

# UNIVERSIDAD POLITÉCNICA DE VALENCIA

**Escuela Técnica Superior De Ingeniería Del Diseño**

Master Universitario en Diseño y Fabricación Integrada Asistidos por Computador



## CONTROL Y OPTIMIZACIÓN DE LA PRODUCCION DE UN PROCESO INDUSTRIAL DE FABRICACIÓN DE PIEZAS.

**TESIS DE MASTER**

Autor:

**Lucia Escudero Andaluz**

Director:

**Antonio Correcher Salvador**

### *Agradecimientos*

*Este proyecto no habría sido posible sin la ayuda de los tutores, profesores y maestros que hasta la fecha han hecho presencia en mi formación, el apoyo incondicional de familiares, amigos y compañeros, a ellos van dirigidas estas palabras.*

*Declaración*

*Declaro que el trabajo aquí presentado es de mi autoría; que no ha sido previamente presentado por ningún grado o calificación profesional; y que he consultado todas las referencias bibliográficas aquí citadas.*

*Lucía Escudero Andaluz*

*Valencia, 28 de Septiembre de 2012*

# Índice

---

<b>1. Introducción.....</b>	<b>7</b>
1.1. <u>¿Qué es automatizar?</u> .....	7
1.2. <u>Niveles de Automatización Industrial</u> .....	8
<b>2. Objetivos.....</b>	<b>15</b>
2.1. <u>Objetivos generales</u> .....	15
2.2. <u>Objetivos específicos</u> .....	16
<b>3. Diseño del Sistema.....</b>	<b>19</b>
3.1. <u>Descripción de la planta</u> .....	19
3.1.1. Descripción del proceso de fabricación de cada pieza.....	21
3.1.2. Descripción de los componentes de la planta.....	23
3.1.2.1. Desbobinadora. ....	23
3.1.2.2. Corte.....	23
3.1.2.3. Taladro.....	24
3.1.2.4. Centro de Pintura.....	24
3.1.2.5. Cintas transportadoras.....	24
3.1.2.6. Zonas de Intercambios.....	25
3.1.2.7. Sensores.....	26
3.2. <u>Simulador</u> .....	28
3.2.1. Estructura del Simulador.....	29
3.2.2. Redes de Petri.....	30
3.2.3. Redes de Petri Coloreadas.....	32
3.2.4. Redes de Petri Coloreadas y Modelado del Simulador.....	33
3.2.5. Interpretación del Modelo en el Simulador.....	37
3.2.6. Modelo de gestión de estados.....	40
3.2.7. Comunicación Simulador-PLC.....	40
3.2.8. Sistema Scada.....	47
<b>4. Sistema de Control.....</b>	<b>49</b>
4.1. <u>Funcionamiento del Autómata</u> .....	51
4.2. <u>Lenguaje de programación: El Grafcet</u> .....	54

4.3.	<u>Especificaciones de Funcionamiento de la planta</u> .....	57
4.3.1.	Especificaciones Generales.....	57
4.3.2.	Especificaciones para cada máquina.....	58
4.4.	<u>Modelo de Subprocesos</u> .....	60
4.5.	<u>Acciones y Sensores en la Planta</u> .....	62
4.6.	<u>Descripción del Sistema de Grafcet</u> .....	65
4.6.1.	Grafcet de producción.....	65
4.6.1.1.	Puesta en marcha.....	65
4.6.1.2.	Bobina.....	65
4.6.1.3.	Corte.....	66
4.6.1.4.	Taladro.....	69
4.6.1.5.	Centro de Pintura.....	71
4.6.1.6.	Cintas.....	73
4.6.1.7.	Intercambiadores.....	78
4.6.2.	Sistema de Prioridades.....	80
4.6.2.1.	Prioridades para las maquinas dentro de cada línea.....	80
4.6.2.2.	Prioridades entre los intercambiadores y las cintas de líneas.....	81
4.7.	<u>Flancos de bajada y restricciones con las acciones como transiciones</u> .....	81
4.8.	<u>Programación con SYSWIN</u> .....	82
<b>5.</b>	<b>Supervisión y Optimización de Fallos</b> .....	<b>85</b>
5.1.	<u>Explicación del problema y variables</u> .....	86
5.1.1.	¿Qué fallos pueden ocurrir en la planta?.....	87
5.1.2.	¿Qué variables interviene en el problema?.....	87
5.1.3.	Posibles soluciones. El Camino Optimo.....	88
5.2.	<u>Búsqueda del algoritmo</u> .....	92
5.2.1.	Tipos de algoritmos.....	93
5.2.2.	Algoritmo de Bellman- Ford.....	96
5.3.	<u>Programa para el cálculo del camino óptimo</u> .....	101
5.3.1.	Programa de Fallos.....	101
5.3.2.	Programa de Variables de Tiempos y Matriz de Decisión...102	
5.3.3.	Programa para el cálculo del camino óptimo.....	106
5.3.3.1.	Programa principal.....	106
5.3.3.2.	Subprogramas.....	109

- Prioridad de Línea.
- Gráfica de simulación de la planta.
- Algoritmo de Bellman-Ford.
- Calculo de la distancia total y de los puntos por donde pasa el camino.
- Matriz de tiempos mínimos.
- Representación gráfica camino para las piezas.

<b>6. Ejemplo de aplicación.....</b>	<b>123</b>
6.1. <u>Diseño y Control del Sistema</u> .....	123
6.2. <u>Secuencia de Fallos</u> .....	128
<b>7. Conclusión y trabajos futuros.....</b>	<b>139</b>
<b>8. Anexo: Grafcet de producción.....</b>	<b>143</b>
8.1. <u>Línea 1</u> .....	143
8.2. <u>Línea 2</u> .....	148
8.3. <u>Línea 3</u> .....	153
8.4. <u>Cintas Intercambio</u> .....	158
<b>9. Bibliografía.....</b>	<b>160</b>



# Introducción

---

## 1.1. ¿Qué es automatizar?

Una definición de automatización puede ser:

*“Automatizar consiste en controlar un proceso mediante el uso de sistemas o elementos computarizados o electromecánicos, y así disminuir la intervención humana en dicho proceso.”*

Gracias a ella se obtienen ventajas como:

- Reemplazo de operadores humanos en tareas repetitivas o de alto riesgo.
- Reemplazo de operador humano en tareas que están fuera del alcance de sus capacidades como levantar cargas pesadas, trabajos en ambientes extremos o tareas que necesiten manejo de una alta precisión
- Incremento de la producción. Al mantener la línea de producción automatizada, las demoras del proceso son mínimas, no hay agotamiento o desconcentración en las tareas repetitivas, el tiempo de ejecución se disminuye considerablemente según el proceso.
- Reducción de costos, puesto que se racionaliza el trabajo, se reduce el tiempo y dinero dedicado al mantenimiento.
- Existe una reducción en los tiempos de procesamiento de información.
- Flexibilidad para adaptarse a nuevos productos.
- Se obtiene un conocimiento más detallado del proceso, mediante la recopilación de información y datos estadísticos del proceso.
- Se obtiene un mejor conocimiento del funcionamiento de los equipos y máquinas que intervienen en el proceso.
- Se asegura una mejora en la calidad del trabajo del operador y en el desarrollo del proceso, aunque dependerá de la eficiencia del sistema implementado.
- Factibilidad técnica en procesos y en operación de equipos.
- Factibilidad para la implementación de funciones de análisis, optimización y autodiagnóstico.

## 1. Introducción

---

- Aumento en el rendimiento de los equipos y facilidad para incorporar nuevos equipos y sistemas de información.
- Disminución de la contaminación y daño ambiental.
- Racionalización y uso eficiente de la energía y la materia prima.
- Aumento en la seguridad de las instalaciones y la protección a los trabajadores.

La automatización industrial permite decidir el grado de autonomía de todo el proceso, desde controlar una simple maquina a planificar todos los pedidos para que el rendimiento y la eficiencia de la planta sea máxima. Aunque cuanto mayor sea el control del sistema más difícil y compleja será la automatización.

### 1.2. Niveles de Automatización Industrial

La automatización de un proceso industrial se puede clasificar en niveles según el grado de autonomía que se quiera dar a la planta. El más elemental consiste en gobernar operaciones sencillas a nivel de maquinaria como pueden ser tareas de vigilancia de tiempos muertos, posicionamiento de piezas o funciones de seguridad. Si en vez de automatizar una sola máquina, se automatizan varias a la vez, se asciende al nivel intermedio.

El siguiente nivel, llamado Nivel 3, se caracteriza por ser un proceso completo, donde además del control elemental de las máquinas, intervienen aspectos como la supervisión y optimización del proceso, gestión de mantenimiento o seguimiento de la producción. Existen varios tipos de control para este nivel:

- a. Control Centralizado**, un ordenador gobierna todos los procesos. No tiene en cuenta temas como seguridad o mantenimiento, solo se centra en que las maquinas funcionen.

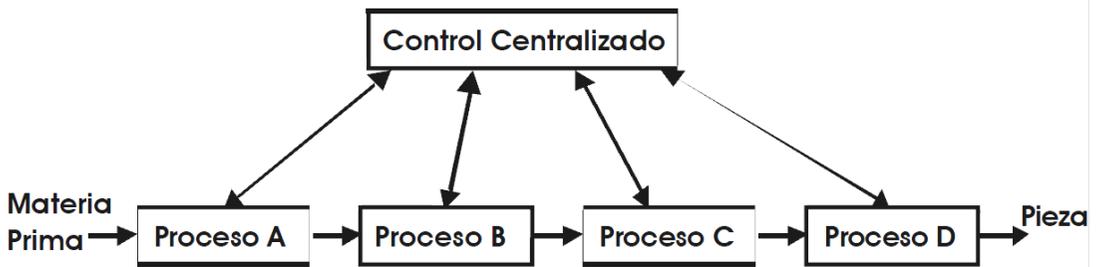


Ilustración 1.1. Control Centralizado

- b. **Control Multicapa**, variedad de control centralizado haciendo dos niveles jerárquicos de control.

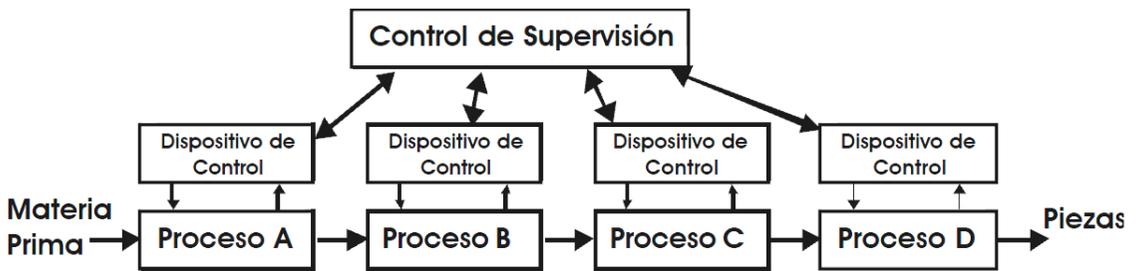


Ilustración 1.2. Control Multicapa

- c. **Control Jerarquizado**, control ampliado del anterior donde existe una jerarquía de control, teniendo en cuenta las tareas de planificación del proceso o seguridad.
- d. **Control Distribuido**, se crean varias unidades de control conectadas entre sí, así si falla una, se puede derivar a otra. Aumenta la seguridad y permite seguir con la producción cuando ocurre algún fallo. El proceso es más lento.

## 1. Introducción

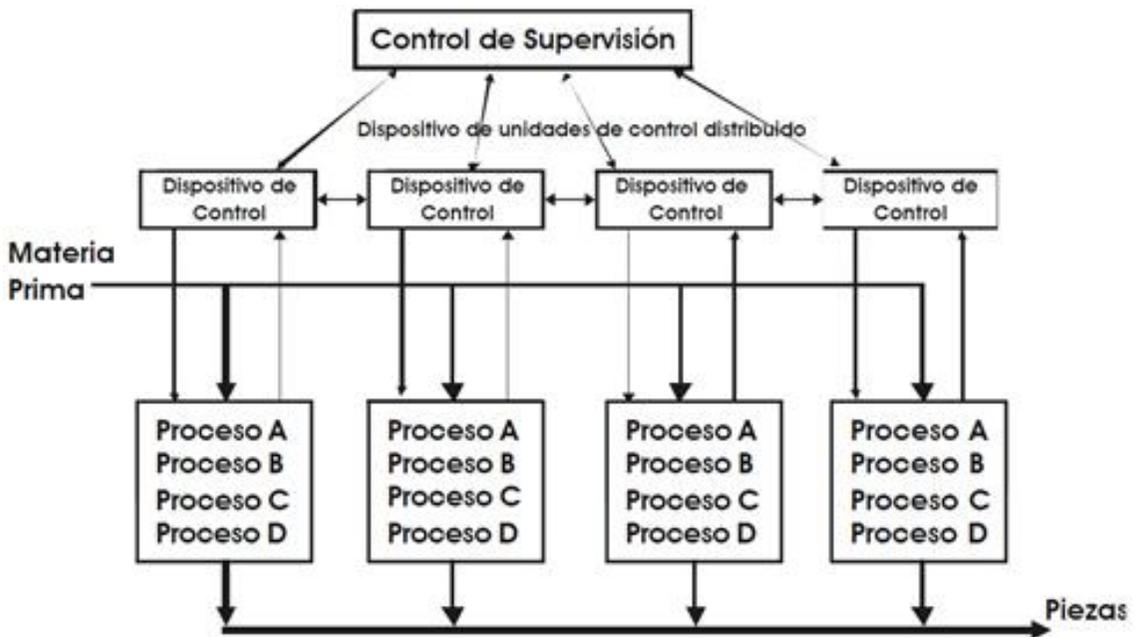


Ilustración 1.3. Control Distribuido

El último nivel de automatización es el Nivel 4 o CIM (Computer Integrated Manufacturing). Este nivel busca dar autonomía total al proceso de producción, automatizando todas las partes que intervienen, desde el control de las máquinas hasta la gestión de pedidos. Los aspectos que se encarga este tipo son:

- Planificación la producción.
- Programación y control de las máquinas para que sigan el plan de producción que ha ideado el planificador.
- Gestión de los pedidos.
- Control de salida de los productos.

Con el concepto de CIM se obtienen las siguientes ventajas:

- Disminución de costes.
- Disminución de Stock.
- Reducción de costes de material.
- Incrementación de la productividad.
- Mejora del control de calidad.

El Nivel 4 o CIM corresponde a una estructura piramidal jerarquizada, donde la base representa el nivel más elemental y sencillo de controlar y según se va ascendiendo por la pirámide va aumentando en grado de complejidad.



Ilustración 1.4. Pirámide Jerarquías

### Niveles de la pirámide

→ **Nivel 0:** representa las máquinas que se utilizan y los sensores y actuadores que lo engloban. En la tesina está representado mediante un simulador creado con el software Labview.

→ **Nivel 1:** está formado por dispositivos lógicos de control, autómatas programables. Este nivel se encarga de gobernar a las máquinas que forman parte del nivel 0, haciendo que sigan una secuencia de acciones determinada con el fin de que el proceso de producción se lleve a cabo. Constituye los elementos de mando y control sobre la maquinaria del nivel cero. Proporciona información de actuación directa al nivel cero y del estado del sistema al nivel 2.

→ **Nivel 2:** Nivel de supervisión. Se realizan por medios informáticos las siguientes tareas:

- Monitorización del proceso.
- Toma de decisión de la mejor solución para minimizar el problema o fallo.

## 1. Introducción

---

- Gestión de alarmas.
- Mantenimiento correctivo y preventivo.

Este nivel emite órdenes de ejecución al Nivel 1 y recibe la situación del estado de dicho nivel. Igualmente recibe los programas de producción del Nivel 3 y le informa de las incidencias ocurridas en la planta.

→ **Nivel 3:** Nivel de planificación. En este nivel se realizan las siguientes tareas:

- Planificación de la producción.
- Optimización del proceso.
- Control de inventario.
- Gestión de recursos de fabricación.

El nivel 3 emite los programas hacia el nivel 2, y recibe de éste las incidencias de la planta. Del nivel 4 recibe información de:

1. Número y cantidad de pedidos.
2. Información acerca de las características del pedido.

Y envía información relativa a:

1. Cumplimiento de programas.
2. Costes de operación.
3. Cambios en el proceso.

→ **Nivel 4:** Nivel corporativo. Realiza las tareas de gestor de pedidos. Entre sus tareas están:

- Gestión comercial y marketing
- Planificación estratégica
- Planificación financiera y administrativa
- Gestión de recursos humanos
- Ingeniería de producto
- Ingeniería de proceso
- Gestión de tecnología
- Gestión de sistemas de información
- Investigación y desarrollo

Este nivel emite al Nivel 3 información sobre la situación comercial (pedidos y previsiones), información de ingeniería de producto y de proceso, etc.

En la tesina se va a aplicar la automatización tipo CIM para su desarrollo, comenzando por diseñar la planta, y a continuación ir ascendiendo por los diferentes niveles hasta el de supervisión de fallos.



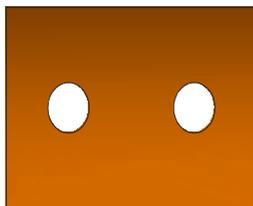
# Objetivos

---

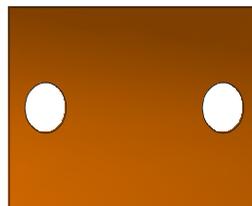
## 2.1. Objetivos Generales

El objetivo general que se persigue en esta tesina es el de crear y automatizar una planta industrial para poder fabricar piezas en serie. El proceso de fabricación tiene que ser lo más eficiente posible, optimizando procesos de producción y tiempos.

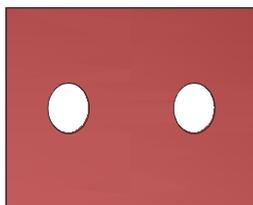
La planta se tiene que diseñar con todos los elementos necesarios para que se fabriquen piezas en serie a partir de bobinas de chapa. Cada pieza debe tener forma rectangular con dos taladros y estar pintadas. Los taladros podrán estar situados en dos posiciones, más cerca o más lejos del centro. Además, la planta debe ofrecer la posibilidad de elegir el color de las piezas entre dos posibles y el tipo de calidad de material se quiere para cada pieza entre tres, de calidad baja, calidad intermedia y calidad alta. Con todas estas opciones, la planta ofrecerá 12 tipos diferentes de productos a elegir para su fabricación en función del tipo de taladro, de calidad del material y del color de la pintura.



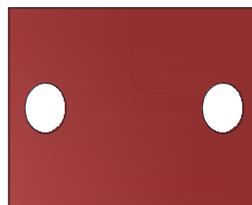
Taladro Tipo 1, Pintura Tipo 1



Taladro Tipo 2, Pintura Tipo 1



Taladro Tipo 2, Pintura Tipo 1



Taladro Tipo 2, Pintura Tipo 2

**Ilustración 2.1. Tipo de Piezas**

### 2.2. Objetivos específicos.

Los objetivos específicos del trabajo consisten en ir diseñando cada uno de los niveles que forman parte del esquema de automatización tipo CIM. Este sistema sigue un esquema jerarquizado en el que se tiene que empezar diseñando los componentes desde la parte inferior de la pirámide hasta arriba.



Ilustración 2.2. Pirámide de Jerarquías

Para esta tesina, el objetivo a cumplir es el diseño de los tres primeros escalones de la pirámide:

1. Diseño de la planta.

El primer objetivo es diseñar la planta utilizando un simulador creado con Labview. La planta tiene que cumplir con todos los requisitos necesarios para que luego el resto de niveles puedan trabajar con ella como si fuera un sistema real. Tiene que estar formada por varias líneas de producción para que se fabriquen piezas con diferente tipo de calidad de material. Cada línea debe tener todas las máquinas que son necesarias para la fabricación de las piezas y colocadas en el orden correcto.

2. Control del sistema

El segundo objetivo es crear el sistema de control para la planta. Para ello primero tiene que estar diseñada la parte del simulador de la planta. Esta parte se tiene que encargar de controlar tanto el movimiento de las máquinas para que cada una realice su trabajo

perfectamente como el proceso para que la secuencia de producción sea la correcta y las piezas se fabriquen sin ningún error.

### 3. Supervisión de fallos y optimización de tiempos.

El tercer objetivo que se va a desarrollar es la creación del sistema de supervisión y optimización de fallos. Para ello tienen que estar listas las partes de diseño de la planta y de control del proceso. La función es dar la mejor opción, el camino que tienen que seguir las piezas cuando aparece una avería en su línea para que el tiempo perdido por el fallo sea el menor posible.

Las especificaciones de diseño hacen que obligatoriamente el proceso de diseño de los niveles lleve una jerarquía de diseño, comenzando con el diseño de la planta, luego el control del proceso y por último la supervisión de fallos.

## 2. Objetivos

---

# *Diseño del Sistema*

---

El diseño de una planta industrial comienza conociendo cuál es el proceso de fabricación del objeto que se quiere fabricar. A partir de esta información, se decide cual son las máquinas que se necesitan en la planta y el orden de ejecución de procesos, así se componen las líneas de producción que van a integrar la planta. El número de líneas va en función de las características del producto o de la estrategia que se quiera seguir en el plan de producción.

En este capítulo se describirán los componentes que forman parte de la planta y las características de funcionamiento del simulador que sustituye a la planta real. La primera parte describe la planta industrial, el proceso productivo que se necesita para la fabricación de las piezas, las líneas que integran la planta y las máquinas que forman parte de cada línea. También se habla de las acciones que tienen que hacer cada máquina.

La segunda parte se centra en el simulador que se ha creado para imitar el comportamiento de una planta real con el ordenador. El simulador está hecho con el software Labview y utiliza un sistema de redes de Petri coloreadas como motor de funcionamiento para que la secuencia del proceso de fabricación se ejecute sin errores. El simulador muestra las líneas de fabricación y las máquinas que integran la planta. Cada máquina realiza los movimientos como si fuera real para ir fabricando las piezas, los movimientos los controla y acciona el autómatas, las maquinas también tienen sensores para la automatización de los movimientos. Para que la simulación parezca más real, en la planta también se podrán simular fallos, cuando se active un fallo la maquina correspondiente se paralizará hasta que se arregle.

## **3.1. Descripción de la planta.**

La planta va a albergar la fabricación de piezas con las siguientes características:

- Piezas rectangulares de chapa.
- Con dos taladros.

### 3. Diseño del Sistema

---

- Pintadas.

El material llega enrollado en bobinas. Hay bobinas de tres tipos según el tipo de calidad del material, se pueden fabricar piezas de calidad 1 que corresponde con la de menor calidad, de calidad 2, calidad intermedia y calidad 3 que representa a la bobina de mayor calidad.



Ilustración 3.1. Tipo de Calidades de Pieza

La planta tiene que dar la posibilidad de elegir entre dos tipos de taladros y dos tipos de colores para dar el acabado a las piezas.

Con todas estas condiciones se decide diseñar la planta de las siguientes características:

- La planta va a estar compuesta por tres líneas de producción que trabajaran en serie donde se fabrican varios tipos de calidad de piezas.
- Cada línea va a estar integrada por:
  1. Una desbobinadora para desenrollar el material de las bobinas.
  2. Una máquina para cortar chapa y hacer las piezas.
  3. Un taladro para hacer los agujeros.
  4. Un centro de pintura para dar el acabado a las piezas.
- Para trasladar piezas de unas máquinas a otras dentro de cada línea se van a utilizar cintas transportadoras.
- La planta va a tener cuatro cintas transportadoras situadas en dos puntos intermedios para poder pasar piezas de unas líneas a otras y así agilizar y aumentar la productividad de la planta.

#### 3.1.1. Descripción del proceso de fabricación de cada pieza.

La fabricación de una pieza comienza cuando la desbobinadora desenrolla la longitud de material necesario. Acto seguido, la cortadora separa las piezas de la bobina, después la cinta 1 lleva la pieza hasta el punto intermedio 1, allí puede continuar por dos caminos, seguir en la misma línea de producción o cambiar de línea a través de los intercambiadores. Si no se cambia de línea, la pieza pasa a la cinta 2, que la lleva hasta la zona de taladrado, siempre que el taladro esté libre. Si se cambia, la cinta que sirve de intercambiador lleva la pieza a la cinta 2 de otra línea de producción y el proceso sigue igual.

En la zona de taladro se hacen dos agujeros a la vez. Los agujeros pueden ser de dos formas diferentes, Modo 1 o Modo 2. Una vez realizados los taladros se pasa la pieza a la cinta 3, la cual transporta la pieza al punto intermedio 2. Al igual que en el 1, la pieza puede cambiar de línea o continuar en la suya. Si continúa en la suya pasa a la cinta 4, y de allí a la zona de pintura si está libre. Cuando la pieza está acabada se pasa a la cinta 5, que la saca de la línea de producción automáticamente, y el proceso queda terminado.

La secuencia de ejecución de fabricación que se tiene que seguir en todo momento en la planta de producción para que la pieza esté bien fabricada es:

**Desenrollar Material → Corte del Material → Taladros → Acabado**

No importa que las piezas cambien de línea, la secuencia tiene que ser la misma. Las máquinas de las tres líneas son iguales. La única diferencia que existe entre las líneas es la calidad del material. Al ser todas máquinas iguales la planta permite que la secuencia de procesos se realice en varias líneas sin existir ningún problema.

Con las condiciones y características que se han descrito, la situación de las líneas y las máquinas en la planta es la siguiente:

### 3. Diseño del Sistema

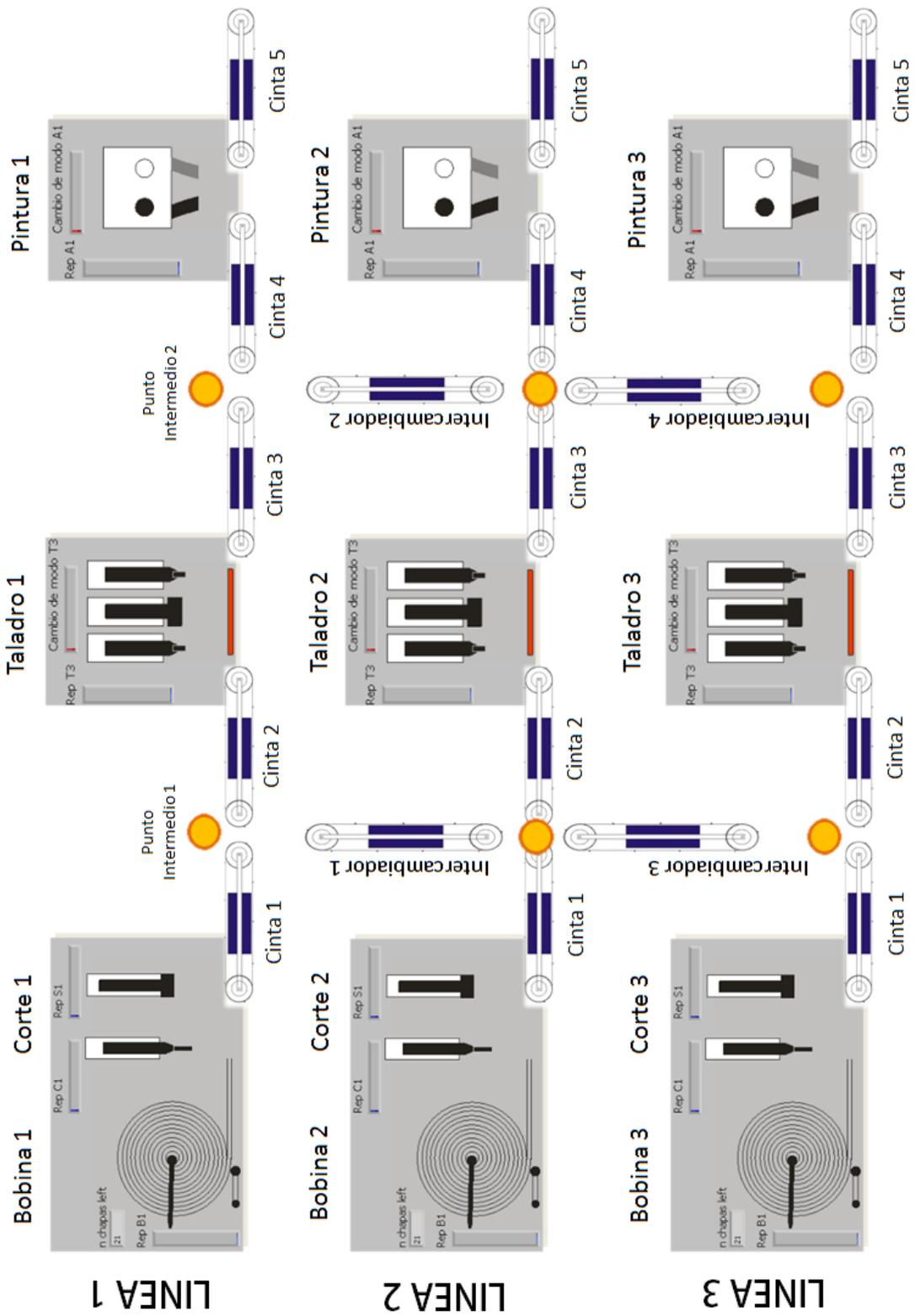


Ilustración 3.2. Distribución Planta Industrial

### 3.1.2. Descripción de los componentes de la planta.

Las máquinas que forman parte de cada línea son similares con el fin de se puedan intercambiar las piezas entre líneas lo más rápido posible y agilizar la producción. Cada línea está formada por una desbobinadora, una máquina de corte de chapa, un taladro y un centro de pintura. Para trasladar las piezas entre maquinas se usan cintas trasportadores.

#### 3.1.2.1. Desbobinadora.

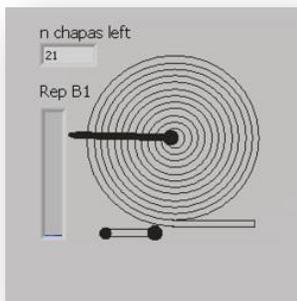


Ilustración 3.3. Desbobinadora

La desbobinadora sujeta la bobina y va desenrollándola según el tiempo que le indica el autómata. En función del tiempo que va desenrollando, la longitud de la pletina es mayor o menor. Cada bobina tiene una capacidad para fabricar 25 piezas.

#### 3.1.2.2. Corte.

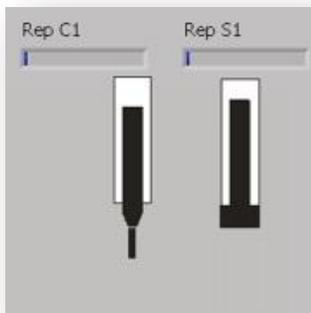


Ilustración 3.4. Máquina de Corte

El corte está formado por una sujeción en forma de cilindro para inmovilizar la pieza y una cizalla. Para realizar el proceso, primero se tiene que haber desenrollado la longitud necesaria de la bobina. Una vez lista la bobina, primero se baja la sujeción, cuando la pieza esta inmóvil, baja la cuchilla hasta que se corta la chapa completamente, se sube la cuchilla, y cuando ha llegado arriba, el cilindro de sujeción sube también para dejar libre la

pieza ya cortada.

#### 3.1.2.3. Taladro

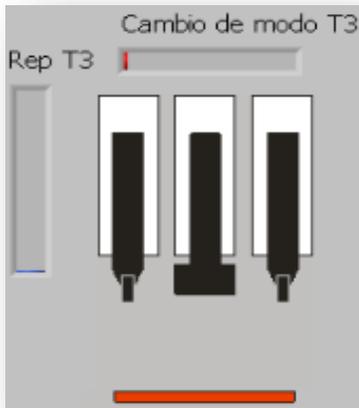


Ilustración 3.5 Taladro

La maquina tiene dos taladros gemelos y una sujeción para piezas. El proceso es parecido al del corte, la diferencia es que bajan a la vez la sujeción y los taladros.

Además el sistema tiene 2 modos de funcionamiento, dependiendo del modo que se escoja hace los agujeros más cerca o más lejos del centro de la pieza. Antes de que las brocas comiencen a bajar para hacer los agujeros, el modo debe estar posicionado.

#### 3.1.2.4. Pintura

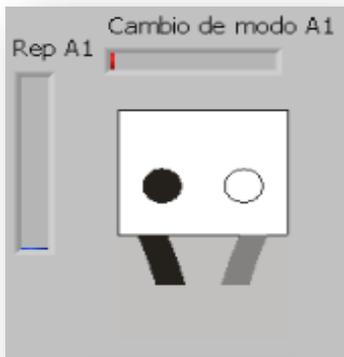


Ilustración 3.6. Centro de Pintura

La pintura da el acabado a la pieza. Puede pintar en dos colores diferentes. La pieza entra en el centro de pintura, dentro se pinta y se seca, cuando está lista sale por la cinta 5.

#### 3.1.2.5. Cintas Transportadoras.

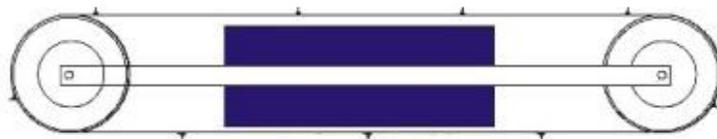


Ilustración 3.7. Cinta Transportadora

Las cintas se utilizan para trasladar las piezas de unas máquinas a otras. Se activan cuando llega una pieza al principio y se detienen cuando llega al final, mientras que no haya pieza para trasladar tienen que estar paradas. En el simulador no están representadas.

Hay dos tipos de cintas, por un lado están las cintas que trasladan piezas dentro de la misma línea de producción. Cada línea tiene 5 cintas distribuidas de la siguiente manera:

- Cinta 1: traslada piezas desde el corte al punto intermedio 1.
- Cinta 2: traslada piezas desde el punto intermedio 1 al taladro.
- Cinta 3: traslada piezas desde el taladro al punto intermedio 2.
- Cinta 4: traslada piezas desde el punto intermedio 2 al centro de pintura.
- Cinta 5: traslada piezas desde el centro de pintura al final de la línea.

Además, también hay otras 4 cintas que sirven para intercambiar las piezas de unas líneas de producción a otras. Los intercambios se pueden ejecutar en los 2 puntos intermedios de cada línea, situados entre la cinta 1 y la cinta 2 para el primer intercambio, y entre la cinta 3 y la cinta 4 para el segundo intercambio.

Estas cintas están diseñadas para que funcionen en ambos sentidos, lo que significa, que la misma cinta se usa para enviar piezas desde la línea 1 a la línea 2 o de la línea 2 a la línea 1.

#### **3.1.2.6. Zona de intercambios.**

El fin de los intercambios es dar más flexibilidad a la planta, de manera que permitan intercambiar piezas entre líneas para:

1. Agilizar la fabricación de las piezas, si sólo se están produciendo piezas de un tipo de calidad, con este sistema se va a poder utilizar las tres líneas disminuyendo el tiempo de producción total y dando uso a unas máquinas que de otra manera estarían apagadas.
2. No parar la producción si alguna máquina se rompe, ya que permite enviar las piezas a las otras líneas mientras se arregla.

### 3. Diseño del Sistema

---

Hay 2 zonas para los intercambios, la zona 1 situada entre la cinta 1 y la cinta 2 y zona 2 situada entre la cinta 3 y la cinta 4 de cada línea. Los intercambios se producen entre las mismas zonas de cada línea, si se envían piezas desde la zona 1 de la línea 1, llegan a la zona 1 de la línea 2.

Es importante que la secuencia de procesos de producción se respete con los intercambios, luego si se envía una pieza desde la cinta 1 porque se acaba de cortar, tiene que llegar a la cinta 2 de la línea de recepción para que se hagan los taladros, no puede llegar a otro punto.

Las zonas de intercambio están divididas en dos partes, por un lado están las piezas que llegan hasta el sensor intermedio, y por otro las piezas que salen desde allí. Las piezas que llegan al punto de intercambio pueden proceder de la misma línea, Cinta 1 o Cinta 3, o de los intercambiadores. Si un intercambio está en proceso, las cintas de línea se tienen que esperar a que acabe el intercambio y la zona intermedia se quede libre.

Cuando la pieza llega al punto intermedio procedente de las cinta de línea también tiene dos opciones, puede continuar en la misma línea, o cambiarse a otra y seguir con el proceso de fabricación en la otra. Se cambie o no, la pieza irá al proceso de producción que le toque en ese momento, si está en la zona intermedia 1, haya intercambio de línea o no, la pieza ira al taladro para que se hagan los agujeros, y si está en la zona intermedia 2 será enviada al acabado.

#### **3.1.2.7. Sensores**

Los sensores hacen un papel fundamental en el control de la planta. Están repartidos por todas las máquinas y cintas. Se activan cuando algún componente de maquina o pieza llega al lugar donde están. Son los encargados de enviar la información al autómatas para que este sepa en cada momento que instrucciones tiene que activar o desactivar.

El sensor de la desbobinadora es un temporizador que marca el tiempo que tiene que estar desenrollando material, también tiene un sensor para indicar cuando se ha terminado una bobina.

El corte tiene 4 sensores, que muestran cuando están arriba o abajo la sujeción y el corte. El taladro sólo tiene 2, para indicar cuando están arriba o

abajo el taladro y la sujeción, en este caso van unidos ambos. La pintura tiene un sensor para indicar cuando la pieza está lista.

Para indicar que una pieza está al final de la cinta 1 o la cinta 3 de cada línea, hay colocados sensores en los puntos intermedios. Estos sensores son muy importantes porque son los encargados de coordinar las piezas para que no se junten cuando llegan de diferentes líneas. Si está encendido este sensor significa que hay una pieza que está esperando que la máquina en la que tiene que entrar termine de hacer el trabajo con otra pieza para que se quede libre. Como consecuencia, ninguna cinta puede enviar piezas a los puntos intermedios mientras el sensor esté activado. Con esto se consigue organizar la planta para que los envíos de piezas se hagan correctamente y no se junten piezas en la misma máquina.

#### 3.2. Simulador.

El sistema de la planta industrial es un simulador creado con Labview que imita a una planta real. Dentro del simulador están representadas las tres líneas de producción con todas las máquinas que forman parte de ellas.

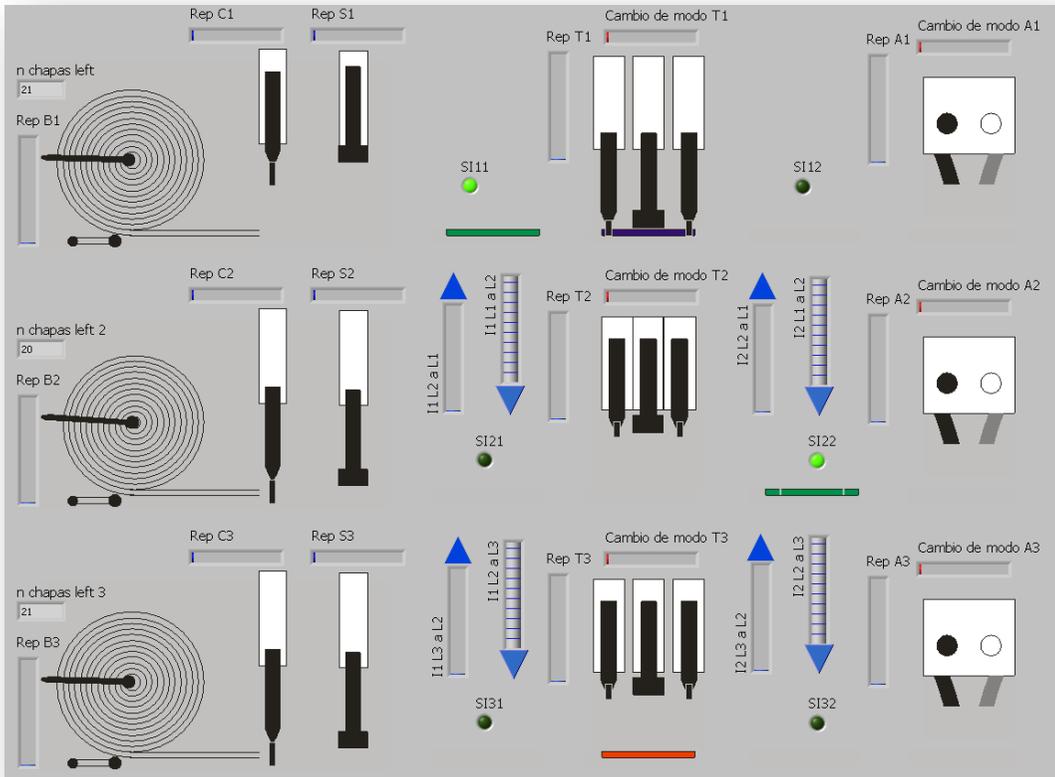


Ilustración 3.8. Simulador

Para que el funcionamiento sea correcto, el simulador tiene que imitar a una planta industrial real. Todas las máquinas tienen que realizar sus acciones como si fueran de verdad, dichas acciones las controlará en el siguiente nivel el autómatas. Se tiene que ir viendo cómo se fabrican las piezas siguiendo la secuencia del proceso y realizando intercambios si son necesarios. El simulador tiene que ser capaz de desenrollar las piezas, realizar el corte, los taladros y por último el acabado sin saltarse ninguna máquina. También tiene que

comportarse como lo haría una maquina cuando se avería, dejando de funcionar y parando la producción en ese punto.

