



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**DESARROLLO DE UNA APLICACIÓN LOW  
COST PARA EL CONTEO DE LA  
PRODUCCIÓN DE UNA EMPRESA TEXTIL  
CON DESARROLLO DE ENCODERS DE  
BAJO COSTE PARA BAJAS VELOCIDADES**

AUTOR: SAMVEL YERITSYAN

TUTOR: RICARDO PIZÁ FERNÁNDEZ

**Curso Académico: 2016-17**

## Resumen

El presente TFG trata el problema de conteo de los metros de producción de vendajes de una fábrica dedicada a la fabricación de productos para el ámbito sanitario situada en la provincia de Valencia. Se parte de una situación inicial donde se encontraba como principal deficiencia la ausencia de un método riguroso y fiable de conteo de la producción diaria y se llega a una resolución del problema por medio de un PLC, unos sensores inductivos y una pantalla táctil conectada al servidor de la empresa.

**Palabras clave:** Automatización industrial, PLC, Sensores, Vendajes



## RESUM

El present TFG tracta d'un problema en el conteo dels metres de producció d'embenats d'una fàbrica dedicada a la fabricació de productes pel àmbit sanitari situat en la província de València. Partint d'una situació on es troba com a principal deficiència la absència d'un mètode rigorós i fiable de conteo de la producció diària i arribant a una resolució del problema mitjançant un PLC, uns sensors inductius i una pantalla tàctil connectada al servidor de la empresa.

**Paraules clau:** Automatització industrial, PLC, sensors, embenats



## **ABSTRACT**

The current Project deals with the problem of counting the meters of production of bandages on a factory dedicated to the manufacture of sanitary products, located in the province of Valencia. Starting on an initial situation where the main deficiency was the absence of a rigorous and reliable counting method of the daily production and reaching a resolution of the problem by means of a PLC, inductive sensors and a touch screen connected to the company's server

**Key words:** Industrial automation, PLC, Sensors, Bandages



## **ÍNDICE**

### **Documentos contenidos en el TFG:**

- Memoria
- Presupuesto
- Planos





# MEMORIA DEL TFG



# ÍNDICE DE LA MEMORIA

## Contenido

1.	Introducción .....	13
2.	Objeto y objetivos .....	13
3.	Antecedentes y justificación .....	14
4.	Metodología .....	15
5.	Marco teórico .....	16
5.1.	Sistema de control.....	16
5.1.1.	Sensores: .....	16
5.1.2.	PLC: .....	17
5.1.3.	Módulo E/S: .....	18
5.2.	Software .....	18
6.	Fase de diseño .....	21
6.1.	Introducción .....	21
6.2.	Elección de componentes .....	21
6.3.	Implementación .....	24
6.3.1.	Grafset:.....	24
6.3.2.	Diagrama de contactos:.....	27
6.3.3.	Diseño de la pantalla: .....	29
6.3.4.	Realización de informe .....	32
6.4.	Tablas de variables: .....	36
7.	Pliego de condiciones .....	43
7.1.	Definición y alcance.....	43
7.1.1.	Objeto del pliego .....	43
7.1.2.	Pasos a seguir: .....	43
7.2.	Condiciones y normas generales.....	43
7.3.	Condiciones particulares .....	44
7.3.1.	PLC: .....	44
7.3.2.	Fuente de alimentación:.....	44
7.3.3.	Módulos de Entrada/Salida .....	45
7.3.4.	PC:.....	45

7.4. Legislación aplicada .....	46
8. Presupuesto.....	47
8.1. Introducción .....	47
8.2. Hardware:.....	47
8.3. Software: .....	48
8.4. Recursos humanos .....	48
8.5. Presupuesto total .....	49
9. Conclusiones.....	51
10. Bibliografía.....	53
Anexos .....	1
Anexo I Pruebas de laboratorio.....	5
Anexo II Macros de Excel VBA .....	9
Planos .....	1

## 1. Introducción

El presente Trabajo Fin de Grado aborda la necesidad de implementación de un sistema de conteo de la producción diaria de bobinas de vendajes en una fábrica textil.

El avance tecnológico permite utilizar sistemas automáticos de conteo aumentando la eficiencia y reduciendo los errores de cálculo. La falta de la información precisa sobre la producción de la fábrica supone pérdidas, aumento de stock y gasto de material innecesario.

Así mismo, la falta de históricos de datos sobre las paradas en las máquinas supone una pérdida de eficacia muy alta, ya que no se puede realizar contramedidas para evitar esos fallos y realizar un mantenimiento preventivo programado.

El proyecto se apoya en las prácticas realizadas por el autor en la empresa CALVO IZQUIERDO S.L. entre diciembre de 2016 y junio de 2017 como parte de sus estudios en el Grado de Ingeniería en Tecnologías Industriales.

## 2. Objeto y objetivos

El objeto del presente Trabajo Fin de Grado es la automatización del conteo de los metros de vendaje producidos por una fábrica dedicada a la fabricación de productos para el ámbito sanitario situada en la provincia de Valencia.

La propuesta pretende implementar un sistema automático que cuente de forma automática los metros producidos por cada máquina y cree una serie de informes con periodicidad diaria, semanal y mensual de la producción de la fábrica. Con ello se contribuirá a su objetivo general, que es la creación de un sistema de conteo riguroso y preciso para disminuir las pérdidas de información y recursos en la empresa apoyado en un histórico de datos para en un futuro abordar la problemática de la planificación de tareas de mantenimiento preventivo.

Como objetivos del proyecto a nivel de resultados prácticos se propone:

- Diagnóstico de la situación actual
- Definición de la propuesta de mejora, con la aprobación de la empresa
- Elaboración de documentación para la implementación de las mejoras propuestas
- Implementación de la propuesta

### 3. Antecedentes y justificación

Previamente en la sala de tejeduría de la fábrica se implementó un sistema de conteo basado en un reloj instalado en cada máquina. El sistema consiste en un reloj contador instalado en cada máquina que cuenta una vuelta en un tiempo determinado, independiente de la máquina. En el momento de instalación se estimó un valor aproximado de metros producidos por vuelta de reloj para cada máquina y se implementó una tabla Excel en la que se recoge la información de cada máquina de forma manual al terminar cada jornada.

El resultado de este sistema de medición no es preciso y causa una pérdida de información y recursos, debido a la falta de precisión de la medición y la falta de rigor en la implementación del informe.

La justificación del presente proyecto es la necesidad de implementar un sistema de conteo de la producción preciso y fiable.

Por otro lado, el presente proyecto tiene una justificación académica, siendo el mismo conducente para su autor a la adquisición de determinadas competencias necesarias para la finalización de sus estudios de Grado en Ingeniería en Tecnologías Industriales.

## 4. Metodología

El presente Trabajo Final de Grado se ha desarrollado en tres fases principales: Análisis de la situación precedente, diseño del nuevo sistema e implementación de este nuevo sistema.

La información requerida para la realización del documento en cuestión se ha obtenido principalmente por consultas a gerencia, apoyo del tutor del TFG, observación directa y toma de datos de las instalaciones.

Las distintas etapas dentro de las fases generales de trabajo son:

### Fase de análisis:

- Estudio de las bases teóricas que sustentan el proyecto.
- Recopilación de información de la empresa mediante consultas y solicitudes de documentación.
- Análisis de la información y estudio del sistema actual de conteo y realización de informe de la producción:
  - Medición actual de la producción diaria
  - Paso de la medición realizada hasta la base de datos final
  - Previsiones de ampliación de producción de la empresa
  - Dimensiones de la sala y la maquinaria

### Fase de diseño:

- Planteamiento de alternativas: propuesta de autómatas y distintos sensores, como distintas formas de conexión. En este apartado se ha realizado un estudio de distintas posibles instalaciones para la implementación del nuevo sistema, bajo criterio principal de reducir costes de la instalación cumpliendo los objetivos mínimos.
- Evaluación de la viabilidad de las propuestas y selección de la más adecuada.
- Adquisición de componentes seleccionados y análisis de funcionamiento en laboratorio. Se realizó en dos subetapas:
  - Pruebas de laboratorio en la universidad
  - Pruebas en una máquina de la empresa
- Instalación en fábrica del sistema completo.



## 5. Marco teórico

En el presente apartado se presentan las bases teóricas en las que se ha apoyado la realización del proyecto.

### 5.1. Sistema de control

Se define como tal un conjunto de elementos que permite controlar una planta industrial.

Abarca varios aspectos como:

- Instrumentación industrial
- Sistemas de transmisión de datos
- Software de control de procesos industriales
- Software de supervisión y monitorización

La instrumentación industrial es el conjunto de elementos que sirve para medir, convertir, transmitir, controlar, etc. las variables de un proceso.

Se puede dividir en dos partes principales:

- Parte operativa:
  - Proceso físico
  - Sensores
  - Actuadores
- Parte de control:
  - PLC
  - Módulos de entrada y salida

#### 5.1.1. Sensores:

Los sensores son los elementos encargados de adquirir la información básica del sistema.

Pueden ser de muchos tipos, pero en el presente TFG se ha centrado el análisis en tres:

- Sensor óptico (F.1):  
Es un conjunto de emisor y receptor de luz. Emite un bit de 1 cuando detecta la luz y 0 cuando no. Es utilizado normalmente como detector de presencia de tipo barrera.
- Sensor capacitivo (F.2):  
Es un sensor de proximidad cuyo funcionamiento se basa en un circuito oscilante RC y las líneas de campo eléctrico que se cierran por el aire.  
Cuando se aproxima un objeto, interfiere con las líneas de campo generando oscilaciones y una variación de la capacidad. Esta variación depende de la constante dieléctrica, volumen y densidad del objeto.

- Sensor inductivo (F.3):

Es un sensor de proximidad basado en el principio de inducción electromagnética y la variación de la misma al interferir con objetos metálicos. Consiste en una bobina que genera un campo magnético al ser atravesada por una corriente eléctrica. Al aproximar un objeto metálico ferromagnético se induce una corriente de Foucault en él, lo que a su vez genera un campo magnético que se opone al del devanado del sensor causando una reducción de la inductancia del mismo. Esta variación es lo que se detecta e indica la presencia de un objeto metálico.



F.1 Sensor óptico



F.2 Sensor capacitivo



F.3 Sensor inductivo

#### 5.1.2. PLC:

El PLC (Programmable Logic Controller) es un computador industrial que alberga el programa, software, de control de los sensores y actuadores. Su diferencia principal respecto a un computador convencional es la alta resistencia y la capacidad de trabajar en condiciones de temperatura y humedad sin verse afectado su funcionamiento.

Está compuesto por:

- CPU (F.4):

Es el hardware encargado de interpretar el programa informático diseñado para el control del proceso.

- Fuente de alimentación (F.5):

La CPU en general trabaja a 3V (DC) o 5V (DC), lo que hace que se requiera de una fuente de alimentación que transforme los 230V (AC) de la red en la tensión requerida DC.

