



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: **2014-15**

Contenidos

Documento n°1: Resumen.....	3
Documento n°2: Memoria.....	5
Documento n°3: Anexo a la memoria: Procesos FisherTechnik (Entradas-Salidas)	47
Documento n°4: Presupuesto.....	53
Documento n°5: Anexo de diseño.....	63
Documento n°6: Manual de programador	77
Documento n°7: Manual de usuario	109
Documento n°8: Planos	127



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº1: Resumen

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

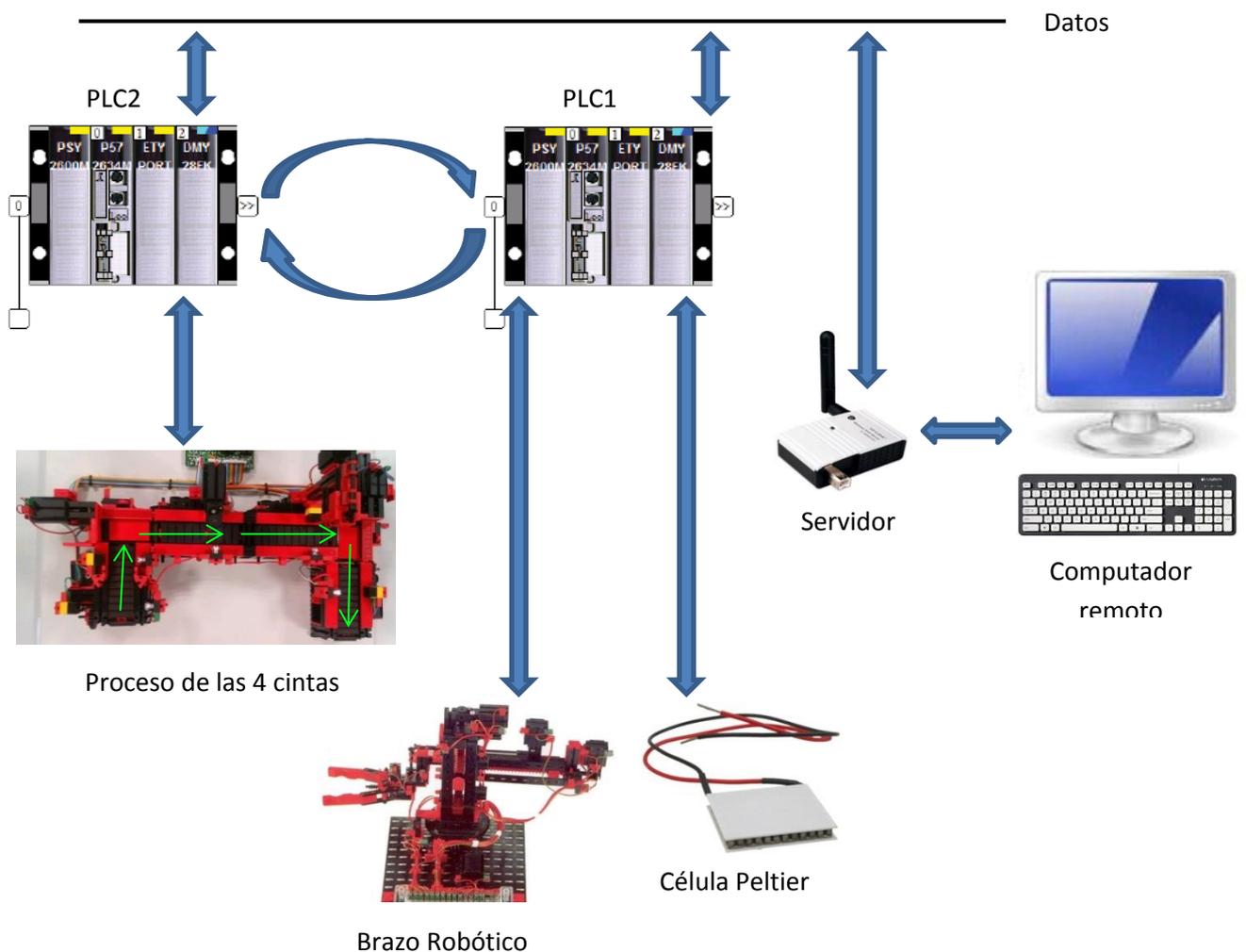
TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: **2014-15**

Resumen

En el presente proyecto se va a trabajar con los autómatas TSX Premium (PLCs), las maquetas de FisherTechnik correspondientes a la línea indexada y al brazo robótico, la célula Peltier y programas como LabVIEW y Unity Pro, pertenecientes a la línea de software profesional y muy presentes en el entorno industrial actual. El objetivo se define como la correcta automatización de un proceso que realizan conjuntamente la línea y el brazo y que está controlado por los autómatas. Será preciso por tanto una comunicación efectiva entre los autómatas para evitar desincronizaciones. Al proceso se le añade un control híbrido que se diseñará para la célula Peltier. El mismo debe proporcionar un control sobre la temperatura de referencia así como un rechazo de las perturbaciones. El proceso será monitorizado y controlado desde un computador remoto que permitirá manejar todas las funciones según unos permisos asignados. Para diferenciar esos permisos se establecerá un dominio donde se crearán las diversas cuentas de usuario atendiendo a la capacidad y jerarquía de cada usuario. Para la monitorización será precisa la creación de un servidor para la publicación de datos.





UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº2: Memoria

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenidos

1 Objeto del Proyecto	10
2 Motivación	12
3 Descripción del funcionamiento deseado	12
4 Especificaciones	15
5 Justificación	16
6 Descripción del Equipo	17
6.1 Introducción.....	17
6.2 Maqueta Línea indexada con dos unidades de mecanizado FisherTechnik.....	17
6.3 Maqueta 3-D-Robot TX FisherTechnik.....	18
6.4 Célula Peltier	18
6.5 PLC.....	21
6.6 TAD.....	22
6.7 Fuente de alimentación variable	22
6.8 Fuente de tensión controlable	22
6.9 LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench).....	23
6.10 Matlab.....	24
6.11 Unity Pro.....	25
7 Desarrollo de la solución	26
7.1 Descripción de la solución obtenida:	26
7.2 Esquema de la solución obtenida.....	27
7.3 Estructura que se ha aplicado.....	28
7.4 Esquema de desarrollo del proyecto	29
7.5 Particularidades en el diseño de los Graficets.....	30
7.5.1 Modos de Marcha.....	30
7.5.2 Modos de Alarma	30
7.5.3 Detección del proceso a realizar en la pieza.....	30
7.6 Comunicación entre autómatas	31
7.7 Aplicación en LabVIEW para la adquisición de datos	32
7.7.1 Preparación.....	32
7.7.2 Toma de datos	33
7.7.3 Identificación del modelo.....	33
7.7.4 Diseño del regulador	33
7.7.5 Validación del regulador:	33
7.8 Implementación del regulador en Unity Pro.....	34

7.8.1	Diseño del bloque de función PI_B.....	34
7.8.2	Validación del comportamiento real/simulado.....	35
7.9	Sistema de usuarios.....	35
7.9.1	Creación dominio jerárquico de usuarios e implementación.....	35
7.10	Publicación de datos	37
7.11	Incorporación al proyecto de las Bound Variables	38
7.12	Monitorización y control remoto	39
7.12.1	Pestaña lateral.....	39
7.12.2	Pestaña Inicio	40
7.12.3	Pestaña Cintas	40
7.12.4	Pestaña Robot.....	41
7.12.5	Pestaña Refrigeración.....	41
7.12.6	Pestaña Histórico.....	42
7.13	Almacenamiento de datos.....	42
8	Conclusiones	43
9	Ampliaciones.....	44
10	Bibliografía	46

Índice de Figuras

Figura 1. Maquetas Fisher-Technik de las cintas y del brazo robótico.....	11
Figura 2. Pestaña de “Cintas” de la pantalla diseñada para monitorizar el proceso.....	11
Figura 3. Montaje de la Célula Peltier en laboratorio	11
Figura 6. Aspecto de la pieza tras el proceso 1	13
Figura 7. Fresadora con cabezal grande practicando el agujero de los cilindros	13
Figura 4. Fresadora con cabezal grande actuando sobre el bloque sólido.....	13
Figura 5. Fresadora con cabezal pequeño actuando sobre el bloque sólido	13
Figura 9. Pieza correcta tras realizar los dos procesos	13
Figura 8. Fresadora con cabezal pequeño realizando los retoques en el cilindro	13
Figura 10. Diagrama de funcionalidades	14
Figura 11. Maqueta de la línea indexada	17
Figura 12. Maqueta del brazo robótico.....	18
Figura 13. Célula Peltier	18
Figura 14. Temperatura cara fría (azul) y temperatura de la cara caliente (rojo) de célula Peltier sin ventilador	19
Figura 15. Temperatura cara fría (azul) y temperatura de la cara caliente (rojo) de célula Peltier con ventilador	19
Figura 16. Equivalencia grados-voltios de los sensores de la célula Peltier	20
Figura 17. Equivalencia porcentaje de actuación-voltios de los actuadores de la célula Peltier	20
Figura 18. Imagen del autómeta	21
Figura 19. Diagrama de comunicación del autómeta	21
Figura 20. Fuente de tensión variable	22
Figura 21. Fuente de tensión controlable	22
Figura 22. Panel frontal (izquierda) y diagrama de bloques de LabVIEW.....	23
Figura 23. Ventana de comandos de Matlab	24
Figura 24. Interfaz de la herramienta Ident de Matlab.....	24
Figura 25. Interfaz de la herramienta Simulink de Matlab	25
Figura 26. Interfaz del programa Unity Pro.....	26
Figura 27. Esquema de interconexión del equipo.....	27
Figura 28. Esquema de desarrollo del proyecto	29
Figura 29. Bloque de función FIFO	31
Figura 30. Diagrama de funcionamiento de la cola FIFO	31
Figura 31. Configuración de la comunicación entre autómetas. Ventana Ethernet	32
Figura 32. Esquema del montaje de la célula Peltier	32

Figura 33. Panel frontal de la aplicación para la toma de datos.....	33
Figura 34. Respuesta simulada del regulador calculado	34
Figura 35. Bloque de función PI_B.....	34
Figura 36. Comparativa entre la respuesta simulada (roja) y la real (azul) del regulador	35
Figura 37. Imagen de las pestañas y los elementos de control de usuario.....	36
Figura 38. Ventana de gestión del dominio.....	36
Figura 39. Diferentes permisos de usuario.....	37
Figura 40. Ventana del servidor.....	37
Figura 41. Bound Variables del proyecto.....	38
Figura 42. Pestaña lateral.....	39
Figura 43. Pestaña Inicio.	40
Figura 44. Pestaña Cintas.....	40
Figura 45. Pestaña Robot.....	41
Figura 46. Pestaña Refrigeración	41
Figura 47. Pestaña Histórico	42
Figura 48. Archivo de datos	42

1 Objeto del Proyecto

El objeto del presente proyecto es el desarrollo y la implementación de la automatización de una planta para el procesado de piezas (acabado superficial y de arranque de viruta) mediante PLC.

En concreto el proyecto cubre las siguientes tareas:

1. Diseño de un sistema SCADA que permitirá la monitorización y gestión mediante computador remoto a tiempo real y el almacenamiento de datos de interés para su posterior análisis. Exactamente, las funciones añadidas al sistema SCADA serán:
 - a. Creación de un dominio con una jerarquía de usuarios.
 - b. Control de los permisos de usuarios.
 - c. Monitorización de los diversos elementos que componen el sistema en diversas pestañas.
 - d. Control permanente de los modos de funcionamiento y la gestión de alarmas.
 - e. Sistema de almacenamiento de datos
2. Diseño e implementación de los automatismos que gobiernen la planta.
3. Diseño de un sistema de control de la temperatura de la máquina de fresado que incorpora la planta. El control se realizará por medio de una célula Peltier.

:

En las *Figuras 1, 2 y 3* se muestra el montaje completo de los diferentes elementos que han servido para desarrollar el proyecto en el laboratorio donde se han utilizado diferentes maquetas para simular el proceso a automatizar.

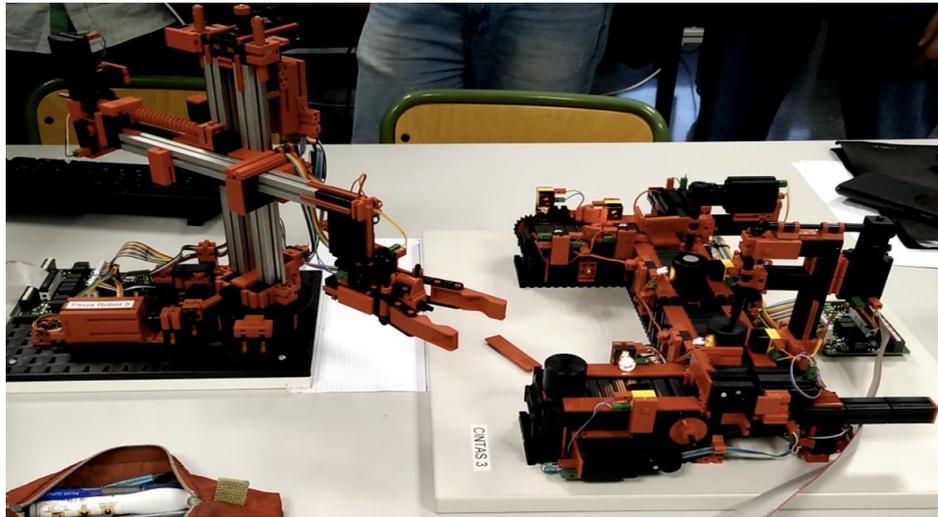


Figura 1. Maquetas Fisher-Technik de las cintas y del brazo robótico



Figura 2. Pestaña de "Cintas" de la pantalla diseñada para monitorizar el proceso



Figura 3. Montaje de la Célula Peltier en laboratorio

2 Motivación

Uno de los procesos más comunes en el mundo de la industria es el proceso de acabado superficial o de arranque de viruta. Las piezas se conforman inicialmente con una tolerancia grande y posteriormente son tratadas con diversas máquinas: fresadoras, taladradoras... De esta forma se consigue mejorar el acabado superficial y dotar a la pieza de las dimensiones funcionales. En la actualidad, muchas empresas utilizan máquinas separadas para conformar la primera forma y para tratarlas superficialmente después. En piezas de pequeñas dimensiones esto no suele ocasionar problemas de transporte, aunque si se almacenan en cajas, el peso de las mismas podría suponer el mismo problema. El problema reside en un peso elevado. Sin embargo, el caso que nos ocupa es el de los bloques motores. Se trata de piezas de grandes dimensiones y con un peso elevado, lo que conlleva la imposibilidad de traslado a mano. Actualmente las empresas utilizan puentes grúa para desplazar piezas tan grandes y ello supone que un operario esté siempre atento al traslado, se encargue de manejar cada vez el brazo grúa y de situar la pieza correctamente en la cinta a la que se transporta. No hace falta mencionar que es un tiempo largo y en ocasiones crea cuellos de botella en la línea de producción. Se ha pensado un nuevo sistema basado en el principio de las células flexibles, para reducir los tiempos de espera y optimizar la producción. Una célula flexible *“representa el intento de diseñar fábricas que sean capaces de funcionar permanentemente de forma automatizada, sin necesidad de la intervención de operadores humanos. Se sustentan, por lo tanto, más en la introducción de la automatización que en la reorganización del flujo del proceso”*.¹

3 Descripción del funcionamiento deseado:

A modo de dotar de aplicación práctica al proyecto, se ha decidido ver en qué campo de la industria actual podría ser el proyecto de interés. Finalmente se ha optado por la fabricación de motores². Se ha decidido que podría aplicarse a una célula flexible para el conformado de un bloque motor a partir del sólido.

El bloque motor en sólido es posicionado en la cinta de carga y avanza hasta la fresadora con el cabezal más grande que se encarga de hacer el fresado más superficial (*Figura 4*) y posteriormente los retoques con la fresa pequeña (*Figura 5*).

El aspecto de la pieza tras el proceso 1 se muestra en la *Figura 6*.

Para comenzar el proceso 2, un brazo robótico desplaza el bloque de nuevo a la cinta de carga. En muchas empresas se utilizan actualmente los puentes grúa para el desplazamiento de las piezas, en este caso se ha optado por el modelo de célula flexible y es un brazo robótico el que desplaza el bloque.

¹ [11] (Rodríguez) Referencia tomada de la siguiente página electrónica:

<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100002/lecciones/taxonomia/fabricacionflexibel.htm>

² [9] (Liebherr) Durante una estancia de realizada en la fábrica que tiene la empresa Liebherr en Ehingen, Alemania, se pudo asistir de primera mano al proceso de fabricación de motores de esta empresa. Basándose en esta experiencia, así como en documentos de presentaciones realizadas por la empresa (*“Abgasgrenzwerte für Dieselmotoren und ihre Auswirkungen auf LWE”* y *“Auszüge aus der Innerbetrieblichen Shweißrichtlinie ISR B – 010 mit praktischen Ergänzungen im Alltag”*) se ha pensado en un proceso de fabricación de motores que sea novedoso a la par que practicable en el mundo de la industria actual.

En el proceso 2 con la fresadora de cabezal grande se practican los agujeros de los cilindros (*Figura 7*). Posteriormente se finalizan con una fresa de menor tamaño (*Figura 8*).

De manera no regulada se aplica al final un análisis metrológico (inspección) para garantizar que las medidas son las correctas y observar en qué estado están los cabezales de la fresadora.

Si todo es correcto, se lleva la pieza al almacén donde espera a que los operarios comiencen con las labores de colocación de los pistones, bielas, árbol de levas, transmisión, etc. La pieza actual tendrá el aspecto que a continuación se muestra en la *Figura 9*³.



Figura 4. Fresadora con cabezal grande actuando sobre el bloque sólido



Figura 5. Fresadora con cabezal pequeño actuando sobre el bloque sólido



Figura 6. Aspecto de la pieza tras el proceso 1



Figura 7. Fresadora con cabezal grande practicando el agujero de los cilindros



Figura 8. Fresadora con cabezal pequeño realizando los retoques en el cilindro



Figura 9. Pieza correcta tras realizar los dos procesos.

³ Las figuras 4-9 han sido tomadas del siguiente vídeo: <https://www.youtube.com/watch?v=QsmileAkE-o>

En la *Figura 10* se presenta un diagrama de funcionalidades que esquematiza la secuencia que sigue el motor.

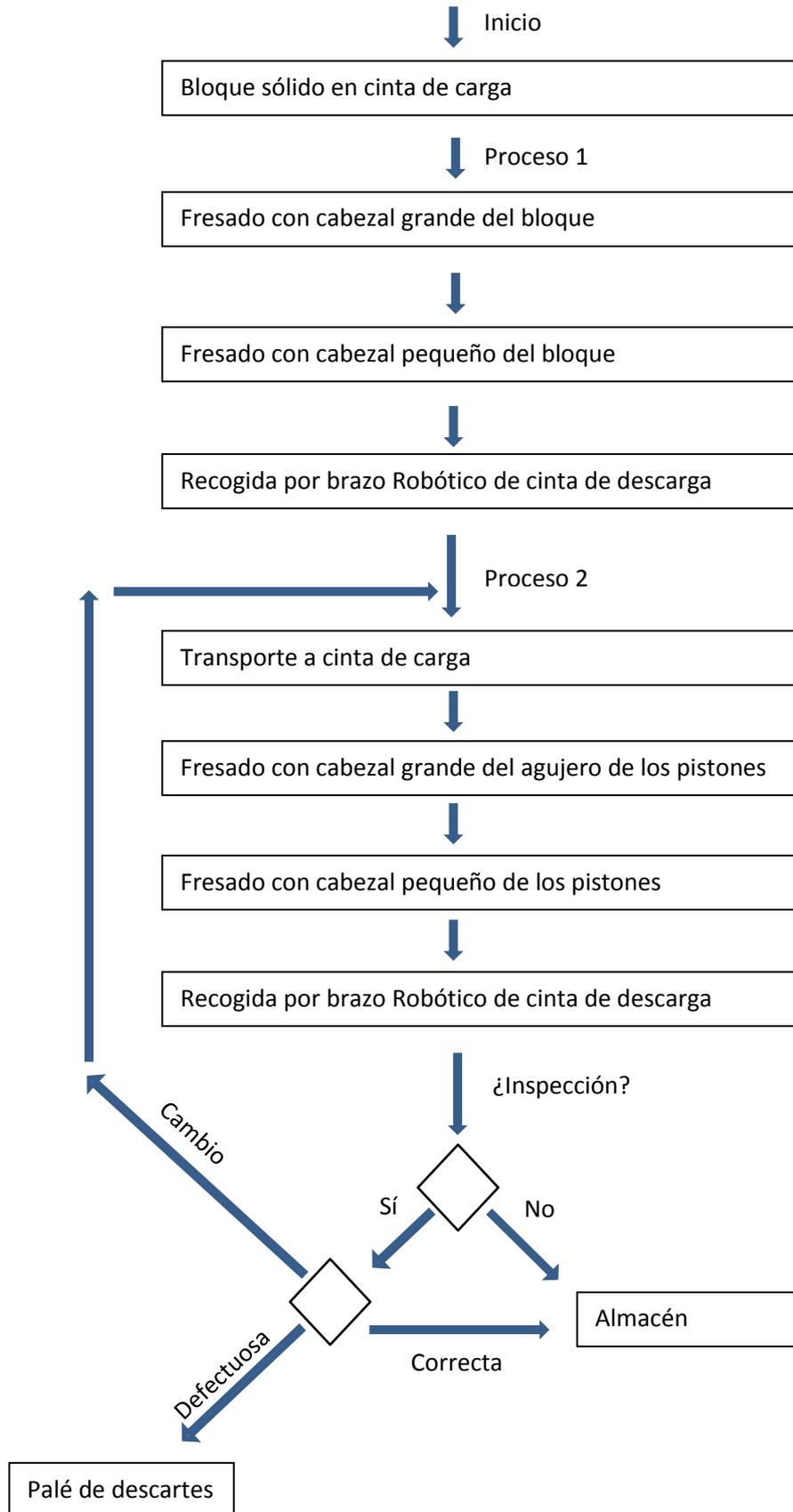


Figura 10. Diagrama de funcionalidades

4 Especificaciones:

A continuación se enumeran las especificaciones particulares de cada una de las partes que componen el proyecto (Cintas, Robot, Peltier y SCADA):

- **Cintas:**

1. Programar las cintas para el funcionamiento normal distinguiendo cuándo se realiza el proceso 1 y cuándo el proceso 2.
2. Distinguir entre dos modos de marcha, uno automático y otro paso a paso en el cual se podrá manejar diferentes elementos de las cintas de manera individual (fresadora, cinta 1...).
3. Controlar mediante los diferentes modos de alarma el funcionamiento de los elementos. Se podrá parar el proceso en su situación actual, reiniciarlo o realizar una inspección de piezas.

- **Brazo Robot:**

1. Programar el funcionamiento normal del robot para que de manera autónoma pueda:
 - Recoger las piezas de la cinta de descarga.
 - Llevar la pieza a la cinta de carga.
 - Llevar la pieza al almacén.
 - Llevar la pieza al palé de descartes.
2. Distinguir entre tres modos de marcha:
 - Automático: Funcionará de manera autónoma hasta que lleve al almacén un número de piezas que previamente se le ha indicado
 - Ciclo a ciclo: Recogerá la pieza y la llevará al siguiente checkpoint.
 - Paso a paso: El Robot podrá ser manejado por un usuario pudiendo éste moverlo en cualquiera de sus 4 grados de libertad.

Cualquier modo de marcha debe mantenerse hasta que se indique un nuevo modo o se cumplan los requisitos.

3. Controlar mediante diferentes modos de alarma el bloqueo del funcionamiento del brazo. Se deben diseñar tres modos para:
 - Parar el funcionamiento donde se encuentre.
 - Reiniciar el sistema del brazo.
 - Detenerse antes de ir a por piezas mientras éstas se inspeccionan y se decide qué hacer con cada una.

- **Peltier:**

1. Identificación del modelo en la zona lineal escogida. Se ha estimado la temperatura de refrigeración, que se tomará como la referencia, en unos 4°C. El modelo debe adecuarse a la zona alrededor de esta referencia. (0-10°C).
2. Cálculo de un controlador PID lo más simple posible.
3. Control efectivo de la temperatura de la cara fría.
4. Implementación de dos modos de funcionamiento:

- Automático: aplicará el controlador calculado anteriormente.
 - Manual: aplicará directamente la acción que se le indique.
- **Scada:**
 1. Proporcionar un sistema de control y supervisión al que se pueda acceder de forma remota.
 2. Control de usuarios y asignación de permisos diferentes para cada uno de ellos, procurando como premisa una gestión de las pantallas fácil e intuitiva.
 3. Visualización del conjunto y de qué procesos se están ejecutando en cada instante.
 4. Visualización particular de las cintas y poder manipular los elementos de las cintas desde el acceso remoto.
 5. Visualización particular del brazo robótico y poder manejar el brazo en sus 4 grados de libertad desde el acceso remoto.
 6. Visualización y control de los parámetros de la Peltier de manera directa así como permitir el almacenamiento de los mismos para el posterior análisis.
 7. Visualización de datos de interés para los usuarios de nivel alto.
 8. Control de los modos de marcha sin que interfieran entre sí.
 9. Control de los sistemas de alarmas procurando que actúen de forma rápida y precisa.

5 Justificación:

El presente proyecto se realiza para acreditar los conocimientos adquiridos en las asignaturas “Sistemas Automáticos”, “Tecnología Automática”, “Ciencia de Materiales”, “Informática” y “Laboratorio de Automatización y Control”.

6 Descripción del Equipo:

6.1 Introducción

En este capítulo se describirán brevemente todos los elementos que conforman el trabajo y sus respectivas funciones. Se ha trabajado con dos maquetas de la marca FisherTechnik, una correspondiente a unas cintas transportadoras con diversas unidades de mecanizado y la otra correspondiente a un brazo robótico. Ambas maquetas se programarán mediante el software de programación Unity Pro M en un PLC (Controlador Lógico Programable). Se ha trabajado también con una célula Peltier con un rango de actuación comprendido entre -34 a 70 °C. Para hacerla funcionar se precisa de algunos elementos accesorios: dos fuentes de tensión, una de ellas variable y otra controlable así como una tarjeta de adquisición de datos para realizar el modelo y su validación.

6.2 Maqueta Línea indexada con dos unidades de mecanizado FisherTechnik

La *Figura 11*⁴ muestra la maqueta de las cintas sobre la que se ha trabajado en laboratorio.



Figura 11. Maqueta de la línea indexada

Dispone de 5 barreras fotoeléctricas de lógica negada, 4 sensores de final de carrera (sin potencial) así como 8 motores de CC para mover cada uno de los elementos.

Una descripción más detallada de las entradas y salidas se podrá encontrar en el anexo a la memoria, 2 *Anexo de E/S de las cintas*.

⁴ (www.atdsas.com)

6.3 Maqueta 3-D-Robot TX FisherTechnik

La *Figura 12*⁵ muestra la maqueta del brazo robótico sobre la que se ha trabajado en laboratorio.



Figura 12. Maqueta del brazo robótico

Brazo Robótico con 4 sensores final de carrera y 4 encoders así como 4 motores de CC bidireccionales. Para más detalle en las entradas y salidas ver el anexo a la memoria, 1 *Anexo de E/S del brazo robótico*.

6.4 Célula Peltier

La *Figura 13*⁶ muestra el aspecto físico de una célula Peltier.

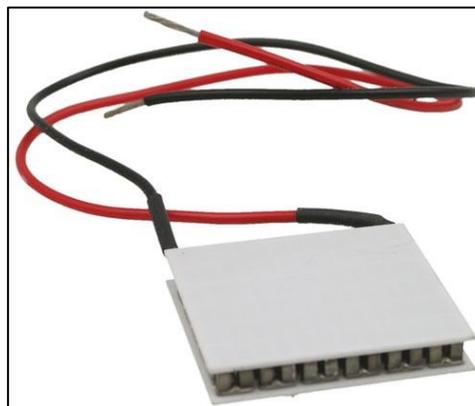


Figura 13. Célula Peltier

Esta célula se basa en uno de los efectos termoeléctricos conocidos, el efecto Peltier⁷. Por medio de corriente eléctrica es capaz de enfriar uno de sus bornes mientras el otro se calienta. Mediante este efecto, puede enfriar el lado frío hasta -34°C . El comportamiento de la Peltier sin embargo, no es lineal como se puede observar en la *Figura 14*.

⁵ (fischer-technik.es)

⁶ (www.faunaexotica.net)

⁷ [15] (Wikipedia) El efecto Peltier se describe pormenorizadamente en el link http://es.wikipedia.org/wiki/Efecto_termoel%C3%A9ctrico

Este ensayo realizado en laboratorio demuestra que, debido a la proximidad entre la cara fría y la caliente, la cara caliente influye sobre la fría calentándola. La solución que se ha adoptado es acoplar un ventilador que disipe el calor y conseguir así una distribución lineal entre la temperatura de la cara fría y la energía aportada. El resultado del mismo ensayo realizado con ventilador se muestra en la *Figura 15*.

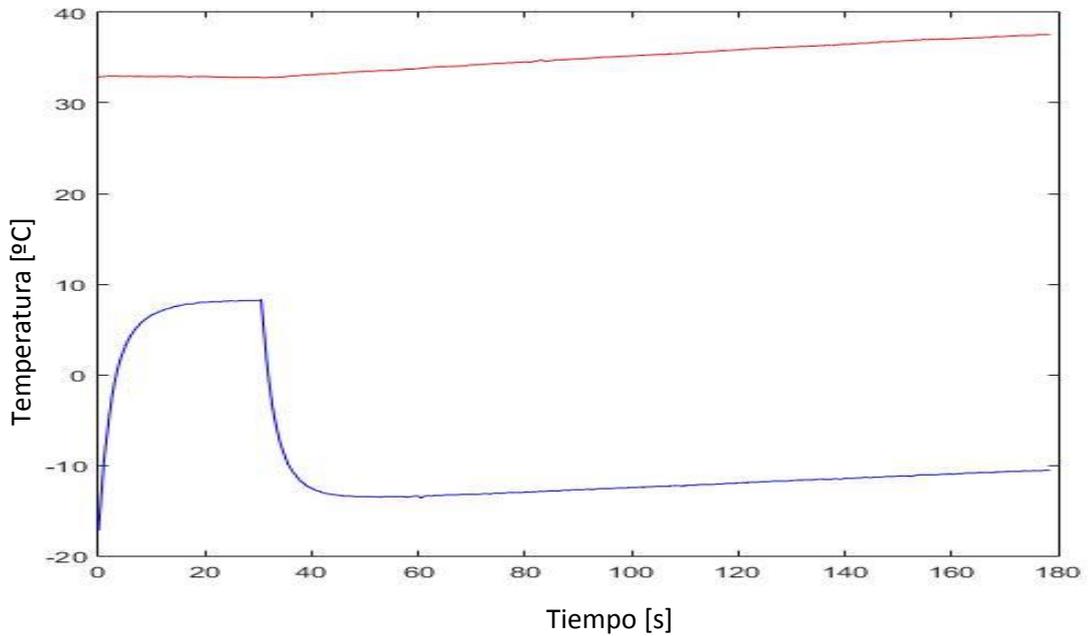


Figura 14. Temperatura cara fría (azul) y temperatura de la cara caliente (rojo) de célula Peltier sin ventilador

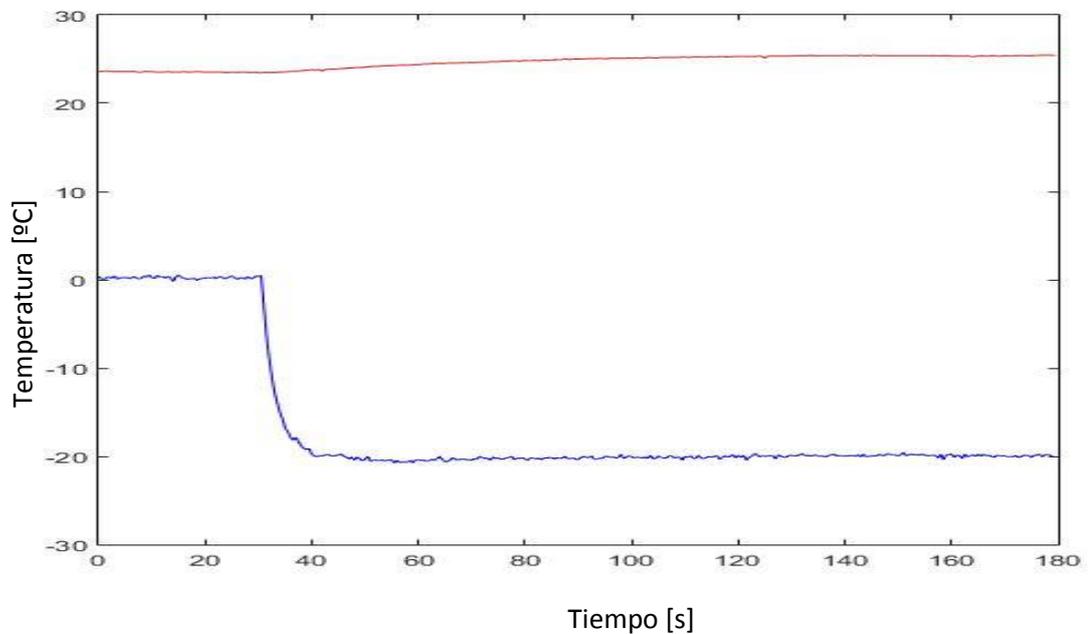


Figura 15. Temperatura cara fría (azul) y temperatura de la cara caliente (rojo) de célula Peltier con ventilador

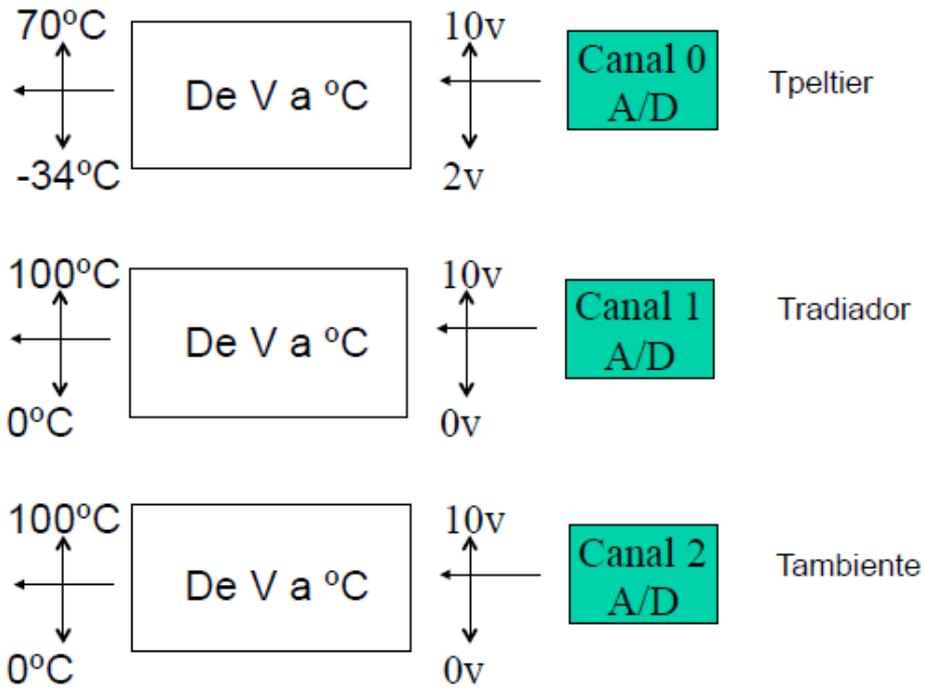


Figura 16. Equivalencia grados-voltios de los sensores de la célula Peltier

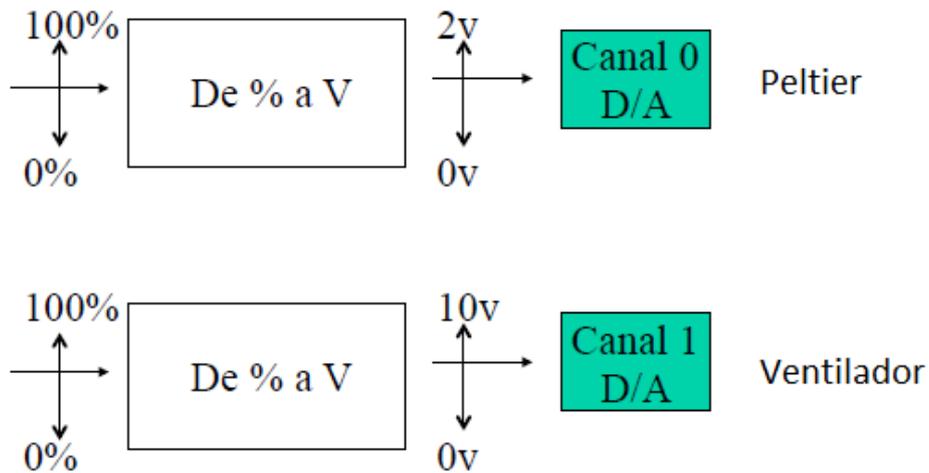
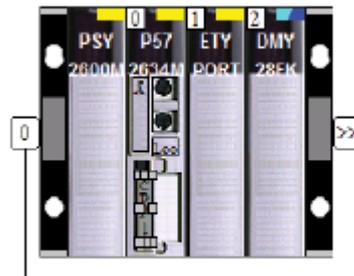


Figura 17. Equivalencia porcentaje de actuación-voltios de los actuadores de la célula Peltier

6.5 PLC



[Figura 18. Imagen del autómata

El autómata que se va a utilizar tiene el aspecto mostrado en la *Figura 18*. Se trata de un autómata TSX Premium que consta de un módulo de entradas/salidas digitales y un BUS CAN con dos nodos. El nodo1 tiene 2 módulos: uno de entradas/salidas digitales y otro de entradas/salidas analógicas. El nodo2 tiene un módulo de entradas/salidas digitales. La programación se realizará mediante el software de programación Unity Pro M y la transmisión de datos se muestra en la *Figura 19*. El módulo del autómata cuenta con 16 entradas y 12 salidas y los módulos de entradas/salidas digitales de cada nodo CAN, que son los que se van a utilizar, tienen 12 entradas y 8 salidas cada uno, tal y como se ve en la *Figura 19*. Cada entrada/salida es a su vez una palabra de 16 bits escalada en el rango 0-10000mV.

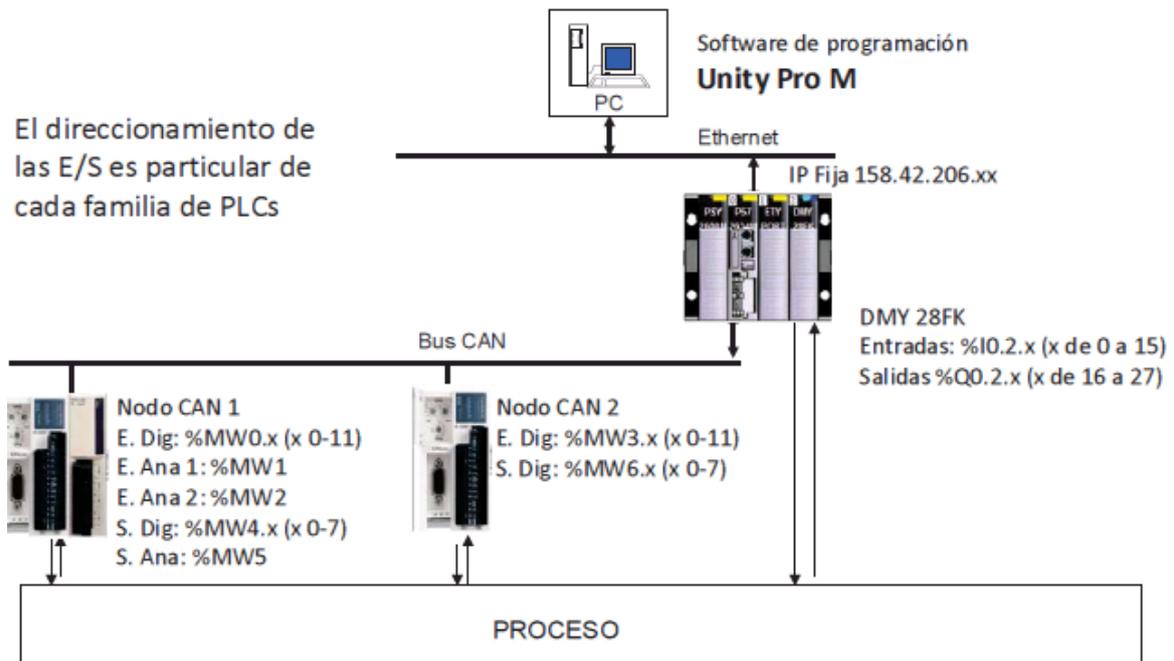


Figura 19. Diagrama de comunicación del autómata

6.6 TAD

La tarjeta de adquisición de datos que se va a emplear es la **ADLink PCI-9112**. Esta tarjeta es necesaria para la comunicación con el PC de los sensores y los actuadores de la célula Peltier. En consecuencia, la tarjeta será necesaria para la obtención y validación de un modelo mediante LabVIEW y Matlab gracias a la librería de la tarjeta de adquisición de datos que se instala en LabVIEW.

También se utilizará para la validación del controlador elegido. Posteriormente ya no será necesaria para la implementación en el PLC pues el propio PLC será el encargado de adquirir los datos.

La conexión realizada para la Peltier se muestra en el [Figura 32](#).

6.7 Fuente de alimentación variable

La fuente de alimentación variable será la encargada de alimentar todos los componentes de la célula Peltier. La tensión que aplica es de +/-15 voltios. Para la obtención del modelo, la potencia del ventilador se manejará desde la tarjeta de adquisición de datos, pero posteriormente, en el montaje final en el PLC se pondrá una tensión constante para el ventilador desde la fuente de tensión variable como se indica en la Figura 20.



Figura 20. Fuente de tensión variable

6.8 Fuente de tensión controlable

Tanto para la obtención del modelo como para la implementación en el PLC, esta fuente de tensión controlable se utiliza para amplificar la tensión aplicada directamente a la célula Peltier x3.

Además, limita de 0 a 5 amperios la intensidad que puede recibir directamente la Peltier, evitando acciones que puedan dañar la Peltier.



Figura 21. Fuente de tensión controlable

6.9 LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench)

LabVIEW es un entorno de desarrollo pensado para la medición y el control que acelera la productividad.

Empieza un lenguaje de programación G propio (programación gráfica) que permite el desarrollo dinámico de aplicaciones (*dynamic dataflow programming*). Dichas aplicaciones reciben el nombre también de **Instrumentos Virtuales** o **VI**. En el presente proyecto se han creado aplicaciones para:

- la toma de medidas por medio de librerías de adquisición de datos.
- creación de una pantalla para el control y monitorización del proceso.

Se ha elegido LabVIEW por diversas razones que se exponen a continuación:

- Amplia cantidad de elementos prediseñados con una apariencia funcional e intuitiva para la elaboración del interfaz de usuario.
- Versatilidad en el manejo conjunto de datos de LabVIEW con otros programas.
- Herramientas convencionales para la depuración de los Vis, ejecución paso a paso, flujo de datos, puntos de ruptura y monitorización, etc.
- Ejecución de los elementos de programación por flujo de datos.
- Gran variedad de librerías para la toma de medidas, cálculo de PIDs, etc.

La apariencia que muestra el programa se divide en dos ventanas, el panel frontal y el diagrama de bloques, este último es donde se programará.

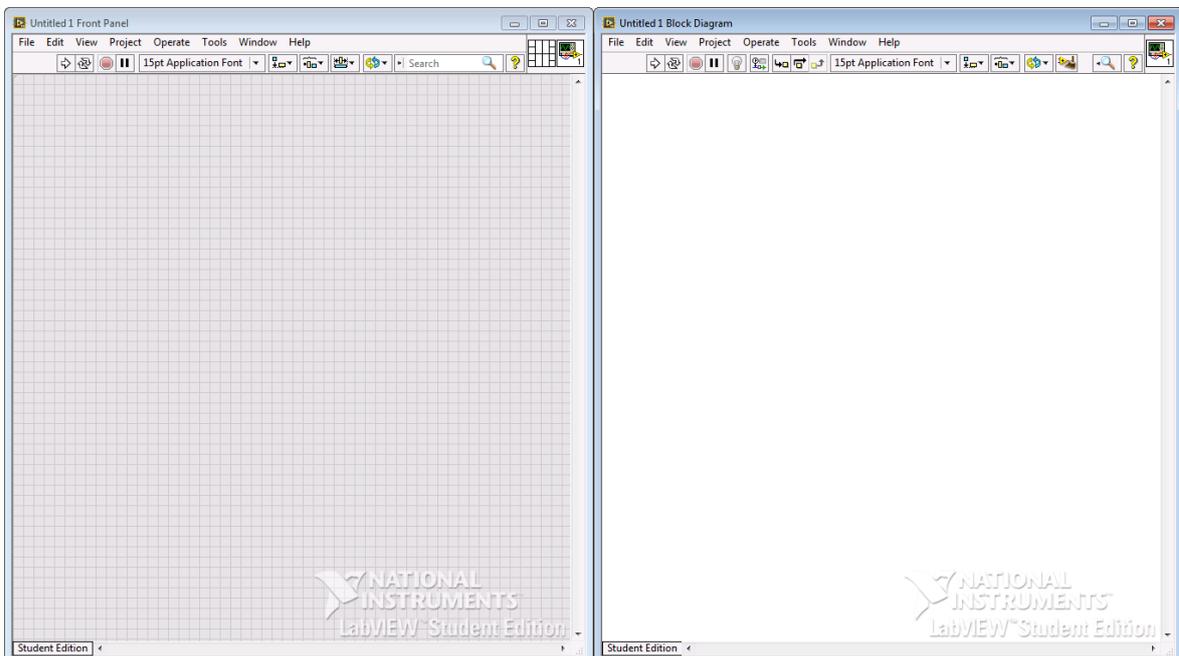


Figura 22. Panel frontal (izquierda) y diagrama de bloques de LabVIEW

6.10 Matlab

Matlab es un software de programación ampliamente extendido en el campo didáctico e industrial. Tiene su propio lenguaje de programación (lenguaje M) y como su nombre indica, es un software muy útil para desarrollos matemáticos. Esta capacidad se empleará para la identificación de modelos matemáticos y la validación posterior.