### 3.2.1. Estructura del simulador

El esquema de funcionamiento que sigue el simulador es el siguiente:

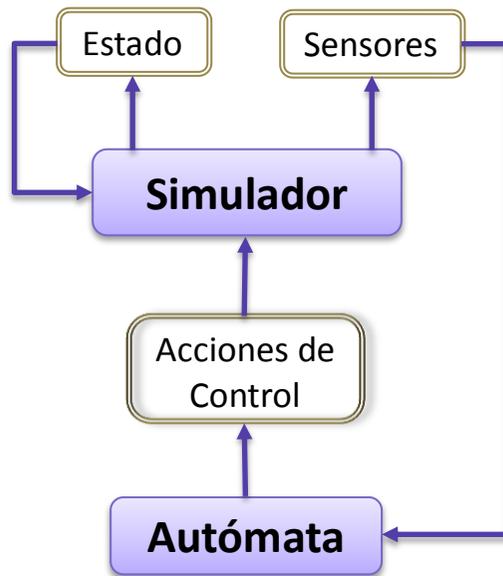


Ilustración 3.9. Esquema funcionamiento simulador

El sistema funciona enviando información de la situación de las acciones que están activadas desde el autómata hacia el simulador para que este ponga en funcionamiento los movimientos de máquinas a las que corresponden las acciones activadas. Por ejemplo si el autómata tiene activada la etapa de bajar taladros, el simulador baja los taladros. Cuando se termina de realizar la acción en el simulador, este modifica la información de acciones realizadas y sensores activados y por un lado, actualiza el simulador al siguiente estado, y por otro, envía la información de los sensores activados para que el autómata pase a las siguientes acciones. Y se vuelve a repetir el proceso

El motor de funcionamiento del simulador es un sistema cuyo modelo está basado en Redes de Petri Coloreadas. Este modelo es el que hace que el sistema

siga la secuencia de fabricación y las piezas no se salten procesos o sigan un orden incorrecto. El modelo indica en cada momento donde esta cada pieza y permite al autómeta accionar sólo los movimientos de la maquina correspondiente. La información se actualiza cada vez que llega una acción de control nueva desde el autómeta y no deja accionar ninguna maquina más para esa pieza hasta que no se han encendido los sensores que indican el final de la parte de fabricación de dicha máquina. Utilizar Redes de Petri Coloreadas tiene una gran ventaja en el sistema, ya que permite hacer un seguimiento a tiempo real de la fabricación de todas las piezas.

La comunicación entre el simulador y el autómeta utiliza el protocolo Hostlink. Este es el protocolo de comunicación para PLC's de la marca Osrom. Su función es por un lado pasar la información de las acciones que están activadas desde el autómeta al simulador, y por otro indicar al autómeta que sensores se van activando en el simulador.

#### **3.1.2. Redes de Petri.**

Las Redes de Petri (PN) son una alternativa de modelado de sistemas, aplicados principalmente hacia el control de procesos, por su facilidad de manejo en el problema de la sincronización de secuencias.

Fueron inventadas por el alemán Karl Adam Petri en 1962. En su tesis doctoral "kommunikation mit automaten" (Comunicación con autómetas), establece los fundamentos para el desarrollo teórico de los conceptos básicos de las PN. Con su ayuda se pueden modelar el comportamiento y la estructura de un sistema y llevar el modelo a condiciones límite, que en un sistema real son difíciles de lograr o muy costosas. La teoría de PN ha llegado a ser reconocida como una metodología establecida en la literatura de la robótica para modelar los sistemas de manufactura flexibles. Comparada con otros modelos de comportamiento dinámico gráficos, como los diagramas de las máquinas de estados finitos, las PN ofrecen una forma de expresar procesos que requieren sincronía. Y quizás lo más importante es que las PN pueden ser analizadas de manera formal y obtener información del comportamiento dinámico del sistema modelado.

Una Red de Petri clásica se concibe como un grafo dirigido que posee dos tipos de nodos principales: los lugares representados por círculos y las transiciones representadas por barras rectangulares. Entre los nodos se ubican los arcos dirigidos, los cuales se encargan de unir las transiciones con los lugares y viceversa. Cada arco dirigido posee un número que indica su peso, el cual determina la cantidad de marcas que consume de un lugar o deposita en un lugar, siempre y cuando se haya disparado una transición habilitada. Los arcos dirigidos sin número se entiende que consumen o depositan una marca. Las marcas se representan en forma gráfica como puntos negros que se ubican dentro de cada lugar.

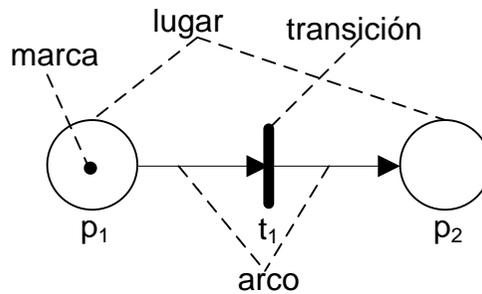


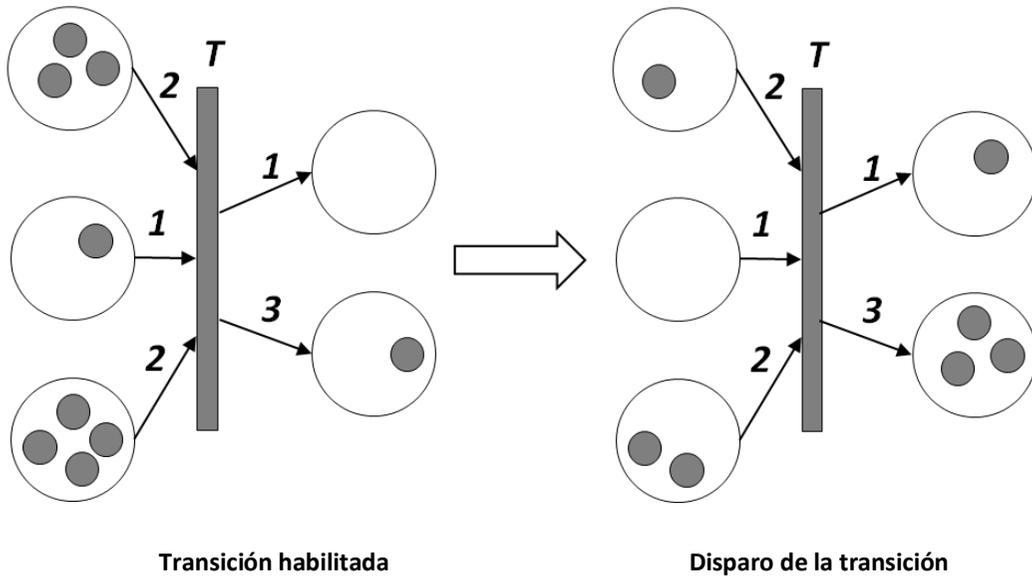
Ilustración 3.10 Componentes Red de Petri

El comportamiento del sistema puede ser descrito en términos de sus estados y sus cambios. En las PN, el estado del sistema, o mejor dicho, el marcado de la PN cambian de acuerdo con las siguientes reglas de disparo o transición:

1. Se dice que una transición está habilitada si cada lugar de entrada antes de dicha transición tiene el número de marcas que indica el peso situado al lado del arco. Si no pone nada se supone que es uno.
2. Una transición habilitada puede o no ser disparada (eso depende solamente del carácter no determinista del evento).
3. El disparo de una transición habilitada remueve “x” marcas de cada lugar de entrada y agrega “y” marcas a cada lugar de salida donde “x” e “y” es el peso de los arco desde el lugar a la transición o viceversa.

En el siguiente ejemplo se muestran las reglas de transición. La transición queda habilitada porque hay más marcas en los lugares de entrada a la

transición que pesos en los arcos dirigidos. Cuando se dispara la transición existe un flujo de marcas desde los lugares de entrada hacia el lugar de salida.



#### 3.2.3. Redes de Petri Coloreadas

Las Redes de Petri Coloreadas (CPN) pertenecen a la familia de las PN, la diferencia viene marcada por las consideraciones en CPN de colores y las funciones lineales asociadas a sus arcos. Las marcas de color representan un atributo o condición que se tiene que cumplir, además de que haya el número de pesos que marcan los arcos para que la transición esté habilitada. Una transición en CPN está en estado habilitado si todos sus nodos de entrada contienen un número de colores igual o mayor que los definidos por el peso del arco. El añadir colores permite a las redes de Petri coloreadas tener más condiciones para que las transiciones estén habilitadas. Se pueden añadir tantos colores como se quiera.

El ejemplo es similar al de las Redes de Petri, pero en este caso sólo estará habilitada la transición cuando se cumpla que todas las entradas tiene el número de marcas del color que indica el peso de cada arco. Cuando se dispara la transición sale desde allí las marcas con las condiciones que tenga el peso del arco correspondiente a cada salida.

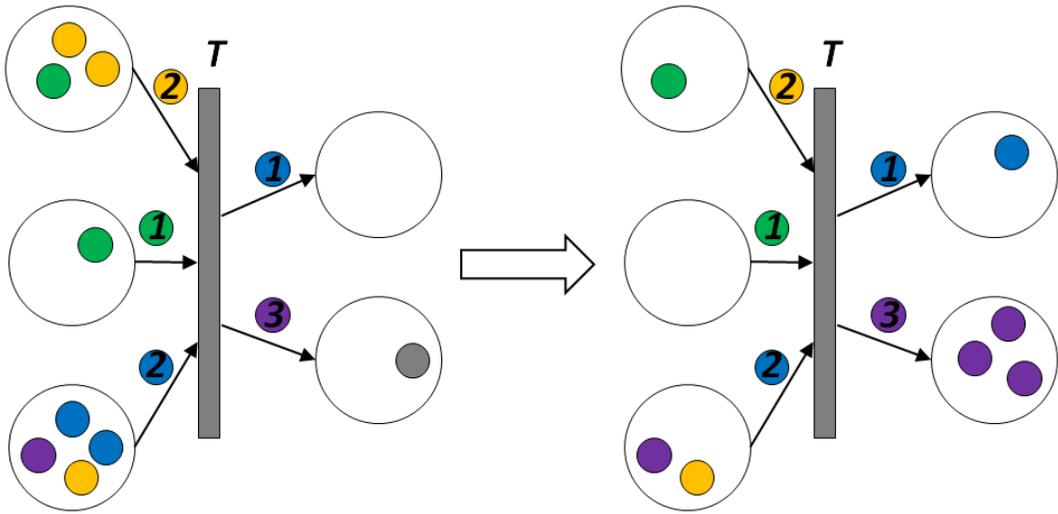


Ilustración 3.11. Disparo Transición CNP

#### 3.2.4. Red de Petri Coloreadas y modelado del simulador.

La Red de Petri Coloreada es el motor de funcionamiento del simulador. Representa el modelo de la secuencia de fabricación de las piezas dentro de las líneas. El programa trabaja con tres grafos iguales, uno para cada línea. En el grafo que representa al modelo, las transiciones representan a las máquinas y los círculos los lugares donde esperan las piezas a que las transiciones estén habilitadas. Los colores indican las condiciones que deben cumplir las piezas para abandonar los círculos y pasar a las transiciones. Cada condición de color representa un estado de la pieza, por ejemplo, la condición para que el taladro haga los agujeros es que la pieza ya esté cortada.

El modelo va pasando ordenadamente por cada una de las máquinas que se necesitan para la fabricación de la pieza esperando hasta que la máquina haga su trabajo para pasar a la siguiente. En todo momento se sigue el orden de la secuencia de fabricación establecido, habilitando cada máquina según se va necesitando, al inicio se habilita la desbobinadora, una vez que ha terminado su trabajo, se habilita la transición de corte, donde se corta, después la transición del movimientos de cintas, la del taladro y por último la de la pintura.

### 3. Diseño del Sistema

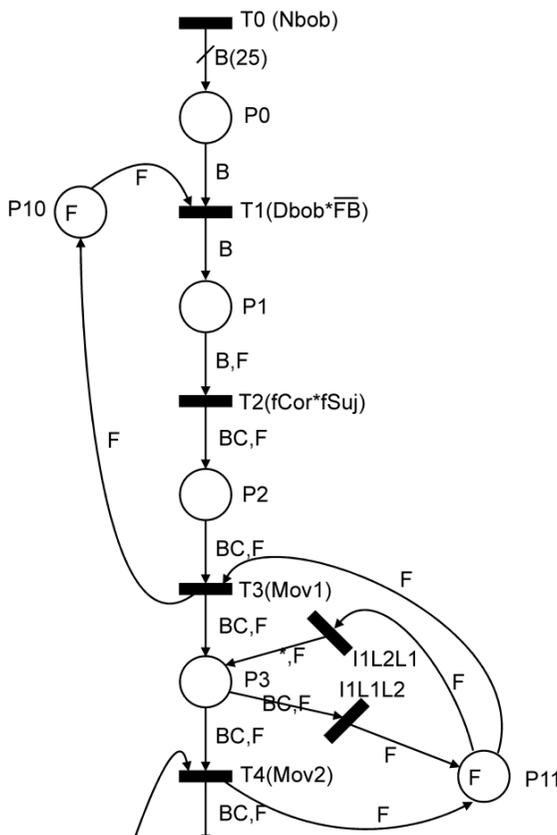


Ilustración 3.12. Grafica CNP Simulador

Cuando se dispara la primera transición se colocan 25 piezas o marcas en el primer círculo y se habilita la segunda transición. A partir de aquí se envían piezas de una en una en orden al resto del diagrama. Al dispararse la segunda transición se activa la función de desenrollar bobina, para sacar la primera pieza. Al terminar la pieza pasa a P1, con la condición B (significa pieza inicial), allí espera a que se habilite y dispare la transición T1 donde se activa la función de corte. Cuando sale la pieza de la transición T1 sale con la condición de BC, (pieza cortada). El proceso continúa así con el resto de máquinas, para que entre en las transiciones se tiene que

cumplir que el proceso que la precede en el orden de fabricación ya esté listo, a la vez, cuando sale de la transición, sale con la condición de que esa parte ya está acabada.

Para que las piezas entren de una en una en las máquinas, y no se junten dos a la vez, se han creado lugares que se llaman forzados. Los forzados son caminos que comienzan en la transición que marca el final del proceso de cada máquina, (la transición que marca la acción para sacar la pieza) y termina en el comienzo de la acción. Durante el tiempo que una pieza está dentro de la máquina, en los lugares de los forzados no hay ninguna marca, deshabilitando la transición de entrada a la máquina, cuando la pieza sale de la maquina se envía una marca al forzado, y este vuelve a habilitar la transición de entrada. Todas las máquinas trabajan con forzados para evitar que entren piezas mientras trabajan.

También se usan forzados en los puntos intermedios 1 y 2, para evitar que junten piezas a la vez de varias líneas.

El modelo final que representa al sistema completo está compuesto por las redes de Petri que forman las tres líneas y las líneas de intercambio que las comunican. El modelo gráfico es el siguiente:

### 3. Diseño del Sistema

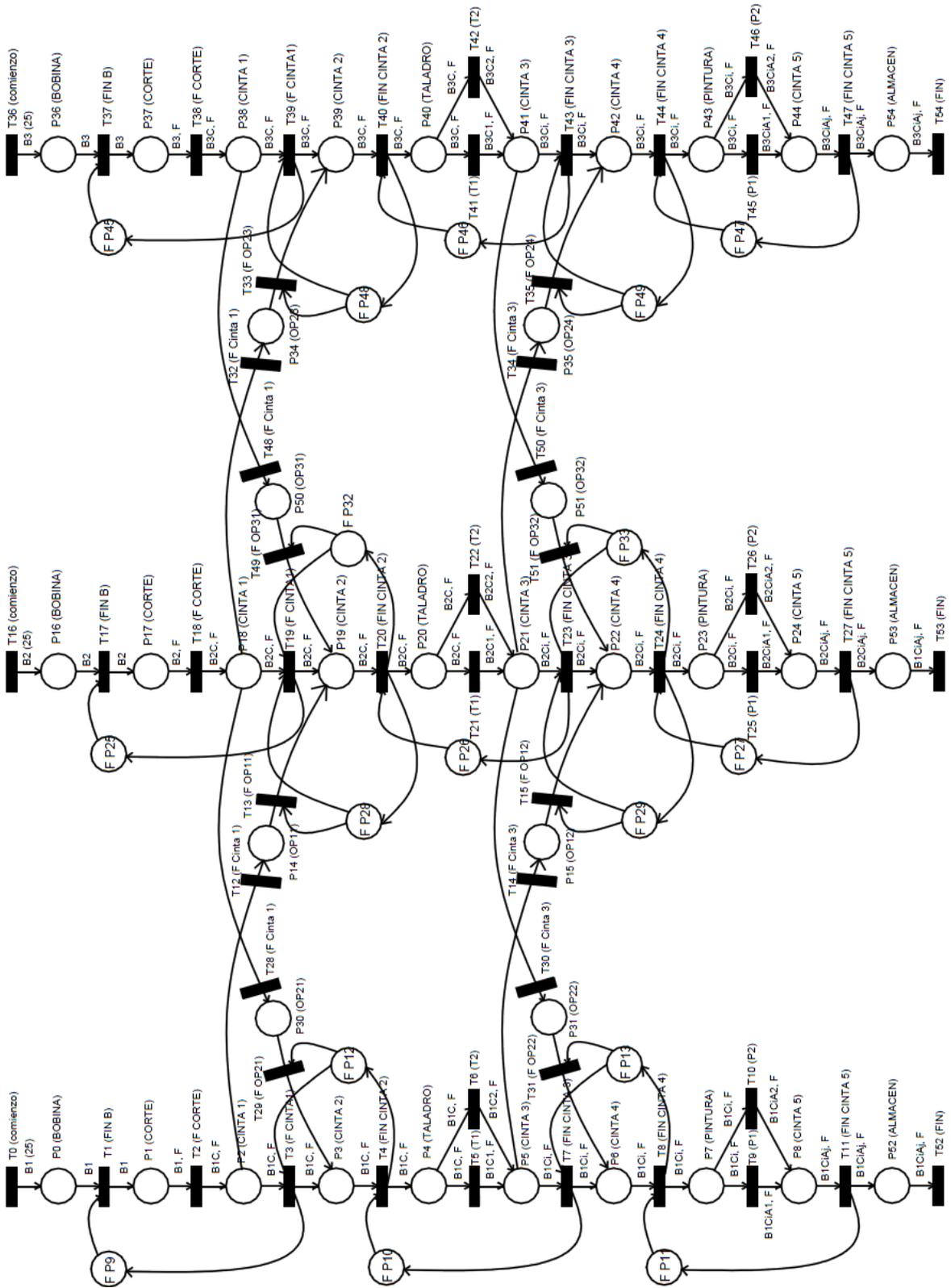


Ilustración 3.13. Red de Petri Coloreada Simulador

### 3.2.5. Interpretación del modelo en el simulador

El simulador interpreta el modelo de las redes de Petri de manera matemática. Las redes de Petri se pueden interpretar de forma gráfica y de forma matemática. La forma gráfica es el esquema anterior. La forma matemática se forma a partir de la ecuación de estados. Esta ecuación expresa la evolución de las marcas a través del tiempo, va cambiando cada vez que se dispara una transición. Para calcular la evolución del sistema, la ecuación utiliza como datos el valor del estado actual escrito en forma de matriz, llamada matriz de disparos y la matriz de incidencias.

La ecuación de estados calcula la posición que van a tener las marcas cuando se dispara una determinada transición:

$$M_{k+1} = M_k + I * T_k$$

Para explicar la ecuación se va a utilizar en siguiente ejemplo:

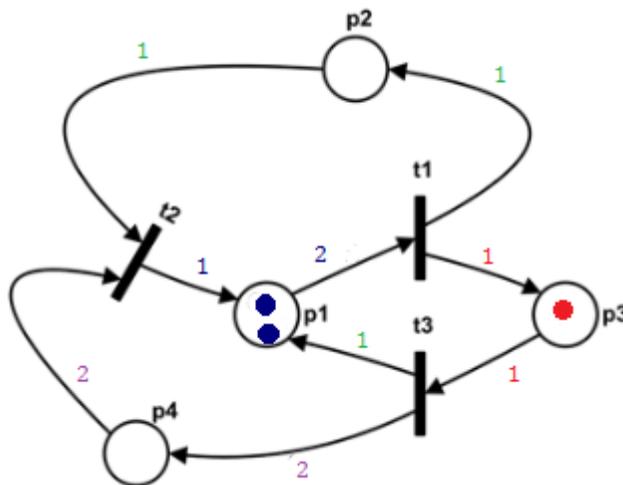


Ilustración 3.14. Ejemplo CNP

- Matriz de marcado k-esima+1,  $M_{k+1}$ .

$M_{k+1}$  es la matriz de marcado k-esima +1, es la matriz que indica las marcas que se van a activar cuando se dispara una determinada transición.

- Matriz de marcado,  $M_k$ .

### 3. Diseño del Sistema

---

$M_k$  es la matriz de marcado actual, indica en qué lugares hay marcas y cuál es su valor actual. Si la red es coloreada, además de poner lo numero en cada lugar, se pone de qué color son.

Matriz de marcado del ejemplo:

$$M_k = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- Matriz de incidencias I.

En una red de Petri con  $n$  transición y  $m$  lugares, la matriz de incidencias es la matriz de enteros de *filas x columnas*; indica el valor de los arcos entre lugares y transiciones de la red. Los valores negativos son los valores de los arcos desde los lugares a las transiciones, los valores positivos representan el valor de los arcos que van desde las transiciones a los lugares.

La matriz de incidencias del ejemplo:

$$I = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{bmatrix}$$

- Matriz de disparo  $T_k$ .

Matriz de una sola columna compuesta por ceros excepto los lugares de transiciones que se quieren disparar que tienen 1. Si quiere disparar la transición 3, la matriz es:

$$T_k = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Según la ecuación de estado cuando se dispara la transición 3 desde el estado actual en el ejemplo, el resultado sería:

$$M_{k+1} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 + 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

Que es el mismo que se halla de manera gráfica.

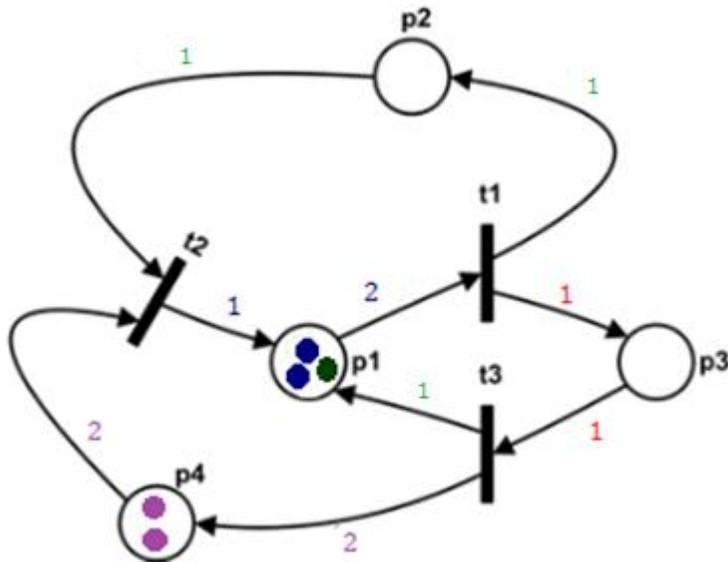


Ilustración 3.15. CNP Disparada

El simulador va ejecutando la ecuación continuamente actualizando los estados. La matriz de disparos que hace falta para que se pueda ejecutar la ecuación se crea a partir de la información que envía el PLC porque dicha matriz la crea el PLC cuando indica al simulador que etapas están activadas y cuáles no.

A la vez que se ejecuta la ecuación de estados, el simulador también va ejecutando otra ecuación llamada *ecuación de salida*. Esta ecuación va indicando los sensores que se tienen que activar para finalizar las acciones. Información que luego se envía al PLC.

Ecuación de Salida.

$$Y_k = S_k * F$$

dónde:

- $S_k$  es la matriz fila que marca los lugares que están activos en el instante  $k$
- $F$  es la matriz que representa a los sensores que tiene la planta. La matriz está formada por el eje  $X$  donde están todos los sensores

colocados ordenadamente y el eje Y dónde están, también ordenados, los lugares de la Red de Petri.

#### **3.2.6. Modelo de gestión de estados**

El sistema trabaja alternativamente cogiendo información del estado del autómatas y del estado del simulador. El proceso comienza cuando el autómatas envía la información de las etapas que están activadas al simulador para que este cree la matriz de disparo, el simulador utiliza esta información para hacer dos cosas:

1. Ejecutar la ecuación de estado, disparando las transiciones que correspondan en ese momento para crear los movimientos en el simulador y actualizar el estado al siguiente instante.
2. Ejecuta la ecuación de salida para ese momento, con lo que se calcula los sensores que se tienen que activar para terminar la acción. Esta información se envía al PLC, el cual ejecuta la transición que toque y pasa a la siguiente etapa.

El proceso es cíclico, donde el autómatas informa al simulador de los estados que están activados en el instante k. El simulador ejecuta la ecuación de estados, disparando las transiciones que le ha indicado el autómatas para pasar al siguiente estado. En el simulador también se ejecuta la ecuación de salida, que indica que sensores se han activado. El número de sensores activados es la información que envía el simulador al autómatas para que actualice su estado y el número de etapas activadas. Y con la nueva situación de estados activados en el autómatas se vuelve a repetir el proceso.

#### **3.2.7. Comunicación Simulador-PLC**

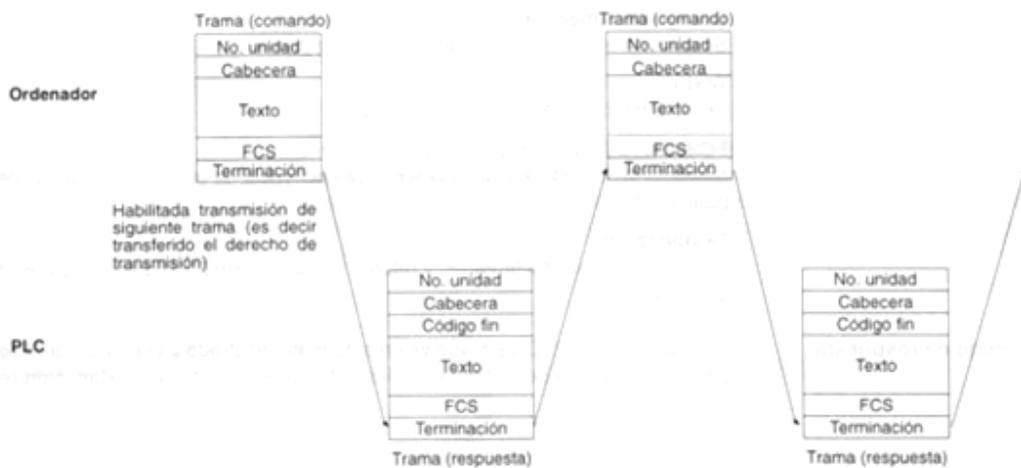
El sistema que utiliza el simulador para comunicarse con el exterior es un sistema estándar que permite comunicarse con cualquier PLC. Esto es así porque el simulador utiliza para conectarse variables globales de entrada y

salida, el protocolo de conexión del PLC solo tiene que interpretar estas variables como si fuesen las acciones reales.

En la tesina se ha utilizado un autómata de la marca Osrom, que utilizan el protocolo Hostlink para la comunicación con el exterior.

#### Protocolo Host link

Éste protocolo establece el procedimiento adecuado para la transmisión de información vía RS-232 y el área y tipo de información accesible.



**Ilustración 3.16. Secuencia del protocolo HostLink**

El protocolo de comunicación hostlink se basa en el envío y recepción de una serie de tramas. Por parte del PC se envía las tramas denominadas "comando" mientras que el PLC siempre devuelve tramas de respuesta donde se encuentra la información solicitada por el comando o simplemente el resultado, exitoso o no, de la operación solicitada.

Primero el PC envía una trama al PLC solicitando una operación (también puede iniciarse desde el PLC). La trama consiste en 5 campos:

- Unidad: número de PLC al que va destinada la trama
- Cabecera: Código de la operación solicitada:
- Texto: Espacio específico para datos de cada operación.

### 3. Diseño del Sistema

---

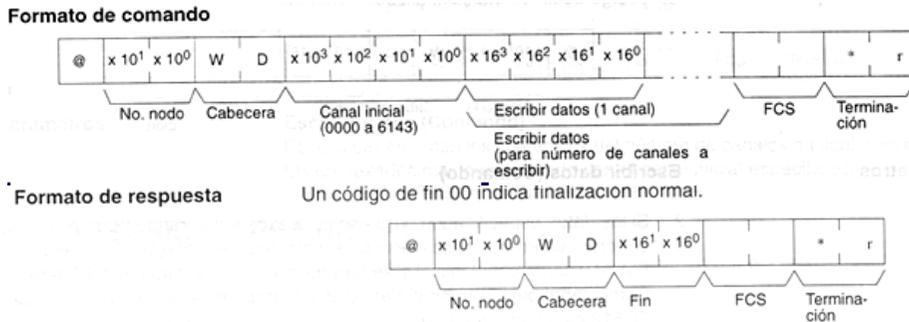
- FCS: Código de seguridad para verificar que la trama recibida es correcta
- Terminación: Dos caracteres que indican el final de la transmisión.

Cuando el PLC con el número indicado recibe la trama, primero comprueba que el valor de FCS es el correcto y responde al PC con otra trama:

- Unidad: número de PLC al que va destinada la trama
- Código fin: Código que indica el éxito o fracaso de la operación y sus motivos.
- Cabecera: Código de la operación solicitada:
- Texto: Espacio específico para datos de cada operación.
- FCS: Código de seguridad para verificar que la trama recibida es correcta
- Terminación: Dos caracteres que indican el final de la transmisión.

La composición de las tramas tanto las de comando como las de respuestas dependen por completo del tipo de operación que se quiera realizar. Esta diferencia reside en el contenido del campo de cabecera y texto de la trama. Hay que diferenciar las operaciones de lectura y de escritura. Dentro de cada una de ellas se emplea un formato de trama distinta según el área de memoria a la que se destinen.

El formato concreto de cada una de las tramas se encuentra definido en el manual de programación del OMRON CQM1. Sin embargo, se va a detallar algunas de las tramas que se emplean de forma frecuente en nuestra aplicación.



**Ilustración 3.17. Formato de las tramas de escritura en memoria DM**

La trama de escritura para el área de memoria DM dispone de todos los campos descritos anteriormente. Como cabecera, este tipo de tramas emplean la cadena “WD”. Posteriormente en el campo de texto se introduce la cadena correspondiente (con cuatro dígitos) al canal inicial donde comienzan a escribirse los datos. Acto seguido se introduce también las cadenas de cuatro dígitos que contienen los datos a escribir. Pueden actualizarse más de un canal con una única orden, siempre que estos sean consecutivos, introduciendo los datos a escribir secuencialmente uno detrás de otro dentro del campo de texto. La trama de respuesta devuelve una cabecera igual a la enviada por el comando y el campo fin nos devuelve los dígitos de control que indican el resultado de la operación. Si la operación se ha efectuado correctamente el campo de fin contendrá la cadena “00” sin embargo, si ha ocurrido algún error, la cadena contendrá un código diferente a “00”. Cada cadena representa a un error diferente. En la siguiente tabla se muestra los códigos y los errores a los que corresponden:

### 3. Diseño del Sistema

Cód. fin	Contenidos	Causa probable	Corrección
00	Finalización normal	---	---
01	No ejecutable en modo RUN	El comando enviado no se puede ejecutar cuando el PLC está en modo RUN.	Comprobar la relación entre el comando y el modo del PLC.
02	No ejecutable en modo MONITOR	El comando enviado no se puede ejecutar cuando el PLC está en modo MONITOR.	
04	Dirección demasiado alta (PLCs CPM1/CPM1A/SRM)	Se ha excedido la dirección más alta del área de programa de usuario.	Comprobar el programa.
0B	No ejecutable en modo PROGRAM	El comando enviado no se puede ejecutar cuando el PLC está en modo PROGRAM.	Este código no está siendo utilizado actualmente.
13	Error de FCS	El FCS es falso. El cálculo del FCS es erróneo o hay influencias adversas de ruido.	Comprobar el método de cálculo de FCS. Si había influencia de ruido, transferir de nuevo el comando.
14	Error de formato	Formato de comando erróneo.	Comprobar el formato y transferir de nuevo el comando.
15	Error de datos de número de entrada	Las áreas para leer y escribir son erróneas.	Corregir las áreas y transferir de nuevo el comando.
16	Comando no soportado	El comando especificado no existe en la dirección especificada. (Leer SV, etc.)	Comprobar dirección e instrucción.
18	Error de longitud de trama	Se ha excedido la longitud máxima de trama.	Dividir el comando en varias tramas.
19	No ejecutable	Parámetros a leer no registrados para comando compuesto (QQ).	Ejecutar QQ para registrar parámetros a leer antes intentar leerlos.
23	Memoria de usuario protegida contra escritura	PLCs CQM1: Pin 1 del interruptor DIP del CQM1 en ON. PLCs CPM1/CPM1A/SRM1: La memoria está protegida en la configuración del PLC	PLCs CQM1: Poner el pin 1 a OFF para ejecutar. PLCs CPM1/CPM1A/SRM1: Cambiar la selección en la configuración del PLC (DM 6602).
A3	Abortado debido a error de FCS en transmitir dato	El error se generó mientras se estaba ejecutando un comando que ocupa más de una trama. <b>Nota:</b> Los datos hasta ese punto se graban en la área apropiada de la CPU.	Comprobar los datos de comando e intentar transferir de nuevo.
A4	Abortado debido a error de formato en transmitir dato		
A5	Abortado debido a error de dato de número de entrada en transmitir dato		
A8	Abortado debido a error de longitud de trama en transmitir dato		
Otro	---	Se ha recibido ruido.	Transferir de nuevo el comando.

Ilustración 3.18. Errores de Host Link

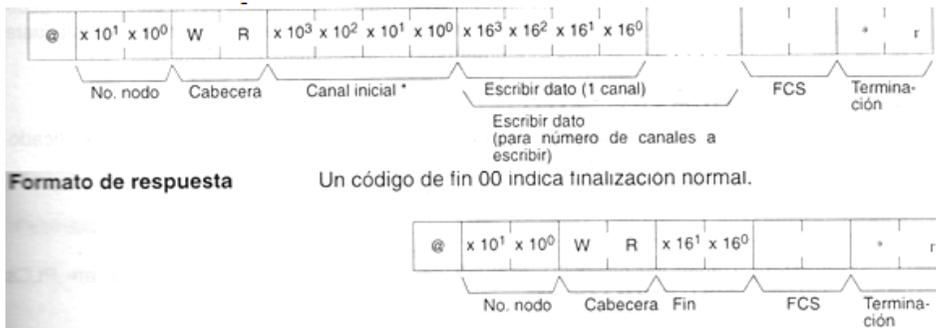


Ilustración 3.19. Formato de las tramas de escritura en memoria IR

La escritura de datos en la memoria interna (IR) es similar a la de la memoria de datos. La única diferencia es que en el campo de cabecera se introducirá la cadena “WR”

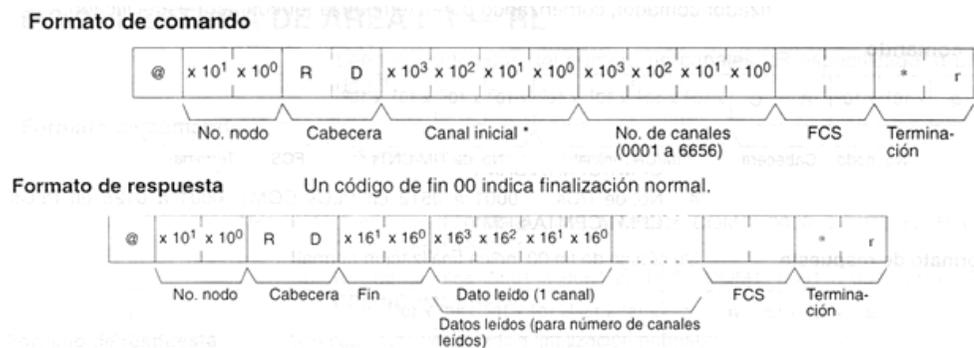
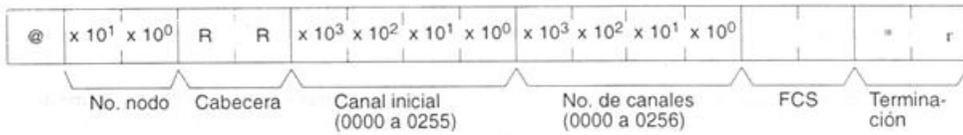


Ilustración 3.20. Formato de las tramas de lectura en memoria DM

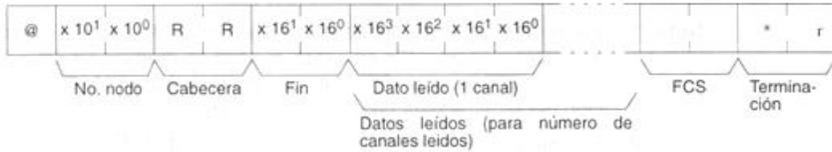
La lectura de datos emplea una estructura ligeramente similar a las vistas para la escritura de datos. La cabecera usa la cadena “RD” al que se le añade el canal inicial (con cuatro dígitos) y una cadena adicional de hasta cuatro caracteres que indica el número de canales consecutivos que se desea leer. El formato de respuesta en este caso, además del código de fin que muestra el resultado de la operación, incluye una cadena de cuatro caracteres por cada canal que haya decidido leer.

### 3. Diseño del Sistema



#### Formato de respuesta

Un código de fin 00 indica finalización normal.

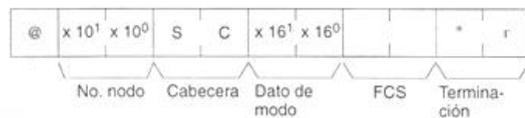


**Ilustración 3.21. Formato de las tramas de lectura de memoria interna (IR)**

De manera similar a lo visto en la lectura de memorias DM, para leer el valor de un registro de la memoria interna se cambia la cabecera con la cadena "RR". El formato de la respuesta es el mismo que el visto en el tipo anterior.

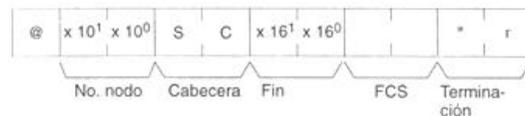
Para el caso en el que se quiere acceder a alguna otra área de memoria del PLC tan sólo hay que consultar el manual del PLC para comprobar la cadena necesaria tanto para el comando como para interpretar la respuesta.

#### Formato de comando



#### Formato de respuesta

Un código de fin 00 indica finalización normal.