Por otro lado, la mayoría de sensores y actuadores trabajan a 12V (DC) o 24V (DC).

Dependiendo del diseño del sistema, se puede tener una, dos o varias fuentes de alimentación.

- Módulos de comunicación:

Son módulos que permiten la conexión del PLC a un dispositivo externo para su manipulación o monitorización.

Hay varios tipos de módulos de comunicación:

- Estándar: RS232, USB, etc.
- Avanzadas: Ethernet, EtherCAST, etc.
- Otros: Profibus, DeviceNET, CAN, etc.

- Módulos especiales

Son módulos que permiten el control de componentes especiales:

- Control de ejes de movimiento
- Gestión de memorias remotas
- Controladores de procesos (PID, etc.)



F.4 CPU



F.5 Fuente de alimentación

### 5.1.3. Módulo E/S:

Los módulos de entrada/salida tienen la funcionalidad de proteger y aislar la parte de control de la parte operativa del sistema. Pueden conectarse directamente al PLC o de forma remota mediante cable RS con un módulo de control remoto.

Internamente dispone de un optoacoplador (un fototransistor junto a un LED infrarrojo) que se encarga de separar los contactos de la entrada del PLC de los bornes de entrada del sensor/actuador, consiguiendo así un aislamiento eléctrico total entre PLC y sensores/actuadores.

El flujo de datos es el que hace la diferencia entre entradas y salidas. Las entradas están reservadas para sensores, mientras que las salidas para actuadores.

## 5.2. Software

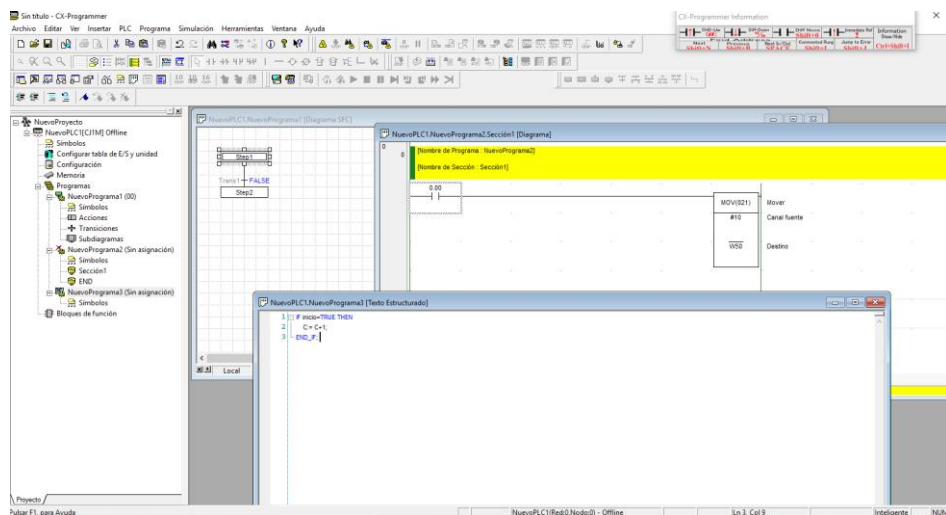
El PLC es un computador programable y para definir el programa se utiliza un software específico dependiendo de la marca y de las prestaciones del mismo.

En el caso particular del presente TFG se ha utilizado el software de OMRON: Cx-Programmer para la programación de la CPU y NB-Designer para el diseño del interfaz de usuario en una pantalla táctil de la serie NB. Así mismo se ha utilizado el paquete de software CX-Server para la implementación de un servidor OPC y la conexión PC/PLC.

### CX-Programmer:

Es un software de programación de autómatas programables. Permite la programación de los autómatas de OMRON en hasta tres lenguajes distintos, dependiendo de las prestaciones del modelo de PLC.

- **Diagrama de contactos o ladder diagram:**  
Es el lenguaje de programación más básico. Es un lenguaje gráfico basado en el lenguaje de relés.
- **GRAFSET:**  
Es un diagrama funcional normalizado que permite modelar el proceso de forma gráfica. Permite una visualización más rápida y fácil del proceso programado.
- **Texto estructurado (ST):**  
Es un lenguaje de programación de alto nivel basado en Pascal. Permite programación compleja con bloques de funciones.



F.6 Vista general de las ventanas de los tres lenguajes de programación en CX-Programmer

### NB-Designer:

Es un software de diseño de pantallas para la serie NB de pantallas táctiles de OMRON. Permite crear las distintas pantallas que se quiera tener en una pantalla NB y ponerlo a prueba en su modo de simulación enlazado a un PLC.

Tiene tres modos de simulación:

- **Simulación Offline:**  
Es un modo de simulación en el que se muestra una de las pantallas diseñadas en el ordenador. Es una simulación puramente visual, para comprobar la posición de los elementos creados en la pantalla.

- Simulación indirecta:  
Permite trabajar directamente con una pantalla NB conectada al ordenador. En este modo de simulación se transfiere el programa a la pantalla una vez y permite realizar cambios y pruebas sin tener que volcar nuevamente el programa.
- Simulación directa:  
Es un modo de simulación en el que se trabaja con una pantalla virtual y un PLC físico conectado al puerto serie del ordenador. Permite visualizar la pantalla antes de adquirirla físicamente. La desventaja del modo es que solamente se puede simular con PLC que permita una conexión al puerto serie del ordenador con el protocolo RS-232.

### CX-Server:

Es el software proporcionado por OMRON para la elaboración de un servidor OPC.

El OPC (*Object linking and embedding for Process Control*) es un protocolo de comunicación entre fuentes de datos (p.e. PLC, módulos E/S, etc.) y “clientes OPC” (HMI, generadores de informes, etc.).

En el caso particular del proyecto actual la fuente de datos es el PLC CJ2M instalado en fábrica y el cliente OPC es el fichero Excel que se encarga de realizar el informe de la producción.

El software incorpora una serie de herramientas ActiveX (F.7) que permite la conexión directa de las variables de entrada y salida del autómatas con celdas individuales de Excel.



F.7 Herramientas ActiveX del pack de software CX-Server

## 6. Fase de diseño

### 6.1. Introducción

La instalación a implementar consiste en un sistema sencillo en el que se dispone de un sensor de lectura en cada máquina que, al activarse, indica una vuelta del eje motor de la misma.

Para ello se ha decidido utilizar un sistema automático basado en PLC (Programable Logic Controller).

### 6.2. Elección de componentes

El primer componente a analizar es el sensor que se utilizará para la toma de datos.

Se ha realizado un análisis de los tres tipos de sensores mencionados anteriormente:

#### Sensor óptico:

Se consideró la opción de utilizar un sensor óptico dado su bajo coste y la alta variedad de distancia de detección, lo que da mayor flexibilidad en la instalación del mismo en las máquinas.

Ópticos	rango	Tensión (v)	precio (€)
<b>Omron EESX1140</b>	14mm	4 dc	4,19€
<b>Optec OPB800W51Z</b>	9.53mm	2 dc	3,91€

*T.1 Sensores ópticos*

Esta propuesta fue rechazada debido a la suciedad que se acumula dentro de la máquina, lo que interfiere de forma drástica en el funcionamiento del mismo, obstruyendo el sensor.

#### Sensor capacitivo:

Los sensores capacitivos son considerablemente más robustos que los ópticos. Tiene un rango de detección aceptable, pero la presencia de pelusa y suciedad en la máquina puede interferir de forma negativa en el funcionamiento, dando mediciones erróneas.

Capacitivos	rango	Tensión (v)	Precio (€)
<b>Autonics CR18-8AC</b>	8mm	10-40 dc	59,19€
<b>CA30CAF16PA</b>	2-20 mm	10-40 dc	42,94€
<b>CD50CNF06No</b>	0-6 mm	10-30 dc	39,5€

*T.2 Sensores capacitivos*

#### Sensor inductivo:

Es la opción más rígida. Al detectar solo materiales ferromagnéticos no se ve afectado por la suciedad acumulada en el interior de la máquina.

Inductivos	distancia(mm)	longitud(mm)	diam.(mm)	tensión(V)	Precio (€)
E2B-M18KN16-M1-B2	16	39	18	12 a 24	24,14
E2B-M30LN30-WP-B1 2M	30	63	30	12 a 24	38,22
E2B-M12KN08-WP-B1 2M	8	30	12	12 a 24	20,97
E2B-M18KN16-WP-B1 2M	16	36	18	12 a 24	22,3
E2A-M18KS08-M1-B1	8	53	18	12 a 24	27,98

*T.3 Sensores inductivos*

Además de los tres tipos de sensores expuestos, se valoró la posibilidad del uso de un Encoder para la medición de las vueltas de eje de las máquinas.

Un Encoder es una clase de transductor que transforma la posición angular de un eje de giro en información digital. Este dispositivo permite realizar una lectura directa del ángulo de giro del eje, por tanto permitiría medir directamente los metros de producción de vendajes. Para la lectura de datos se requiere un tipo de módulo de entradas de alta frecuencia que permita realizar lecturas rápidas. Estos módulos son módulos especiales con un coste muy elevado. En el proyecto actual basta con medir vueltas completas o medias vueltas, por lo que no queda justificado la inversión en el uso de un elemento de alta precisión y, por tanto, se ha descartado esta opción.

De las distintas opciones se eligió el sensor ESA- M18KS08-M1-B1 ya que tiene una distancia de detección aceptable y un mayor aislamiento eléctrico que la serie E2B, sin tener un precio excesivamente elevado.

### **PLC:**

El siguiente componente a elegir fue el PLC. Se estudiaron varias opciones que pudieran cumplir con las necesidades de la empresa. Los criterios para la elección del mismo fueron un bajo coste de la instalación y la posibilidad de programación en un lenguaje de fácil manejo para modificaciones posteriores.

	compatible	E/S	Red	Memoria (Datos/Programa)	Precio (€)
Omron CP1L Relé	CP1H, CJ1 y CS1	30 (18 dig) Ampliable - 160	USB	32k/10k	444,33
Omron CP1E Relé	CP1H, CJ1 y CS1	30 (18 dig) Ampliable- 150	USB	32k/8k	332,09
Omron CJ1M	CQM1	Max 640 E/S	DeviceNet	32k/5k	373,43
Omron CJ2M	Todos	Max 2560 E/S	Ethernet	64k/5k	754,32

T.4 Modelos de PLC estudiados

Se eligieron dos modelos aptos para las condiciones de trabajo de la empresa en particular:

- CP1LM60DRD: Un PLC compacto con 35/24 E/S, suficientes para las condiciones actuales y con posibilidad de incorporar máquinas posteriormente. La desventaja de esta opción es la falta de posibilidad de programar con GRAFCET.
- CJ2MCPU31: Un PLC de mayor categoría, permite la programación en los tres lenguajes expuestos anteriormente. Tiene el inconveniente de no disponer de E/S directa.

