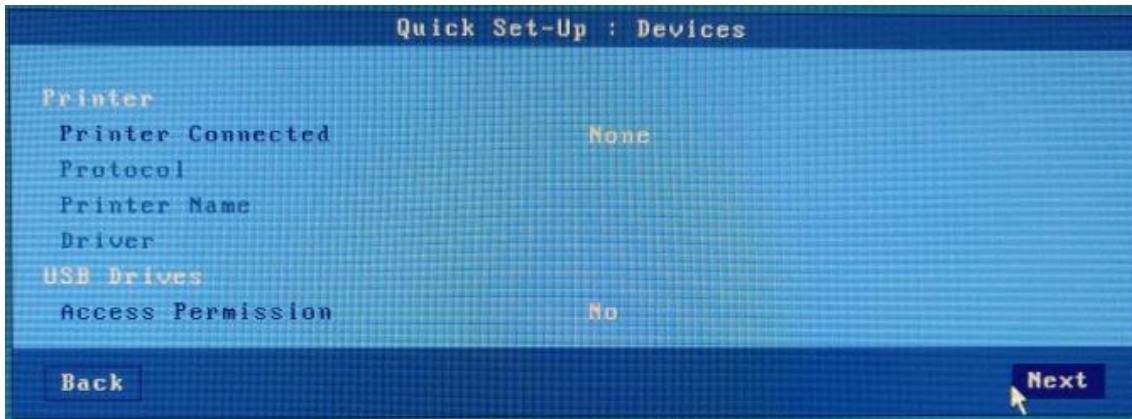
Se aceptan los cambios en Next.



A continuación se configura la sesión de escritorio remoto para conectarse a la máquina virtual del cliente. Debe seleccionarse Microsoft TSE/RDS en el tipo de sesión.

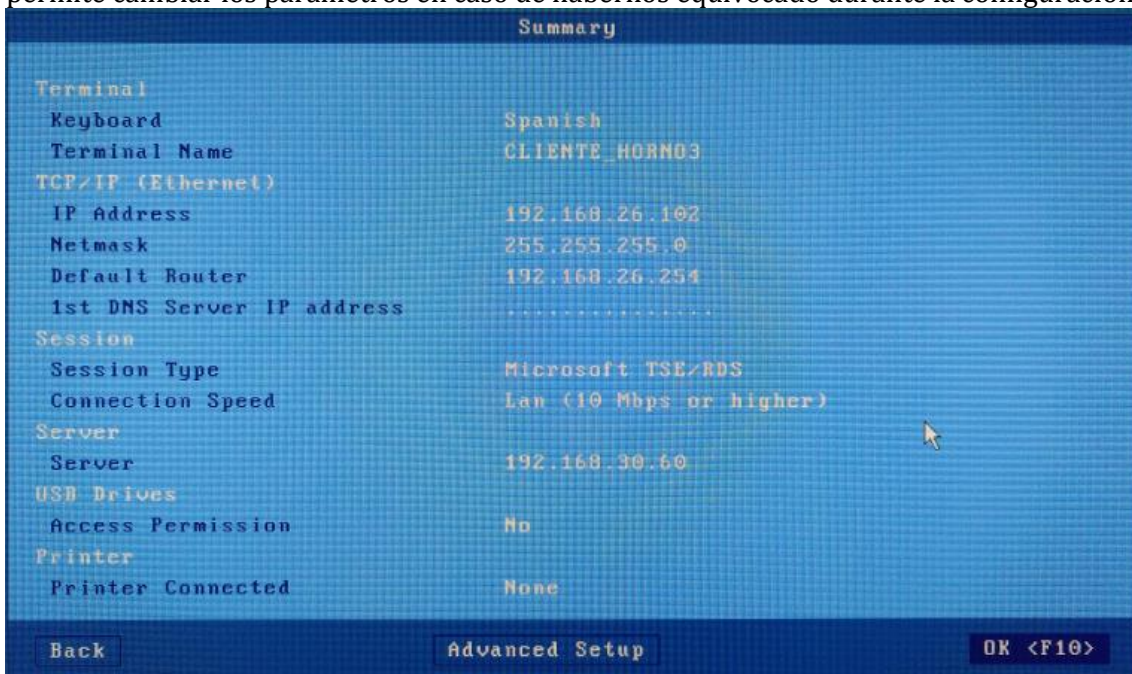


Se aceptan los cambios en Next.



No se permite conexión de impresoras en este equipo, y por motivos de seguridad tampoco se permite dar acceso a los puertos USB.  
Se aceptan los cambios en Next.

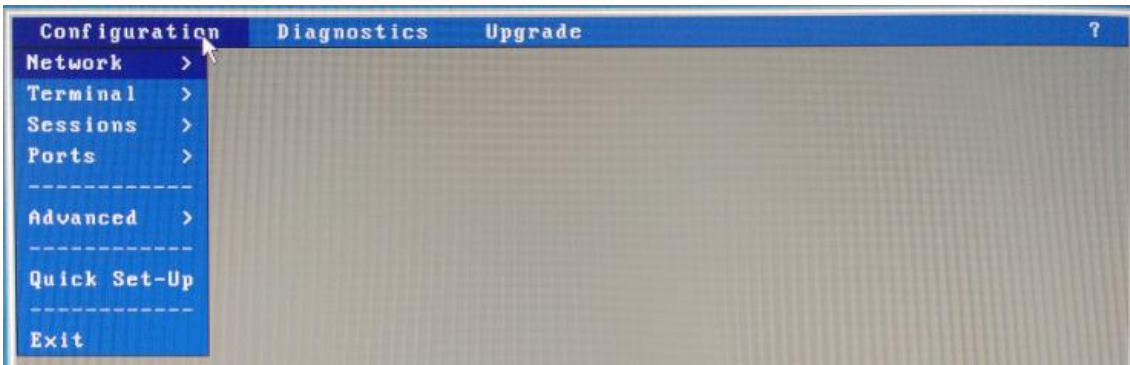
Por último aparece un menú con el resumen de la configuración realizada desde el que se permite cambiar los parámetros en caso de habernos equivocado durante la configuración.



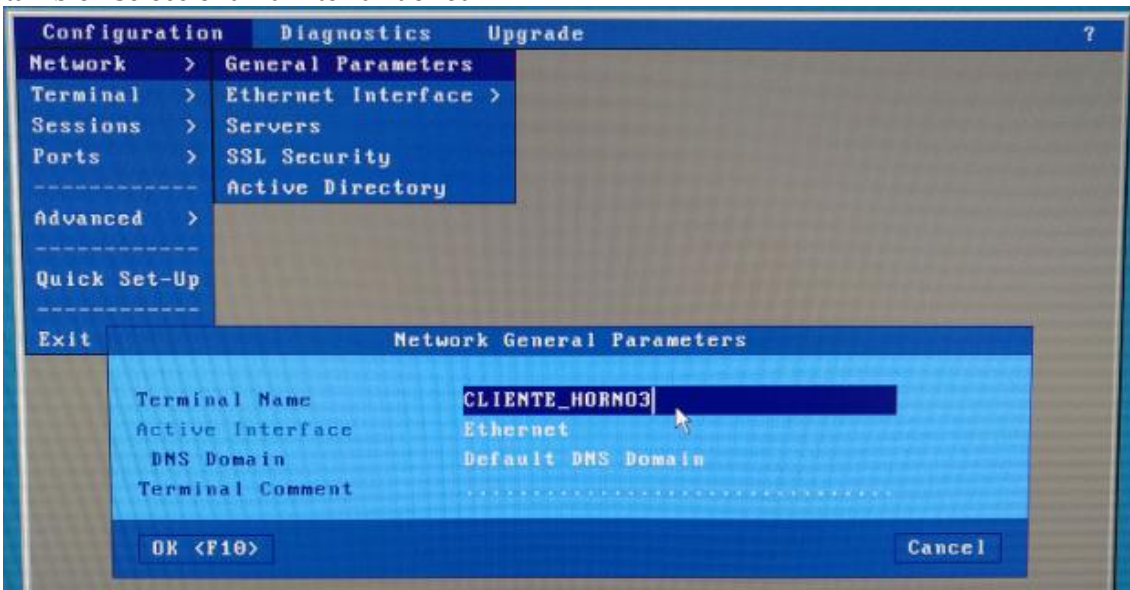
Se aceptan los cambios en Next o pulsando la tecla F10.  
El equipo se reinicia y automáticamente abrirá la sesión de escritorio remoto

### 2.3.7.2. Configuración desde el menú

Si se desea configurar el dispositivo sin emplear el Quick Set-Up o simplemente desea modificarse algún parámetro de configuración del mismo puede realizarse desde el menú de configuración al que se podrá acceder en cualquier momento con la sesión de escritorio remoto cerrada pulsando la siguiente combinación de teclas: **Ctrl+Alt+Esc**



En Network -> General Parameters podemos cambiar el nombre del equipo así como también seleccionar la interfaz de red.