**Ilustración 3.22. Formato de las tramas de lectura de memoria interna (IR)**

Por último, en cuanto a las tramas queda destacar las tramas de los comandos especiales. Estas tramas se emplean para ejecutar operaciones especiales en el PLC como puede ser el cambio de modo de funcionamiento. Esta trama se caracteriza por tener una cabecera con la cadena "SC" e

incorporar como texto una cadena de dos caracteres que indica el modo al que se cambiara el autómata según se indica en la siguiente tabla.

Cod.	Modo
00	Modo stop
11	Modo Run
10	Modo Monitor

Tabla 1. Modos de funcionamiento del PLC

### 3.1.8. Sistemas Scada

El simulador tiene la opción de utilizarse como si fuera un sistema Scada para el control y supervisión de fallos dentro de un sistema real.

Para la realización de la tesina, se ha utilizado el simulador como sistema donde la comunicación entre el PLC y el simulador será cíclica, el PLC envía información al simulador de la situación de los estados utilizando globales y el simulador responde enviando información de los sensores que se han activado para que el PLC pase al siguiente estado.

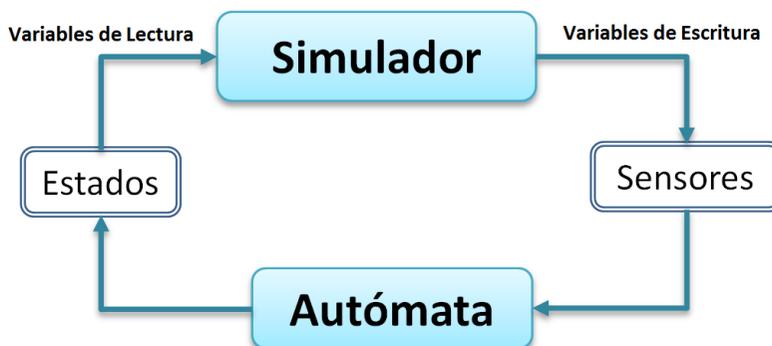


Ilustración 3.23. Esquema Funcionamiento del Sistema

Las variables que envía el autómata hacia el simulador son de lectura. El simulador crea variables de escritura, que lee el autómata.

### 3. Diseño del Sistema

---

Si el autómata se conecta a un sistema real, el ciclo de comunicación es el mismo pero se sustituye el simulador por el sistema real. La comunicación con el simulador se basa en variables globales de lectura y escritura en un sistema normal. Si se conecta el PLC al simulador y la información que llega desde los sensores del sistema real, se pasa también al simulador en forma de variables de lectura, se crea un sistema Scada, que permite controlar y supervisar el proceso obteniendo información de la planta en todo momento.

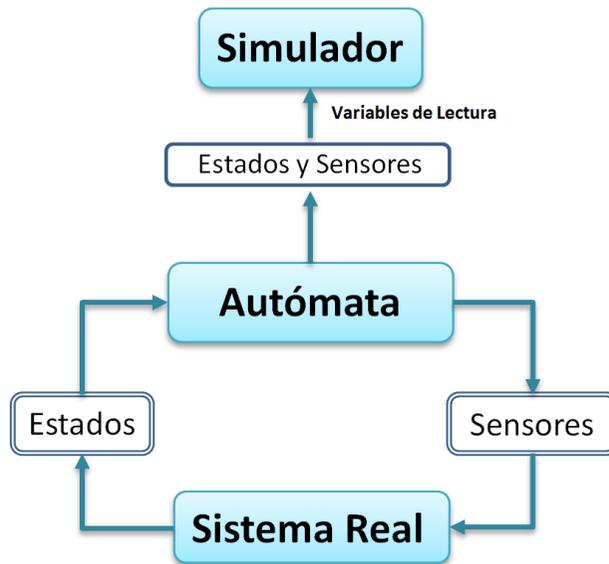


Ilustración 3.24. Esquema Funcionamiento Scada

# Control del Sistema

---

El segundo escalón en la automatización de la planta industrial es el control de las máquinas. El control debe realizarse de una manera automática y a tiempo real, sin la necesidad de que ninguna persona intervenga en el proceso y siguiendo una secuencia lógica para que la fabricación de cada pieza sea correcta.

El control de las máquinas apareció por la necesidad de realizar los procesos de una forma más rápida y así disminuir los tiempos de fabricación. Al principio se hacía de manera electromecánica, usando elementos como relés y contactores y uniéndolos con cable, pero este tipo de automatización tenía muchos problemas, como por ejemplo la cantidad de metros de cable que hacían falta o el elevado precio que suponía hacer una instalación así. Además el control no llegaba a ser eficiente, ya que si se producía algún cambio en la línea de producción, había que volver a cambiar todo el sistema y los componentes se solían estropear con bastante frecuencia. A finales de la década de los 60 se produjo un giro de 180 grados con la aparición de los semiconductores y los primeros circuitos integrados, los cuales permitieron disminuir notablemente el número de elementos, logrando sistemas de menor tamaño y desgaste y mayor fiabilidad. Unos años después llegó el microprocesador y con ello el autómata programable o PLC. Actualmente, el autómata es el encargado de coordinar y dirigir a las máquinas que forman parte del proceso productivo.

Su función principal es hacer que el conjunto de máquinas que están a su cargo realicen las operaciones de una manera secuencial, siguiendo las instrucciones que le han dado anteriormente y sin que aparezcan errores en los procesos.



Ilustración 4.1. Pirámide de Jerarquías

## 4. Control del Sistema

---

Con el control se sube un escalón en la pirámide de jerarquías del sistema de automatización tipo CIM, llegando al Nivel 1. La misión de este nivel es únicamente la de controlar y coordinar las máquinas que están dentro del Nivel 0 siguiendo las indicaciones que le dan los niveles superiores. Las instrucciones que debe seguir el autómatas para que cada pieza siga la ruta que ha calculado el Nivel de Planificación y así, el tiempo total de fabricación de todas las piezas sea el mínimo. El Nivel 1 también es el encargado de recoger la información acerca del estado del Nivel 0 y de las incidencias que puedan ocurrir, como el fallo de alguna máquina, e informar al Nivel de Supervisión para que éste muestre como solucionar el problema de la manera más eficiente.

El autómatas tiene que coordinar dos tipos de secuencias diferentes. Por un lado está la secuencia de proceso, que es el camino que tiene que seguir a través de las máquinas cada pieza. El orden es el siguiente:

**1º Desenrollar material.**

**2º Corte de la chapa.**

**3º Taladrar.**

**4º Pintura.**

Cada pieza tiene que seguir este orden obligatoriamente. No puede activar la pintura antes que el taladro, y enviar la pieza primero a la pintura y luego hacer que retroceda hasta el taladro. Tampoco puede hacer taladros sin que haya ninguna pieza en la mesa de trabajo o desenrollar material si en la zona de corte aún hay pieza. Para que la planta funcione con normalidad, el autómatas tiene que ejecutar esta secuencia sin que haya alteraciones.

La segunda secuencia es la que tiene que ver con las acciones que tiene que seguir cada máquina para hacer su trabajo. El PLC también se encarga de coordinar los movimientos que tiene que seguir cada máquina para que la pieza quede bien fabricada.

La planta tiene una dificultad añadida en la coordinación de las secuencias porque las líneas no trabajan independientemente, si no que pueden intercambiarse piezas entre ellas a través de los puntos de

intercambio usando los intercambiadores. Esto quiere decir que una pieza que está cortada en la línea 1, por ejemplo, puede pasar a la línea 2 para que se hagan los taladros y luego a la línea 3 para que se pinte. El autómata además de controlar las secuencias del proceso y las de las máquinas, debe controlar los intercambios de piezas entre líneas, procurando que dichos intercambios sean desde las líneas correctas, en el momento que toca y sin que se junten varias piezas en el mismo punto.

El autómata es el que ejecuta las acciones y hace que se muevan los elementos, pero necesita que antes le indiquen las instrucciones que tiene que seguir. Para ello existen varios tipos de programación en automatización. En este caso se ha utilizado lenguaje en Grafcet, (siglas de Grafica de Control de Etapas y Transiciones), el lenguaje en Grafcet permite hacer un modelo de la secuencia a controlar, contemplando las entradas, las acciones a llevar a cabo y las señales que indican cuando efectuar cada acción.

La programación con grafcet es un tipo de lenguaje estándar y bastante fácil de usar, pero los fabricantes de autómatas no lo suelen usar en sus software, tienen sus lenguajes propios desarrollado a partir de los grafcet.

Los pasos que se han seguido para crear el control de la planta han sido:

1. Programación en grafcet.
2. Traducción de los programas en lenguaje grafcet al software que tiene el autómata para el ordenador.
3. Transferencia el programa del ordenador al PLC, comprobación que no hay errores a nivel de programación.
4. Verificación de que el programa sigue la secuencia de procesos correcta y todos los elementos se ejecutan bien.

### **4.1. Funcionamiento del autómata**

El autómata o PLC, es el encargado de hacer que todas las máquinas de la línea de producción funcionen siguiendo una secuencia lógica. Es el aparato que gobierna y controla todo el proceso de producción.

## 4. Control del Sistema

---

Cualquier sistema de control está formado por dos partes, la parte operativa y la parte de control. La parte operativa es la parte que realiza el trabajo, las máquinas que forman parte del proceso. La parte lógica o de control es el autómata, el cual gobierna a la parte operativa. Ambas partes se comunican a través de los actuadores y los captadores.

- Los actuadores son los elementos que realizan las acciones. Puede ser el motor que mueve una cinta o el que hace que suba y baje el taladro. Todos estos elementos cumplen las órdenes que le envía el PLC. Los actuadores es la vía de comunicación desde la parte lógica a la operativa.
- Los captadores o sensores recogen información a tiempo real del estado del proceso desde la parte operativa y la envía a la parte lógica para que esta la procese, tome decisiones e indique que actuadores se tienen que activar o desactivar en cada momento. Casi todos los sensores que se han utilizado en este caso son de posición, para marcar donde están las piezas o los actuadores.

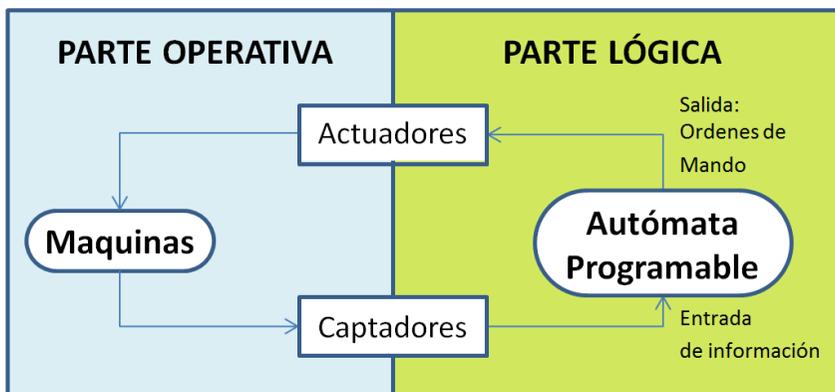


Ilustración 4.2. Funcionamiento del Autómata

### Partes del Autómata

Para que el autómata funcione es necesario que tenga:

- Un microprocesador.
- Módulos de entrada/ salida.

El microprocesador o microcontrolador está compuesto por:

- a. Memoria ROM, (Read Only Memory), memoria sólo de lectura, tiene programas que no se pueden borrar como el de arranque del PLC.
- b. Memoria RAM, (Random Access Memory), memoria para leer y escribir. En ella se graban los programas que debe ejecutar el autómata mientras trabaja. También guarda la información procedente de los captadores.
- c. Parte Lógica, es la parte del microprocesador que hace las operaciones, analiza la información procedente de la parte operativa y ejecuta las instrucciones a seguir según le indica el programa.

Los módulos de entrada/salida sirven para comunicar el PLC con el exterior, con la parte operativa. En el módulo de entrada están conectados los sensores; por aquí llega la información al PLC del estado de la parte operativa que después es enviada al procesador. En el módulo de salida van conectados los accionadores. Desde aquí, el PLC va ejecutando las órdenes que le envía el procesador para activar o desactivar el elemento que toca en cada momento.

Ambos módulos funcionan mediante registros. Un registro es un conjunto de huecos o pines que están numerados y donde van conectados los captadores y los actuadores. La comunicación entre ellos y el PLC se crea asociando cada número de hueco y registro a la instrucción que corresponde cuando se pasa el programa en graficet al software del autómata. Por ejemplo, si se conecta el motor que desenrolla la bobina al hueco 3 del registro número 4 de salida en el autómata, cuando se está escribiendo esta instrucción en el programa del ordenador, hay que indicarle que esa instrucción corresponde a la dirección 4.3. (registro 4, hueco 3) de salida, para que así, cuando se ponga en funcionamiento el autómata y llegue a esa etapa o instrucción active o desactive el hueco.

En función del modelo de autómata, pueden llevar otros componentes adicionales como tarjetas de red, memorias externas o pequeñas pantallas.

Cuando se pone en marcha el autómata, el funcionamiento que sigue es simple. Va leyendo las instrucciones que le indica el programa creado

anteriormente de manera secuencial. Eso quiere decir, que primero lee la primera instrucción, realiza la acción que marca, como bajar el taladro o poner en funcionamiento una cinta y cuando acaba comienza con la siguiente. Para marcar el inicio o final de cada instrucción están los sensores. Son las condiciones que indican cuando empezar una acción o acabarla y pasar a la siguiente. Por ejemplo, cuando el taladro ha llegado abajo, hay un sensor que se enciende e indica que la acción ya ha termina y que hay que pasar a la siguiente instrucción. La secuencia sigue así sucesivamente hasta que llega al final del programa, activando las acciones como indican las instrucciones y pasando a la siguiente cuando lo marcan los sensores.

El PLC usado en la tesina es de la marca OMRON, modelo CQM1.



Ilustración 4.3. Imagen Automata

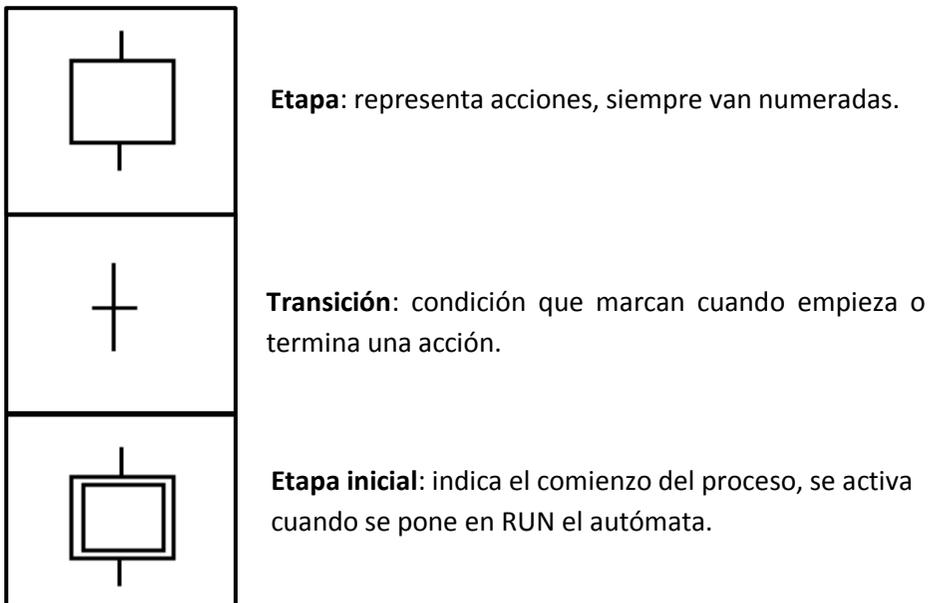
### 4.2. Lenguaje de programación: El Grafcet.

Los Grafcet son diagramas normalizados que permiten crear un modelo de un proceso para automatizar, contemplando entradas, acciones a realizar y transiciones intermedias que activan o desactivan las acciones.

Este lenguaje inicialmente fue propuesto para documentar la etapa secuencial de los sistemas de control de procesos de eventos discretos. No fue concebido como un lenguaje de programación de autómatas, sino un tipo de grafo para elaborar el modelo pensando en la ejecución directa del automatismo. Varios fabricantes empezaron a utilizarlo en sus PLC's de gama

alta, lo que lo convirtió en un potente lenguaje gráfico de programación para autómatas, adaptado a la resolución de sistemas secuenciales. En la actualidad no tiene una amplia difusión como lenguaje, puesto que la mayoría de los autómatas no pueden programarse directamente en este lenguaje, pero se ha universalizado como herramienta de modelado que permite el paso directo a programación.

### Componentes del diagrama



### Principios Básicos

Para realizar el programa correspondiente a un ciclo de trabajo en lenguaje grafcet, se deben de tener en cuenta los siguientes principios básicos:

- Se descompone el proceso en etapas que serán activadas una tras otra.

## 4. Control del Sistema

- A cada etapa se le asocia una o varias acciones que sólo serán efectivas cuando la etapa esté activa.
- Una etapa se activa cuando se cumple la condición de transición y está activa la etapa anterior.
- El cumplimiento de una condición de transición implica la activación de la etapa siguiente y la desactivación de la etapa precedente.
- Nunca puede haber dos etapas o condiciones consecutivas, siempre deben ir colocadas de forma alterna.
- 

El programa va activando cada una de las etapas y desactivando la anterior conforme se van cumpliendo cada una de las condiciones. Las acciones se realizan en función de la etapa activa a la que están asociadas. Por ejemplo, la etapa 1 se activa tras arrancar el programa, al cumplirse la "Condición 1", se activa la etapa 2, se desactiva la 1, y se realiza la "Acción 1".

Las acciones son los movimientos que deben hacer las máquinas como bajar el pistón de la sujeción para el corte. La condición o fin de cada acción lo marcan los sensores instalados en toda la planta. Para el caso de la sujeción para el corte, el fin de la acción lo indica un sensor colocado en la parte de debajo de la máquina, que se enciende cuando el pistón ha llegado abajo, desactivando esta etapa y activando la siguiente.

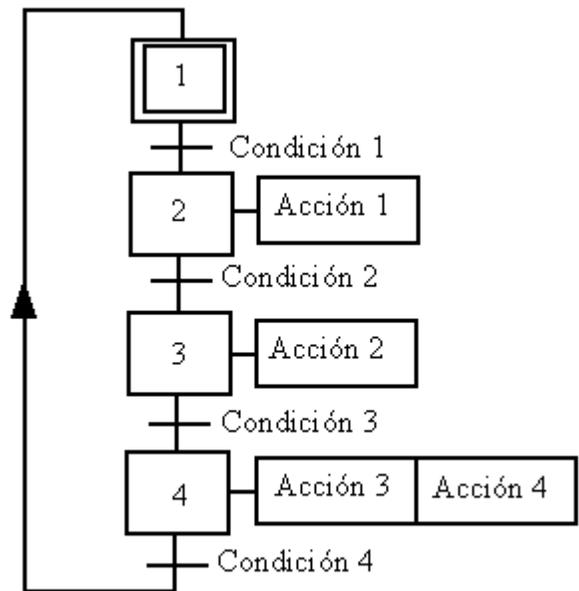


Ilustración 4.4. Grafset

### Tipos de grafset: producción, marcha y seguridad.

Existen tres tipos de grafset: de producción, de seguridad y de marcha.

- Seguridad: son los grafctet encargados de velar por la seguridad del sistema. Por lo general cuando hay un problema activan alarmas, congelan acciones y sensores y esperan a que se arregle el problema para seguir con la producción o reiniciar el sistema, dependiendo de la situación. En el caso de la planta de producción no se han usado porque de esta parte se encarga el nivel de supervisión de fallos.
- Marcha: su función sólo es la de iniciar y finalizar el sistema. Normalmente es un botón el encargado de poner en marcha todo. Las acciones de este tipo de grafctet están subordinadas a las de los grafctet de seguridad.
- Producción: son los grafctet que se encargan de mover las máquinas para que las piezas se puedan fabricar. Van subordinados inmediatamente por los grafctet de marcha, que son los que les indican cuando tienen que empezar a trabajar y un escalón por encima, por los grafctet de seguridad, que son los grafctet que toman el control cuando hay algún problema.

### **Creación de los Grafctet.**

Al igual que cualquier tipo de programación, la forma de crear grafctet para usarlo en automatización es libre y cada persona puede pensar de forma diferente, siendo igual de válidas todas. Lo importante es que el resultado final sea que el proceso se ejecute correctamente y sin fallos, sin que los elementos que forman parte del proceso se muevan a destiempo y puedan ocasionar accidentes, o que no se muevan y no realicen su trabajo. Es imprescindible que la secuencia se ejecute en el orden correcto, da igual como sea la programación que se ha creado siempre y cuando se cumpla esto.

### **4.3. Especificaciones de funcionamiento de la planta**

#### **4.3.1. Especificaciones generales**

Para que la planta industrial funcione correctamente, el nivel de control a través de los grafctet debe cumplir las siguientes condiciones:

## 4. Control del Sistema

---

- a. La planta debe ser capaz de producir piezas continuamente sin que haya interrupciones o alguna pieza se quede atascada en una máquina.
- b. Debe de poder intercambiar piezas entre diferentes líneas sin que aparezca ningún problema como por ejemplo enviar piezas a la línea que no corresponde o que llegue a destiempo.
- c. Que lleve el orden de secuencia de proceso correcta, siendo esta la de bobina-corte-taladro-pintura.
- d. Cada máquina debe seguir la secuencia de movimientos que le toca y no activar ninguno de sus elementos a destiempo o cuando no sea su turno.
- e. Ninguna maquina debe funcionar cuando no tenga piezas para procesar.
- f. No puede entrar ninguna pieza a alguna maquina mientras haya otra dentro, tiene que espera a que la maquina termine de trabajar con la primera para que pueda pasar.
- g. Debe haber una prioridad en los intercambios de piezas entre líneas para evitar que lleguen dos piezas a la vez al mismo punto.
- h. Si se estropea una máquina debe existir una alternativa para arreglar el problema, y que la producción no se pare más del tiempo necesario.

### 4.3.2. Especificaciones para cada maquina

Para que cada pieza se fabrique correctamente dentro de las máquinas, cada una tiene que seguir una secuencia de acciones. Además tiene que cumplir varias condiciones para que el proceso se desarrolle correctamente.

1. Desbobinadora.
  - Se tiene que colocar nueva bobina cuando no haya.
  - Tiene que ir sacando piezas de la bobina de una en una solo cuando la máquina de corte esté vacía.
2. Corte

- Sólo se puede poner en funcionamiento cuando se haya desenrollado material de la bobina.
- La secuencia de funcionamiento siempre tiene que ser bajada de la sujeción, bajada del corte, subida del corte, subida de la sujeción. Otra combinación de movimientos no corta el material.

### 3. Taladro

- Para que se puedan hacer los taladros la pieza tiene que estar ya cortada.
- El taladro solo puede activarse cuando llegue una pieza desde la cinta 2.
- Antes de ponerse a funcionar tiene que comprobar que está en el modo correcto y cambiarse si no está.
- La secuencia consiste en bajar el taladro y la sujeción hasta que el sensor indique que ya están los agujeros hechos y luego subir taladro.

### 4. Pintura

- La pintura sólo se puede activar cuando llega una pieza desde la cinta 4.
- La secuencia de funcionamiento consiste en comprobar que está en el modo de pintura correcto, y luego activar la función de pintura hasta que la pieza queda lista. Automáticamente la cinta 5 tiene que sacar la pieza de la línea.
- La pieza sólo se pinta siempre y cuando ya esté cortada y los taladros hechos.

### 5. Cintas 1 y 3

- La cinta 1 sólo se activa cuando haya una pieza cortada. La cinta 3 cuando el taladro haya terminado su trabajo, para poder hacer el traslado en ambas los puntos intermedios tienen que estar libres y que los intercambiadores no estén enviando piezas desde otras líneas.

### 6. Cinta 2

- Tiene que enviar las piezas desde el punto intermedio 1 al taladro siempre que el taladro esté libre.
7. Cinta 4
- Solo puede trasladar piezas al centro de pintura cuando este libre.
8. Intercambiadores
- Tienen que trasladar las piezas desde unas líneas a otras siempre que el punto de destino este libre.
  - Los intercambios se tienen que producir sin que se altere el orden de secuencia de fabricación de las piezas.

### 4.4. Modelo de subprocesos

A la hora de ponerse a pensar cual era la mejor manera de diseñar los graficet, se analizó la situación desde dos puntos de vista. El primero es de manera local, mirando que la planta estuviera formada por un conjunto de máquinas que necesitaban que algo las indicase que secuencia de movimientos tenían que seguir para fabricar bien cada pieza. El segundo punto de vista fue de manera global, observando que la planta estaba formada por 3 líneas de producción, y que cada línea estaba compuesta por 4 máquinas diferentes. Luego había que pensar la forma en que cada pieza pasara por todas las máquinas en el orden correspondiente. Además las 3 líneas debían funcionar a la vez, sin que aparecieran errores como fabricar piezas en la línea 1 cuando le tocaba a la 2. Este aspecto era muy importante, porque la bobina que se usa en cada línea tiene una calidad diferente, si hay un pedido de la calidad intermedia, (la que corresponde a la línea 2) y se hace con la de calidad baja (la línea 1) las piezas no serían válidas aunque estuviesen bien fabricadas.

Las encargadas de llevar las piezas de unas máquinas a otras son las cintas transportadoras. Cada línea de producción también tiene cintas entre las líneas para intercambiar piezas. Es importante que estén perfectamente sincronizadas para que saquen las piezas cuando la máquina ha terminado de realizar su trabajo y no lleven piezas mientras la máquina esté trabajando.

Todos los transportes se tienen que producir en el momento correcto. Además tiene que existir una prioridad entre las cintas de línea y las que se usan de intercambio entre líneas porque puede darse la situación de que lleguen a la vez dos piezas a un punto intermedio, una de la línea de producción y otra de un intercambio.

Otra cosa a tener en cuenta a la hora de pensar en cómo hacer los grafcet es que las maquinas se pueden romper y dejar de funcionar, pero la línea donde están dichas máquinas y el resto de la planta tienen que seguir funcionando. Eso es un problema, porque cuando se estropea alguna máquina, su grafcet queda paralizado y si el funcionamiento del resto de máquinas dependen de él o están relacionados de alguna manera, también dejarán de funcionar.

Para solucionar estos inconvenientes lo que se ha hecho ha sido crear los grafcet de forma independiente, mediante subprocesos. Esto quiere decir que para cada máquina de cada línea se ha hecho un grafcet diferente. Así, el corte de la línea 1 tiene un grafo para él, el taladro de la línea 2 otro, y así sucesivamente para todos los componentes de la planta, incluidas todas las cintas. El inconveniente de hacer grafcet por subprocesos, es que al final el número de grafcet es elevado, aunque de tamaño pequeño, ya que las acciones que tiene que hacer cada componente no llegan a más de 5 o 6. La ventaja es que da una flexibilidad muy grande al proceso de producción porque tal y como está diseñada la planta, esta forma de diseño permite ir intercambiando las piezas de unas líneas a otras sin ninguna restricción. Así, si se quiere cortar un pieza en la línea 1, taladrar en la línea 2 y pintar en la línea 3, sólo hace falta utilizar los grafcet de corte de la línea 1, de taladro de la línea 2 y de pintura de la línea 3 y los de las cintas que necesitan para cambiar las piezas de unas máquinas a otras.

Este sistema soluciona el problema de tener que parar la línea de producción cuando se avería alguna máquina, porque permite desviar las piezas a otras líneas, y luego volverlas a llevar a la línea. Esta sería una ruta alternativa si se rompe el taladro de la línea 1.

## 4. Control del Sistema

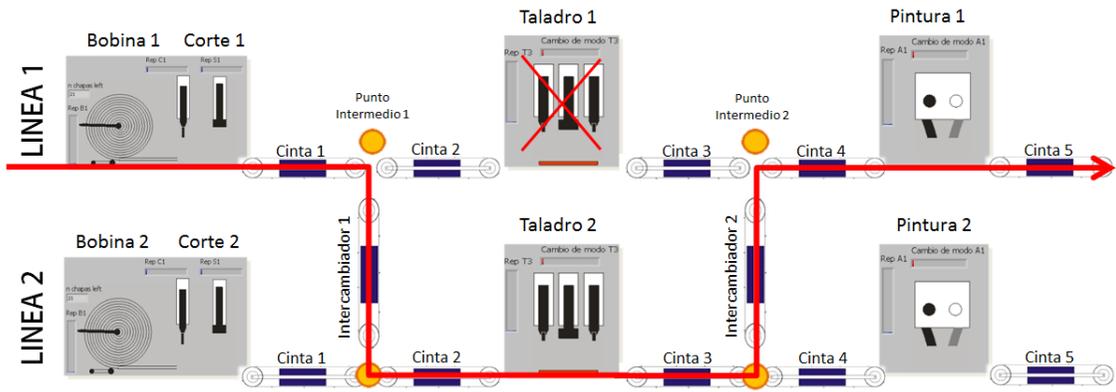


Ilustración 4.5. Ruta Alternativa

Otra ventaja de programar mediante subprocesos es que la producción es más rápida que de otra manera. Al estar programado de manera local, una pieza no tiene que esperar a entrar en la línea hasta que otra haya salido, sino que sólo tiene que esperar a que la máquina anterior acabe su trabajo.

### 4.5. Acciones y sensores en la planta

Cada máquina y cinta tiene un conjunto de acciones y sensores que son necesarios conocer para poder hacer la programación. A continuación se enumeran todos. Las tres líneas de producción tienen las mismas máquinas, así que usan el mismo tipo de accionadores y sensores.

	TIPO	Abrev.	DESIGNACIÓN	CARACTERÍSTICAS
BOBINA	Acción	Sb	Desenrollar Bobina	Desenrolla bobina.
		Nb	Nueva bobina	Cambia la bobina cuando se acaba.
	Sensor	In	Inicio	Marca el inicio de las operaciones
			Temporizador	La desbobinadora no lleva sensores, utiliza un temporizador para medir el tiempo debe estar activada la acción, en función de ese tiempo se desenrolla más o menos material.
CORTE	Acción	BSuj	Bajar sujeción	Baja el pistón de sujeción
		SSuj	Subir sujeción	Sube el pistón de sujeción
		BCor	Bajar corte	Baja la cuchilla de corte

		SCor	Subir corte	Sube la cuchilla de corte
	Sensor	Sar	Sujeción arriba	Se activa cuando el pistón de sujeción está completamente arriba
		Sab	Sujeción abajo	Se activa cuando el pistón de sujeción está completamente abajo
		Car	Corte arriba	Se activa cuando el corte está completamente arriba
		Cab	Corte abajo	Se activa cuando el corte está completamente abajo
TALADRO	Acción	BTa	Bajar taladro	Baja el taladro y la sujeción a la vez
		STa	Subir taladro	Sube el taladro y la sujeción a la vez
	Sensor	Tar	Taladro arriba	Se activa cuando el taladro está arriba
		Tab	Taladro abajo	Se activa cuando el taladro ha llegado abajo
PINTURA	Acción	Ppi	Pintar pieza	Se pone en funcionamiento la pintura
	Sensor	Fa	Fin de pintura	Marca cuando termina se termina de pintar la pieza.
CINTAS	Acción	Mov 1	Movimiento Cinta 1	Mueve la cinta 1
		Mov 2	Movimiento Cinta 2	Mueve la cinta 2
		Mov 3	Movimiento Cinta 3	Mueve la cinta 3
		Mov 4	Movimiento Cinta 4	Mueve la cinta 4
		Mov 5	Movimiento Cinta 5	Mueve la cinta 5
	INT1	Intercambiador entre líneas 1 y 2	Mueve intercambiador 1 desde la línea 1 a la 2	
	INT1	Intercambiador entre líneas 1 y 2	Mueve intercambiador 1 desde la línea 2 a la 1	
	INT2	Intercambiador entre líneas 1 y 2	Mueve intercambiador 2 desde la línea 1 a la 2	
	INT 2	Intercambiador entre líneas 1 y 2	Mueve intercambiador 2 desde la línea 2 a la 1	
	INT3	Intercambiador entre líneas 3 y 2	Mueve intercambiador 3 desde la línea 3 a la 2	
	INT 3	Intercambiador entre líneas 3 y 2	Mueve intercambiador 3 desde la línea 2 a la 3	

#### 4. Control del Sistema

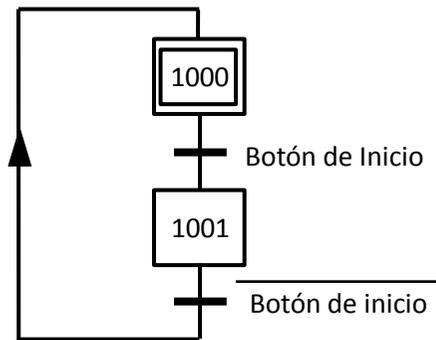
	INT 4 L3L2	Intercambiador entre líneas 3 y 2	Mueve intercambiador 4 desde la línea 3 a la 2
	INT 4 L2L3	Intercambiador entre líneas 3 y 2	Mueve intercambiador 4 desde la línea 2 a la 3
Sensor	SI11	Sensor intermedio 1 línea 1	Se activa cuando llegan piezas de la línea 1, cinta 1 o el intercambiador 1 y se apaga cuando se lleva la pieza la cinta 2 o el intercambiador 1
	SI12	Sensor intermedio 2 línea 1	Se activa cuando llegan piezas de la línea 1, cinta 3 o el intercambiador 2 y se apaga cuando se lleva la pieza la cinta 4 o el intercambiador 2
	SI21	Sensor intermedio 1 línea 2	Se activa cuando llegan piezas de la cinta 1 de las líneas 1 ó 3 o los intercambiadores 1 ó 3 y se apaga cuando se lleva la pieza la cinta 2 o los intercambiadores 1 ó 3
	SI22	Sensor intermedio 2 línea 2	Se activa cuando llegan piezas de la cinta 3 de las líneas 1 ó 3 o los intercambiadores 2 ó 4 y se apaga cuando se lleva la pieza la cinta 4 o los intercambiadores 2 ó 4
	SI31	Sensor intermedio 1 línea 3	Se activa cuando llegan piezas de la línea 3, cinta 1 o el intercambiador 3 y se apaga cuando se lleva la pieza la cinta 2 o el intercambiador 3
	SI32	Sensor intermedio 2 línea 3	Se activa cuando llegan piezas de la línea 3, cinta 3 o el intercambiador 4 y se apaga cuando se lleva la pieza la cinta 4 o el intercambiador 4

**4.6. Descripción del Sistema de Grafcet.**

4.6.1. Grafcet de Producción

4.6.1.1. Puesta en marcha.

Este grafcet simplemente sirve para poner en marcha o parar los grafcet de producción.



4.6.1.2. Bobina

Cuando se activa el botón de inicio, se activa la acción de desenrollar bobina (Sb) y la desbobinadora comienza a desenrollar material. La acción termina cuando ha pasado el tiempo marcado por el temporizador. Una vez hecho esto se tiene que espera a que el corte realice su trabajo y saque la pieza fuera a través de la cinta 1 para volver a desenrollar más material.

a. Secuencia de movimientos

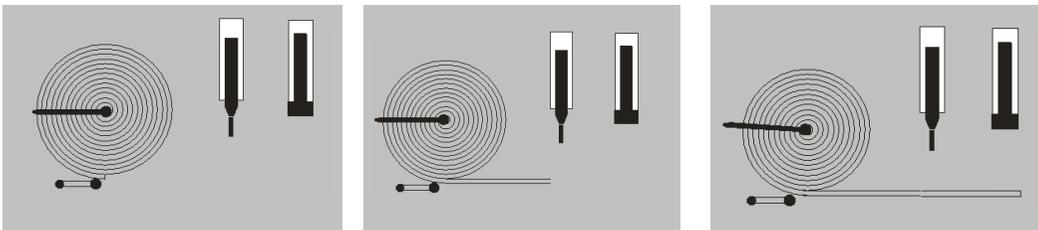


Ilustración 4.6. Secuencia Movimientos Desbobinadora

### b. Grafcet

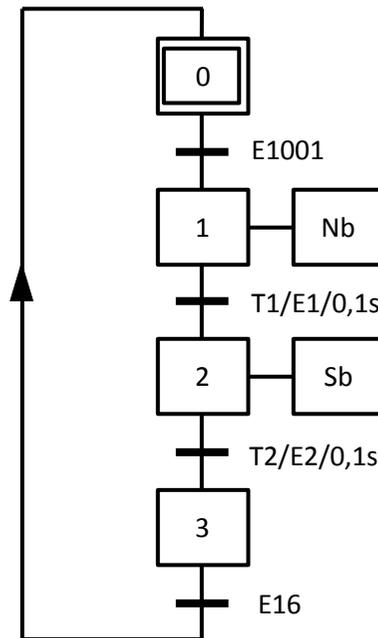


Ilustración 4.7. Grafcet Bobina

El proceso arranca cuando se activa la etapa 1000 de grafcet de marcha. Primero comprueba si hace falta bobina nueva (Nb). Sólo se cambia si hace falta. Luego desenrolla material (Sb) durante el tiempo que le marca el temporizador 2 (T2) y espera a que el grafcet de corte llegue a la etapa 16 (E16) para volver a empezar.

#### 4.6.1.3. Corte

El corte está formado por dos componentes, un cilindro que sirve de sujeción para el material y una cuchilla. Mientras que no se estén utilizando, ambos deben estar arriba, para comprobar que es así, los sensores de corte arriba (Car) y sujeción arriba (Sar) tienen que estar encendidos. Solamente puede ponerse en marcha cuando la desbobinadora ha terminado de desenrollar material. En ese momento, se activa la acción bajar sujeción (BSuj) y la sujeción baja hasta que se activa el sensor de sujeción abajo (Sab). Inmediatamente después se activa la acción de bajar el corte (BCor) hasta

que se activa el sensor de corte abajo (Cab), eso significa que la cuchilla ha llegado abajo y ha cortado el material. El siguiente paso es subir el corte con la acción de subir corte (SCor) hasta que se vuelva a activar el sensor de corte arriba (Car). Y por último subir la sujeción con subir sujeción (SSuj) también hasta que se encienda el sensor de sujeción arriba (Sar). Con esto la secuencia de corte estaría acabada, lo único que queda es espera a que la cinta 1 se lleve la pieza.

a. Secuencia de movimientos.