Para la implementación del CJ2M se estudió dos opciones:

- PLC CJ2MCPU31 centralizado con módulos E/S incorporados:

	Cantidad	Precio	Dto	Importe
CJ2MCPU31	1	876,5	20	701,2
CJ1WPA202	1	169,2	20	135,36
CJ1WID211	1	179,4	20	143,52
CJ1WID201	1	146	20	116,8
CJ1WOD212	1	210,2	20	168,16
CJ1WOD202	1	55,85	20	168,16

T.5 Tabla de componentes para CJ2M centralizado

- PLC CJ2MCPU31 con módulos remotos de E/S (20/20):



	Cantidad	Precio	Dto	Importe
CJ2MCPU31	1	876,5	20	701,2
CJ1WPA202	1	169,2	20	135,36
NXEIC202 ETHERNET	3	361,4	20	867,36
NXID4442 8 E	2	73,1	20	116,96
NXOD4256 8 S	2	81,2	20	129,92
NXID3417 4 E	1	55,85	20	44,68
NXOD3256 4S	1	66	20	113,61
S8VKG03024	3	54,1	30	113,61

T.6 Tabla de componentes para CJ2M con módulos E/S remotos

Se consideró la opción de PLC CJ2MCPU31 con módulos remotos de E/S para reducir los costes de instalación de cableado, pero el sistema así diseñado presenta un sobrecoste de 789.95 € respecto al PLC centralizado.

Así, teniendo en cuenta las ventajas y desventajas presentadas para las distintas opciones, se decidió optar por la opción de un PLC CJ2MCPU31 centralizado con módulos de E/S incorporados.

#### **Pantalla táctil:**

Se optó por la pantalla táctil NB7WTWD1B 7" de OMRON debido a la compatibilidad directa con el resto de los componentes y la diferencia de precio respecto a la serie NS de pantallas de OMRON.

### **6.3. Implementación**

El programa principal se ha desarrollado con el software de OMRON, Cx-Programmer. Se han empleado dos lenguajes de programación:

#### **6.3.1. Grafcet:**

El objetivo principal del proyecto es la creación de un sistema totalmente automatizado de conteo de la producción. Por ello se ha implementado el GRAFCET (F.9) con el siguiente funcionamiento:

Al encender el PLC se pasa automáticamente al estado de lectura de los sensores (Step2) en el que el autómatas permanece hasta que se active el sensor inductivo correspondiente o se detecte una parada.

El sensor inductivo está implementado en la máquina junto a un anillo con un "diente" metálico de forma que imita el funcionamiento de un Encoder (F.8), dando una lectura de bit 1 al detectar el "diente" metálico, indicando una vuelta de eje. Si el sensor tarda más de un tiempo determinado (una aproximación ponderada con un margen de seguridad del tiempo de una vuelta de eje (temp1)) en

## DESARROLLO DE UNA APLICACIÓN LOW COST PARA EL CONTEO DE LA PRODUCCIÓN DE UNA EMPRESA TEXTIL CON DESARROLLO DE ENCODERS DE BAJO COSTE PARA BAJAS VELOCIDADES

realizar la lectura, significa que la máquina se ha detenido, con lo que el sistema determina que la máquina ha pasado a un estado de parada (step3).

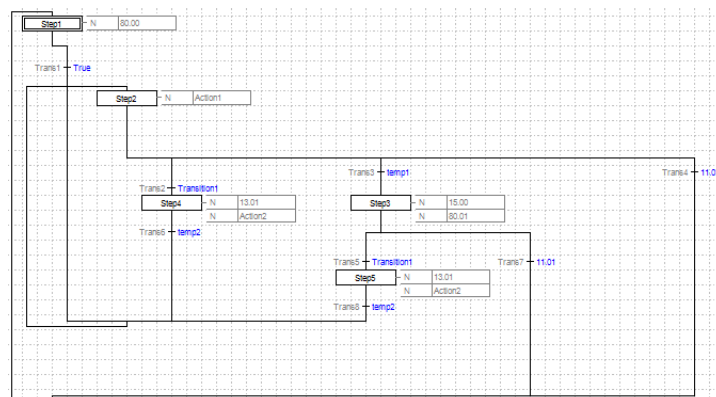


**F.8** Sensor inductivo con anillo con diente metálico en la máquina

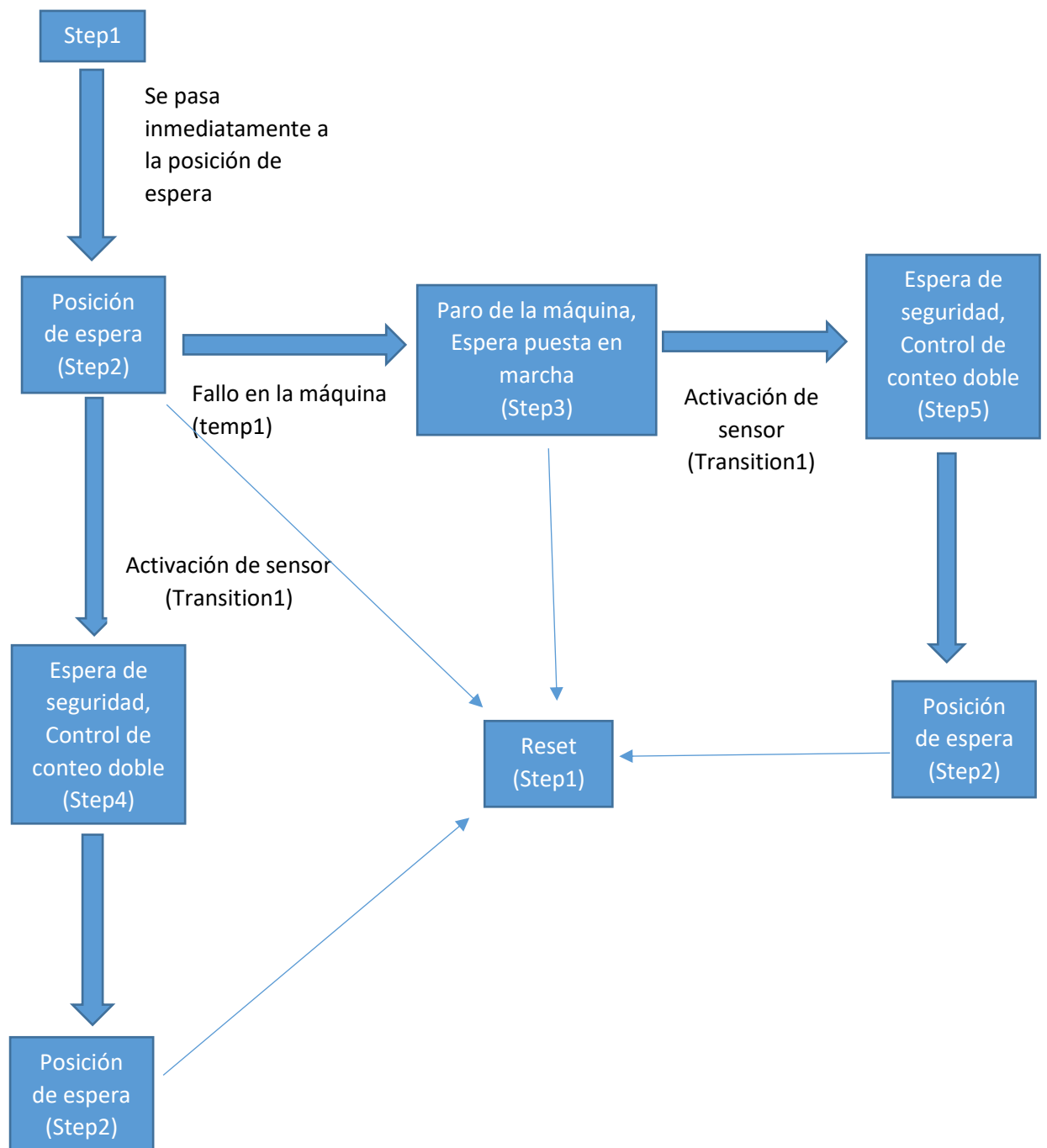
En caso de entrar en el modo de parada, el autómata permanece en espera hasta recibir una nueva lectura del sensor, lo que se interpreta como que la máquina se ha reparado y se ha puesto en marcha.

Al activarse el sensor, tanto en el modo parada (step5) como en el modo regular de funcionamiento (step4), se pasa a un estado auxiliar en el que el autómata permanece durante un segundo (temp2) para evitar lecturas dobles debido a las vibraciones de la máquina.

Por otro lado, cada máquina dispone de un botón “Reset” en la pantalla que se representa en el GRAFCET como “11.xx” (xx varía en función de la máquina a la que está asociado el botón).



**F.9** GRAFCET implementado en CX-Programmer



6.3.2. Diagrama de contactos:

Todas las acciones secundarias y la interacción con PC/Pantalla se realizan a través de diagramas de contactos.

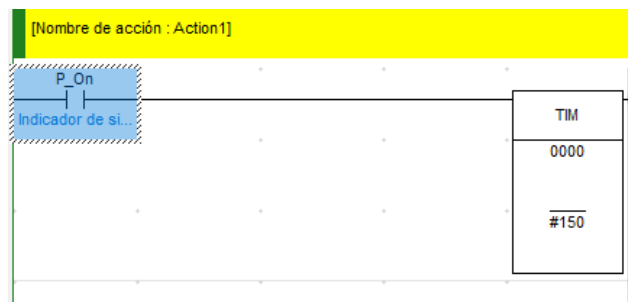
Transition1:

Es una transición en diagrama de relés que se activa al tener un flanco de subida en el sensor correspondiente en cada máquina.



F.10 Transition1

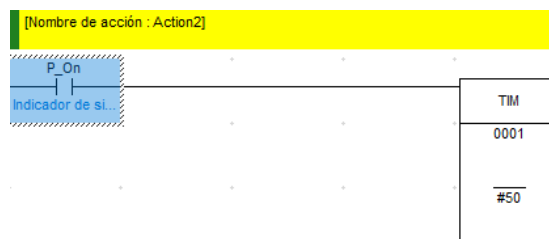
Action1: Es una acción en diagrama de relés que se encarga de activar el temporizador para la activación del estado de “paro”. Se corresponde con la salida “temp1”. “temp1” es una variable local que está asociada a un temporizador correspondiente a cada máquina.



F.11 Action1

Action2: Es una acción en diagrama de relés que se encarga de activar el temporizador de seguridad para evitar lecturas dobles y erróneas debido a las vibraciones de la máquina.