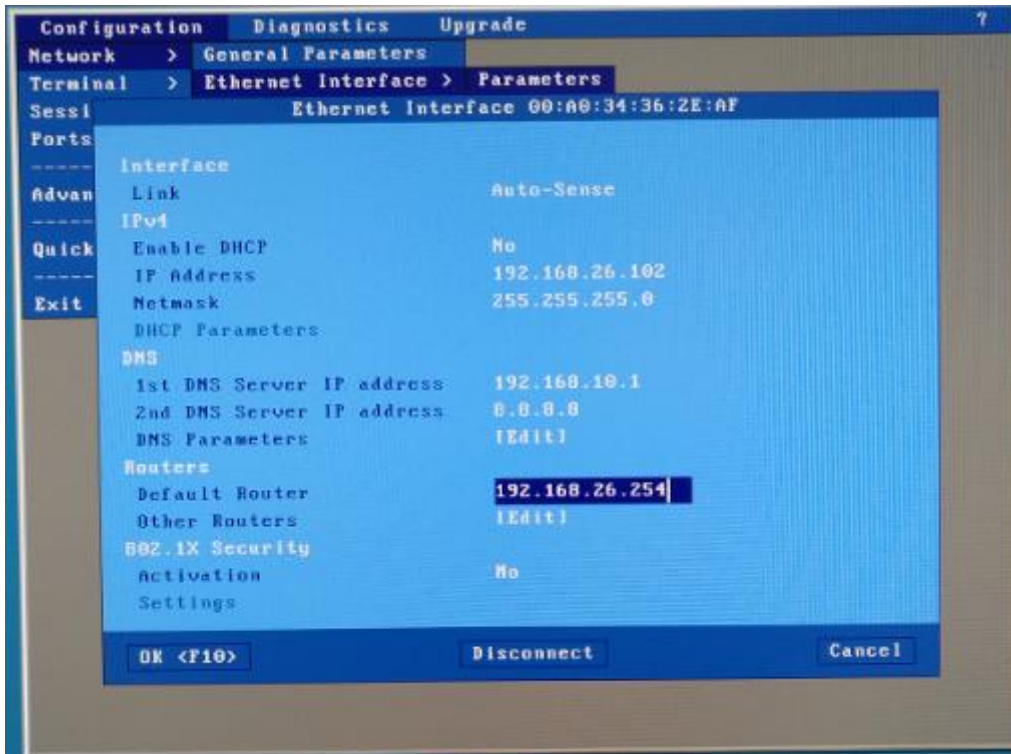


Se aceptan los cambios pulsando en el botón OK o la tecla F10 del teclado físico.

A continuación se configura la interfaz de red en Network -> Ethernet Interface -> Parameters.

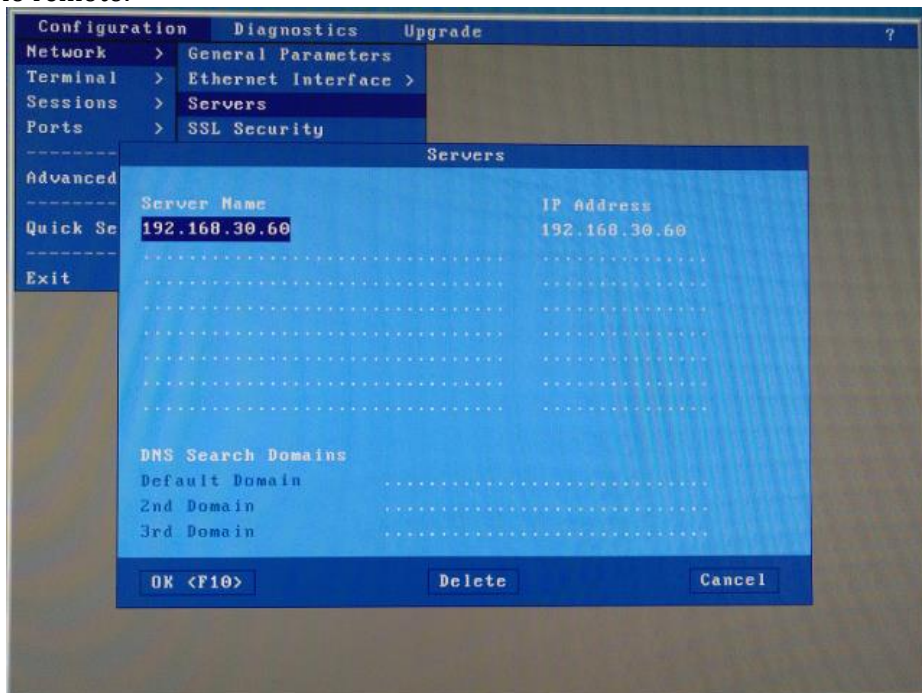
Se cubren los campos de la dirección IP Ipv4, máscara de subred y puerta de enlace tal y como se muestra en la imagen.





Se aceptan los cambios pulsando en el botón OK o la tecla F10.

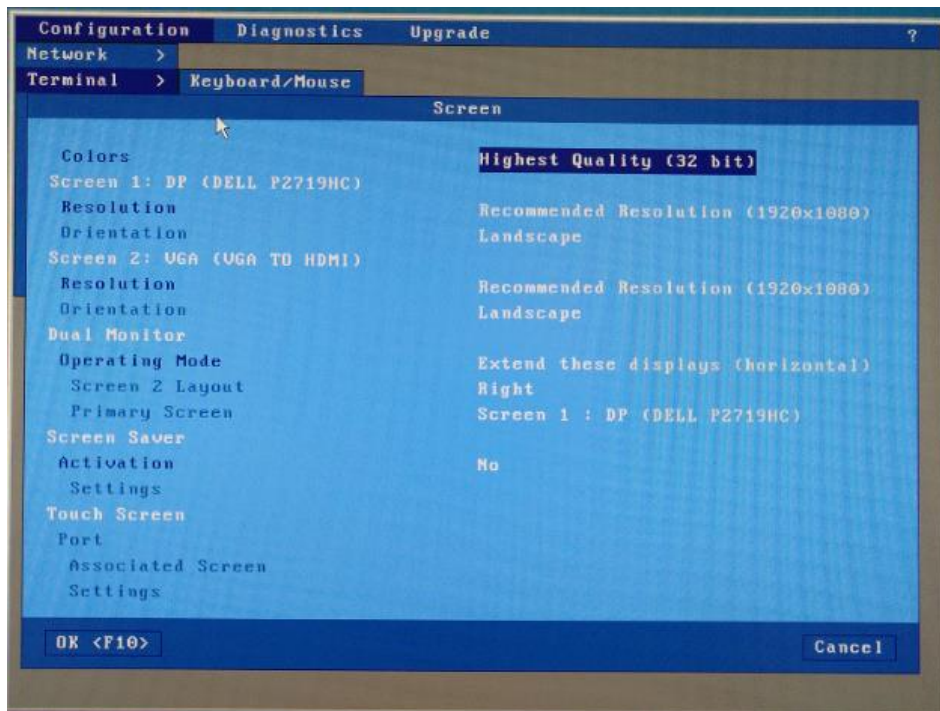
En Network -> Servers se configura la dirección IP del servidor al que se desea conectar por escritorio remoto.