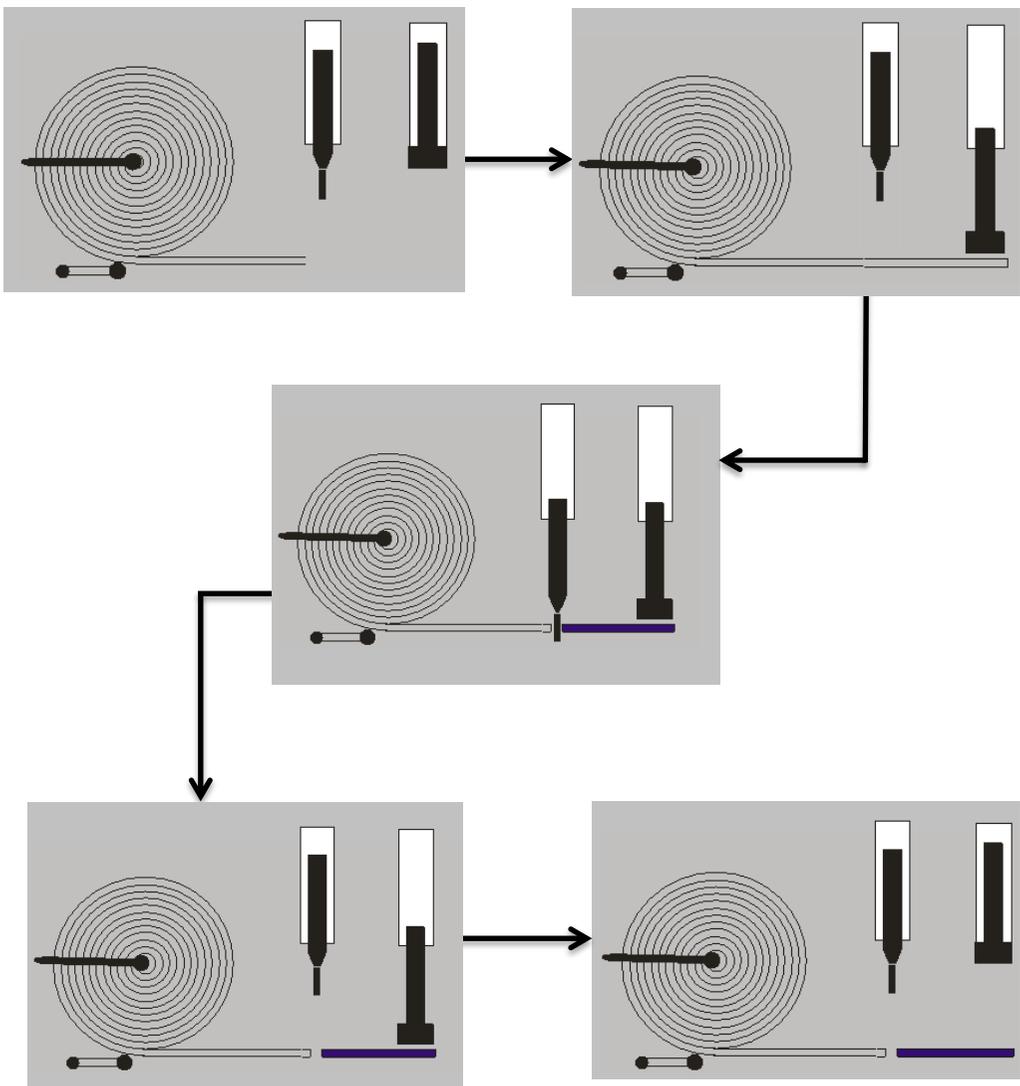


Ilustración 4.8. Secuencia de Movimientos Corte

### b. Grafcet

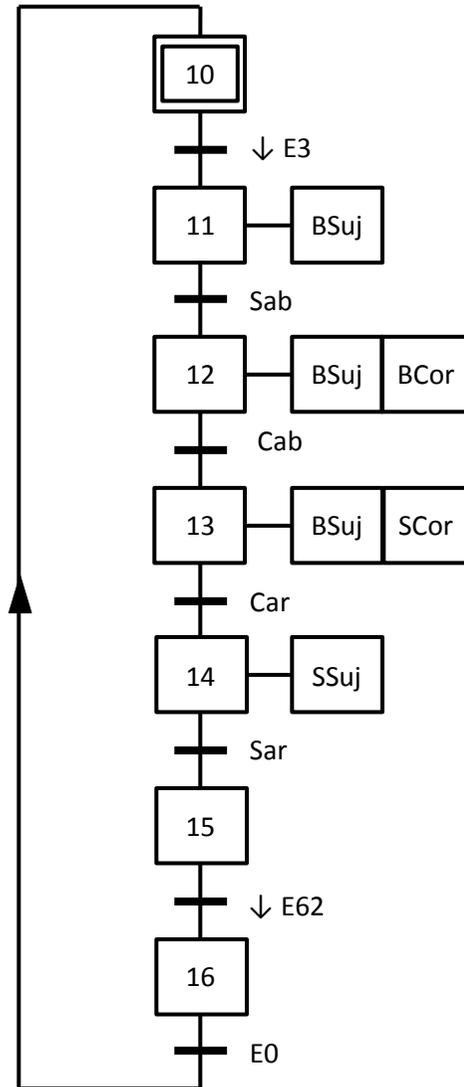


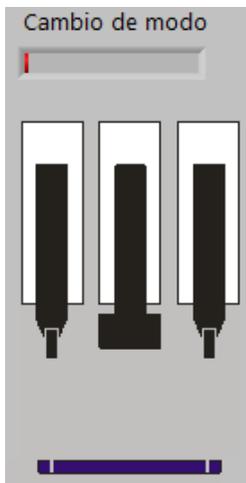
Ilustración 4.9. Grafcet Corte

El grafcet empieza cuando se activa la etapa 3 (E3) del grafcet de la desbobinadora, eso quiere decir que ya se ha desenrollado material. Realiza la secuencia desde la etapa 11 hasta la 14 y luego espera en primer lugar a que la etapa 62 haya terminado (E62), que significa que la cinta 1 se ha llevado la pieza, y después de que el grafcet de la desbobinadora saque material, vuelva a cero (E0), para volver a comenzar la secuencia.

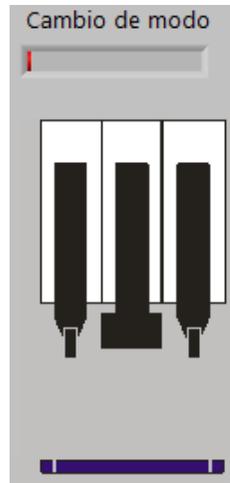
4.6.1.4. Taladro

El proceso del taladro comienza cuando llegan piezas procedentes de la cinta 2. Lo primero que hace el PLC es comprobar que está en el modo correcto, MODO 1 o MODO2, si no es así, se cambia. Una vez en el modo correcto, el taladro y la sujeción comienzan a bajar a la vez, para ello se activa la acción de bajar taladro (BTa) hasta que se enciende el sensor de taladro abajo (Tab), eso significa que ya están hecho los agujeros, asique sólo queda activar la etapa de subir taladro (STa) hasta que se enciende el sensor de taladro arriba (Tar) para finalizar el proceso y esperar a que la cinta 3 se lleve la pieza para poder empezar a trabajar con otra. Mientras tanto el taladro y la sujeción deben permanecer arriba todo el tiempo.

a. Tipos de Modos



Modo 1



Modo 2

b. Secuencia de Movimientos

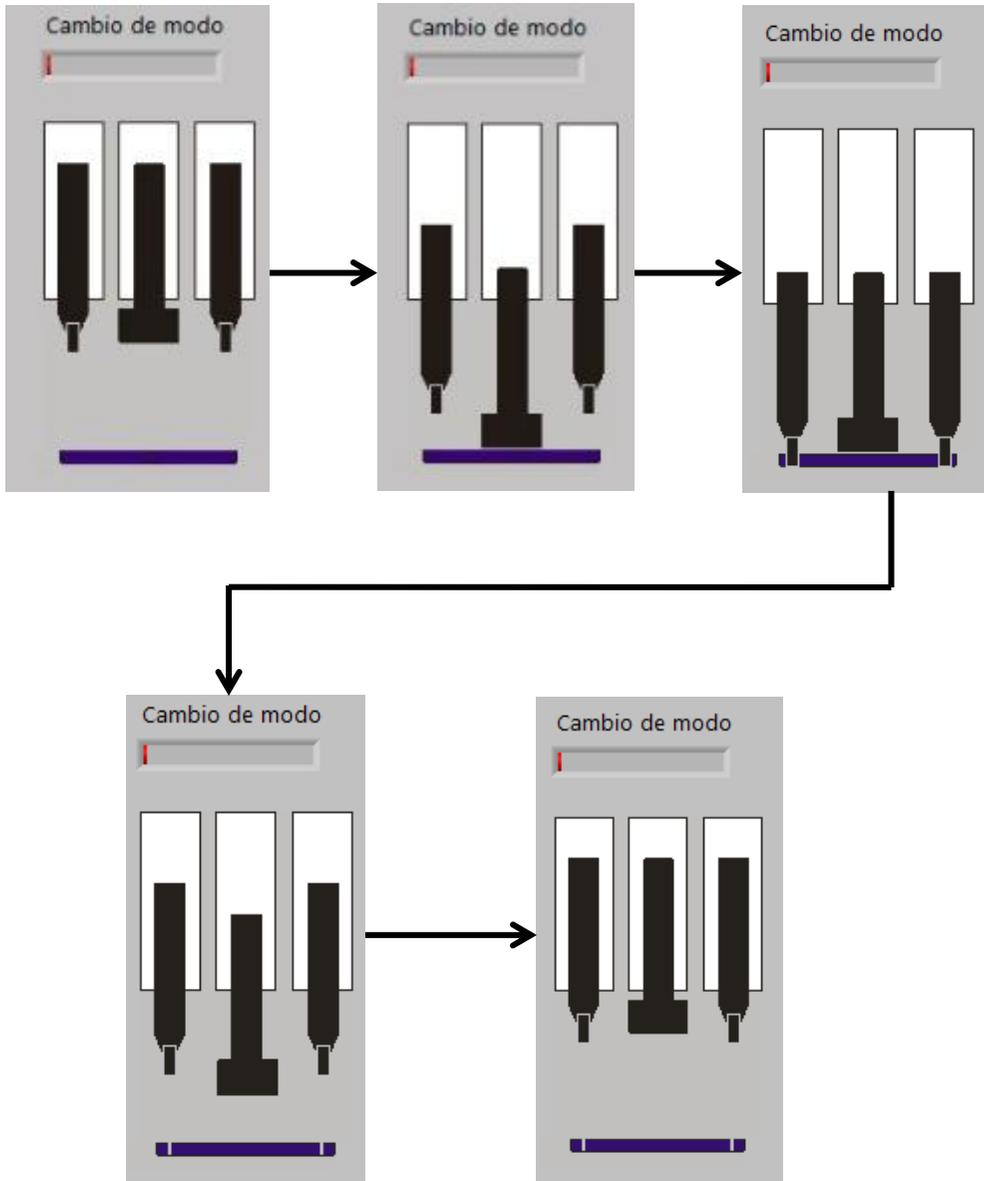


Ilustración 4.10 Secuencia de movimientos Talador

## c. Grafcet

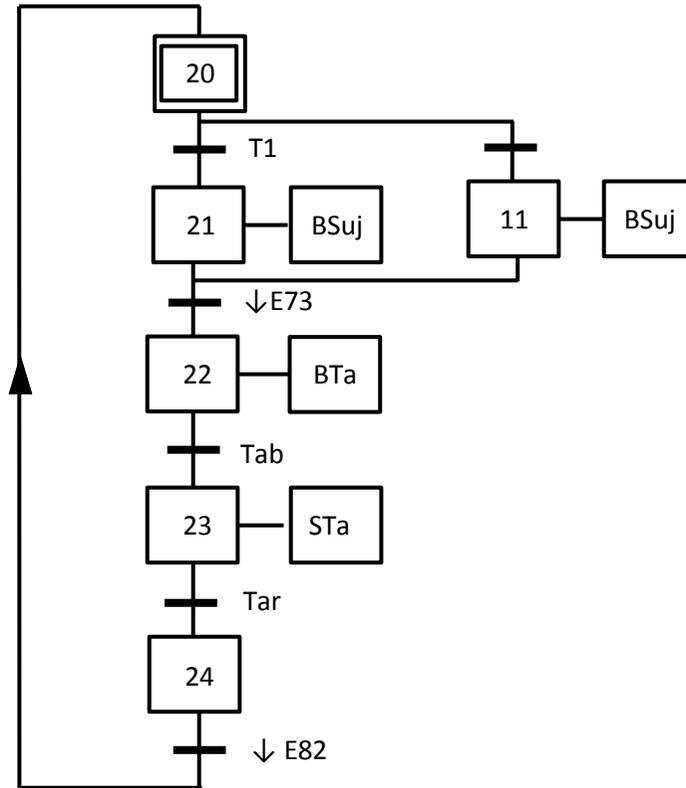


Ilustración 4.11. Grafcet Taladro

La secuencia empieza cuando la etapa 73 del grafcet de la cinta 2 ha terminado ( $\downarrow E73$ ), cuando la cinta 2 ha trasladado la pieza hasta el taladro. Se realizan las acciones de bajar y subir taladro (etapas 22 y 23) y por último espera a que la etapa 82 del grafcet de la cinta 3 termine ( $\downarrow E82$ ), lo cual significa que la cinta 3 se ha llevado la pieza.

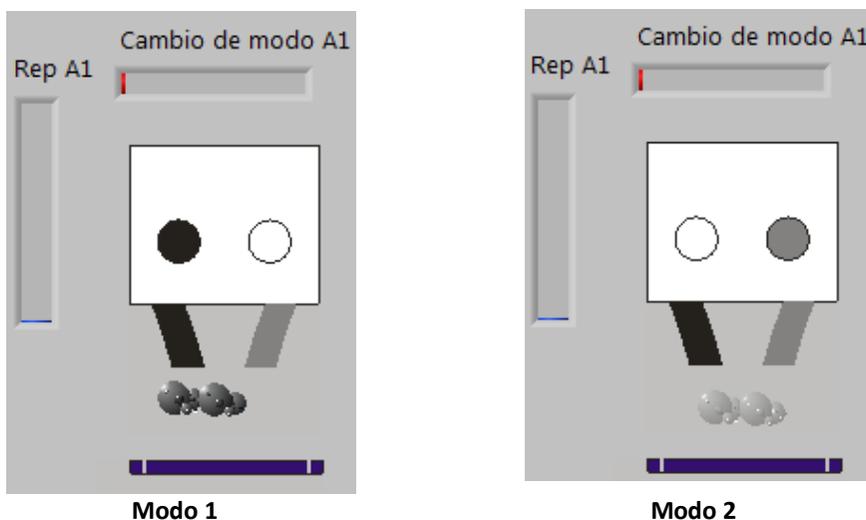
## 4.6.1.5. Centro de Pintura

La pintura es el último paso para fabricar la pieza. El proceso comienza cuando llega una pieza desde la cinta 4 e igual que en taladro, primero comprueba que la máquina está en el modo que corresponde.

## 4. Control del Sistema

Después entra la pieza en la zona de pintura y se activa la acción de pintar pieza (PPI) hasta que se enciende el sensor de fin de pintura (Fa), con lo que la pieza ya está acabada. Automáticamente se pone en funcionamiento la cinta 5 (Mov 5) y saca la pieza de la zona de producción.

### a. Modos de funcionamiento



### b. Secuencia de movimientos



Ilustración 4.12. Secuencia de Movimientos Pintura

## c. Grafcet

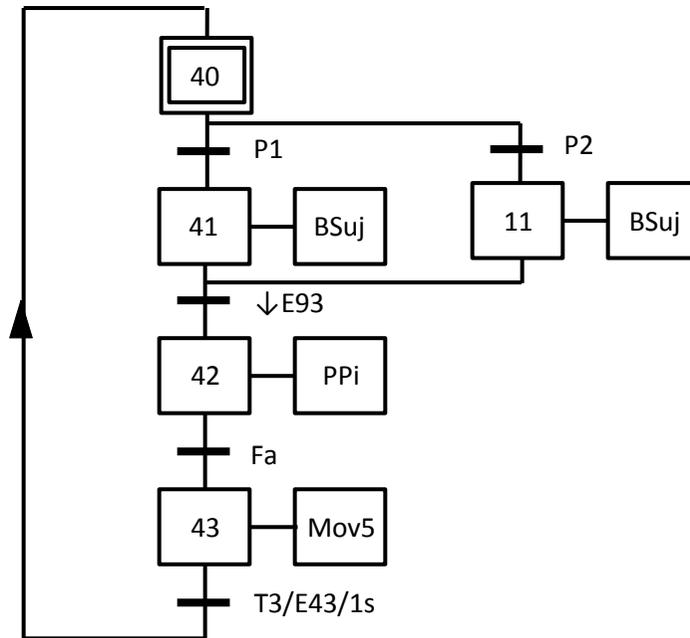


Ilustración 4.13. Grafcet Centro de Pintura

El grafcet primero comprueba que está el modo correcto. Después espera a que la etapa 93 del grafcet de la cinta 4 haya terminado ( $\downarrow E93$ ). Eso quiere decir que la cinta 4 le ha enviado una pieza. La etapa 42 (E42) realiza la secuencia de pintar la pieza hasta que se enciende el sensor Fa y después activa automáticamente la cinta 5 (E43). La última transición es un temporizador que marca el tiempo que tiene que estar funcionando la cinta 5 para que saque la pieza de la línea. Luego la secuencia vuelve a empezar.

4.6.1.6. Cintas.

Las cintas son las encargas de transportar las piezas desde unos puestos a otros.

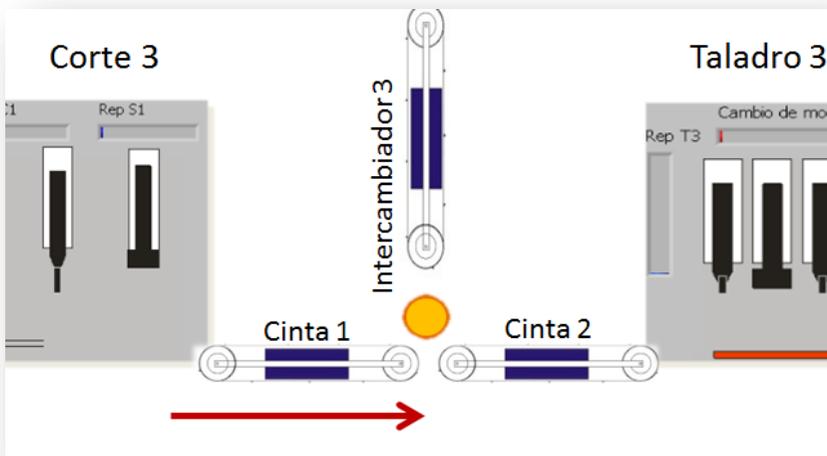
Cinta 1. Transporta las piezas desde el puesto de corte a la zona intermedia 1. Se pone en funcionamiento únicamente cuando el proceso de corte ha terminado su trabajo, activando movimiento 1 (Mov 1) y se para al

## 4. Control del Sistema

encenderse el sensor intermedio 1 (SI1). Existen dos condiciones que se deben cumplir para que la cinta se mueva:

1. Que el sensor intermedio esté apagado, eso quiere decir que no hay ninguna pieza en ese punto.
2. Que el intercambiador que lleva piezas a ese punto desde otra línea no esté activado, por lo tanto, no esté llevando piezas en ese momento.

a. Secuencia de movimientos



a. Grafcet

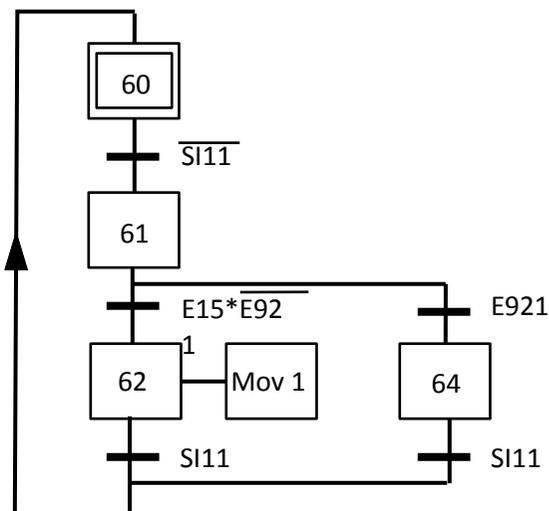


Ilustración 4.15. Grafcet Cinta 1

El graficet comienza en la etapa 60 (E60), comprueba que no hay ninguna pieza en el punto intermedio 1 (SI11). Si no hay, pasa a la etapa 61 (E61), donde tiene que elegir entre dos opciones. Si se está enviando pieza por el intercambiador, está activada la etapa 921 (E921), pasa a la etapa 64 (E64), donde espera a que se termine el intercambio y se active el sensor SI11 para volver a comenzar la secuencia. Si no hay intercambio (E921) y está activada la etapa 15 (E15), quiere decir que hay una pieza lista para ser enviada, se activa la etapa 62 (E62), y se pone en funcionamiento la cinta 1 hasta que se activa el sensor SI11, y habrá acabado la secuencia.

Cinta 2. Lleva las piezas desde el punto intermedio 1 hasta el taladro. Se activa cuando se enciende el sensor intermedio 1 (SI11), primero comprueba que el taladro está libre y el modo de funcionamiento del taladro es el correcto. Si es así, se activa el movimiento 2 (Mov 2) desplazando la pieza hasta el taladro, si no es así se espera a que quede libre. El sensor SI11 se apaga cuando la pieza se mueve, dejando libre el puesto para la siguiente.

### a. Secuencia de Movimientos

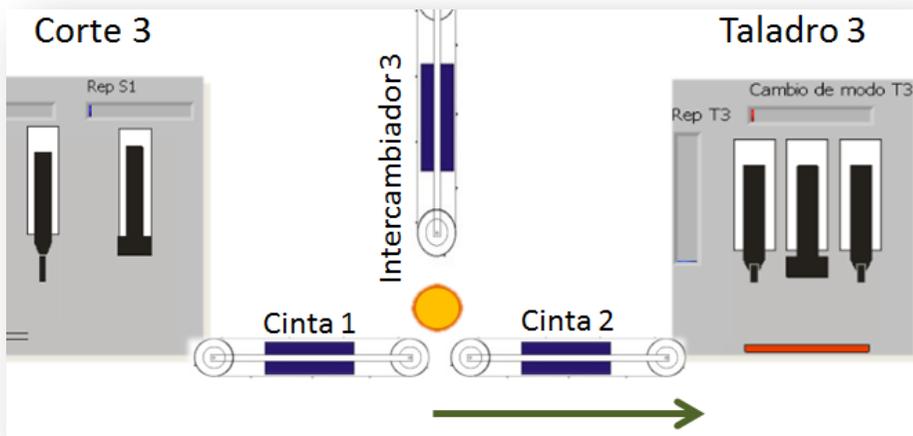


Ilustración 4.16. Secuencia Movimiento Cinta 2

### b. Grafcet

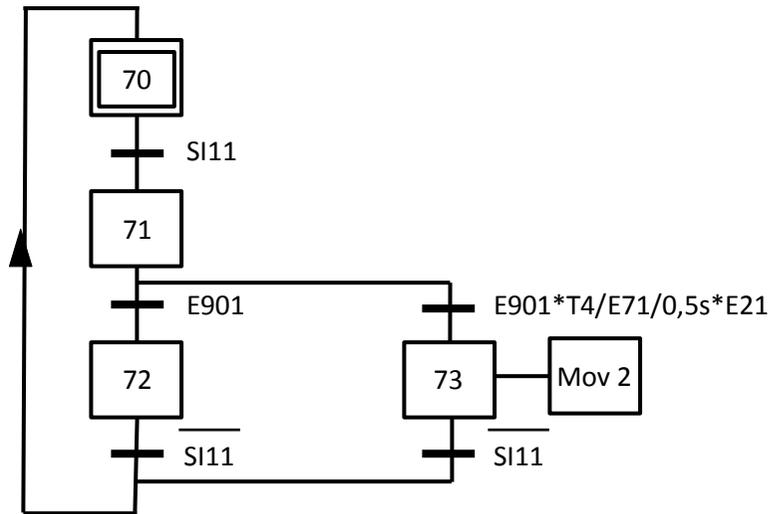


Ilustración 4.17. Grafcet Cinta 12

El proceso comienza cuando se activa el sensor SI11 porque ha llegado una pieza al punto intermedio 1. Después decide qué camino tomar. Si esta activada la etapa 901 (E901), se está enviando una pieza desde esta línea a otra, luego espera a que termine el proceso (se apague el sensor SI11) y vuelve a empezar. Si no está activa la etapa durante lo que marque el temporizador y la etapa 21 esté activada (E21) porque el taladro está libre, se activa la etapa 73 (E73) poniendo en funcionamiento la cinta 2. Una vez que el punto intermedio queda libre, (SI11), se vuelve al principio.

Cinta 3. El funcionamiento de la cinta 3 es similar al de la cinta 1. Solo se activa cuando el taladro ha terminado su secuencia, mediante la etapa Mov 3 hasta que se enciende el sensor intermedio 2 (SI2). Las restricciones para que pueda mover la cinta son las mismas que en el caso de la cinta 1.

## a. Grafcet

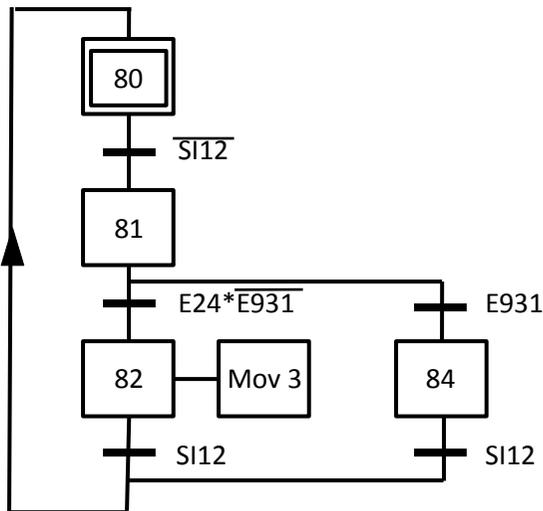


Ilustración 4.18. Grafcet Cinta 3

La estructura del grafcet es igual que la de la cinta 1, pero cambian los sensores. Ahora las piezas llegan al punto intermedio 2, así que se activa y desactiva el SI12, la cinta de intercambio es la 2, y la etapa asociada a esta es la 931 (E931). La otra acción que marca que se ponga en funcionamiento la cinta 3 es la etapa 24 (E24), etapa correspondiente al final del proceso del taladro.

Cinta 4. La cinta 4 mueve las piezas desde el punto intermedio 2 a la zona de pintura sólo cuando esta esté libre y en el modo correcto, en tal caso, se activa el movimiento 4 (Mov 4) hasta que se apaga el sensor intermedio 2 (SI2) y la cinta queda lista para el siguiente transporte.

### a. Grafcet

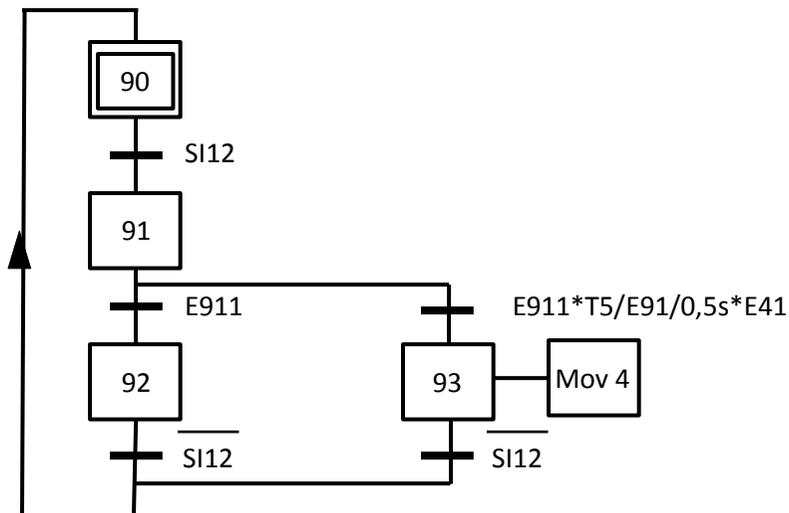


Ilustración 4.19. Grafcet Cinta 4

El grafcet de la cinta 4 es como el del 2, pero cambiando las transiciones como en el grafcet de la cinta 3.

Cinta 5. Es la encargada de sacar las piezas de la línea de producción. Se activa automáticamente cuando se activa el sensor de final de pintura (Fa), poniendo en funcionamiento el movimiento 5 (Mov 5) hasta sacar la pieza fuera.

Esta cinta no tiene grafcet propio, va incluido en el grafcet de la pintura.

#### 4.6.1.7. Intercambiadores.

Son cintas también, pero que transportan las piezas entre las líneas de producción. Se activan cuando el sensor intermedio desde donde tiene que salir la pieza está encendido, en ese caso, se pone en funcionamiento, el movimiento (INT). La misma cinta puede llevar piezas en ambos sentidos, por ejemplo, desde la línea 1 a la 2 y desde la 2 a la 1.

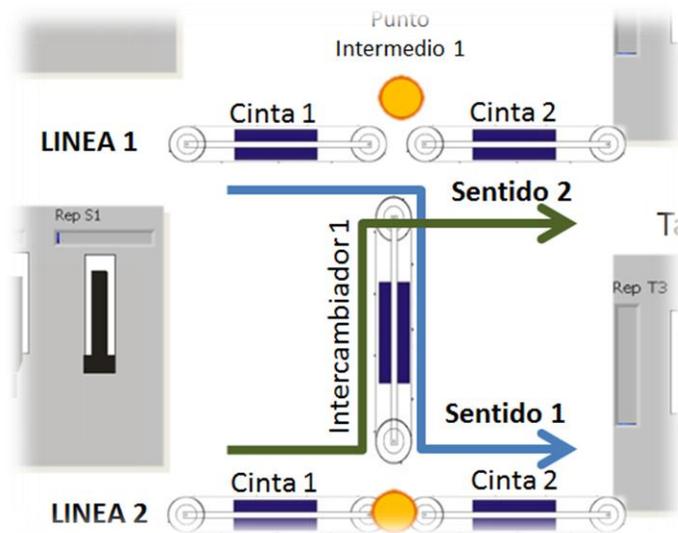


Ilustración 4.20 Intercambiador de Líneas

a. Grafcet Intercambiador 1

Intercambiador Línea 1 a Línea 2

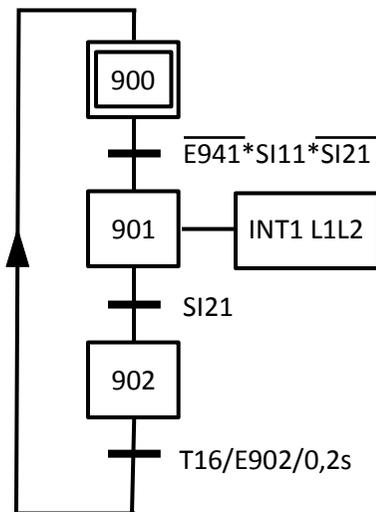


Ilustración 4.21. Grafcet Intercambio L1L

Intercambiador Línea 1 a Línea 2

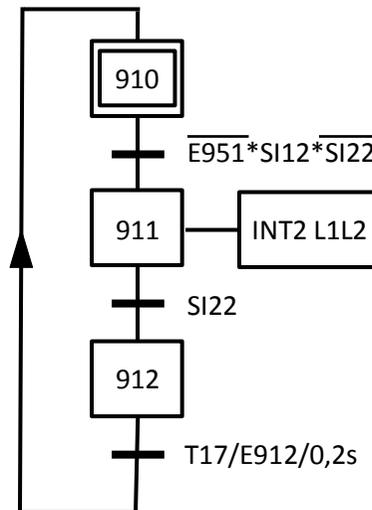


Ilustración 4.22. Grafcet Intercambio L2L1

El intercambio desde la línea 1 a la 2 sólo se produce si la etapa 941 (E941) no está activada. Esta etapa corresponde a la acción de llevar piezas desde la línea 3 a la línea 2, si el punto intermedio 1 de la línea 2 (SI21) está

libre y el de la línea 1 activado (SI11). Entonces se activa la etapa 901 (E901), llevando la pieza desde la línea 1 a la línea 2. La etapa 902 (E902) en principio no es necesaria, pero el software del autómeta da fallos si los graficet sólo tienen dos etapas asique la solución ha sido añadir otra con un temporizador de tiempo muy pequeño.

En el intercambio desde la línea 2 a la 1 ocurre al contrario, para que se active la etapa 921 (E921) es necesario que esté el sensor de intermedio 1 de la línea 1 apagado (SI11), y el sensor del punto intermedio 1 de la línea 2 (SI21) encendido. Para que el proceso se acabe se tiene que encender el sensor de la línea 1 (SI11).

### **4.6.2. Sistema de Prioridades.**

#### **4.6.2.1. Prioridades para las maquinas dentro de cada línea.**

Para que todas las maquinas funcionen coordinadas dentro de cada línea y no se junten piezas en la misma maquina hay que establecer un orden de prioridades, para ello, es necesario que la producción vaya en función del tiempo que tarda cada máquina en realizar su trabajo. La programación se ha pensado de tal manera que las maquinas o cintas sólo envíen piezas al siguiente proceso cuando este esté libre. Por ejemplo, la desbobinadora sólo puede desenrollar chapa siempre y cuando el corte no esté cortando ninguna pieza o la cinta 2 sólo puede enviar piezas al taladro siempre que este no esté ocupado. Si es así, la pieza se tiene que esperar a que el taladro termine de trabajar con la anterior para poder entrar. Cada máquina o cinta está condicionada en función de la máquina o cinta que la precede, la cinta 5 no tiene que espera a nada, y el centro de pintura tampoco, pero la cinta 4 si, se activa en función de que el centro de pintura esté libre o no. La cinta 3 le ocurre igual, sólo se puede activar cuando el punto intermedio 2 esté vacío, y así con el resto de máquinas. Con esto se evita que entren piezas en las maquinas mientras están trabajando con otra.

### 4.6.2.2. Prioridades entre los intercambiadores y las cintas de líneas.

Otro problema sobre prioridades que aparece en la planta es cuando hay intercambios y envíos de cinta en la misma línea. Puede ocurrir el caso que se envíen a la vez piezas por la cinta y el intercambiador y se junten en el punto intermedio, esto hay que evitarlo. Para ello se ha dado prioridad a los envíos de los intercambiadores frente a las cintas. Luego si está en proceso un intercambio, la cinta que está dentro de la línea donde llega la pieza queda bloqueada hasta que este se produce y la pieza ha salido del punto intermedio.

Un caso especial de intercambios son los que llegan a la línea 2. A ella pueden llegar piezas desde la línea 1 y desde la línea 3, por lo que puede ocurrir que lleven piezas a la vez desde ambas líneas. Para evitar que suceda esto se ha restringido el intercambiador 1. Así, desde la línea 1 sólo se pueden enviar piezas a la línea 2 siempre que no se estén enviando desde la línea 3. Se ha decidido que la prioridad sea del intercambiador 3 porque este lleva piezas de mayor calidad.

### **4.7. Flancos de bajada y restricciones con los acciones como transiciones.**

La forma de conseguir coordinar todas las máquinas y que no se junten piezas en el mismo sitio es condicionar unos grafcet a otros utilizando las etapas como transiciones. De esta manera, por ejemplo, si se quiere que la cinta 4 sólo envíe piezas a la pintura cuando no está trabajando, en la transición anterior al movimiento de la cinta del grafcet de la cinta 4 se pone la condición de que esté activada la etapa 41. Así, la etapa que acciona el movimiento en el grafcet de la cinta 4 sólo se activará cuando en el otro grafcet esté activada la etapa 41, etapa que corresponde al inicio de la secuencia, cuando en el centro de pintura no hay ninguna pieza.

En algunas de las transiciones de los grafcet, al lado de la etapa que está condicionando la transición aparece una flecha hacia abajo (↓). Esta flecha representa a un flanco de bajada que significa que la condición se cumple cuando la etapa finaliza.

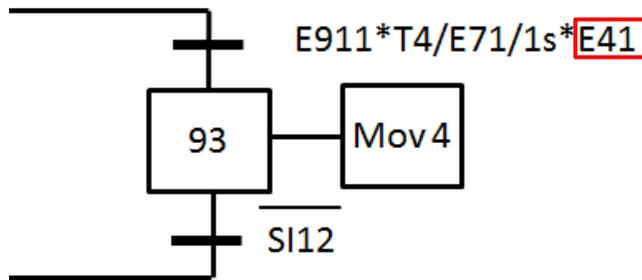


Ilustración 4.23. Restricción con etapas en las transiciones

#### 4.8. Programación con SYSWIN.

El software que utiliza el autómata se llama Syswin. Es un programa propio del fabricante del autómata. El modo de programación es gráfico. Cuando se abre aparece una zona donde están los menús y otra donde está la hoja de trabajo. Para crear un programa, se escoge el componente que se quiera colocar en la hoja de trabajo del menú de la izquierda y se arrastra hasta la hoja de trabajo.

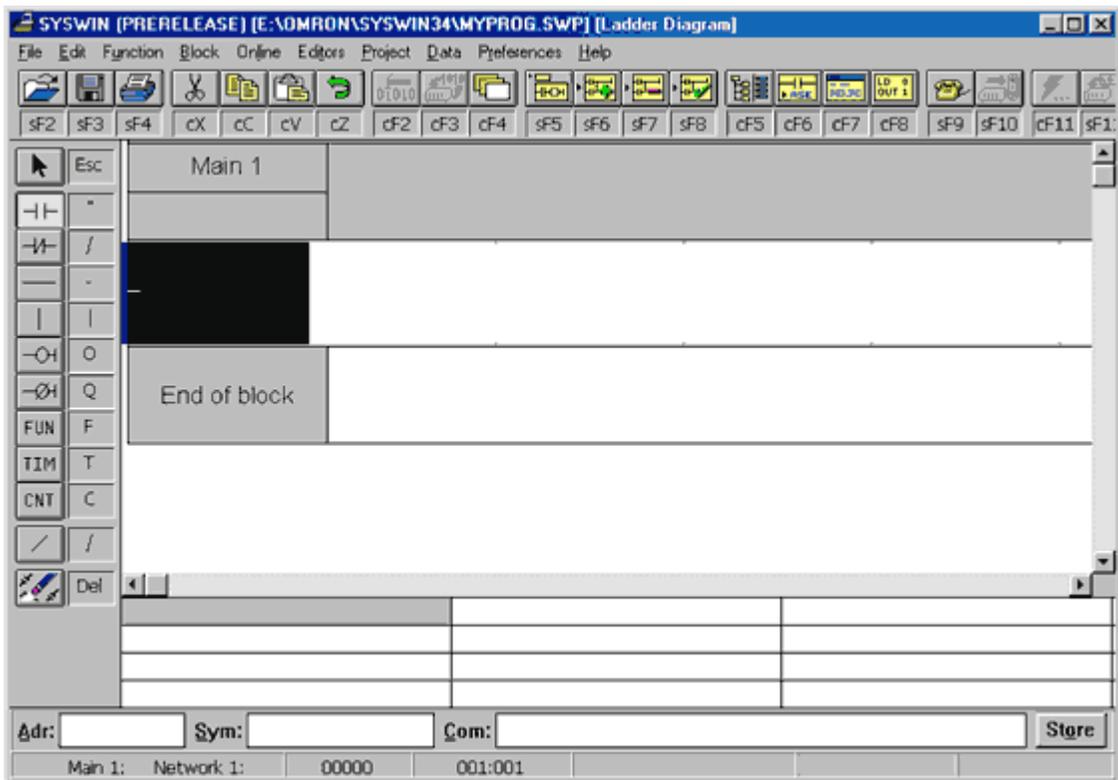


Ilustración 4.24. Imagen Programa SYSWIN

Todo se representa mediante contactos, tanto transiciones como etapas. Todos los contactos trabajan con direcciones de registros, luego hay que asignarles direcciones a todas las etapas, todas las transiciones y todas las acciones asociadas a las etapas.

El sistema a seguir para hacer los programas es fácil. Hay que ir metiendo etapa por etapa, y en cada etapa indicar cuál es la etapa anterior, las transiciones o condiciones que se tienen que cumplir para que se pase a esa etapa y la etapa que viene después, todo con contactos. Y a cada contacto asignarle una dirección de registro.

También hay que asociar a cada etapa la acción de le corresponde en otra línea.

Cada grafcet se va programando en una página diferente. Hay que tener cuidado en cómo se colocan las paginas porque el autómata va ejecutándolas de arriba abajo. Si hay etapas condicionadas de unos grafcet a otros, para que al autómata realice bien la condición, hay que poner primero el grafcet de la etapa y luego el grafcet donde está la transición con la etapa condicionada.



# *Supervisión de Fallos*

---

Con la parte de control de la planta acabada, ya se ha conseguido que todas las maquinas se muevan y realicen la secuencia de pasos correcta para poder fabricar la pieza y que el proceso de fabricación se lleve a cabo de una manera automática. El siguiente paso para automatizar la planta está relacionado con la supervisión de fallos. Una vez que la planta funciona correctamente, hay que controlar que no ocurran fallos, y si ocurren que se solucionen de la forma más rápida.

Estos fallos por lo general son roturas o paradas de máquina, que hacen detener las líneas donde se encuentran, retrasando la producción y por lo tanto creando pérdidas de tiempo y dinero. Este capítulo se centra en crear un sistema que detecte donde se ha producido el fallo y dé la mejor alternativa para seguir con la producción reduciendo el tiempo perdido al mínimo.