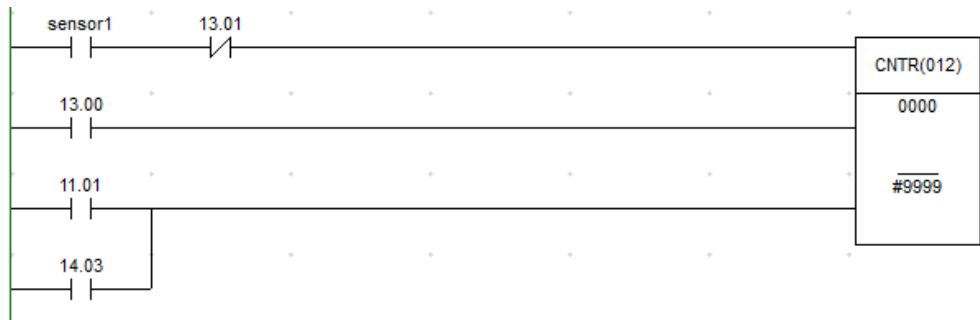
Activa la variable local “temp2”, que está asociada a un temporizador correspondiente a cada máquina.



F.12 Action2

Contadores:

Todos los contadores de las máquinas se han implementado con el bloque “CNTR (012)” en diagramas de relés. El bloque de funciones “CNTR (012)” corresponde con un contador bidireccional (suma/resta) aunque en el proyecto actual se ha empleado únicamente el bit de suma del mismo.



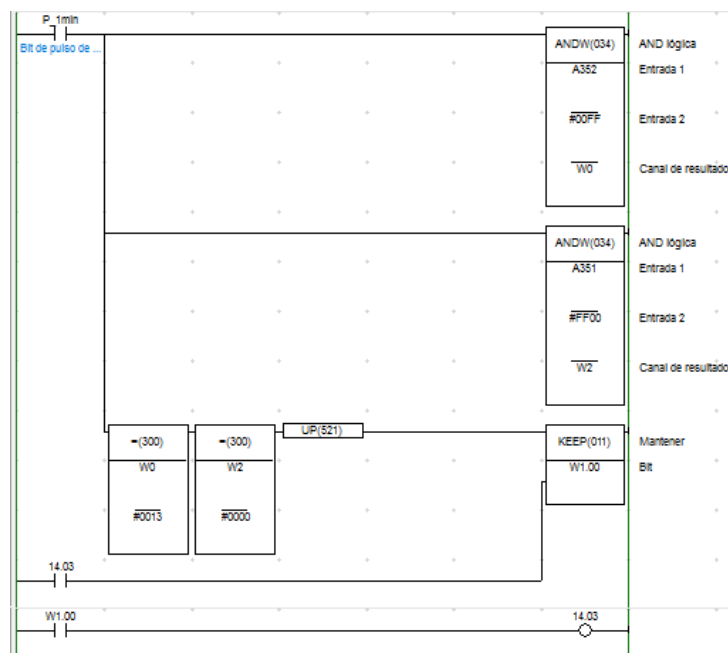
F.13 Contador de vueltas de eje

Reinicio de contadores entre jornadas:

En el caso particular del proyecto se desea realizar registros de la producción por jornadas, por lo que se ha implementado un programa en diagrama de contactos que activa la salida “14.03” asociada al reset de los contadores.

Hay posibilidad de tener tres jornadas en un mismo día, con un mínimo de dos, por lo que los contadores se resetean tres veces al día: a las 6:00, a las 14:00 y a las 22:00.

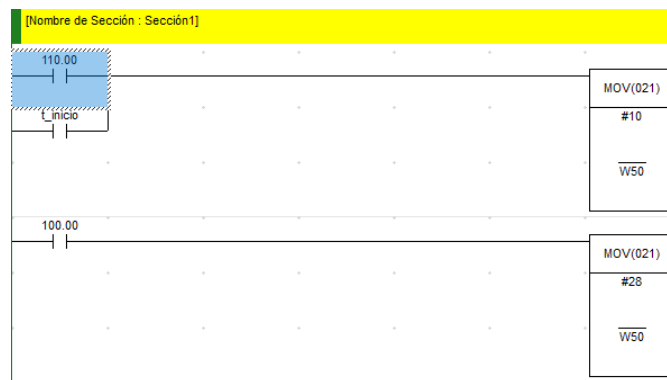
El diagrama compara la hora actual con las tres horas anteriormente dichas y, en caso de coincidir, activa la salida “14.03”.



F.14 Reinicio de contadores entre jornadas

### Conexión con pantalla táctil:

La interacción con la pantalla táctil se realiza a través de la dirección de palabra “W50”. Esta interacción es únicamente para la realización de cambio de las distintas pantallas desarrolladas en NB-Designer. Por medio de un diagrama de relés se ha implementado un programa que escribe, a través de la función “MOV (021)” el número de pantalla a la que se desea cambiar en “W50” al activar los distintos contactos. El resto de variables y E/S se transfiere de forma automática a la pantalla por medio de una conexión EtherNET directa.



**F.15** Diagrama de contactos encargado de cambio de pantallas

Todo el programa en conjunto (tanto Grafcet, como diagramas de contactos) se vuelca al PLC a través de una conexión USB directa al ordenador.

Todas las variables de Entrada/Salida utilizadas se detallan en el apartado “6.4 Variables E/S”.

### 6.3.3. Diseño de la pantalla:

El paso siguiente es diseñar la pantalla táctil. Como se dijo anteriormente, en este proyecto se ha utilizado la pantalla NB7W-TW01B de OMRON. El diseño de la misma se ha implementado a través del software NB-Designer de OMRON.

Se ha definido veintiuna pantallas correspondientes a:

Pantalla de inicio:

Es una pantalla (F.16) de bienvenida en la que se permanece durante unos segundos, después de encender el equipo, antes de ir a la pantalla principal de las máquinas.

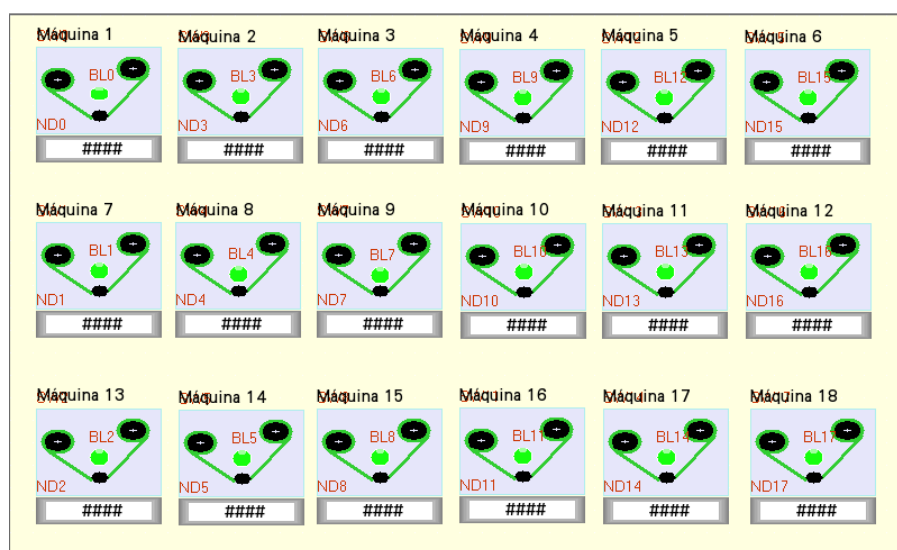


F.16 Pantalla de inicio

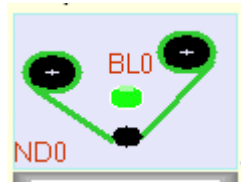
Pantalla Principal:

En esta pantalla (F.17) se dispone de información breve sobre el estado de cada máquina. Se dispone:

- Botón booleano (F.18) asociado a cada pantalla correspondiente a cada máquina. Está asociado al CIO\_bit (Input/Output Bit) de salida correspondiente al cambio de pantalla de cada máquina en CX-Programmer. Es un botón momentáneo, es decir, activa el estado 1 en la salida correspondiente al ser pulsado y el estado 0 en reposo.
- Junto al botón booleano se dispone de un indicador LED (F.18: *BL0*) que muestra el estado de la máquina. Está asociado a una salida de tipo CIO\_bit correspondiente a cada máquina. Este led indica si la máquina está en funcionamiento o en estado de parada. Se enciende una luz verde cuando el bit está a nivel 0 y una luz roja parpadeante cuando el bit está a nivel 1. El bit a nivel 1 indica la parada de la máquina, esto se debe a que está asociado a una salida en CX-Programmer que se activa (Bit de salida nivel 1) al entrar en modo de parada.
- Contador numérico (F.19) asociado al contador de vueltas de eje correspondiente a cada máquina.



F.17 Pantalla principal



F.18 Botón booleano junto a LED indicador



F.19 Contador numérico

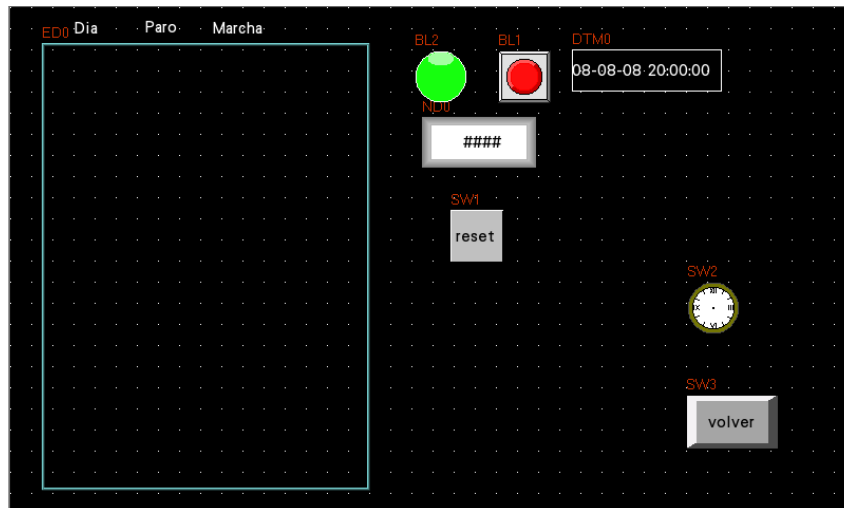
### Máquina#X:

Son las dieciocho pantallas correspondientes a cada máquina. En estas pantallas (F.20) se dispone de información detallada del estado de cada máquina.

Se dispone:

- Contador numérico correspondiente al contador de vueltas de eje asociado a cada máquina. Es idéntico al contador numérico dispuesto en la pantalla Principal.
- LED indicador del estado de la máquina, idéntico al LED indicador dispuesto en la pantalla principal.
- LED indicador del estado del sensor (F.20: *BL1*) (activado/desactivado). Es un LED similar al LED indicador del estado de la máquina, sirve para comprobar el correcto funcionamiento de los sensores.
- Botón “reset” para reiniciar el contador de la máquina correspondiente (F.20: *Reset*). Es un botón booleano asociado al bit de salida correspondiente al “reset” del contador asociado a la máquina.
- Fecha actual. Se muestra en un panel la fecha y la hora actual en formato DD/MM/AAAA HH/mm/ss.
- Botón “Volver” para volver a la “Pantalla principal”. Un botón booleano asociado a la salida de bit correspondiente a la activación de la “Pantalla principal” en CX-Programmer.
- Botón en forma de reloj para acceder a la pantalla de configuración del reloj de la pantalla táctil.
- Una tabla en la que se registran las paradas de la máquina correspondiente. En esta tabla se puede visualizar de forma directa la fecha y la hora de parada y de puesta en marcha de cada máquina. La tabla se borra al reiniciar la pantalla.