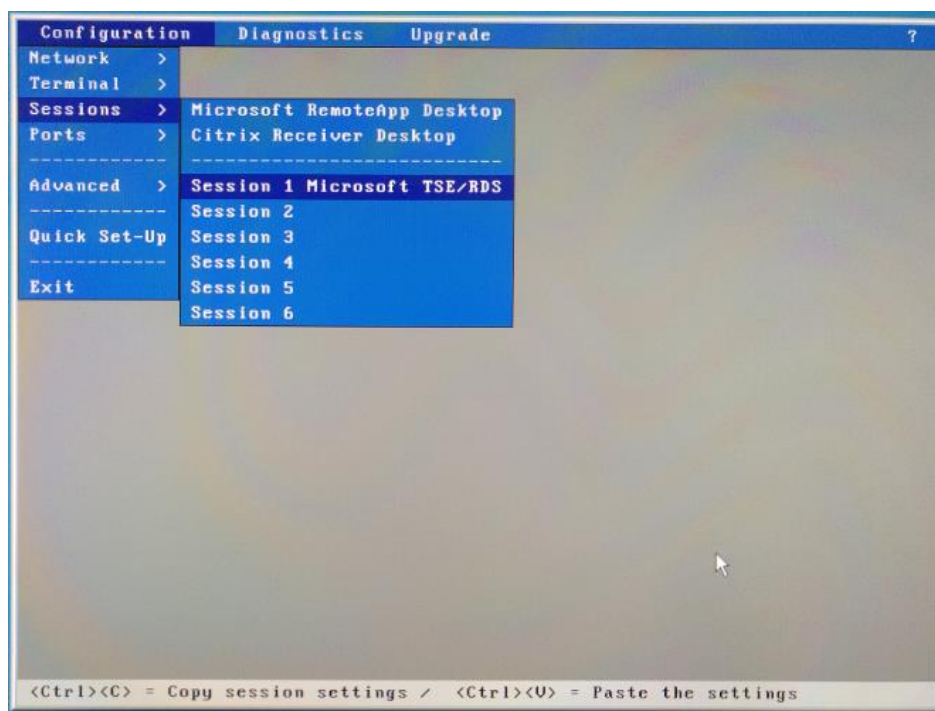
Se aceptan los cambios pulsando en el botón OK o la tecla F10.

En Terminal -> Screen se configuran las resoluciones de ambos monitores y se selecciona el modo de operación con dos monitores si es el caso.

La resolución de los monitores debe ser 1920x1080 y en caso de tener dos monitores ha de seleccionarse el modo de operación extendido.

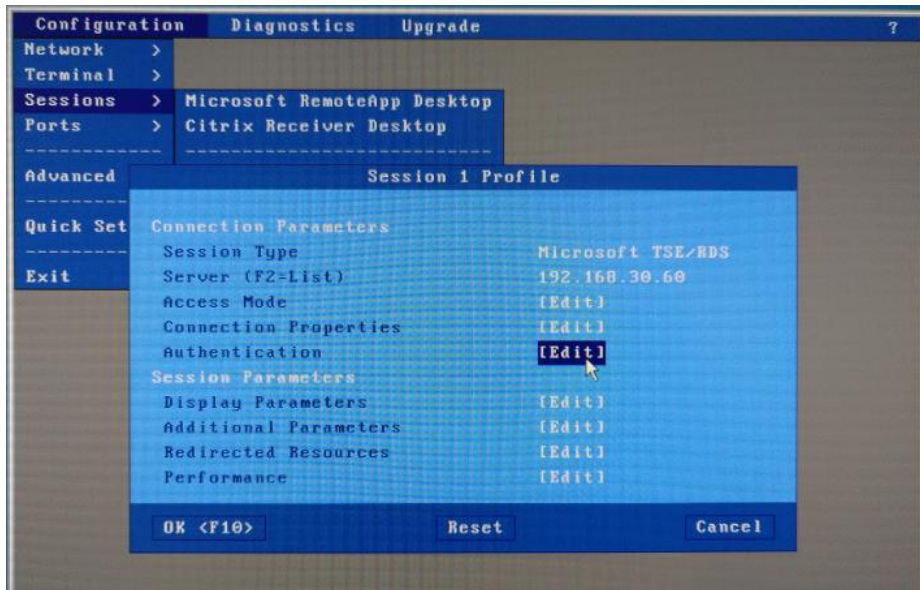


Se aceptan los cambios pulsando en el botón OK o la tecla F10.



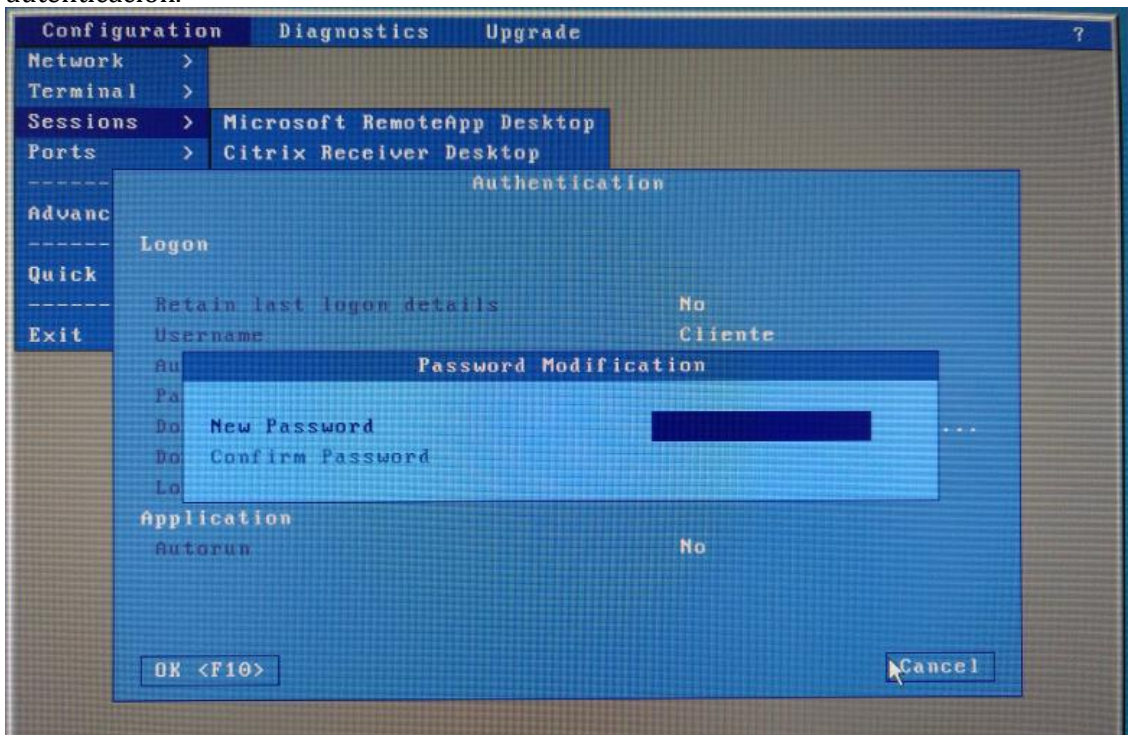
A continuación se configura la sesión de escritorio remoto seleccionando en el tipo de sesión Microsoft TSE/RDS.

En Server se selecciona de la lista el servidor definido anteriormente en Network->Servers



Si se desea que el equipo se conecte al servidor por escritorio remoto y se autentique automáticamente se debe editar la autenticación de la sesión seleccionada.