Se empleará el software Matlab R2015a con las extensiones:

- Herramienta Ident (*System Identification Toolbox*)
 - Se emplea para la identificación de modelos matemáticos a partir de un patrón.
- Herramienta Simulink
 - Se emplea para simular sistemas.

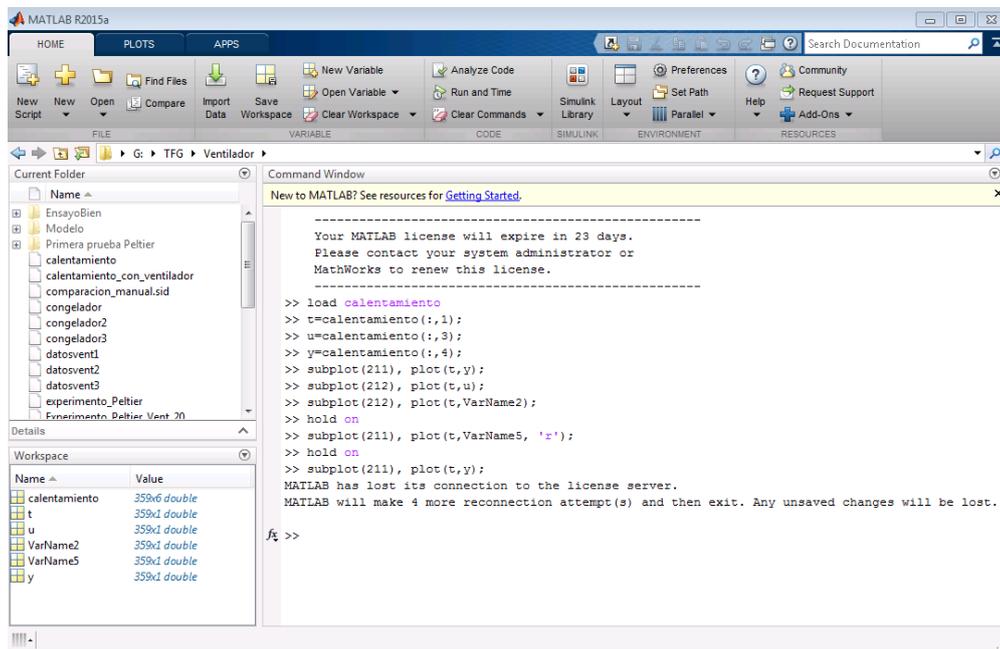


Figura 23. Ventana de comandos de Matlab

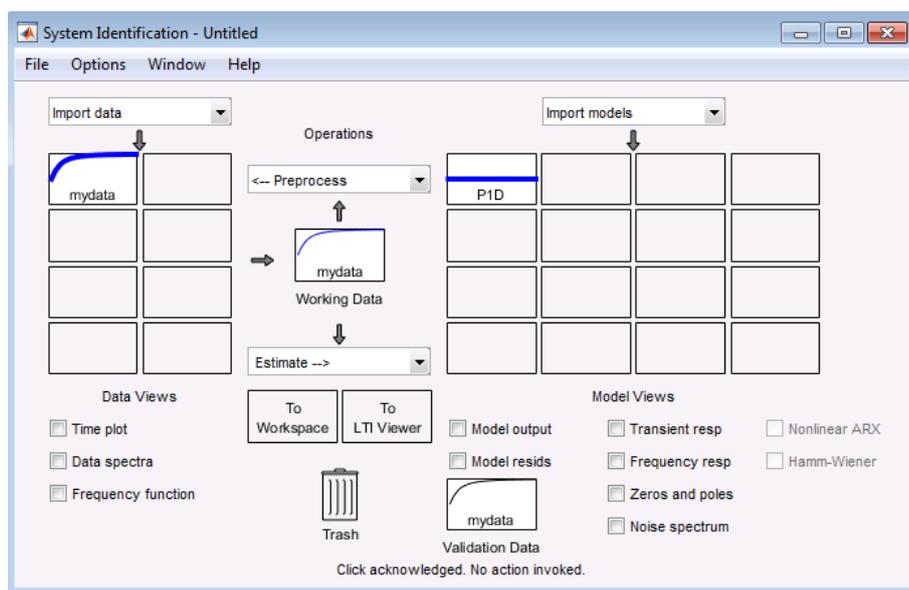


Figura 24. Interfaz de la herramienta Ident de Matlab

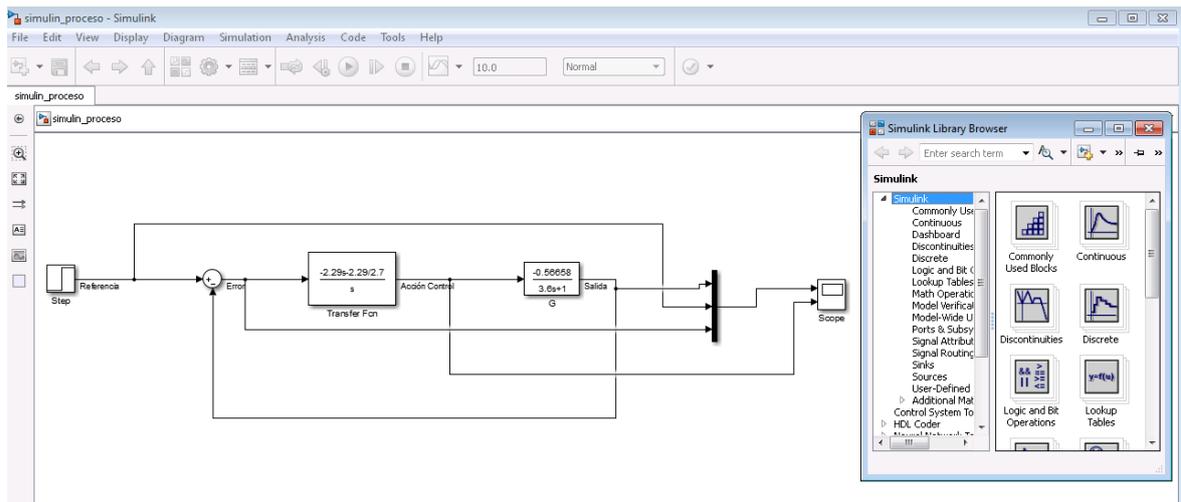


Figura 25. Interfaz de la herramienta Simulink de Matlab

6.11 Unity Pro

Unity Pro es un software de programación avanzado. El programa que finalmente se desarrollará será el resultado de la aplicación de los conocimientos sobre los lenguajes de programación que permite Unity Pro y que a continuación se detallan (SFC, LD y ST):

- SFC (*Secuencial Functional Chart*): También conocido como lenguaje Grafcet.
- LD (*Ladder*): También se conoce como programación en contactos o programación mediante operadores lógicos.
- ST (*Structured Text*): “Lenguaje estructurado de alto nivel, que es estructurado en bloques y sintácticamente similar a Pascal, aunque sus raíces provienen de Ada y C, pero más intuitivo para el ingeniero de control”⁸.

Mediante el sistema “Etapa-Transición” se puede crear un programa que controle las entradas y salidas del PLC.

La interfaz del Unity Pro será la que se presenta en la Figura 26.

⁸ [3] Universidad Nacional abierta y a distancia

Esta cita hace referencia a la siguiente página electrónica:

http://datateca.unad.edu.co/contenidos/2150512/ContenidoLinea/leccin_1633__texto_estructurado_structured_text__st.html

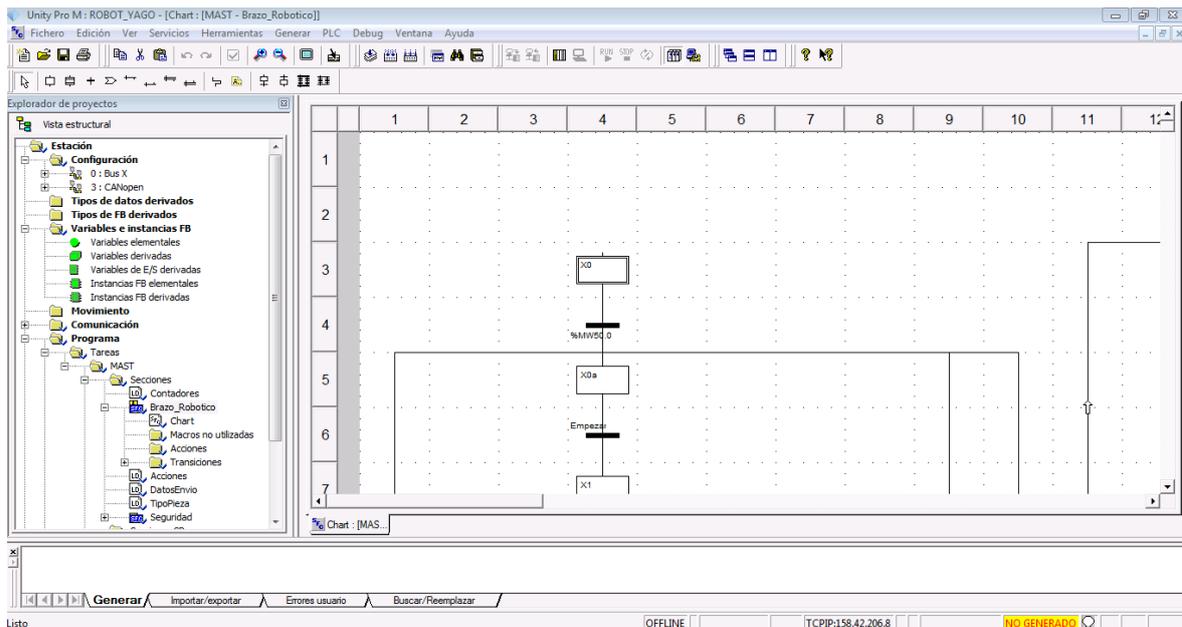


Figura 26. Interfaz del programa Unity Pro

7 Desarrollo de la solución

7.1 Descripción de la solución obtenida:

Se dispone de una maqueta de cintas transportadoras y un brazo robótico, razón por la que se ha adaptado el proceso real para poder simularlo en el laboratorio con las maquetas. Se utilizarán dos autómatas, uno para la maqueta del brazo robótico (PLC1) y otro para la maqueta de las cintas (PLC2).

En la *Figura 27* se puede apreciar cómo debe realizarse la conexión de los diferentes equipos. De igual manera, las flechas representan la dirección de los datos.

La línea indexada alterna entre el proceso 1 y el proceso 2 para cada pieza. A partir de ahora el proceso 1 se denominará Acabado Superficial y el proceso 2 recibirá el nombre de Cilindrado.

Es necesario que cada bloque motor realice el acabado superficial seguido del cilindrado.

Por lo tanto uno de los problemas que habrá que enfrentar es la necesidad de “marcar” cada pieza al inicio de la cinta de carga con el proceso que debe seguir.

La maqueta de las cintas debería ser mucho más compleja en la fresadora para poder simular correctamente el proceso de fresado, pero no siendo ésta la finalidad del proyecto, se ha optado finalmente por simular cada fresado con una acción de las unidades de mecanizado de las cintas de unos segundos de duración. A continuación se detallan los 2 procesos (Acabado superficial y Cilindrado):

- **Acabado Superficial:**

El fresado con cabezal grande se realizará en la primera unidad de mecanizado de la línea indexada y se simulará tan complejo proceso por varios segundos actuando seguidos cada uno por pequeños tiempos muertos (simularán la recolocación de la fresa en posiciones diferentes).

El fresado con cabezal pequeño se realizará con la segunda unidad de mecanizado de la línea indexada y se simulará con un tiempo de actuación sin interrupciones.

Posteriormente será recogido el bloque motor con el acabado superficial ya realizado por el brazo robótico que lo llevará de nuevo a la cinta de carga para que realice el segundo proceso, el de cilindrado.

- **Cilindrado:**

El fresado con cabezal grande para hacer los agujeros de los cilindros se realizará con la primera unidad de mecanizado con un proceso ininterrumpido de larga duración.

El fresado con cabezal pequeño se realizará nuevamente en la segunda unidad de mecanizado con una actuación idéntica a la del proceso de acabado superficial.

La pieza, ya acabada, será recogida nuevamente por el brazo robótico y transportada a una caja donde se almacenará.

Un esquema del trayecto realizado por una pieza desde que se deposita inicialmente en la cinta de carga hasta que es retirada por el brazo robótico y trasladada a la caja de almacenaje se puede ver en la [Figura 10](#).

7.2 Esquema de la solución obtenida

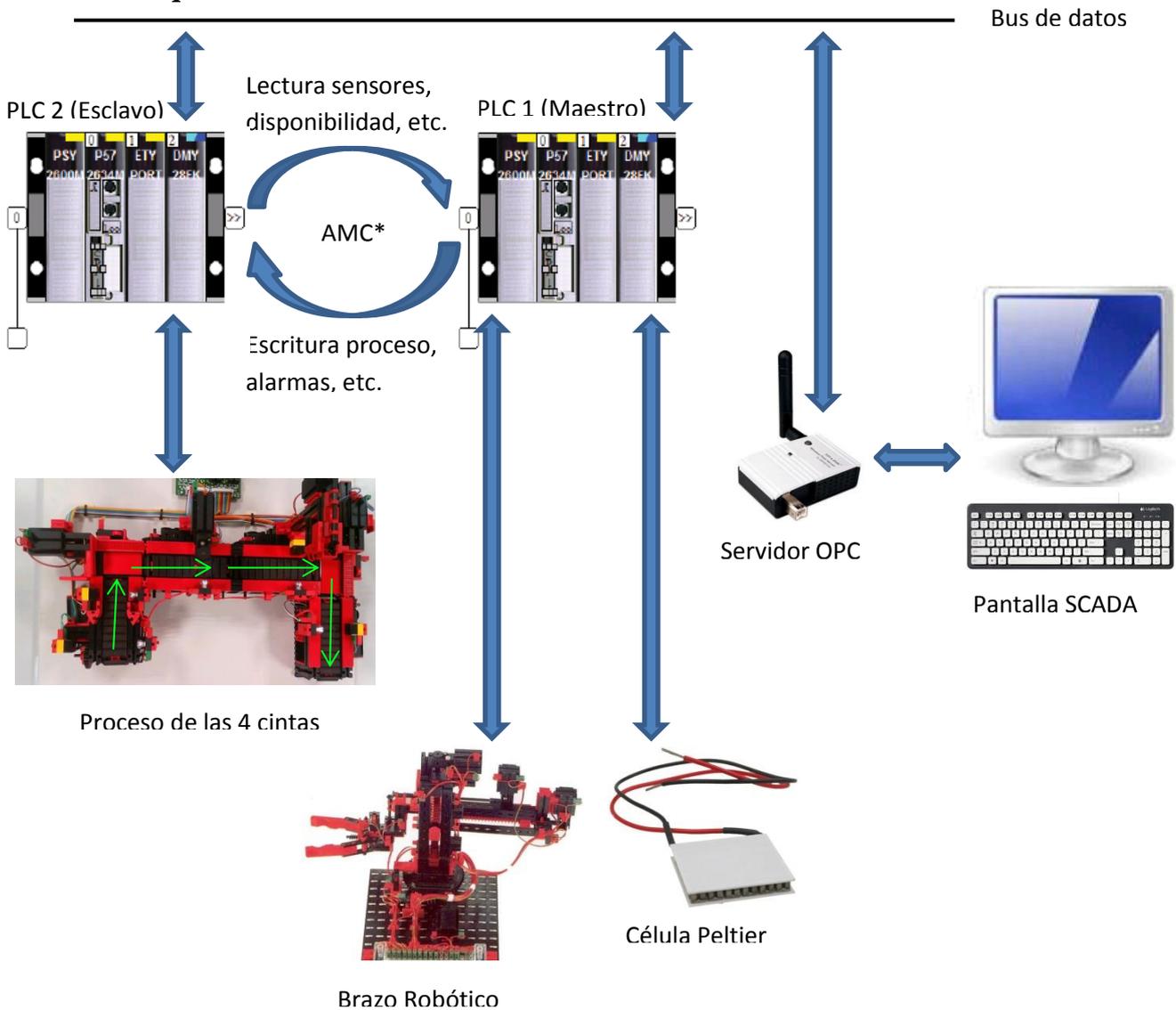


Figura 27. Esquema de interconexión del equipo

*AMC=Acceso a memoria compartida

7.3 Estructura que se ha aplicado

Para poder programar el proceso de fabricación se ha escogido una estructura de Grafkets parciales jerarquizados, tanto para la maqueta de las cintas, como para la del brazo robótico.

En el caso de las cintas, cada una de ellas se maneja por separado, por ello, a cada una le corresponde un Grafket diferente para controlarlas según lo deseado. Además, existe otro Grafket con mayor jerarquía que es el de seguridad y desde el que se fuerza al Grafket principal. Una descripción más detallada se tiene en el manual de programador Grafkets de la Línea indexada.

En el caso del brazo robótico se tiene un único Grafket que controla el funcionamiento general, además de un segundo que se encarga de la inicialización del Robot. Éste sólo se pone en marcha cuando el de funcionamiento general se lo ordena. Existe otro Grafket con mayor jerarquía que controla los modos de marcha y fuerza al de funcionamiento general. Además, existe otro Grafket con la mayor jerarquía, el de seguridad, que fuerza a todos los anteriores. Una descripción más detallada se tiene en el manual de programador Grafkets del Brazo Robótico.

En el laboratorio se dispone de dos maquetas que harán posible la simulación de todo el proceso. Una maqueta de unas cintas con una fresadora y una taladradora y un brazo robótico con 4 grados de libertad. Además de la simulación del proceso, se implementará una función añadida. Las máquinas de mecanizado que se han mencionado, fresadora y taladradora, son máquinas que trabajan con arranque de viruta, con lo que el contacto de metal con metal puede ocasionar un calentamiento que produzca la pérdida de propiedades del material como puede ser la dilatación por exceso de temperatura. Para conseguir tolerancias muy bajas, requeridas para un motor, este efecto termofísico resultaría inaceptable por lo que en el mundo industrial se practica un método llamado refrigeración por taladrina que consiste en que, mientras la herramienta trabaja, por un conducto situado junto a la herramienta se lanza un líquido refrigerante a presión que puede ser agua o aceite y que enfría el material y hace que el arranque de viruta ocurra a la temperatura óptima. Para simular esto, en el laboratorio se dispone de una célula Peltier que se programará y controlará para que, cuando la fresadora trabaje, la cara fría esté a la temperatura óptima y el resto del tiempo se mantenga a temperatura ambiente. Resumiendo, significa que cuando la fresadora esté actuando, la célula Peltier se enfría simulando el líquido refrigerante.

Además, se pretende crear un sistema SCADA para el control de cada uno de los elementos del sistema en tiempo real así como su directa supervisión e incluso el almacenamiento de los datos para el posterior análisis.

En la *Figura 28* se tiene un esquema del proceso seguido para la elaboración del proyecto.

7.4 Esquema de desarrollo del proyecto

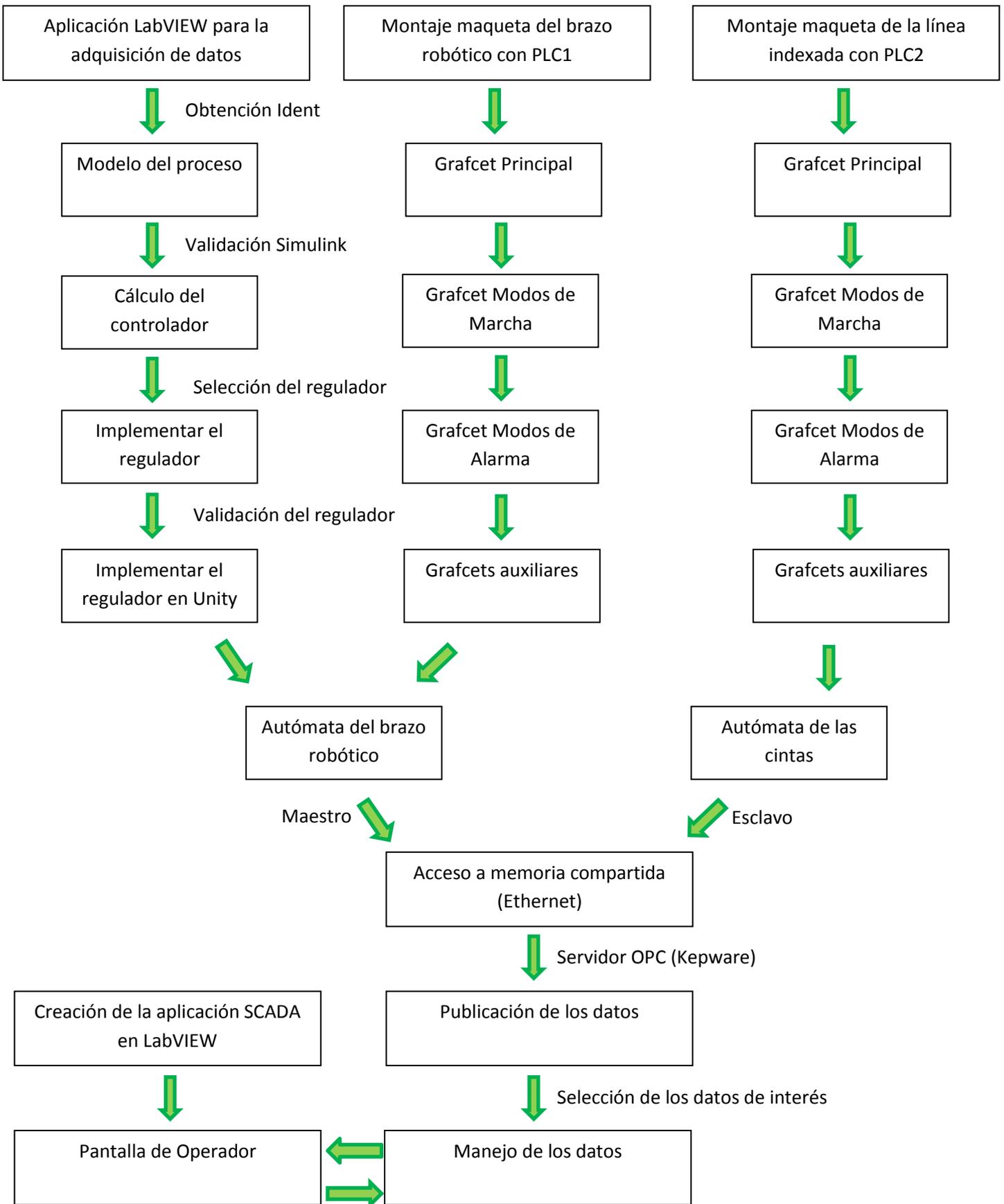


Figura 28. Esquema de desarrollo del proyecto

7.5 Particularidades en el diseño de los Graficets:

7.5.1 Modos de Marcha:

- **Automático:** Funcionamiento continuado del proceso en el que se le indica el número de piezas que ha de realizar y el proceso se realiza de forma ininterrumpida hasta que termina las piezas, cuando se detiene hasta nueva orden. También se puede salir simplemente pulsando cualquier otro modo de marcha o desactivando el modo automático.
- **Ciclo a ciclo:** Si se quiere controlar la ejecución de cada pieza individualmente, este ciclo permite que el usuario sea quien dé la orden cada vez que el brazo robótico tenga una pieza esperando en la cinta de descarga.
- **Paso a Paso:** Desde este modo se puede controlar la activación manual de las cintas, de la fresadora, de la taladradora así como mover el brazo como se desee en sus 4 grados de libertad.

7.5.2 Modos de Alarma

- **Inicializar:** Se debe parar el proceso y volver a iniciarse desde el principio.
- **Congelar/Parar:** Debe detener el proceso donde esté y no realizar ninguna acción. Posteriormente debe volver a reanudarse donde se quedó.
- **Salir:** Interrumpe el proceso y no se puede volver a iniciar hasta que se desconecte el sistema y se vuelva a conectar.
- **Inspección:** Evita que el robot vaya a por la pieza disponible hasta que ésta sea inspeccionada por un operario. Éste seleccionará si la pieza es correcta, hay que volverla a pasar por el proceso 2 o hay que desecharla.

Como se ha mencionado con anterioridad, cada pieza tiene que ser marcada en la cinta de carga con el proceso que va a seguir, o bien realiza el acabado superficial, o bien el cilindrado. Es imprescindible que cada pieza tenga una “marca” del proceso que se le ha de practicar.

7.5.3 Detección del proceso a realizar en la pieza

Explicación del bloque de función FIFO:

El conformado de un bloque motor funcional requiere de una ‘marca’. Esto podría, a priori, parecer sencillo si se sabe cuántas piezas van a entrar en las cintas.

La complicación en el proceso reside en que no se sabe cuántas piezas van a entrar en cada tanda, con lo que no se puede resolver con un simple contador. Es por ello, un diseño más óptimo y más complejo que uno en el que siempre se le proporcione la misma cantidad de piezas por tanda. Para la resolución de este problema, se ha recurrido a un bloque de función disponible en Unity Pro, el FIFO (First In First Out).

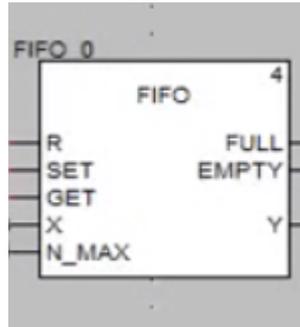


Figura 29. Bloque de función FIFO

Este bloque almacena la marca que se le va poniendo a cada pieza y cuando esa pieza llega a la cinta de descarga, saca el primer dato que tenía almacenado y que todavía no hubiera sacado. Un esquema de funcionamiento se muestra en la *Figura 30*.

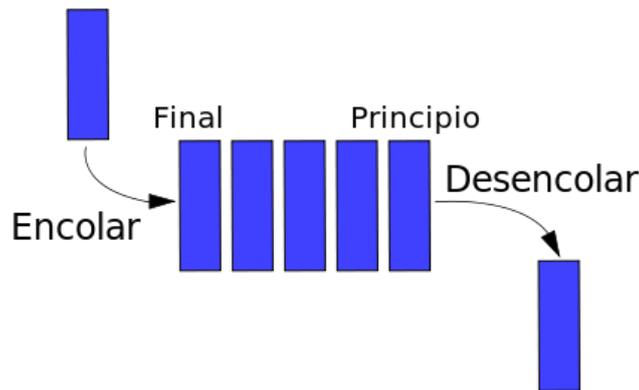


Figura 30. Diagrama de funcionamiento de la cola FIFO

7.6 Comunicación entre autómatas

Tanto el PLC que controla la línea indexada como el que controla el brazo robótico han de estar en comunicación continua (ver *Figura 27*). Existen datos de importancia como los sensores de carga y descarga que se deben conocer en ambos autómatas para el correcto funcionamiento del conjunto. Cabe destacar que la comunicación es una de las funciones más importantes en el diseño de este proyecto, dado que una desincronización no sería aceptable en absoluto. Por lo tanto, ambos PLCs deben trabajar juntos y ello se consigue enviando las variables de interés entre los dos autómatas.

Se ha elegido como medio de comunicación el acceso a memoria compartida mediante “Ethernet”, en el que se han direccionado las variables internas de interés de cada sistema (del robot y de las cintas) a puestos de memoria que no estaban asignados ni a sensores, ni a acciones, ni otras posibles variables internas de los autómatas. Se ha considerado que el brazo robótico es el “Maestro” en la comunicación entre los autómatas, ya que es el encargado de establecer los modos de marcha y las emergencias, mientras que las cintas representan el rol de “Esclavo”.

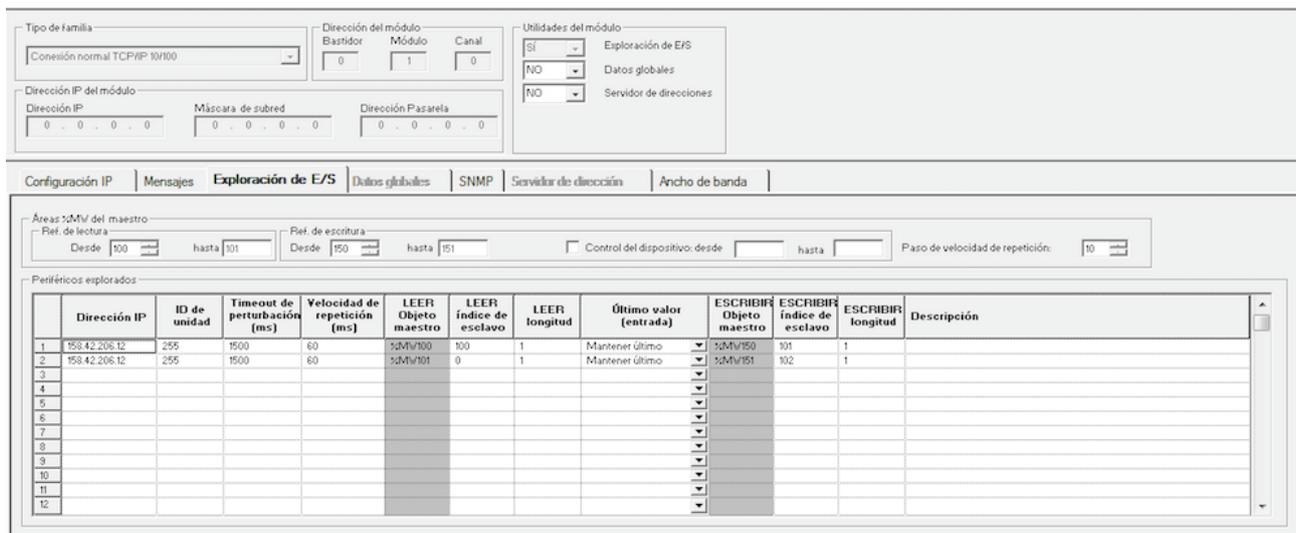


Figura 31. Configuración de la comunicación entre autómatas. Ventana Ethernet

En el manual de programador 2.2.11 Comunicación se detalla cada una de las variables compartidas entre autómatas.

7.7 Aplicación en LabVIEW para la adquisición de datos

7.7.1 Preparación:

Como se ha mencionado anteriormente, lo primero que se debe hacer es modelar el comportamiento de la célula Peltier de forma que se pueda “predecir” matemáticamente. Se conectará la tarjeta de adquisición de datos (de ahora en adelante TAD) según la Figura 32.

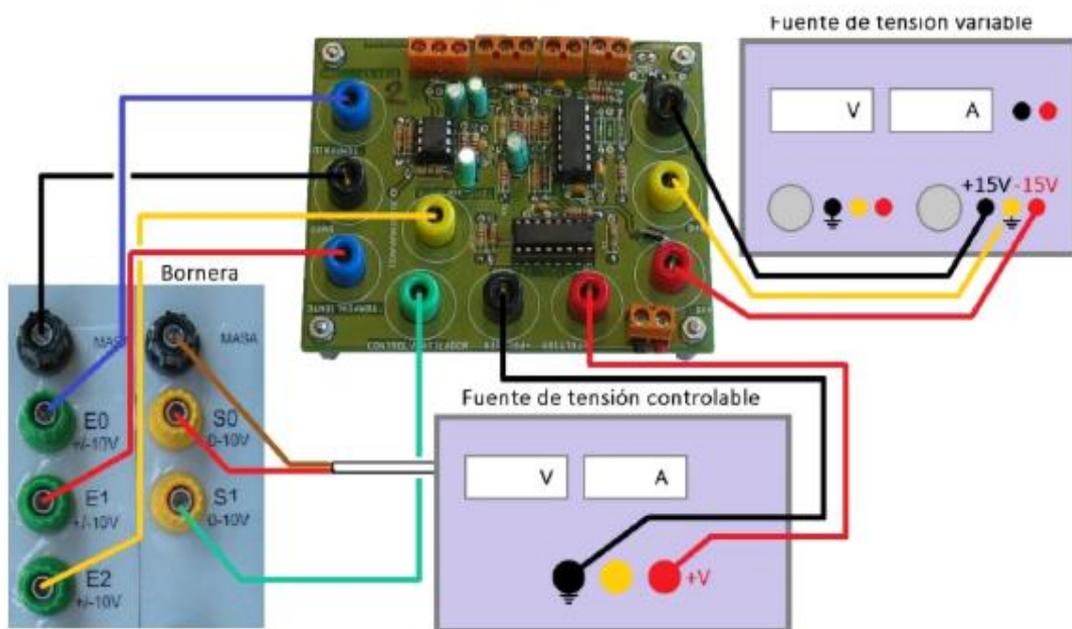


Figura 32. Esquema del montaje de la célula Peltier

7.7.2 Toma de datos:

Se ha diseñado en LabVIEW⁹ la siguiente aplicación que permite leer y mostrar por pantalla a tiempo real los datos que se están tomando de los sensores y las acciones que se están aplicando. También permite cambiar variables como la referencia, la acción del ventilador y ver cómo se comporta el sistema. Todos estos datos se almacenan en un archivo que es posible abrir con Matlab para su procesamiento.

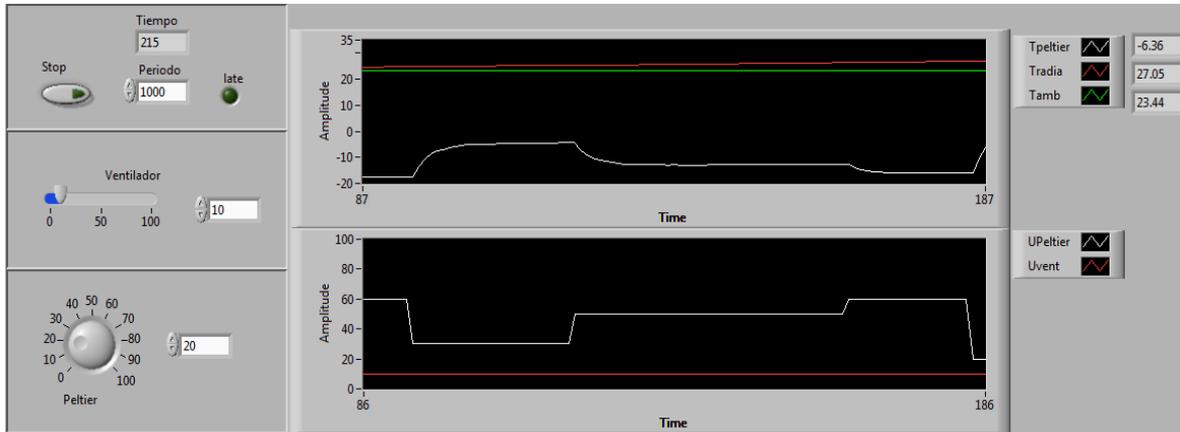


Figura 33. Panel frontal de la aplicación para la toma de datos

7.7.3 Identificación del modelo

El modelo de los que se ha calculado que mejor cumple las especificaciones de diseño es el que a continuación se muestra:

$$G(s) = \frac{-0.56658}{3.6 \cdot s + 1}$$

El proceso seguido para la obtención y validación del modelo se detalla en el anexo de diseño 2 *Identificación del modelo*.

7.7.4 Diseño del regulador

El regulador que cumple mejor las especificaciones de diseño y por el que finalmente se ha optado es el que a continuación se muestra:

$$G(r) = \frac{-2.29 \cdot (1 + \frac{1}{2.7})}{s}$$

Se trata de un regulador con acción proporcional y acción integral (PI).

El proceso seguido para la obtención y validación del regulador se detalla en el anexo de diseño en los 3 *Cálculo del regulador* y 4 *Validación del regulador*.

7.7.5 Validación del regulador:

Para la validación del regulador se utilizará una herramienta de Matlab descrita con anterioridad, Simulink.

⁹ Para el diseño se ha partido de la plantilla "identplantilla.vi". Autor: Juan Manuel Herrero Durá.

Se incorporará el modelo calculado y el regulador como se explica en el anexo de diseño. Incorporando también un patrón de entradas, se verá si el comportamiento se ajusta adecuadamente. Como se puede apreciar en la *Figura 34*, el sistema de control consigue que el regulador se adapte a la referencia lo suficientemente rápido y sin presentar oscilaciones (comportamiento sobreamortiguado).

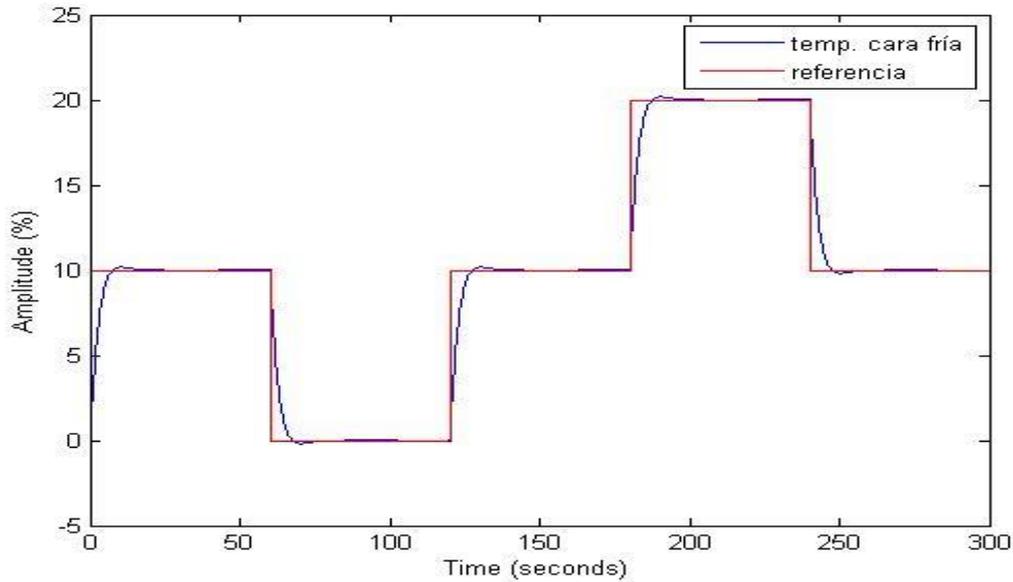


Figura 34. Respuesta simulada del regulador calculado

7.8 Implementación del regulador en Unity Pro

7.8.1 Diseño del bloque de función PI_B¹⁰:

El aspecto del bloque ya implementado será el siguiente:

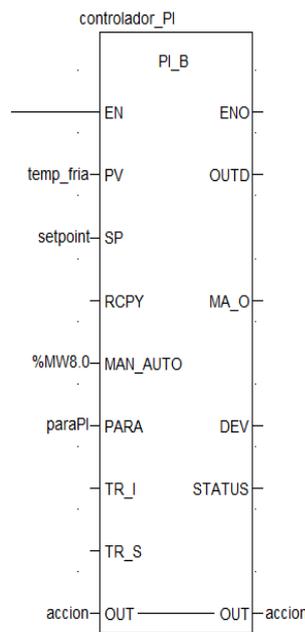


Figura 35. Bloque de función PI_B

¹⁰[13](Simarro, Anexos asignatura Lac. ETSII, 2014-2015)

En el manual de programador 2.2.10 *Implementación del PID* (Plano 8): se detallarán cada uno de los parámetros.

7.8.2 Validación del comportamiento real/simulado:

Resta una última comprobación que se realizará nuevamente mediante la herramienta Simulink. Esta comprobación se realizará como una comparación entre el comportamiento simulado del regulador y el comportamiento real.

De esta forma se podrá rechazar o aceptar el regulador viendo si el comportamiento es el mismo que se esperaba de la primera simulación.

Los resultados se aprecian en la *Figura 36*.

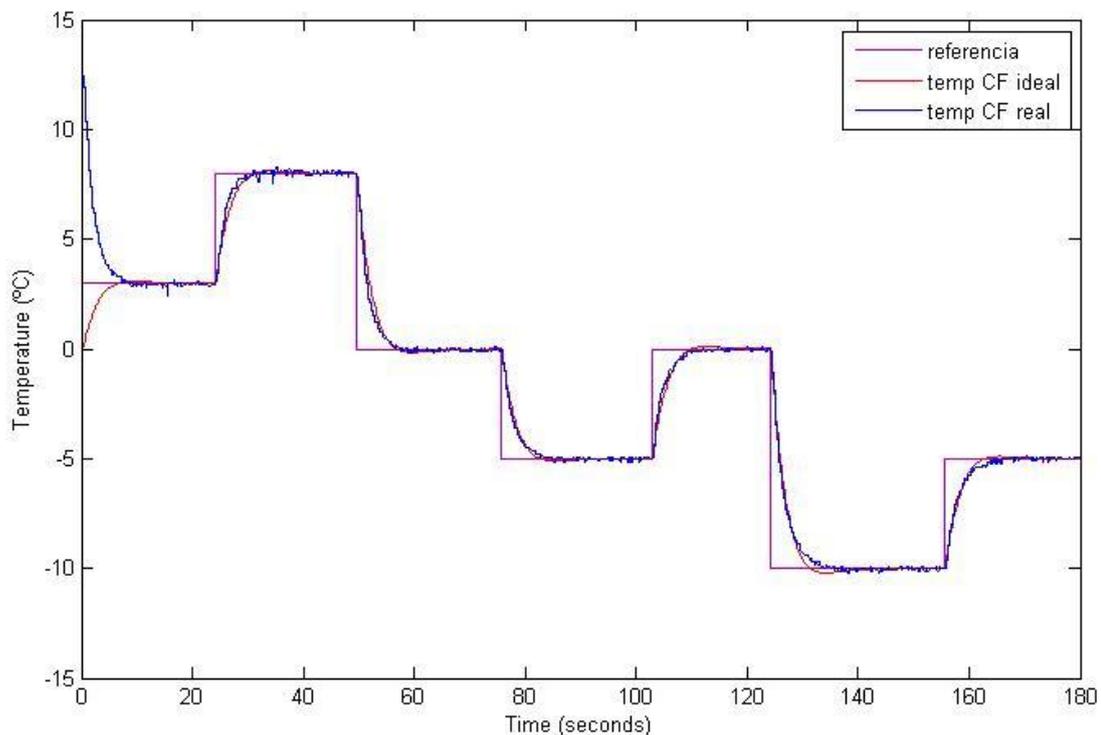


Figura 36. Comparativa entre la respuesta simulada (roja) y la real (azul) del regulador

En rojo se aprecia la salida simulada y en azul la salida real.

Como se observa, ambas señales transcurren casi idénticas por lo que se puede concluir que el comportamiento real será muy similar al comportamiento simulado.

En el anexo de diseño 5.2 *Validación del sistema de control* se detalla cómo se ha obtenido realizado y analizado dicho ensayo.

7.9 Sistema de usuarios:

7.9.1 Creación dominio jerárquico de usuarios e implementación¹¹

Se ha decidido crear un sistema de usuarios mediante el cual gestionar los permisos de seguridad. Mediante este sistema los usuarios podrán:

¹¹ Para ello se han seguido las indicaciones proporcionadas por [5] (Herrero, Apuntes de la asignatura LAC, 2014-2015)

- Identificarse con diferentes permisos
- Comprobar que la conexión es correcta
- Desconectarse una vez finalizada la sesión o al cerrar el programa
- Saber en todo momento el usuario actualmente conectado

Se ha procurado una interfaz sencilla pero funcional a la que poder acceder desde todas las pestañas. Los elementos principales de la interfaz de usuario se muestran en la siguiente *Figura 37*.

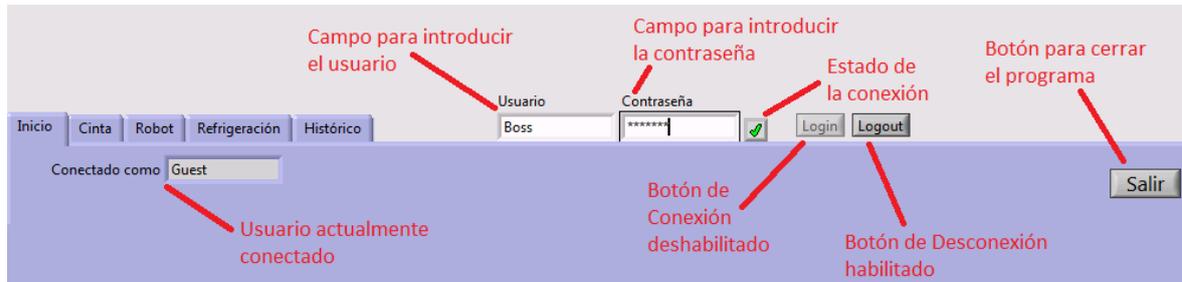


Figura 37. Imagen de las pestañas y los elementos de control de usuario

Se crearán en el dominio 4 usuarios además del usuario “Administrador” que se crea por defecto y desde el cual se pueden gestionar el resto de cuentas. Los usuarios creados se muestran en la *Figura 38*.