Dada la distribución de la planta, pueden existir dos soluciones o alternativas cuando se produce un fallo de una máquina. La primera es esperar a que se arregle la máquina y después seguir con la fabricación. La segunda consiste en desviar las piezas a otras líneas de la planta a través de los intercambiadores. La elección entre una u otra dependerá del tiempo que se tarde en realizar cada una, siendo la más óptima la que menos tiempo tarde. Desviar piezas a otras líneas tiene un problema, ya que es posible que las otras líneas estén trabajando también y por lo tanto habría que pensar que línea de fabricación tiene prioridad de paso.

Para gestionar este nivel y solucionar los problemas expuestos se ha creado un programa con el software Labview. El motor central del programa es el algoritmo de Bellman-Ford. Su función es buscar la opción que menos tiempo tardan las piezas en recorrer todas las maquinas teniendo en cuenta variables como el tiempo de funcionamiento de cada máquina o el de arreglo. En la parte visual, el programa muestra el tiempo que se va a tardar en fabricar las piezas y los puntos por donde pasa mediante una

## 5. Supervisión de Fallos

---

representación gráfica. El programa también proporciona la posibilidad de escoger que línea tiene prioridad de paso frente a las otras cuando ocurre algún fallo y ambas tienen que utilizar las mismas máquinas.

Siguiendo con el esquema de la estructura piramidal para automatización tipo CIM, el siguiente nivel es el nivel de supervisión. Este nivel es el encargado de gestionar la planta cuando aparecen fallos.



Ilustración 5.1. Pirámide de Jerarquías

El nivel recibe la situación del estado y posibles fallos desde el nivel 1 y emite órdenes de ejecución para solucionarlo. Igualmente recibe los programas de producción del Nivel 3 y le informa de las incidencias ocurridas en la planta.

### 3.2. Explicación del problema y variables.

El problema que se quiere solucionar en este capítulo es saber cuál es el mejor camino alternativo dentro de la planta cuando aparece un fallo en una línea y aún quedan piezas por fabricar, para que el tiempo de fabricación perdido por el fallo sea el mínimo posible. Los fallos son averías de las máquinas.

Cuando se avería una máquina, el nivel 1 no sabe qué hacer, ya que sólo ha sido programado para que ejecute una serie de instrucciones. Necesita a alguien que le indique que pasos tiene que seguir. El nivel de

supervisión es el encargado de indicarle al nivel de control como tiene que actuar cuando aparece un fallo. Para ello hay que conocer todas las opciones que hay y elegir cual es la mejor, en la que menos tiempo de ejecución se pierde. Analizando la planta, existen dos alternativas:

1. La línea puede esperar a que se arregle el fallo y continuar entonces por donde se había quedado. Esta solución es buena cuando el tiempo de arreglo es pequeño.
2. Desviar las piezas que aún quedan por fabricar hacia otras líneas. Esta es la mejor opción cuando se va a tardar mucho en arreglar el fallo o es urgente que se acabe con el pedido.

### 5.1.1. ¿Qué fallos pueden ocurrir en la planta?

Los fallos aparecen cuando alguna de las máquinas se avería y hay que arreglarla. En el simulador las maquinas o elementos de las maquinas que puede averiarse son:

- La bobinadora
- El corte
- La sujeción del corte
- El taladro
- El centro de pintura

### 3.2.2. ¿Qué variables interviene en el problema?

Para poder solucionar el problema primero hay que conocer las variables que forman parte de dicho problema. Como lo que se busca es optimizar el tiempo para que la línea esté parada el menor tiempo posible, todas las variables son tiempos que intervienen en el proceso de producción. Cada línea está compuesta por los siguientes tiempos:

- Tiempo de producción de cada máquina: es el tiempo que tarda cada máquina en fabricar su parte de la pieza.
- Tiempo de transporte de las cintas: tiempo que tarda cada cinta en transportar una pieza.
- Tiempo de transporte de los intercambiadores: tiempo que tardan los intercambiadores en enviar una pieza de una línea a otra.

## 5. Supervisión de Fallos

- Tiempo de arreglo de cada máquina: el tiempo teórico que te tardaría en arreglar la máquina.

### 3.2.3. Posibles soluciones. El camino óptimo

La opción de intercambiar piezas entre las líneas aporta mucha flexibilidad para solucionar el problema porque gracias a ellas se pueden desviar las piezas a las otras líneas. La consecuencia de ello es que aparecen caminos alternativos al camino normal que sigue cada pieza cuando se fabrica. Por ejemplo, si se avería el taladro de la línea 1, estas son tres opciones que se podrían seguir:

Opción 1. Esperarse a que se arregle el taladro y luego continuar por la línea 1. Este caso se dará cuando se tarde poco en arreglar la máquina, y por lo tanto el tiempo total de fabricación de las piezas y el arreglo sea menos que enviar las piezas a otras líneas.

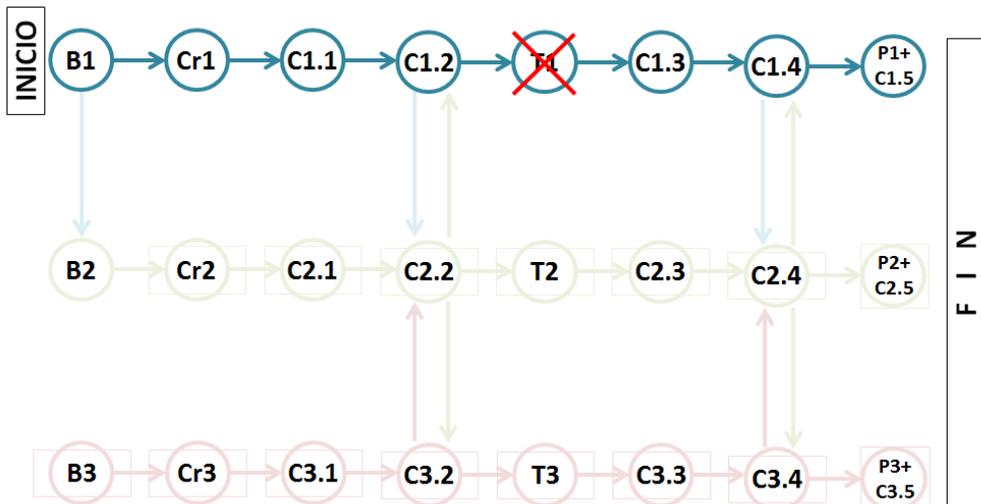


Ilustración 5.2. Opción 1 Camino

Opción 2. Desviar las piezas hacia la línea 2 y que se terminen de fabricar allí.

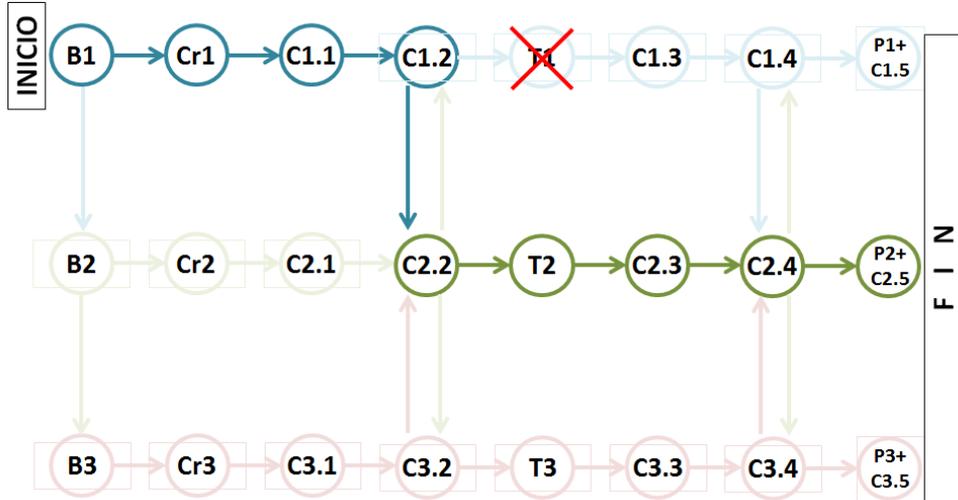


Ilustración 5.3. Opción 2 Camino

Opción 3. Enviar las piezas a través de los intercambiadores a la línea 3 para que se hagan los taladros allí y devolver las piezas a la línea 1 para que se pinten.

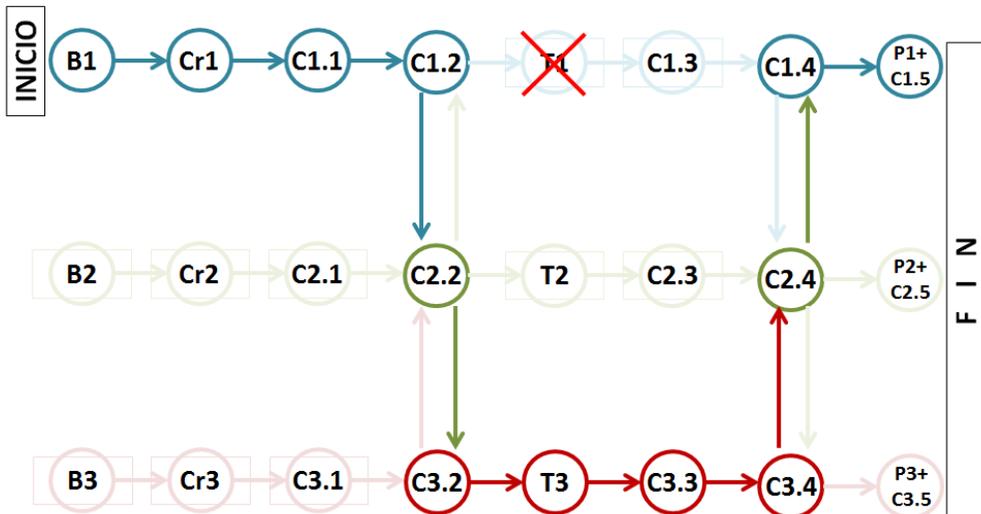


Ilustración 5.4. Opción 3 Camino

## 5. Supervisión de Fallos

La línea que tiene más combinaciones de caminos es la línea 1 porque se pueden empezar a fabricar piezas desde cualquier línea porque es la línea que trabaja con peor calidad de material, luego, y si hace falta, se pueden sustituir estas piezas por otras de mayor calidad. La línea 2 tiene una combinación menos, porque las piezas en este caso no pueden estar fabricadas con material de peor calidad, (línea 1), pero si se pueden fabricar con las de la línea 3. La línea que tiene menos combinaciones de caminos es la línea 3, ya que todas las piezas de este tipo de calidad solo se pueden empezar a fabricar desde aquí.

En las salidas, las piezas pueden salir por cualquier línea, no es necesaria que las piezas de calidad correspondiente a la línea 1 salgan por la línea 1, si no que pueden salir por las otras dos.

El número de combinaciones se puede visualizar gráficamente, representando cada línea mediante un árbol de posibilidades. Así se muestran los posibles caminos que se pueden seguir. Los círculos representan las máquinas o cintas y las flechas la dirección de recorrido de las piezas.

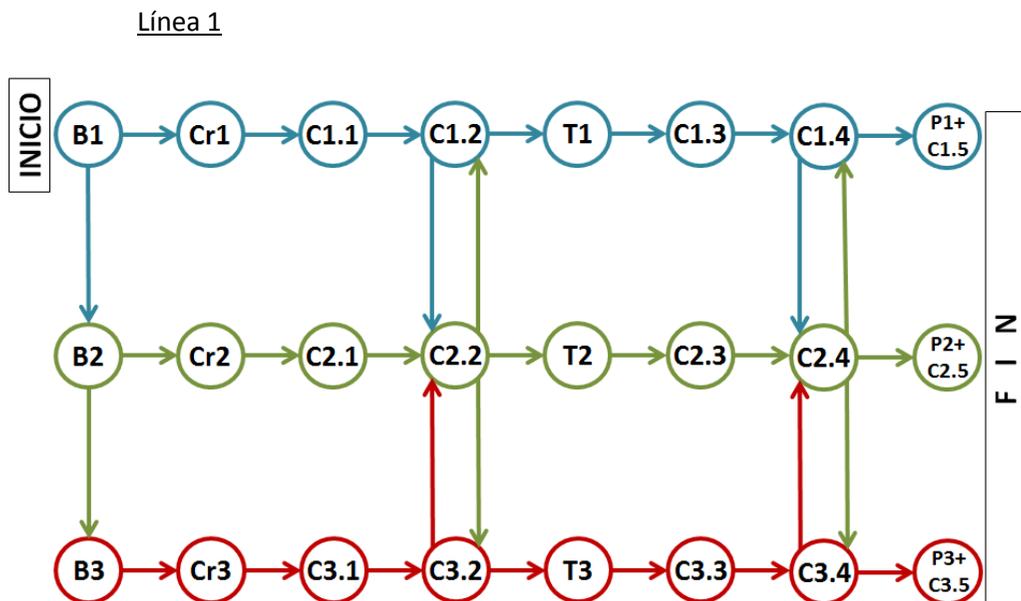


Ilustración 5.5. Árbol de Posibilidades Línea 1

Línea 2

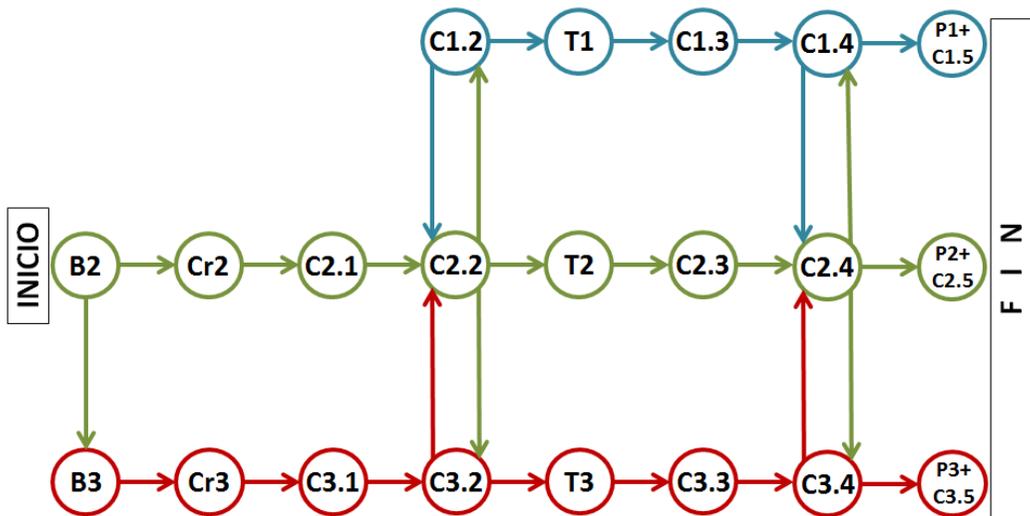


Ilustración 5.6. Árbol de Posibilidades Línea 2

Línea 3

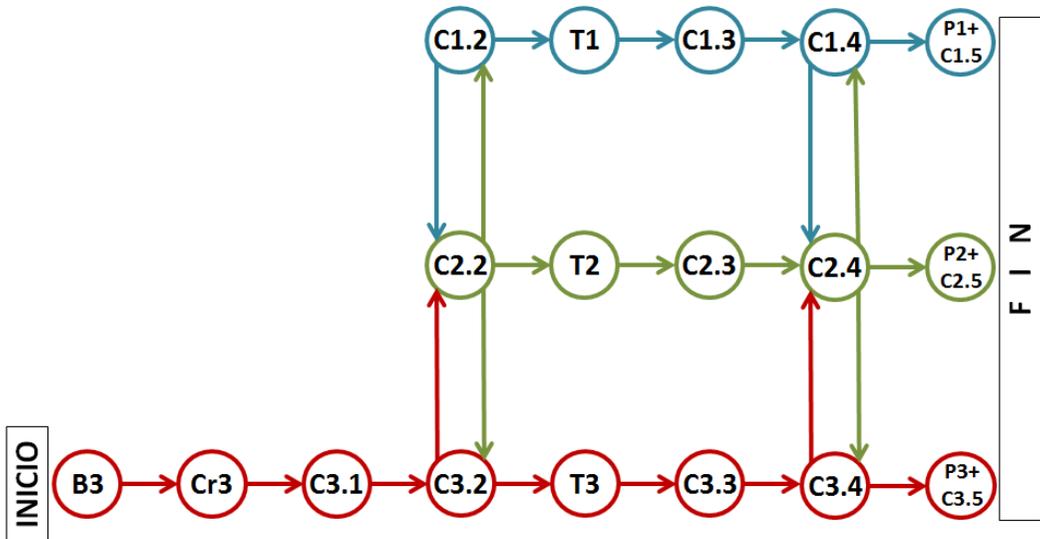


Ilustración 5.7. Árbol de Posibilidades Línea 3

Que se escoja un camino u otro dependerá del tiempo en que se tarde en fabricar la piezas por cada uno. El camino óptimo será en el que menos tiempo se recorra la secuencia de Desbobinadora-Corte-Taladro-Pintura, teniendo en cuenta la cantidad de piezas que aún quedan por fabricar en la

línea de la máquina que se ha averiado y el tipo de prioridad que esté activada. Ahora se necesita un instrumento que decida cuál de todos los caminos es el bueno. De ello se encarga el algoritmo.

### 3.3. Búsqueda de algoritmos.

El eje central de este nivel es el algoritmo, el cual se encarga de hallar cual es el camino en el que cada pieza tarda menos tiempo en recorrer la secuencia de fabricación entre todas las combinaciones siguiendo la secuencia de fabricación.

El algoritmo que se adecua a nuestras necesidades, es aquel que es capaz de sumar todos los tiempos que forman parte de cada camino y entre todos los caminos, indicar cual el más pequeño, el que tiene el menor tiempo. Hay muchos tipos de algoritmos que resuelven infinidad de problemas, por lo que hay que empezar acotando la búsqueda. Existe una familia de algoritmos denominada “path optim” o “Problema del camino más corto”, que se centran en solucionar el problema de encontrar dentro de un grafo el camino entre dos vértices o nodos, de tal manera que la suma de los pesos de las aristas que lo constituyen sea mínima. Los grafos son las representaciones graficas del problema, para este caso sirven de los diagramas de las líneas 1,2 y 3.

Dentro de la familia de los algoritmos “path optim” se han encontrado los siguientes algoritmos:

- Algoritmo de Búsqueda A estrella
- Algoritmo de Floyd-Warhsall
- Algoritmo de Johnson
- Algoritmo de Dijkstra
- Algoritmo de Bellman-Ford

Cada uno resuelve el problema de una manera diferente, pero eso no quiere decir que sea mejor a peor, simplemente que tarda más o necesitan más recursos.

### 5.2.1. Tipos de algoritmos.

#### Algoritmo de Búsqueda A estrella

El algoritmo “A estrella” es un algoritmo de búsqueda para grafos que encuentra el camino de menor coste entre un nodo inicial y otro final. Usa una función heurística (denotada  $f'(n)$ , es una aproximación a  $f(n)$ , función que proporciona la verdadera evaluación de un nodo) para determinar el orden en que la búsqueda visita cada nodo en el árbol. La mencionada función es la suma de otras dos funciones: una función que indica el coste del camino seguido hasta un cierto nodo (denotada  $g(n)$ ) y una estimación admisible de la distancia hasta la meta ( $h'(n)$ ). La función de evaluación resulta entonces

$$f(n) = g(n) + h'(n)$$

Empezando en un nodo inicial dado, el algoritmo expande el nodo con el menor valor de  $f'(n)$ . El algoritmo mantiene un conjunto de soluciones parciales almacenadas en una cola de prioridad. La prioridad asignada a un camino  $x$  viene determinada por la función  $f'(n)$ . El proceso continúa hasta que una meta tiene un valor  $f'(n)$  menor que cualquier otro nodo en la cola (o hasta que el árbol ha sido completamente recorrido).

La complejidad computacional del algoritmo está íntimamente relacionada con la calidad de la heurística que se utilice en el problema. En el peor caso, con una heurística de pésima calidad, la complejidad será exponencial, mientras que en el caso mejor, con una buena  $h'(n)$ , el algoritmo se ejecutará en tiempo lineal.

#### Algoritmo de Floyd-Warshall

El algoritmo de Floyd-Warshall compara todos los posibles caminos a través del grafo entre cada par de vértices. El algoritmo es capaz de hacer esto con sólo  $V^3$  ( $V$  representa el número de vértices) comparaciones. Esto es importante considerando que puede haber hasta  $V^2$  aristas en el grafo, y que cada combinación de aristas se prueba. Lo hace mejorando paulatinamente

## 5. Supervisión de Fallos

una estimación del camino más corto entre dos vértices, hasta que se sabe que la estimación es óptima.

El algoritmo comienza creando la matriz de distancias  $D$ . La iteración se realiza sobre nodos intermedios, es decir, para todo elemento de la matriz se prueba si lo mejor para ir de  $i$  a  $j$  es a través de un nodo intermedio elegido o como estaba antes. Esto se prueba con todos los nodos de la red. Una vez probados todos los nodos de la red como nodos intermedios, la matriz resultante da la mejor distancia entre todo par de nodos.

Se puede definir el  $\text{caminominino}(i,j,k)$  como:

$$\text{caminominino}(i,j,k) = \min(\text{caminominino}(i,j,k-1), \text{caminominino}(i,k,k-1) + \text{caminominino}(k,j,k-1))$$

$$\text{caminominino}(i,j,0) = \text{pesoarista}(i,j)$$

La fórmula ejecuta primero  $\text{caminominino}(i,j,1)$ , para todos los pares  $(i, j)$ , y se usa para después encontrar  $\text{caminominino}(i,j,2)$  para todos los pares  $(i, j)$ , el proceso continua hasta que  $k=n$ , y se encuentre el camino más corto para todos los pares de vértices  $(i, j)$ .

Para que haya coherencia numérica, Floyd-Warshall supone que no hay ciclos negativos (de hecho, entre cualquier pareja de vértices que forme parte de un ciclo negativo, el camino mínimo no está bien definido porque el camino puede ser infinitamente pequeño). No obstante, si hay ciclos negativos, Floyd-Warshall puede ser usado para detectarlos.

### Algoritmo de Johnson

$$D = \begin{bmatrix} 0 & 3 & 5 & 1 & \infty & \infty \\ 3 & 0 & \infty & \infty & 9 & \infty \\ 5 & \infty & 0 & 7 & 7 & 1 \\ 1 & \infty & 7 & 0 & \infty & 4 \\ \infty & 9 & 7 & \infty & 0 & \infty \\ \infty & \infty & 1 & 4 & \infty & 0 \end{bmatrix}$$

Su nombre viene de Donald B. Johnson, quien fue el primero en publicar la técnica en 1977. El algoritmo de Johnson es una forma de encontrar el camino más corto entre todos los pares

Ilustración 5.8. Matriz de Distancias

de vértices de un grafo dirigido disperso. Permite que las aristas tengan pesos negativos, si bien no permite ciclos de peso negativo. Funciona utilizando el algoritmo de Bellman-Ford para hacer una transformación en el grafo inicial que elimina todas las aristas de peso negativo, permitiendo por tanto usar el algoritmo de Dijkstra en el grafo transformado.

### **Algoritmo de Dijkstra**

El algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado desde un vértice origen al resto de vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

La idea subyacente en este algoritmo consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene. El algoritmo es una especialización de la búsqueda de costo uniforme, y como tal, no funciona en grafos con aristas de costo negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajarían el costo general del camino al pasar por una arista con costo negativo).

### **Algoritmo de Bellman-Ford**

El algoritmo de Bellman-Ford, genera el camino más corto en un grafo dirigido ponderado (en el que el peso de alguna de las aristas puede ser negativo). El algoritmo de Dijkstra resuelve este mismo problema en un tiempo menor, pero requiere que los pesos de las aristas no sean negativos.

El Algoritmo de Bellman-Ford es, en su estructura básica, muy parecido al algoritmo de Dijkstra, pero en vez de seleccionar vorazmente el nodo de peso mínimo aun sin procesar para relajarlo. Simplemente relaja

todas las aristas, y lo hace  $|V|-1$  veces, siendo  $|V|$  el número de vértices en el grafo. Las repeticiones permiten a las distancias mínimas recorrer el árbol, ya que en la ausencia de ciclos negativos, el camino más corto solo visita cada vértice una vez. A diferencia de la solución voraz, la cual depende de la suposición de que los pesos sean positivos, esta solución se aproxima más al caso general.

### 3.3.2. Algoritmo de Bellman-Ford.

Para hacer este trabajo se ha escogido el algoritmo de Bellman-Ford por ser el que mejor se ajusta a las necesidades requeridas.

El algoritmo sigue los siguientes pasos de manera iterativa:

- a. Inicializa todas las distancias o costos mínimos a infinito, excepto el nodo inicial que vale cero.
- b. Actualiza el paso anterior, aplicando la siguiente fórmula en cada nodo

$$d_x(y) = \min_v [c(x, v) + d_v(y)]$$

Donde:

- $D_x(y)$  es el costo del camino con menor coste del nodo  $x$  al nodo  $y$ .
  - $C(x,v)$  es el costo del predecesor hacia en nodo  $x$ .
  - $D_v(y)$  es el costo del predecesor hacia el nodo inicial.
- c. Se repite el paso anterior hasta que no se produzca ningún cambio de un paso a otro, con lo que se habrá concluido el algoritmo.

El algoritmos solo funciona con grafos dirigidos. Los grafos dirigidos son grafos normales donde las aristas que unen los vértices son flechas que indican la dirección en que se puede ir desde un vértice al siguiente. Esta es una de las razones por la que se ha escogido este algoritmo para hacer el trabajo, ya que es necesario que el grafo sea dirigido para que se pueda seguir la secuencia de fabricación correctamente.

La siguiente figura representa el grafo dirigido para la línea 1. Los vértices representan a cada una de las máquinas. Las flechas son las aristas que marcan la dirección que tienen que seguir las piezas desde un puesto al siguiente. Como la fabricación de las piezas tiene que seguir una secuencia específica, las flechas son las encargadas de que así ocurra. Por ejemplo, en la planta, cuando sale una pieza del corte de la línea 1 al siguiente lugar donde va es la cinta 1, luego el grafo tiene que tener una flecha desde el corte a la cinta.

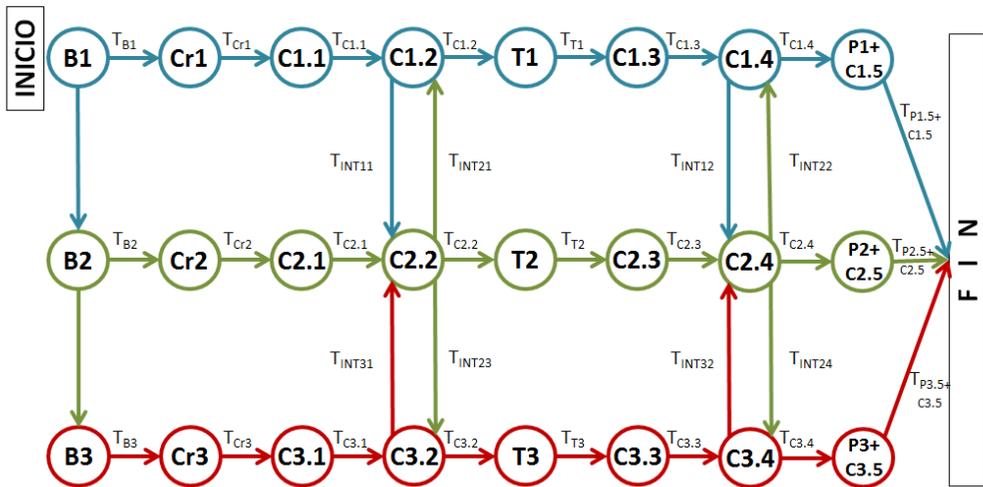


Ilustración 5.9. Grafo Dirigido Línea 1

Las letras que hay encima de cada flecha son los pesos. El peso es una magnitud que mide la distancia de un vértice al siguiente, puede ser cualquier cosa, dinero, masa, longitud... En este caso es el tiempo que tarda una pieza en ir desde una maquina a la siguiente, por ejemplo, la flecha que va desde B1 a Cr1 representa el tiempo que tarda la desbobinadora en sacar material y dejarlo listo para que el corte pueda trabajar con él. El peso es la variable que usa el algoritmo para hacer los cálculos. Para el caso de la planta, los pesos son los tiempos que tarda una pieza en ir desde una maquina a la siguiente.

## 5. Supervisión de Fallos

Una vez diseñado el grafo, para poder aplicar el algoritmo es necesario construir la matriz de decisiones. Esta matriz representa de una manera numérica los pesos del grafo y da la posibilidad de trabajar con ella matemáticamente para resolver el problema. Los ejes de coordenadas son los vértices que forman parte del gráfico. El eje vertical representa el origen, el vértice desde donde salen las flechas, y el eje horizontal el destino, el vértice donde llegan las flechas. Donde se cruzan ambos vértices dentro de la matriz es el lugar donde está el valor del peso. Por ejemplo, para un grafo pequeño como el siguiente, si se quiere saber cuál es el valor del peso que va desde el vértice 2 al 3, solo hay que buscarlo en la matriz.

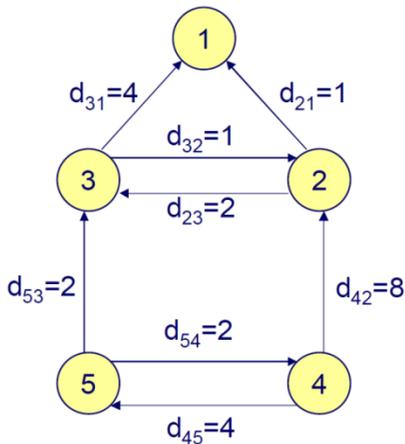


Ilustración 5.10. Grafo Ejemplo

<i>destino</i>	1	2	3	4	5
<i>origen</i> \					
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	1	0	2	$\infty$	$\infty$
3	4	1	0	$\infty$	$\infty$
4	$\infty$	8	$\infty$	0	4
5	$\infty$	$\infty$	2	2	0

Ilustración 5.11. Matriz de Decisión Ejemplo

Los pares de vértices que no están unidos en el grafo mediante flechas o el sentido es el contrario al de la dirección que marca la flecha no tienen ninguna relación, luego se supone que la distancia entre ellos es infinita. La distancia entre los mismos vértices de origen y destino es cero.

En la planta de producción cada línea tiene su matriz de decisión. La de la Línea 1 está formada por 26 variables, de las cuales 24 representan a las máquinas y las cintas de todas las líneas porque las piezas pueden ir por cualquier lugar, las otras dos variables son el inicio y el final. La matriz de la Línea 2, tiene 25 variables porque falta la de la desbobinadora 1, la matriz de la Línea 3, 24 variables porque no tiene ni la desbobinadora 1 ni la desbobinadora 2.

## 5. Supervisión de Fallos

Esta es la matriz de decisión de la línea 1, la mayoría de las casillas tienen el valor de infinito porque las maquinas que unen no tienen ninguna relación, como puede ser ir desde la pintura 3 al taladro 2. Solo tienen tiempos numéricos las que si tienen relación porque la pieza cuando se fabrica pasa de una a la otra.

distancias	INICIO	B1	CR1	C1.1	C1.2	T1	C1.3	C1.4	P1	B2	CR2	C2.1	C2.2	T2	C2.3	C2.4	P2	B3	CR3	C3.1	C3.2	T3	C3.3	C3.4	P3	FIN
INICIO	0	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
B1	Inf	0	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
CR1	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C1.1	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C1.2	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
T1	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C1.3	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C1.4	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
P1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1
B2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
CR2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C2.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C2.2	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf
T2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C2.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C2.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf
P2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1
B3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
CR3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C3.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
C3.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf	Inf
T3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf	Inf
C3.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf	Inf	Inf
C3.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1	Inf
P3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	5
FIN	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

**Ilustración 5.12. Matriz de Decisión Línea 1**

Cuando se tiene completada la matriz de decisión se aplica la fórmula del algoritmo siguiendo los pasos arriba indicados.

### **Algoritmo de Bellman-Ford mediante programación.**

En problemas que tienen pocas variables se puede calcular el algoritmo a mano, pero en cuando el número de variables y nodos comienza a crecer los cálculos se vuelven muy engorrosos porque hay que hacer muchas iteraciones. Es más rápido usar un programa informático para que

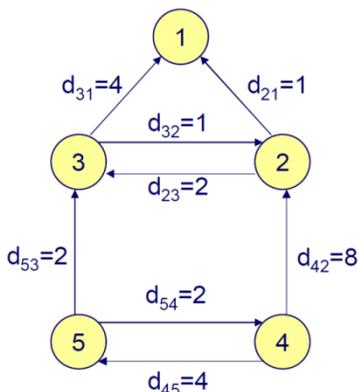
## 5. Supervisión de Fallos

haga los cálculos. La forma de programar este algoritmo informáticamente es bastante sencilla.

```

BellmanFord(Grafo G, nodo_origen s)
  for v ∈ V[G] do
    distancia[v]=INFINITO
    predecesor[v]=NIL
  distancia[s]=0
  for i=1 to |V[G]-1| do
    for (u, v) ∈ E[G] do
      if distancia[v]>distancia[u] + peso(u, v) then
        distancia[v] = distancia[u] + peso(u, v)
        predecesor[v] = u
    for (u, v) ∈ E[G] do
      if distancia[v] > distancia[u] + peso(u, v) then
        print ("Hay ciclo negativo")
      return FALSE
    return TRUE
  
```

El programa utiliza la matriz de decisiones para hallar la distancia mínima entre los vértices. Extrae de la matriz la fila correspondiente al vértice origen y la columna del vértice de destino y las suma por parejas. De todos los resultados escoge el más pequeño y lo coloca en otra matriz llamada matriz de distancias en la posición donde se cruzaban la fila y la columna utilizadas. Este proceso lo repite hasta que la matriz queda completada. En la matriz de distancias quedan reflejadas todas las distancias desde unos vértices a otros. Para el ejemplo de la matriz pequeña, si quiero saber cuál es el valor de la distancia desde el nodo 4 al 3 sumaría las fila 4 y la columna 3, y de todos los resultados me quedaría con el más pequeño, el cual colocaría en la casilla 4,3 de la matriz de distancias.



FILA 4	∞	8	∞	0	4
COLUMNA 3	∞	2	0	∞	2
Resultado	∞	10	∞	∞	6

destino /origen	1	2	3	4	5
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	1	0	2	$\infty$	$\infty$
3	4	1	0	$\infty$	$\infty$
4	$\infty$	8	$\infty$	0	4
5	$\infty$	$\infty$	2	2	0

**Matriz de Decisión**

destino /origen	1	2	3	4	5
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	1	0	2	$\infty$	$\infty$
3	2	1	0	$\infty$	$\infty$
4	8	8	6	0	4
5	4	3	2	2	0

**Matriz de Distancias**

Haciendo un resumen, el algoritmo coge la matriz de decisión y a partir de ella crea otra llamada matriz de distancias donde se reflejan todas las distancias de unos vértices a otros. En el trabajo el algoritmo se va a ejecutar el con software Labview.

### 3.4. Programación para el cálculo del camino óptimo

Toda la parte de programación de este nivel para automatizar la planta se ha hecho con el software Labview. La programación del nivel lo componen tres programas con funciones diferentes, pero que están relacionados y son necesarios para poder llegar al final de la solución. El primer programa es el de fallos, sirve para indicar cuando se ha averiado una máquina. El segundo es el de variables de tiempo y matriz de decisión, se encarga de crear la matriz de decisión a partir de las variables de tiempos. El tercero es el programa de cálculo del camino óptimo. Los tiempos que usa el segundo programa para crear la matriz dependen de la información que le envíe el programa de fallos. A la vez dicha matriz la utiliza el programa de cálculo de camino óptimo.

#### 5.3.1. Programa de Fallos

Este programa sirve para simular los fallos que pueden aparecer en la planta. Todos los fallos tienen que ver con averías de máquinas. Las maquinas o partes de máquinas que se pueden averiar son:

- La bobinadora.
- El corte.

## 5. Supervisión de Fallos

---

- La sujeción del corte.
- El taladro.
- El centro de pintura (acabado).

El panel está compuesto por tres hileras de botones, que representan las averías de las máquinas de cada línea de producción.

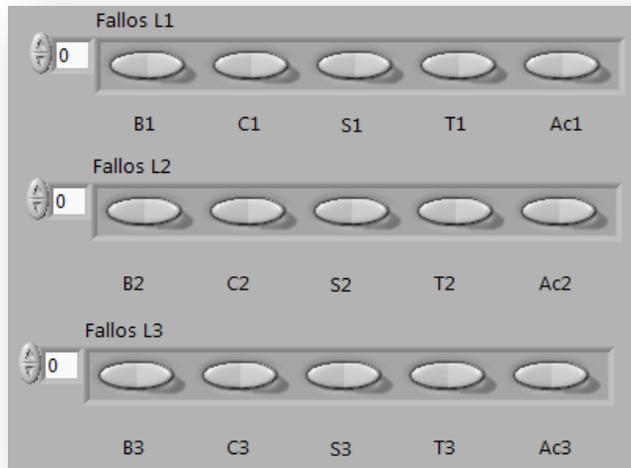


Ilustración 5.13. Panel de Fallos

### 5.3.2. Programa Variables de tiempo y matriz de decisión

El segundo programa sirve para crear la matriz de decisión que luego utiliza el algoritmo para realizar los cálculos.

Para poder crear la matriz, primero hay que hacer un análisis de las variables que la forman. Está claro que todas las variables tienen que ser tiempos porque lo que se quiere es encontrar cuál es el camino por el cual cada pieza tarda menos en fabricarse. La matriz tiene que estar formada por todos los tiempos por los que puede pasar la pieza desde que empieza a fabricarse hasta que llega al final de línea, o sea, los tiempos que tarda cada máquina en hacer su trabajo, los que tarda cada cinta en enviar la pieza desde un puesto al siguiente o el tiempo que se tardaría en arreglar la máquina cuando falla. Se han clasificado en los siguientes grupos:

- Tiempo de producción de cada máquina: es el tiempo que tarda cada máquina en fabricar su parte de la pieza.
- Tiempo de transporte de las cintas: tiempo que tarda cada cinta en transportar una pieza desde un puesto al siguiente.
- Tiempo de transporte de los intercambiadores: tiempo que tardan los intercambiadores en enviar una pieza de una línea a otra.
- Tiempo de arreglo de cada máquina: el tiempo teórico que se tardaría en arreglar la máquina.

Excepto las variables de arreglo, el resto de variables representan el tiempo por pieza, luego para calcular el tiempo de la producción que queda por fabricar hay que multiplicarlas todas por el número de piezas que aún faltan.

En el Front Panel de Labview están las variables de entrada, todos los tiempos que hacen faltan. El panel está dividido en cuatro zonas que representan a las tres líneas y a las cintas de intercambio. En las zonas de línea están, por un lado la variable de número de piezas que faltan por fabricar y por otro las variables de tiempo de funcionamiento de cada línea y cada cinta y el tiempo de reparación de cada máquina. En la parte inferior derecha está la zona para las variables de tiempo de los intercambiadores de líneas.

The image shows a control panel for time variables, divided into four sections: LINEA 1, LINEA 2, LINEA 3, and CINTAS INTERCAMBIO. Each section contains several variables with numerical input fields and spinners.

Variable	Value
Numero de piezas 1	1
B1A	0
B1F	4
Cr1SUJ	6
Cr1COR	5
Cr1F	1
CAMBIO T1 T1A	0
T1F	1
CAMBIO P1 P1A	0
P1F	1
C1.1	1
C1.2	1
C1.3	1
C1.4	1
C1.5	1
Numero de piezas 2	1
B2A	4
B2F	1
Cr2SUJ	4
Cr2COR	9
Cr2F	1
CAMBIO T2 T2A	0
T2F	6
CAMBIO P2 P2A	0
P2F	5
C2.1	1
C2.2	1
C2.3	1
C2.4	1
C2.5	1
Numero de piezas 3	1
B3A	5
B3F	1
Cr3SUJ	4
Cr3COR	4
Cr3F	1
CAMBIO T3 T3	1
T3F	9
CAMBIO P3 P3A	0
P3F	5
C3.1	1
C3.2	1
C3.3	1
C3.4	1
C3.5	1
O1.1	2
O2.1	2
O2.3	2
O3.1	2
O1.2	2
O2.2	2
O2.4	2
O3.2	2

Ilustración 5.14. Panel Variables de Tiempo.