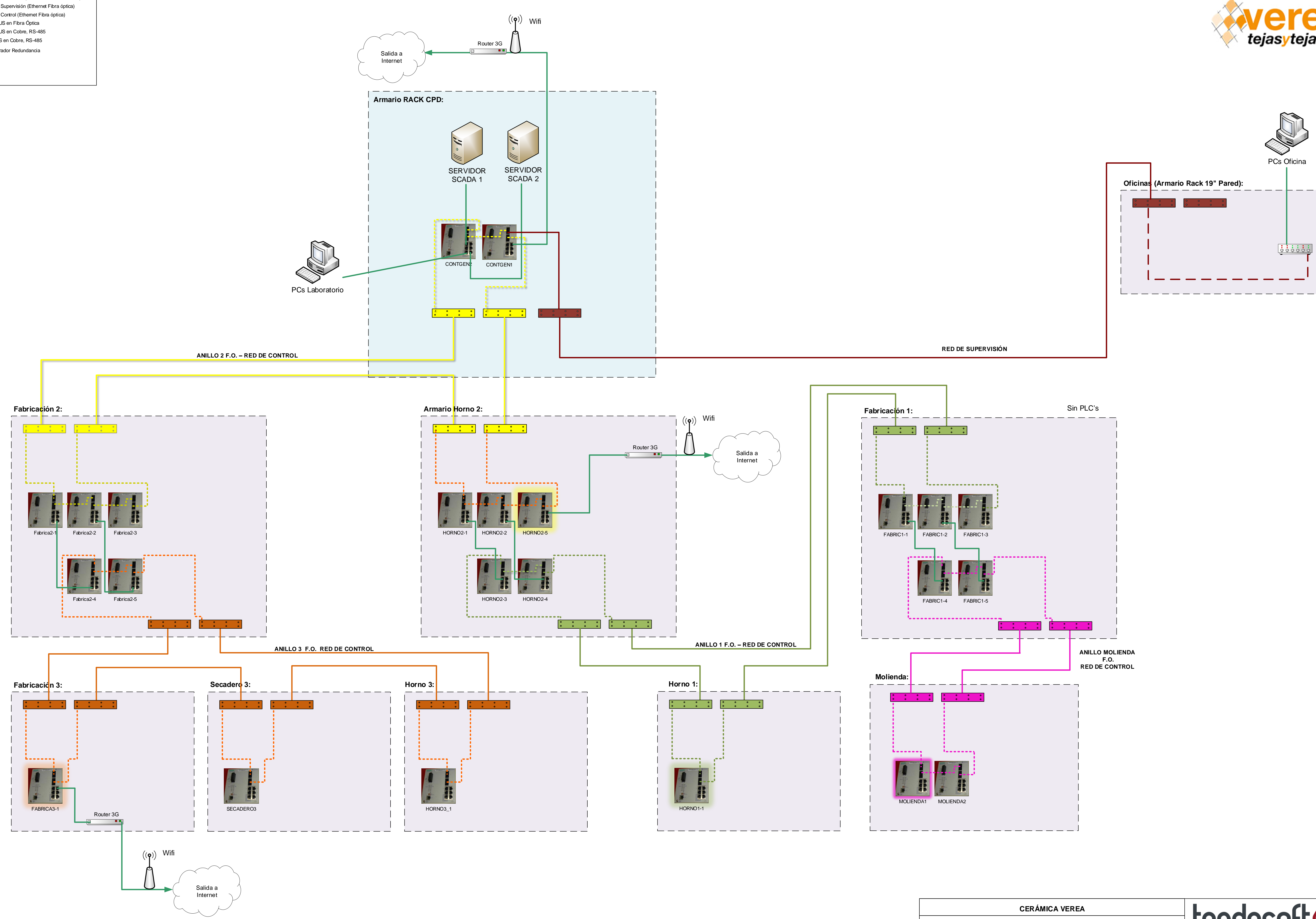
Se debe seleccionar "No" en conservar los últimos detalles de inicio de sesión y a continuación introducir el nombre de usuario y contraseña con la que se realizará autenticación.



Por último se aceptan los cambios pulsando en el botón OK o la tecla F10 y se reinicia el dispositivo.

### 3. PLANOS

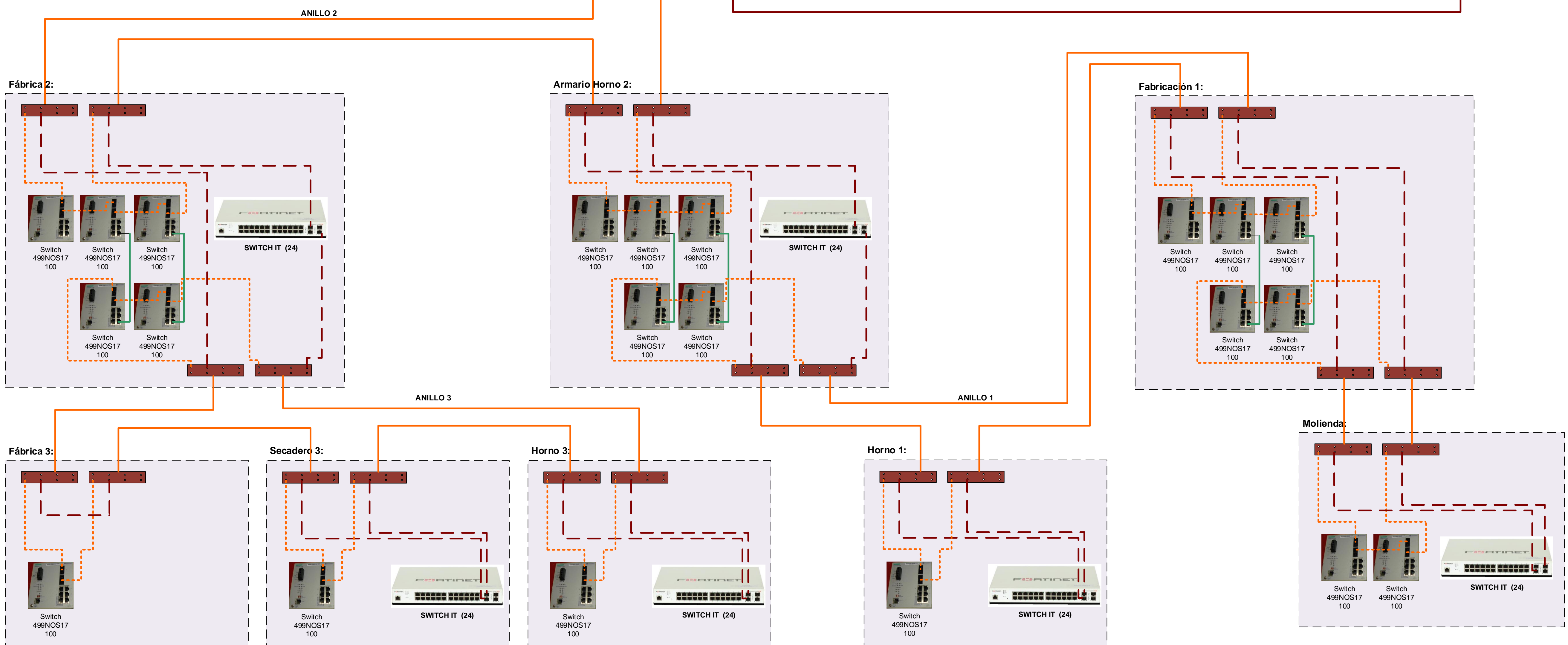
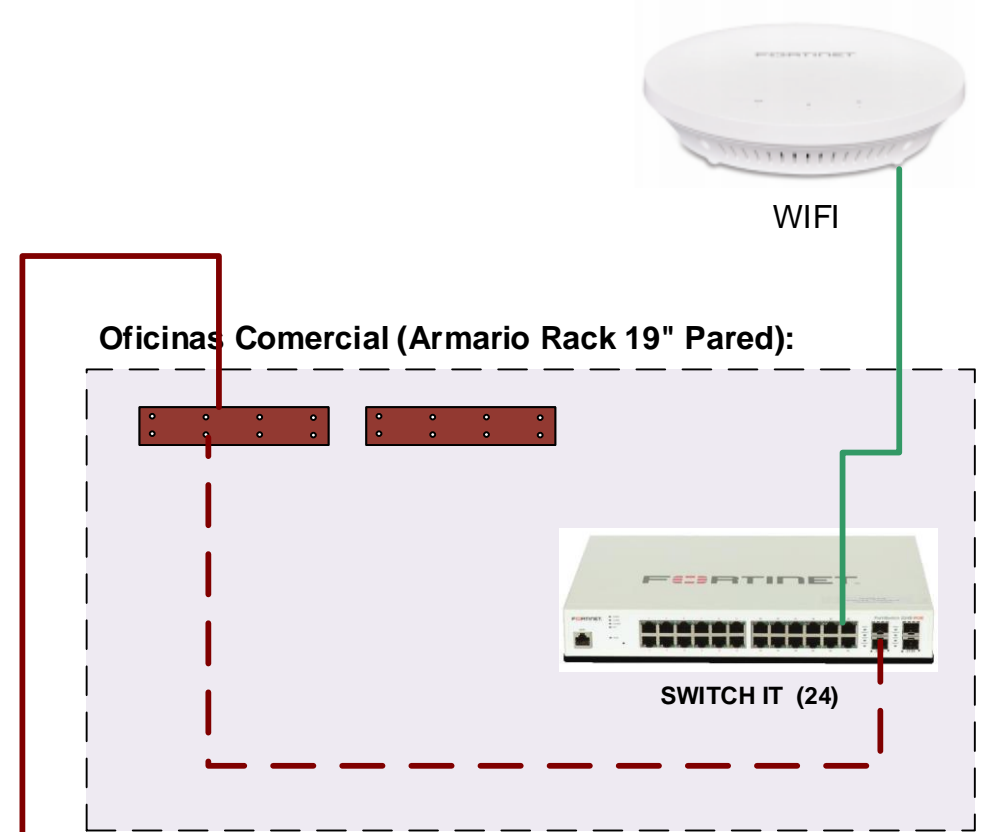
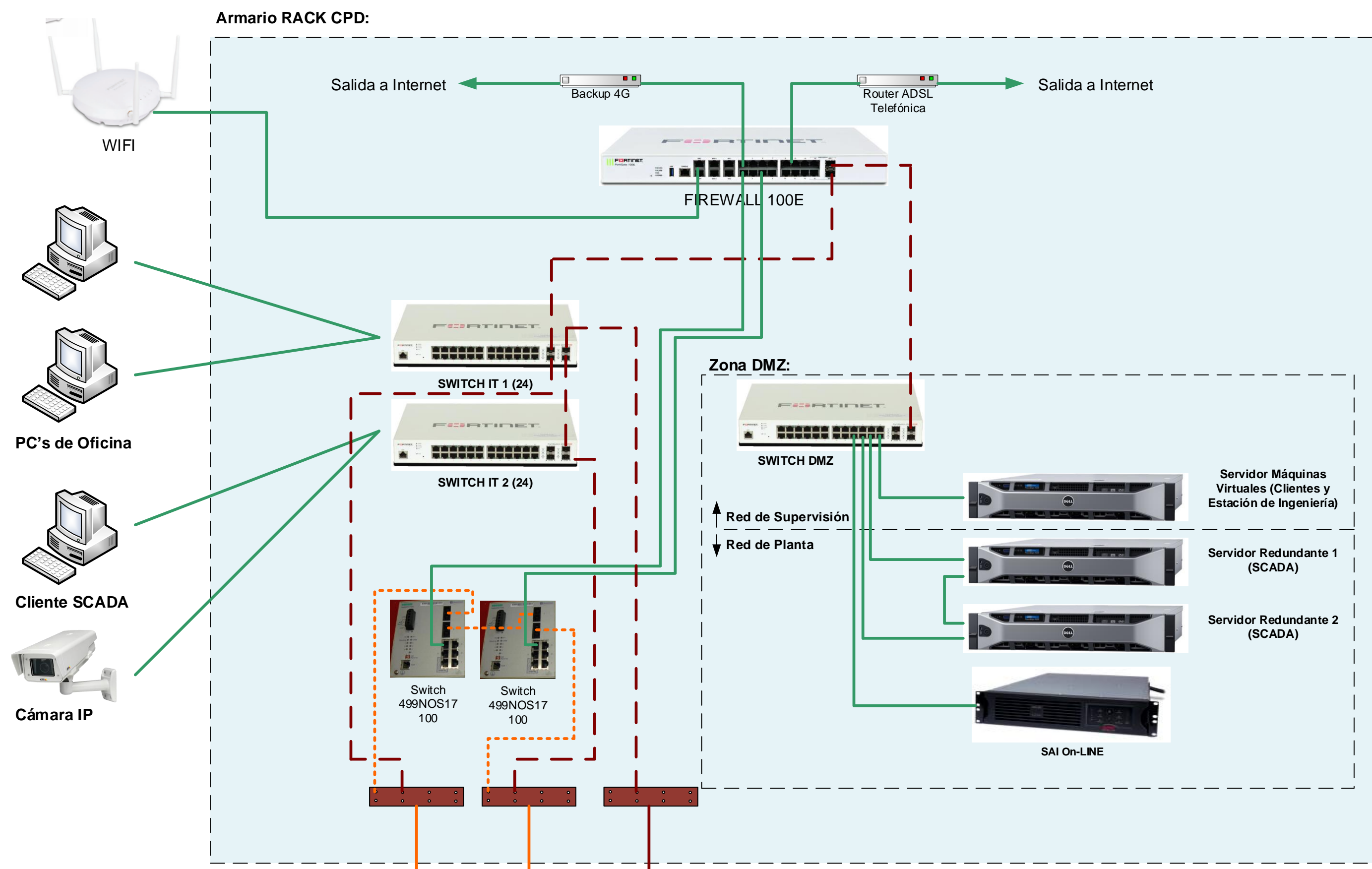
- Legenda:**
- Red Ethernet Cobre (Cable norma PROFINET)
  - Anillo de Supervisión (Ethernet Fibra óptica)
  - Anillo de Control (Ethernet Fibra óptica)
  - PROFIBUS en Fibra Óptica
  - PROFIBUS en Cobre, RS-485
  - MODBUS en Cobre, RS-485
  - RMX Administrador Redundancia