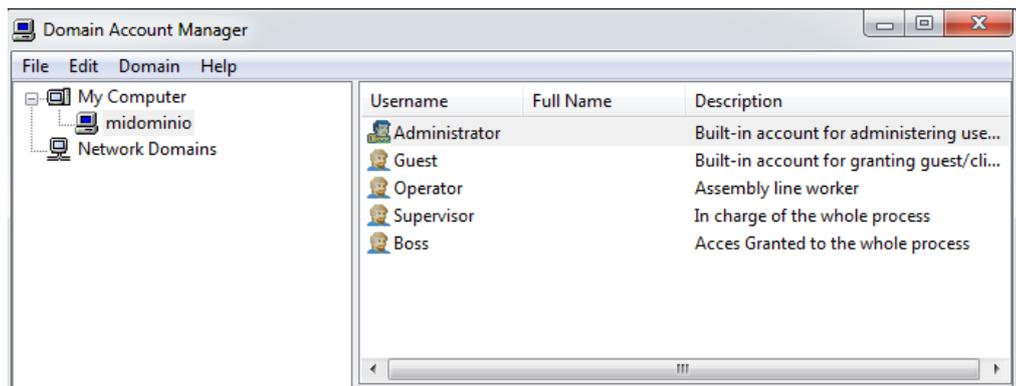


Figura 38. Ventana de gestión del dominio

- **Guest:** No tendrá ningún permiso asignado por defecto, pero el administrador puede asignarle los permisos que considere. Con esta cuenta se puede organizar una presentación a medida para un cliente.
- **Operator (operario):** Tendrá acceso a la pestaña de Inicio, a la de Cintas, a la de Brazo y a la de Refrigeración, al modo automático de la Peltier, así como a los botones de puesta en marcha y detención del panel lateral. También podrá decidir entre el modo de marcha automático o el ciclo a ciclo.
- **Supervisor (Operario avanzado):** Tendrá acceso a las mismas pantallas y permisos del operario pero incorporaría también la gestión manual de la refrigeración y se le habilitaría el modo de marcha paso a paso y los botones de manejo manual de todos los elementos de la línea indexada y del brazo robótico.
- **Boss (Jefe Ingeniero):** Tendrá acceso habilitado a todas las pestañas, además es el administrador del sistema por lo que podrá agregar o borrar a otros usuarios. Es el único usuario con acceso a la pestaña del archivo histórico.

Una muestra de los permisos asignados se puede apreciar en la *Figura 39*.

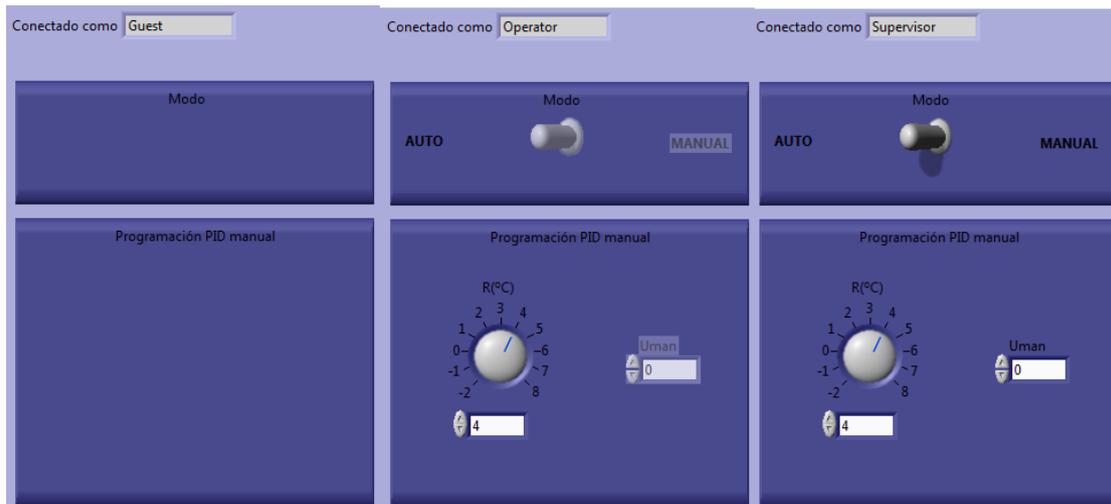


Figura 39. Diferentes permisos de usuario

7.10 Publicación de datos:

Uno de los requisitos para que se pueda controlar el sistema desde una pantalla remota es que haya una conexión entre el autómatas y el programa que monitoriza los datos.

Esto se realiza mediante una conexión por servidor OPC¹².

Se ha definido pues un aparato (device) que sería el autómatas TSX Premium con el que se manejan tanto las cintas como el brazo robótico.

Dentro de ese tipo de aparato se han direccionado los dos puertos con los que se trabajará bajo el nombre de CINTAS y ROBOT. Dentro de cada uno se publican las variables de memoria que interesan para que otros programas puedan acceder. La pantalla del servidor queda como se muestra en la *Figura 40*.

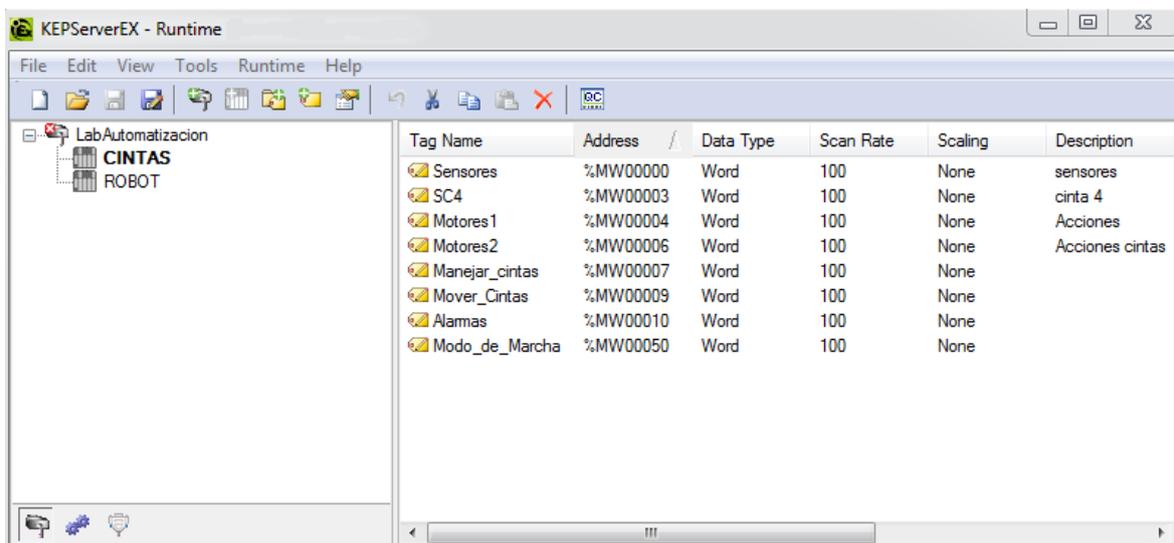


Figura 40. Ventana del servidor.

¹² Para la creación del servidor OPC se han seguido las indicaciones incluidas en los documentos:

[14] (Simarro, Apuntes de la asignatura LAC. ETSII, 2014-2015)

[13] (Simarro, Anexos asignatura Lac. ETSII, 2014-2015)

7.11 Incorporación al proyecto de las Bound Variables¹³

Una vez publicados los datos mediante el servidor, habrá que acceder a ellos desde LabVIEW. Esto es preciso para posteriormente poder manejar los datos desde el programa de la pantalla de usuario, de forma que cualquier botón que se pulse en la pantalla origine un cambio en la variable del autómeta.

La *Figura 41* muestra el aspecto de la librería creada junto con las variables que se manipularán en el diagrama de bloques.

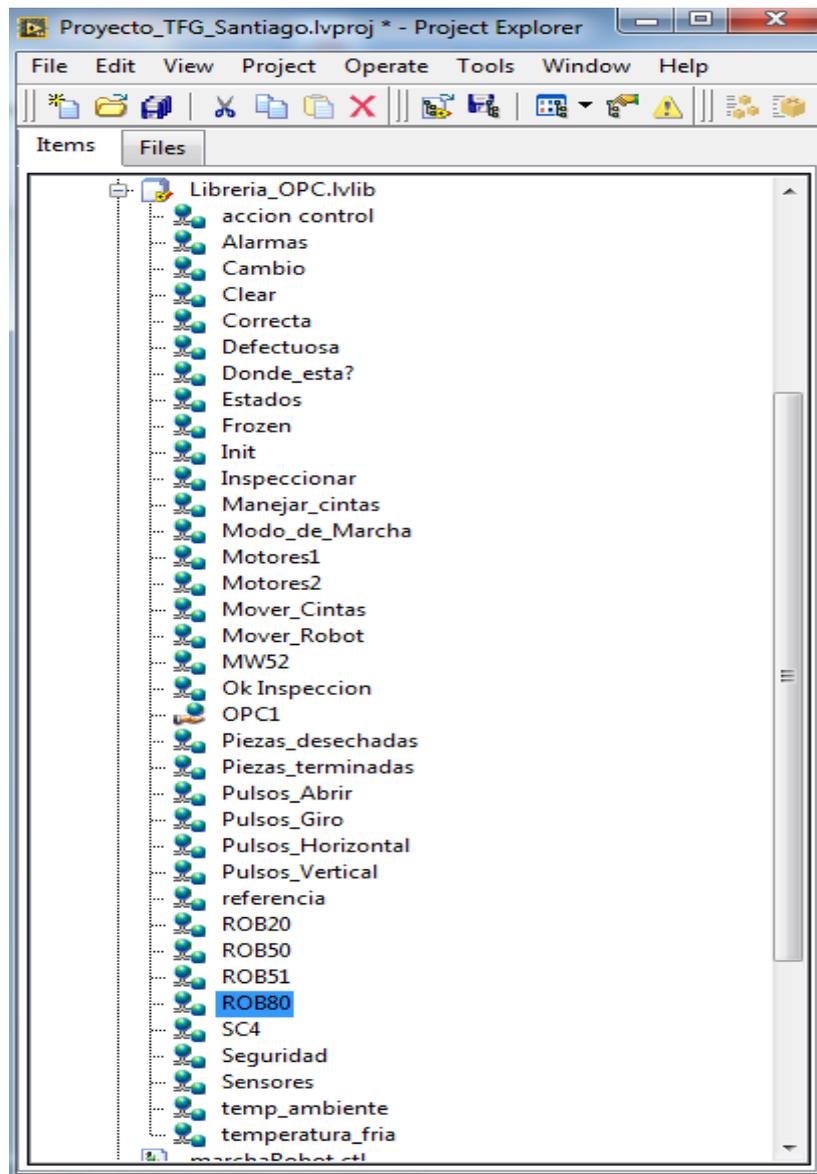


Figura 41. Bound Variables del proyecto

¹³En el documento [13] (Simarro, Anexos asignatura Lac. ETSII, 2014-2015) se explica detalladamente cómo crear las librerías necesarias y acceder a los datos del servidor

7.12 Monitorización y control remoto

Uno de los puntos más importantes de la creación del proyecto es la posibilidad de controlarlo desde una aplicación conectada a la red de los autómatas.

La posibilidad de una Interfaz humano-máquina (HMI) permitirá a los diferentes usuarios interactuar con el proceso de producción bien en la línea de producción mismo (pantalla de operario), bien desde un despacho observando a tiempo real cómo está avanzando el proceso y otras particularidades. La posibilidad de conectarse desde un computador remoto no situado inmediatamente junto al proceso abre las puertas a un nuevo modo de gestión y control con un importante añadido de funcionalidad. Es un diseño más óptimo.

La recepción de señales es tan importante como la manipulación de salidas, por lo que en esta interfaz se producirá un intercambio en ambos sentidos de la información.

Para la mayor facilidad en el manejo de la pantalla de usuario se ha optado por distribuir en pestañas cada elemento. Además, se mostrarán siempre en un lateral diversos controles como los de alarma o los modos de marcha de forma que se puedan manipular en todo momento. A continuación se mostrarán las diversas pestañas con una breve explicación sobre su función.

7.12.1 Pestaña lateral

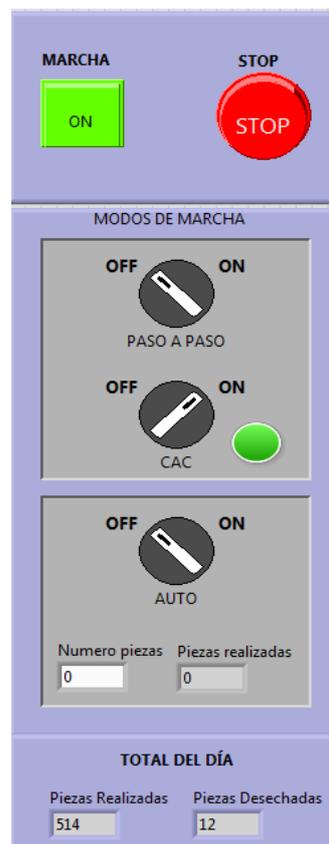


Figura 42. Pestaña lateral.

En la parte superior se distinguen dos botones para la puesta en funcionamiento y la parada de todo el proceso.

El cuadro de modos de marcha incluye botones para la conexión o desconexión de los mismos así como indicadores de la realización de piezas en el día que ocupa.

Este cuadro permanecerá visible en todo momento en el lateral.

7.12.2 Pestaña Inicio

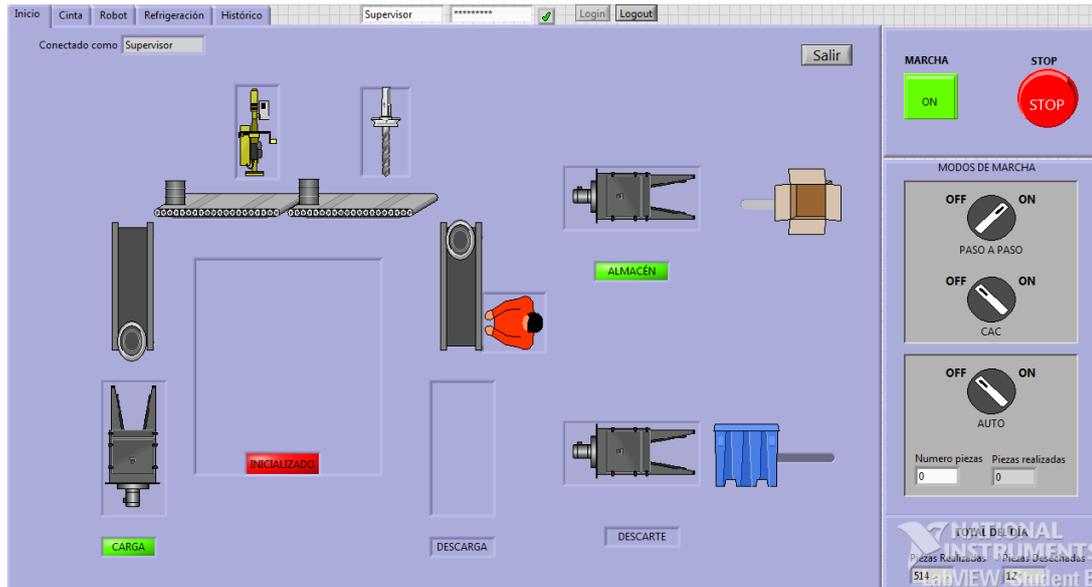


Figura 43. Pestaña Inicio.

Será la pantalla que se mostrará durante el funcionamiento normal junto a los modos de marcha y alarma a un lateral, como ya se ha señalado anteriormente. Desde esta pantalla se podrá observar donde se encuentra el proceso en todo momento y no se podrá manipular el mismo. (Se muestran varios elementos encendidos como apoyo visual que durante el funcionamiento no coincidirán encendidos simultáneamente).

7.12.3 Pestaña Cintas

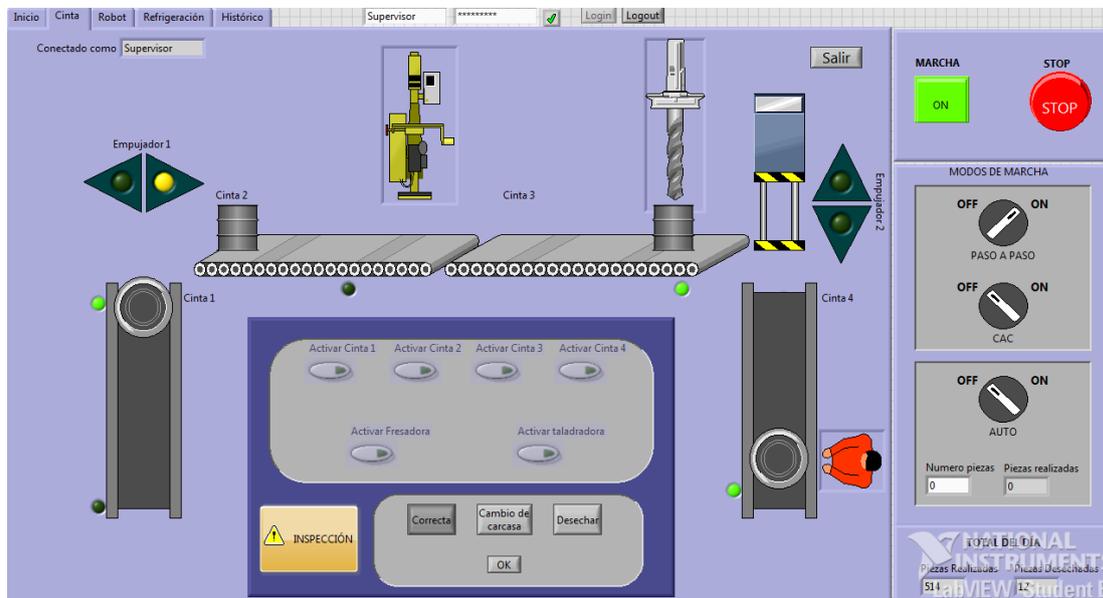


Figura 44. Pestaña Cintas.

Desde esta pantalla se podrán observar más detenidamente algunos elementos de las cintas y su acción actual como los sensores de las cintas o el movimiento que realiza el empujador. También se podrá manejar el modo de inspección así como algunos elementos de las cintas individualmente si se tienen los permisos necesarios.

7.12.4 Pestaña Robot

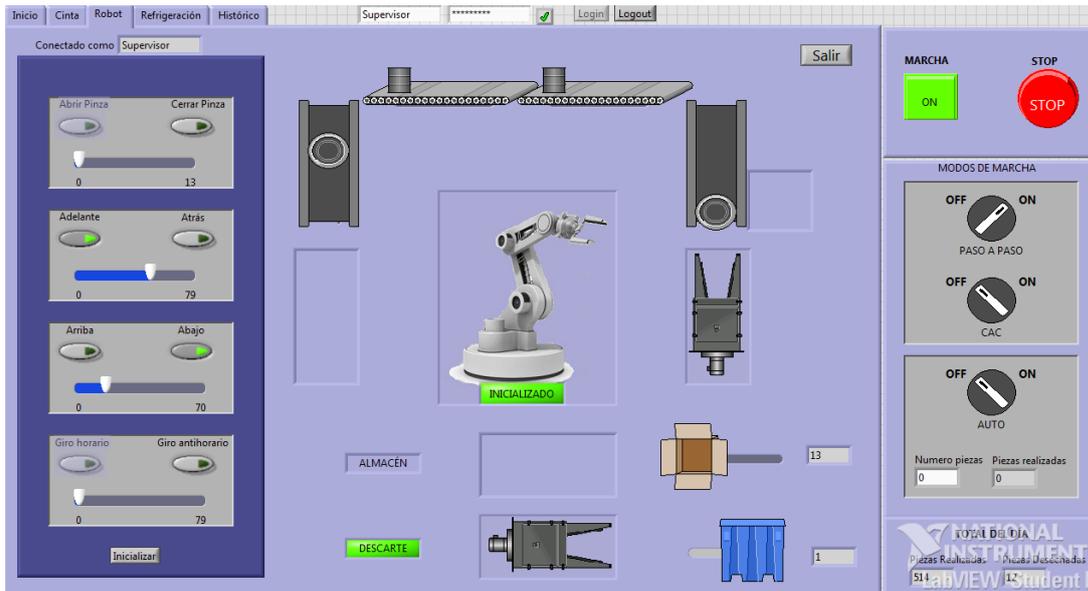


Figura 45. Pestaña Robot.

Esta pantalla permite, de manera análoga a la de las cintas, visualizar las posiciones del brazo de manera más detallada y manejarlo manualmente si se tiene los permisos necesarios. La imagen es sólo orientativa, habrá elementos que no coincidirán encendidos simultáneamente.

7.12.5 Pestaña Refrigeración

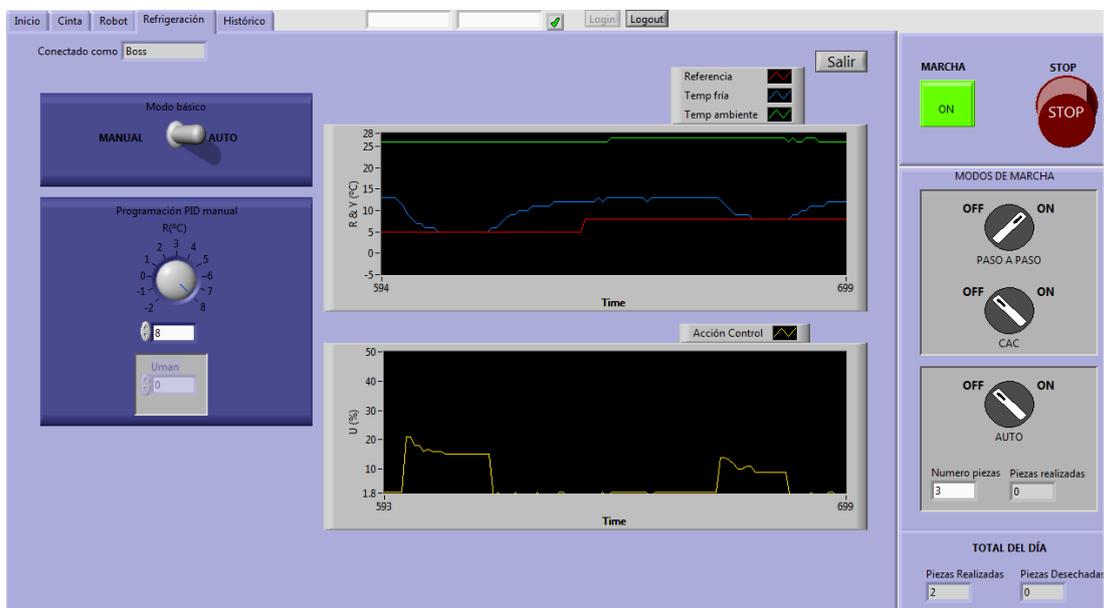


Figura 46. Pestaña Refrigeración.

Desde esta pantalla se podrá controlar la célula Peltier y ver gráficamente los valores de los sensores de temperatura con respecto a una referencia que se puede cambiar. También se puede observar gráficamente la acción de control que se aplica para que la cara fría de la célula Peltier llegue a la referencia marcada.

7.12.6 Pestaña Histórico

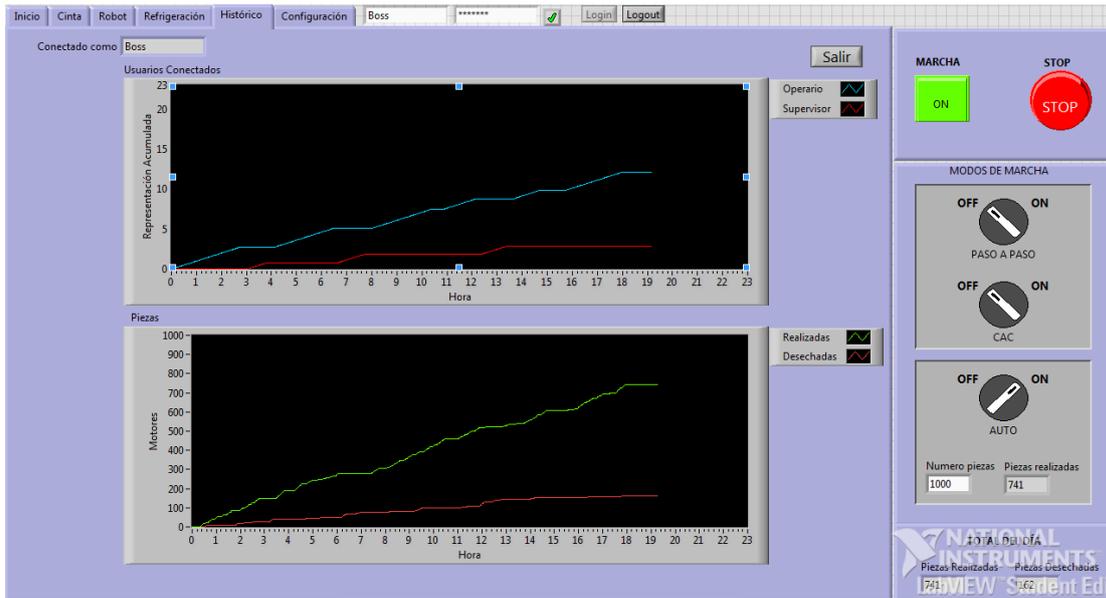


Figura 47. Pestaña Histórico.

En esta pantalla se muestran gráficamente el tiempo que han estado conectados los usuarios *Operario* y *Supervisor* a lo largo de la jornada. En la segunda gráfica se muestran las piezas realizadas y desechadas a lo largo de la jornada.

7.13 Almacenamiento de datos

Se almacenarán todos los datos que se consideren de interés de la célula Peltier. Se han considerado de interés las siguientes variables:

- Tiempo
- Valor de la referencia
- Temperatura de la cara fría
- Temperatura ambiente
- Acción aplicada a la célula Peltier

Time(s)	Ref	Tfría	Tamb	acción
56.400	4.000	12.000	26.000	22.000
56.600	4.000	12.000	26.000	22.000
56.800	4.000	10.000	26.000	21.000
57.000	4.000	10.000	26.000	21.000
57.200	4.000	9.000	26.000	21.000
57.400	4.000	9.000	26.000	21.000
57.600	4.000	9.000	26.000	21.000
57.800	4.000	8.000	26.000	20.000
58.000	4.000	8.000	26.000	20.000
58.200	4.000	7.000	26.000	19.000
58.400	4.000	7.000	26.000	19.000
58.600	4.000	7.000	26.000	19.000
58.800	4.000	6.000	26.000	18.000

Figura 48. Archivo de datos.

En la *Figura 48* se muestra el aspecto que tendrá el archivo de texto una vez finalizado el programa y almacenados los datos. Este archivo se podrá cargar en Matlab como se explica en el anexo de diseño 2 *Identificación del modelo*.

8 Conclusiones

El presente trabajo cumple la integridad de las especificaciones planteadas a su inicio, descritas en el punto 4 *Especificaciones*. En efecto, los objetivos previstos fueron los señalados para Cintas, Robot, Célula Peltier y pantalla SCADA. Su cumplimiento permite extraer las siguientes conclusiones:

1. Se ha procurado un sistema SCADA que resulte amigable de forma que el operario cometa el mínimo número de errores posibles y ello se ha conseguido desarrollando las siguientes funciones:
 - Pantalla distribuida en pestañas temáticas (Inicio, Cintas, etc.).
 - Modos de marcha y alarma permanentemente accesibles a un lateral de la pantalla.
 - Desactivación automática de modo de marcha obsoleto tras actualizar el nuevo.
 - Impedimento de forzar las máquinas fuera de su rango de trabajo mediante la desactivación de los botones pertinentes de forma automática cuando se alcanza dicho límite.
 - Asignación de permisos, para que los usuarios menos cualificados no puedan acceder a funciones críticas (que supongan un riesgo para el equipo tanto material como personal).
 - Interacción usuario-programa mediante una petición de confirmación en situaciones delicadas por medio de ventanas emergentes.
2. Se ha empleado un hardware industrial como los autómatas TSX Premium. Se trata de un sistema robusto y flexible que permite manejar procesos como las maquetas FisherTechnik que se han empleado en el presente proyecto. Estos autómatas son capaces de reproducir el proceso repetitivamente con un número muy reducido de situaciones de fallo.
3. Tras los conocimientos adquiridos en materia de comunicación entre autómatas se puede implementar un sistema supervisor en el cual ambos autómatas estén permanentemente chequeando la actuación del homólogo y, en caso de que no se reciba respuesta o la respuesta no sea la esperada, den las señales de alarma pertinentes.
4. El control implementado para la célula Peltier cumple la totalidad de las especificaciones de diseño que se establecieron y es capaz de adecuarse a la referencia en todo el intervalo efectivo (0-10°C) y rechazar perturbaciones.
5. Se han diseñado aplicaciones con software profesional (LabVIEW y Unity Pro). Para ello se ha precisado de los conocimientos adquiridos en asignaturas de automática e informática ampliados con la investigación en el campo de la programación. La variedad de lenguajes de programación que se emplean en el presente proyecto (C, G, M, Grafset...) favorece un resultado óptimo que se resuelve en:
 - Pantalla de operario con elementos visualmente agradables (imágenes) e intuitiva.
 - Automatización de las maquetas con un lenguaje normalizado (Grafset) con una estructura jerarquizada que permite la correcta gestión de situaciones de alarma.
6. Se ha conseguido de manera satisfactoria programar las dos maquetas (línea indexada y brazo robótico) de manera que ambas realicen:
 - Desplazamiento de las piezas por un recorrido previamente diseñado. En el presente proyecto se ha logrado que ese recorrido se asemeje al máximo posible al realizado durante el proceso de fabricación de los motores de combustión interna (MCI) en plantas industriales.
 - Colocación de las piezas en diferentes puntos de control atendiendo a un criterio de selección de tipo de pieza cedido al usuario. Mediante un sistema de inspección

intuitivo es posible definir una pieza como correcta, reparable o desechable y que las maquetas actúen en consecuencia.

- Funcionamiento diferente dependiendo del modo de marcha que se seleccione. Se ha conseguido que las maquetas funcionen ininterrumpidamente durante el modo automático, una única vez durante el modo ciclo a ciclo y cedan el control de todos los elementos al usuario durante el modo paso a paso.
 - Bloqueo de las acciones tras la activación de un modo de alarma. Cualquier alarma que active el usuario garantiza que la acción que realizan las maquetas en ese momento queda interrumpida, lo que supone un factor de seguridad importante. Tras percibir una situación de riesgo, cualquier usuario puede detener el proceso instantáneamente y sólo pulsando un botón e impedir daños considerables.
7. El control de usuarios diseñado permite manejar diferentes permisos así como restringir los accesos no deseados. Supeditando el inicio de la pantalla de operario a una identificación inicial, se impide que personas ajenas al proceso puedan manipularlo y producir daños.
 8. El diseño de una aplicación que ejecuta las órdenes de manera independiente permite aligerar el peso del programa y la velocidad de ejecución. En la pantalla SCADA, cada bucle se trata por separado con lo que se consigue que el conjunto realice una función compleja, pero cada bucle realice una función simple, lo que da lugar a un considerable menor número de errores. Además, los errores que pueden surgir son fácilmente subsanables y localizables. Realizar el programa con esta estructuración me ha permitido aprender a descomponer los problemas complejos en problemas más simples y asumibles.

En resumen, se trata de un proyecto en el que se aprende a aplicar los conocimientos adquiridos durante la carrera a un entorno industrial donde se trabaja con hardware y software profesional, equipo robótico avanzado (maquetas FisherTechnik), mecanismos de control híbrido (célula Peltier) y aplicaciones diseñadas para la adquisición de datos e interacción con el usuario (SCADA).

Se trata de un proyecto práctico a la vez que interesante y los medios visuales de las maquetas y pantalla facilitan la comprensión y alcance de lo que se está trabajando en el proyecto.

9 Ampliaciones

El presente proyecto está cerrado pero durante el mismo han surgido diversas ideas por las limitaciones del equipo (memoria del autómatas, rigidez del diseño de las maquetas, etc.) o bien que debido a limitaciones de tiempo y objeto del proyecto, no se han podido llevar a cabo. A continuación se enumeran varias de estas ideas:

- **Sistemas de autogestión de alarmas:** Sistemas optimizados que sean capaces de retirar una pieza de la cola de producción en cualquier posición que se encuentre y continuar el proceso de manera autónoma. Esto se podría conseguir añadiendo un nuevo automatismo al diseño como por ejemplo un robot que se activara cada vez que se detecte un defecto y se le mandara a la posición donde se encuentra la pieza a retirarla. Otra alternativa sería incorporar a la maqueta medios audiovisuales, sensores led o sonidos que se activaran en el momento en que algún fallo fuera detectado. De esta forma se optimizaría todavía más el diseño que se ha planteado y se conseguiría hacer más fácil el trabajo a un operario que es lo que se ha planteado desde el principio como premisa.

- **Sistemas de trazabilidad:** Mecanismos por los cuales quedarían registrados de cada pieza cuándo se hizo y quién la realizó. Se trata de un sistema muy extendido en el mundo industrial para la óptima detección de productos defectuosos y depuración de responsabilidades. La forma más sencilla de realizar dicho proceso consistiría en realizar una firma, bien sea física o electrónica a cada pieza. Cada pieza tendría asociado un código de forma que se podría localizar el momento exacto de su producción así como quién la hizo y otros datos de interés como el proveedor que gestionaba en ese momento el material con el que se hace, etc.
- **Posicionamiento de la pieza en los tres ejes:** Para aplicaciones industriales de complejidad elevada no basta con un autocentraje cónico como el que hay instalado en las cintas. Muchas veces se trabaja con material frágil o con unas dimensiones y peso considerables que hay que manejar con extremada precaución, como en el caso de los motores que se propone a modo de ejemplo en el presente trabajo. Dichas piezas deben ser posicionadas al milímetro para que las operaciones que sobre ellas se realizan produzcan los resultados deseados. El arranque de viruta es un proceso peligroso y complejo y hasta el más mínimo error en el posicionamiento de la herramienta podría suponer un cambio en el ángulo de ataque, lo que supondría que la viruta podría crecer en direcciones no deseadas. Esto sería gravísimo y podría causar lesiones serias o importantes daños en los equipos.
Para realizar el centraje se debería disponer de sensores y tecnología más sofisticados.
- **Control avanzado de las unidades de mecanizado:** En el presente proyecto se simula la actuación de la fresadora y la taladradora como un giro del elemento de la maqueta correspondiente.
Esto en la realidad supone una complejidad mucho mayor, en la que hay que definir los diámetros del cabezal de cada herramienta de la fresadora, los tiempos que actúan así como el cambio de cabezal, la duración de las plaquitas y también la posición exacta en los tres ejes en los que ha de fresar en cada momento.
Sería un interesante proyecto unir los conocimientos de automatización adquiridos a los que ya se adquirieron en asignaturas como ciencia de materiales o Sistemas de Producción y Fabricación y desarrollar una maqueta que cumpliera con las especificaciones tanto del área de la ciencia de los materiales como del área de la automática.
- **Depuración del código:** Inicialmente se desarrollaron las aplicaciones partiendo de unos conocimientos básicos de cada programa y a medida que se ha avanzado, dichos conocimientos han aumentado considerablemente. A medida que los conocimientos del programa se iban incrementando se lograba optimizar el código simplificando y comprimiendo cada vez más la programación y por tanto los espacios de memoria utilizados, lo que agiliza el proceso cada vez más. Una revisión exhaustiva del proyecto y las líneas de código permitiría la detección de comandos repetitivos y/u obsoletos. Muchos de estos problemas o procesos no óptimos están localizados pero debido a que se trata de código introducido durante los primeros pasos, los cambios supondrían una reorganización del proyecto.
Se concluye pues que el nivel de experiencia del programador es un factor muy importante a la hora de diseñar un programa.

10 Bibliografía

- [1] Bondia, J., & Sala, A. (2013-2014). *Comportamiento dinámico de sistemas*. Valencia: SPUPV.
- [2] Chacón, H. Q. (2014). *SCADA en LabVIEW - Animacion de una faja transportadora*. Obtenido de <https://www.youtube.com/watch?v=Ay2182-Agp8>
- [3] distancia, U. N. (s.f.). *Lenguaje de programación ST*. Obtenido de http://datateca.unad.edu.co/contenidos/2150512/ContenidoLinea/leccin_1633__texto_e_structurado_structured_text__st.html
- [4] Herrero, J. M. (2013-2014). Apuntes de la asignatura TAU. *Prácticas con el motor de corriente continua*. Valencia: UPV.
- [5] Herrero, J. M. (2014-2015). Apuntes de la asignatura LAC. *Ident y Control con Labview.pdf y Extras.pdf*. Valencia: UPV.
- [6] Herrero, J. M. (s.f.). *Tutorial Simulink, Identificación y validación de Fdts*. Recuperado el 2015, de <http://politube.upv.es/play.php?vid=54901>
- [7] Instruments, N. (2010). *Getting started with LabVIEW*. Obtenido de <http://ni.com>
- [8] Instruments, N. (2012). *Getting started with the LabVIEW Datalogging and Supervisory Control Module*. Obtenido de <http://ni.com>
- [9] Liebherr. (s.f.). *Auszüge aus der Innerbetrieblichen Schweißrichtlinie ISR B – 010 mit praktischen Ergänzungen im Alltag y Abgasgrenzwerte für Dieselmotoren und ihre Auswirkungen auf LWE*. Ehingen, Alemania.
- [10] Morales, L. M. (2014-2015). PFC. ETSII. *Desarrollo de una aplicación de control de temperatura de una célula Peltier mediante LabVIEW*. Valencia: UPV.
- [11] Rodríguez, F. B. (s.f.). *Universidad Nacional de Colombia*. Recuperado el 2015, de <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100002/lecciones/taxonomia/fabricacionflexibel.htm>
- [12] Salcedo, J. V. (2013-2014). Apuntes de la asignatura TAU. ETSII. *Graficets*. Valencia: UPV.
- [13] Simarro, R. (2014-2015). Anexos asignatura Lac. ETSII. *Anexo 00 Prácticas de TAU con el PLC TSX-Premium.pdf, Anexo 02 modbus_ethernet (KEPServerEX5).pdf y Anexo 04 NI-Tutorial-7906-es.pdf*. Valencia: UPV.
- [14] Simarro, R. (2014-2015). Apuntes de la asignatura LAC. ETSII. *Seminario Automatización de Procesos.pdf y Trabajo Automatización de Procesos.pdf*. Valencia: UPV.
- [15] *Wikipedia*. (s.f.). Obtenido de Información sobre sensores tipo termopar: http://es.wikipedia.org/wiki/Efecto_termoel%C3%A9ctrico



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº3: Anexo a la memoria

Procesos FisherTechnik (Entradas-Salidas)

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenidos

1 Anexo de E/S del brazo robótico	50
2 Anexo de E/S de las cintas	51

Índice de tablas

Tabla 1. Entradas del brazo robótico	50
Tabla 2. Salidas del brazo robótico.....	50
Tabla 4. Variables de comunicación del brazo robótico.....	51
Tabla 5. Entradas de las cintas	51
Tabla 3. Variables de alarma del brazo robótico	51
Tabla 6. Salidas de las cintas	51
Tabla 7. Variables de comunicación de las cintas	52
Tabla 8. Variables implementadas del PID en las cintas.....	52

1 Anexo de E/S del brazo robótico

Entrada	Descripción	Dirección TSX
Spinza	final de carrera referencia de la pinza	%MW0.0
Satrás	final de carrera referencia brazo agarre	%MW0.2
Sarr	final de carrera referencia vertical	%MW0.4
Sgiro	final de carrera referencia de giro	%MW0.5
PulsosP	Contador de pulsos de la pinza	%MW0.1
PulsosH	Contador de pulsos del brazo de agarre	%MW0.3
PulsosV	Pulsos encoder movimiento vertical	%MW0.7
PulsosG	Pulsos encoder movimiento giratorio	%MW3.1
SentidoV	Sentido del movimiento vertical	%MW0.6
SentidoG	Sentido del movimiento de giro	%MW3.0
Binit	Botón para inicializar el brazo robótico y las cintas	%MW10.0
BFrozen	Botón congelar el brazo robótico y las cintas	%MW10.2
Cam	Botón para señalar si hay que volver a realizar el proceso 2	%MW10.8
Def	Botón para señalar que la pieza es defectuosa	%MW10.9
OK	Botón para señalar que ya se ha realizado la inspección	%MW10.11
XP	Variable interna para saber qué proceso asignar a la siguiente pieza que entra	%MW20
YP	Variable interna para saber qué proceso fue asignado a la pieza que sale	%MW21
Auto	Botón para seleccionar el modo automático	%MW50.1
CAC	Botón para seleccionar el modo ciclo a ciclo	%MW50.2
PaP	Botón para seleccionar el modo paso a paso	%MW50.3
BCAC	Botón para que se realice un ciclo	%MW50.4
Piezas	Piezas realizadas durante el modo automático	%MW51
PiezasTer	Piezas realizadas desde el inicio de la sesión	%MW53
PiezasDes	Piezas desechadas desde el inicio de la sesión	%MW54
PiezasCom	Se activa si se completa el número de piezas seleccionado en el modo automático	%MW60
P	Pulsos de la pinza	%MW11
H	Pulsos del movimiento brazo agarre	%MW12
V	Pulsos del movimiento vertical	%MW13
G	Pulsos del movimiento giratorio	%MW14
Marcha	Botón que enciende el brazo robótico y las cintas	%MW 50.0
RecogidaP	1 si está recogiendo el brazo una pieza en la cinta de descarga	%MW150.0
DejadaP	1 si ha dejado el brazo una pieza en la cinta de carga	%MW150.1

Tabla 1. Entradas del brazo robótico

Salida	Descripción	Dirección TSX
AP	Motor de apertura de pinza	%MW4.2
CP	Motor de cierre de pinza	%MW4.3
AdH	Motor del brazo de la pinza hacia adelante	%MW4.4
AtH	Motor del brazo de la pinza hacia atrás	%MW4.5
AbV	Motor movimiento vertical hacia abajo	%MW4.6
ArrV	Motor movimiento vertical hacia arriba	%MW4.7
HG	Motor movimiento giratorio horario	%MW6.2
AhG	Motor movimiento giratorio antihorario	%MW6.3

Tabla 2. Salidas del brazo robótico

Alarmas	Descripción	Dirección TSX
Inicializar	Reinicia los graficets.	%MW10.0
Congelar	Para los graficets en la etapa que estén y detiene las acciones.	%MW10.2
Inspección	Botón para señalar si se quiere hacer una inspección	%MW10.10

Tabla 3. Variables de alarma del brazo robótico

Comunicación	Descripción	Dirección TSX
ICC	Indicador de pieza en el sensor de la cinta de carga	%MW101.6
IEmp	Indicador de pieza en el sensor del empujador	%MW101.4
IDispCC	Indicador de disponibilidad de la cinta de carga. 1 mientras está disponible	%MW100.0
IHayPCD	Indicador de pieza en cinta de descarga. 1 mientras hay pieza	%MW100.1

Tabla 4. Variables de comunicación del brazo robótico

2 Anexo de E/S de las cintas

Entrada	Descripción	Dirección TSX
SE1Del	Sensor de final de carrera frontal del empujador 1	%MW0.0
SE1Atrás	Sensor de final de carrera trasera del empujador 1	%MW0.1
SE2Del	Sensor de final de carrera frontal del empujador 2	%MW0.2
SE2Atrás	Sensor de final de carrera trasera del empujador 2	%MW0.3
SE	Fototransistor empujador 1	%MW0.4
Sfresadora	Fototransistor fresadora	%MW0.5
SCC	Fototransistor de la cinta de carga	%MW0.6
STaladradora	Fototransistor de la taladradora	%MW0.7
SCD	Fototransistor de la cinta de descarga	%MW3.0
Proc	Variable interna para saber qué proceso fue asignado a la pieza que sale	%MW20.1
DispCC	Variable interna para saber si la cinta de carga tiene pieza(0) o está disponible(1)	%MW100.0
HayPCD	Variable interna que vale 1 cuando hay una pieza esperando a ser recogida en la cinta de descarga	%MW100.1

Tabla 5. Entradas de las cintas

Salida	Descripción	Dirección TSX
ME1Del	Motor del empujador 1 hacia adelante	%MW4.2
ME1Atrás	Motor del empujador 1 hacia atrás	%MW4.3
ME2Del	Motor del empujador 2 hacia adelante	%MW4.4
ME2Atrás	Motor del empujador 2 hacia atrás	%MW4.5
C1	Motor de la cinta transportadora de carga	%MW4.6
C2	Motor de la cinta transportadora de la fresadora	%MW4.7
C3	Motor de la cinta transportador de la taladradora	%MW6.3
C4	Motor de la cinta transportadora de salida/descarga	%MW6.5
Fresadora	Motor de la fresadora	%MW6.2
Taladradora	Motor de la taladradora	%MW6.4
Luces	Habilitar sensores y empujadores	%MW6.6

Tabla 6. Salidas de las cintas

Comunicación	Descripción	Dirección TSX
Marcha_Cintas	Indicador de la variable Marcha del brazo robótico	%MW101.2
IPaP	Indicador de la variable PaP del brazo robótico	%MW101.3
IXP	Indicador del proceso a aplicar a la siguiente pieza que entre. De la variable XP del brazo	%MW102.0
lcambio	Indicador de la variable cambio del brazo robótico	%MW102.1
IDejadaP	Indicador de si la pieza está siendo dejada en la cinta de carga	%MW101.0
IRecogidaPCD	Indicador de si la pieza está siendo recogida de la cinta de descarga	%MW101.1

Tabla 7. Variables de comunicación de las cintas

PID	Descripción	Dirección TSX
SCaraFría	Sensor de la célula Peltier de la temperatura de la cara fría [mV]	%MW1
SCaraFría2	SCaraFría/77	%MW29
SCaraFría3	SCaraFría/77 – 60	%MW30
Temp_Fría	Real [SCaraFría3] [°C]	
A/M	Botón para seleccionar entre el modo automático(PID) o el modo manual(Usuario)	%MW8.0
Actúa_fresadora	Variable interna booleana que vale 1 cuando la fresadora está en uso	%MW8.1
Accion	Variable real con la accion a aplicar por el controlador (REAL) [°C]	
Accion2	Int [Accion]	%MW31
Uman	Valor de la accion que introduce el usuario al que debe ajustarse cuando está seleccionado el modo manual	%MW35
Accion3	Accion en número entero que se aplicará. Se tomará o la del PID o la manual	%MW33
Salida	20*Accion3 [mV]	%MW5
Setpoint	Valor al cual deberá ajustarse la señal (Real)	
Referencia	Variable en la que se almacena el valor introducido por el usuario al que debe ajustarse el proceso mientras funcione la fresadora (Real)	
Para_PI	Parámetros previamente calculados del regulador	

Tabla 8. Variables implementadas del PID en las cintas



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº4: Presupuesto

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenido

1	Introducción.....	56
2	Personal directamente implicado.....	56
3	Equipo Hardware amortizado	57
4	Equipo Software amortizado.....	58
5	Precios unitarios	58
6	Precios descompuestos	59
7	Presupuesto de ejecución material.....	62

Índice de tablas

Tabla 1. Mano de obra del ingeniero industrial	56
Tabla 2. Mano de obra del técnico.....	57
Tabla 3. Mano de obra del director de proyecto	57
Tabla 4. Tabla de precios del equipo Hardware.....	57
Tabla 5. Tabla de precios del equipo Software.....	58
Tabla 6. Precios unitarios	58
Tabla 7. Precio del estudio previo y redacción del presupuesto inicial	59
Tabla 8. Precio del diseño de la aplicación de la línea indexada.....	59
Tabla 9. Precio del diseño de la aplicación del brazo robótico.....	59
Tabla 10. Precio del modelado de la célula Peltier	60
Tabla 11. Precio del diseño del controlador de la célula Peltier	60
Tabla 12. Precio de la implementación del controlador en la aplicación principal.....	60
Tabla 13. Diseño del SCADA	61
Tabla 14. Precio de la redacción del manual de usuario	61
Tabla 15. Precio de la redacción del documento relativo al desarrollo del proyecto	61
Tabla 16. Precio de los técnicos.....	61
Tabla 17. Ocupación del personal directamente implicado.....	62
Tabla 18. Tiempo empleado de los equipos en el proyecto	62
Tabla 19. Presupuesto de ejecución material.....	62

1 Introducción

El presupuesto que se redactará será el correspondiente a la realización de las dos aplicaciones para el control de las maquetas y para la monitorización del proceso. Se parte de la base de que el cliente ha adquirido el proceso previamente y precisa de alguien que realice la programación como se ha explicado con anterioridad. Cualquier gasto en autómatas, maquetas, célula Peltier u otros elementos necesarios para la elaboración del proyecto corre a cargo del cliente.