F.20 Pantalla "Máquina#X"

#### 6.3.4. Realización de informe

El último paso para terminar la programación del sistema es la elaboración de una "Macro" en Excel para el registro y el procesamiento automático de la lectura de los sensores.

Para la elaboración del informe se han realizado consultas a la gerencia de la empresa y se ha realizado una adaptación del formato previo a la instalación del nuevo sistema.

En el caso particular de la empresa en la que se ha realizado el proyecto, los informes se guardan en ficheros Excel en los que se guarda la información por columnas de cada máquina.

El Excel incluye las siguientes columnas:

- Número de máquina.
- Metros producidos por jornada.
- Tiempo de funcionamiento de la máquina a lo largo de la jornada.
- Rendimiento de la máquina.
- Número de paradas.

En la elaboración del nuevo sistema se ha añadido la siguiente información:

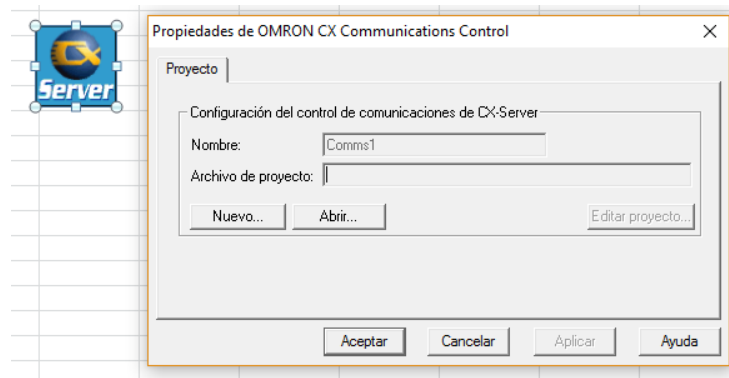
- Vueltas de eje de la máquina a lo largo de la jornada.
- Diámetro del eje de la máquina.
- Número de cabezas de la máquina.
- Hora de parada y hora de puesta en marcha de la máquina

El nuevo informe se basa en un fichero Excel con dos Macros (abreviado de macroinstrucción). La primera se encarga de tomar los datos de lectura de las máquinas, es decir, anota las vueltas de eje de la máquina y las horas de parada y de puesta en marcha a lo largo de la jornada. La segunda es una macro encargada de procesar la información de la "Hoja1" del Excel y mostrarlo en la "Hoja2" del mismo fichero.

**Toma de datos:**

Para la toma de datos de las máquinas se ha implementado una conexión PLC-PC a través del servidor OPC proporcionado por OMRON en su paquete de software Cx-Server.

El primer paso es la creación del Servidor OPC añadiendo las distintas variables utilizadas en CX-Programmer a un archivo de servidor. Para ello se agrega en Excel la herramienta “OMRON CX Communications control” y se accede al panel de propiedades (F.21) haciendo click derecho encima y eligiendo “OMRON CX Communications control Object/properties”.

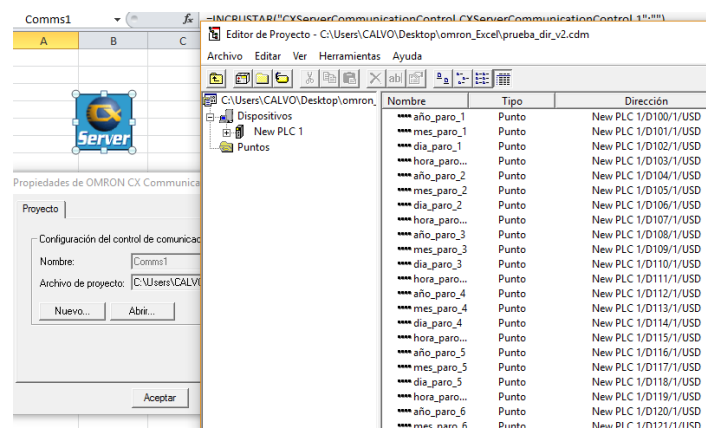


F.21 Propiedades de OMRON CX Communications Control

En esta pantalla se puede abrir un archivo de CX-Server creado externamente a partir de un programa creado en CX-Programmer exportando todos los símbolos del mismo o crear un nuevo archivo.

En ambos casos se puede editar el archivo presionando “Editar proyecto”. Al presionar dicho botón se accede a la ventana de CX-Server en la que se puede añadir/modificar propiedades del PLC con el que se trabaja, así como la vía de conexión entre el PLC y el PC.

Una vez agregado el PLC se procede a agregar las distintas variables utilizadas en el proyecto.



F.22 Editor de proyecto de CX-Server

### **Macros:**

Como se ha mencionado anteriormente, la elaboración de informes se realiza a través de dos macros de Excel.

Las Macros son conjuntos de instrucciones en Excel que se ejecutan de forma secuencial a través de una orden de ejecución. En el presente proyecto se ha empleado dos órdenes de ejecución:

- OMRON CX Timer control: Es una herramienta ActiveX del paquete de software CX-Server de OMRON. Ejecuta las líneas de programa escritas en ella de forma periódica. En esta misma herramienta se ha escrito el primer macro del programa.
- Botón (Control de formulario): Es una de las herramientas de “Desarrollo” incorporadas en Excel. Activa el macro asociado al ser pulsado y ejecuta todas las instrucciones una única vez.

Las Macros se programan en Microsoft VB (Visual Basic). Es un lenguaje de programación dirigido por eventos, en el que se va ejecutando las instrucciones de forma secuencial.

Se va a proceder a continuación a explicar el funcionamiento de las Macros elaborados en el proyecto actual:

### **Hoja1:**

En esta hoja, como se ha dicho anteriormente, se va a guardar la información de lectura de los sensores.

La estructura de la tabla es tal que en las filas aparece la información asociada a cada máquina, en las primeras columnas (F.23) aparece:

- Nº de máquina
- Contador
- Nº de cabezas
- Diámetro
- Producción por jornada

Así mismo en la primera fila se escribe la jornada actual junto a la fecha (DD/MM/AA). En la primera fila del archivo también aparece la hora actual (HH/mm). Esta hora se usa para determinar la jornada actual.

En las columnas posteriores (F.24) se anota de forma consecutiva las horas de parada y de marcha de cada máquina en la fila correspondiente (HH/mm).





Máquina	sensor
1	0.00
2	0.01
3	0.02
4	0.03
5	0.04
6	0.05
7	0.06
8	0.07
9	0.08
10	0.09
11	0.10
12	0.11
13	0.12
14	0.13
15	0.14
16	0.15
17	2.00
18	2.01

*T.7 Variables de entrada asociados a los sensores correspondientes a cada máquina*

Estas variables corresponden a variables de entrada físicas, es decir se corresponde con los módulos de Entrada/Salida incorporados en el PLC. El "0" indica que es el primer módulo. En particular, en este proyecto se han utilizado dos módulos de entrada y dos módulos de salida, de forma alternada, tal que los módulos "0" y "2" son de entrada, mientras que los módulos "1" y "3" son de salida.

En el proyecto actual se han utilizado únicamente los módulos de entrada, pero se han dispuesto también de los módulos de salida para poder instalar posteriormente luces de emergencia en cada máquina que se enciendan al producirse una parada.

**Variables internas:**

Son variables virtuales que se ha utilizado en el propio programa para pasar de etapas o para activar ciertos elementos como contadores y temporizadores.

Estado: Son las variables de bit empleadas para indicar el estado de funcionamiento de la máquina.

Reg. Parada: Es la variable de bit que se utiliza para guardar las horas de parada y de puesta en marcha. Al tener una parada y entrar en modo de parada se activa la variable. Al tener un flanco de subida, se registra la hora de parada y, por consiguiente, al producirse un flanco de bajada se registra la hora de puesta en marcha.

Máquina	Estado	Reg. Parada	reset	cont_suma(n)
1	15.00	80.01	11.01	13.01
2	15.01	81.01	11.02	13.02
3	15.02	82.01	11.03	13.03
4	15.03	83.01	11.04	13.04
5	15.04	84.01	11.05	13.05
6	15.05	85.01	11.06	13.06
7	15.06	86.01	11.07	13.07
8	15.07	87.01	11.08	13.08
9	15.08	88.01	11.09	13.09
10	15.09	89.01	11.10	13.10
11	15.10	90.01	11.11	13.11
12	15.11	91.01	11.12	13.12
13	15.12	92.01	11.13	13.13
14	15.13	93.01	11.14	13.14
15	15.14	94.01	11.15	13.15
16	15.15	95.01	12.00	14.00
17	16.00	96.01	12.01	14.01
18	16.01	97.01	12.02	14.02
			14.03	

*T.8 Variables internas utilizadas en el proyecto*

Reset: Es la variable de bit que se utiliza para reiniciar los contadores. A parte de las variables asociadas a cada máquina se dispone de una variable (14.03) que reinicia los contadores al inicio de cada jornada.

Cont\_suma(n): Son las variables de bit que se activan después de la activación del sensor. Mientras estas variables estén activas el contador correspondiente no sumará. Esta variable es la que se utiliza para evitar el conteo doble o conteo erróneo en la máquina asociada.

### **Variables E/S**

Son variables que se podrían considerar tanto de salida como de entrada. Son las que conectan la pantalla táctil con el PLC y sirven para cambiar de una pantalla a otra en NB-Designer.

Máquina	cambio pantalla
1	100.00
2	100.01
3	100.02
4	100.03
5	100.04
6	100.05
7	100.06
8	100.07
9	100.08
10	100.09
11	100.10
12	100.11
13	100.12
14	100.13
15	100.14
16	100.15
17	101.00
18	101.01

*T.9 Variables E/S para el cambio de pantallas en NB- designe*

**Direcciones de memoria:**

Las direcciones de memoria son identificadores de localizaciones de memoria en las que los programas informáticos pueden almacenar datos. En estas direcciones se guardan las horas de parada y puesta en marcha. Conectan el PLC y el PC a través del servidor OPC.

A parte de las direcciones de memoria mencionadas, en el proyecto se ha utilizado la dirección de palabra “W50” para el cambio de pantallas en NB-Designer. En esta palabra se guarda, a través de CX-Programmer y al activar una de las variables correspondientes, el número de la pantalla a la que se quiere acceder.