CERÁMICA VEREA			
PLANO DE RED INICIAL			
Hoja 1 de 1	Rev A	21/03/2018	- Sistema WinCC

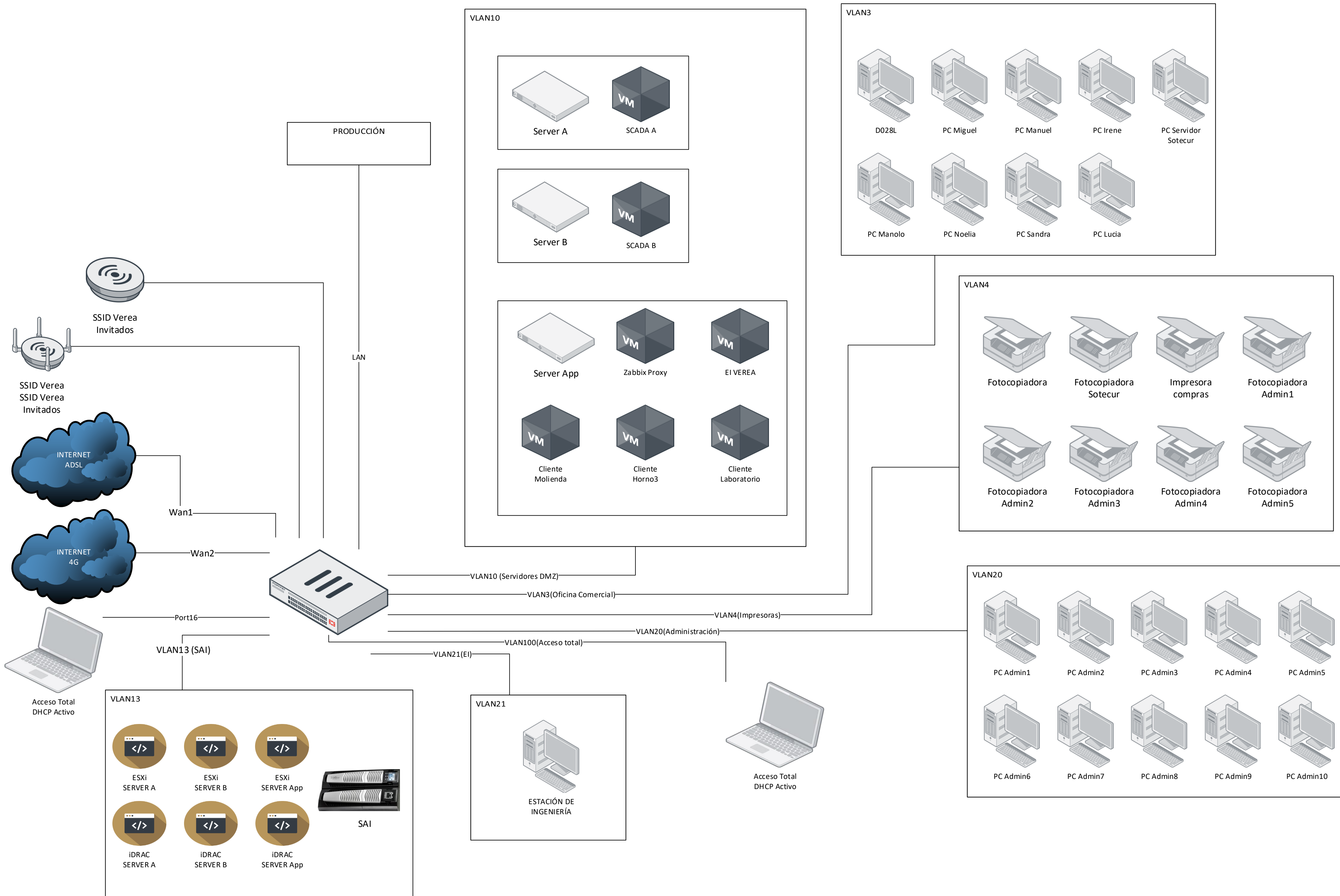
\*Este plano es propiedad de Técnicas de Soft S.A. Queda prohibida su reproducción total o parcial sin previa autorización.