Se realizará un análisis de:

- Personal directamente implicado
- Equipo Hardware amortizado
- Equipo Software amortizado
- Precios unitarios y descompuestos

Finalmente se presentará el presupuesto de ejecución material así como un análisis de los tiempos empleados por cada recurso.

2 Personal directamente implicado

Los recursos humanos empleados serán:

- Ingeniero Industrial especializado en Automatización, encargado de la realización del proyecto.
- Director de Proyecto, encargado de dirigir y resolver los conflictos que surjan durante el proyecto.
- Técnico, encargado del correcto mantenimiento de las instalaciones y recursos utilizados.

El gasto por hora del **Ingeniero** se detalla en la Tabla 1

Conceptos	Total(€)
Salario Base(225 días al año)	31212,66
Dietas	575,70
Seguridad social((Base cotización, contingencias comunes, accidentes trabajo, desempleo, fondo garantía salarial y formación profesional)	9363,80
Pluses (Transporte, desgaste herramientas, prendas trabajo)	600,00
Horas extra (10/mes)	2677,20
<i>Total al año</i>	44429,36
<i>Total a facturar por jornada (8h)</i>	197,46
<i>Total a facturar por hora</i>	24,68

Tabla 1. Mano de obra del ingeniero industrial

El gasto por hora del **Técnico** se detalla en la Tabla 2

Conceptos	Total(€)
Salario Base(225 días al año)	22555,92
Dietas	268,20
Seguridad social((Base cotización, contingencias comunes, accidentes trabajo, desempleo, fondo garantía salarial y formación profesional)	6766,78
Pluses (Transporte, desgaste herramientas, prendas trabajo)	200,00
Horas extra (10/mes)	2204,40
<i>Total al año</i>	31995,30
<i>Total a facturar por jornada (8h)</i>	142,20
<i>Total a facturar por hora</i>	17,78

Tabla 2. Mano de obra del técnico

El gasto por hora del **Director de Proyecto** se detalla en la Tabla 3

Conceptos	Total(€)
Salario Base(225 días al año)	37030,00
Dietas	872,40
Seguridad social((Base cotización, contingencias comunes, accidentes trabajo, desempleo, fondo garantía salarial y formación profesional)	11109,00
Pluses (Transporte, desgaste herramientas, prendas trabajo)	1000,00
Horas extra (10/mes)	3392,40
<i>Total al año</i>	53403,80
<i>Total a facturar por jornada (8h)</i>	237,35
<i>Total a facturar por hora</i>	29,67

Tabla 3. Mano de obra del director de proyecto

3 Equipo Hardware amortizado

Se ha estimado 3 años de amortización y unas horas laborables de 1000 horas trabajando (lo que supondría un uso de más de 1 hora al día).

$$\text{Precio a amortizar} = \frac{\text{Precio del equipo (€)}}{1000} \text{ €/h}$$

Equipo Hardware	Precio total (€)	Precio a amortizar por jornada (€/hora)
Ordenador	490,00	0,54
Fuente tensión controlable	425,00	0,43
Fuente de tensión variable	254,95	0,25

Tabla 4. Tabla de precios del equipo Hardware

4 Equipo Software amortizado

Equipo Software	Precio (€)	Precio a amortizar por jornada (€/hora)
LabVIEW 2013	3310,00	3,68
DSC Module	2950,00	3,28
Matlab	2000,00	2,96
System Identification Toolbox	1000,00	1,48
Control System Toolbox	1000,00	1,48
Simulink	3000,00	4,44
Unity Pro	3500,00	3,89
Windows 7	80,00	0,09
Microsoft Office	80,00	0,09
Kepware Server	2000,00	2,22

Tabla 5. Tabla de precios del equipo Software

La licencia de Matlab tiene una duración de 2 años, por lo que para el precio de amortización se han estimado 675 horas útiles de trabajo.

El resto de licencias tienen una duración de 3 años para los que se ha decidido aplicar una amortización de 1000 horas.

$$\text{Precio a amortizar} = \frac{\text{Precio de la licencia (€)}}{\text{Tiempo de amortización (h)}} \text{ €/h}$$

5 Precios unitarios

A continuación se muestran los costes relativos a los precios por unidad de obra

Nº	Descripción	Precio (€)
1	Estudio previo, redacción de presupuesto inicial	952,91
2	Diseño aplicación principal cintas	1911,77
3	Diseño aplicación principal brazo robótico	3184,54
4	Modelado del sistema de refrigeración	1015,62
5	Diseño del controlador para el modelo de refrigeración	1705,98
6	Implementación del controlador en la aplicación principal	331,34
7	Diseño de la aplicación para control remoto y monitorización	2562,93
8	Redacción del manual de usuario	1777,52
9	Redacción documentos de desarrollo del proyecto	156,69
10	Técnicos	1181,91

Tabla 6. Precios unitarios

6 Precios descompuestos

Se muestran seguidamente los precios descompuestos relativos a costes de fabricación.

Estudio previo y redacción de presupuesto inicial				
Concepto	Precio unitario (€)	Cantidad	Unidades	Total (€)
Ingeniero	24,68	35,00	horas	863,90
Director de proyecto	29,67	3,00	horas	89,01
TOTAL				952,91

Tabla 7. Precio del estudio previo y redacción del presupuesto inicial

Diseño aplicación principal cintas				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	60,00	horas	1480,98
Director de Proyecto	29,67	5,00	horas	148,34
Ordenador	0,54	60,00	horas	32,67
Unity Pro	3,89	60,00	horas	233,33
Kepware	2,22	5,00	horas	11,11
Windows 7	0,09	60,00	horas	5,33
TOTAL				1911,77

Tabla 8. Precio del diseño de la aplicación de la línea indexada

Diseño aplicación principal brazo robótico				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	100,00	horas	2468,30
Director de Proyecto	29,67	8,00	horas	237,35
Ordenador	0,54	100,00	horas	54,44
Unity Pro	3,89	100,00	horas	388,89
Kepware	2,22	12,00	horas	26,67
Windows 7	0,09	100,00	horas	8,89
TOTAL				3184,54

Tabla 9. Precio del diseño de la aplicación del brazo robótico

Modelado del sistema de refrigeración				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	30,00	horas	740,49
Director de Proyecto	29,67	3,00	horas	89,01
Ordenador	0,54	25,00	horas	13,61
Fuente tensión controlable	0,47	30,00	horas	14,17
Fuente de tensión variable	0,28	30,00	horas	8,50
Matlab	2,96	20,00	horas	59,26
System Identification Toolbox	1,48	10,00	horas	14,81
LabVIEW 2013	3,68	20,00	horas	73,56
Windows 7	0,09	25,00	horas	2,22
TOTAL				1015,62

Tabla 10. Precio del modelado de la célula Peltier

Diseño del controlador para el modelo de refrigeración				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	45,00	horas	1110,73
Director de Proyecto	29,67	5,00	horas	148,34
Ordenador	0,54	45,00	horas	24,50
Fuente tensión controlable	0,47	45,00	horas	21,25
Fuente de tensión variable	0,28	45,00	horas	12,75
Matlab	2,96	40,00	horas	118,52
Control System Toolbox	1,48	15,00	horas	22,22
Simulink	4,44	30,00	horas	133,33
LabVIEW 2013	3,68	30,00	horas	110,33
Windows 7	0,09	45,00	horas	4,00
TOTAL				1705,98

Tabla 11. Precio del diseño del controlador de la célula Peltier

Implementación del controlador en la aplicación principal				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	10,00	horas	246,83
Director de Proyecto	29,67	1,00	horas	29,67
Ordenador	0,54	8,00	horas	4,36
Fuente tensión controlable	0,47	10,00	horas	4,72
Fuente de tensión variable	0,28	10,00	horas	2,83
Unity Pro	3,89	8,00	horas	31,11
Windows 7	0,09	8,00	horas	0,71
Kepware	2,22	5,00	horas	11,11
TOTAL				331,34

Tabla 12. Precio de la implementación del controlador en la aplicación principal

Diseño de la aplicación para control remoto y monitorización				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	70,00	horas	1727,81
Director de Proyecto	29,67	5,00	horas	148,34
Ordenador	0,54	70,00	horas	38,11
LabVIEW 2013	3,68	70,00	horas	257,44
DSC Module	3,28	70,00	horas	229,44
Kepware	2,22	70,00	horas	155,56
Windows 7	0,09	70,00	horas	6,22
TOTAL				2562,93

Tabla 13. Diseño del SCADA

Redacción del manual de usuario				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	5,00	horas	123,41
Director de Proyecto	29,67	1,00	horas	29,67
Ordenador	0,54	5,00	horas	2,72
Windows 7	0,09	5,00	horas	0,44
Microsoft Office	0,09	5,00	horas	0,44
TOTAL				156,69

Tabla 14. Precio de la redacción del manual de usuario

Redacción documentos de desarrollo del proyecto				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Ingeniero	24,68	40,00	horas	987,32
Director de Proyecto	29,67	5,00	horas	148,34
Ordenador	0,54	40,00	horas	21,78
Windows 7	0,09	40,00	horas	3,56
Microsoft Office	0,09	40,00	horas	3,56
LabVIEW 2013	3,68	2,00	horas	7,36
Unity Pro	3,89	2,00	horas	7,78
Kepware	2,22	1,00	horas	2,22
TOTAL				1181,91

Tabla 15. Precio de la redacción del documento relativo al desarrollo del proyecto

Técnicos				
Concepto	Precio unitario (€)	Cantidad	Unidad	Total (€)
Técnico de Mantenimiento	17,78	100,00	horas	1777,52
TOTAL				1777,52

Tabla 16. Precio de los técnicos

Se muestran las horas invertidas por el personal y los materiales utilizados. Para la posterior planificación temporal del proyecto es importante saber estos datos para decidir cuantas unidades de cada elemento tomar con el fin de distribuir de forma efectiva las tareas.

Personal implicado en el proyecto	Horas
Ingeniero	395,00
Director de proyecto	36,00
Técnico	100,00

Tabla 17. Ocupación del personal directamente implicado

Elemento	Horas
Ordenador	353,00
Fuente de tensión controlable	85,00
Fuente de tensión variable	85,00
LabVIEW 2013	122,00
DSC Module	70,00
Matlab	60,00
System Identification Toolbox	10,00
Control System Toolbox	15,00
Simulink	30,00
Unity Pro	170,00
Windows 7	353,00
Microsoft Office	45,00
Kepware Server	93,00

Tabla 18. Tiempo empleado de los equipos en el proyecto

7 Presupuesto de ejecución material

Nº	Descripción	Precio (€)
1	Estudio previo, redacción de presupuesto inicial	952,91
2	Diseño aplicación principal cintas	1911,77
3	Diseño aplicación principal brazo robótico	3184,54
4	Modelado del sistema de refrigeración	1015,62
5	Diseño del controlador para el modelo de refrigeración	1705,98
6	Implementación del controlador en la aplicación principal	331,34
7	Diseño de la aplicación para control remoto y monitorización	2562,93
8	Redacción del manual de usuario	1777,52
9	Redacción documentos de desarrollo del proyecto	156,69
10	Técnicos	1181,91
TOTAL		14781,21

Tabla 19. Presupuesto de ejecución material

El presupuesto de ejecución material asciende a la cantidad de **catorce mil setecientos ochenta y un euros con veintiún céntimos**.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº5: Anexo de diseño

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenidos

1	Introducción.....	66
2	Identificación del modelo.....	66
2.1	Identificación manual	66
2.1.1	Zona caliente:	67
2.1.2	Zona fría	68
2.2	Identificación mediante la herramienta Ident y validación	69
3	Cálculo del regulador.....	71
4	Validación del regulador	72
5	Implementación del controlador y validación del comportamiento real	73
5.1	Implementación en Labview	73
5.2	Validación del sistema de control	73
6	Ampliaciones.....	75

Índice de figuras

Figura 1. Carga de un archivo en Matlab	66
Figura 2. Ensayo de la zona de trabajo	67
Figura 3. Escalón negativo de la zona caliente	67
Figura 4. Escalón positivo de la zona caliente.....	68
Figura 5. Escalón negativo de la zona fría	68
Figura 6. Escalón positivo de la zona fría	69
Figura 7. Bondad de los modelos	69
Figura 8. Modelos calculados mediante la herramienta Ident	70
Figura 9. Comparación del ajuste de todos los modelos calculados	70
Figura 10. Esquema del PID.....	71
Figura 11. Diagrama de bloques de SIMULINK para la simulación del regulador	72
Figura 12. Simulación del regulador	72
Figura 13. Diagrama de bloques de SIMULINK para comparación de reguladores	74
Figura 14. Comparativa de reguladores simulado y real	74
Figura 15. Comparación de acción ideal-real.....	75
Figura 16. Desconexión del ventilador	76
Figura 17. Acción tras desconexión del ventilador.....	76

1 Introducción

Se van a realizar los ensayos¹⁴ necesarios para obtener un controlador efectivo y sus parámetros. Los parámetros serán introducidos en el autómata, para que éste sea el encargado no sólo de controlar la línea indexada y el Robot 3-D sino también de controlar el comportamiento de la Peltier.

Para todos los ensayos se tomará una acción del ventilador del 50% para evitar el fenómeno por el cual, debido a la proximidad entre las caras de la Peltier, la cara caliente influye en la cara fría calentándola e impidiendo un modelo lineal de temperaturas que es lo que interesa. Aún con el ventilador, no se puede asegurar dicho comportamiento en todo el rango de actuación de la célula Peltier, por lo que únicamente se trabajará con el rango de temperaturas comprendido entre 0 y 13 °C, que por otro lado, resulta más que suficiente para la aplicación que luego se le dará a la Peltier.

2 Identificación del modelo

2.1 Identificación manual

Una vez realizado el ensayo, los datos se guardan en un archivo que hay que cargar desde Matlab¹⁵. Se han tomado varios archivos, el de Ensayo y el de Validación y se han extraído sus primeras dos columnas en variables. Representarán el tiempo y la referencia.

También se extraerá la cuarta columna que representa el valor del proceso, es decir, la temperatura de la cara fría en todo momento

La *Figura 1* presenta el aspecto de la carga del archivo

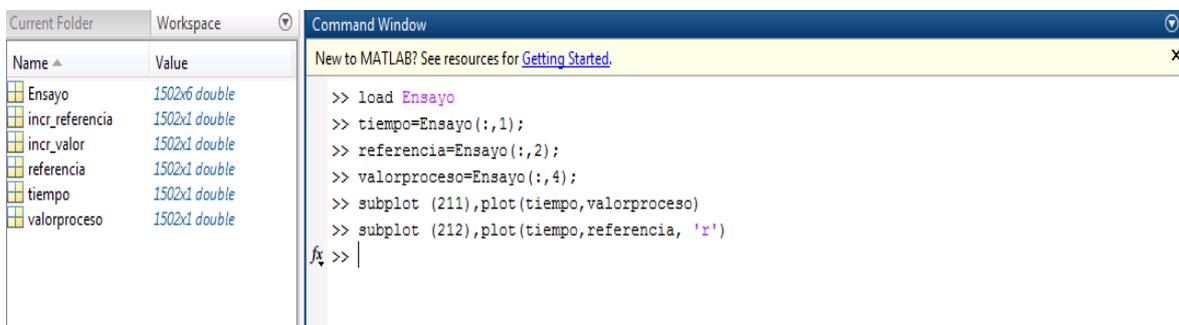


Figura 1. Carga de un archivo en Matlab

La imagen que se muestra en la *Figura 2* será el valor del proceso en función de los cambios efectuados en la referencia.

¹⁴ La guía para la realización de los ensayos la encontramos en [5] (Herrero, Apuntes de la asignatura LAC, 2014-2015)

¹⁵ Una guía práctica de cómo realizar estas funciones de Matlab la podemos encontrar en [6] (Herrero, Tutorial Simulink, Identificación y validación de Fdts)

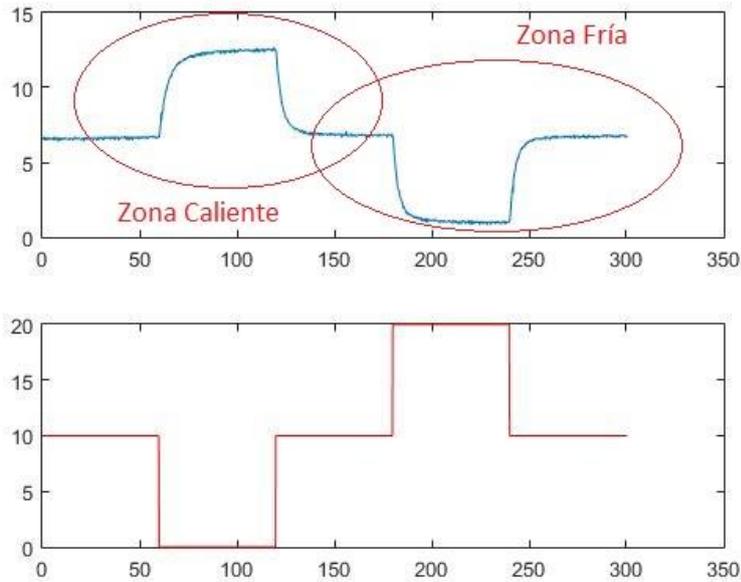


Figura 2. Ensayo de la zona de trabajo

Se ha dividido en dos zonas que inicialmente se estudiarán por separado para luego realizar un modelo conjunto que se ajuste lo mejor posible al comportamiento de la célula Peltier en la zona comprendida entre los 0 y los 13 °C.

En ambas zonas hay un escalón positivo y uno negativo respectivamente, para los cuales se calculará el modelo por separado.

El modelo responderá a la fórmula

$$G(s) = \frac{K}{1 + T_s \cdot s} \cdot e^{(-\theta \cdot s)}$$

2.1.1 Zona caliente:

- Escalón negativo

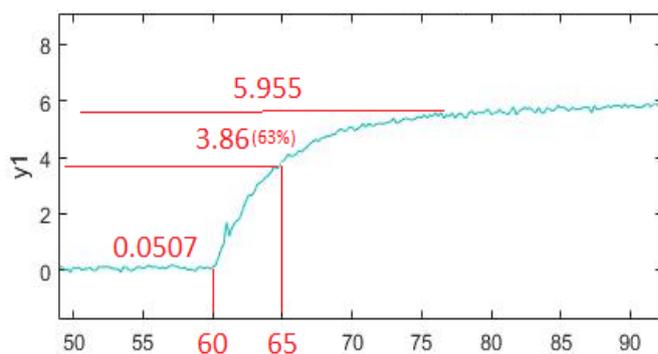


Figura 3. Escalón negativo de la zona caliente

La ganancia del sistema se calculará como:

$$K = \frac{5.955 - 0.0507}{0 - 10} = -0.5904$$

La constante de tiempo se definirá como:

$$T_s = 65 - 60 = 5s$$

El modelo calculado será:

$$G(1) = \frac{-0.5904}{1 + 5 \cdot s}$$

- **Escalón positivo**

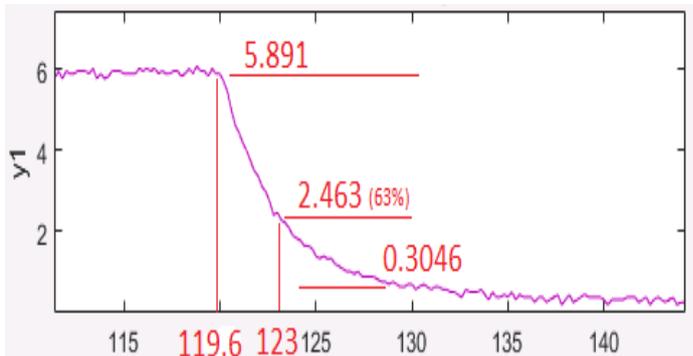


Figura 4. Escalón positivo de la zona caliente

La ganancia del sistema se calculará como:

$$K = \frac{0.3046 - 5.891}{10 - 0} = -0.55864$$

La constante de tiempo se definirá como:

$$T_s = 123 - 119.6 = 3.4s$$

El modelo calculado será:

$$G(2) = \frac{-0.55864}{1 + 3.4 \cdot s}$$

2.1.2 Zona fría

- **Escalón negativo**

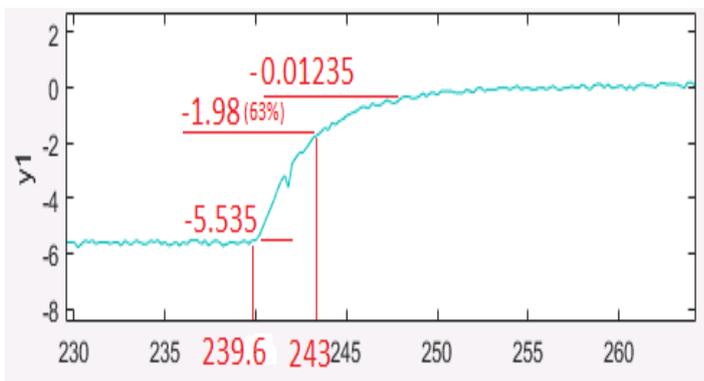


Figura 5. Escalón negativo de la zona fría

La ganancia del sistema se calculará como:

$$K = \frac{-0.01235 + 5.535}{10 - 20} = -0.55227$$

La constante de tiempo se definirá como:

$$T_s = 243 - 239.6 = 3.4s$$

El modelo calculado será:

$$G(3) = \frac{-0.55227}{1 + 3.4 \cdot s}$$

- **Escalón positivo**

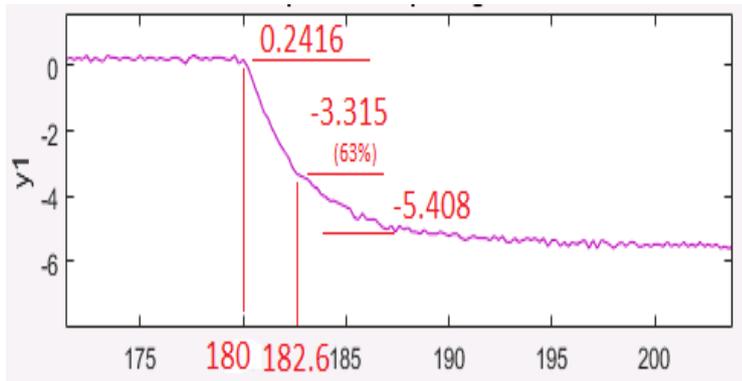


Figura 5. Escalón positivo de la zona fría

La ganancia del sistema se calculará como:

$$K = \frac{-5.408 + 0.2416}{20 - 10} = -0.56496$$

La constante de tiempo se definirá como:

$$T_s = 182.6 - 180 = 2.6s$$

El modelo calculado será:

$$G(4) = \frac{-0.56496}{1 + 2.6 \cdot s}$$

En ninguno de los modelos se aprecia un retardo significativo, por lo que en todo momento responde a un modelo sobreamortiguado sin retardo¹⁶.

Realizando el promedio de los cuatro modelos anteriormente calculados se obtendrá un modelo con la fórmula que se muestra a continuación:

$$G(\text{Promedio}) = \frac{-0.56658}{1 + 3.6s}$$

2.2 Identificación mediante la herramienta Ident y validación

Mediante la herramienta Ident se puede comprobar la bondad de los diseños realizados. Los resultados de la comparativa se muestran en la Figura 7.

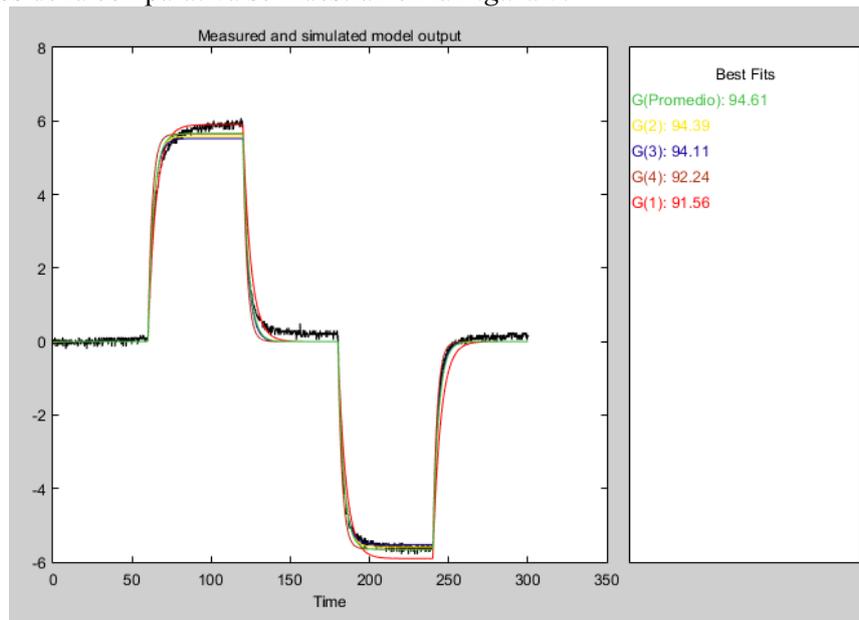


Figura 7. Bondad de los modelos

¹⁶ [1] (Bondia & Sala, 2013-2014)

Se aprecia que todos los modelos se ajustan con una exactitud superior al 90%, lo cual es un ajuste bueno.

Es el modelo promedio el que sin embargo logra un mejor ajuste, cercano al 95%.

Mediante la herramienta Ident también se puede calcular el modelo de manera automática¹⁷ y los modelos resultantes se muestran en la *Figura 8*.

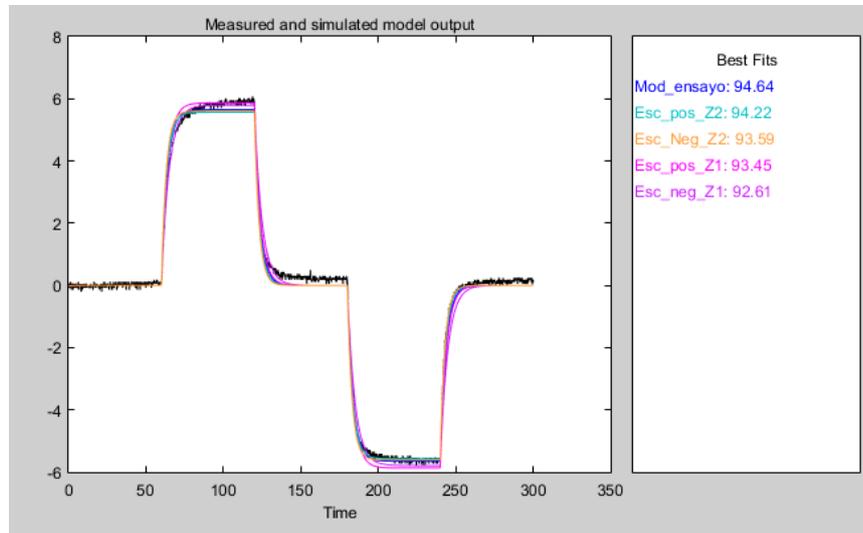


Figura 8. Modelos calculados mediante la herramienta Ident

Como se aprecia, los resultados son muy similares a los obtenidos en el cálculo a mano y, al igual que en éste, el modelo que mejor se ajusta es el que se ha realizado tomando como datos de partida todos los correspondientes al ensayo.

Se procederá a realizar un siguiente ensayo con diferentes cambios de referencia para validar los modelos obtenidos y ver cual consigue un mejor ajuste tanto en el modelo del ensayo como en el de validación.

Los resultados se muestran en la *Figura 9*

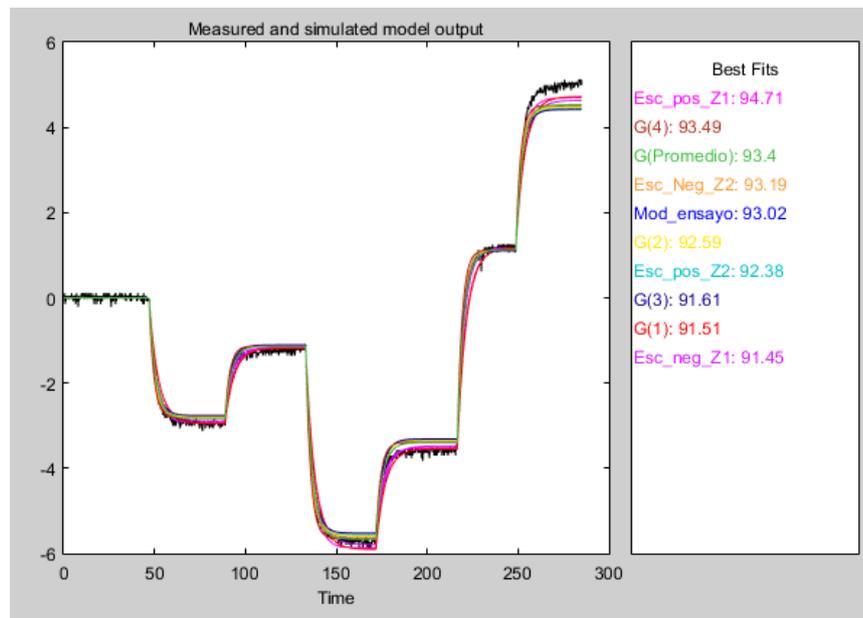


Figura 9. Comparación del ajuste de todos los modelos calculados

¹⁷ Se ha seguido las indicaciones proporcionadas por el documento [5] (Herrero, Apuntes de la asignatura LAC, 2014-2015) Y [6] (Herrero, Tutorial Simulink, Identificación y validación de Fdts)

Como se observa, todos los resultados resultarían aceptables puesto que el ajuste es reseñable (90%).

Los parámetros de cada uno de los modelos calculados no varían significativamente, por lo tanto se concluye que cualquier modelo ajustaría el comportamiento real de la célula Peltier con tolerancias pequeñas.

Se ha optado por seleccionar el modelo G(Promedio) puesto que en ensayo y validación presenta de los mejores ajustes.

El modelo elegido será:

$$G(\text{Promedio}) = \frac{-0.56658}{1 + 3.6s}$$

3 Cálculo del regulador

Dado que por las características del proceso no se precisa acción derivativa, el PID más sencillo será un controlador con acción proporcional y acción integral, un PI.

Los parámetros del mismo se calcularán para la fórmula del PID:

$$G(r) = K_c \cdot \frac{(s + T_d \cdot s^2 + \frac{1}{T_i})}{s}$$

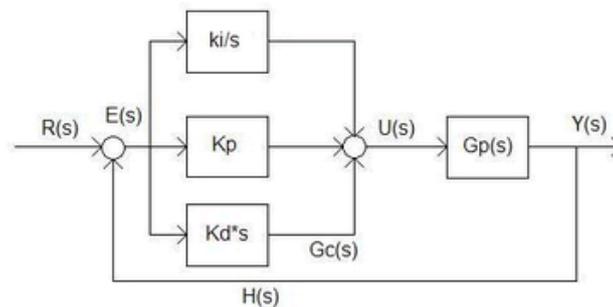


Figura 10. Esquema del PID

Se va a utilizar el método Ciancone para el cálculo del controlador.

Las razones por las que se ha elegido este regulador son:

- Minimiza el IAE (Integral absoluta de error) ante cambios de referencia o perturbación.
- Considera incertidumbres en los parámetros del modelo de $\pm 25\%$.
- Considera limitaciones en la variación de la variable manipulada

Se calculan los parámetros¹⁸:

$$K_c = -2.29$$

$$T_i = 2.7 \text{ segundos}$$

El regulador completo será:

$$G(r) = -2.29 \cdot \frac{(s + \frac{1}{2.7})}{s}$$

¹⁸ [5] (Herrero, Apuntes de la asignatura LAC, 2014-2015)

4 Validación del regulador

Mediante la herramienta Simulink de Matlab se va a validar el regulador calculado. Se construirá el sistema de control como se muestra a continuación¹⁹ (Figura 11):

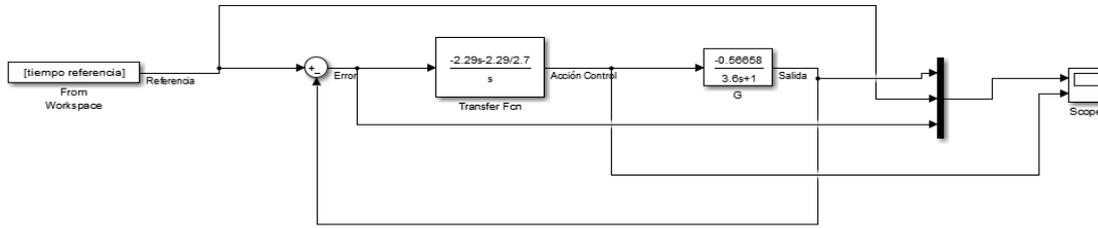


Figura 11. Diagrama de bloques de SIMULINK para la simulación del regulador

Cargando el archivo de validación y definiendo las variables tiempo y referencia como ya se ha hecho anteriormente, se podrán ver los cambios en la referencia y como el controlador calculado se ajusta a esta referencia (parte superior, curva en azul) y la acción de control que va aplicando en cada instante (parte inferior, en azul):

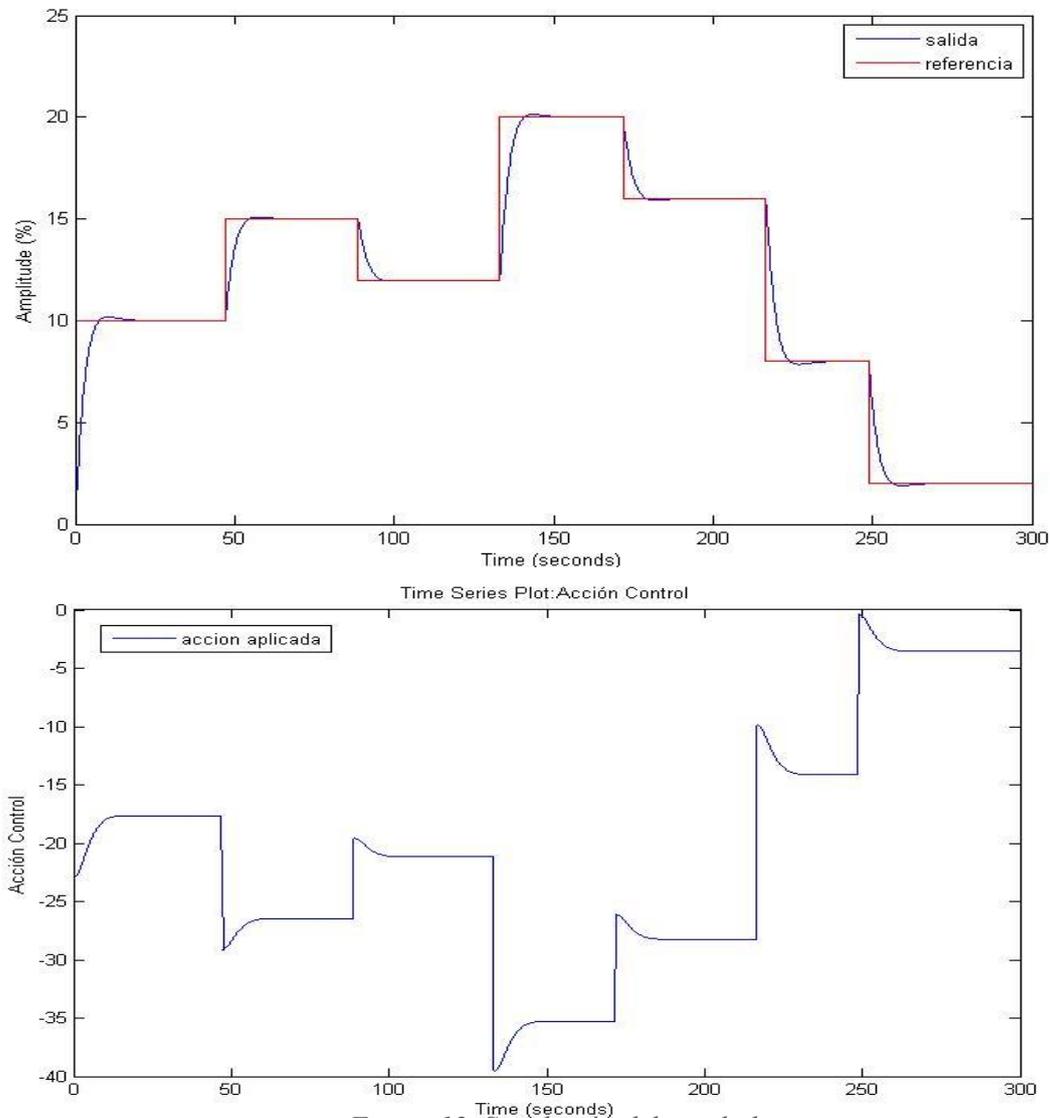


Figura 12. Simulación del regulador

¹⁹[5] (Herrero, Apuntes de la asignatura LAC, 2014-2015)

Se puede concluir que el regulador es lo suficientemente rápido y se ajusta correctamente a la referencia sin presentar oscilaciones (comportamiento sobreamortiguado). Por ello se elige este regulador y no se precisa otro con mejores prestaciones.

5 Implementación del controlador y validación del comportamiento real

5.1 Implementación en LabVIEW

Se va a diseñar una aplicación auxiliar en LabVIEW para aplicar el control a la célula Peltier y estudiar posteriormente si el comportamiento era el esperado o por el contrario el comportamiento real del regulador difiere del comportamiento simulado anteriormente mediante Simulink.

La aplicación²⁰, similar a la ya empleada para la identificación del modelo, incorpora un cuadro de programación donde se especificará el siguiente código:

```
if (man==1)
E=R-yanterior; //Error será igual a la referencia menos el valor anterior
Up=P*E; //acción proporcional
Ui=Uianterior+P*T*E/Ti; //acción integral
Ud=P*Td*(E-Eanterior)/T; //acción derivada
Eanterior=E;
u=Up+Ui+Ud;
else
u=Uman;
end
if(u>100) //antiwindup
u=100;
Ui=Uianterior;
end
if(u<0) //antiwindup
u=0;
Ui=Uianterior;
end
yanterior=y;
```

R=referencia.

y=valor de la temperatura de la cara fría.

u=acción de control del 0 al 100%.

man=botón para seleccionar entre la acción calculada por el PID cuando es 'true' y la acción proporcionada manualmente, Uman, cuando es 'false'.

El código incorpora una acción antiwindup para saturar la acción de control y evitar la acumulación de error.

5.2 Validación del sistema de control

Desde el panel frontal de la aplicación se podrán manipular los parámetros del PID.

Se han introducido los parámetros calculados mediante el método Ciancone y realizado un ensayo con varios cambios en la referencia (escalones). Posteriormente, mediante la herramienta Simulink, se someten a comparación los datos de la respuesta real con la respuesta simulada de Simulink con la siguiente estructura de diagrama de bloques (*Figura 13*):

²⁰Las bases del diseño de la aplicación se tomarán del instrumento virtual (VI) Controlplantilla.vi; Autor: Juan Manuel Herrero Durá

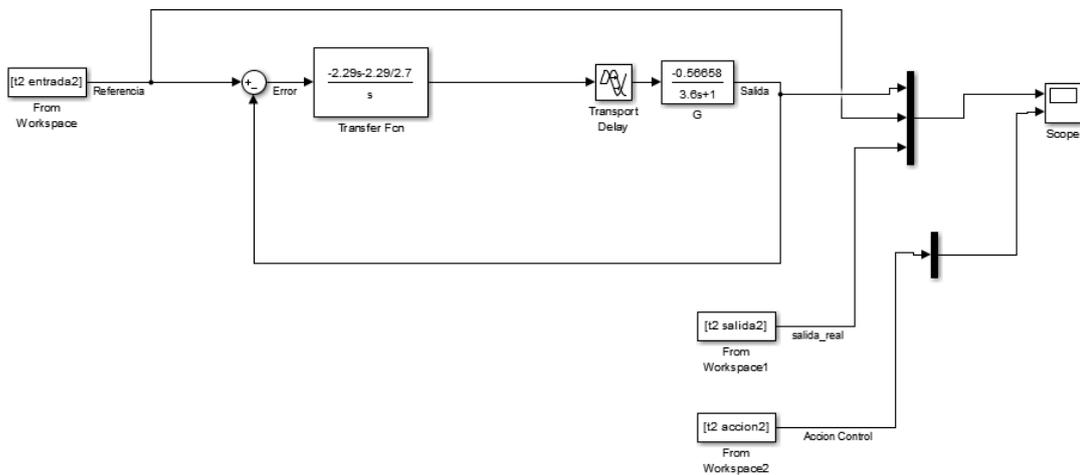


Figura 13. Diagrama de bloques de SIMULINK para comparación de reguladores
 Por el Scope se mostrarán las diversas señales como en la Figura 14

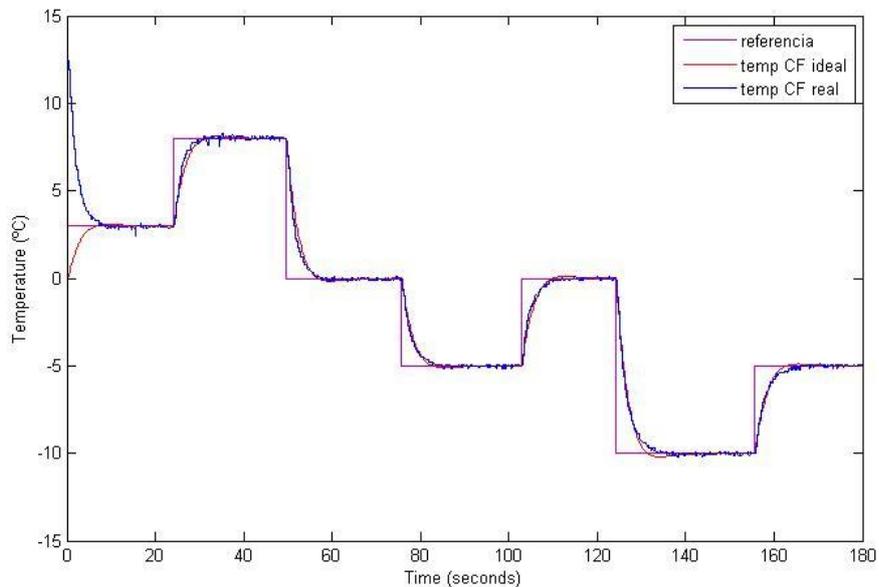


Figura 14. Comparativa de reguladores simulado y real

Se observa que el comportamiento real (azul) y el simulado (rojo) son parejos. La respuesta real es suficiente para el proceso que se desarrolla en el presente proyecto y se implementará finalmente en el autómata para que controle de manera efectiva la célula Peltier como se explicará en el manual de programador 2.2.10 Implementación del PID (Plano 8):.

En la Figura 15 se aprecia que la acción de control real (rojo) y la simulada (azul) son similares. Los escalones negativos de la acción de control representan calentamientos de la cara fría de la célula Peltier. En el proceso real se precisa menos acción para calentar que en el simulado. Eso es debido a que en Valencia, lugar donde se realizan los ensayos, la temperatura ambiente es elevada y debido a los efectos termodinámicos se calienta con más facilidad. Lo contrario ocurre en los escalones positivos de la acción de control, que implican un enfriamiento de la cara fría de la célula Peltier. Ocurre lo mismo a lo descrito con anterioridad pero al contrario, por lo tanto, la acción real que hay que aplicar para que se enfríe debe ser mayor que la simulada.