Máquina	D. parada		D. marcha	
1	103	300	203	336
2	107	302	207	338
3	111	304	211	340
4	115	306	215	342
5	119	308	219	344
6	123	310	223	346
7	127	312	227	348
8	131	314	231	350
9	135	316	235	352
10	139	318	239	354
11	143	320	243	356
12	147	322	247	358
13	151	324	251	360
14	155	326	255	362
15	159	328	259	364
16	163	330	263	366
17	167	332	267	368
18	171	334	271	370

*T.10 Direcciones de memoria del PLC utilizadas en el proyecto*

**Contadores:**

Las funciones de contador tienen asociado un bloque de memoria específico en el que se guarda el valor de cada contador utilizado.

En particular, en este proyecto se han utilizado dieciocho contadores asociados a cada máquina para el conteo de las vueltas de eje.

**Temporizadores:**

Al igual que los contadores, los temporizadores están asociados a un bloque de memoria específico. Los temporizadores activan un bit de salida al alcanzar el tiempo establecido.

En el proyecto actual se han utilizado treinta y seis temporizadores.

Cada máquina tiene dos temporizadores asociados:

- El primer temporizador está asociado al modo de parada. En caso de activarse el bit de salida correspondiente se activa la transición para la activación del modo de parada.
- El segundo es el temporizador de seguridad de 1 s para evitar conteos dobles.

Máquina	contador	temp1	temp2
1	C0	T0	T1
2	C1	T3	T4
3	C2	T6	T7
4	C3	T9	T10
5	C4	T12	T13
6	C5	T15	T16
7	C6	T18	T19
8	C7	T21	T22
9	C8	T24	T25
10	C9	T27	T28
11	C10	T30	T31
12	C11	T33	T34
13	C12	T36	T37
14	C13	T39	T40
15	C14	T42	T43
16	C15	T45	T46
17	C16	T48	T49
18	C17	T51	T52

*T.11 Contadores (C#x) y temporizadores (T#x) asociados a la máquina*



## 7. Pliego de condiciones

### 7.1. Definición y alcance

#### 7.1.1. Objeto del pliego

El objetivo de este pliego de condiciones es definir las condiciones técnicas que se han de cumplir para la correcta materialización del proyecto descrito en este documento.

El objetivo del presente trabajo fin de grado es la implementación de un sistema de conteo de metros de producción de una fábrica de material medicinal textil totalmente automatizado.

Las soluciones propuestas en este documento pueden ser modificadas siempre que el sistema cumpla con su objetivo.

#### 7.1.2. Pasos a seguir:

Para la correcta implementación del proyecto propuesto se han de seguir los siguientes pasos:

1. Adquisición del PLC, pantalla táctil y sensores
2. Comprobación del correcto funcionamiento de cada componente
3. Implementación de los programas de funcionamiento para cada componente
4. Interconexión de los componentes principales
5. Prueba de funcionamiento del sistema en laboratorio
6. Instalación de los componentes en fábrica y conexión de los mismos a una máquina
7. Prueba de funcionamiento del sistema en fábrica

El proyecto se ha de llevar a cabo por una persona cualificada, capaz de localizar y afrontar los problemas que surjan durante la programación, instalación y comprobación del funcionamiento de la instalación, así como los problemas que puedan surgir durante el funcionamiento posterior.

### 7.2. Condiciones y normas generales

Todos los componentes empleados en el proyecto son materiales normalizados. En caso de fallo o rotura se han de sustituir por un material de mismas características para asegurar el correcto funcionamiento del sistema.

En caso de necesidad de sustitución de algún componente se ha de realizar bajo supervisión y autorización del encargado del proyecto. En caso contrario toda la responsabilidad en caso de fallo del sistema recaerá en la persona que actúe sin autorización.

### 7.3. Condiciones particulares

El proyecto descrito en este documento utiliza ciertos componentes y software con requisitos de funcionamiento determinados. En caso de necesidad de sustituir alguno de los componentes por uno de distintas características a las mencionadas en este documento, se realizará las modificaciones necesarias al sistema para su correcto funcionamiento. El director del proyecto no se responsabiliza por las consecuencias que pueda traer dichos cambios, ya que el proyecto actual se ha diseñado para su funcionamiento continuo únicamente con componentes y programas descritos aquí.

Los componentes del proyecto son:

#### 7.3.1. PLC:

En el proyecto se ha utilizado el PLC CJ2MCP31 de OMRON.

#### Características técnicas:

Tensión de trabajo	5 Vdc
Tipo de red	Ethernet
Conexión COM	USB 2.0
Capacidad de programa	5000 pasos
Lenguaje de programación	Diagrama de relés SFC Texto estructurado

*T.12 Características técnicas de CJ2M-CPU31*

#### 7.3.2. Fuente de alimentación:

El PLC mencionado requiere de una fuente de alimentación externa para su funcionamiento. Así mismo se requiere una fuente de alimentación para el funcionamiento de los sensores.

- Fuente de alimentación PLC: En este proyecto se ha empleado la fuente de alimentación CJ1W-PA202 con las siguientes características técnicas:

Tensión de salida	5 Vdc
Corriente de salida	400 mA
Potencia de salida	14 W

*T.13 Características técnicas de CJ1W-PA202*

- Fuente de alimentación sensores: En este proyecto se ha utilizado una fuente de alimentación estándar de las siguientes características:

Tensión de línea	230 Vac
Tensión de salida	24 Vdc
Potencia de salida	75 W

*T.14 Características técnicas que ha de cumplir la fuente de alimentación para sensores*

Esta misma fuente se emplea para la alimentación de la pantalla táctil

La fuente de alimentación tiene que cumplir con las normativas de seguridad ante:

- Cortocircuitos
- Sobretensiones
- Perturbaciones de red

### 7.3.3. Módulos de Entrada/Salida

Los módulos de entrada y salida han de ser módulos digitales de 24Vdc en los bornes de salida. En el proyecto actual se ha utilizado los módulos de entrada CJ1WID211 y CJ1WID201 y los módulos de salida CJ1WOD212 Y CJWOD202.

### 7.3.4. PC:

El ordenador personal (PC) empleado es de gran importancia ya que a través de él se realiza la interconexión de los distintos componentes del sistema, así como se vuelca los programas al PLC y la pantalla táctil.

La pantalla táctil no es un elemento de gran importancia, su elección ha sido determinada por el criterio de mínimo coste, pero la serie elegida define el programa a emplear para su diseño.

El software utilizado define las características mínimas necesarias y recomendadas para el PC a emplear para su desarrollo.

En este proyecto se ha utilizado el siguiente software:

- CX-Programmer
- NB-Designer
- CX-Server
- Excel

Cada programa tiene sus requisitos mínimos y recomendados, por tanto, se elegirá un ordenador que cumpla con los requisitos de hardware más restrictivo:

### **Hardware:**

Microprocesador	Intel Pentium II
Memoria RAM	512 MB
Disco duro	20 GB
Conexión a red/COM	USB 2.0 Ethernet

*T.15 Hardware utilizado en el proyecto*

#### **7.4. Legislación aplicada**

Para la implementación del proyecto descrito se ha de tener en cuenta la siguiente legislación:

- Reglamento Electrónica de Baja Tensión
- RD 1580/2006, de 22 de diciembre, a través del cual se regula la compatibilidad electromagnética de los equipos eléctricos y electrónicos
- Normativa UNE en referencia a utilización y mantenimiento de autómatas

## 8. Presupuesto

### 8.1. Introducción

Con el objetivo de realizar una estimación del coste del proyecto, se expone a continuación la documentación detallada del presupuesto del mismo.

En el presupuesto se detalla los costes de los recursos empleados en la implementación del proyecto.

Los recursos se dividen en tres bloques:

- Software: todos los programas utilizados en el proyecto
- Hardware: todos los elementos físicos que componen el sistema
- Recursos humanos: las personas que han colaborado en la implementación del proyecto

### 8.2. Hardware:

Como se ha mencionado anteriormente, en el proyecto se ha utilizado:

- Ordenador personal. En el proyecto se ha utilizado un ordenador portátil, por lo que no se incluye en el presupuesto ni una pantalla ni teclado.
- Ratón
- PLC
- Módulo de E/S
- Pantalla táctil
- Sensores
- Cableado con conectores

Los precios de cada componente se detallan en la tabla siguiente:

Concepto	Cantidad(U)	Precio/U (€/U)	Precio total (€)
<b>Ordenador personal</b>	1	450	450
<b>Ratón</b>	1	9,9	9,9
<b>PLC</b>	1	701,2	701,2
<b>Módulo E/S</b>	16 E	1	143,52
	8 E	1	116,8
	16 S	1	168,16
	8 S	1	168,16
<b>Pantalla táctil</b>	1	487,5	487,5
<b>Sensores</b>	18	43,05	774,9
<b>Cableado con conectores</b>	18	10,41	187,38
<b>TOTAL (€):</b>			<b>3207,52 €</b>

T.16 Costes de Hardware



### 8.3. Software:

En el proyecto actual se ha utilizado el siguiente software:

- CX-Programmer
- CX-Server
- NB-Designer(gratuito)
- Excel

En este paquete también se incluiría el sistema operativo, pero su precio va incluido en el ordenador adquirido.

La empresa ya disponía parte del software utilizado en el proyecto, por lo que éstos aparecen con coste "0" en el presupuesto.

El gasto detallado relacionado a cada programa viene reflejado en la tabla siguiente:

Concepto	Cantidad(U)	Precio/U (€/U)	Precio total (€)
CX- Programmer	1	0	0
Excel	1	0	0
CX-Server	1	427	427
NB-Designer	1	0	0
TOTAL (€):			<b>427 €</b>

*T.17 Coste de Software*

### 8.4. Recursos humanos

Este apartado del presupuesto hace referencia al salario de las personas implicadas en el desarrollo del proyecto.

En el proyecto actual interviene un recurso humano, el ingeniero técnico que a la vez cumple el papel de proyectista y desarrollador del proyecto. El ingeniero técnico tiene un salario de 25 € a la hora, pactado por los sindicatos.

Para realizar el presupuesto del trabajo realizado por el ingeniero técnico, se ha separado las tareas realizadas según la siguiente tabla:

Tarea	Horas invertidas (h)	Precio/hora (€/h)	Precio total (€)
Estudio previo	15	25	375
Diseño	40	25	1000
Implementación	50	25	1250
Comprobación	20	25	500
Montaje	15	25	375
Comprobación Montaje	5	25	125
Total horas (h):			<b>145 h</b>
Total (€):			<b>3625 €</b>

T.18 Coste total de la mano de obra

#### 8.5. Presupuesto total

A continuación, se resume el coste total del proyecto, teniendo en cuenta los costes parciales reflejados en apartados anteriores:

Concepto	Precio (€)
Hardware	3207,52
Software	427
Recursos humanos	3625
Total (€):	<b>7259,52 €</b>

T.19 Coste total del proyecto



## 9. Conclusiones

Al finalizar el proyecto, se puede afirmar el cumplimiento satisfactorio de los objetivos establecidos. Se ha creado un sistema automatizado de conteo de la producción eficaz con un coste muy reducido. Por otro lado, el proyecto ha servido tanto para la adquisición de conocimientos en el ámbito de programación de automatismos industriales, como para la comprensión del funcionamiento de los mismos.