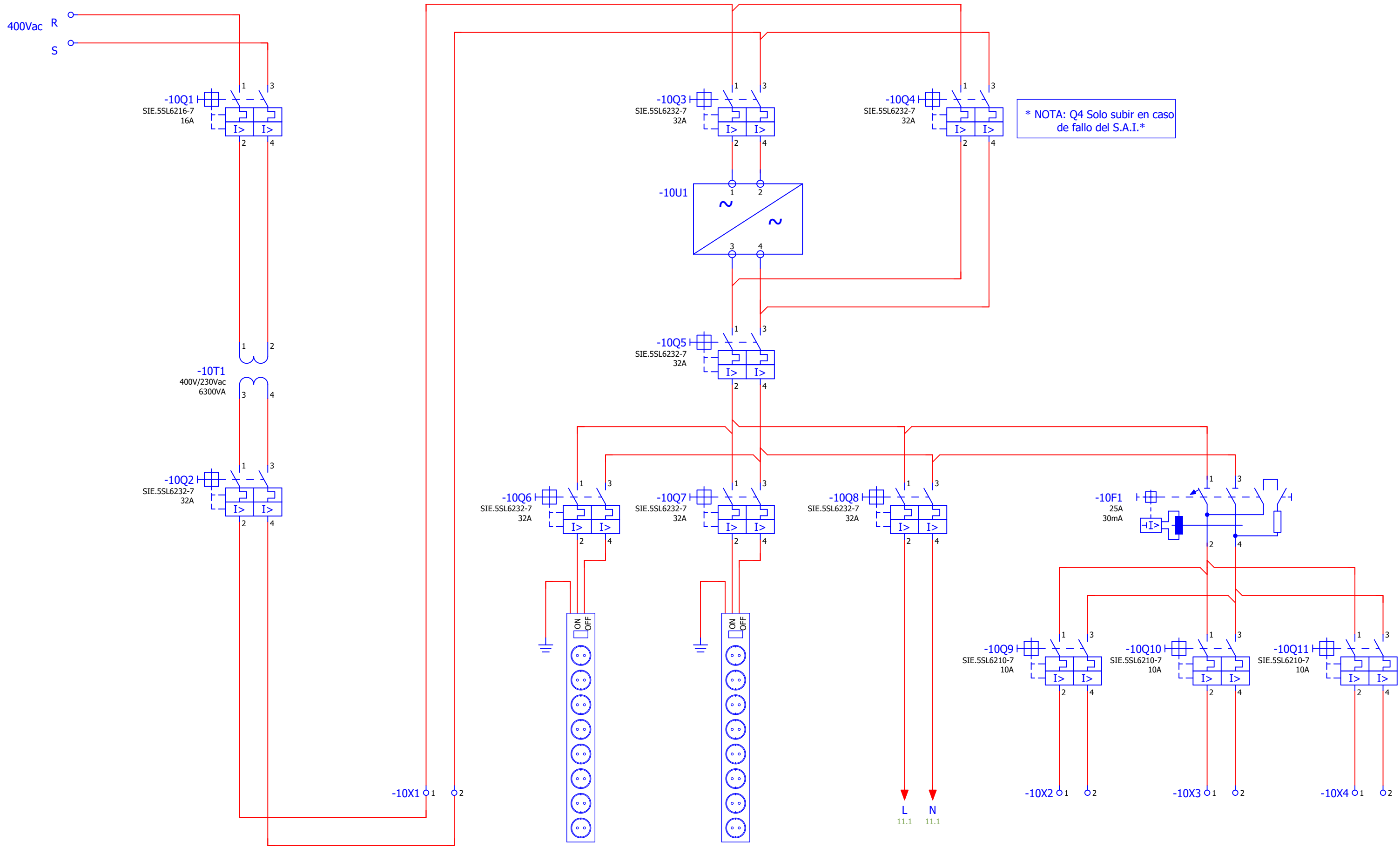
- Legenda:**
- Red Ethernet Cobre (Cable norma PROFINET)
  - Anillo de Supervisión (Ethernet Fibra óptica)
  - Anillo de Control (Ethernet Fibra óptica)
  - PROFIBUS en Fibra Óptica
  - PROFIBUS en Cobre, RS-485
  - MODBUS en Cobre, RS-485



<b>CERÁMICA VEREA</b>			
<b>TOPOLOGÍA SISTEMA DE AUTOMATIZACIÓN</b>			
Hoja 1 de 1	Rev B	07/03/2018	- Sistema WinCC

\*Este plano es propiedad de Técnicas de Soft S.A. Queda prohibida su reproducción total o parcial sin previa autorización.





\* NOTA: Q4 Solo subir en caso de fallo del S.A.I.\*

TOMA DE REGLETAS Nº1

TOMA DE REGLETAS Nº2

A FUENTES DE ALIMENTACIÓN 24Vdc

OFICINA

TELEFONÍA

RESERVA

0	07/18	E.MURIAS	EMISIÓN	J.J.P.E.	J.J.P.E.
REV.	FECHA	DIBUJADO	DESCRIPCIÓN	VERIFIC.	APROB.