El programa coge todas las variables para crear las matrices de decisión que luego utilizara el algoritmo para hacer los cálculos. Hay tres matrices de decisión diferentes, una para cada línea. No son iguales, ya que las combinaciones de camino de cada línea tampoco lo son.

En cada matriz tiene que estar incluidos los siguientes tiempos:

- Tiempos de funcionamiento de las máquinas y cintas de la línea de la matriz.
- Tiempos de arreglo de las máquinas de esa línea.
- Tiempos de funcionamiento de las otras dos líneas.
- Tiempos de arreglo de las máquinas de las otras líneas.
- Tiempos de intercambio de la línea de la matriz con las otras líneas.

Además, los tiempos de funcionamiento de máquinas, cintas e intercambio tienen que estar multiplicados por el número de piezas que faltan por fabricar para saber el tiempo total.

En la matriz, las variables de origen y destino son las máquinas y cintas y los tiempos son los pesos para ir de unas máquinas a otras. Todos los tipos de tiempos están incluidos de manera automática cuando se crea la matriz excepto los tiempos de arreglo, que se suman a los de funcionamiento sin estar multiplicados por el número de piezas que faltan por fabricar, solamente se suman cuando la maquina esté averiada.

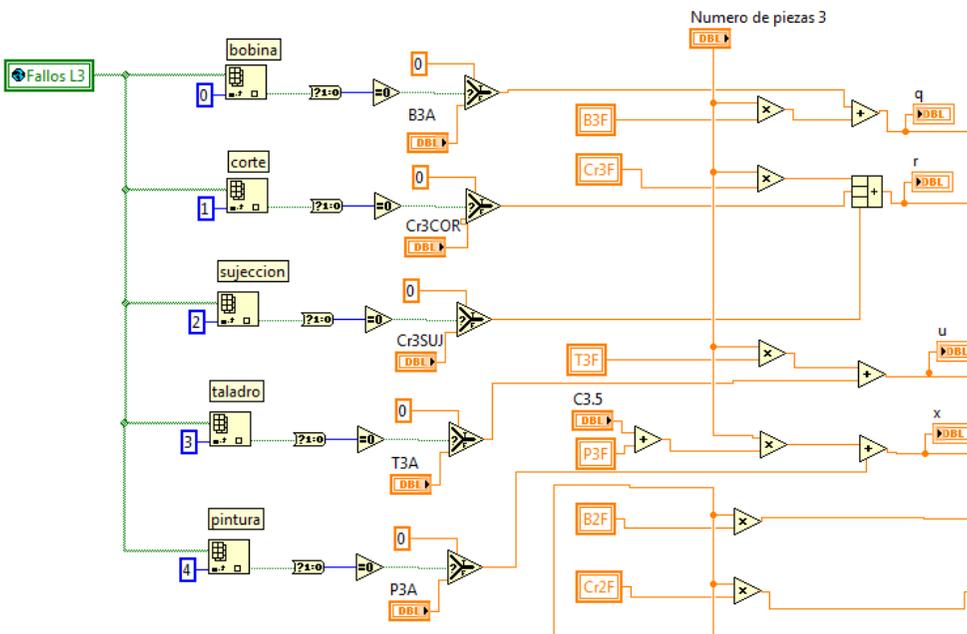


Ilustración 5.15. Programa Matriz de Decisión

De esta manera, ya quedan contempladas las dos opciones que hay cuando aparece un fallo dentro de la matriz de cada línea. Se pueden desviar las piezas a otra línea, mediante las variables de funcionamiento de las otras líneas o esperar a que se arregle la máquina, utilizando las variables de arreglo de la línea.

Para formar la matriz hay que colocar individualmente cada tiempo de cada máquina en el lugar que le corresponde. La matriz estará bien siempre y cuando los elementos que la forman reflejen lo siguiente:

- Para la línea que corresponde con la matriz, los tiempos de funcionamiento multiplicados por el número de piezas más los tiempos de arreglo de las máquinas que estén averiadas.
- Para el resto de líneas, los tiempos de funcionamiento multiplicado por el número de piezas que faltan aún más los tiempos de avería por si se estropean también.
- Los huecos donde las variables no tienen ninguna relación tienen que valer infinito.

### 5.3.3. Cálculo del camino óptimo

Este es el programa más importante del nivel de supervisión. Es el encargado de mostrar cual es el camino que tienen que seguir las piezas cuando se avería una máquina para que el tiempo perdido sea el mínimo. Para ello, a partir de los datos de la matriz de decisión creada en el programa de variables, el algoritmo de Bellman crea la matriz de distancias y con ella una gráfica que simula a la planta y las líneas donde se muestra el resultado, el camino óptimo.

#### 5.3.3.1. Programa Principal

Cada línea tiene un programa propio para mostrar cual es el camino óptimo cuando se avería algún componente de su línea. En todos, la pantalla principal está compuesta por varias partes. La primera es una gráfica que simula la planta, en ella los puntos representan a las máquinas y las cintas. Las aristas de colores son las uniones entre las máquinas y las de rojo el camino que tienen que seguir las piezas. La segunda es un indicador numérico que muestra el tiempo total que se va a tardar en fabricar las piezas que faltan. También un botón para activar la prioridad de paso y otro para encender o apagar la línea, según se esté usando o no. Por último tiene

dos indicadores para mostrar si las prioridades de las otras líneas están encendidas.

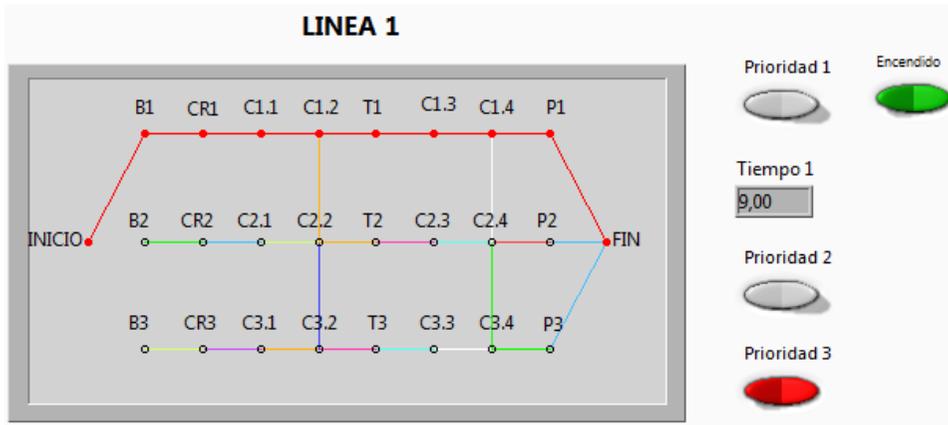


Ilustración 5.16. Panel Sistema de Optimización Línea 1

La parte de programación está compuesta por el programa principal y dentro de él, por varios subprogramas que se encargan de realizar los siguientes trabajos:

1. Representación gráfica de la planta. Para poder tener una noción visual de por qué maquinas tiene que ir la pieza.
2. Calculo de la matriz de distancias usando el algoritmo.
3. Calculo de prioridades de línea. Este programa marca cuál de las líneas tiene prioridad a la hora de fabricarse.
4. Extracción de la matriz de tiempos mínimos. La matriz de tiempos mínimos muestra los tiempos de las maquinas por donde pasa la ruta que ha dado el algoritmo. Esta matriz luego se utilizara para el sistema de prioridades de línea.
5. Cálculo del tiempo de fabricación y los puntos por donde pasa el camino.
6. Representación gráfica del camino que tienen que seguir las piezas.

### Representación Programa Principal

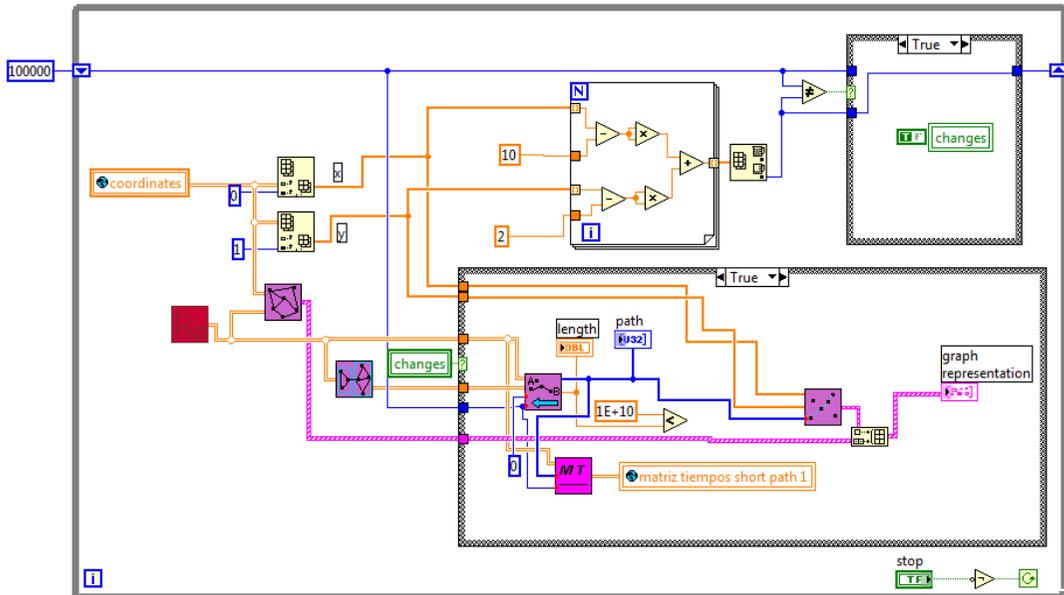


Ilustración 5.17. Programa Optimización de Caminos

El programa principal ejecuta toda la programación, organiza a los subprogramas, une las variables que tienen relación entre ellos y muestra los resultados en el panel principal.

Tiene dos variables de entrada, la matriz de coordenadas y la matriz de decisión que se creó en programa de variables. La matriz de coordenadas es una matriz que indica la posición de las máquinas dentro de la gráfica que simula a la planta. Está vinculada al programa mediante una variable global. Las coordenadas de cada punto tienen que estar ordenadas para que coincidan con el orden que tienen las variables en la matriz de decisión, porque cada coordenada representa a una máquina en el gráfico. La segunda variable de entrada es la matriz de decisión del programa de variables y matriz, forma parte del subprograma de prioridades de línea.

El funcionamiento del programa es el siguiente:

El programa lee los datos de las coordenadas, los descompone separando la X por un lado y la Y por otro y halla la distancia mínima desde todos los punto al punto final, que en este caso es el (10,2). De todos los valores calculados coge el mínimo, y el puesto que ocupa este dentro de la matriz de coordenadas lo compara con otro puesto al azar. Si el valor no coincide, (eso es lo que se quiere) se activa la variable de changes. La cual activa la otra parte del programa.

La otra parte del programa solo se activa si la variable de “changes” está activada. Si no es así, el programa no hace ningún cálculo. Esta parte está formada por los subprogramas arriba indicados. El cálculo comienza en el subprograma de prioridades de línea desde donde sale la matriz de decisión modificada. Con la matriz de decisión modificada y la matriz de coordenadas se crea la gráfica que representa a la planta. Paralelamente, usando también la matriz de decisión modificada, el subprograma del algoritmo de Bellman crea la matriz de distancias. A partir de la matriz de distancias, con el subprograma de cálculo de tiempo minino y puntos, halla el tiempo que van a tardar en fabricarse las piezas y se indica por qué coordenadas va el camino que ha dado el algoritmo. Conociendo los puntos por donde va el camino, por un lado se crea la matriz de tiempos mínimos, que luego utilizan los programas de prioridades, y por otro se dibuja la gráfica con el camino, trabajo que realiza el subprograma de dibujo del camino. Por último se une esta grafica con la gráfica que representa a la planta y el programa queda terminado.

### 5.3.3.2. Descripción de los Subprogramas

#### **Prioridad de línea.**

Este programa está pensado para poder decidir que línea tiene prioridad cuando una maquina se avería y la producción de esa línea tiene que pasar a otra línea que a la vez también está trabajando. Por ejemplo, si se estropea alguna máquina de la línea 3 y hay que desviar las piezas que aún quedan por fabricar a la línea 2, hay dos opciones:

1. Que se termine de fabricar las piezas que aún quedan de la línea 2, y luego pasar por esta línea las de la línea 3.

2. Parar la producción de la línea 2 para poder fabricar las de la línea 3, y una vez terminadas, continuar con las de la línea 2.

El criterio para seleccionar una línea u otra como línea principal de producción lo tendrá que decidir el operario en función de que línea tiene preferencia para que se termine de fabricar antes. Este programa realiza los cálculos para obtener la matriz de decisión modificada que luego utilizara el algoritmo para obtener el camino óptimo en cada línea. Dependiendo de la línea que esté como principal, el programa da unos resultados u otros. La línea principal es la que tiene preferencia y va a pasar primera en cualquier línea. Si las otras líneas están trabajando con sus piezas, y las piezas de la línea principal van por sus líneas, estas tendrán que esperarse hasta que la línea principal haya terminado para continuar con su producción.

Al existir tres líneas, hay que establecer una jerarquía para decidir cuál pasa primero, segundo y tercero. El criterio que se ha seguido ha sido clasificarlas por el tipo de calidad del material, pasando primero el que mayor calidad tiene. La línea que pasa como preferente se puede elegir a través de los interruptores del panel principal, dependiendo de la línea que este como principal, la segunda y la tercera cambian. La razón de esto, es que si nos ponemos en el peor de los casos, cuando se averían dos líneas y hay que desviar toda la producción de las tres líneas a la línea que funciona, ¿Cuál debería pasar primero? ¿Y segundo? ¿Y tercero? El siguiente criterio lo decide.

1. Si esta como línea principal la línea 3, la de mayor calidad, como segunda línea está la línea 2 y como tercera la línea 1.
2. Si como primera línea está la línea 2, como segunda esta la línea 3 y como tercera la línea 1.
3. Si como línea principal está la línea 1, como segunda estaría la línea 3 y como tercera la línea 2.

Cada línea de producción tiene su programa propio porque cada línea tiene unas preferencias diferentes.

El programa trabaja sumando matrices en función del botón de prioridad que este encendido, creando una matriz de decisión modificada, que es la que utiliza el algoritmo para hacer los cálculos. La matriz de decisión modificada está formada por la suma de la matriz de decisión de la

línea que corresponde y las matrices de tiempos mínimos de las otras líneas. La matriz de tiempos mínimos es una matriz de igual dimensión que la matriz de decisión, dentro de ella todos los elementos son cero excepto los lugares por donde va el camino que ha dado el algoritmo de la línea a la que corresponde, que muestra el valor del tiempo en cada puesto.

El programa puede funcionar con una línea encendida, con dos líneas o con las tres. Dependiendo del caso, dará unos resultados u otros. Si están encendidas las tres líneas, el programa funciona así:

La línea que está como preferente utiliza la matriz de decisión que ha salido del programa de variables, la matriz no se modifica, pasa directamente al algoritmo para que haga los cálculos.

La línea que esta segunda en la jerarquía, el programa de prioridades suma la matriz de tiempos mínimos de la línea de preferencia y la matriz de decisión que ha proporcionado el programa de variable para la segunda línea. El resultado es la matriz de decisiones modificada, en cuyos puntos donde se juntan las dos líneas se han sumado los tiempos de funcionamiento de ambas para saber el tiempo que se tarda en fabricar las dos líneas.

Para la línea que está tercera en la jerarquía, el programa suma las matrices de tiempos mínimos de las otras dos líneas más la matriz de decisión de esta, creando la matriz de decisión modificada. E igual que para la segunda línea, en los puntos donde coinciden los caminos de las tres líneas, se suman todos los tiempos para saber cuánto se tardaría en fabricar la producción total por ahí.

Por defecto esta la línea 3 como línea principal.

### **Gráfica de simulación de la planta.**

Este subprograma sirve para crear la gráfica que simula la planta. La grafica está formada por tres hileras de puntos horizontales, que representan a cada una de las maquinas o cintas de las tres líneas. La situación de los puntos dentro de la gráfica es la que se ha puesto en la matriz de coordenadas. Todos los puntos están unidos con aristas de colores.



la Y de cada coordenada de la matriz y se unen mediante un cluster para formar la gráfica. El resultado es el siguiente:

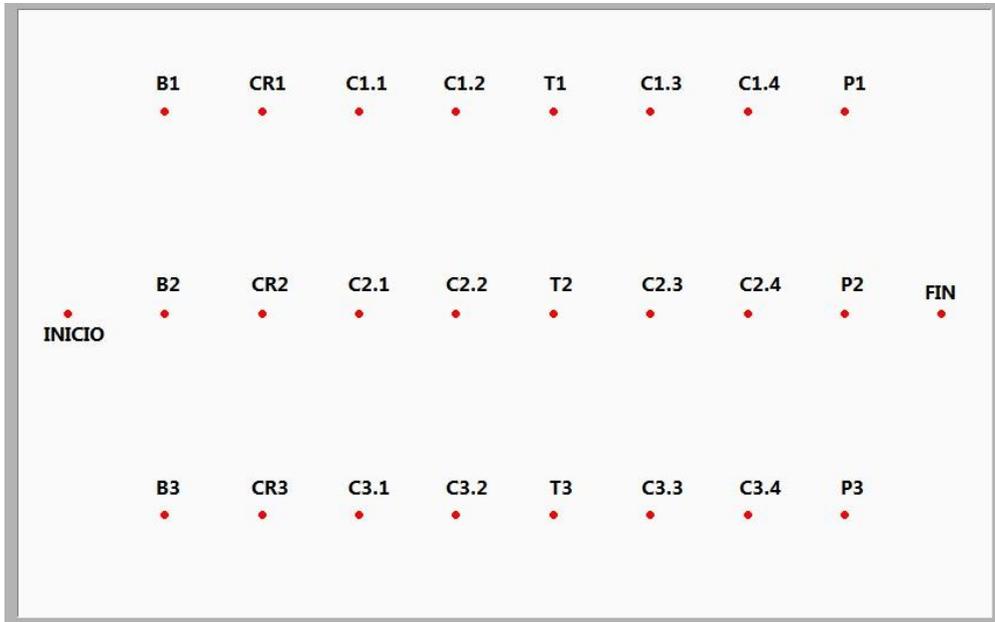


Ilustración 5.20 Representación Puntos

La segunda grafica sirve para dibujar las aristas que unen los puntos. La programación de esta parte es un poco más compleja que la de los puntos porque hay que pintar solo las aristas que unen puntos por donde pueden trasladarse piezas. Se han utilizado como datos de entrada una gráfica en blanco y las matrices de decisión y coordenadas. Para que la arista se dibuje en la gráfica tiene que cumplir la condición de que tenga un valor mayor que 0 y menor que  $1E10$  en la matriz de decisión, por lo tanto solo unirá las aristas que tienen un valor de tiempo en la matriz, las variables que valen infinito quedan excluidas. El programa va recorriendo la matriz posición por posición, aquellas posiciones donde se cumple la condición son los puntos de la matriz de coordenadas que tiene que unir. Si por ejemplo ha detectado que en la posición (1,2) de la matriz de decisión se cumple la condición, coge las coordenadas 1 y 2 de la matriz de coordenadas y las une.

### Algoritmo de Bellman-Ford

## 5. Supervisión de Fallos

El algoritmo de Bellman-Ford es el que calcula cual es el camino que tiene que hacer cada pieza para que el tiempo de fabricación sea el mínimo.

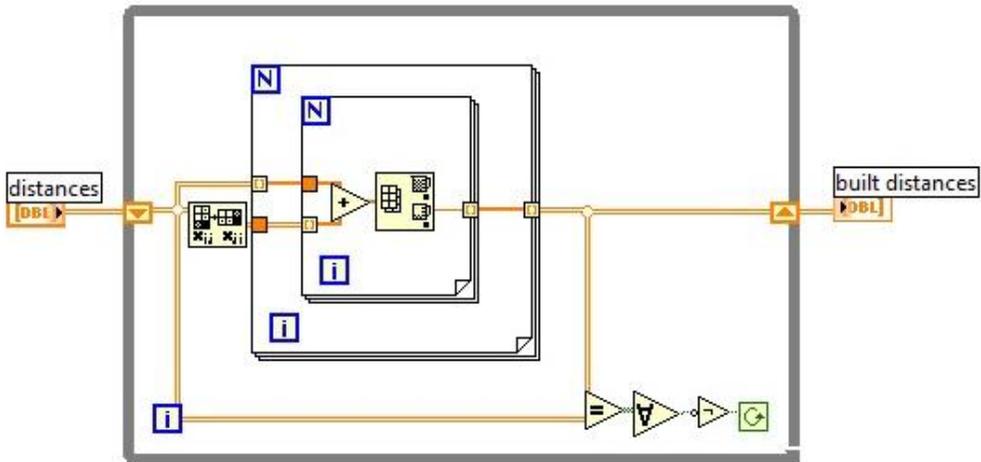


Ilustración 5.21. Programa

El programa parte de la matriz de decisiones y construye la matriz de distancias, llamada en el programa “built distances”. Para ello va cogiendo cada fila y cada columna de la matriz de distancias, y las suma por parejas, el primer elemento de la fila, con el primer elemento de la columna, el segundo con el segundo, así con todos. De todos los valores que ha cogido, se queda con el más pequeño, que es la distancia mínima entre la variable que representa a la fila y la variable de la columna. Luego escribe el dato hallado en el lugar de la matriz de distancias donde se cruzan la fila y la columna.

El programa repite iterativamente este proceso con todas las filas y columnas hasta que la matriz de distancias queda completada.

Una vez que la matriz de distancias esta completada, si se quiere saber cuál es la distancia mínima entre dos puntos, solo hay que mirar en ella donde se cruzan la fila del punto de origen y la columna del punto de destino y la matriz lo dirá.

**Cálculo de la distancia total y de los puntos por donde pasa el camino.**

El cuarto subprograma tiene dos misiones, calcular cuánto tiempo se tarda en fabricar la producción que aún queda y mostrar porqué puntos de la gráfica pasa el camino de manera matricial.

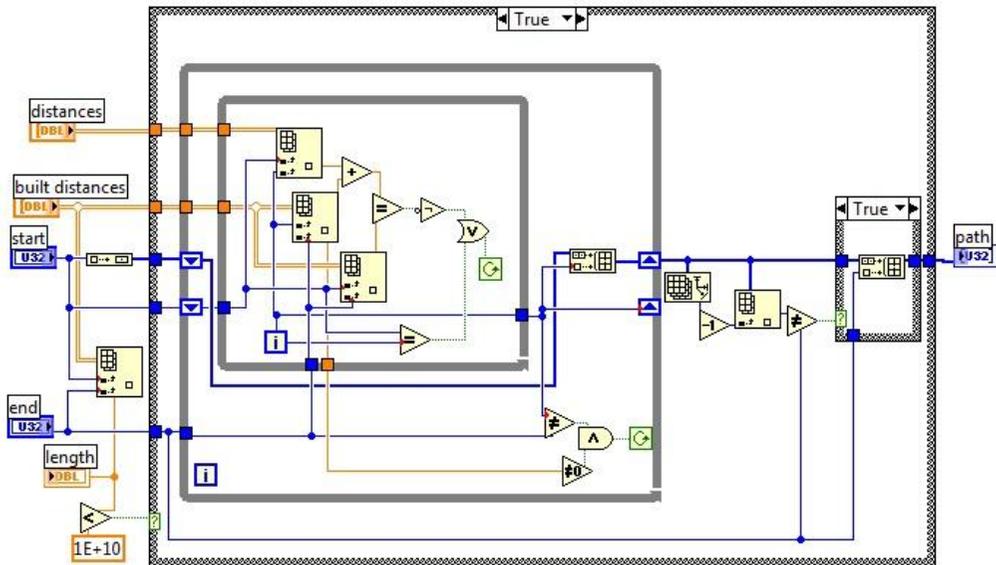


Ilustración 5.21. Programa

Las variables de entrada que usa son:

- La matriz de decisión.
- La matriz que distancias, calculada a partir del algoritmo.
- Un valor de inicio que en este caso es 0.
- Y un valor de final que representa a la última variable que tiene la matriz de decisión.

El tiempo que se tarda en fabricar las piezas es fácil de hallar. El valor ya está calculado en la matriz de distancias, así que ahora sólo hay que extraerlo y colocar un indicador que lo muestre. La casilla de la matriz de distancias donde está dicho valor es donde se cruzan la fila de la variable de inicio, que es el origen, (fila 0) y la columna de final, que es el destino (columna 26).

La búsqueda de los puntos por donde va el camino es algo más complicado. Ahora hay que averiguar cuál han sido los tiempos de la matriz

## 5. Supervisión de Fallos

---

de decisión que ha sumado el algoritmo para formar la matriz de distancias entre los puntos de inicio y final y luego crear una matriz donde ponga cuales son, para poder dibujar la gráfica del camino óptimo.

El planteamiento del problema ha sido el siguiente:

Cuando la matriz de decisión crea la matriz de distancias lo hace calculando las distancias mediante iteraciones. En la primera iteración calcula las distancias desde el vértice de inicio a los vértices donde llegan las flechas que salen desde él, compara los resultados y se queda con el valor más pequeño. En la siguiente iteración coge el vértice donde llega la flecha del valor mínimo hallado en la iteración anterior como vértice de salida y vuelve a hacer lo mismo, busca los valores de las flechas que salen de él y se queda con el más pequeño. Así sucesivamente hasta que llega al vértice del que no salen flechas, que es el vértice final.

El planteamiento de este programa es hacer el camino inverso, e ir desde el final buscando cuales han sido los vértices que ha utilizado en algoritmo para hallar el camino.

El programa compara los resultados hallados en la matriz de distancias con los de la matriz de decisiones desde el punto final hacia atrás. La situación del último punto en ambas matrices se conoce, es donde se cruza la fila de inicio con la columna de final. Ahora se van a buscar los dos valores que se han utilizado para hallar ese resultado. Estos valores pueden ser la suma del valor desde el primer vértice al segundo, más el valor desde el segundo vértice al final. El valor de los dos elementos es fácil de hallar, porque la suma tiene que ser igual al resultado final. El valor desde el vértice origen al segundo vértice que forma el camino tiene que estar en la fila de inicio de la matriz de decisión. El valor desde el segundo vértice hasta el final tiene que ser un valor dentro los que hay en la columna final de la matriz de distancias porque a esta columna llegan las distancias desde todos los puntos hasta el final. El lugar que ocupe el valor de la suma de elementos de la fila de inicio de la matriz de decisión más la columna final de la matriz que distancias, que coincide con el valor desde el inicio al final de la matriz de distancias será el segundo vértice que forma el camino. Para explicar esto vamos a utilizar el ejemplo del grafo sencillo ya mostrado.

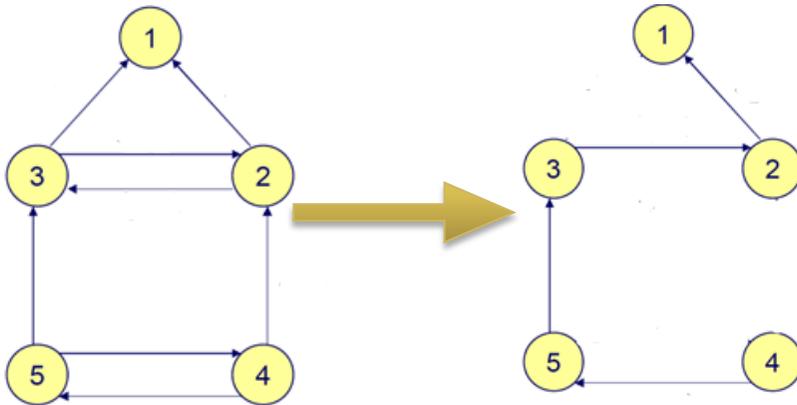


Ilustración 5.22. A partir del valor de los pesos del grafo se va a hallar el camino

destino /origen	1	2	3	4	5
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	1	0	2	$\infty$	$\infty$
3	4	1	0	$\infty$	$\infty$
4	$\infty$	8	$\infty$	0	4
5	$\infty$	$\infty$	2	2	0

**Matriz de Decisión**

destino /origen	1	2	3	4	5
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	1	0	2	$\infty$	$\infty$
3	2	1	0	$\infty$	$\infty$
4	8	8	6	0	4
5	4	3	2	2	0

**Matriz de Distancias**

El planteamiento es el siguiente, el valor del camino desde el vértice 4(inicio) al 1 (final) en la matriz de distancias es 8, es lo que se tarda en recorrer el camino.

Ahora el recorrido se puede dividir en dos partes y la suma de los tiempos de los dos tiene que ser 8. La primera parte del recorrido va desde el inicio al segundo vértice y la segunda parte va desde el segundo vértice al final.

Luego el valor del tiempo desde el origen al segundo vértice está en la fila 4 de la matriz de decisión. Y el valor desde el segundo vértice al final está en algún lugar de la columna 1 de la matriz de distancia.

## 5. Supervisión de Fallos

---

FILA 4	$\infty$	8	$\infty$	0	4
COLUMNA 1	0	1	2	8	4
Resultado	$\infty$	$\infty$	$\infty$	8	8

El valor que está situado en la misma posición el número de fila no vale porque es el valor desde el inicio al final. El valor que sí que es válido es el del quinto lugar, por lo tanto el segundo punto del camino es el vértice 5. Para seguir averiguando puntos se toma ahora como punto origen el vértice 5 y como final el vértice 1 y se vuelve a realizar los mismos pasos.

### Matriz de tiempos mínimos

La matriz de tiempos mínimos es una matriz que muestra solo los tiempos que forman parte del camino que ha dado el algoritmo. La razón de crear esta matriz es que es necesaria para el programa de prioridades porque muestra cual es el camino que ha seguido el algoritmo y los tiempos que ha utilizado. Esta es la matriz que se suma a la matriz de decisión que da el programa de variables para calcular la matriz de decisión modificada que luego usa el algoritmo en los casos en que la línea no es la línea de preferencia. Si la línea de producción que esta como preferente, la que pasa primera por cualquier línea, es la línea 3 y tiene que usar las máquinas de la línea 2 porque se ha averiado alguna de la línea 3, la línea 2 se tendría que espera a que la línea 3 termine para seguir con la producción suya. Luego para hacer los cálculos de esta línea hay que sumar los tiempos de la línea 3 más los tiempos de la línea 2. Los tiempos de la línea 3 son los tiempos que muestra la matriz de tiempos mínimos de esta línea, los tiempos por donde pasa el camino dado por el algoritmo.

La matriz tiene la misma dimensión que la matriz de decisiones. Toda la matriz está llena de ceros, excepto los lugares por donde pasa el camino dado por el algoritmo, que están los tiempos que tiene la matriz de decisión, los tiempos que se tarda en pasar de una maquina a la siguiente.

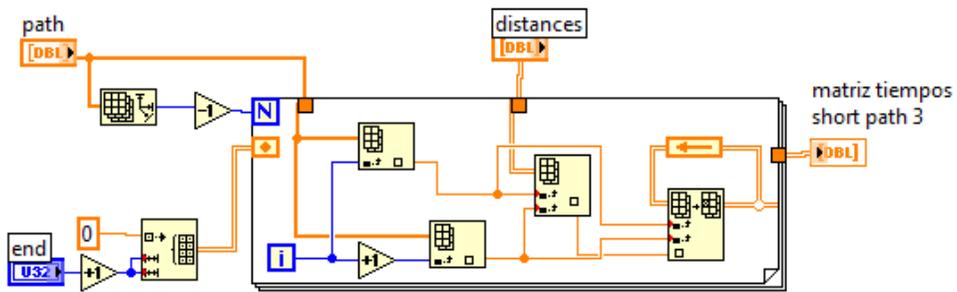


Ilustración 5.23. Programa

El programa primero crea la matriz de ceros, con la misma dimensión que la matriz de decisión. Utiliza la matriz de puntos del camino dado para buscar en la matriz de decisión el lugar de la variable de tiempo que luego se va a colocar en el mismo lugar pero en la matriz de tiempos mínimos. El lugar de la variable del primer tiempo de la matriz de decisión está situado en el sitio donde se cruza la fila número que indica el primer elemento y la columna número que marca el segundo elemento de la matriz de puntos. Luego, si los dos primeros números de la matriz de puntos son el 2 y el 3, la variable de tiempo que va desde el primer punto al segundo es el tiempo situado en el cruce de la fila 2 con la columna 3 de la matriz de decisión. Si el segundo y tercer punto de la matriz de puntos son el 3 y el 4, el tiempo para pasar de la segunda variable a la tercera será el que está colocado en el cruce de la fila 3 con la columna 4 de la matriz de decisión. Después de hacer todas las iteraciones el resultado de la matriz de tiempos mínimos es algo así:



matriz de puntos, y se unen con una función de cluster, y la gráfica queda dibujada.

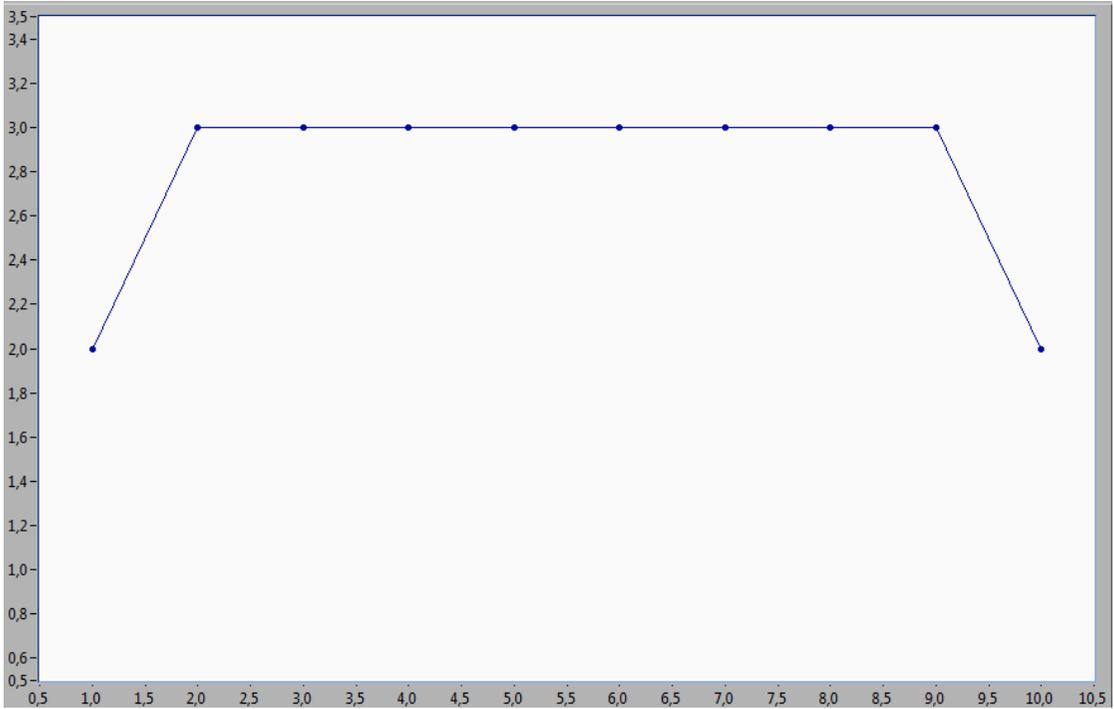


Ilustración 5.25 Representación Camino



# Ejemplos de Aplicación.

A continuación se van a exponer varios casos de ejemplos de aplicación sobre la planta industrial. Los ejemplos son por un lado de diseño y control de la planta y por otro de la optimización de caminos cuando ocurre un fallo.

## 6.1. Diseño y Control de la planta.

En los ejemplos de diseño y control se ha tratado de enseñar la versatilidad de la planta poniendo varios casos de opciones de producción.

### Caso 1. Producción normal.

En el primer caso, las tres líneas funcionan a la vez, cada una produciendo el tipo de pieza que le corresponde.

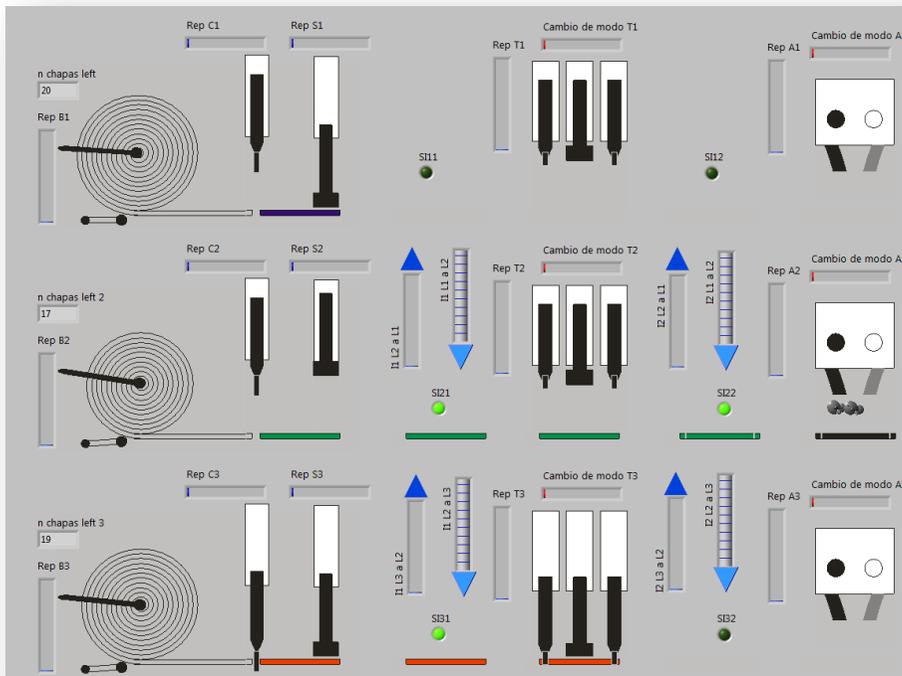


Ilustración 6.1. Producción Normal

### Caso 2. Diversidad de producción.

En este caso se quiere mostrar la diversidad de opciones de producción que permite la planta. Gracias a las cintas instaladas como intercambiadores de piezas, se puede enviar piezas de unas líneas a otras, por lo tanto, se pueden usar el taladro y pintura de cualquier línea para fabricar todos los tipos de piezas. Cuando se produce el envío, las piezas de las cintas de intercambio tienen prioridad frente a las de línea, lo que permite que las piezas de intercambio entren antes a las de línea.

El diseño de la programación de graficet mediante subprocesos, cada máquina tiene un graficet asociado, hace que los intercambios de piezas entre líneas sean posibles porque el autómatas controla las máquinas y no las líneas de producción de la planta.

En la imagen se puede observar que las tres líneas además de fabricar las piezas que le corresponden, están fabricando piezas de calidad 2 (piezas verdes) y calidad 3 (piezas rojas).