Los conocimientos adquiridos durante la realización del proyecto son de gran valor para el proyectista, ya que la automática tiene cada vez una mayor presencia en el ámbito industrial. Los lenguajes de programación utilizados son lenguajes estandarizados empleados por la gran mayoría de autómatas industriales, por lo que su conocimiento y la agilidad de uso de las herramientas proporcionadas para su empleo son de gran importancia para las empresas del sector.

Por ello, como conclusión final, se puede afirmar que la realización del proyecto ha servido para la formación del proyectista como ingeniero y en el ámbito de la automatización industrial.

Por último, me gustaría agradecer a **D. Carlos Calvo Errando**, mi tutor en la empresa CALVO IZQUIERDO SL en la que se ha realizado el proyecto, tanto por sus consejos para la mejora del proyecto, como por aportar toda la información necesaria sobre la situación en fábrica para el correcto desarrollo e instalación del proyecto. Así mismo, agradecer a **D. Ricardo Pizá Fernández**, mi tutor del Trabajo Final de Grado, por su ayuda en la mejora del proyecto y la orientación dada en momentos de duda a lo largo de la realización del mismo. También quisiera agradecer a mis familiares y amigos que me han apoyado en todo momento, dándome la confianza y fuerzas para seguir mejorando.



## 10. Bibliografía

- CX-One manual de introducción:  
[https://assets.omron.eu/downloads/manual/es/r135\\_cx-one\\_instruction\\_manual\\_es.pdf](https://assets.omron.eu/downloads/manual/es/r135_cx-one_instruction_manual_es.pdf)
- NB- Designer manual de usuario:  
<http://www.miel.si/wp-content/uploads/2013/09/NB-Designer-manual.pdf>
- CX-Lite Manual de usuario:  
[http://www.omronkft.hu/nostree/pdfs/software/cx-server\\_light/cx-s-lite\\_net\\_doc.pdf](http://www.omronkft.hu/nostree/pdfs/software/cx-server_light/cx-s-lite_net_doc.pdf)
- Introducción a VBA en Excel 2010:  
[https://msdn.microsoft.com/es-es/library/office/ee814737\(v=office.14\).aspx](https://msdn.microsoft.com/es-es/library/office/ee814737(v=office.14).aspx)



# **Anexos**





# Índice de Anexos:

<u>Anexo I Pruebas de laboratorio</u> .....	5
<u>Anexo II Macros de Excel VBA</u> .....	9



# **Anexo I**

# **Pruebas de**

# **laboratorio**

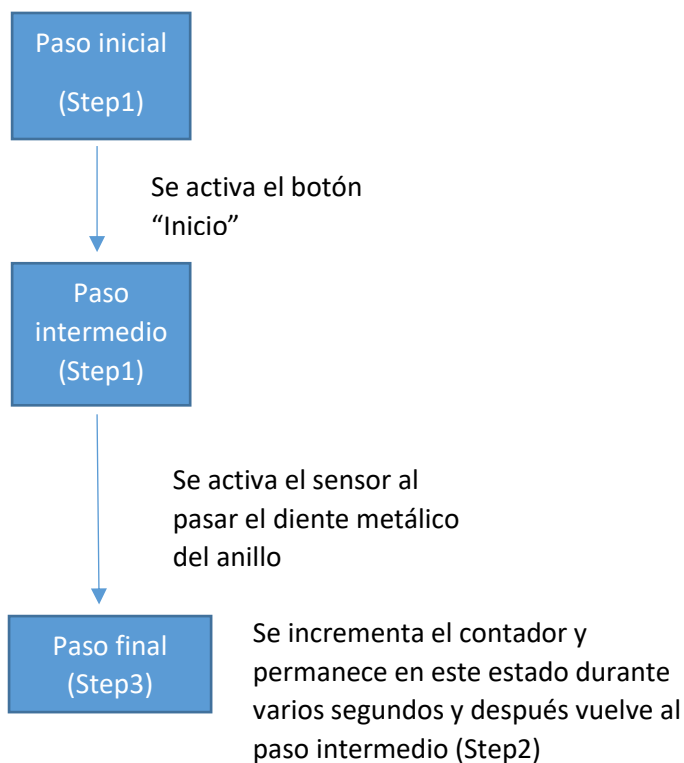


## Prueba de sensor

En primer lugar, se ha diseñado un soporte para acoplar el sensor inductivo en la máquina, así mismo se ha diseñado un anillo con diente metálico para convertir el sensor en un Encoder de coste bajo.

Ambas piezas se han impreso mediante una impresora 3D para las pruebas de funcionamiento.

Para comprobar el funcionamiento del sensor se ha diseñado un GRAFCET sencillo en Cx-Programmer. El programa consiste en un GRAFCET de tres estados en el que al pulsar el botón de “inicio” se pasa a un estado intermedio donde se espera la señal del sensor. Al activar el sensor, el programa pasa a un estado con temporizador y suma uno en el contador asociado.



Con el mismo funcionamiento del programa se comprueba también el funcionamiento del PLC.

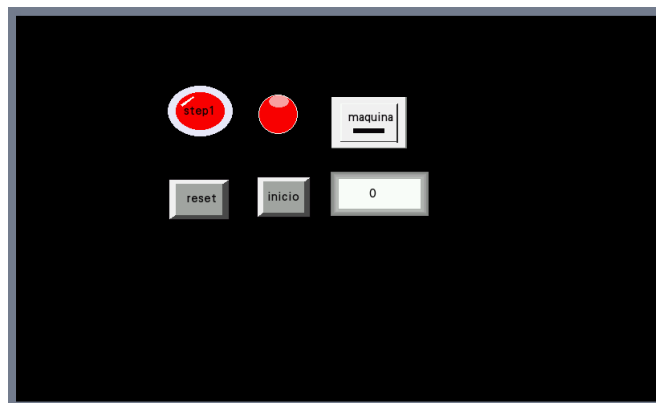
## Prueba de pantalla táctil

Para probar el funcionamiento adecuado de la pantalla se ha programado dos pantallas sencillas con elementos básicos.

### Pantalla 1:

En esta pantalla se dispone de:

- Una bombilla que indica la activación del estado inicial de la “máquina”
- Una bombilla que indica que la máquina está en estado de espera después de la activación del sensor
- Botón “Inicio” que simula la puesta en marcha de la “máquina”
- Botón “reset” que devuelve la máquina al estado inicial y reinicia el contador
- Botón “Máquina” que sirve para cambiar a la pantalla siguiente (*Pantalla 2*)
- Contador numérico



F.27 Pantalla 1

### Pantalla 2:

En esta pantalla se dispone de:

- Contador numérico
- Bombilla indicadora del estado del sensor
- Botón “reset” que sirve para reiniciar el contador y volver la “máquina” al estado inicial
- Botón “volver” para volver a la pantalla inicial (*Pantalla 1*)

# **Anexo II**

# **Macros de**

# **Excel VBA**





## Hoja1: Macro de recopilación de datos

Esta Macro es la encargada de enlazar el servidor OPC al fichero Excel, almacenando la información de lectura de los sensores.

A continuación, se describe el programa de forma más detallada:

- **Cabecera:** En primer lugar, se escribe la cabecera que consiste en la primera fila del Excel en la que aparece: "Contador", "Nº Cabezas", "Diámetro", "Producción", seguido de la fecha y hora en el instante actual en formato DD/MM/AA HH/mm

```
If Cells(1, 2) = 0 Then
    Cells(1, 2) = "Contador"
    Cells(1, 3) = "NºCabezas"
    Cells(1, 4) = "Diámetro"
    Cells(1, 5) = "Producción"
End If
Cells(1, 6) = Comms1.Value("dia_actual")
Cells(1, 7) = Comms1.Value("mes_actual")
Cells(1, 8) = Comms1.Value("año_actual")
Cells(1, 9) = Comms1.Value("hora_actual")
Cells(1, 10) = Comms1.Value("min_actual")
```

### F.1 Script de cabecera

Como se puede observar, parte de la cabecera se escribe solo la primera vez que se escriba en el fichero.

- **Búsqueda de posición de escritura:** El programa compara la fecha y la hora actual con el registro del fichero para localizar la fila en la que debe escribir.

```
' Busca el final del dia , 21 por 3 jornadas = 63
While Cells(3 + n, 1) <> 0
    n = n + 63
Wend
If n <> 0 Then
    m = n - 63
End If

' Si la fecha cuanda y la hora concuerdan entonces determiina posicion
If Cells(3 + m, 2) = Cells(1, 6) And Cells(3 + m, 3) = Cells(1, 7) And Cells(3 + m, 4) = Cells(1, 8) Then
    n = m
ElseIf Cells(1, 9) < 6 Then
    n = m
End If
|
' va a la jornada del dia que toca
If Cells(1, 9) > 6 And Cells(1, 9) < 14 Then
    n = n
Cells(3 + n, 1) = "jornada1"
ElseIf Cells(1, 9) > 14 And Cells(1, 9) < 22 Then
    n = n + 21
Cells(3 + n, 1) = "Jornada2"
ElseIf Cells(1, 9) > 22 And Cells(1, 9) < 6 Then
    n = n + 42
Cells(3 + n, 1) = "Jornada3"
End If
```

### F.2 Script de búsqueda de la fila de escritura

El script está preparado de forma que escribe las 3 jornadas independientemente de las jornadas que se haya tenido durante el día en cuestión. Esto viene impuesto por la petición de la gerencia de la empresa en la que se ha realizado el proyecto.