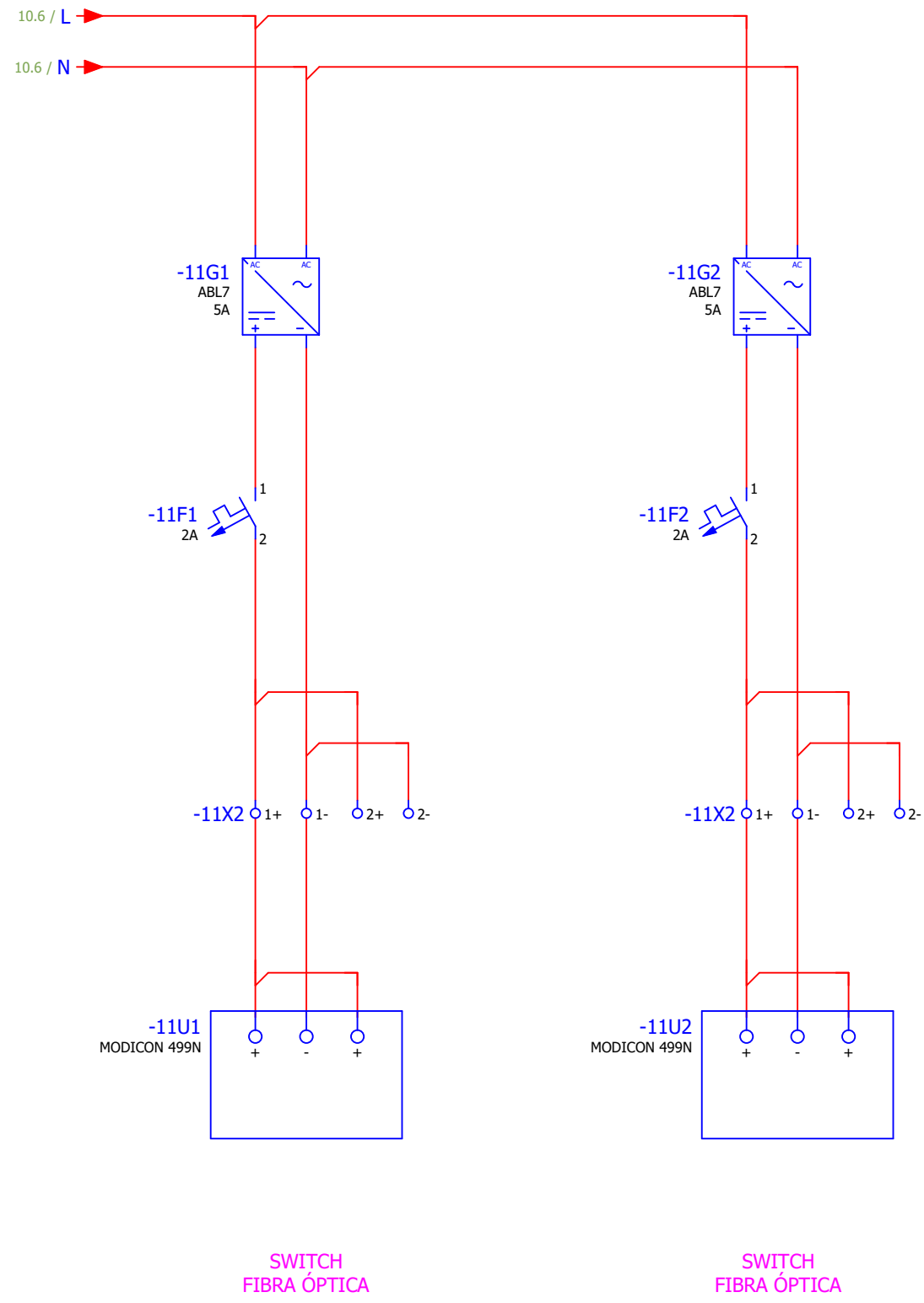




Instalación:	VEREA
Sistema:	MODERNIZACIÓN RED DE PLANTA
Equipo:	ARMARIO RACK

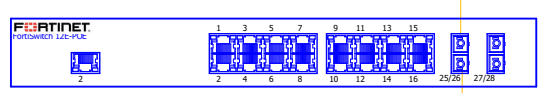
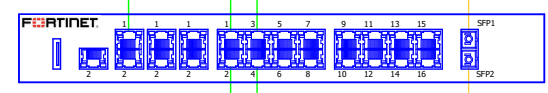
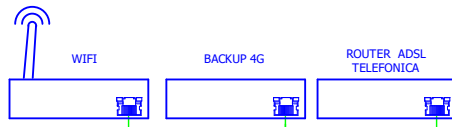
ALIMENTACIONES	
----------------	--

Colección	
Hoja 1	De 3
Autor: Enrique Murias Fernández	
Fecha creación: 03/07/2018	

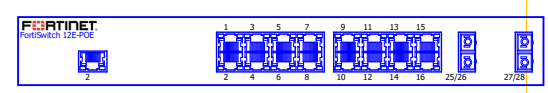
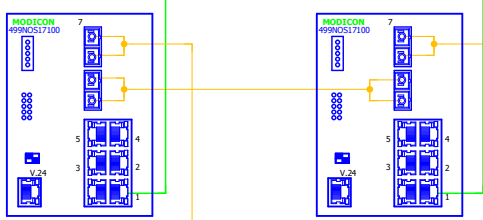




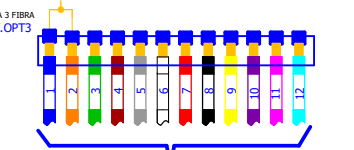
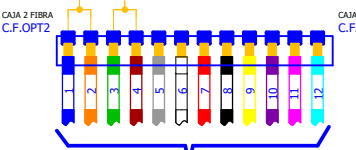
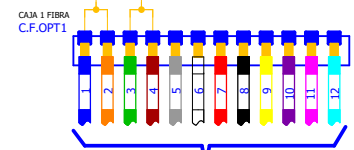
						 	Instalación: VEEA	ALIMENTACIONES	Colección
							Sistema: MODERNIZACIÓN RED DE PLANTA		Hoja 2 De 3
0	07/18	E.MURIAS	EMISIÓN	J.J.P.E.	J.J.P.E.		Equipo: ARMARIO RACK		Autor: Enrique Murias Fernández
REV.	FECHA	DIBUJADO	DESCRIPCIÓN	VERIFIC.	APROB.			Fecha creación: 03/07/2018	



PC's DE OFICINA



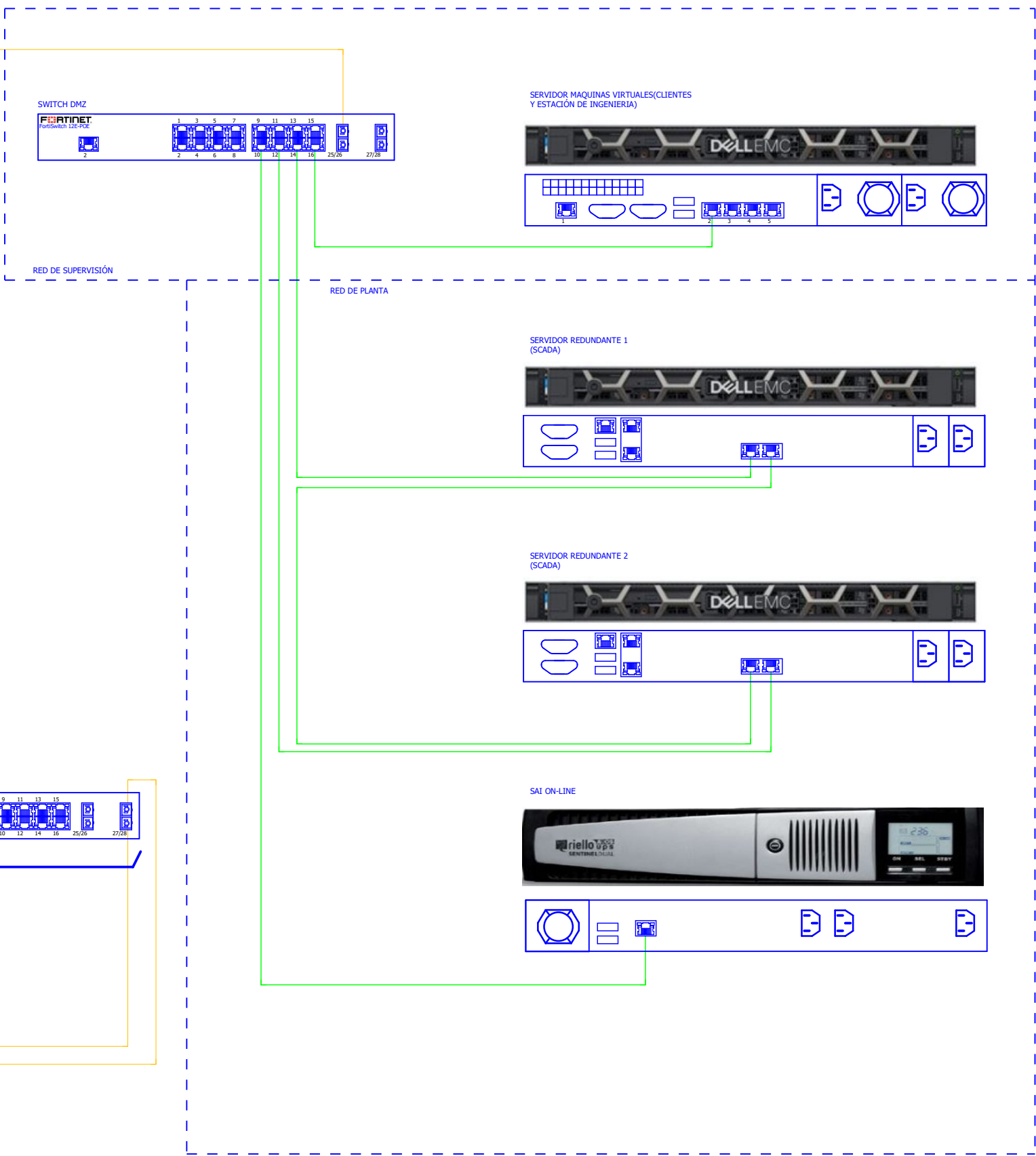
CLIENTE ESCADA



MANGUERA DE FIBRA OPTICA MULTIMODO 12 PARES DESDE FABRICA 2

MANGUERA DE FIBRA OPTICA MULTIMODO 12 PARES ARMARIO HORNO

MANGUERA DE FIBRA OPTICA MULTIMODO 12 PARES DESDE OFICINA COMERCIAL



0	07/18	E.MURIAS	EMISIÓN	J.J.P.E.	J.J.P.E.
REV.	FECHA	DIBUJADO	DESCRIPCIÓN	VERIFIC.	APROB.