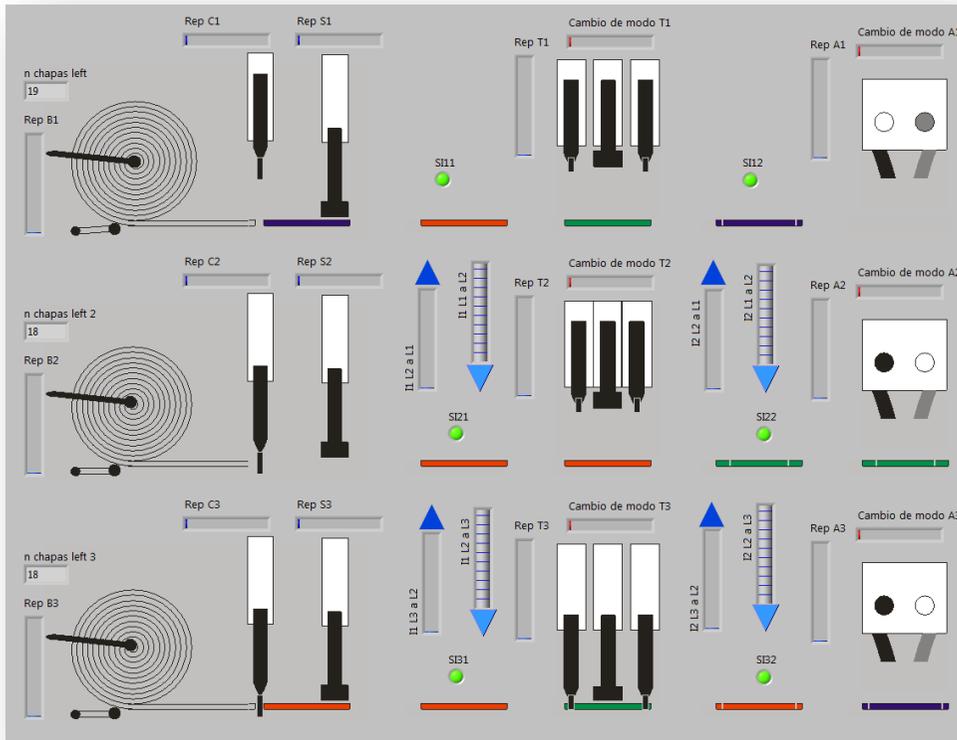


Ilustración 6.2. Diversidad de Posibilidades

### Caso 3. Agilidad de Producción.

Los intercambiadores se pueden utilizar para agilizar la producción. Se puede dar el caso que una de las líneas no tenga que producir piezas de su tipo y las máquinas estén parada. Esas máquinas se pueden utilizar para fabricar piezas de otras líneas y así reducir el tiempo de producción total y aumentar la productividad de la planta y las máquinas.

En el ejemplo se utilizan la línea 3 y la línea 2 para producir piezas de tipo 2 a la vez para agilizar la producción porque cada una tiene que ir pintada de un color diferente. Mientras tanto la línea 1 sigue con su producción.

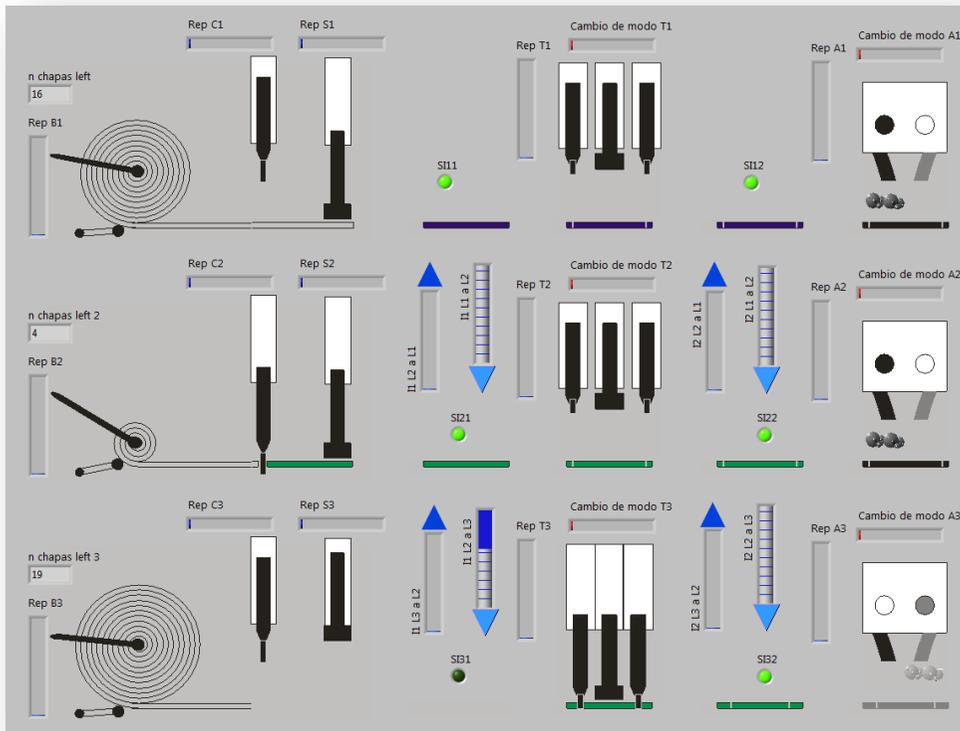


Ilustración 6.3. Agilidad de Producción

### Caso 4. Avería de máquinas.

Si se avería alguna máquina, las piezas de esa línea que aún quedan por fabricar se pueden desviar al resto de líneas utilizando los intercambiadores mientras se arregla la avería.

Esto permite que la producción de esa línea no esté parada durante el tiempo que dura la avería.

En este caso, se ha averiado el taladro de la línea 1 y el centro de pintura de la línea 2, las piezas que se estaban fabricando por la línea 1 se han desviado primero a la línea 2 para hacer los taladros y luego a la línea 3 para ser pintadas. Esta es una opción, pero también se podrían enviar directamente a la línea 3 o en vez de hacer el acabado en la línea 3 pasarlas otra vez a la línea 1. El encargado de decidir cuál es el camino que tarda menos tiempo es el optimizador de fallos.

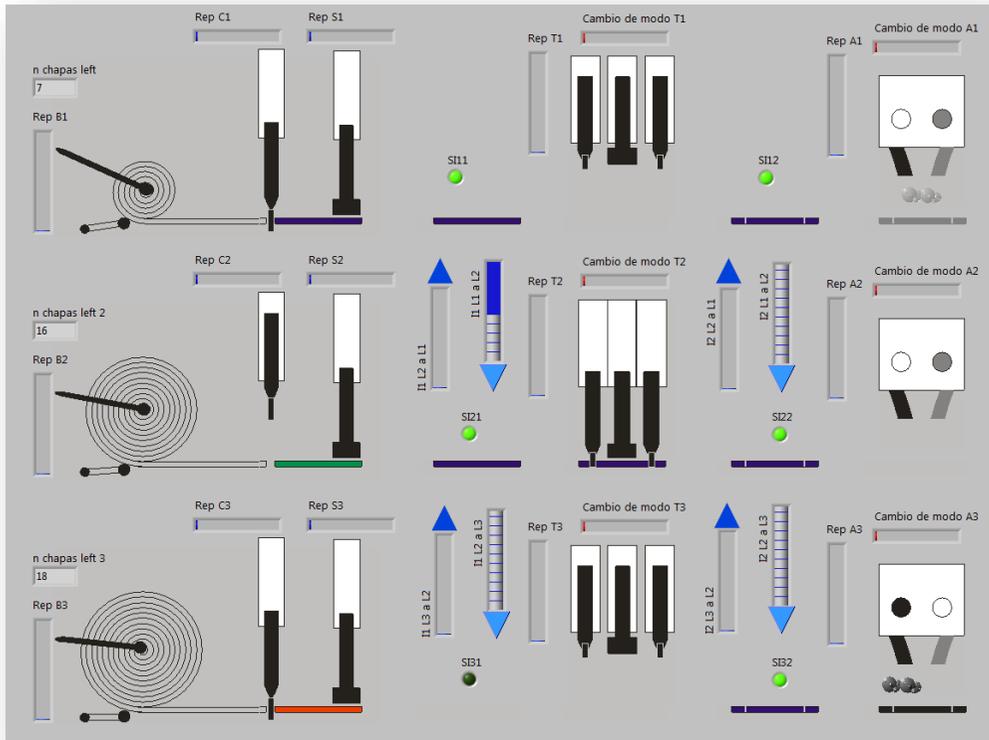


Ilustración 6.4. Avería del Taladro 1 y la Pintura 2

### 6.2. Supervisión de fallos.

A continuación se van a exponer varios casos en los que actúa el nivel de supervisión de fallos. Este nivel entra en funcionamiento cuando aparecen averías en alguna de las máquinas, el sistema optimiza los caminos de producción de piezas para que el tiempo de fabricación sea el mínimo, dependiendo de la línea que tenga prioridad de paso.

Para simplificar los cálculos y que todo sea más fácil de entender, se va a suponer que todas las máquinas y las cintas de línea tardan 1 segundo en hacer su trabajo y los intercambiadores tardan 5 segundos en enviar piezas desde unas líneas a otras. Los tiempos de arreglo de cada máquina están puestos al azar. El número de piezas a fabricar por línea también es 1. Todas las variables están reflejadas en el panel de tiempos.

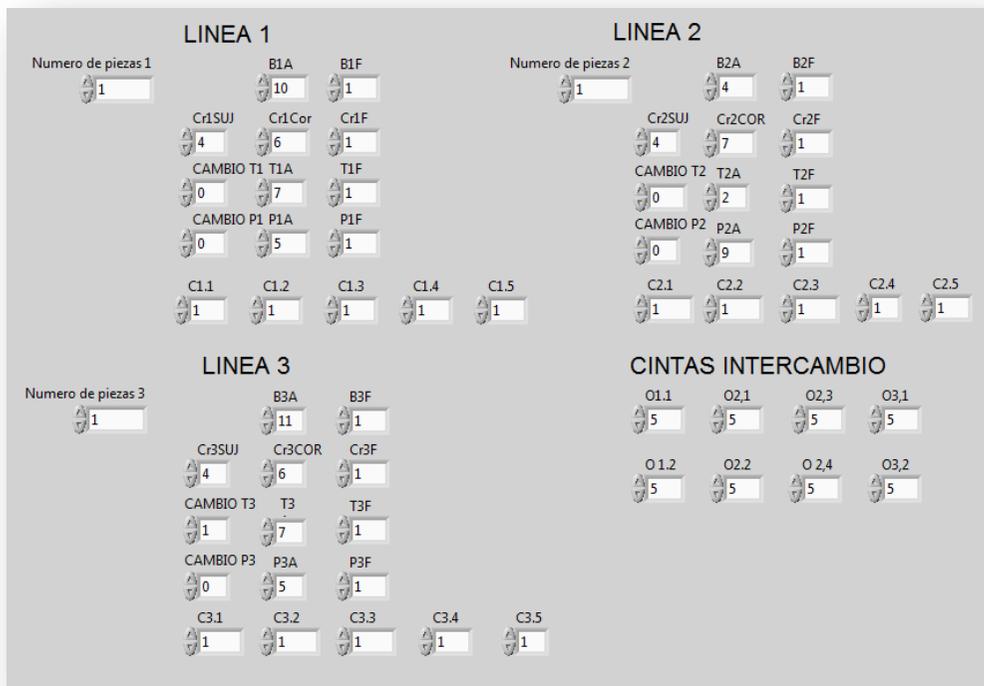
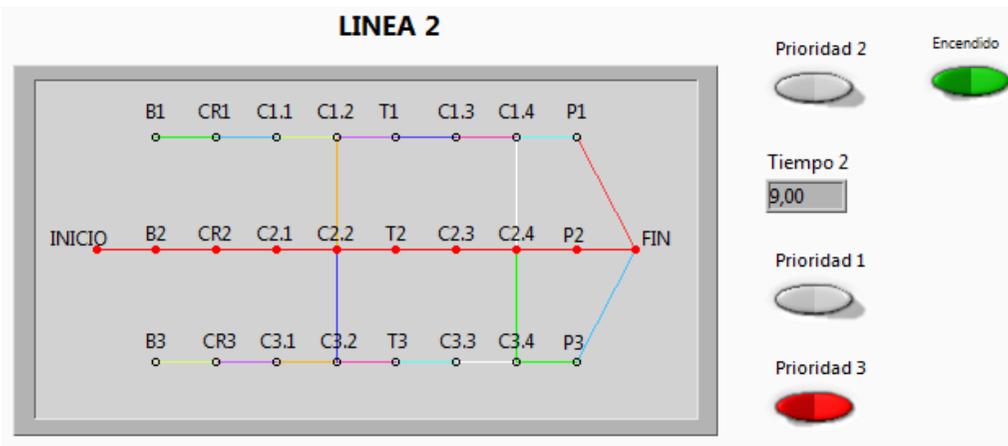
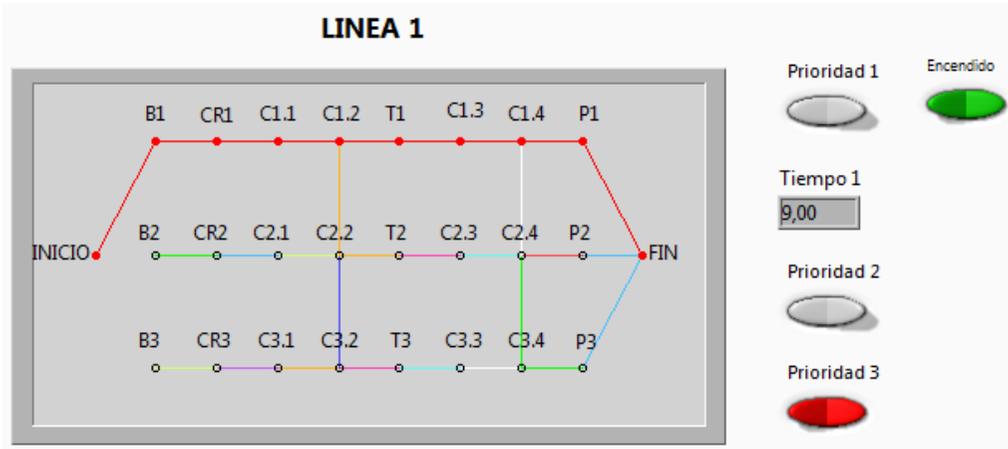


Ilustración 6.5. Panel de Tiempos

Caso 1. La planta funciona sin ninguna avería

En este caso la planta funciona correctamente y no hay ninguna avería, se están fabricando piezas por las tres líneas. La suma del tiempo del proceso de fabricación son 9 segundos, 4 segundos por cada una de las maquinas por donde pasa y 5 de las 5 cintas de línea.



## 6. Ejemplos de Aplicación

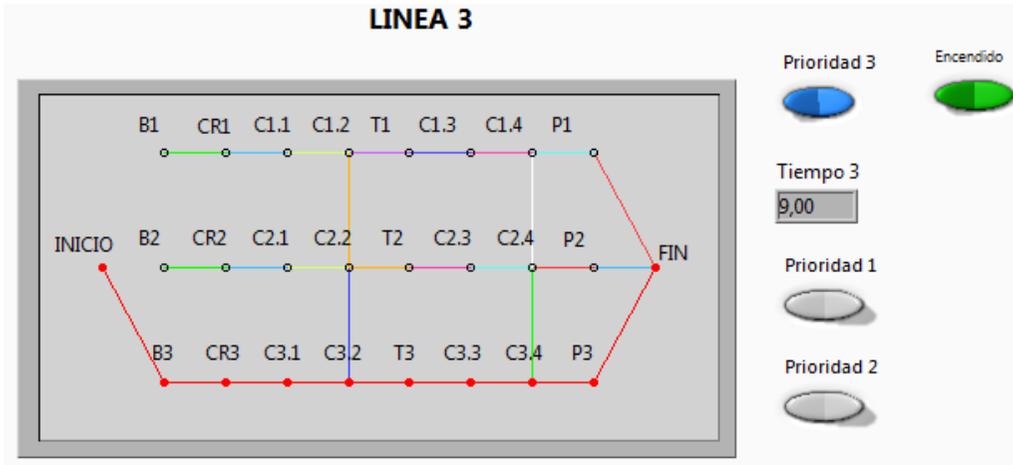
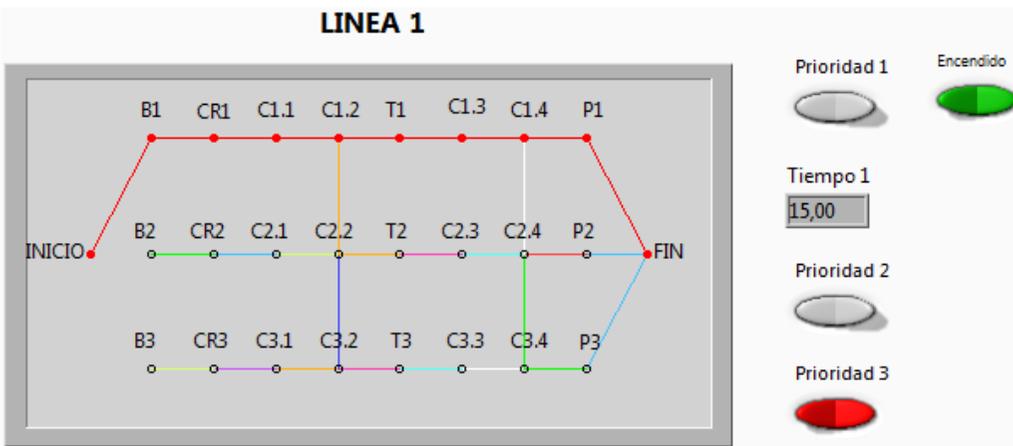


Ilustración 6.6. Funcionamiento sin fallos

### Caso 2. Fallo de una máquina y sistema de prioridades.

Se va a suponer que se avería el corte de la línea 1. El tiempo de arreglo es de 6 segundos. Si esta activada la prioridad 3, donde siempre tiene preferencia la línea 3, luego la 2 y por último la 1, el programa indica que la opción de menor tiempo de fabricación es esperar a que se arregle la máquina y luego continuar la producción de la línea 1.



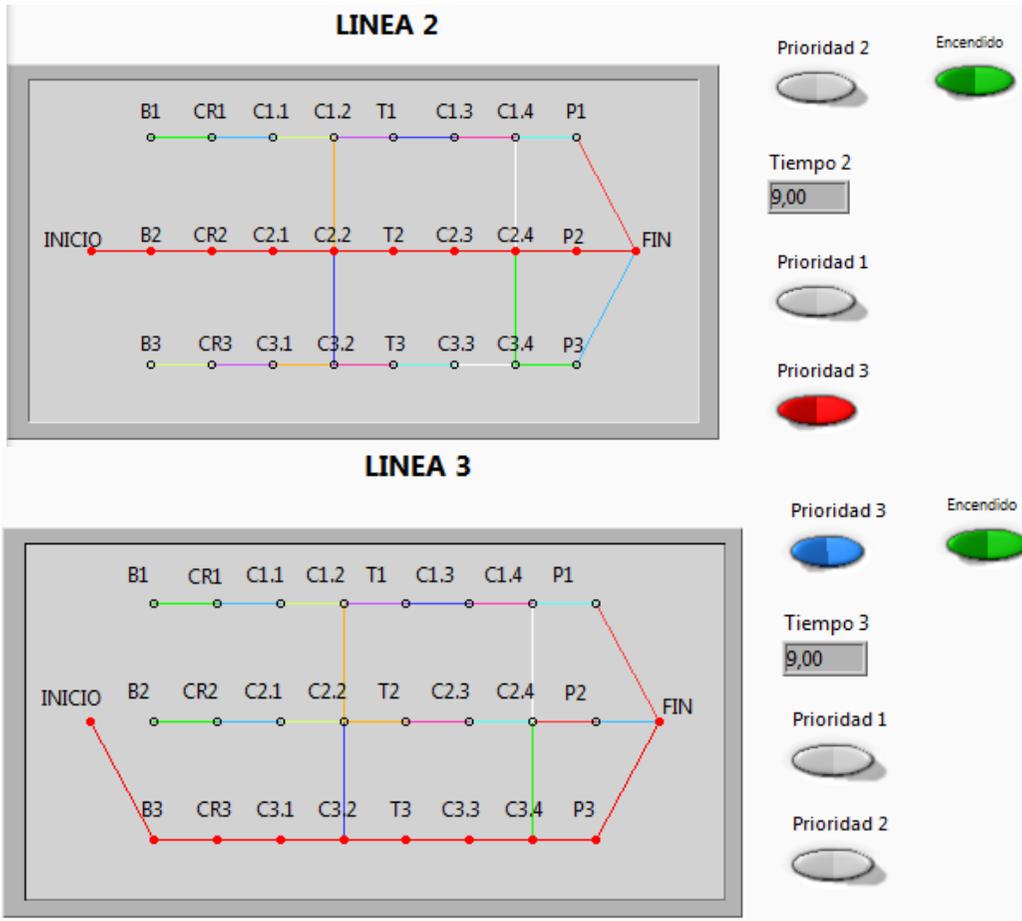


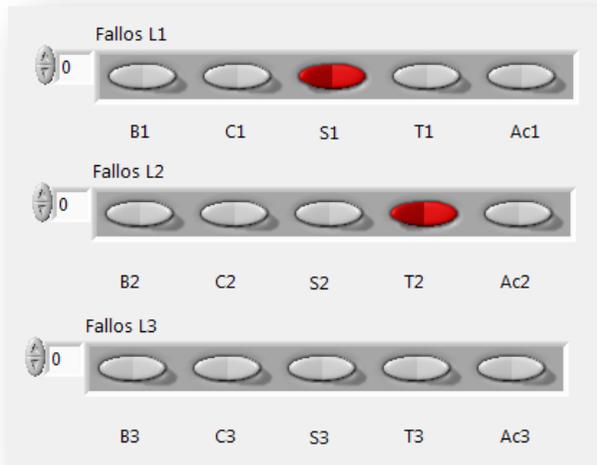
Ilustración 6.7. Fallo Línea 1, Prioridad 3

En cambio, si tiene la prioridad la línea 1, los caminos de las piezas cambian. Las piezas de calidad 1 pasan a la línea 2, donde el tiempo de fabricación es de 9 segundos y las piezas de calidad 2, comienzan en su línea cuando ya han pasado las de la línea 1 y después del corte se desvían a la línea 1 para terminar el proceso.



### Caso dos. Se averían dos máquinas de líneas diferentes.

Se va a suponer que se avería la sujeción de la línea 1 y se tarda en reparar 6 segundos y el taladro de la línea 2 que tarda en arreglarse 2 segundos.



**Ilustración 6.9. Panel de fallos**

Si la prioridad la tiene la línea 1, las piezas de la línea 1 pasan a la línea 3, las de la 2 se esperan a que se arregle la avería de su línea y las de la 3 esperan a que se fabrique la 1 y luego pasan:

## 6. Ejemplos de Aplicación

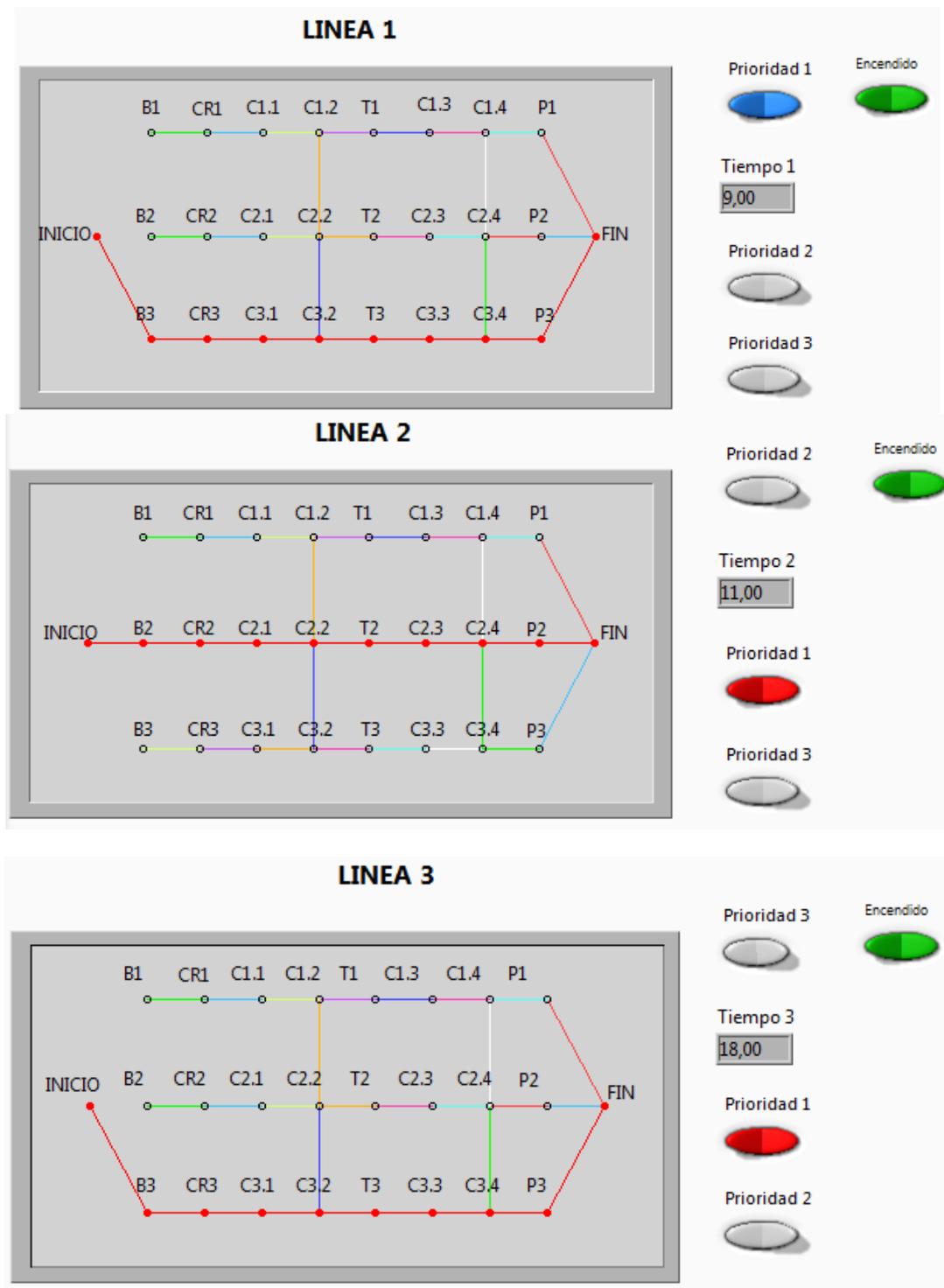
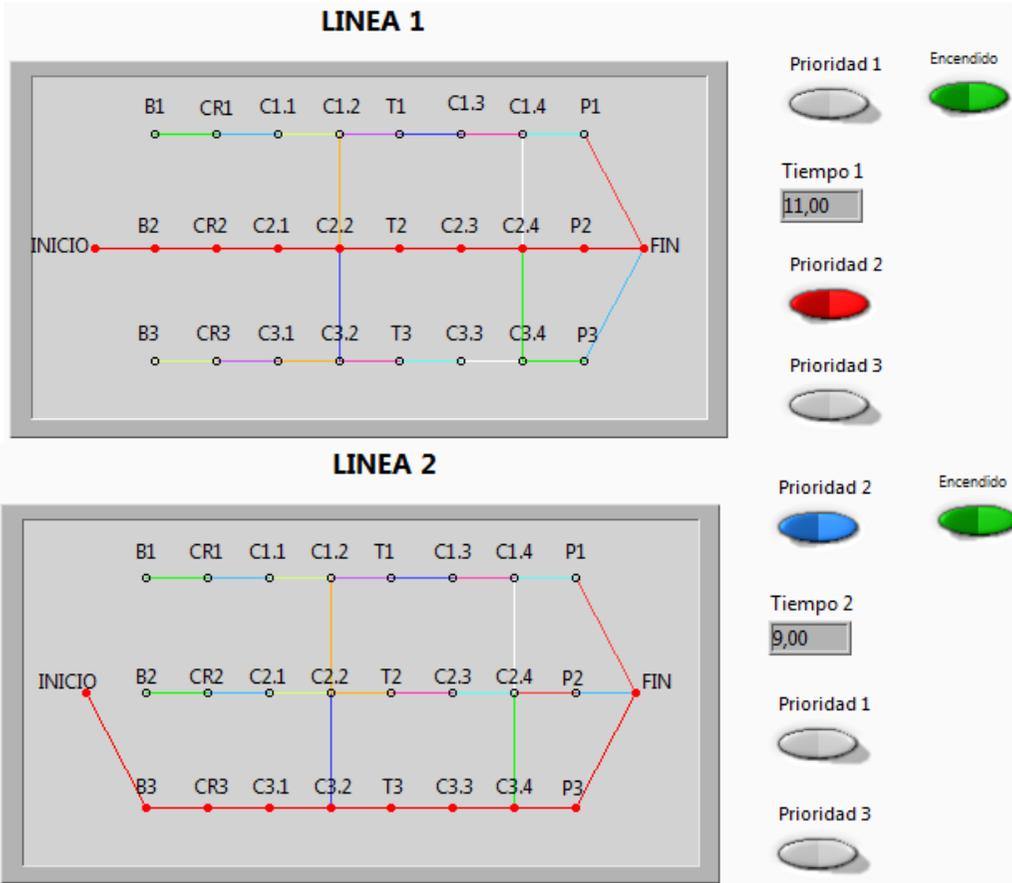
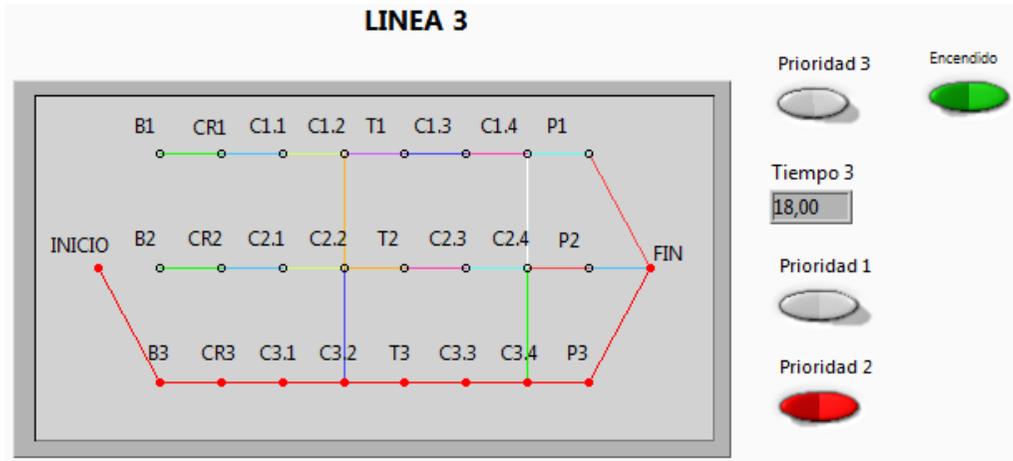


Ilustración 6.10. Fallos Línea 1 y Línea 2, Prioridad 1

## 6. Ejemplos de Aplicación

En cambio si la prioridad la tiene la línea 2, las piezas de la línea 1 deben ir por la línea 2 y esperarse a que se arregle la avería, las de la línea 2 pasan a la 3, porque esta no tiene ninguna avería, y las de la línea 3 se tienen que esperar a que fabriquen las de calidad 2 para fabricar las suyas.





**Ilustración 6.11. Fallos Línea 1 y Línea 2, Prioridad 2**

El programa también tiene la opción de trabajar cuando alguna de las líneas de fabricación no está funcionando, el resto de líneas la pueden utilizar como alternativa.

#### Caso 4. Alguna de las líneas no está activa.

Vamos a suponer que no hay pedido para los productos de calidad tipo 1, si se averían el taladro de la línea 2 y la pintura de la línea 3, cuando tienen prioridad la línea 3 esto es lo que pasaría:

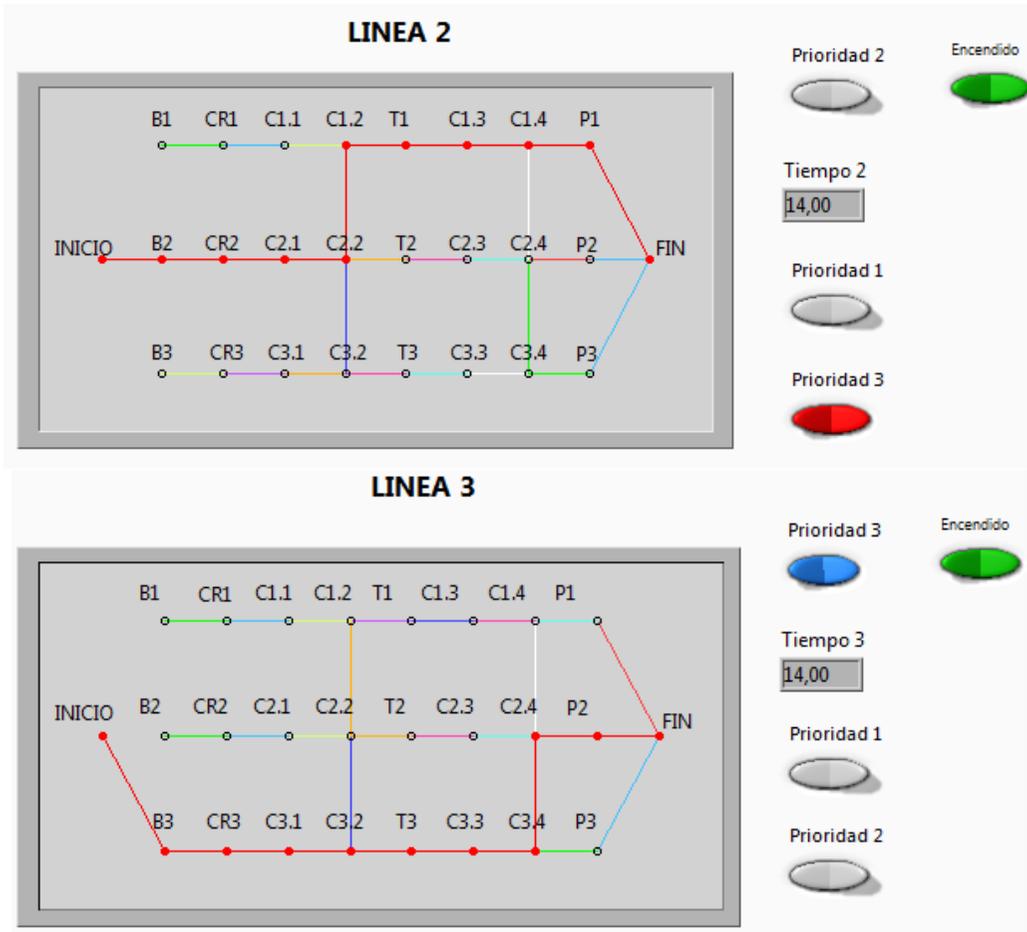


Ilustración 6.12. Línea 1 apagada

Las piezas de la línea 3 pasan a la dos en el punto intermedio 2, y las piezas de la línea 2 pasan a la línea 1 en el punto intermedio 1, y así ambos evitan la zona donde están las averías.



# *Conclusiones y Trabajos Futuros*

---

## **Conclusiones**

En la presente tesina se ha desarrollado la automatización parcial de una planta industrial. Hasta el momento, la planta está diseñada, es capaz de fabricar las piezas automáticamente, gracias a un autómatas y al sistema de control. Cuando aparecen averías de maquinas la parte de supervisión de fallos actúa indicando cual es la opción óptima para que se pierda el menor tiempo de fabricación.

El proyecto comienza al conocer que tipo de piezas se querían fabricar y que características tenían. A partir de esta información se ha investigado cual era el proceso de producción que había que seguir y las máquinas que hacían falta para diseñar la planta, el número de líneas que hacían falta, la colocación de la maquinaria... La idea siempre ha sido que la planta pudiera ser lo más flexible posible para poder reducir tiempos de producción, eso se consiguió colocando cintas transportadoras que intercambiaban piezas entre líneas. Además estas líneas también servían para reducir el tiempo de espera cuando se avería alguna máquina.

La planta se ha creado utilizando una simulación donde se recrea la planta de producción con las líneas y maquinas tal como se ha pensado en la parte de diseño. La planta se comporta como si fuera una planta real. Sigue la secuencia de procesos sin cometer errores porque utiliza un modelo de la secuencia modelado con Redes de Petri Coloreadas, que hace que cada pieza vaya pasando por cada uno de los estados de fabricación solamente cuando se han completado los anteriores.

La planta fabrica las piezas automáticamente sin la ayuda de ninguna persona. El proceso de control lo realiza el autómatas. Este va indicando las acciones que se tienen que ir sucediendo para que la pieza se vaya fabricando. El autómatas utiliza la programación mediante Grafset. Cada

máquina tiene su graficet asociado que muestra la secuencia de acciones que tienen que seguir para que su parte de la fabricación de la pieza se haga correctamente.

La supervisión de fallos calcula y simula cual es el camino que tiene que seguir cada línea cuando aparece un fallo en alguna máquina. Gracias a este sistema los tiempos de fabricación son menores, haciendo ganar productividad a la planta.

### **Trabajos Futuros**

Como trabajos futuros se proponer continuar con la automatización de la planta e integrando todas sus partes completamente.

Dentro de la pirámide que representa a la automatización tipo CIM, aún quedan dos niveles por desarrollar. El primero, el nivel 3, nivel de planificación debe ocuparse de las siguientes tareas:

- Programación de la producción
- Gestión de compras
- Análisis de costes de fabricación
- Control de inventarios
- Gestión de recursos de fabricación
- Gestión de calidad
- Gestión de mantenimiento.

El nivel 4 corresponde a la gestión empresarial, y en él se deben realizar estas tareas

- Gestión comercial y marketing
- Planificación estratégica
- Planificación financiera y administrativa
- Gestión de recursos humanos
- Ingeniería de producto
- Ingeniería de proceso
- Gestión de tecnología

- Gestión de sistemas de información
- Investigación y desarrollo

También se propone que se desarrolle un poco más el nivel de supervisión de fallos, actualmente, los cálculos se realizan basándose en los tiempos de producción, pero también se pueden introducir otras variables como el coste por pieza para dar más versatilidad a la planta.