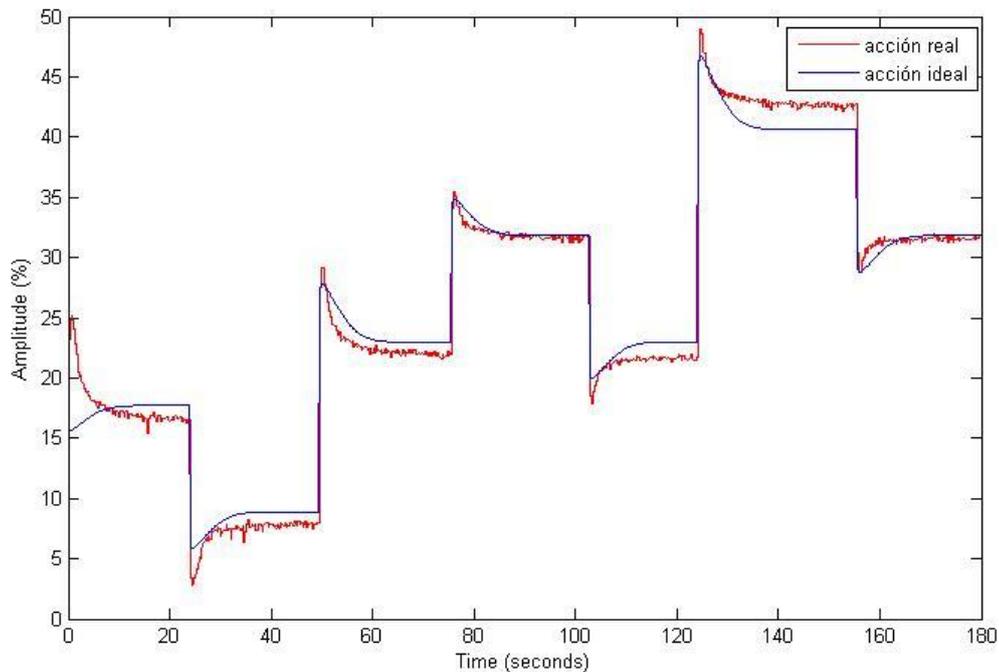


Figura 15. Comparación de acción ideal-real

6 Ampliaciones

Durante el presente trabajo se ha tomado la temperatura externa al proceso como una variable que no afecta al mismo. Sin embargo, se trata también de una variable de interés que influye en nuestro proceso, pues influye en el rendimiento del ventilador.

El calor disipado por un ventilador es directamente proporcional la siguiente fórmula:

$$Q = k \cdot (T_{rad} - T_{amb})$$

Donde k es una constante del ventilador, T_{rad} es la temperatura del radiador y T_{amb} es la temperatura ambiente. Si en algún momento la temperatura ambiente iguala o excede la temperatura del radiador, el ventilador dejaría de disipar calor, por lo que ocurriría el fenómeno descrito en las [Figuras 14 y 15 de la memoria](#).

Ocurriría que la cara caliente influiría sobre la fría, que se calentaría. De esta forma, la cara fría no sería capaz de mantener la temperatura de referencia que se le proporciona y esto influiría de manera negativa en todo el proceso pues no refrigeraría como se necesita.

El presente regulador ha sido calculado para mantener en todo momento la referencia.

Para demostrar este hecho, se ha realizado un ensayo desconectando el ventilador y viendo cómo reacciona el valor de la temperatura de la cara fría y la acción de control aplicada por el regulador. El resultado se muestra en las [Figuras 16 y 17](#).

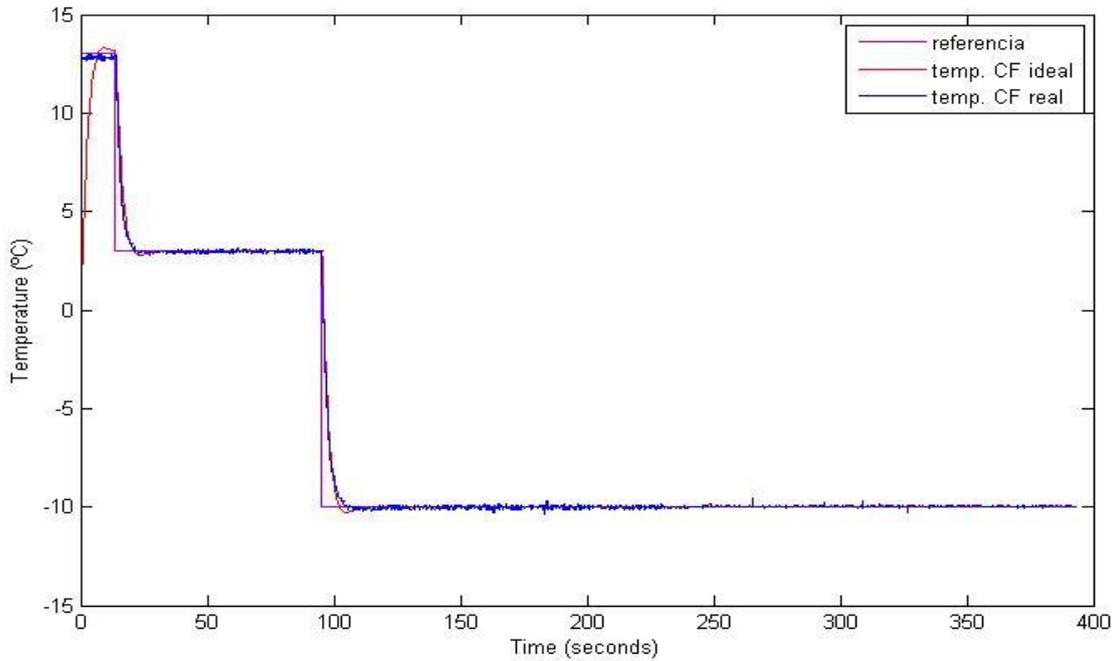


Figura 16. Desconexión del ventilador

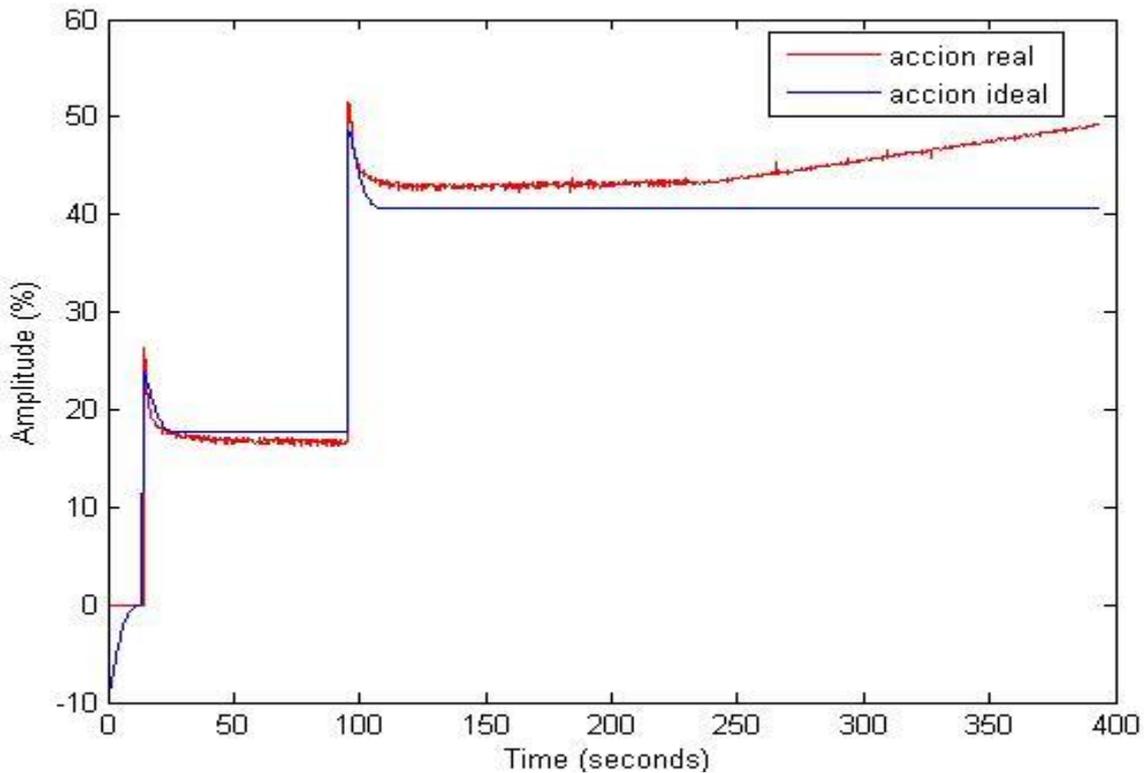


Figura 17. Acción tras desconexión del ventilador

Se aprecia que el valor de la temperatura de la cara fría se mantiene en la referencia, que es lo que interesa para el correcto funcionamiento del proceso.

La acción de control real (rojo) y la simulada (azul) difieren sustancialmente.

Esto se debe a que el regulador va aumentando la acción de control progresivamente para mantener la temperatura de la cara fría a la referencia.

Se concluye que el regulador es correcto también para situaciones inicialmente no contempladas como un cambio brusco de la temperatura externa.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº6: Manual de programador

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenidos

1 Introducción.....	82
2 Unity Pro	82
2.1 Graficets del Brazo Robótico.....	82
2.1.1 Graficet Principal (Plano 1).....	82
2.1.2 Auxiliares Brazo (Plano 2).....	83
2.1.2.1 Inicialización.....	83
2.1.2.2 Disponibilidad de cintas (DCR).....	83
2.1.2.3 Proceso (PRO)	83
2.1.2.4 Modos de Marcha	84
2.1.3 Modo de Alarma (Plano 3).....	84
2.1.4 Contadores de pulsos (Plano 3).....	85
2.1.5 FIFO (Plano 4)	85
2.1.6 Contadores (Plano4).....	86
2.2 Graficets de la Línea indexada.....	86
2.2.1 Activación sensores (Plano 7)	86
2.2.2 Cinta de carga/Cinta 1 (Plano 5)	86
2.2.3 Cinta de la fresadora/Cinta 2 (Plano 5)	87
2.2.4 Cinta de la taladradora/Cinta 3 (Plano 6)	87
2.2.5 Cinta de descarga/Cinta 4 (Plano 6).....	87
2.2.6 Disponibilidad de cintas (Plano 6)	88
2.2.7 Modos de marcha (Plano 6).....	88
2.2.8 Modos de alarma (Plano 7)	88
2.2.9 Fifo (Plano 7)	88
2.2.10 Implementación del PID (Plano 8).....	88
2.2.10.1 Entradas	89
2.2.10.2 Salidas.....	89
2.2.10.3 Parámetros del regulador	89
2.2.11 Comunicación.....	90
3 LabVIEW.....	91
3.1 Bloque de inicialización de variables	91
3.2 Bloque de inicio de sesión	91
3.3 Bloque del programa principal	92
3.3.1 Bucle principal	93
3.3.2 Bucle de movimiento manual.....	96

3.3.3 Bucle de movimiento fluido de cintas 1 y 4	97
3.3.4 Bucle de movimiento de las cintas 2 y 3	98
3.3.5 Bucle de Seguridad.....	98
3.3.6 Bucle Peltier	99
3.3.7 Bucle Inspección	101
3.3.8 Bucle Paso a Paso.....	102
3.3.9 Bucle Desactivación de modos de marcha	104
3.3.10 Bucle de control de usuarios	107

Índice de figuras

Figura 1. Brazo en posición de “Inicializado”	82
Figura 2. Brazo en cinta de descarga.....	83
Figura 3. Fresadora realizando uno de los procesos	84
Figura 4. Bloques de función de SFC Chart Management.....	85
Figura 5. Bloque de función FIFO	85
Figura 6. Línea indexada en marcha.....	87
Figura 7. Parámetros del PID	89
Figura 8. Ethernet del autómeta	90
Figura 9. Diagrama de los datos compartidos	90
Figura 10. Diagrama de bloques. Inicialización de variables	91
Figura 11. Ventana emergente de inicio de sesión.....	92
Figura 12. Código ventana emergente de inicio de sesión	92
Figura 13. Código y aspecto en la pantalla	93
Figura 14. Código de los modos de marcha	94
Figura 15. Código para representación gráfica.....	94
Figura 16. Aspecto del bucle principal completo.....	95
Figura 17. Botones de movimiento manual en pantalla.....	96
Figura 18. Código del bucle de movimiento manual.....	96
Figura 19. Desplazamiento de pieza por cinta en pantalla	97
Figura 20. Código del bucle de movimiento de las cintas 1 y 4	97
Figura 21. Fresadora actuando en pantalla	98
Figura 22. Código del bucle de movimiento de cinta 2 y 3	98
Figura 23. Código del bucle de seguridad	99
Figura 24. Controles de la célula Peltier en pantalla	99
Figura 25. Código del bucle de la Peltier.....	100
Figura 26. Bloque Case del bucle Peltier.....	100
Figura 27. Código del bucle Inspección.....	101
Figura 28. Controles de inspección en pantalla	101
Figura 29. Modo de marcha “Paso a Paso” activo	103
Figura 30. Modo de marcha “Paso a Paso” no activo	103
Figura 31. Código del bucle Paso a Paso	104
Figura 32. Cambio de modo de marcha	105
Figura 33. Confirmación del cambio realizado	105
Figura 34. Nuevo modo de marcha activo.....	105

Figura 35. Código parcial bucle de desactivación de modos de marcha	106
Figura 36. Código completo del bucle de desactivación de modos de marcha	107
Figura 37. Código bucle de control de usuarios	108

1 Introducción:

En el presente documento se detalla el código empleado en Unity Pro y en LabVIEW para conformar las aplicaciones. Para la parte del Unity se empleará la ayuda gráfica de los planos.

2 Unity Pro

2.1 Graficets del Brazo Robótico

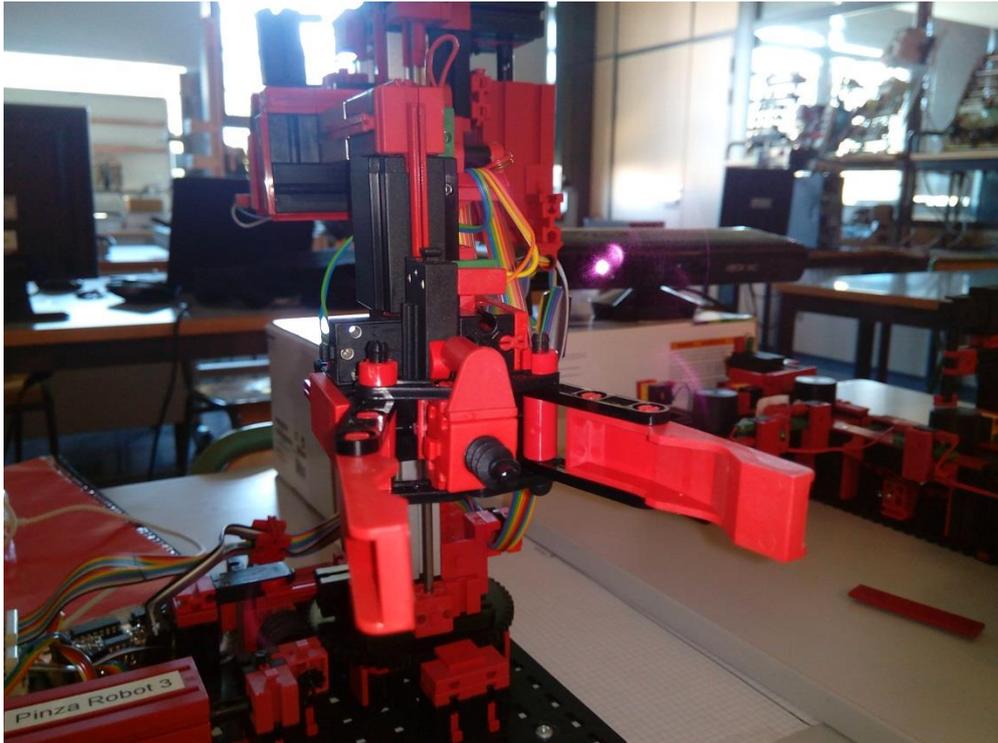


Figura 1. Brazo en posición de “Inicializado”

2.1.1 Graficet Principal (Plano 1):

Cuando se pone en marcha entra en la primera etapa, donde espera hasta que está en la posición de “Inicializado”. Seguidamente recibe un modo de marcha y se desplaza hasta la cinta de descarga. Cuando está situado en el sitio exacto (*Figura 2*), cierra la pinza hasta agarrar la pieza. Posteriormente espera uno de los tres casos siguientes:

- **Realizar el proceso 1 o acabado superficial:** Recibe el proceso del FIFO o bien la señal de cambio y se desplaza hasta la cinta de carga o cinta 1 para dejar la pieza. Seguidamente vuelve a la etapa 1.
- **Realizar el proceso 2 o cilindrado:** Recibe el proceso del FIFO y se desplaza a la caja de almacenaje siempre que no se hubiera seleccionado cambio de pieza o pieza defectuosa previamente. Una vez en la posición de la caja, suelta la pieza y seguidamente vuelve a la etapa 1 (Inicializado).
- **Llevar a descarte:** Cuando se seleccione pieza defectuosa, independientemente del proceso que debiera realizar la pieza, ésta es llevada a un palé de descartes. Seguidamente vuelve a la etapa 1.

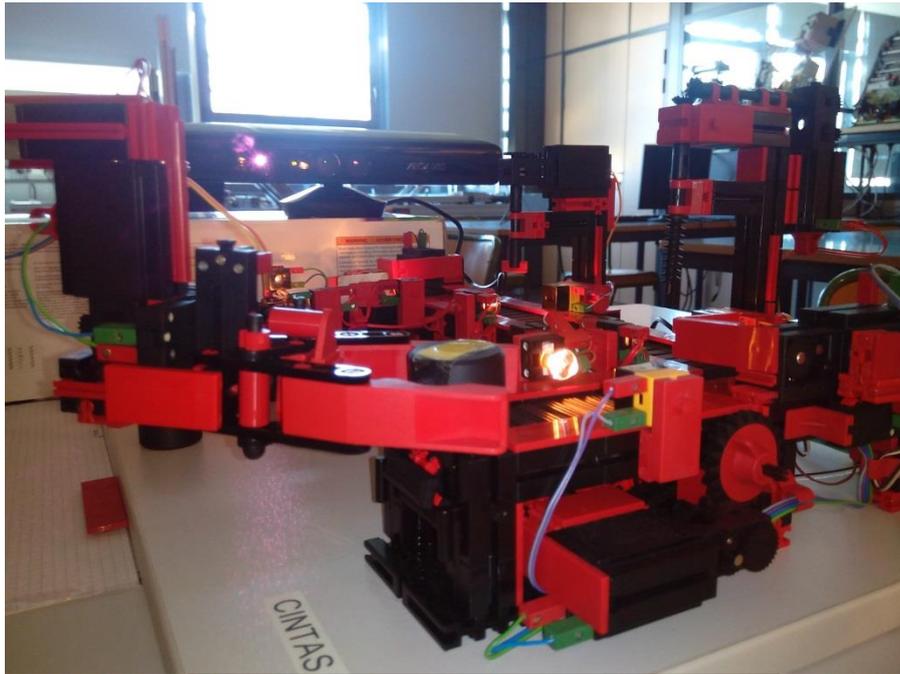


Figura 2. Brazo en cinta de descarga

2.1.2 Auxiliares Brazo (Plano 2):

2.1.2.1 Inicialización (Figura 1):

Inicialmente se encuentra en una etapa vacía hasta que el Grafcet principal entra en la etapa 1 o bien se pulsa desde la pantalla el botón de Inicializar. Comienza un proceso por el que el brazo se desplaza a la posición de “Inicializado” que es el lugar en el espacio en el cual todos los sensores finales de carrera están activos. Cuando esto ocurre, este Grafcet vuelve a su etapa inicial.

2.1.2.2 Disponibilidad de cintas (DCR):

- **Cinta de carga:** Mientras los fototransistores no detectan pieza, la cinta 1 o de carga se encuentra disponible. Cuando detectan o bien el brazo deja pieza en la cinta de carga, la cinta de carga se encuentra ocupada.
- **Cinta de descarga:** Cuando la cinta de descarga detecta una pieza, entra en la etapa de “pieza lista para ser recogida”. Cuando el brazo la recoge, vuelve a la etapa de “No hay pieza”.

2.1.2.3 Proceso (PRO):

- **Determinar la maqueta que deja la pieza:** Cuando se detecta una pieza en la cinta de carga hace falta saber si esa pieza es nueva y no ha realizado ningún proceso (la deja la cinta) o si esa pieza ya ha pasado por el proceso 1 (la deja el brazo).
- **Selección de proceso:** Sabiendo quién la deja, se sabe también cuál es el siguiente proceso que ha de realizar. Si la deja la cinta, esa pieza se “marcará” con un 0 de que no ha realizado ningún proceso todavía. Si la deja el brazo robótico se “marca” con un 1 de que ya ha realizado un proceso.

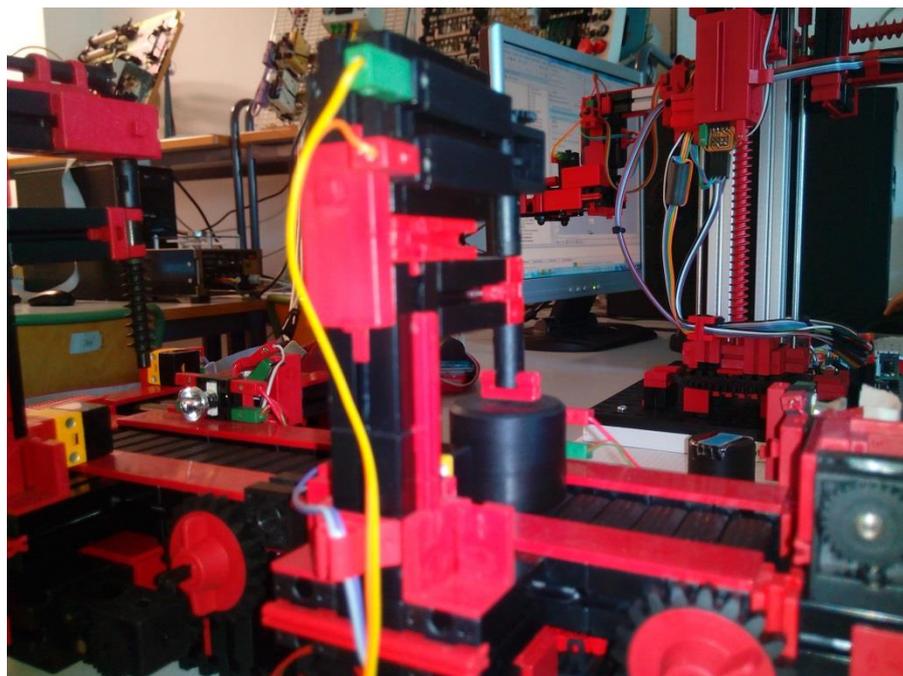


Figura 3. Fresadora realizando uno de los

2.1.2.4 Modos de Marcha:

En este Grafcet se selecciona entre los 3 modos de marcha posibles:

- **Automático:** Entra en este modo cuando se selecciona en la pantalla el botón “Auto”. Sale del modo cuando se han realizado las piezas indicadas o bien se desactiva el botón “Auto”.
- **Paso a Paso:** Entra en el modo cuando se pulsa el botón “PaP” de la pantalla y sale cuando se desactiva.
- **Ciclo a ciclo:** Entra en el modo cuando se activa el botón “CAC” y sale del modo cuando se desactiva.

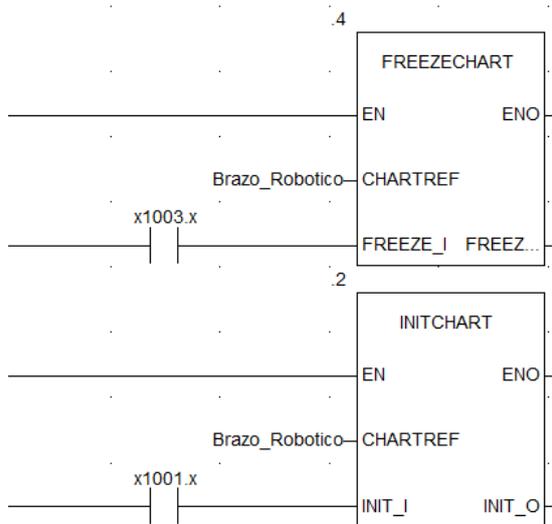
2.1.3 Modo de Alarma (Plano 3):

En este Grafcet se selecciona entre los 3 modos de alarma posibles:

- **Inspección**
- **Inicializar**
- **Congelar**

En todos estos modos se entra pulsando el botón correspondiente de la pantalla. Para inicializar o congelar todo el robot y las cintas se han empleado sendos bloques de función²¹ presentes en Unity que realizan esta función. A continuación se muestra una imagen de ambos bloques.

²¹ Para más información sobre los bloques de función mencionados pulsar sobre el bloque correspondiente con el botón derecho y seleccionar la “Ayuda en tipo” presente en el programa Unity Pro.



*En "Brazo_Robótico" se encuentran los Graficets de los Planos 1 y 2 (Principal y los auxiliares).

Figura 4. Bloques de función de SFC Chart Management

2.1.4 Contadores de pulsos (Plano 3)

Siguen la estructura de un bloque contador de Unity. Y permitirán llevar la cuenta de los pulsos que realice el brazo en cualquiera de sus 4 grados de libertad.

2.1.5 FIFO (Plano 4):

Este bloque resulta muy útil, pues es la herramienta mediante la cual el sistema decidirá de forma autónoma si la pieza que ha recorrido la línea indexada ha realizado el acabado superficial o el cilindrado y debe ser transportada por el brazo a la cinta de carga o a la caja de almacenaje respectivamente.

Este bloque de función sí se explicará con un grado superior de detalle puesto que es de especial relevancia para el proyecto.

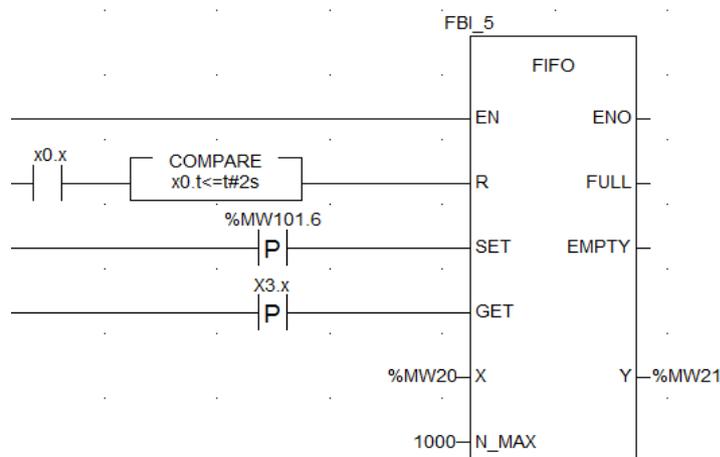


Figura 5. Bloque de función FIFO

- **R (Reset):** uniéndolo a un sensor o etapa se programará un reinicio de la memoria, cada vez que el Grafset pase por esa etapa o sensor. Si se activa correctamente se verá la salida 'EMPTY' en verde.
- **SET:** Cuando la línea del set sea cierta, el FIFO se guardará el dato que hay en la entrada 'X'. Para tomar un único dato, es aconsejable poner un flanco de subida del sensor que active el set.
- **GET:** Cuando se cumple esta condición, el FIFO muestra en la salida 'Y' el primer dato que ha guardado y todavía no ha mostrado. Al igual que en el caso anterior, es aconsejable trabajar con flancos de subida/bajada.
- **X:** el número que se guarda cuando se activa el set.
- **N_Max:** Número de datos máximos que va a almacenar, cuando $N \geq N_Max$ se activa la salida 'FULL' y no se toman más datos.

2.1.6 Contadores (Plano4):

Se crearán tres contadores más:

- **Contador de Piezas Auto:** cuando se alcance el número de piezas que se indicó por pantalla, se activará la salida.
- **Piezas terminadas:** Este contador nos dirá el número de piezas que se han terminado correctamente hasta el momento.
- **Piezas desechadas:** Este contador nos dirá el número de piezas que se han desechado hasta el momento.

2.2 Grafsets de la Línea indexada

2.2.1 Activación sensores (Plano 7):

Activa los sensores cuando el programa se pone en marcha.

2.2.2 Cinta de carga/Cinta 1 (Plano 5)

Cuando se pone en marcha, entra en una primera etapa en la que comprueba que el empujador 1 esté en su final de carrera trasera y si no lo está, es movidos hasta esa posición. Cuando se detecta una pieza al inicio de la cinta de carga, ésta se mueve hasta que la pieza llega al sensor situado frente al empujador 1. Si el empujador está en su posición trasera, la pieza avanza y es desplazada por el empujador hasta la segunda cinta. Durante este desplazamiento se activa también la cinta 2 para evitar que la pieza quede atascada.

Tras finalizar el movimiento vuelve al estado de espera de una nueva pieza.

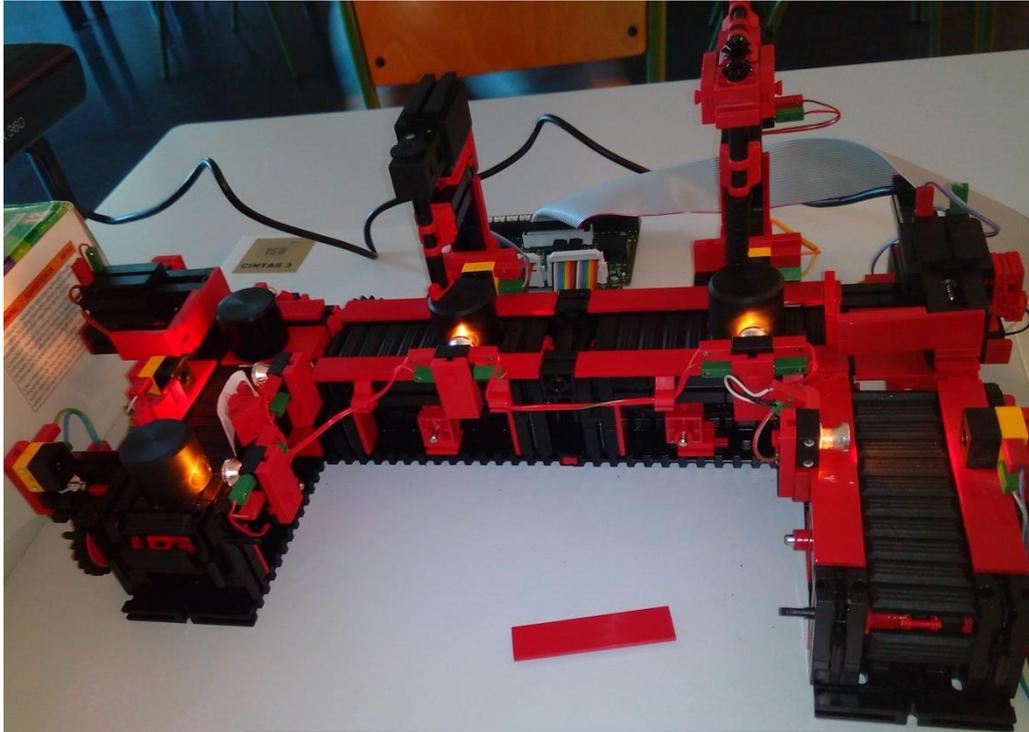


Figura 6. Línea indexada en marcha

2.2.3 Cinta de la fresadora/Cinta 2 (Plano 5)

Una vez se pone en marcha, espera a que el empujador 1 desplace una pieza y activa la cinta 2 hasta que la pieza se encuentra en la posición de la fresadora. Una vez aquí, diferencia entre los dos procesos que ha de realizar:

- **Realizar el proceso 1 o acabado superficial:** La fresadora actúa unos segundos, luego realiza un tiempo de espera y finalmente vuelve a actuar unos segundos más.
- **Realizar el proceso 2 o cilindrado:** La fresadora actúa ininterrumpidamente durante unos segundos.

Una vez realizado uno de los procesos y comprobado que la taladradora no tiene ninguna pieza, se activa la cinta 2 y la cinta 3 simultáneamente hasta que la pieza llega al sensor de la taladradora. Cuando esto ocurre, se considera que la cinta 2 vuelve a estar disponible de nuevo y se desplaza el empujador 1 hacia su posición trasera.

2.2.4 Cinta de la taladradora/Cinta 3 (Plano 6)

Una vez se pone en marcha, se activará la cinta 3 cuando la pieza haya realizado el proceso que le tocara en la cinta 2. La pieza llega a la taladradora, que se activa durante unos segundos. Cuando se detecte el empujador 2 en la posición trasera, la cinta 3 desplazará la pieza al empujador y éste la desplazará a su vez a la cinta 4 mientras activa la cinta 4 para evitar que se atasque la pieza.

2.2.5 Cinta de descarga/Cinta 4 (Plano 6)

Una vez se ponga en marcha, se realiza una primera comprobación de la posición del empujador 2. En caso de que se encuentre en la posición frontal, se moverá a la trasera.

Cuando el empujador 2 desplace una pieza a la cinta 4, ésta se pondrá en marcha hasta el sensor que indica final de la cinta. Se mantendrá el movimiento durante unos instantes para que la pieza quede colocada en el borde de la cinta y el acceso del brazo se vea facilitado.

2.2.6 Disponibilidad de cintas (Plano 6):

Este Grafcet es análogo al del brazo robótico con el mismo nombre.

Se precisa en los dos Grafcets debido al sistema empleado de comunicación entre autómatas, acceso a memoria compartida.

2.2.7 Modos de marcha (Plano 6):

En este caso sólo se distingue entre el modo automático y el modo paso a paso, desde el que se podrá manipular ciertos elementos de las cintas. En el modo automático el funcionamiento es ininterrumpido.

2.2.8 Modos de alarma (Plano 7):

En este caso sólo se consideran 2 modos:

- **Inicializar**
- **Congelar**

El funcionamiento es análogo al descrito en el modo de alarma del brazo robótico, empleando los mismos bloques de función.

2.2.9 Fifo (Plano 7):

Análogamente al Fifo del brazo robótico, este bloque permite que el sistema decida de forma autónoma si la pieza que recorre la línea indexada ha de realizar el acabado superficial o el cilindrado.

2.2.10 Implementación del PID (Plano 8):

Por razones de memoria, el PID se ha dispuesto en el Grafcet de las cintas y no en el del Brazo Robótico como inicialmente se había previsto. Esto no altera de ningún modo el funcionamiento previsto.

Con los parámetros calculados en el anexo de diseño, se ha optado por utilizar un bloque de función presente en Unity y que haría las funciones de un regulador PID²².

El regulador que se ha calculado es para parámetros con sus unidades en grados para las entradas y la acción de control se calculará en porcentaje del 0 al 100%.

Además los datos deben ser números reales (Particularidad del bloque de función PI_B)

Sin embargo, los sensores dan la lectura de las temperaturas en milivoltios que es la unidad en la que se almacenan los datos en los espacios de memoria del autómata como se ha explicado en la descripción del equipo de la memoria.

Consecuentemente, los actuadores precisan la acción en milivoltios.

Además, en los espacios de memoria del autómata los datos se guardan en números enteros, con lo cual habrá que realizar conversiones de enteros a reales y viceversa.

²² El bloque seleccionado en Unity será un bloque de función PI_B que hará las funciones de un regulador PID. Dicho bloque se ha programado siguiendo las indicaciones del documento *Anexo 00 Prácticas de TAU con el PLC TSX-Premium.pdf*; [13] (Simarro, Anexos asignatura Lac. ETSII, 2014-2015) y [4] (Herrero, Apuntes de la asignatura TAU, 2013-2014)

2.2.10.1 Entradas:

Según la [Figura 16 de la memoria](#) la relación entre milivoltios y grados se calcula según una sencilla regla de tres:

$$\begin{aligned} 70^{\circ}\text{C} &= 10000 \text{ mV} \\ -34^{\circ}\text{C} &= 2000 \text{ mV} \end{aligned}$$

$$T [^{\circ}\text{C}] = 0.013 \cdot T[\text{mV}] - 60 = \frac{1}{77} \cdot T[\text{mV}] - 60$$

La lectura del sensor se almacena en la primera entrada analógica del autómata.

Se realiza la operación de división seguida de la diferencia, ambas mediante bloques de función como se observa en el *plano 8*.

Tras el cálculo, se ha de convertir el número entero a real para que el bloque PI_B pueda operar con él sin problemas.

2.2.10.2 Salidas:

Según la [Figura 17 de la memoria](#) la relación entre el porcentaje de actuación y milivoltios se calcula igualmente según una sencilla regla de tres:

$$\begin{aligned} 100\% &= 2000 \text{ mV} \\ 0\% &= 0 \text{ mV} \end{aligned}$$

$$U [\text{mV}] = 20 \cdot U[\%]$$

Esta operación se realiza mediante los bloques de función que se observan en el plano 8.

Antes del cálculo, se debe convertir la acción calculada por el regulador PI_B a un número entero para poder hacer la multiplicación y almacenarlo transmitirlo después a la salida analógica del autómata.

2.2.10.3 Parámetros del regulador

Se crea una variable (Para_PI) con la estructura que define el bloque de función PI_B.

Accedemos a esta variable desde el apartado “Variables Elementales” del Unity Pro como se muestra a continuación:

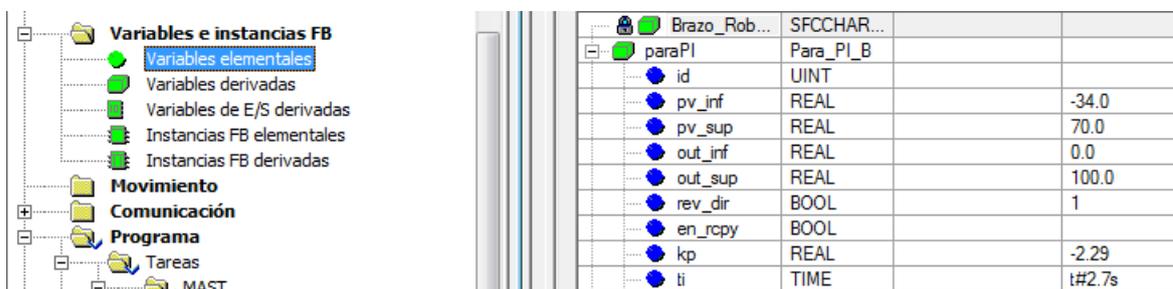


Figura 7. Parámetros del PID

Desde este panel podremos fijar los parámetros calculados en el anexo de diseño.

2.2.11 Comunicación

Configuración IP Mensajes Exploración de E/S Datos globales SNMP Servidor de dirección Ancho de banda												
Áreas %Mv del maestro												
Ref. de lectura			Ref. de escritura			<input type="checkbox"/> Control del dispositivo: desde			hasta			Paso de velocidad de repetición: 10
Desde 100			hasta 101			Desde 150			hasta 152			
Periféricos explorados												
	Dirección IP	ID de unidad	Timeout de perturbación (ms)	Velocidad de repetición (ms)	LEER Objeto maestro	LEER índice de esclavo	LEER longitud	Último valor (entrada)	ESCRIBIR Objeto maestro	ESCRIBIR índice de esclavo	ESCRIBIR longitud	Descripción
1	158.42.206.12	255	1500	60	%Mv100	100	1	Mantener último	%Mv150	101	1	disponibilidad de cintas
2	158.42.206.12	255	1500	60	%Mv101	0	1	Mantener último	%Mv151	102	1	tipo de proceso a seguir
3	158.42.206.12	255	1500	20	%Mv102	8	0	Mantener último	%Mv152	10	1	Alarmas
4												

Figura 8. Ethernet del autómat

La lectura y la escritura se realizan desde el maestro que en el caso que ocupa es el brazo robótico.

Se leerá:

- Si hay pieza en la cinta de carga o en la cinta de descarga
- Los sensores de la línea indexada, de los que concretamente interesan el sensor de la cinta de carga y el sensor del empujador 1.

Se escribirá:

- Si el brazo robótico recoge la pieza de la cinta de descarga y si el brazo transporta la pieza a la cinta de carga
- Tipo de proceso que ha de seguir la siguiente pieza que entre en la cadena
- Los botones de alarmas así como si la pieza es correcta, hay que cambiarla o si es defectuosa

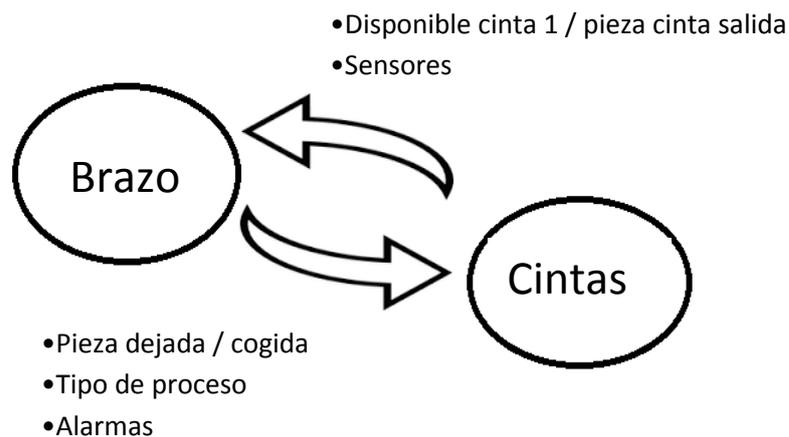


Figura 9. Diagrama de los datos compartidos

3 LabVIEW

El programa ha sido estructurado de forma secuencial en tres bloques:

- Bloque de inicialización de variables
- Bloque de inicio de sesión
- Bloque del programa principal

3.1 Bloque de inicialización de variables

Este bloque es muy sencillo y su función reside en desconectar el usuario que hubiera conectado de la anterior sesión e inicializar una serie de variables que serán importantes posteriormente, en el bloque de inicio de sesión.

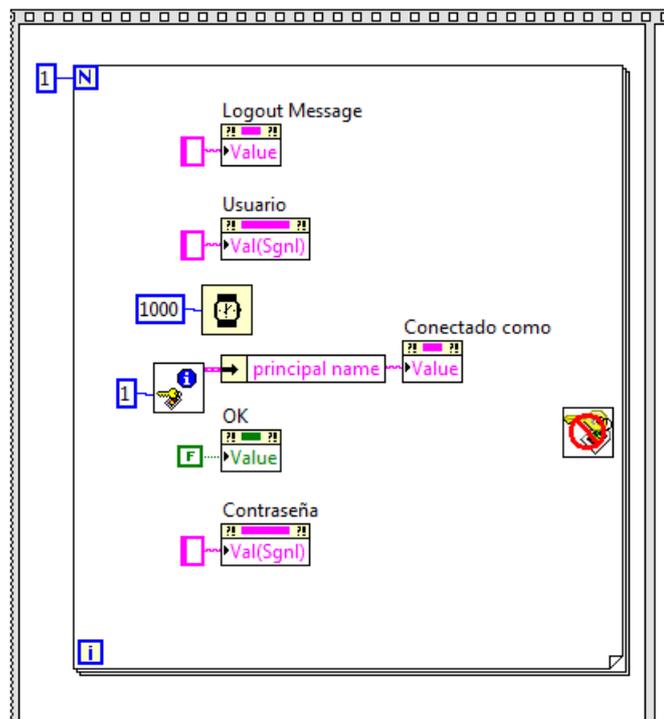


Figura 10. Diagrama de bloques.
Inicialización de variables

3.2 Bloque de inicio de sesión

Como su propio nombre indica, la función de este bloque será que el usuario se identifique antes de empezar a ejecutar el programa.

Cuando se inicia el programa saldrá un cuadro de diálogo pidiendo que el usuario se identifique.

Si la identificación es correcta, pasa al siguiente bloque, el principal, que es realmente el programa para la monitorización del proceso.

En caso de ser incorrecta, el cuadro de diálogo volverá a aparecer hasta que la identificación sea correcta.

Huelga decir que mientras el usuario no está identificado, no se puede acceder a ninguna de las pestañas de la pantalla.

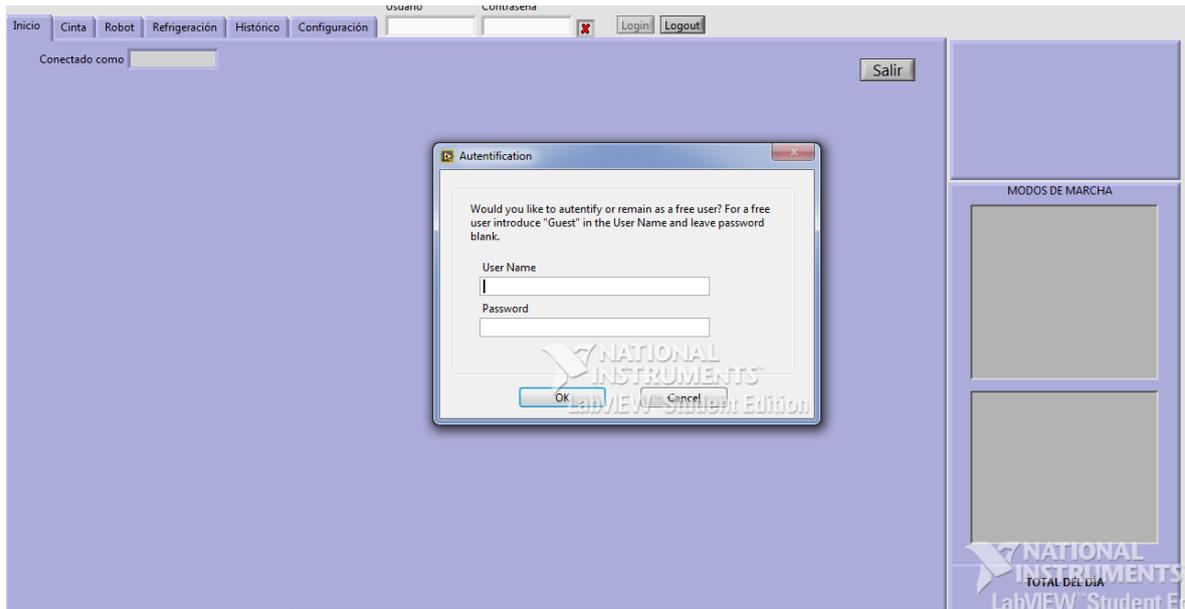


Figura 11. Ventana emergente de inicio de sesión

El código implementado será el que se muestra en la Figura 12.