- Información de las máquinas: Se escribe la información de las máquinas en sus columnas correspondientes (nº de máquina, diámetro de eje, nº de cabezas de la máquina).

```

If Cells(4 + n, 1) = 0 Then
Cells(4 + n, 1) = "maq 1"
Cells(5 + n, 1) = "maq 2"
Cells(6 + n, 1) = "maq 3"
Cells(7 + n, 1) = "maq 4"
Cells(8 + n, 1) = "maq 5"
Cells(9 + n, 1) = "maq 6"
Cells(10 + n, 1) = "maq 7"
Cells(11 + n, 1) = "maq 8"
Cells(12 + n, 1) = "maq 9"
Cells(13 + n, 1) = "maq 10"
Cells(14 + n, 1) = "maq 11"
Cells(15 + n, 1) = "maq 12"
Cells(16 + n, 1) = "maq 13"
Cells(17 + n, 1) = "maq 14"
Cells(18 + n, 1) = "maq 15"
Cells(19 + n, 1) = "maq 16"
Cells(20 + n, 1) = "maq 17"
Cells(21 + n, 1) = "maq 18"
End If

```

**F.3** Script de Nº de máquinas

```

If Cells(4 + n, 3) = 0 Then
Cells(4 + n, 3) = 2
Cells(5 + n, 3) = 2
Cells(6 + n, 3) = 2
Cells(7 + n, 3) = 2
Cells(8 + n, 3) = 2
Cells(9 + n, 3) = 4
Cells(10 + n, 3) = 4
Cells(11 + n, 3) = 3
Cells(12 + n, 3) = 2
Cells(13 + n, 3) = 2
Cells(14 + n, 3) = 2
Cells(15 + n, 3) = 2
Cells(16 + n, 3) = 2
Cells(17 + n, 3) = 2
Cells(18 + n, 3) = 2
Cells(19 + n, 3) = 2
Cells(20 + n, 3) = 4
Cells(21 + n, 3) = 4
End If

```

**F.4** Script de Nº de cabezas

```

If Cells(4 + n, 4) = 0 Then
Cells(4 + n, 4) = 5.25
Cells(5 + n, 4) = 5.25
Cells(6 + n, 4) = 5.14
Cells(7 + n, 4) = 5.25
Cells(8 + n, 4) = 5.127
Cells(9 + n, 4) = 5.8
Cells(10 + n, 4) = 5.8
Cells(11 + n, 4) = 1
Cells(12 + n, 4) = 5.8
Cells(13 + n, 4) = 5.8
Cells(14 + n, 4) = 5.8
Cells(15 + n, 4) = 6
Cells(16 + n, 4) = 5.75
Cells(17 + n, 4) = 5.75
Cells(18 + n, 4) = 5.9
Cells(19 + n, 4) = 8
Cells(20 + n, 4) = 5.8
Cells(21 + n, 4) = 5.8
End If

```

**F.5** Script de diámetro del eje

- Fecha: Se escribe la fecha en la que se ha tomado los datos, en formato DD/MM/AA.

```

If Cells(3 + n, 2) = 0 Then
Cells(3 + n, 2) = Comms1.Value("dia_actual")
Cells(3 + n, 3) = Comms1.Value("mes_actual")
Cells(3 + n, 4) = Comms1.Value("año_actual")
End If

```

**F.6** Script de escritura de la fecha de toma de datos

La información de las máquinas, tanto como la fecha de toma de datos se escribe una única vez por jornada. Esto se debe a que son valores constantes en una misma jornada y no es necesario actualizar su información. De esta forma se optimiza el programa y se ahorra tiempo de procesamiento.

- Contadores: Se anota el valor de los contadores asociados a cada máquina.

```

If Comms1.Value("maq1") > Cells(4 + n, 2) Then
Cells(4 + n, 2) = Comms1.Value("maq1")
End If

```

**F.7** Script de escritura del valor del contador de la máquina 1

En F.7, "maq1" es la variable del servidor OPC asociada al contador de la máquina 1. El script está hecho de forma que actualiza el valor de la tabla únicamente cuando hay un incremento del contador. Esto se hace para asegurar que, en caso de reiniciarse los contadores antes del tiempo previsto, no se pierda la información de la producción de la jornada. El script se repite para cada una de las 18 máquinas.

- Hora de inicio de jornada: Al iniciar la jornada se reinician los contadores y se pasa a escribir en nuevas filas. Por ello y para evitar tener paradas y marchas en horas fuera del rango de la jornada, se escribe la hora de inicio de la jornada como hora de parada en la columna correspondiente.

```
If Cells(3 + n, 15) = 0 Then ' If 03 = 0 then
Cells(3 + n, 15) = "Hora Parada"
Cells(3 + n, 16) = "Min parada"
For i = 1 To 18
Cells(3 + n + i, 15) = Cells(1, 9)
Cells(3 + n + i, 16) = 0
Next i
```

*F.8 Script de escritura de la hora de inicio de jornada*

Cabe la posibilidad de que una máquina no pare en ningún momento entre las tres jornadas, por lo que se comprueba el estado de la máquina y en caso de que la máquina estuviera en funcionamiento se escribe una hora de marcha igual que la hora de inicio de jornada, evitando así datos erróneos de rendimiento.

```
If Comms1.Value("EstadoMaq1") = False Then Cells(3 + n + 1, 17) = Cells(1, 9)
Cell(3 + n + 1, 18) = 0
End If
```

*F.9 Script de comprobación del estado de la máquina 1*

En F.9, "EstadoMaq1" es la variable en el servidor OPC asociada al estado de la máquina 1. El script se repite para cada una de las 18 máquinas.

- Hora de parada: Cuando se produce una parada en la máquina se registra la hora de parada. El script descrito en F.10 es la parte del Macro que se encarga de anotarlo en la última columna libre en la parte del registro de paradas/marchas. Funciona de forma que anota las paradas consecutivas en columnas consecutivas, comprobando que la hora almacenada sea mayor que la hora de la última puesta en marcha.

```
While Cells(4 + n, 16 + a_p) <> 0 Or Cells(4 + n, 15 + a_p) <> 0
a_p = a_p + 4
Wend
If a_p <> 0 Then
a_p_1 = a_p - 4
ElseIf a_p = 0 Then
a_p_1 = a_p
End If
i = Cells(4 + n, 16 + a_p_1)
j = Cells(4 + n, 18 + a_p_1)
k = Cells(4 + n, 17 + a_p_1)
If i <> Comms1.Value("min_paro_1") And (Comms1.Value("min_paro_1") >= j OR Comms1.Value("hora_paro_1")>k) And k <> 0 Then
While Cells(4 + n, 16 + a_p) <> 0 Or Cells(4 + n, 15 + a_p) <> 0
a_p = a_p + 4
Wend
Cells(3 + n, 15 + a_p) = "Hora paro"
Cells(3 + n, 16 + a_p) = "Min paro"
Cells(4 + n, 15 + a_p) = Comms1.Value("hora_paro_1")
Cells(4 + n, 16 + a_p) = Comms1.Value("min_paro_1")
End If
```

*F.10 Script de escritura de la hora de parada de la máquina 1*

En F.10, “min\_paro\_1” y “hora\_paro\_1” son las variables del servidor OPC asociadas a las direcciones de memoria en las que se almacenan los minutos y las horas de parada respectivamente de la máquina 1. El script se repite para cada una de las 18 máquinas.

- Hora de puesta en marcha: Al ponerse en marcha la máquina se registra la hora de puesta en marcha y se anota de forma similar a la “*Hora de parada*”.

```

While Cells(4 + n, 18 + a_i) <> 0 Or Cells(4 + n, 17 + a_i) <> 0
a_i = a_i + 4
Wend
If a_i <> 0 Then
a_i_1 = a_i - 4
ElseIf a_i = 0 Then
a_i_1 = a_i
End If
i = Cells(4 + n, 18 + a_i_1)
j = Cells(4 + n, 16 + a_i)
k = Cells(4 + n, 15 + a_i)
If i <> Comms1.Value("min_init_1") And (Comms1.Value("min_init_1") >= j OR Comms1.Value("hora_init_1")>k) And k <> 0 Then
While Cells(4 + n, 18 + a_i) <> 0 Or Cells(4 + n, 17 + a_i) <> 0
a_i = a_i + 4
Wend
Cells(3 + n, 17 + a_i) = "Hora marcha"
Cells(3 + n, 18 + a_i) = "Min marcha"
Cells(4 + n, 17 + a_i) = Comms1.Value("hora_init_1")
Cells(4 + n, 18 + a_i) = Comms1.Value("min_init_1")
End If

```

**F.11** Script de escritura de la hora de puesta en marcha de la máquina 1

En F.11, “min\_init\_1” y “hora\_init\_1” son las variables del servidor OPC asociadas a las direcciones de memoria en las que se almacenan los minutos y las horas de puesta en marcha respectivamente de la máquina 1. El script se repite para cada una de las 18 máquinas.

## Hoja2: Macro de procesamiento de datos

Esta Macro es la encargada de procesar los datos de la “*Hoja1*” y reescribirlos de forma que facilite la lectura de la información, con información añadida.

El primer paso es copiar los datos de lectura de las máquinas correspondientes a las primeras 5 columnas de la “*Hoja1*” para su posterior procesamiento. A continuación, se escribe la duración de cada jornada y se procesa los datos de “*Hora de parada*” y “*Hora de marcha*” de forma que queden en formato “HH:mm” para poder sumar los tiempos de parada de forma directa y restarlos a la duración total de la jornada.

La duración de jornada es de 8 horas de forma predeterminada, aunque se puede cambiar de forma manual en la casilla del Excel correspondiente.

Como último, se calcula el rendimiento de la máquina y las paradas totales durante la jornada.

```

Sheets("Hoja1").Select
Sheets("Hoja1").Range("A:E").Select
Selection.Copy
Sheets("Hoja2").Select
Sheets("Hoja2").Range("A:E").Select
ActiveSheet.Paste
Application.CutCopyMode = False
    
```

*F.12 Script de copia de datos de Hoja1*

```

If Cells(3 + b + i, 6) = 0 Then
Cells(3 + b + i, 6) = 8 & ":" & 0
End If
C = 0
l = 0
n = 0
r = 0
k = 0
While Sheets("Hoja1").Cells(3 + b + i, 16 + l) <> 0
Sheets("Hoja2").Cells(3 + b, 15 + n) = "Hora Paro"
Sheets("Hoja2").Cells(3 + b + i, 15 + n) = Sheets("Hoja1").Cells(3 + b + i, 15 + l) & ":" & Sheets("Hoja1").Cells(3 + b + i, 16 + l)
l = l + 4
n = n + 2
C = C + 1
Wend
Cells(3 + b + i, 9) = C
l = 0
n = 0
While Sheets("Hoja1").Cells(3 + b + i, 18 + l) <> 0
Sheets("Hoja2").Cells(3 + b, 16 + n) = "Hora marcha"
Sheets("Hoja2").Cells(3 + b + i, 16 + n) = Sheets("Hoja1").Cells(3 + b + i, 17 + l) & ":" & Sheets("Hoja1").Cells(3 + b + i, 18 + l)
If Sheets("Hoja2").Cells(3 + b + i, 15 + n) <> 0 Then
k = Sheets("Hoja2").Cells(3 + b + i, 16 + n) - Sheets("Hoja2").Cells(3 + b + i, 15 + n)
r = r + k
End If
l = l + 4
n = n + 2
Wend
Cells(3 + b + i, 7) = Cells(3 + b + i, 6) - r
Cells(3 + b + i, 8) = (1 - (Cells(3 + b + i, 6) - Cells(3 + b + i, 7)) / Cells(3 + b + i, 6)) * 100
Cells(22 + b, 5) = Cells(22 + b, 5) + Cells(3 + b + i, 5)
Next i
End If
    
```

*F.13 Script de procesamiento de las horas de parada y de puesta en marcha*



# Planos