Instalación:	VEREA
Sistema:	MODERNIZACIÓN RED DE PLANTA
Equipo:	ARMARIO RACK

COMUNICACIONES

Colección	
Hoja	3 De 3
Autor:	Enrique Murias Fernández
Fecha creación:	03/07/2018

#### 4. PRESUPUESTO

CATEGORÍA	DESCRIPCIÓN	PRECIO UNITARIO	CANTIDAD	TOTAL
LICENCIAS	ACT KEY/WINDOWS PRO 10 32BIT/64 YR ONLN	177,86 €	3	533,58 €
	SQL SERVER STANDARD EDITION 2017 LICENSE	875,78 €	1	875,78 €
	WINDOWS SERVER 2016 STANDARDED,ROK,16CORE	531,14 €	2	1.062,28 €
	WINDOWS SERVER 2016 STANDARD	298,55 €	1	298,55 €
	WINCC/ARCHIVE V7.4, 1500 ARCHIVE TAGS, OPCION P. WINCC V7.4, SW RUNTIME, LIC.INDIVIDUAL, CLAVE DE LIC. EN MEMORIA USB	1.739,72 €	2	3.479,45 €
	PAQUETE DE 10 LICENCIAS CLIENTE WINCC/WEB NAVIGATOR V7.4	4.073,48 €	1	4.073,48 €
	LICENCIA SIMATIC WINCC V7.4 RT 65536 (65536 POWER TAGS)	3.916,63 €	2	7.833,26 €
	LICENCIA SIMATIC WINCC REDUNDANCY V7.4	2.564,51 €	1	2.564,51 €
	PAQUETE MIGRACION FL WEB NAVIGATOR	2.333,75 €	1	2.333,75 €
	DELL-FK-DXD9H - 520 CADDY FOR 2.5" HD	15,75 €	1	15,75 €
				<b>23.070,38 €</b>
EQUIPOS	ADAPTADOR VGA A HDMI CON AUDIO	18,88 €	2	37,76 €
	DELL RATON OPTICO MS116 NEGRO	8,14 €	3	24,41 €
	DELL TECLADO MULTIMEDIA NEGRO	13,39 €	3	40,16 €
	DELL P2719HC DELL MONITOR 27" NEGRO	243,81 €	5	1.219,05 €
	CLIENTE LIGERO AXEL AX90	348,60 €	3	1.045,80 €
	SERVICE-PACK AXEL AX90 5-Y CAMBIO ADELANT	42,00 €	3	126,00 €
	SERVIDOR DELL POWEREDGE R440	2.811,60 €	2	5.623,19 €
	SERVIDOR DELL POWEREDGE R640	4.397,64 €	1	4.397,64 €
	BATERIAS SAI RIELLO AUTONOMIA 30MINUTOS 70%	2.046,24 €	2	4.092,48 €
				<b>16.606,50 €</b>
EQUIPOS DE RED	FIREWALL FORTIGATE 100E	3.135,50 €	1	3.135,50 €
	SWITCH FORTISWITCH 124E (24BOCAS)	324,42 €	11	3.568,62 €
	PUNTO DE ACCESO WIFI FORTIAP 221E	286,65 €	1	286,65 €
	PUNTO DE ACCESO WIFI FORTIAP 223E (CON ANTENAS EXTERNAS)	318,50 €	1	318,50 €
				<b>7.309,27 €</b>
RACK	GUIA PASACABLE 1U CON TAPA	7,44 €	13	96,78 €
	RACK 9U 600X550X501 1F 1V NEGRO	350,00 €	1	350,00 €
	BAND FO 19" 24SC SX24 LC DX	32,33 €	1	32,33 €
	TARJETA RED RIELLO	136,34 €	1	136,34 €
	PUERTAS	572,88 €	1	572,88 €

	BANDEJA EXTRAIBLE RACK	25,16 €	1	25,16 €
	CABLE ALIMENTACION IEC M-H 1,8	2,94 €	1	2,94 €
	ELECTRONIC LOCK HORIZONTAL 1-4	29,40 €	2	58,80 €
	REGLETA 19" 8 SCHUKOS= EDCARS8	14,18 €	2	28,35 €
	TERMOSTATO ANALOGICO PARA RACKV2	18,04 €	1	18,04 €
	TAPA PARA TERMOSTATO	1,34 €	1	1,34 €
	VENTILACION TECHO RACK	15,12 €	1	15,12 €
	PATCH PANEL CAT6 STP	14,47 €	4	57,88 €
	GUIA PASACABLE 1U CON TAPA	7,44 €	9	67,00 €
	LM ASCPS3 PASAMUROS SIMPLEX OM3 SC	0,62 €	24	14,87 €
	KIT DE GUIAS UNIVERSL PARA RACK	58,80 €	2	117,60 €
	OBTURADOR MULTI 9 LONGITUD 1m	11,37 €	1	11,37 €
	TAPA CIEGA 1U NEGRA	2,63 €	6	15,75 €
	TAPA CIEGA 2U NEGRA	4,64 €	1	4,64 €
				<b>1.627,19 €</b>
<b>MATERIAL ELÉCTRICO Y OTROS</b>	CONECTOR RJ45 HUTP CAT6 SL AMP	3,26 €	50	162,75 €
	BANDEJA FIJA 19" 4 PUNTOS F1000	18,01 €	1	18,01 €
	CLAVIJA CAUCHO NEGRA IP44 LEGRAND	5,46 €	6	32,76 €
	CONECTOR HEMBRA AEREA 10SIME	1,47 €	6	8,82 €
	LATIGUILLO F.O. LC/PC-LC/PC MULTIMODO DUPLEX 50/125	8,40 €	6	50,40 €
	LM 020D5LCPSCP LC/UPC SC/UP 50/125 2M DP	7,67 €	20	153,30 €
	CAJA 6 MODULOS ESTANCA	26,09 €	1	26,09 €
	CAJA 8 MODULOS ESTANCA M.G.	40,61 €	1	40,61 €
	INTERPLAST DE 21 M TUBO	6,29 €	25	157,24 €
	ROLLOS CINTA AISLANTE PCV	1,18 €	10	11,76 €
	CONECTOR H FTP CAT-6 180º LSA	7,71 €	60	462,42 €
	LAT. F.O. MULTIMODO DUP 2MT	6,83 €	2	13,65 €
	PIGTAL FO SC/PC MULT OM2	3,11 €	40	124,32 €
	CANDADO BLOQUEO PARA DISPOSITIVO	14,78 €	5	73,92 €
	BLOQUEO DE MANETAS PARA INTERRUPTOR 5TE8	3,89 €	5	19,43 €
	1150/2.5 CABLE ALIMENTACION SCHUKO A IEC-320 2.5m	4,21 €	10	42,11 €
	1290/4 CONECTOR TELEFONICO RJ - 11 4 VIAS	0,07 €	8	0,59 €
AUTOTRANSFORMADOR II 6,3KVA 400/230V EN CAJA METALICA	323,40 €	1	323,40 €	
				<b>1.721,57 €</b>
<b>MANO DE OBRA</b>	DESARROLLO INGENIERO CUALIFICADO (€/hora)	32,00 €	1480	47.360,00 €
	PUESTA EN MARCHA INGENIERO CUALIFICADO (€/hora)	36,00 €	680	24.480,00 €
	MONTAJE EN TALLER TÉCNICO ELECTRICISTA (€/hora)	26,00 €	160	4.160,00 €
	PUESTA EN MARCHA TÉCNICO ELECTRICISTA (€/hora)	28,00 €	240	6.720,00 €
				<b>82.720,00 €</b>

<b>IMPREVISTOS</b>	2%	<b>2.661,10 €</b>
<b>BENEFICIO INDUSTRIAL</b>	8%	<b>10.857,28 €</b>
<b>I.V.A</b>	21%	<b>30.780,39 €</b>
<b>TOTAL</b>		<b>177.353,6 7 €</b>

El presupuesto total correspondiente al proyecto "DESARROLLO Y SEGURIDAD SCADA DE UNA PLANTA CERÁMICA" asciende a la cantidad de CIENTO SETENTA Y SIETE MIL TRESCIENTOS CINCUENTA Y TRES EUROS Y SESENTA Y SIETE CÉNTIMOS (177.353,67 €).