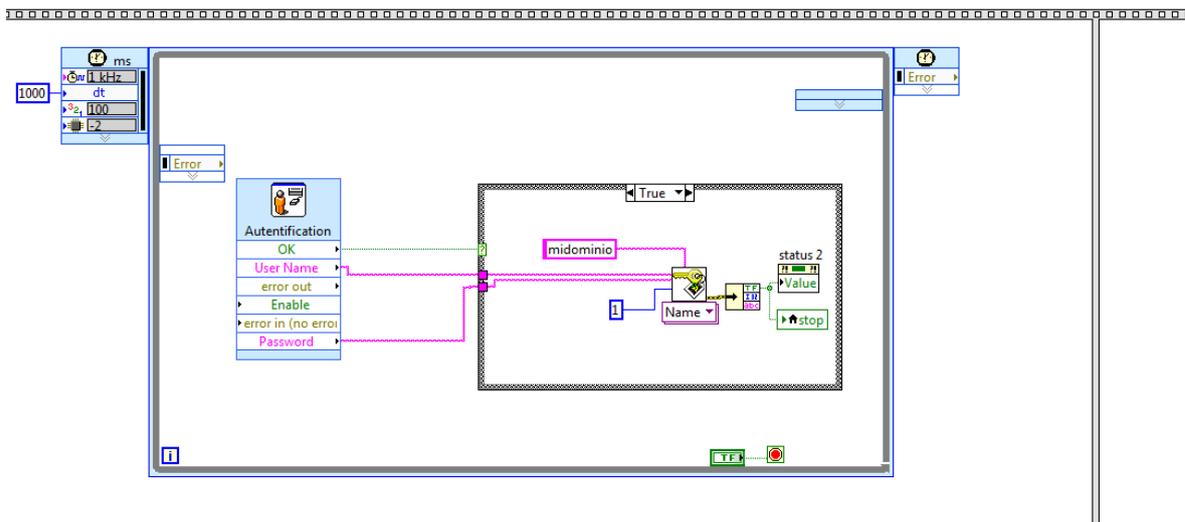


Figura 12. Código ventana emergente de inicio de sesión

3.3 Bloque del programa principal

Está estructurado a su vez en diez bucles ejecutándose simultáneamente.

Se trata de una estructura compleja, por lo que se describirá brevemente la función de cada uno de los bucles para la comprensión del funcionamiento general. Dada la limitación existente en la longitud del presente documento, no se podrá entrar en detalles específicos sobre que se realiza en cada elemento de los bucles, por ello sólo se presentará la función desarrollada por el conjunto de los elementos.

3.3.1 Bucle principal

En este bucle se desarrolla principalmente la transmisión de datos desde el proceso de las maquetas hasta la pantalla.

Mediante este bucle se puede mostrar por pantalla dónde se encuentra la pieza en todo momento. Esta función se realiza mediante las Bound Variables²³, que son las portadoras de los datos desde el programa Unity Pro al LabVIEW y por medio de las cuales se activan los botones de la pantalla del LabVIEW.



Bound Variables

En función del valor desencriptado de las bound variables se sabe que botón debe estar activo o no y consecuentemente su imagen en el panel frontal estará activa o no. Un ejemplo se muestra en la siguiente *Figura 13*.

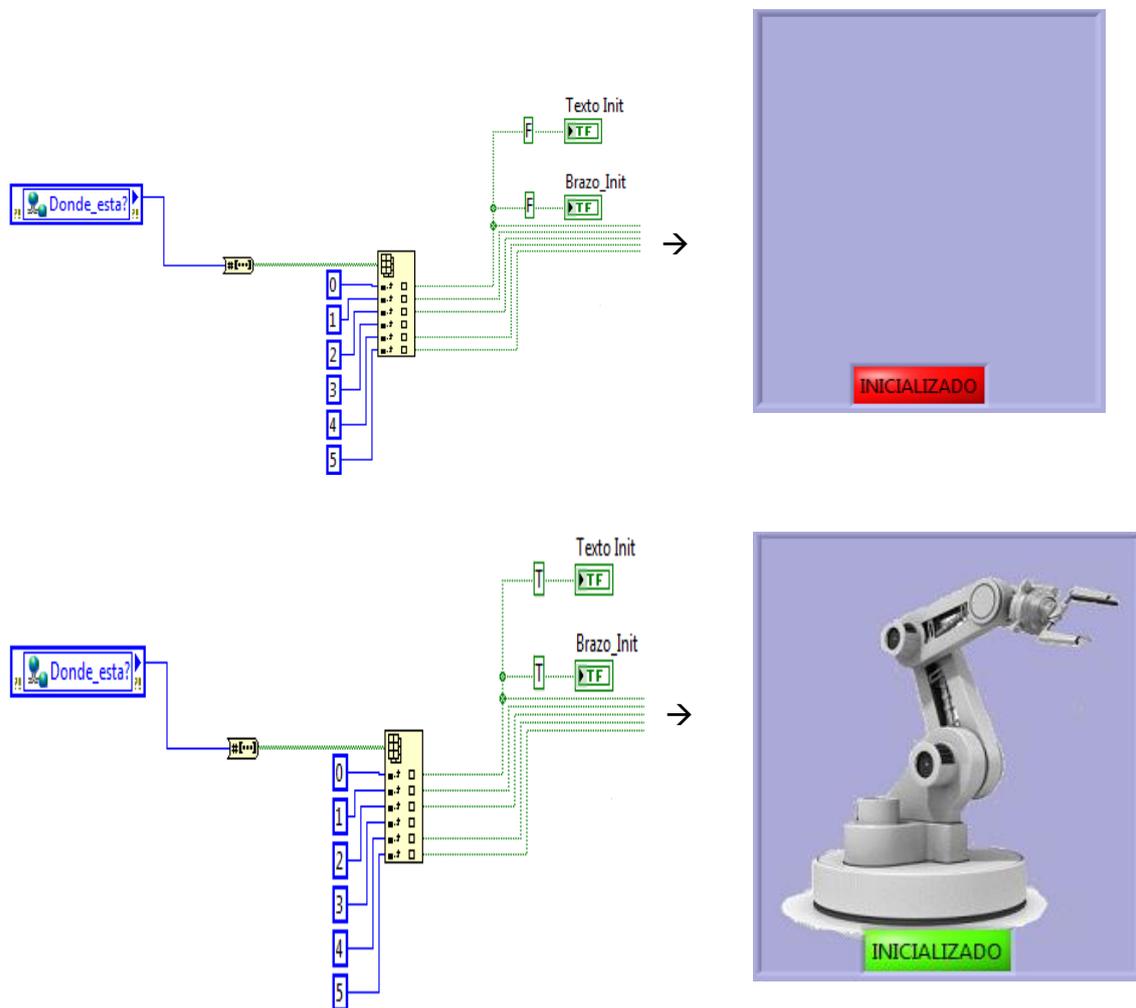


Figura 13. Código y aspecto en la pantalla

²³ Las Bound Variables y su funcionamiento se explican en el documento de "Seminario Automatización de Procesos"; [14] (Simarro, Apuntes de la asignatura LAC. ETSII, 2014-2015)

A través de una Bound Variable de este bucle (ROB50) se transmite el estado de los modos de marcha (Figura 14).

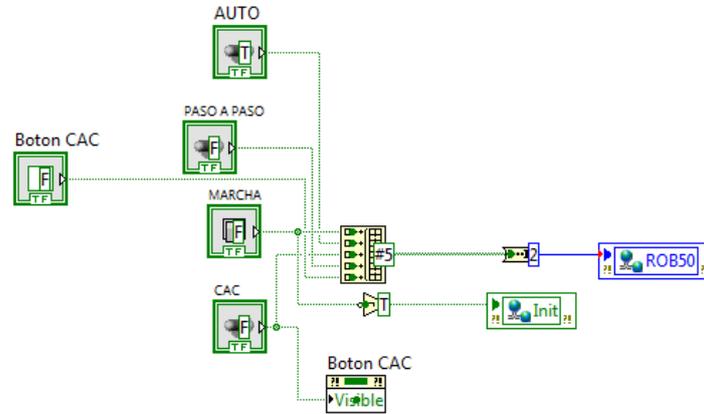


Figura 14. Código de los modos de marcha

De igual forma se dibuja un gráfico con las piezas terminadas y las desechadas (Figura 15).

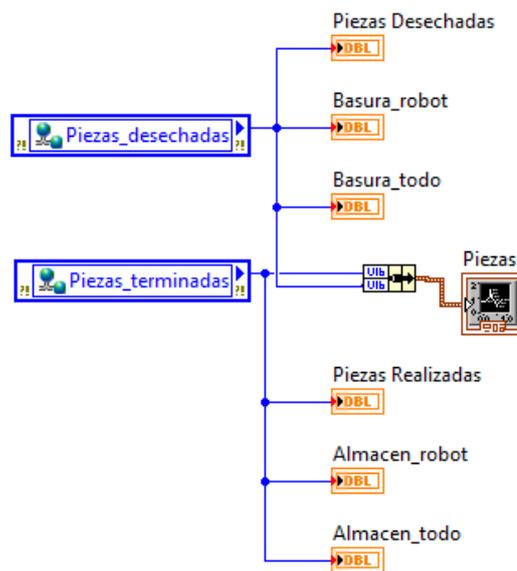


Figura 15. Código para representación gráfica

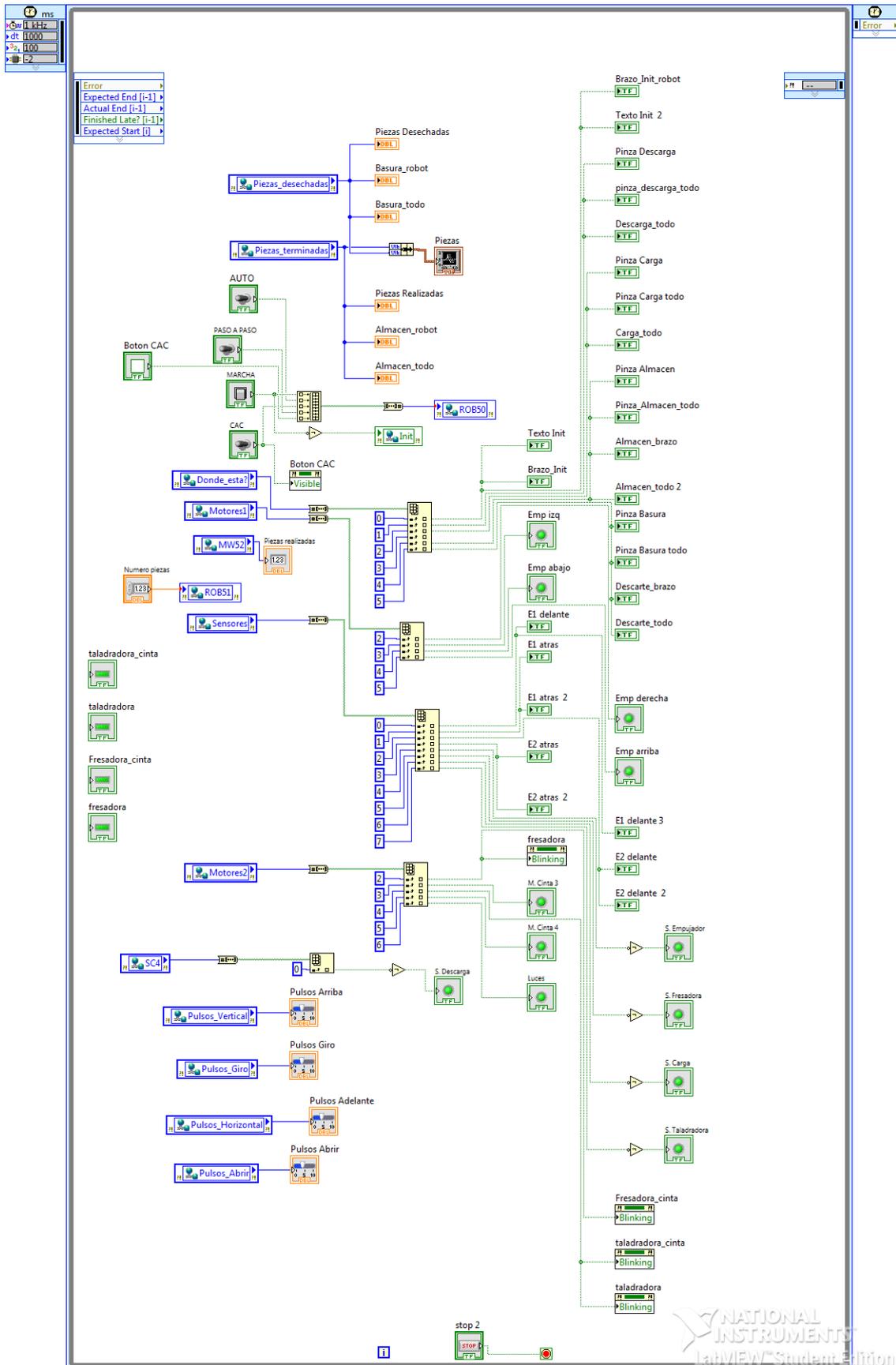


Figura 16. Aspecto del bucle principal completo

3.3.2 Bucle de movimiento manual:

Cuando el modo de marcha seleccionado sea el paso a paso se podrá mover cada uno de los elementos de la línea indexada y del brazo robótico de manera independiente.

La restricción de que sólo se pueda en el modo paso a paso se realizará con otro bucle que se mostrará posteriormente.

El bucle que se está describiendo permite que pulsando el botón de la pantalla correspondiente se pueda mover el elemento que se desea.

Un ejemplo para hacerlo más visual se muestra en la siguiente imagen:

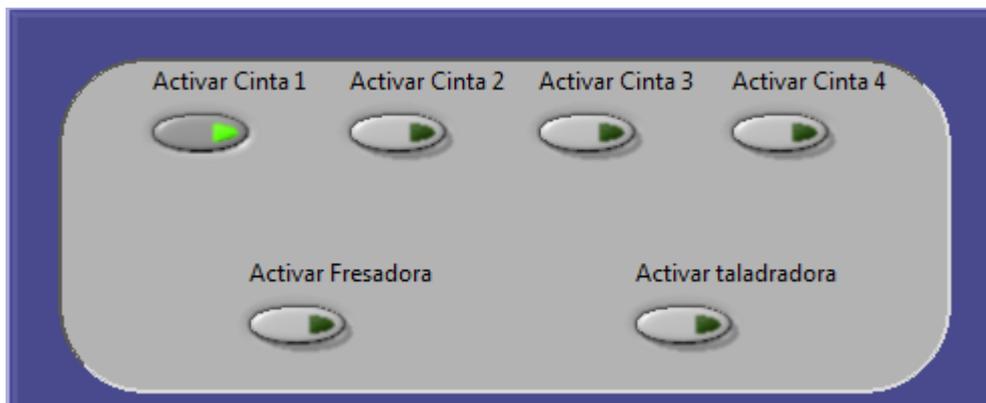


Figura 17. Botones de movimiento manual en pantalla

Al pulsar el botón de la Figura 17 se pretende que la cinta 1 se mueva y ello se consigue con el bucle de movimiento manual que se muestra en la Figura 18.

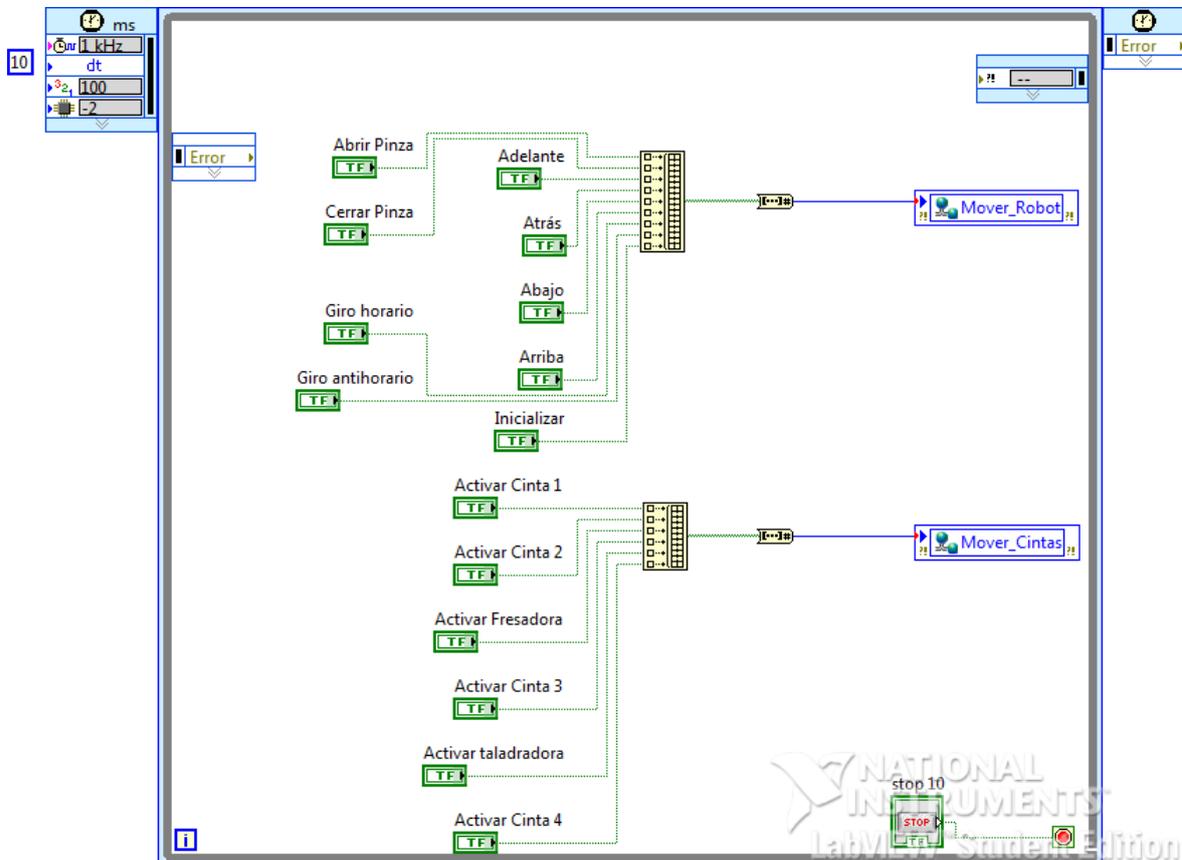


Figura 18. Código del bucle de movimiento manual

Este bucle se ejecuta más rápido (10 ms) para que los datos se transmitan más rápido y el movimiento se realice de forma fluida.

3.3.3 Bucle de movimiento fluido de cintas 1 y 4

Mediante el bucle que se describe ahora se consigue que en la pantalla del operario se vea cómo va avanzando la pieza por la cinta, de forma que aporta una ayuda visual al operario de por dónde está la pieza en cada instante.

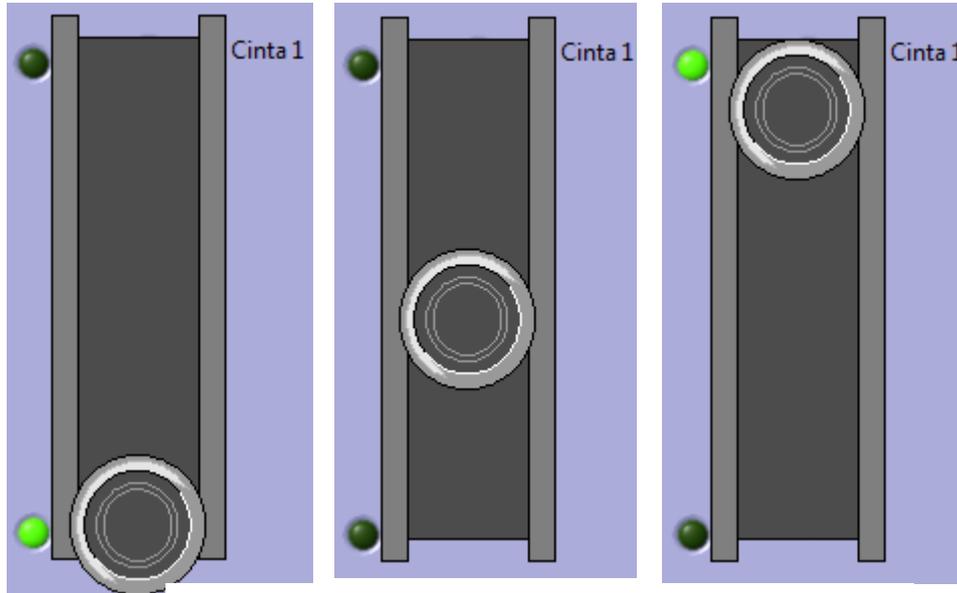


Figura 19. Desplazamiento de pieza por cinta en pantalla

Este efecto que visualmente aporta una valiosa ayuda se consigue con el código implementado²⁴ en la Figura 20.

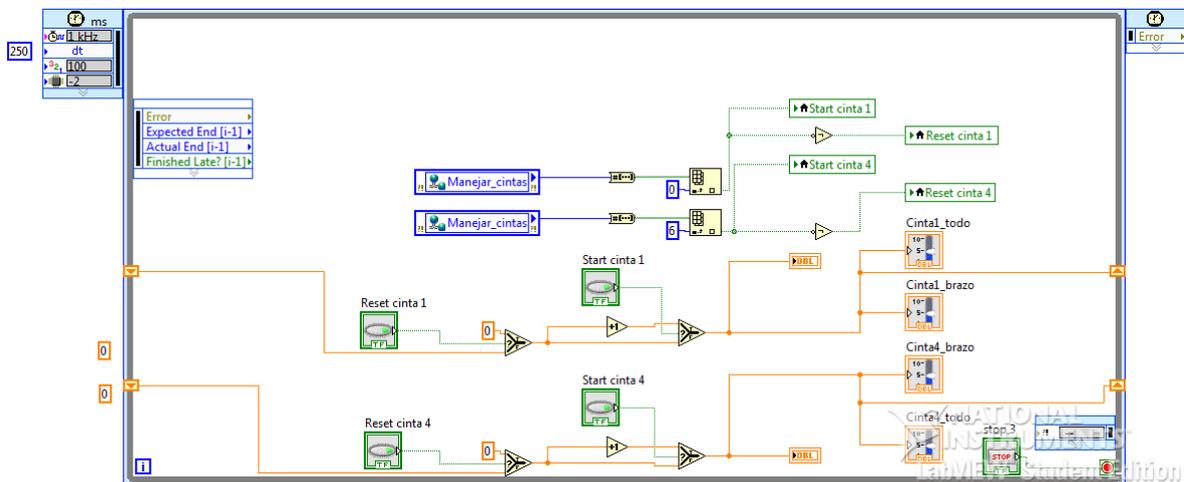


Figura 20. Código del bucle de movimiento de las cintas 1 y 4

²⁴ Para la realización de estos bloques se ha tomado como referencia el vídeo del siguiente enlace: <https://www.youtube.com/watch?v=Ay2182-Agp8>, donde se explica detalladamente cómo realizar este tipo de movimiento dinámico. [2] (Chacón, 2014)

3.3.4 Bucle de movimiento de las cintas 2 y 3

De forma análoga al anterior, se ha realizado un bucle para el movimiento de las otras dos cintas con la única variación de que se ha incorporado un paso extra por el cual la pieza se queda parada en las unidades de mecanizado mientras éstas están actuando.

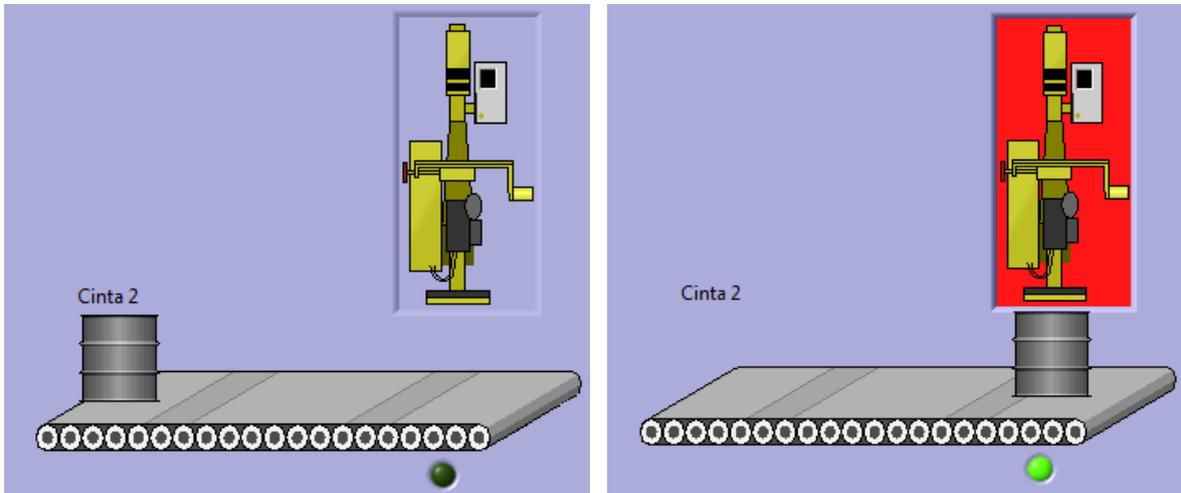


Figura 21. Fresadora actuando en pantalla

Por este motivo es por el que hay que crear dos bucles diferenciados para las cintas de carga y descarga y para las que llevan incorporadas las unidades de mecanizado.

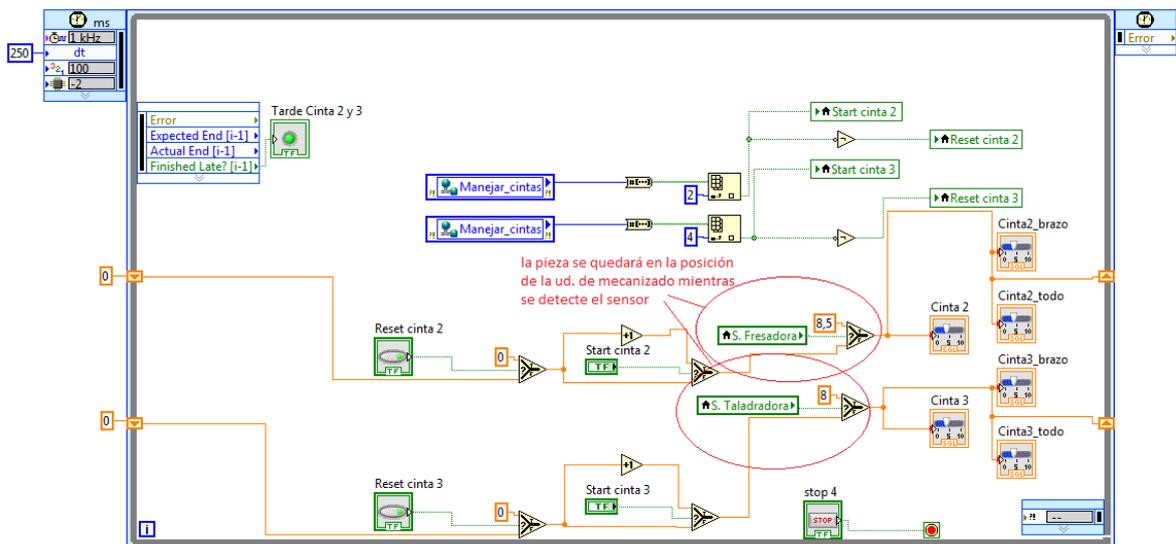


Figura 22. Código del bucle de movimiento de cinta 2 y 3

3.3.5 Bucle de Seguridad

El siguiente bucle es imprescindible, ya que es el que maneja los botones de seguridad (Marcha, Stop, Inspección).

La respuesta ante un cambio en cualquiera de estos botones ha de ser casi instantánea. Por ello se ha escogido un tiempo de refresco muy bajo (50ms).

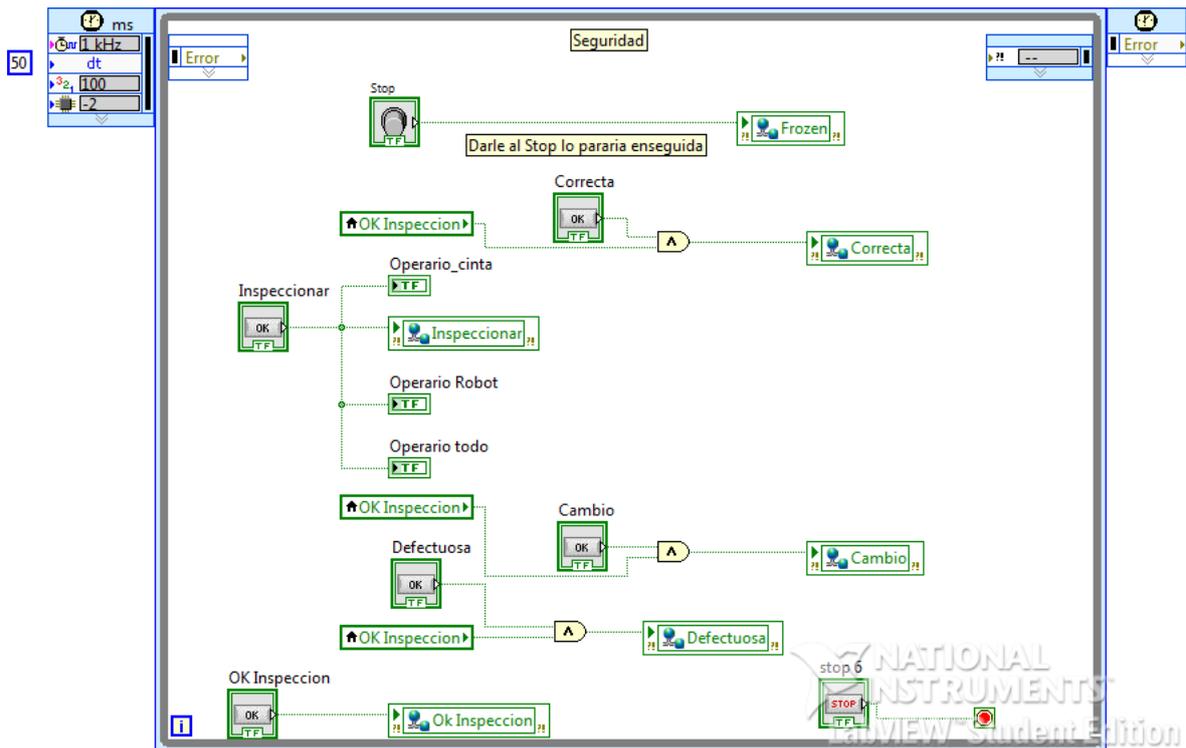


Figura 23. Código del bucle de seguridad

3.3.6 Bucle Peltier

Mediante este bucle se permite variar la referencia, así como la acción manual en el caso de que se esté en modo manual de la Peltier (sino estaría el control “disabled and grayed”²⁵, deshabilitado). En este bucle se muestran gráficamente parámetros interesantes de la célula Peltier y el resultado se observa en *Figura 24*.

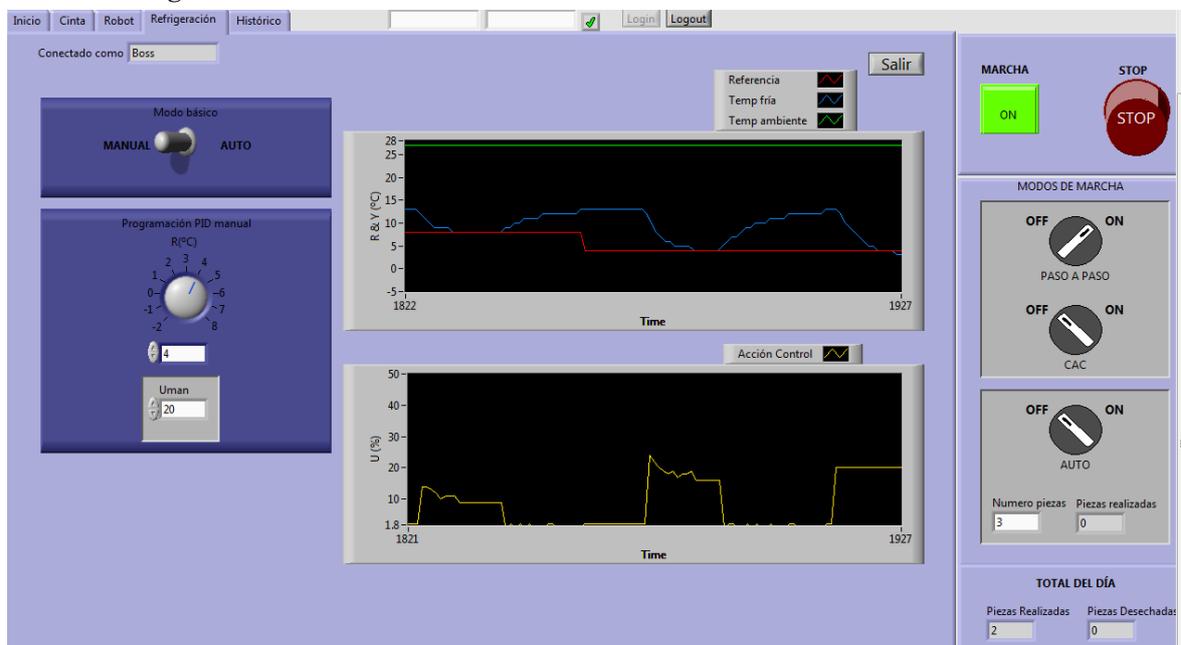


Figura 24. Controles de la célula Peltier en pantalla

²⁵ Los Property Nodes son elementos existentes en LabVIEW con una gran utilidad. Para más información de cualquier elemento de LabVIEW pulsar conjuntamente “Ctrl + H” para abrir el panel de ayuda y seguidamente mover el cursor sobre el elemento del que se requiera información. [7] (Instruments, Getting started with LabVIEW, 2010)

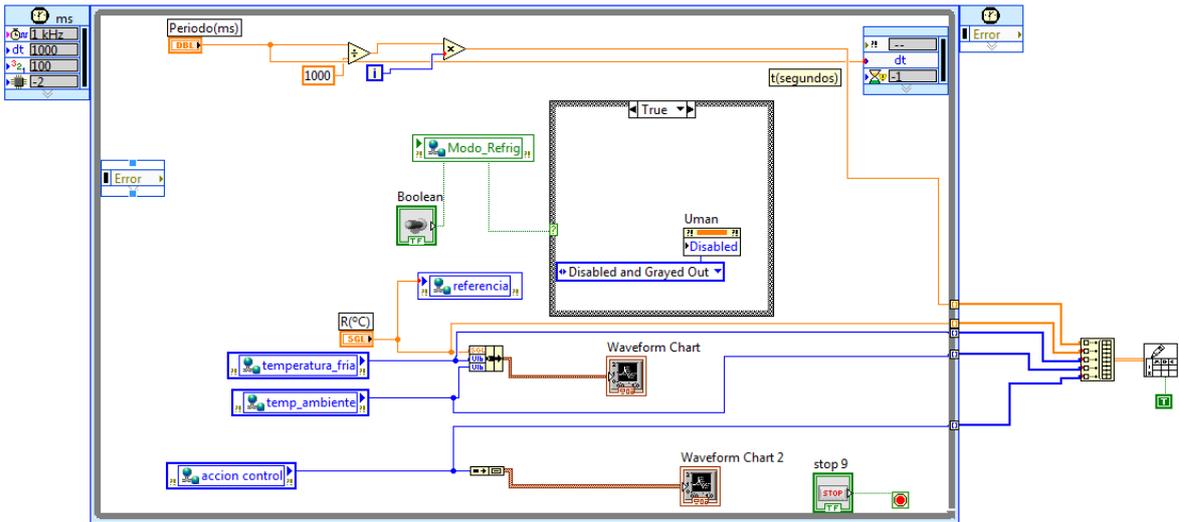


Figura 25. Código del bucle de la Peltier

La función del bloque “Case” es la de habilitar/deshabilitar la acción manual dependiendo de si el modo manual está activo o no respectivamente.

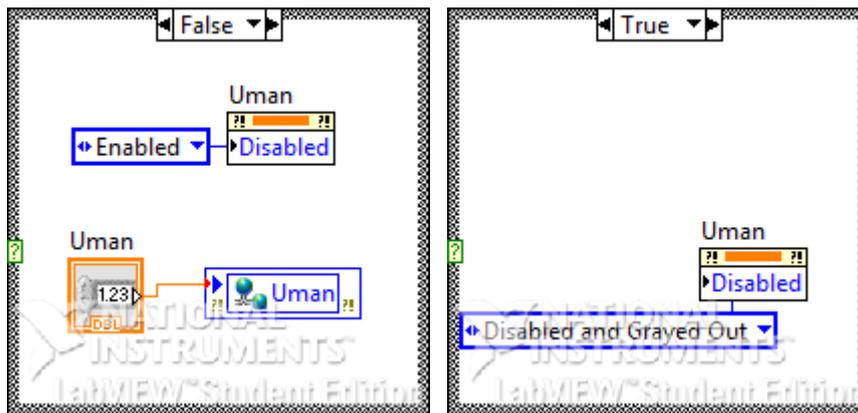


Figura 26. Bloque Case del bucle Peltier

3.3.7 Bucle Inspección

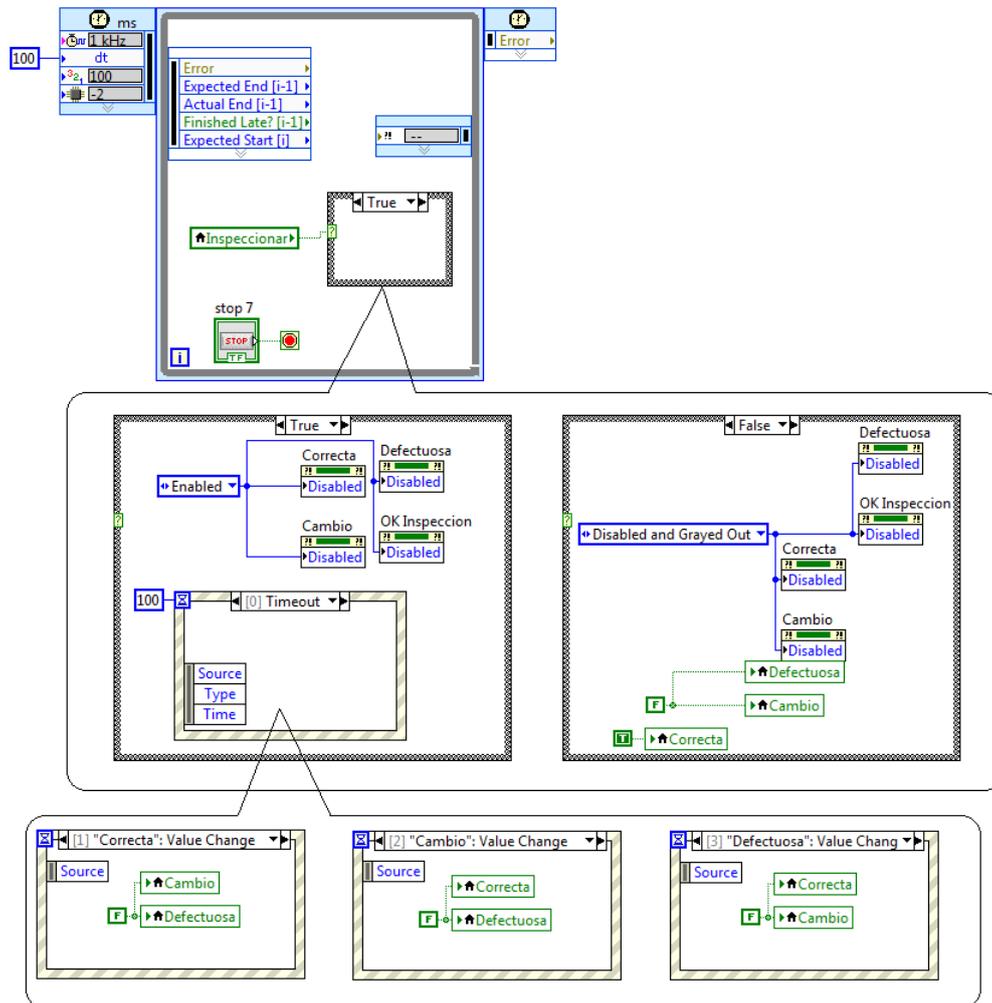


Figura 27. Código del bucle Inspección

Este bucle se emplea para habilitar o deshabilitar los botones de correspondientes a la inspección si ésta está activa o no respectivamente. Se muestra una imagen en la Figura 28.



Figura 28. Controles de inspección en pantalla

La razón por la que este bucle es necesario es puramente funcional.

Dado que se persigue un diseño de pantalla intuitivo, ésta es la mejor forma de implementarlo, ya que si no está seleccionada la inspección, no habrá manera de equivocarse y marcar una pieza como incorrecta.

Esto se consigue con el “Case” mostrado en la *Figura 27*.

Con el bucle “Event” se consigue que una pieza sólo pueda ser marcada como correcta, cambio o defectuosa, pero nunca las dos a la vez. Esto suma a la sencillez en el manejo del programa.

Además, el programa no realiza la acción correspondiente al tipo de pieza seleccionado hasta que se pulse el botón “OK”.

De esta forma también se evitan selecciones accidentales.

3.3.8 Bucle Paso a Paso

Previamente se ha señalado que las funciones de movimiento manual sólo deben estar activas cuando el modo paso a paso esté activo.

Eso se consigue mediante este bucle, que habilita o deshabilita los botones de movimiento manual dependiendo de si está o no seleccionado el modo de marcha “Paso a Paso”.

Además, en el caso de que el brazo esté al máximo de movimiento en cualquiera de sus grados de libertad (por ejemplo, en la posición superior), en el bucle se deshabilita la acción que haría que el brazo saliera de su posición tope (en el ejemplo antes mencionado se deshabilitará la acción “Arriba”).

Esta última función mencionada va en la línea de conseguir una pantalla intuitiva y sencilla.

El usuario nunca será capaz de mover el Brazo Robot más allá de los topes que éste tiene marcados en sus cuatro grados de libertad.

Se previene por tanto que un despiste ocasional pueda causar graves daños en el material.

En las imágenes siguientes se proporciona una ayuda visual para la comprensión de la función que realiza este bucle.

De forma visual se puede observar lo anteriormente expuesto en las *Figuras 29 y 30*.

El código implementado para realizar estas funciones se muestra en la *Figura 31*.