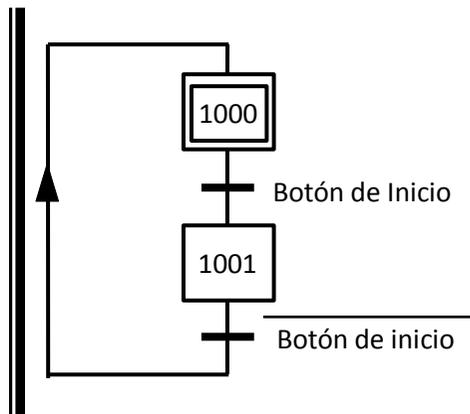


# ANEXO: Grafcet de Programación

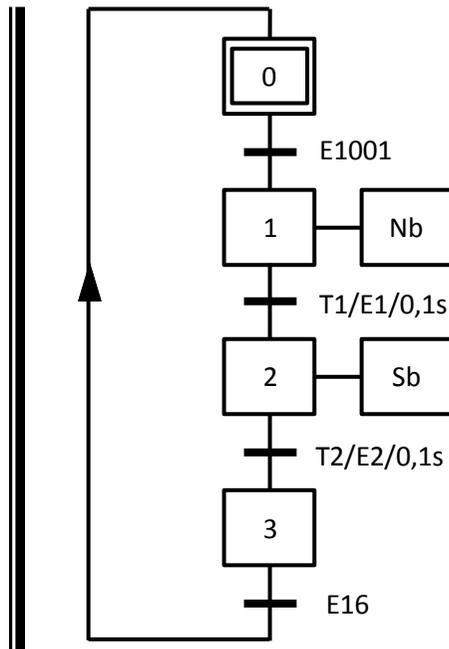
---

## Línea 1

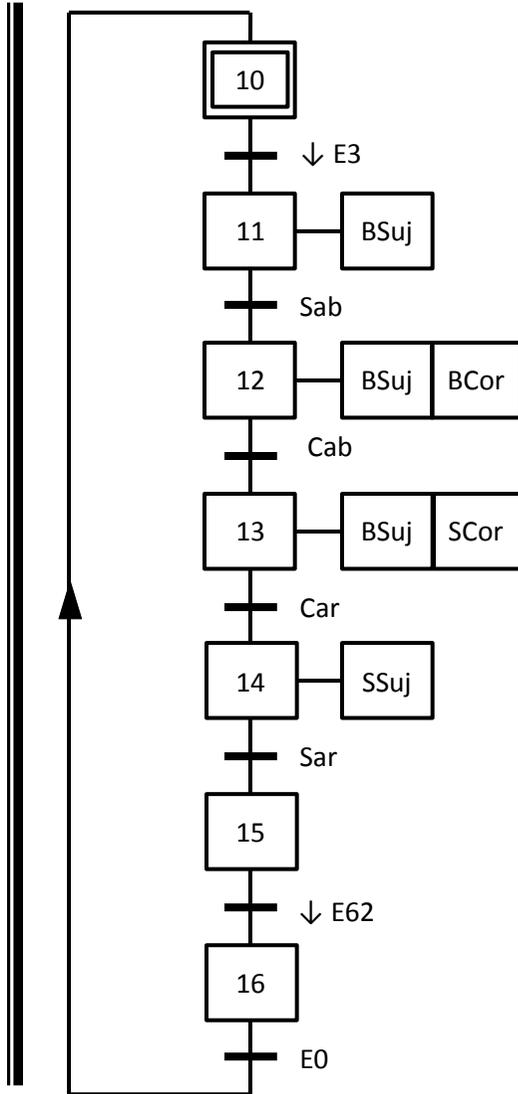
### 1. Puesta en Marcha



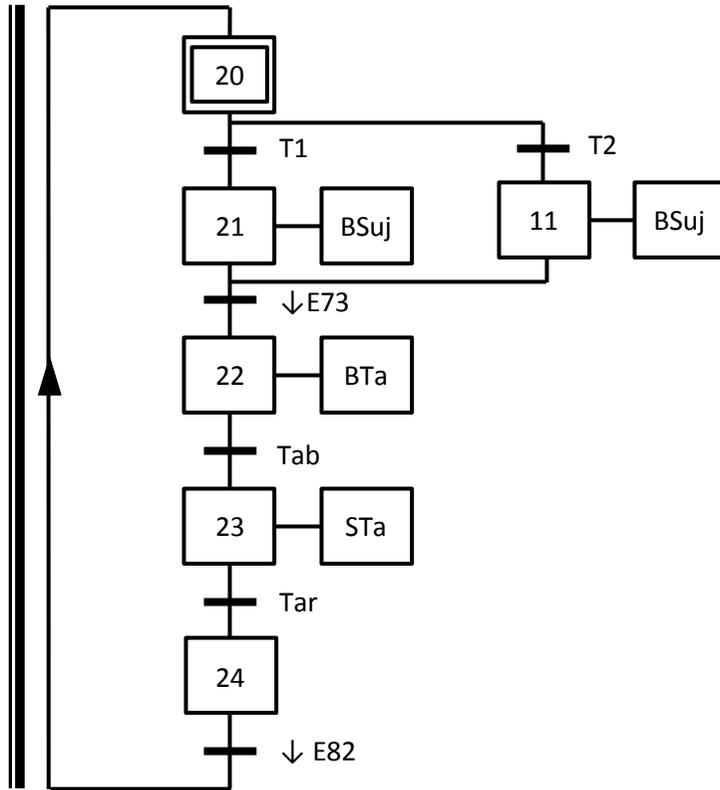
### 2. Bobina



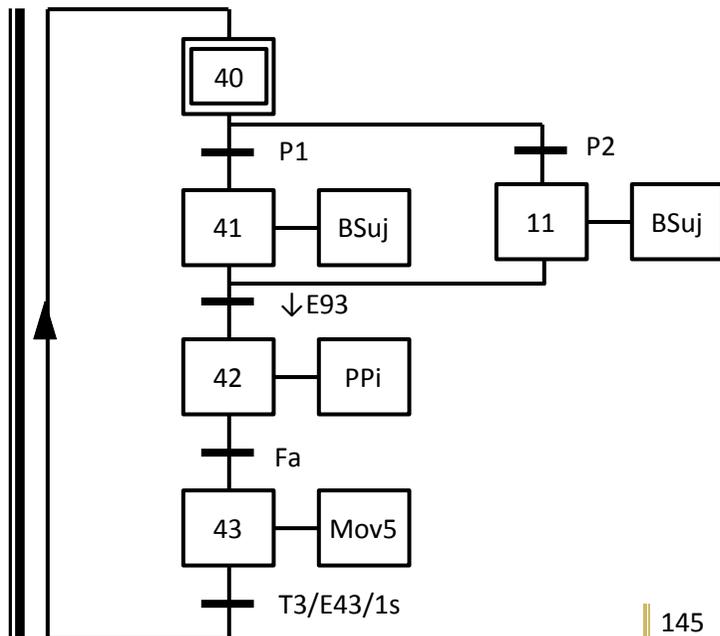
3. Corte.



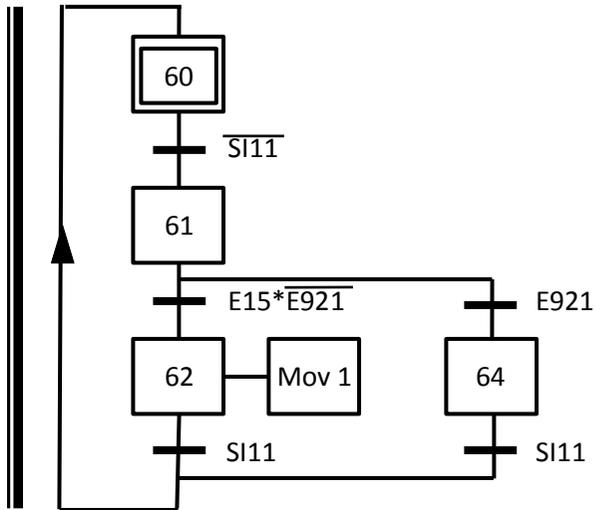
4. Taladro.



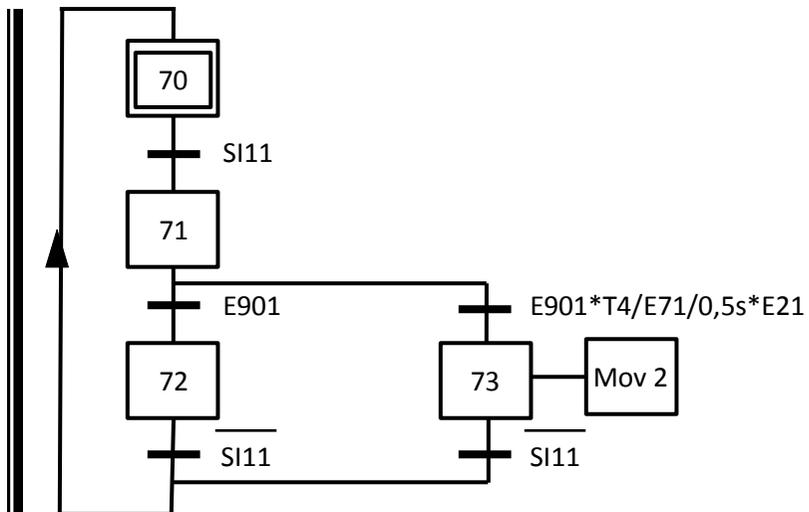
5. Pintura.



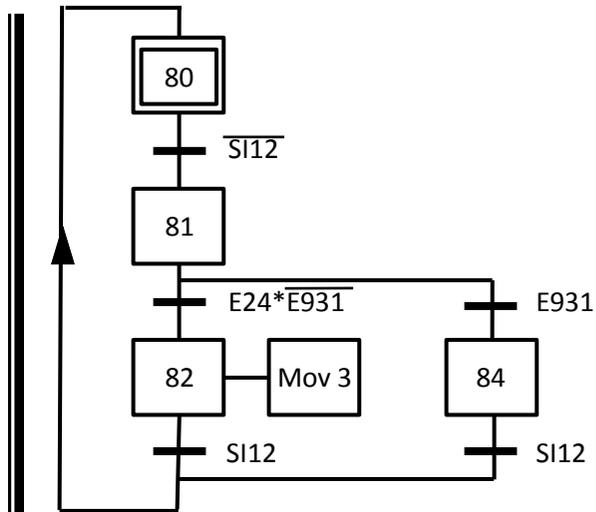
6. Intermedio 1, llegada de piezas.



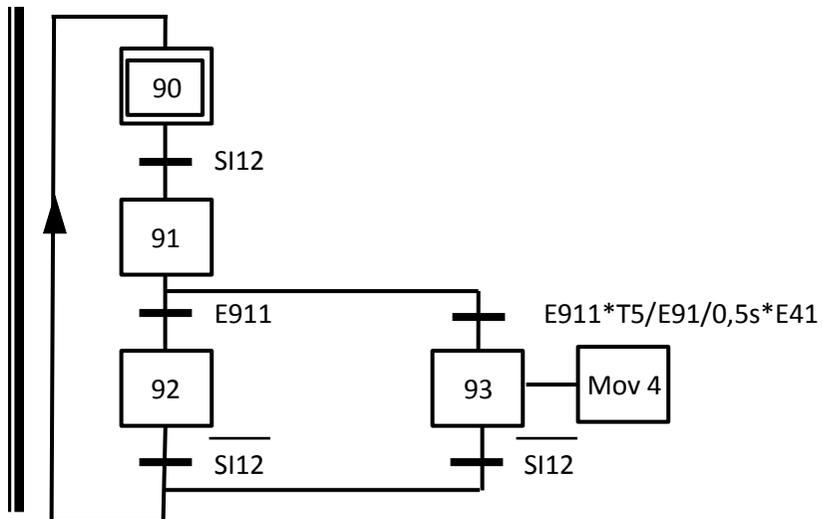
7. Intermedio 1, salida de piezas.



8. Intermedio 2, llegada de piezas.

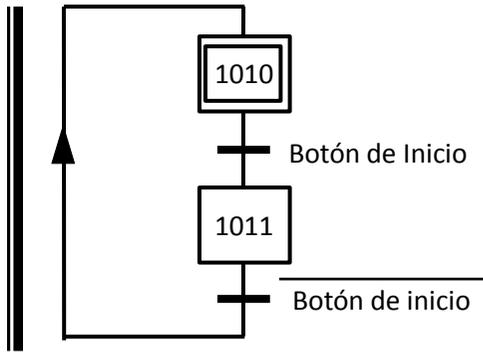


9. Intermedio 2, salida de piezas.

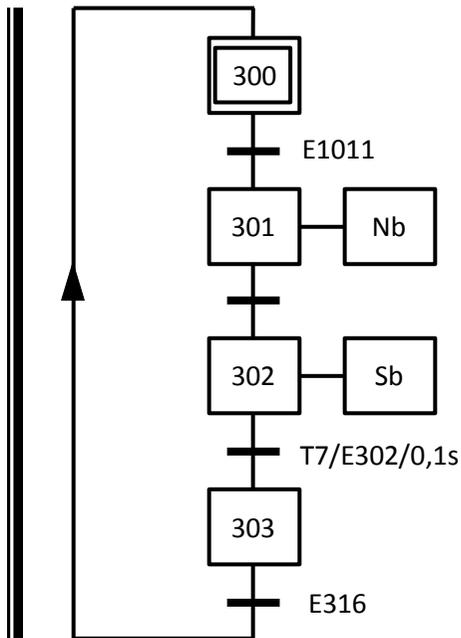


## Línea 2

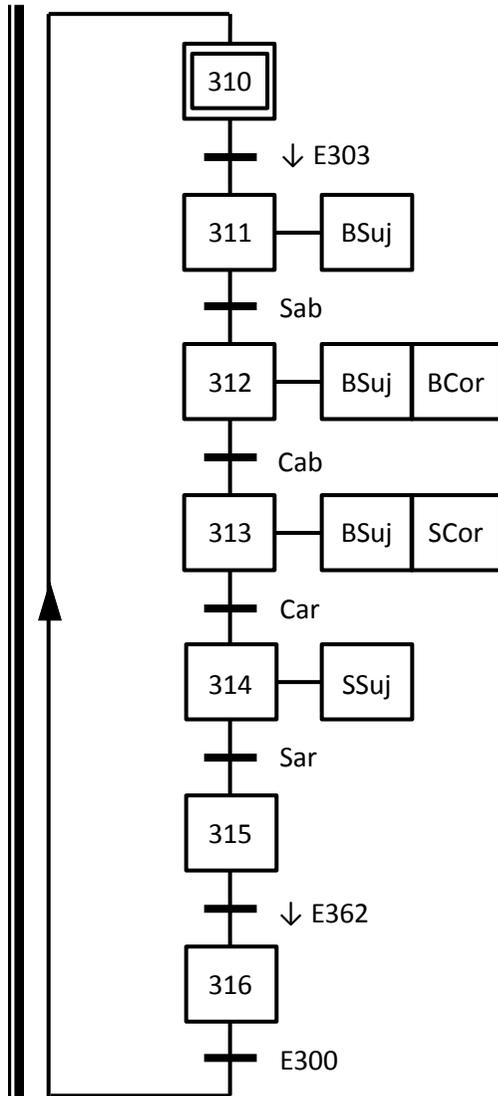
### 1. Puesta en Marcha



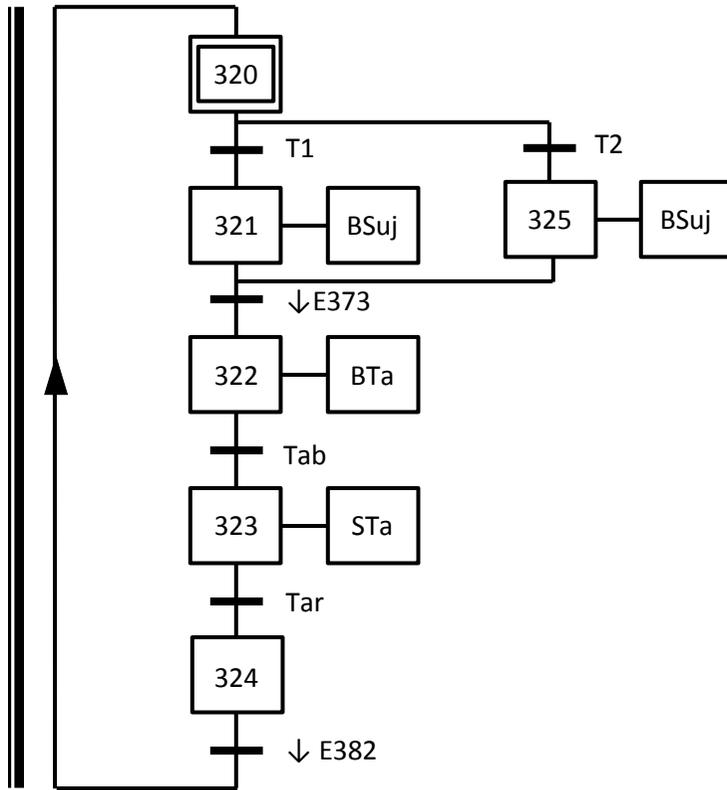
### 2. Bobina



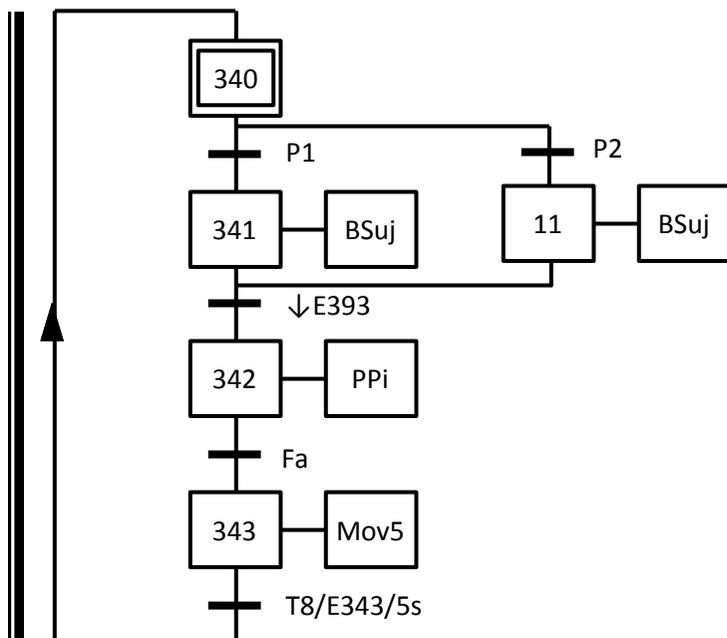
3. Corte



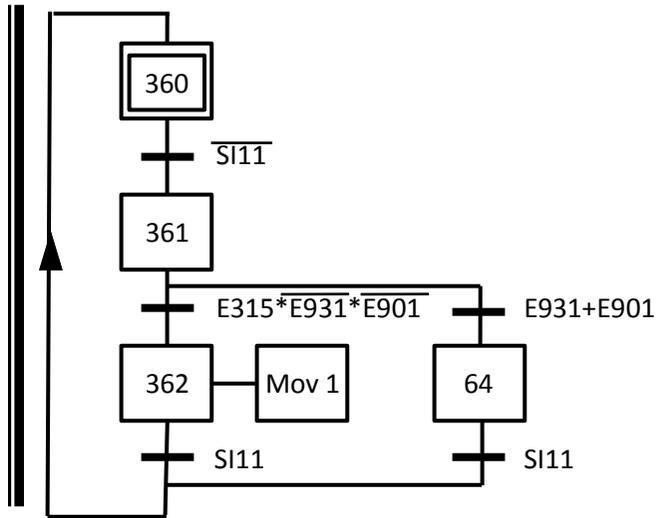
4. Taladro



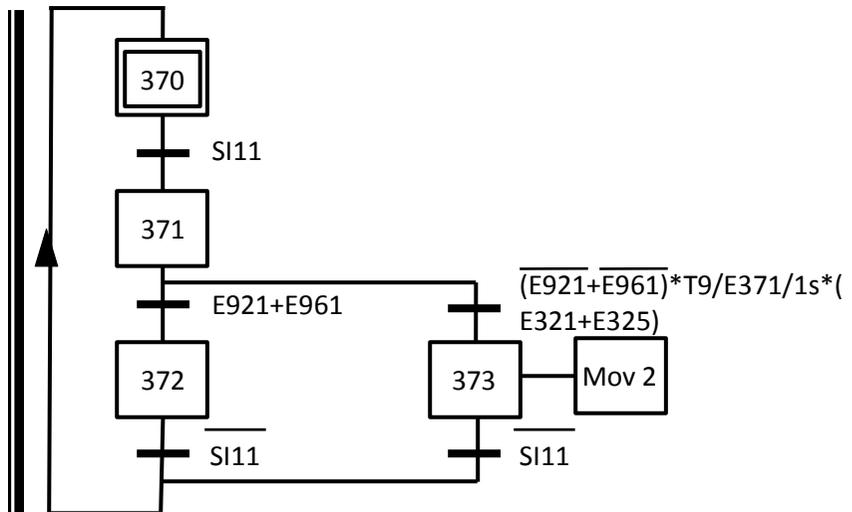
5. Pintura



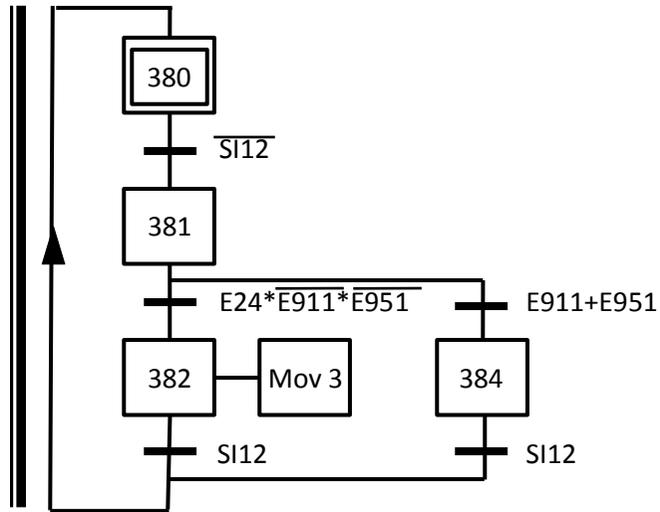
6. Intermedio 1, llegada de piezas



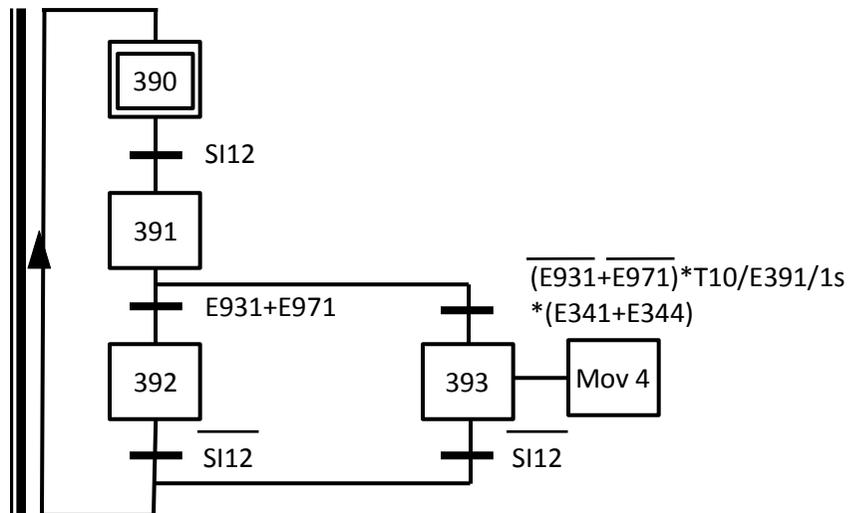
7. Intermedio 1, salida de piezas



8. Intermedio 2, llegada de piezas

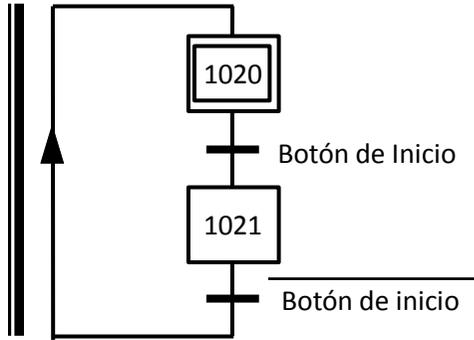


9. Intermedio 2, salida de piezas

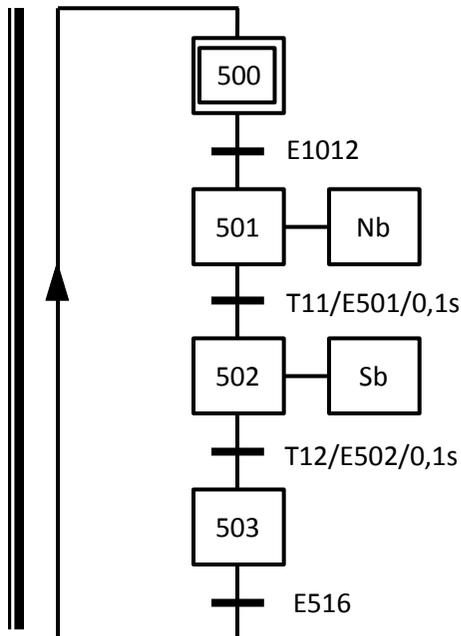


### Línea 3

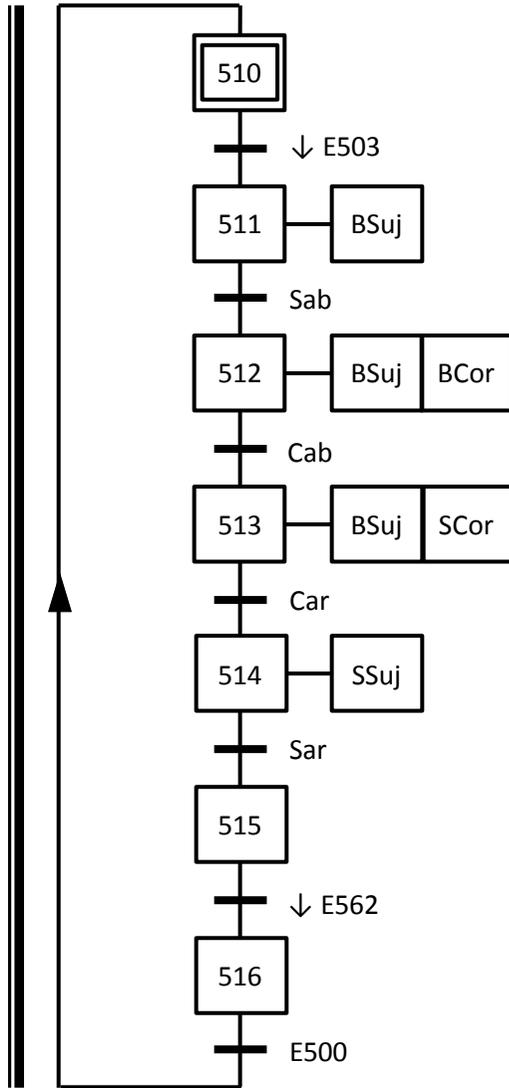
#### 1. Puesta en Marcha



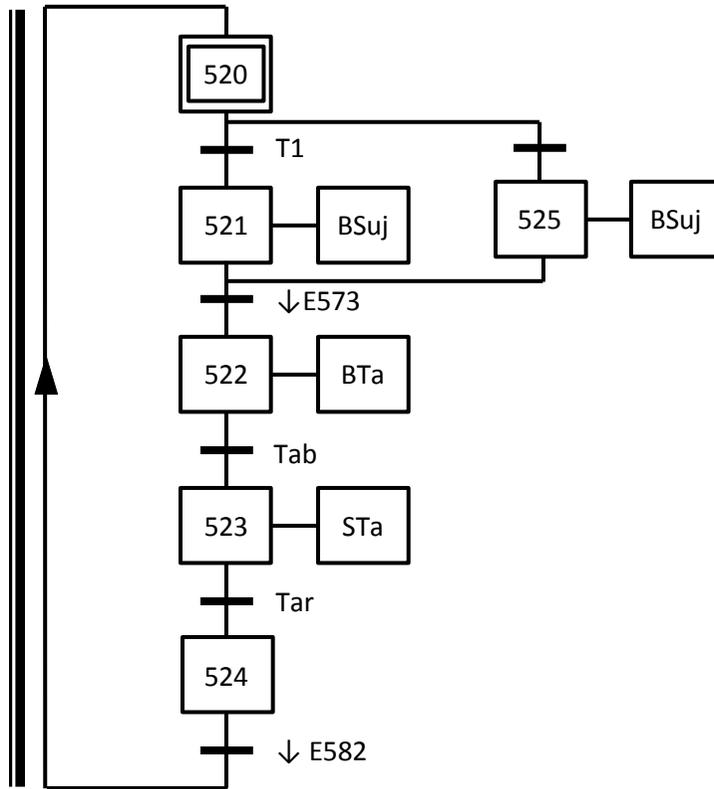
#### 2. Bobina



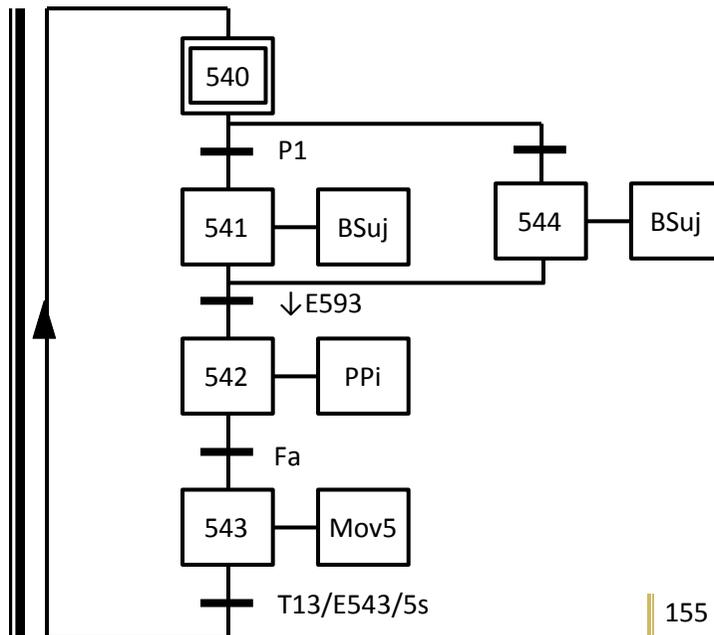
3. Corte



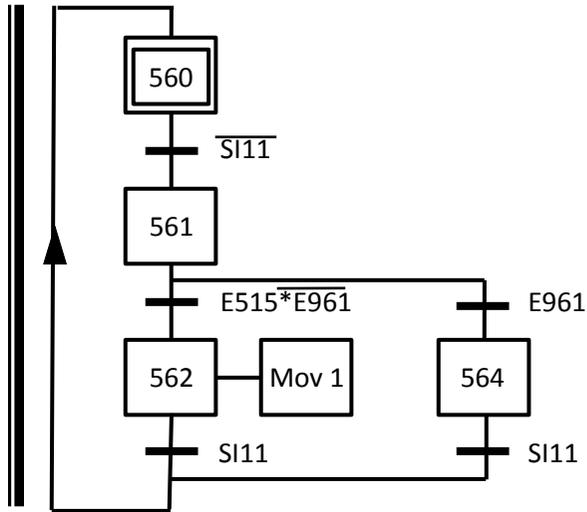
4. Taladro



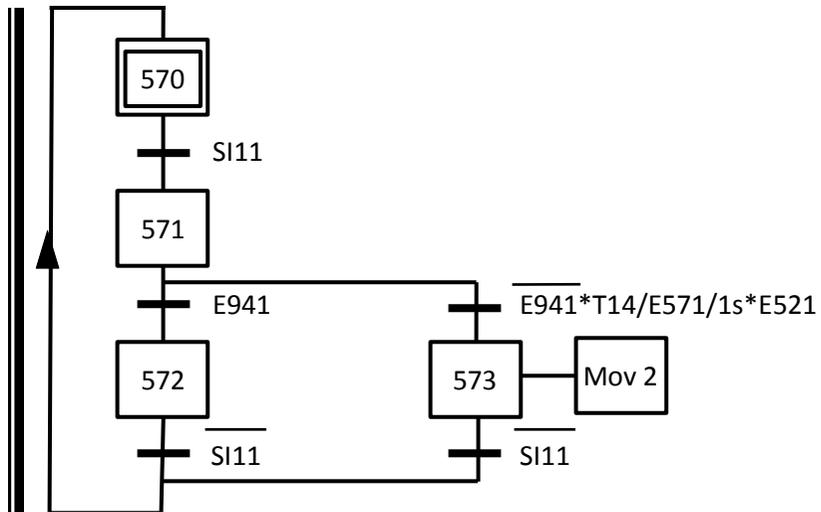
5. Pintura.



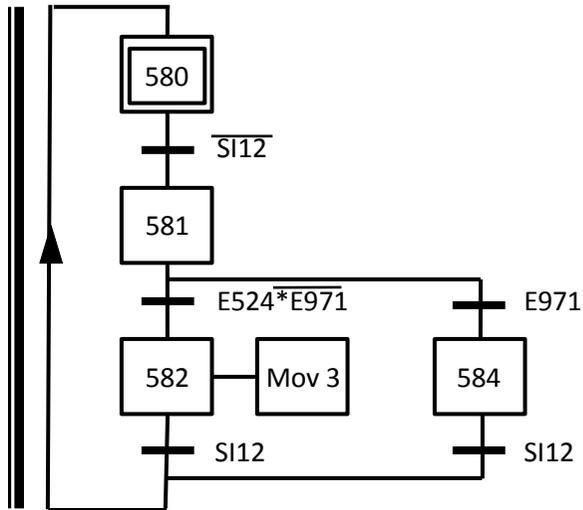
6. Intermedio 1, llegada de piezas.



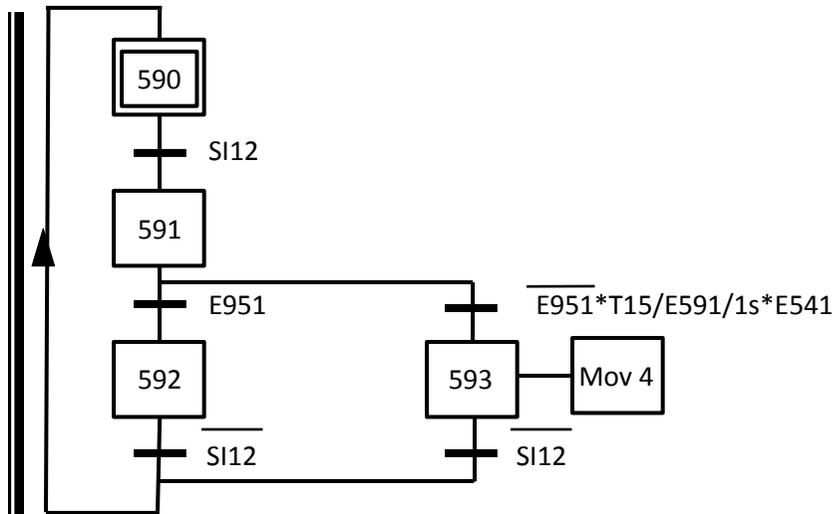
7. Intermedio 1, salida de piezas



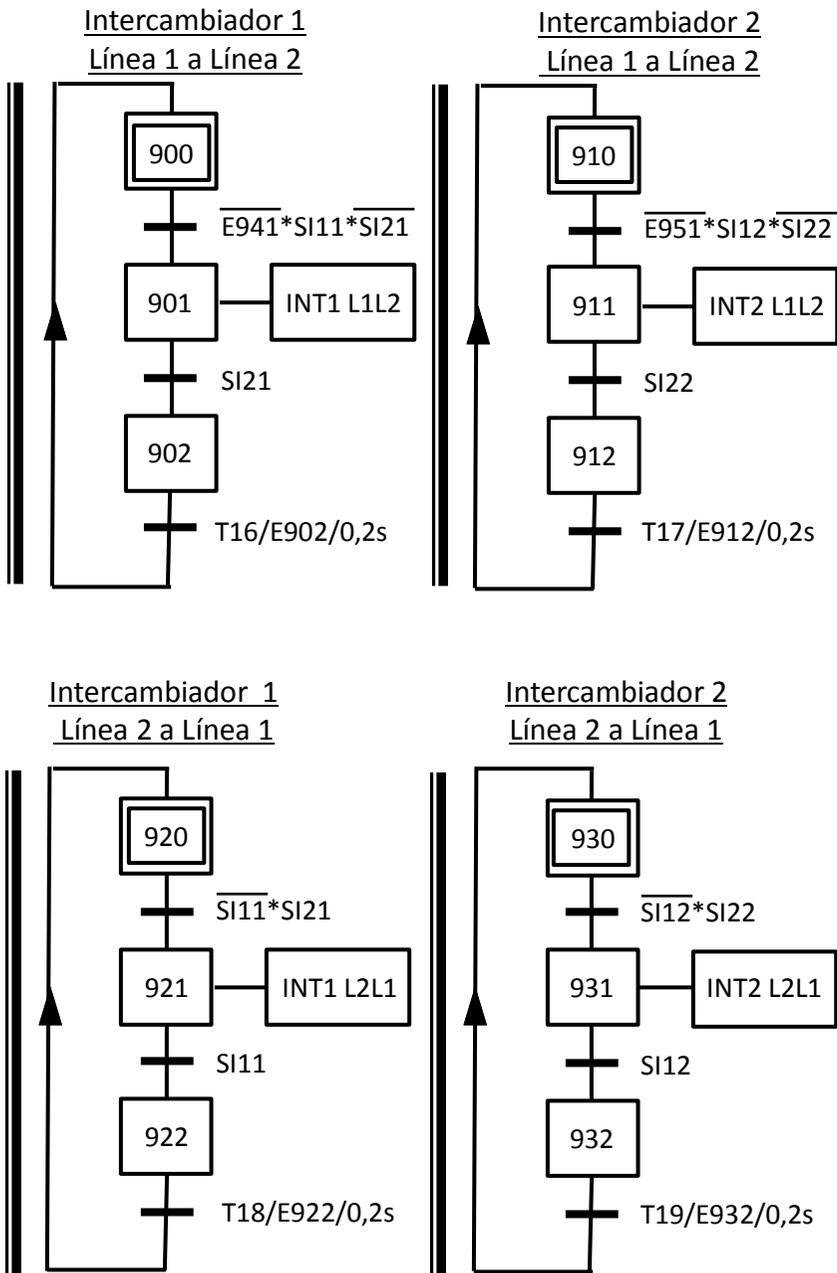
8. Intermedio 2, llegada de piezas



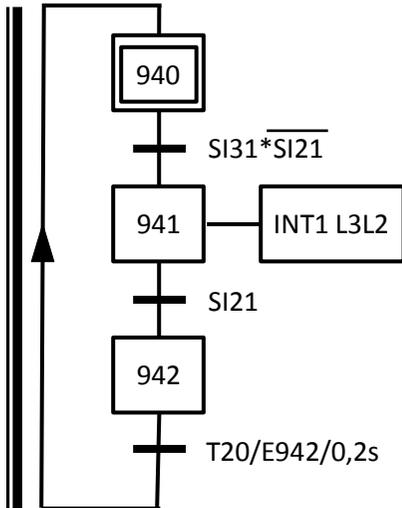
9. Intermedio 2, salida de piezas



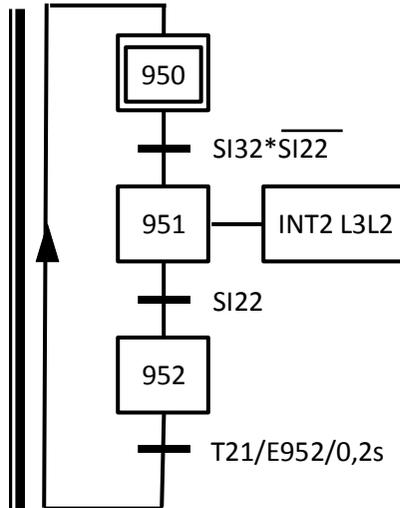
**Intercambiadores**



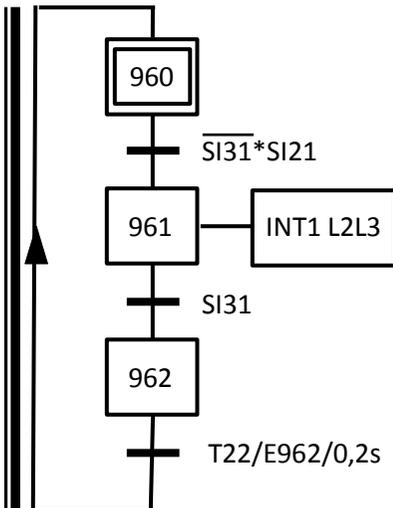
Intercambiador 1  
Línea 3 a Línea 2



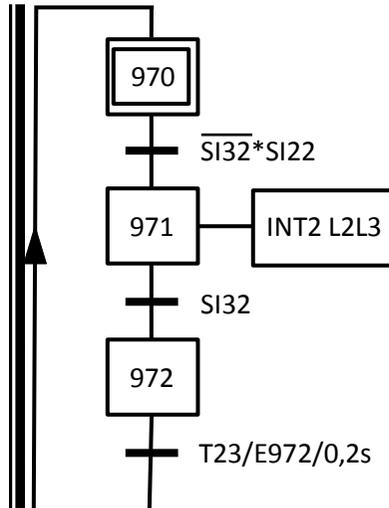
Intercambiador 2  
Línea 3 a Línea 2



Intercambiador 1  
Línea 2 a Línea 3



Intercambiador 2  
Línea 2 a Línea 3



# Bibliografía

---

- **Redes de Petri: Modelado e implementación de algoritmos para autómatas programables**, Luis Diego Murillo Soto.
- **Autómatas Programables Osrom CQM1/CPM1/CPM1A/SRM1**. Manual de Programación.
- **Tutorial Básico Labview**.
- **Lenguajes de programación**, Paloma Prieto.
- **Lenguajes Formales y autómatas**, Ramón Brena Pinedo.
- **“Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use”**. Volume 1. Springer\_Verlag, Berlin 1997. [JENSEN, 1997] Kurt Jensen.
- **Estructuras de datos y caminos, Apuntes Universidad de Cantabria**, Michael González Harbour.
- **Sistemas Industriales Distribuidos, Apuntes de Teoría**, Universidad de Valencia.
- **Camino más corto a partir de múltiples fuentes en un grafo**, Joemmanuel Ponce Garrido.
- ***On a Routing Problem*** in Quarterly of Applied Mathematics, Richard Bellman.
- ***Flows in Networks***, Princeton University Press, 1962. Lestor R. Ford jr., D. R. Fulkerson.
- ***Introduction to Algorithms***. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.