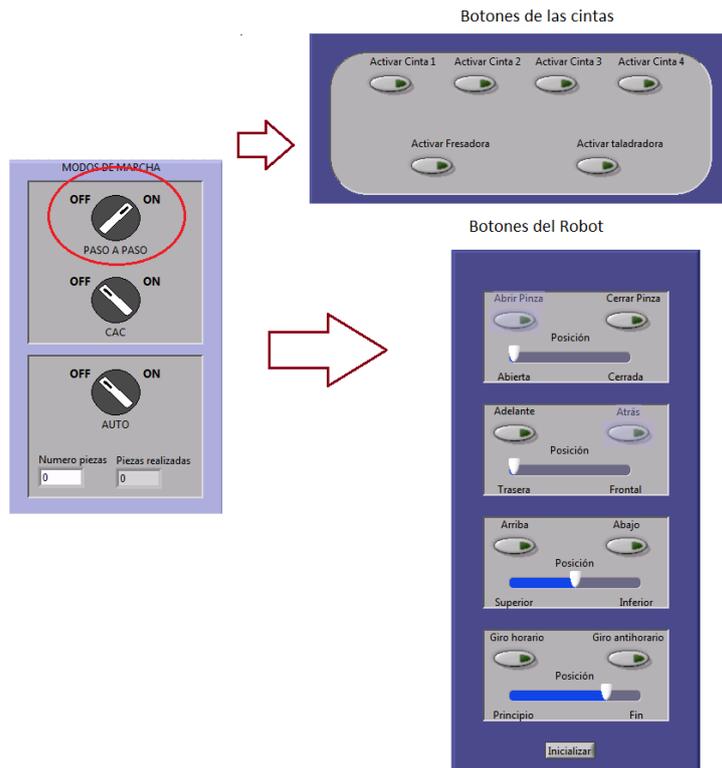


Figura 29. Modo de marcha "Paso a Paso" activo

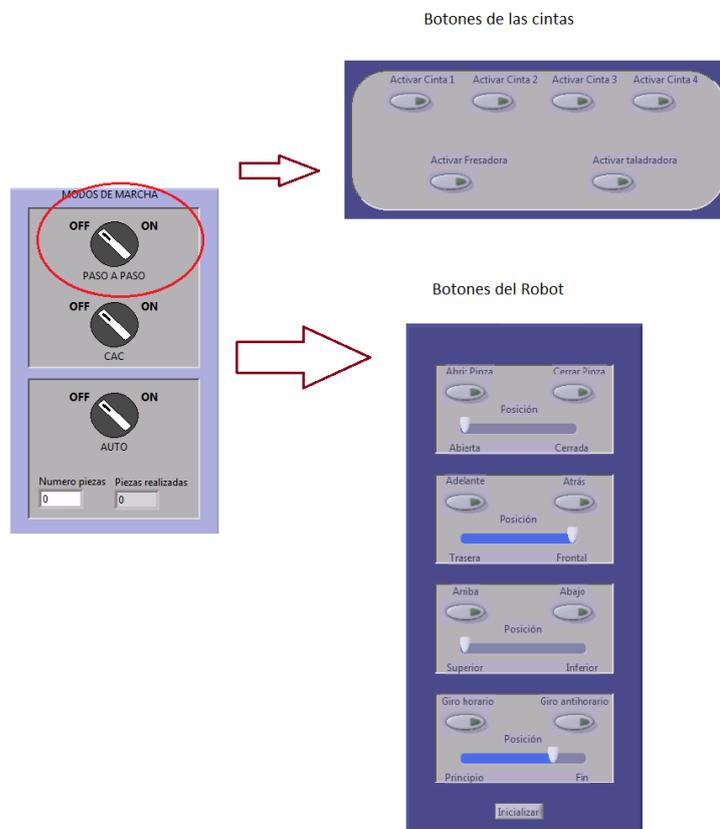


Figura 30. Modo de marcha "Paso a Paso" no activo

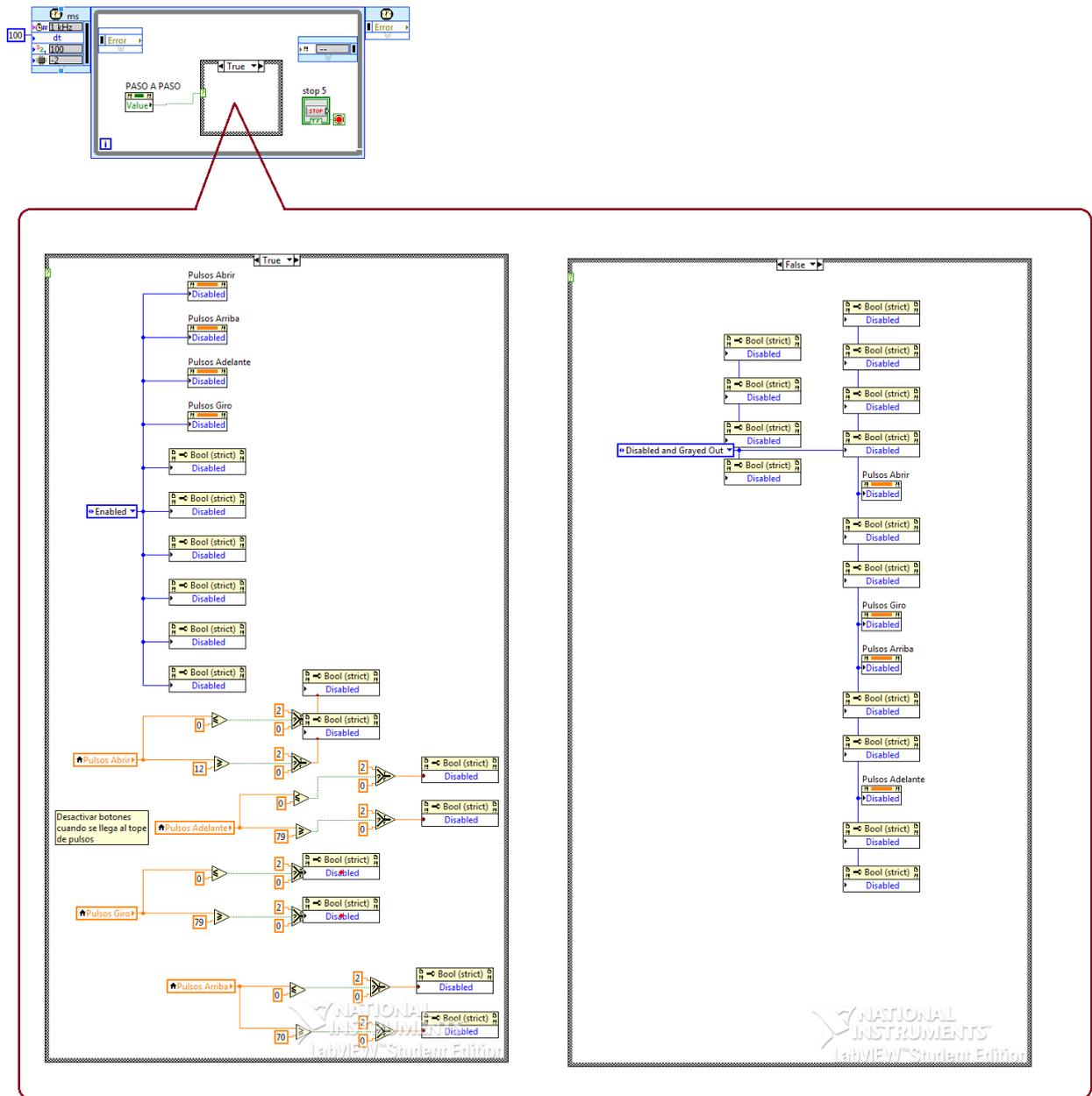


Figura 31. Código del bucle Paso a Paso

3.3.9 Bucle Desactivación de modos de marcha

Con el fin en mente de hacer la aplicación todavía más intuitiva, se ha diseñado este bucle. La función que realiza es, de manera secuencial, comprobar si ha habido algún cambio en el valor de algún modo de marcha (si se ha pulsado el botón de automático por ejemplo) y verificar si había hasta el momento otro modo de marcha seleccionado. Si así fuere, se abre un cuadro de diálogo en la pantalla pidiéndole al usuario si quiere validar el cambio que ha efectuado. Si el usuario quiere cambiar de modo de marcha, se desactivará el modo de marcha que hubiera seleccionado hasta el momento. De esta forma se evita que pueda haber dos modos de marcha funcionando de manera simultánea, lo que podría ocasionar problemas de funcionamiento. Contribuye a crear el clima de sencillez en la aplicación que se ha buscado desde el inicio. Un error del usuario pulsando sin querer otro modo de marcha al seleccionado inicialmente es fácilmente subsanable mediante el sistema de cuadro de diálogo que ofrece la opción de cancelar el cambio que se ha realizado por error.

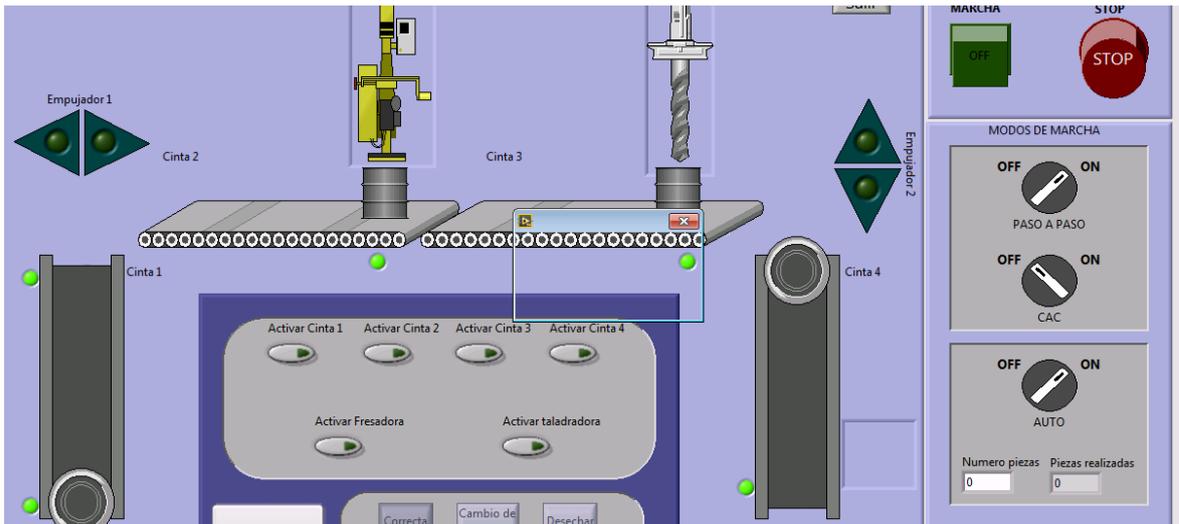


Figura 32. Cambio de modo de marcha

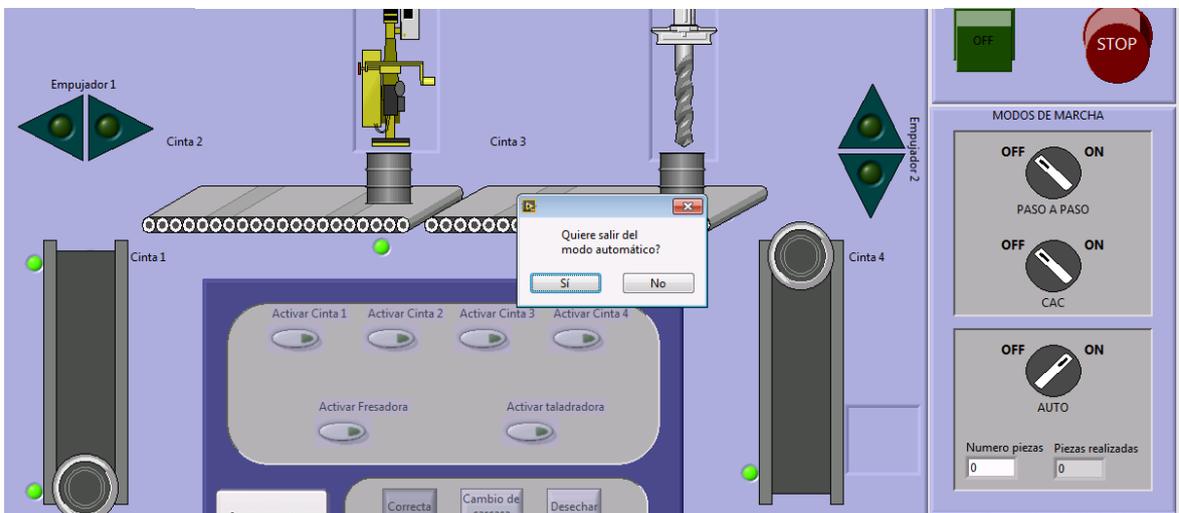


Figura 33. Confirmación del cambio realizado

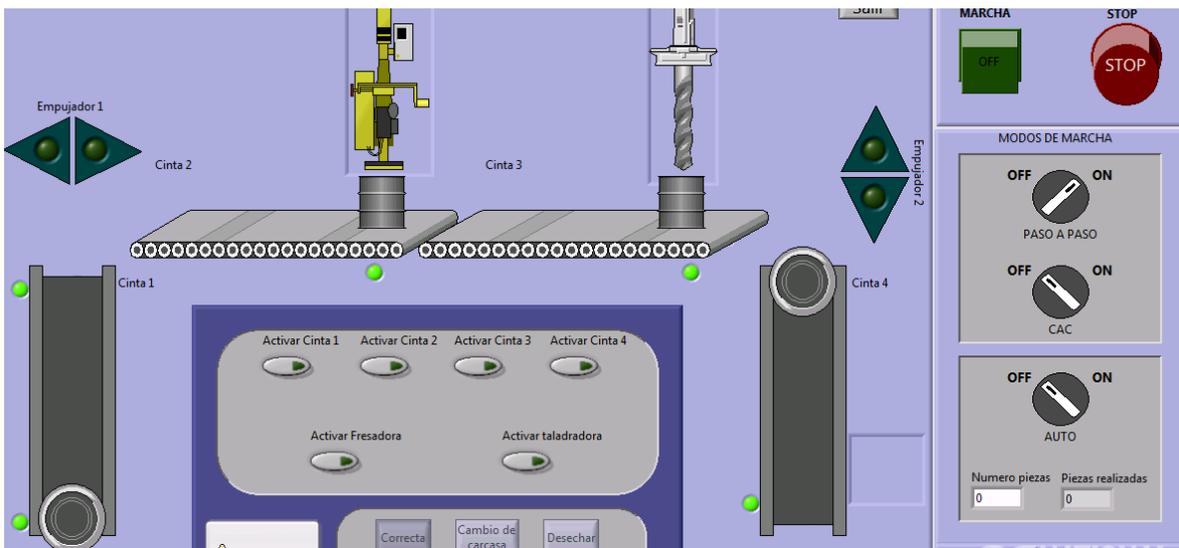


Figura 34. Nuevo modo de marcha activo

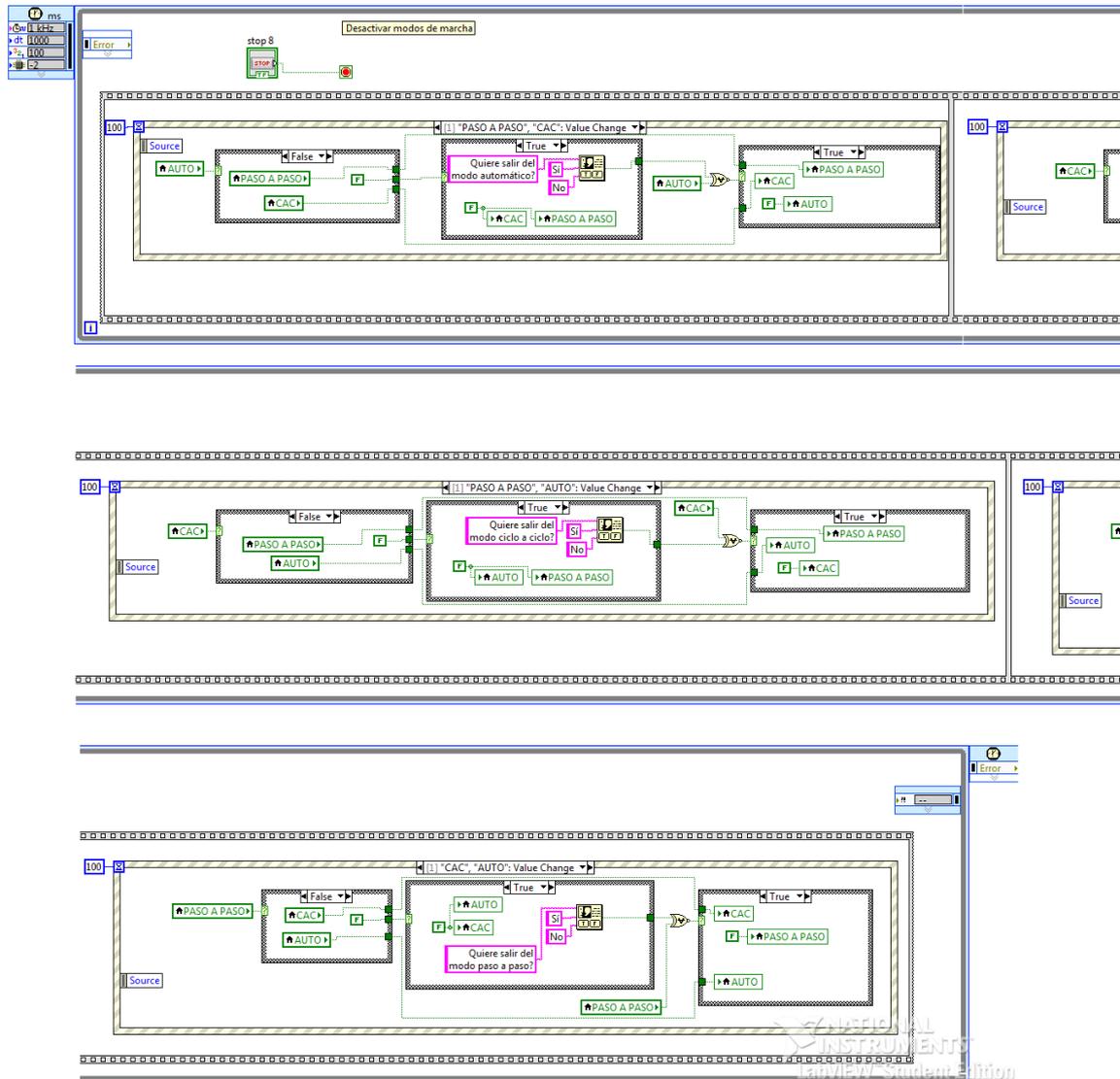


Figura 36. Código completo del bucle de desactivación de modos de marcha

*Se ha cortado la imagen para adaptarla a las medidas de las que se dispone.

3.3.10 Bucle de control de usuarios

Mediante este bucle lo que se pretende es gestionar un dominio de usuarios. Esto se logra gracias a las funciones disponibles en el DSC Module de LabVIEW. Estas funciones permitirán:

- Conectarse a un usuario al dominio previamente creado.
- Desconectarse cuando finalice la sesión.
- Saber qué usuario se encuentra conectado en cada momento.

La gestión de las cuentas de usuario se logra con el código que se muestra en la *Figura 37*.

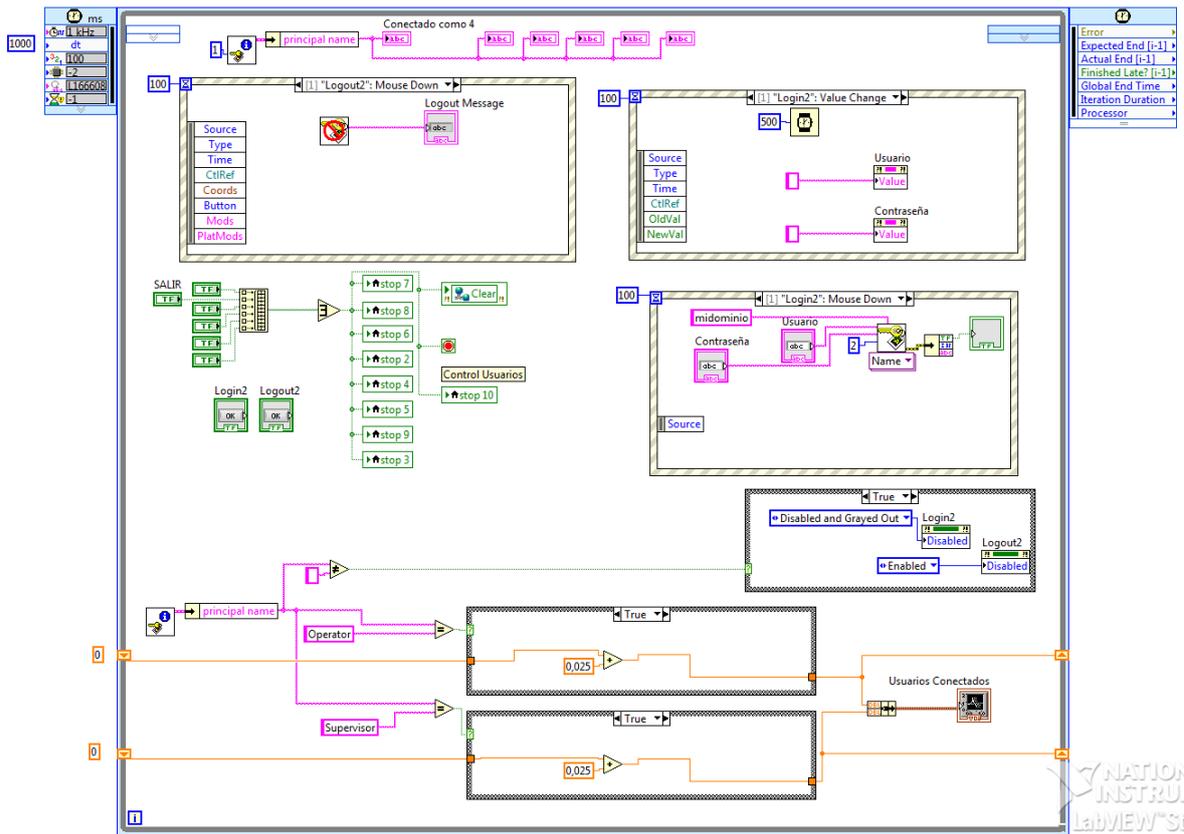


Figura 37. Código bucle de control de usuarios



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº7: Manual de usuario

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenidos

1 Montaje de las maquetas	113
2 Montaje de la Peltier	113
3 Requerimientos.....	114
3.1 Hardware.....	114
3.1.1 Unity Pro	114
3.1.2 LabVIEW	114
3.2 Software	114
4 Esquema de inicio de aplicaciones	115
5 Iniciar el programa en Unity.....	115
6 Iniciar el servidor	116
7 Iniciar la pantalla SCADA	117
7.1 Pestañas e inicio de sesión durante el programa.....	117
7.2 Pestaña Lateral.....	118
7.2.1 Observaciones a considerar de la pestaña lateral	119
7.3 Pestaña Inicio.....	120
7.4 Pestaña de cintas	121
7.4.1 Observaciones a considerar en la pestaña Cintas	122
7.5 Pestaña Robot	122
7.5.1 Observaciones a considerar en la pestaña Robot.....	123
7.6 Pestaña Refrigeración	124
7.6.1 Observaciones de la pestaña Refrigeración.....	124
7.7 Pestaña Histórico	125
7.7.1 Observaciones a considerar en la pestaña Histórico.....	125
8 Almacenamiento de datos.....	125

Índice de figuras

Figura 1. Montaje de las maquetas en laboratorio.....	113
Figura 2. Imágenes del montaje de la Peltier en laboratorio	113
Figura 3. Secuencia de inicio de programas.....	115
Figura 4. Barra de herramientas del Unity Pro	115
Figura 5. Establecer dirección del autómeta	116
Figura 6. Establecer dirección del autómeta en el servidor	116
Figura 7. Botón Ejecutar	117
Figura 8. Ventana emergente de identificación	117
Figura 9. Pestañas y control de usuario	117
Figura 10. Elementos de la pestaña lateral	118
Figura 11. Elementos de la pestaña Inicio.....	120
Figura 12. Elementos de la pestaña Cintas	121
Figura 13. Elementos de la pestaña Robot.....	122
Figura 14. Elementos de la pestaña Refrigeración.....	124
Figura 15. Elementos de la pestaña Histórico.....	125
Figura 16. Archivo de datos.	126

Índice de tablas

Tabla 1. Requerimientos Hardware	114
Tabla 2. Elementos de la pestaña lateral.....	119
Tabla 3. Elementos de la pestaña Inicio	120
Tabla 4. Elementos de la pestaña Cintas.....	121
Tabla 5. Elementos de la pestaña Robot	123
Tabla 6. Elementos de la pestaña Refrigeración	124
Tabla 7. Elementos de la pestaña Histórico	125

1 Montaje de las maquetas



Figura 1. Montaje de las maquetas en laboratorio

El montaje de las maquetas ha de realizarse como se indica en la *Figura 1*. De esta forma queda garantizado el correcto manejo de las piezas por parte de las dos maquetas.

2 Montaje de la Peltier



Figura 2. Imágenes del montaje de la Peltier en laboratorio

La célula Peltier debe ser montada utilizando el equipo siguiente:

- Entradas/Salidas analógicas del autómata
- Fuente de tensión controlable
- Fuente de tensión variable

Una ilustración del montaje se aprecia en la *Figura 2*.

3 Requerimientos

3.1 Hardware

3.1.1 Unity Pro

Unity Pro admite las plataformas Hardware siguientes:

- Modicon M340
- Premium
- Atrium
- Quantum

3.1.2 LabVIEW

LabVIEW precisa de los siguientes requerimientos mínimos²⁶:

Procesador	Pentium 4/M o equivalente
RAM	1 GB
Resolución de Pantalla	1024 x 768 píxeles
SO	Windows 8.1/8/7/Vista (32 bits y 64 bits) Windows XP SP3 (32 bits) Windows Server 2012 R2 (64 bits) Windows Server 2008 R2 (64 bits) Windows Server 2003 R2 (32 bits)
Espacio en Disco	5 GB (Incluye controladores predeterminados del DVD de Controladores de Dispositivos de NI)

Tabla 1. Reaquerimientos Hardware

3.2 Software

- Sistema operativo:
- Windows 8.1/8/7/Vista (32 bits y 64 bits)
 - Windows XP SP3 (32 bits)
 - Windows Server 2008/2012 R2 (64 bits)
 - Windows Server 2003 R2 (32 bits)
 - OS X 10.7, 10.8 ó 10.9
 - Linux kernel 2.4x, 2.6x ó 3.x y GNU C Library (glibc) v.251 para arquitectura Intel x86_64
- LabVIEW versión 13.0 o superior con módulo DSC 2013 (Datalogging and Supervisory Control Module 2013).
 - Editor de textos compatible con .xls (archivo de texto Excel de Microsoft Office)

²⁶ Se ha seleccionado como sistema operativo estándar el Windows. Si se dispone de un equipo cuyo sistema operativo sea diferente, en el siguiente enlace se podrá comprobar si cumple los requerimientos hardware necesarios : <http://www.ni.com/labview/requirements/esa/>

4 Esquema de inicio de aplicaciones

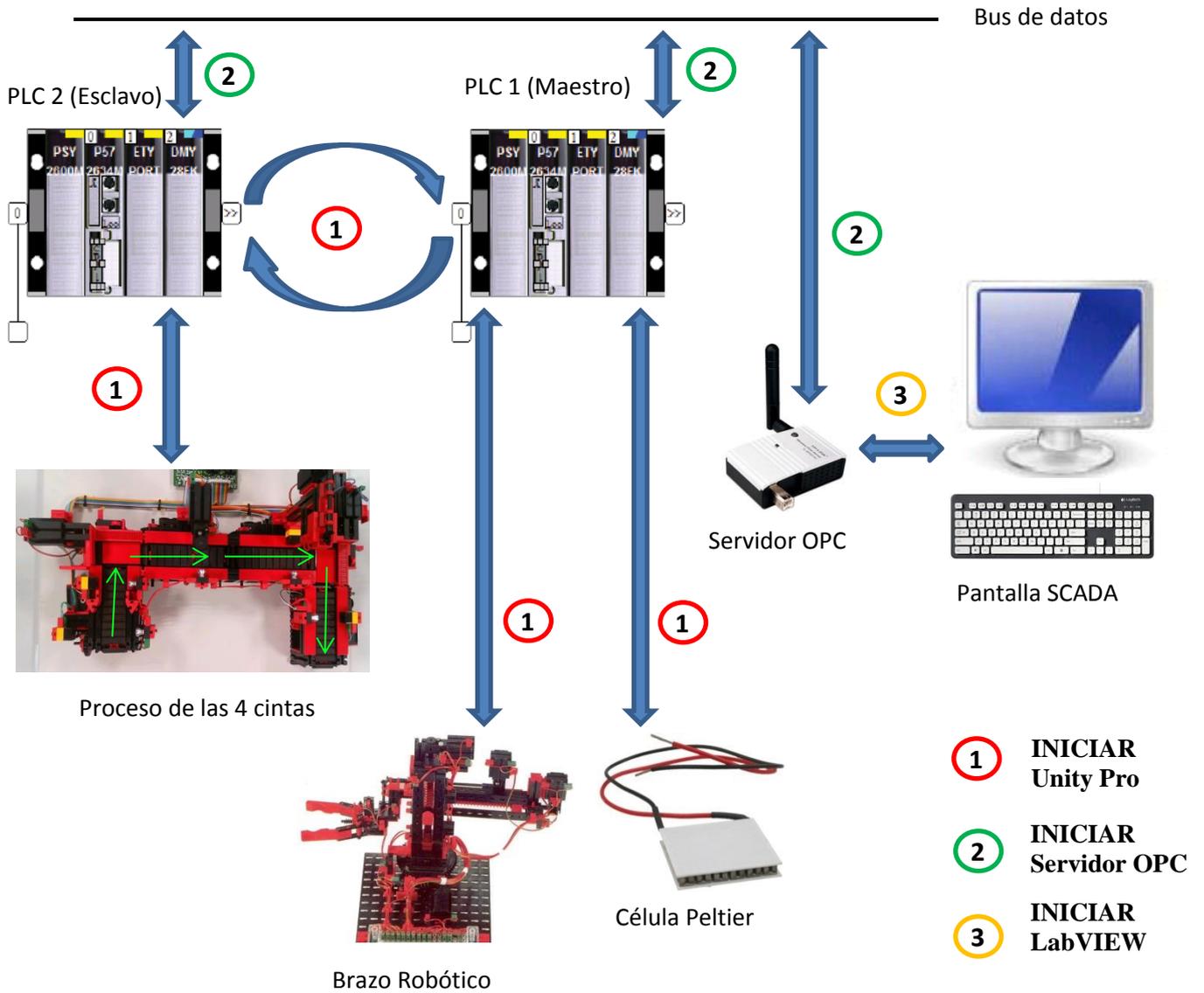


Figura 3. Secuencia de inicio de programas

5 Iniciar el programa en Unity

Estando correctamente conectadas las maquetas, se abren los dos programas, el referente al brazo robótico y el de las cintas desde el Unity Pro. Es necesario ver la dirección IP de los dos autómatas y establecerla en el Unity como se muestra en la siguiente Figura 4

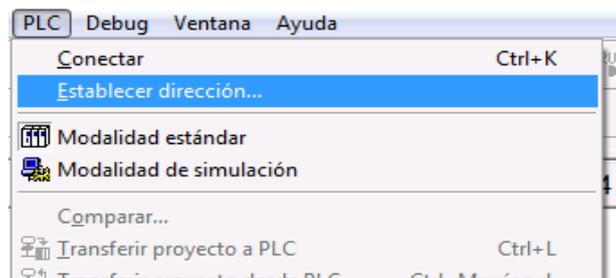


Figura 4. Barra de herramientas del Unity Pro

Aparecerá la siguiente pestaña donde se introducirá la dirección del autómata para el brazo robótico y análogamente se debe realizar el mismo procedimiento para el programa de las cintas.

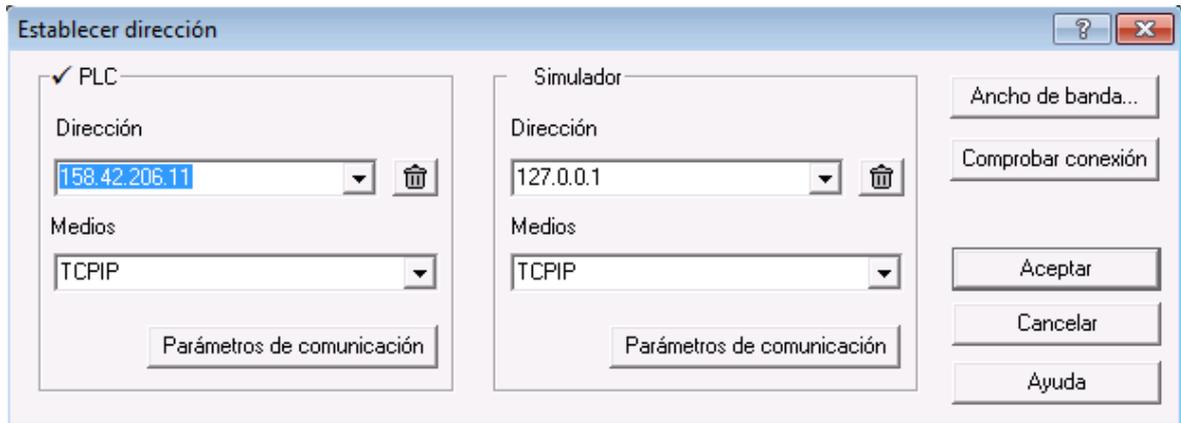


Figura 5. Establecer dirección del autómata

6 Iniciar el servidor

El siguiente paso será abrir el programa KEPServerEX.

Dentro del programa se debe abrir el archivo “*tfg_yago*”.

La única comprobación que se ha de hacer es ver si la dirección de ambos aparatos coincide con la dirección de los autómatas.

Dando a click derecho sobre el aparato y posteriormente al botón Properties se accede directamente al cuadro donde podremos ver la siguiente imagen (Figura 6).

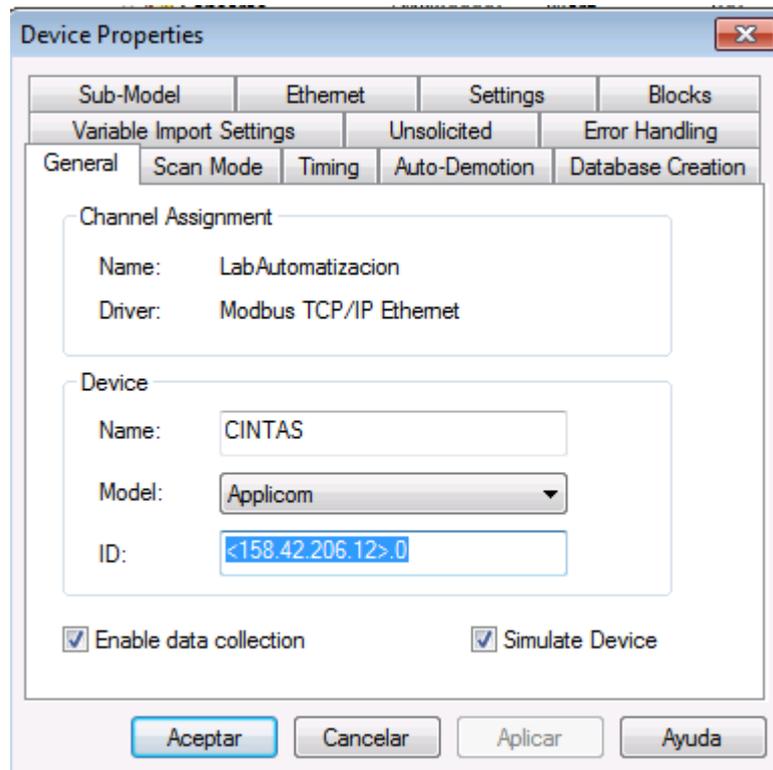


Figura 6. Establecer dirección del autómata en el servidor

En la ventana de la Figura 6 se podrá establecer la dirección del autómata.

7 Iniciar la pantalla SCADA

Para la ejecución del programa se precisa de la instalación previa de LabVIEW y el módulo DSC de LabVIEW.

Con la instalación ya completa, se podrá abrir el proyecto que tendrá el nombre de “Proyecto_Pantalla”. Dentro del proyecto existe un archivo llamado “Pantalla_yago”, que es el que se ha de abrir seguidamente.

Una vez abierto, se ha de pulsar el botón ejecutar, que es el que se muestra en la Figura 7.



Figura 7. Botón Ejecutar

El programa empleará unos segundos en cargar las librerías y seguidamente se iniciará.

Cada usuario tendrá una contraseña predefinida y deberá identificarse para acceder a los permisos que tenga asignados. Por esta razón, antes de poder acceder a las pestañas y gestionar desde las mismas los diferentes elementos, el usuario deberá realizar una identificación previa.

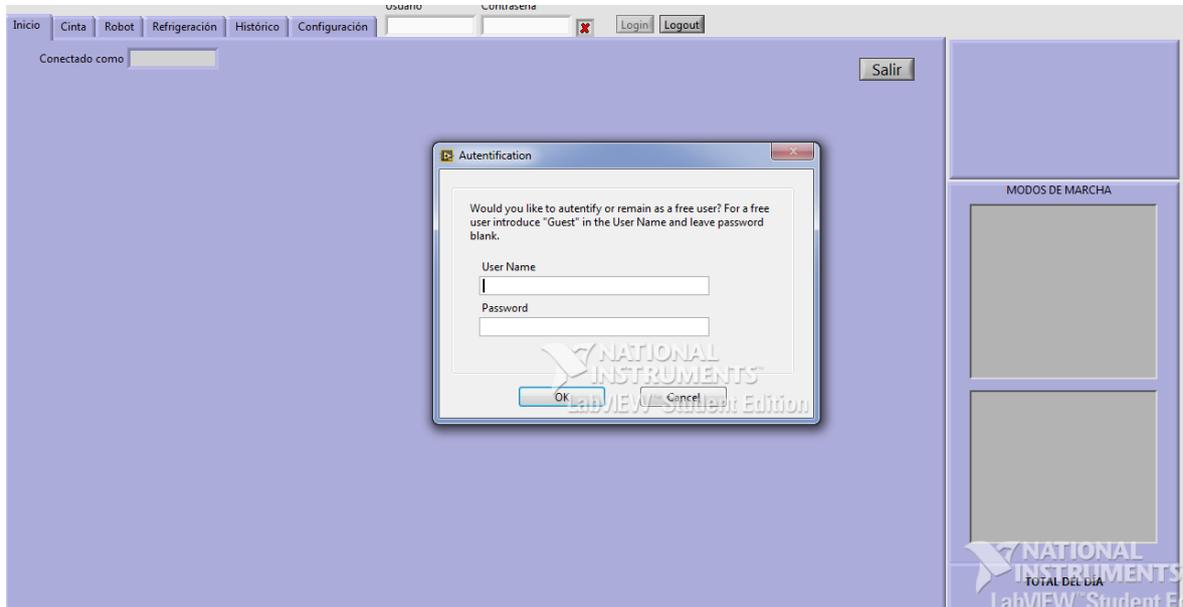


Figura 8. Ventana emergente de identificación

Si la identificación es correcta, ya se podrá acceder a las diferentes pestañas.

Por el contrario, si la identificación no fuera correcta, la ventana de autenticación permanecerá abierta hasta que algún usuario se identifique correctamente.

7.1 Pestañas e inicio de sesión durante el programa

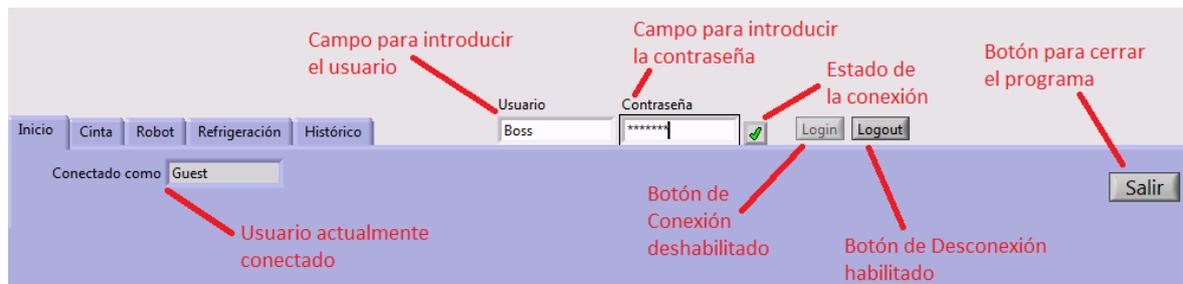


Figura 9. Pestañas y control de usuario

En la *Figura 9* se muestran las 5 pestañas en las que se divide el programa:

- **Inicio:** Aporta una visión global.
- **Cintas:** Posibilita el manejo de elementos específicos de las cintas.
- **Robot:** Posibilita el manejo en los 4 grados de libertad del brazo robótico
- **Refrigeración:** Permite la gestión de la célula Peltier.
- **Histórico:** Permite la visión a tiempo real de algunos datos de interés del proyecto.

Existirá además una pestaña lateral que será siempre accesible con los modos de marcha y los modos de alarma.

7.2 Pestaña Lateral

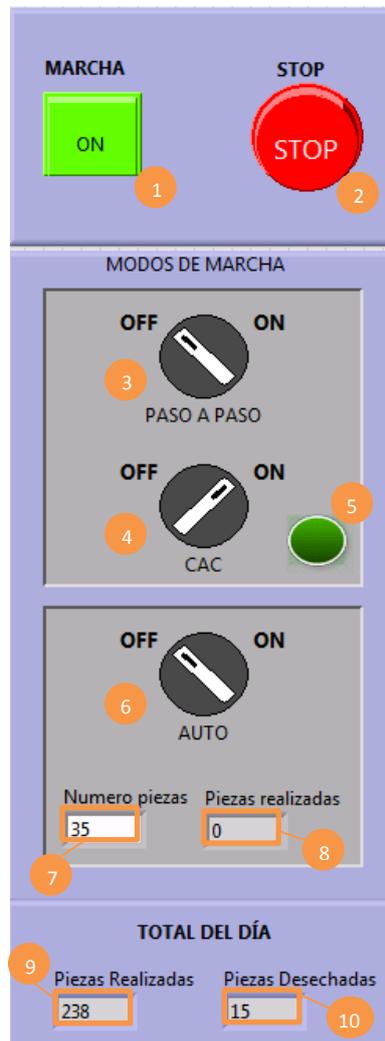


Figura 10. Elementos de la pestaña lateral

Nº	Descripción de la Pestaña Lateral
1	Botón de marcha
2	Botón de Stop
3	Modo de marcha Paso a Paso
4	Modo de marcha Ciclo a Ciclo
5	Botón de ciclo a ciclo
6	Modo de marcha Automático
7	Control de número de piezas a realizar
8	Indicador de piezas realizadas en modo automático
9	Indicador de piezas terminadas en la jornada actual
10	Indicador de piezas desechadas en la jornada actual

Tabla 2. Elementos de la pestaña lateral

7.2.1 Observaciones a considerar de la pestaña lateral:

- **Botón de marcha:** Permitirá activar la maqueta de las cintas y la del brazo robótico. Si se desactiva el botón de marcha durante el ciclo de uso, se reiniciarán ambas maquetas y no se pondrán en marcha hasta que se vuelva a activar este botón.
- **Botón de Stop:** Cuando se pulsa para el funcionamiento de las maquetas en el lugar donde éstas se encuentren. Cuando se vuelve a desactivar, las maquetas siguen su funcionamiento desde donde se quedaron. Este botón sería el equivalente a un botón de pánico.
- **Modo de marcha Paso a Paso:** Permitirá el control manual de los elementos de las dos maquetas de manera independiente. Los usuarios que tienen permiso para acceder a este modo de marcha son por defecto:
 - Supervisor
 - Boss
- **Modo de marcha Ciclo a Ciclo:** En este modo, el brazo requerirá una confirmación (5) cada vez que deba acudir a retirar una pieza de la cinta de descarga. Los usuarios que tienen permiso para acceder a este modo de marcha son por defecto:
 - Operator
 - Supervisor
 - Boss
- **Modo de marcha automático:** Ambas maquetas funcionarán de manera ininterrumpida hasta que se complete el número de piezas a realizar (7) en el modo automático o bien se cambie de modo de marcha. Los usuarios que tienen permiso para acceder a este modo de marcha son por defecto:
 - Operator
 - Supervisor
 - Boss
- Cuando se desee cambiar de modo de marcha habiendo uno ya seleccionado, se abrirá una ventana pidiendo la confirmación de dicho cambio. En caso de que se confirme, el modo de marcha actual se desactivará automáticamente. En caso de que se cancele, el modo de marcha actual seguirá activo y se cancelará el cambio.

7.3 Pestaña Inicio

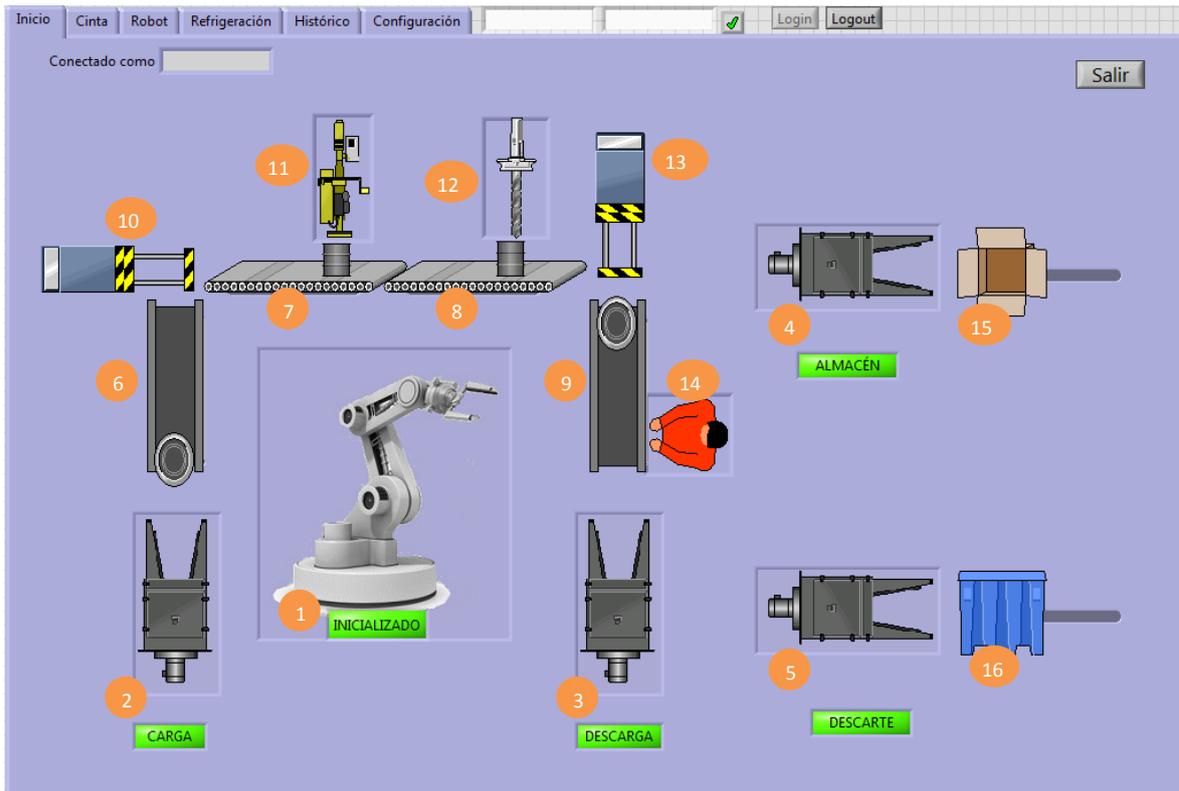


Figura 11. Elementos de la pestaña Inicio

Nº	Descripción Pestaña Inicio
1	Brazo en posición de "Inicializado"
2	Brazo dirigiéndose a posición de cinta de carga
3	Brazo dirigiéndose a posición de cinta de descarga
4	Brazo dirigiéndose a posición de almacén
5	Brazo dirigiéndose a posición de descarte
6	Indicador del avance de la pieza por la cinta de carga
7	Indicador del avance de la pieza por la cinta de la fresadora
8	Indicador del avance de la pieza por la cinta de la taladradora
9	Indicador del avance de la pieza por la cinta de descarga
10	Indicador de la posición del empujador 1
11	Indicador de actuación de la fresadora
12	Indicador de actuación de la taladradora
13	Indicador de la posición del empujador 2
14	Indicador de presencia de operario durante una inspección
15	Indicador de pieza recibida en almacén
16	Indicador de pieza recibida en descarte

Tabla 3. Elementos de la pestaña Inicio

7.4 Pestaña de cintas

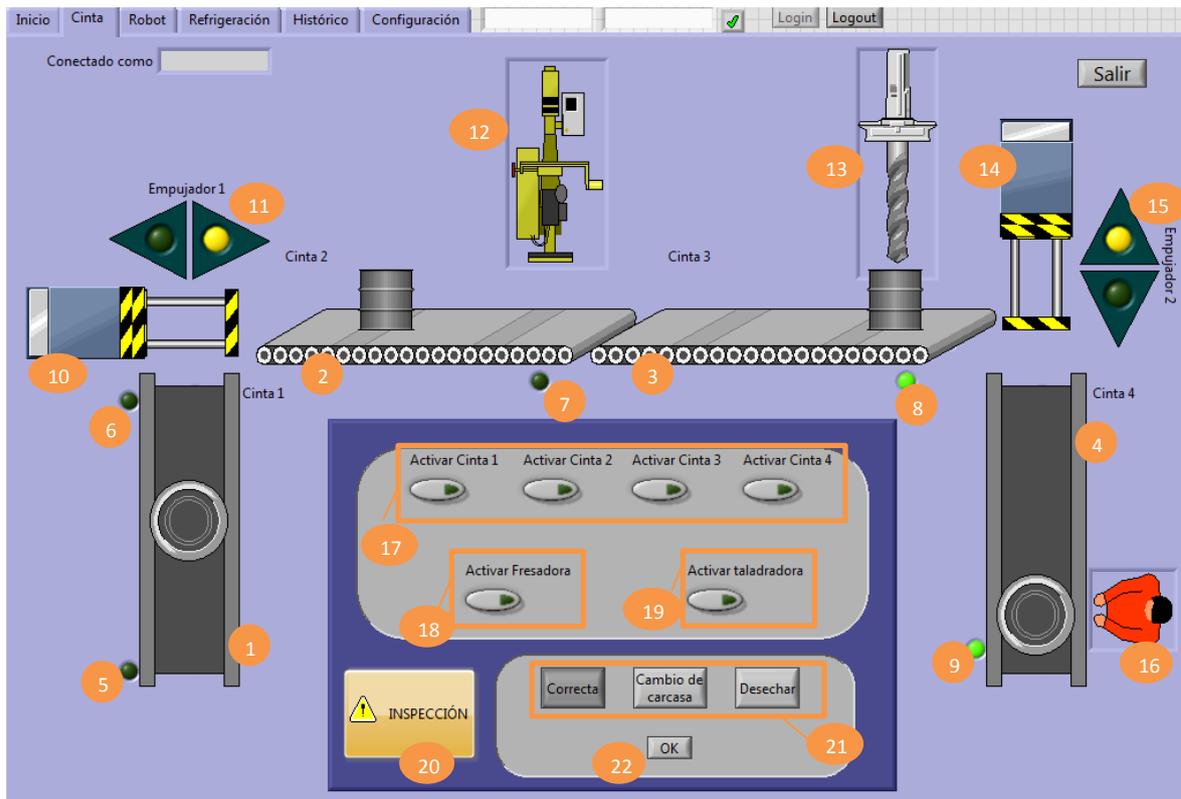


Figura 12. Elementos de la pestaña Cintas

Nº	Descripción de la pestaña Cintas
1	Indicador del avance de la pieza por la cinta 1
2	Indicador del avance de la pieza por la cinta 2
3	Indicador del avance de la pieza por la cinta 3
4	Indicador del avance de la pieza por la cinta 4
5	Sensor de pieza al inicio de la cinta de carga
6	Sensor de pieza en el empujador 1
7	Sensor de pieza en la fresadora
8	Sensor de pieza en la taladradora
9	Sensor de pieza al final de la cinta de descarga
10	Indicador de la posición del empujador 1
11	Indicador de la dirección del movimiento del empujador 1
12	Indicador de actuación de la fresadora
13	Indicador de actuación de la taladradora
14	Indicador de la posición del empujador 2
15	Indicador de la dirección del movimiento del empujador 2
16	Indicador de presencia de operario durante la inspección
17	Controles para activar las cintas manualmente
18	Control para activar la fresadora manualmente
19	Control para activar la taladradora manualmente
20	Botón de Inspección
21	Controles para seleccionar el tipo de pieza
22	Control para confirmar el tipo de pieza

Tabla 4. Elementos de la pestaña Cintas

7.4.1 Observaciones a considerar en la pestaña Cintas:

- Esta pestaña es accesible a los siguientes usuarios:
 - Operator.
 - Supervisor.
 - Boss.
- Los controles manuales (17-19) sólo estarán habilitados mientras el modo de marcha seleccionado sea el modo Paso a Paso.
- Los controles para selección de tipo de pieza (21-22) sólo estarán habilitados mientras el botón de Inspección esté activo.
- Cuando se selecciona un tipo de pieza ya sea como correcta, cambio o defectuosa, el anterior tipo que estuviera seleccionado se desactiva automáticamente.
- Para confirmar el tipo de pieza es necesario pulsar el botón de confirmación (22).

7.5 Pestaña Robot

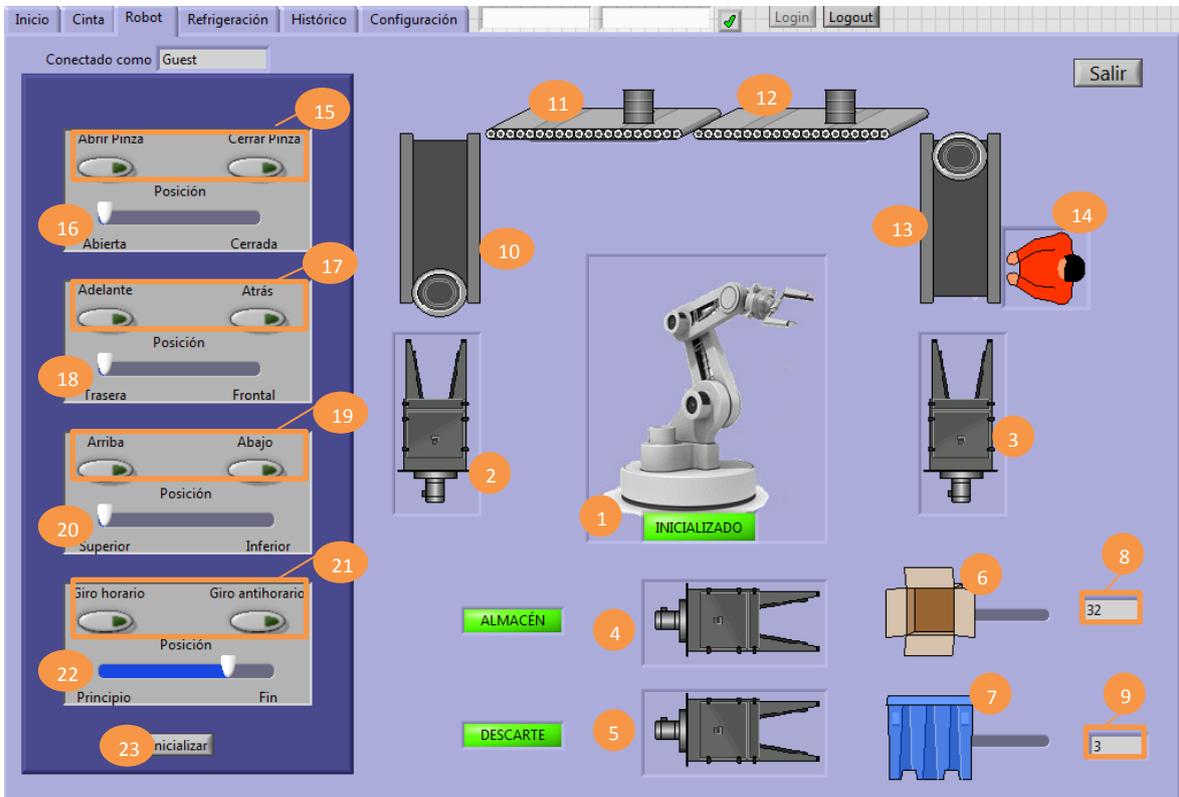


Figura 13. Elementos de la pestaña Robot

Nº	Descripción de la pestaña Robot
1	Brazo en posición de "Inicializado"
2	Brazo dirigiéndose a posición de cinta de carga
3	Brazo dirigiéndose a posición de cinta de descarga
4	Brazo dirigiéndose a posición de almacén
5	Brazo dirigiéndose a posición de descarte
6	Indicador de pieza recibida en almacén
7	Indicador de pieza recibida en descarte
8	Indicador de piezas terminadas en la jornada actual
9	Indicador de piezas desechadas en la jornada actual
10	Indicador del avance de la pieza por la cinta de carga
11	Indicador del avance de la pieza por la cinta de la fresadora
12	Indicador del avance de la pieza por la cinta de la taladradora
13	Indicador del avance de la pieza por la cinta de descarga
14	Indicador de presencia de operario durante una inspección
15	Controles para abrir/cerrar la pinza
16	Indicador de la posición relativa de la pinza
17	Controles para desplazar el brazo de agarre
18	Indicador de la posición relativa del brazo de agarre
19	Controles para desplazar el brazo en sentido vertical
20	Indicador de la posición vertical relativa del brazo
21	Controles para desplazar el brazo en el sentido de giro
22	Indicador de la posición de giro relativa del brazo
23	Control para llevar el brazo a su posición de "Inicializado"

Tabla 5. Elementos de la pestaña Robot

7.5.1 Observaciones a considerar en la pestaña Robot

- Esta pestaña es accesible a los siguientes usuarios:
 -Operator.
 -Supervisor.
 -Boss.
- Los controles manuales (15-23) sólo estarán habilitados mientras el modo de marcha seleccionado sea el modo Paso a Paso.
- Aunque los indicadores (16, 18, 20, 22) se encuentren deshabilitados fuera del modo de marcha Paso a Paso, se podrá ver la posición relativa del brazo.

7.6 Pestaña Refrigeración

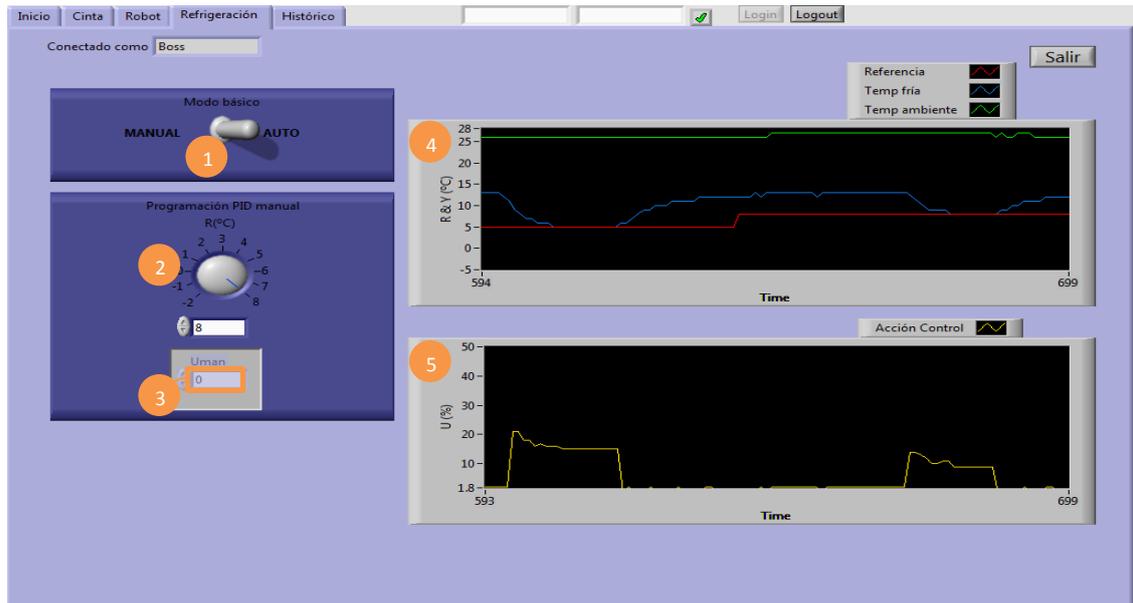


Figura 14. Elementos de la pestaña Refrigeración

Nº	Descripción de la pestaña Refrigeración
1	Botón de selección de modo Automático/Manual
2	Control de la referencia
3	Control de la acción manual
4	Gráfico de los sensores de la célula Peltier
5	Gráfico de la acción que aplica la Peltier

Tabla 6. Elementos de la pestaña Refrigeración

7.6.1 Observaciones de la pestaña Refrigeración:

- Esta pestaña es accesible a los siguientes usuarios:
 - Supervisor
 - Boss.

7.7 Pestaña Histórico

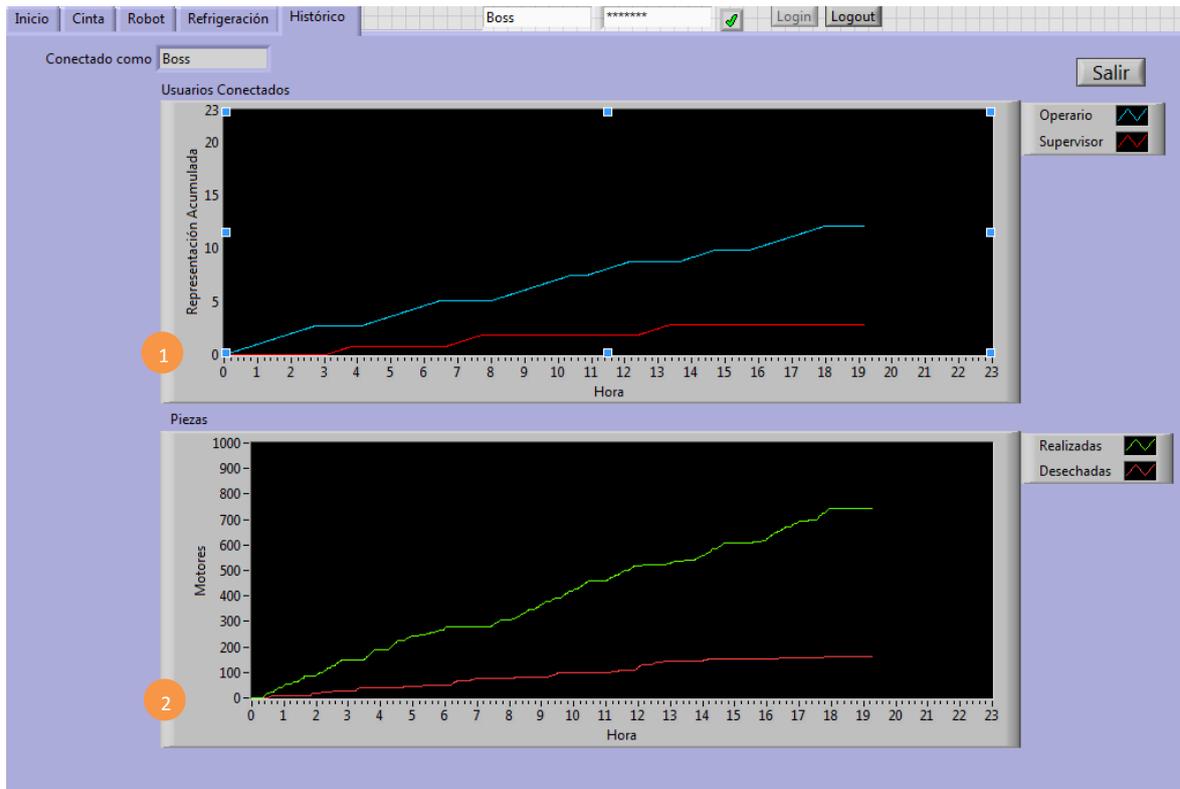


Figura 15. Elementos de la pestaña Histórico

Nº	Descripción de la pestaña Histórico
1	Gráfico que muestra la conexión de los usuarios "Operator" y "Supervisor" a lo largo del día.
2	Gráfico que muestra las piezas terminadas y las desechadas a lo largo del día.

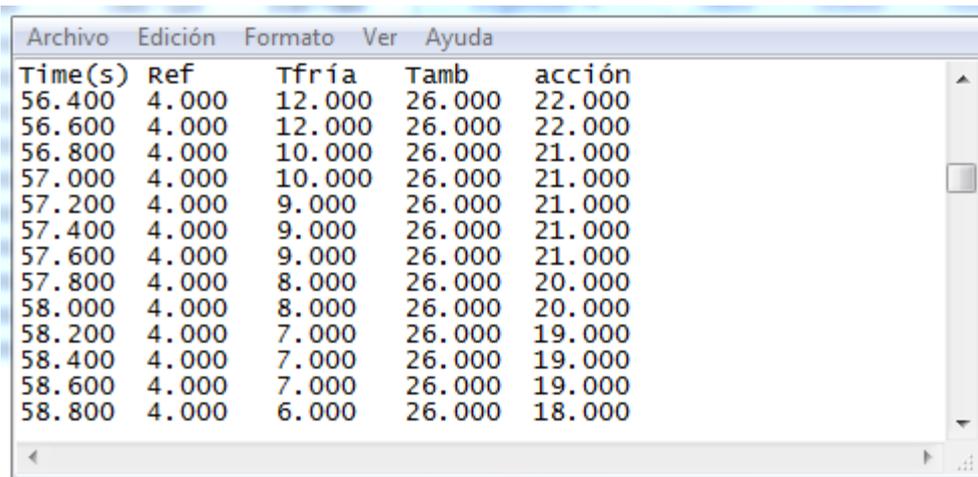
Tabla 7. Elementos de la pestaña Histórico

7.7.1 Observaciones a considerar en la pestaña Histórico:

- Esta pestaña es accesible a los siguientes usuarios:
-Boss.

8 Almacenamiento de datos

Una vez se pulse el botón “Salir”, presente en todas la pestañas, nos aparecerá un cuadro de diálogo donde especificaremos la ruta donde se van a guardar los datos del ensayo realizado. Este archivo contendrá los datos de la célula Peltier de interés (Figura 16).



Time(s)	Ref	Tfría	Tamb	acción
56.400	4.000	12.000	26.000	22.000
56.600	4.000	12.000	26.000	22.000
56.800	4.000	10.000	26.000	21.000
57.000	4.000	10.000	26.000	21.000
57.200	4.000	9.000	26.000	21.000
57.400	4.000	9.000	26.000	21.000
57.600	4.000	9.000	26.000	21.000
57.800	4.000	8.000	26.000	20.000
58.000	4.000	8.000	26.000	20.000
58.200	4.000	7.000	26.000	19.000
58.400	4.000	7.000	26.000	19.000
58.600	4.000	7.000	26.000	19.000
58.800	4.000	6.000	26.000	18.000

Figura 16. Archivo de datos.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC

Documento nº8: Planos

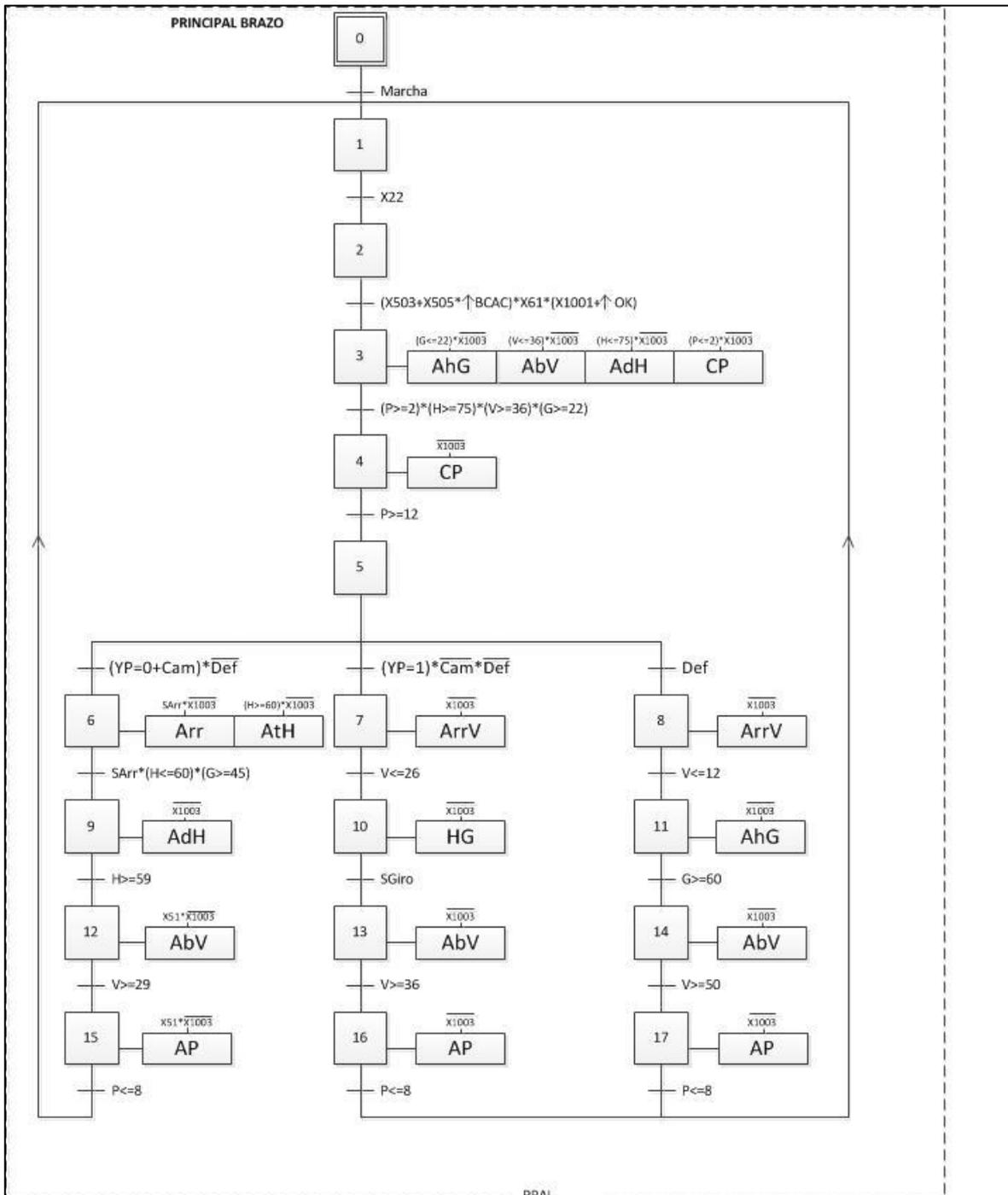
AUTOR: SANTIAGO FERNÁNDEZ RODILLA

TUTOR: RAÚL SIMARRO FERNÁNDEZ

COTUTOR: JUAN MANUEL HERRERO DURÁ

Índice de contenido

Plano 1. Grafcet principal del brazo robótico.....	129
Plano 2. Grafcets auxiliares del brazo robótico.....	130
Plano 3. Grafcet de alarma y contadores de pulsos.....	131
Plano 4. FIFO brazo robótico y contadores de piezas.....	132
Plano 5. Grafcets cintas 1 y 2 (carga y fresadora).....	133
Plano 6. Grafcets cintas 3 y 4 (taladradora y descarga) y auxiliares.....	134
Plano 7. Grafcet de alarma y FIFO de las cintas.....	135
Plano 8. PID y parámetros.....	136



TRABAJO FINAL DE GRADO:
DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC Y MONITORIZACIÓN POR SCADA

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

DIRECTORES: RAÚL SIMARRO FERNÁNDEZ
JUAN MANUEL HERRERO DURÁ

PLANO:
1

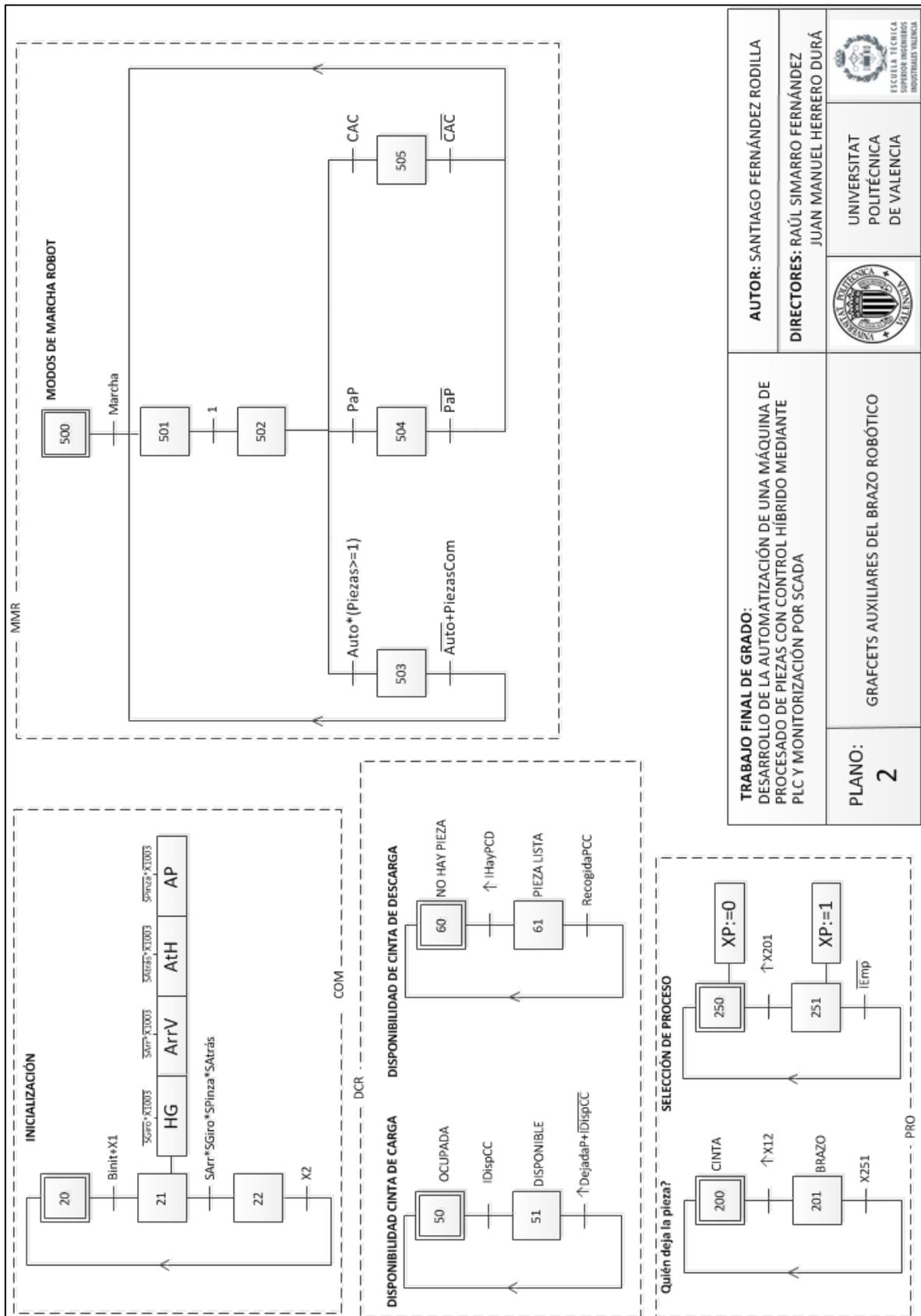
GRAFICET PRINCIPAL DEL BRAZO ROBÓTICO



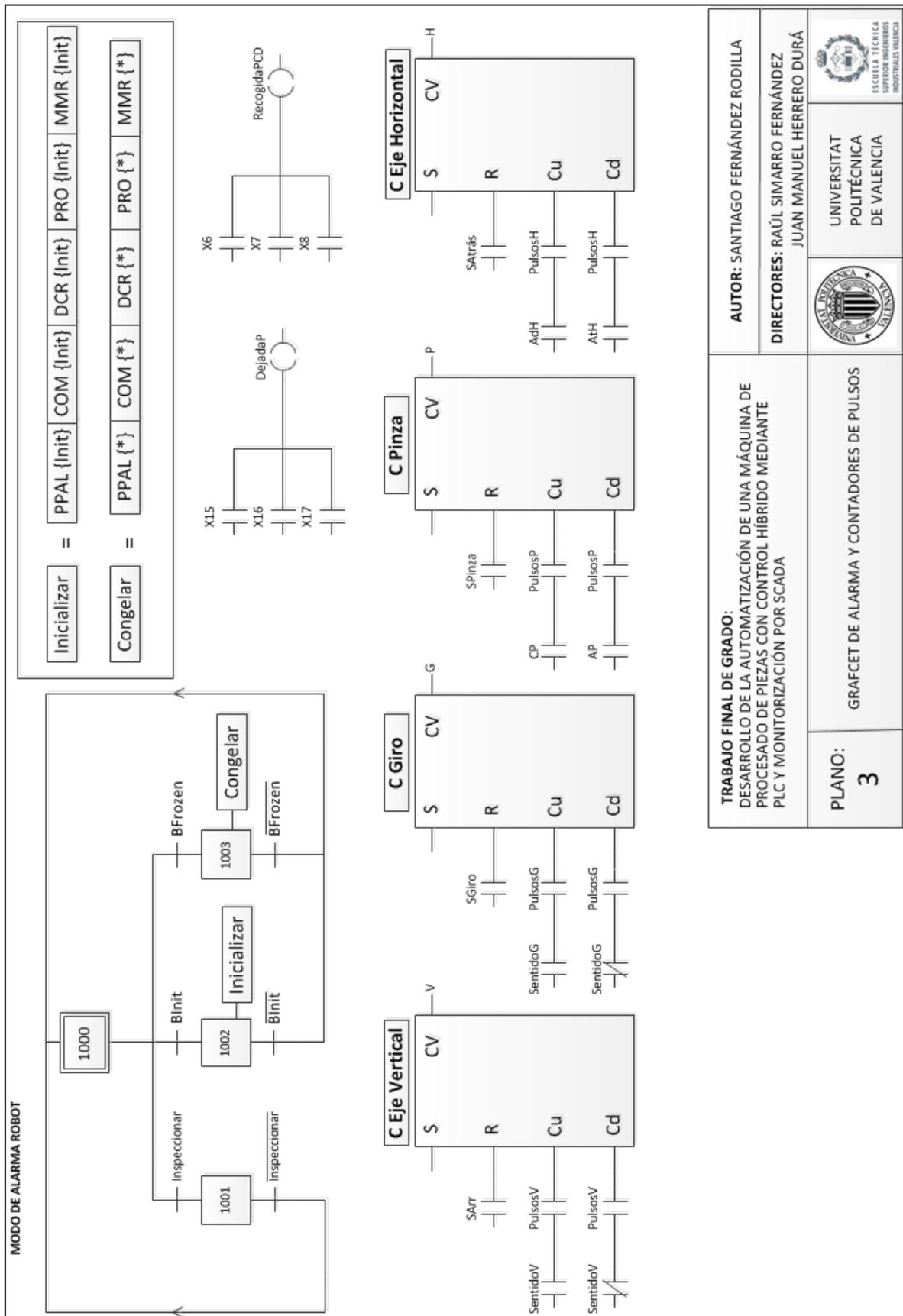
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

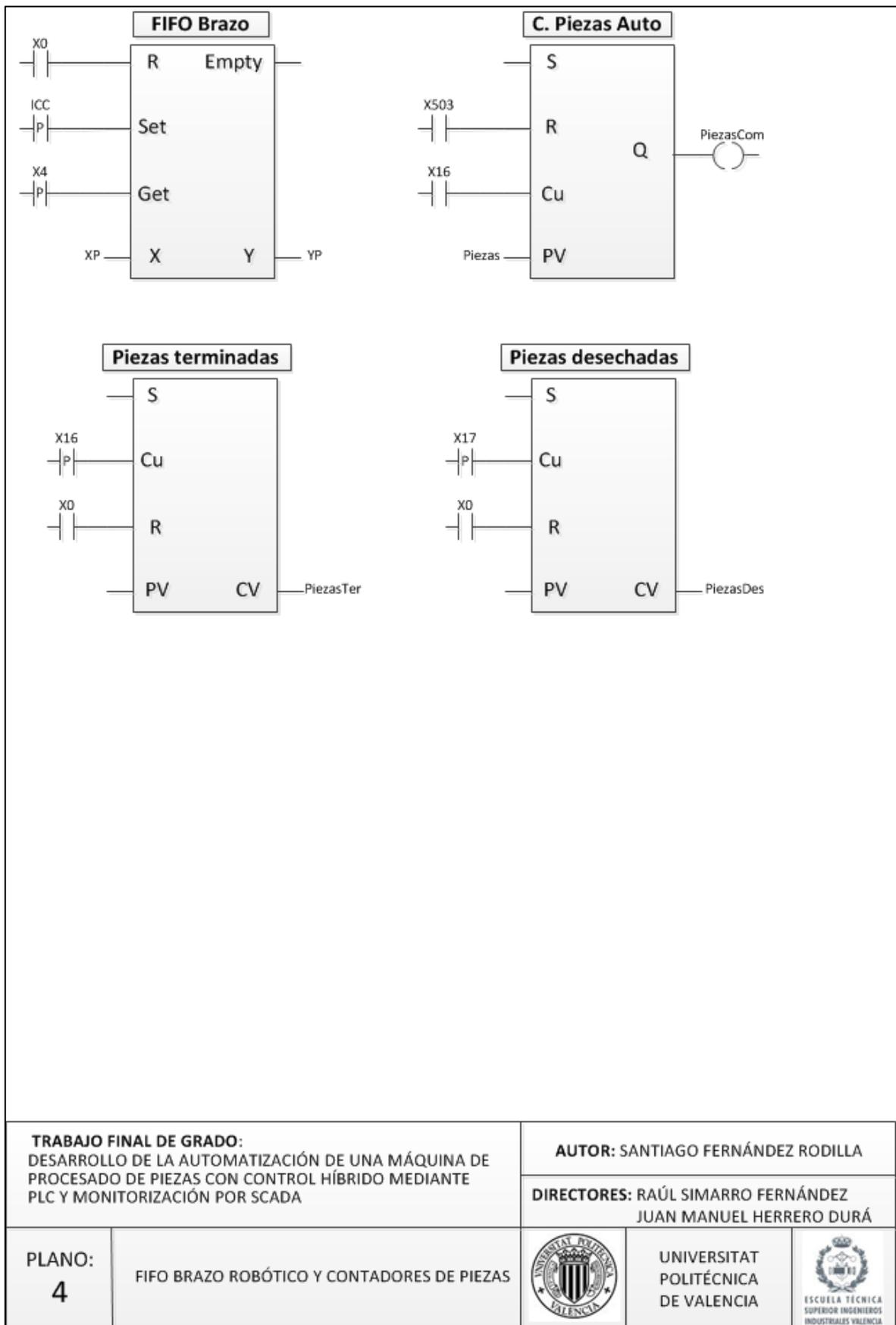


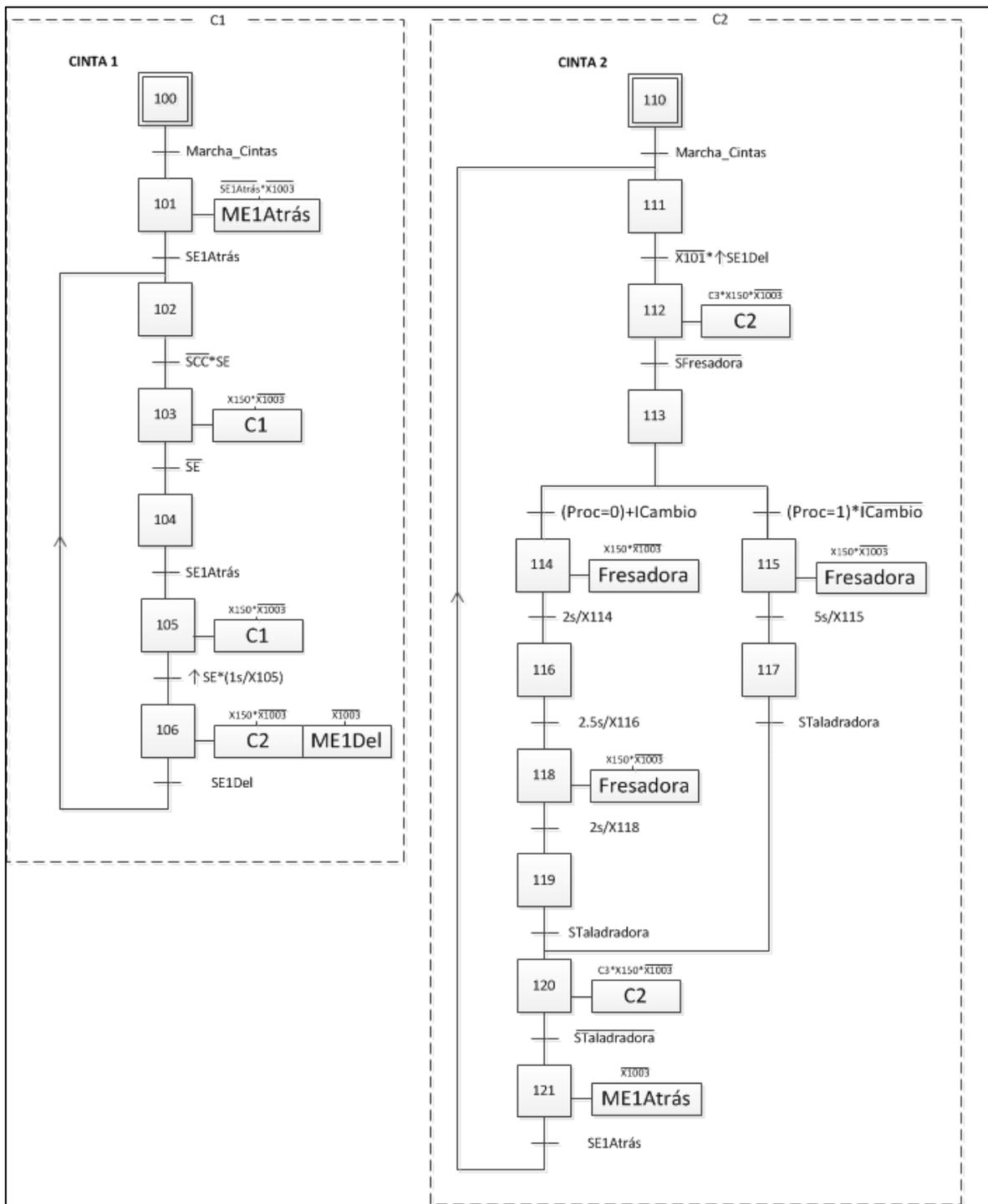
ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALÈNCIA



TRABAJO FINAL DE GRADO: DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC Y MONITORIZACIÓN POR SCADA	AUTOR: SANTIAGO FERNÁNDEZ RODILLA	
	DIRECTORES: RAÚL SIMARRO FERNÁNDEZ JUAN MANUEL HERRERO DURÁ	
PLANO: 2	GRAFCETS AUXILIARES DEL BRAZO ROBÓTICO	
		UNIVERSITAT POLITÈCNICA DE VALÈNCIA
		







TRABAJO FINAL DE GRADO:
DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC Y MONITORIZACIÓN POR SCADA

AUTOR: SANTIAGO FERNÁNDEZ RODILLA

DIRECTORES: RAÚL SIMARRO FERNÁNDEZ
JUAN MANUEL HERRERO DURÁ

PLANO:
5

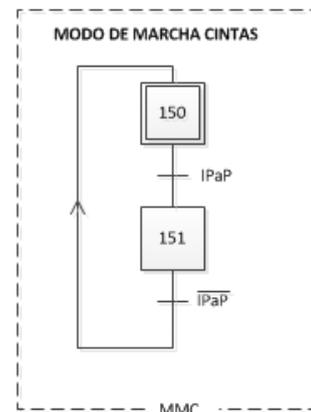
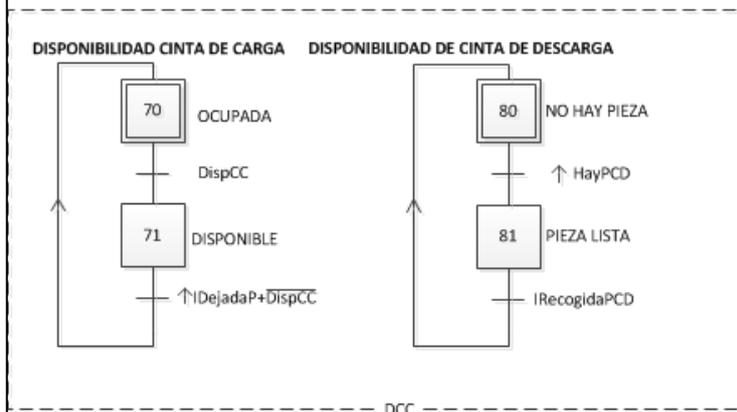
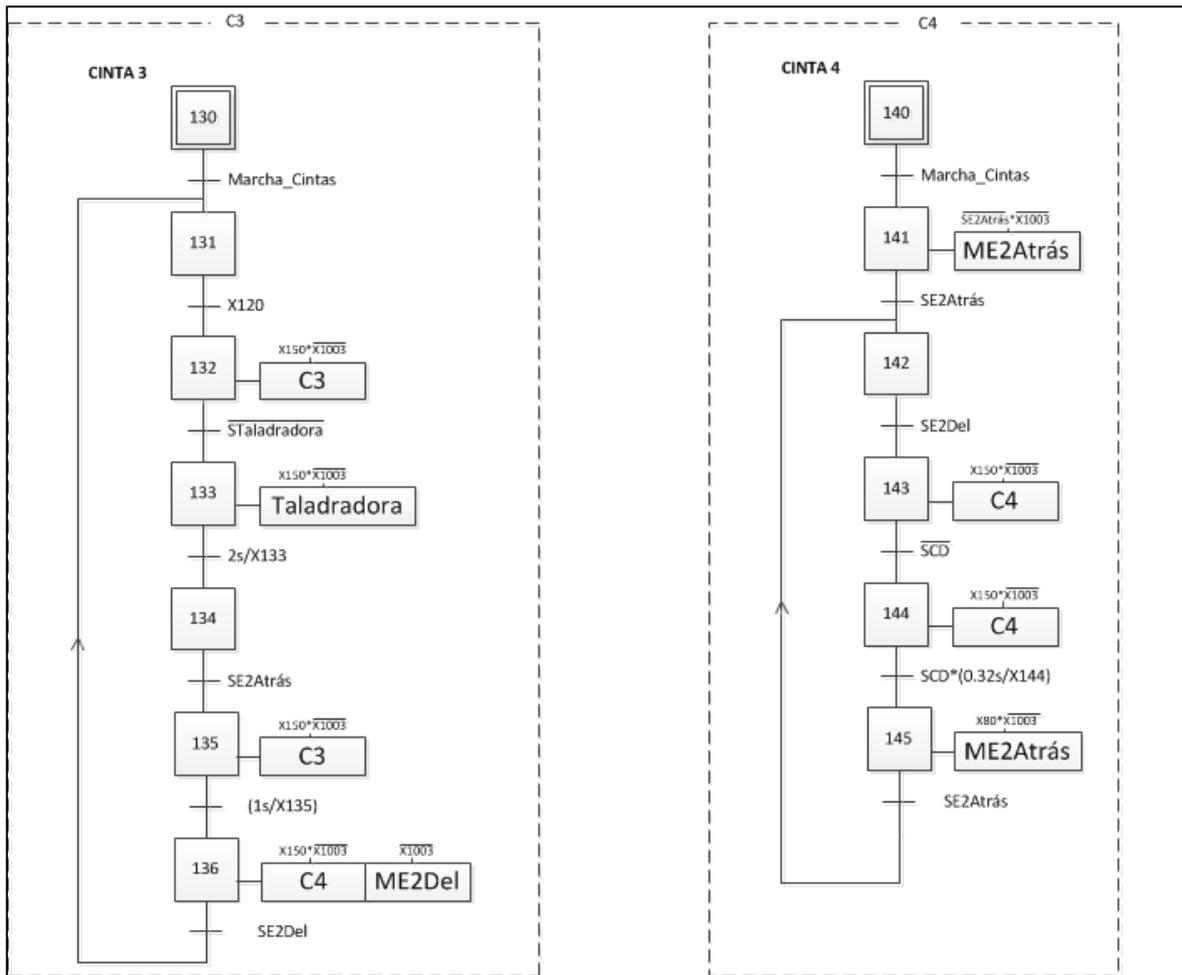
GRAFSETS CINTAS 1 Y 2 (CARGA Y FRESADORA)



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALÈNCIA



TRABAJO FINAL DE GRADO: DESARROLLO DE LA AUTOMATIZACIÓN DE UNA MÁQUINA DE PROCESADO DE PIEZAS CON CONTROL HÍBRIDO MEDIANTE PLC Y MONITORIZACIÓN POR SCADA		AUTOR: SANTIAGO FERNÁNDEZ RODILLA	
		DIRECTORES: RAÚL SIMARRO FERNÁNDEZ JUAN MANUEL HERRERO DURÁ	
PLANO: 6	GRAFCETS CINTAS 3 Y 4 (TALADRADORA Y DESCARGA) Y AUXILIARES		UNIVERSITAT POLITÈCNICA DE VALÈNCIA 

