

**UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR**

**FACULTAD DE INGENIERÍA Y GESTIÓN**  
**ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA Y**  
**TELECOMUNICACIONES**



**“DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ REMOTA PARA LA  
EDUCACIÓN A DISTANCIA DE LA ESTACIÓN AUTOMATIZADA DE  
PROCESOS INDUSTRIALES EN LA UNIVERSIDAD NACIONAL  
TECNOLÓGICA DE LIMA SUR, 2020”**

**TRABAJO DE SUFICIENCIA PROFESIONAL**

Para optar el Título Profesional de

**INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**PRESENTADO POR EL BACHILLER**

CHAMBI APAZA, YHON CRISTIAN

Villa El Salvador  
2020

## **DEDICATORIA**

El presente trabajo se lo dedico a mis padres que con todo el esfuerzo pudieron inculcarme buenos hábitos que gracias a ellos pude concluir con éxito la carrera profesional

## **AGRADECIMIENTOS**

En primer lugar, agradezco a Dios por darme la oportunidad de seguir aquí, por concederme el placer de tener amistades excelentes que me apoyan y me motivan a seguir adelante es gracias a todos ellos que de alguna forma u otra son y serán parte de mi especial consideración a ellos les agradezco todo.

## INDICE

Lista de Figuras .....	vi
Lista de Tablas .....	ix
RESUMEN .....	10
INTRODUCCIÓN.....	11
OBJETIVOS .....	12
a. General.....	12
b. Específicos.....	12
CAPÍTULO I: MARCO TEÓRICO .....	13
1.1 Bases Teóricas .....	13
1.1.1 Interfaz Remota.....	13
1.1.2 Educación a Distancia.....	15
1.1.3 Estación de Automatización de Procesos Industriales .....	17
1.1.4 Marco Teórico Específico .....	21
1.2 Definición de términos básicos.....	24
CAPÍTULO II: Metodología de Desarrollo del Trabajo Profesional.....	26
2.1 Delimitación temporal y espacial del trabajo .....	26
2.2 Determinación y análisis del problema.....	26
2.3 Modelo de solución propuesto .....	27
2.3.1 Validación de la Conexión Remota a Distancia de Docentes y Estudiantes con Google Meet y Anydesk.....	29
2.3.2 Diseño y desarrollo de la Interfaz Remota en Matlab .....	32
2.3.3 Desarrollo y prueba de la Tarjeta de Adquisición Electrónica para el Control del Sistema Automatizado.....	44
2.3.4 Validación y ejecución del Control del Sistema Automatizado de Procesos Industriales.....	59
2.4 Resultados .....	82
Conclusiones.....	83
Recomendaciones .....	84



Bibliografía .....	85
Anexos.....	87

## LISTA DE FIGURAS

Figura 1. Modelo para el laboratorio remoto de comunicaciones .....	14
Figura 2. Plataforma de educación a Distancia .....	15
Figura 3. Proceso Automatizado.....	17
Figura 4. Proceso Automatizado continuo para el etiquetado de botellas .....	19
Figura 5. Esquema General del Proceso .....	20
Figura 6. Ventana de Control y Monitorización .....	20
Figura 7. Etapas de la Elaboración del Proyecto.....	27
Figura 8. Diagrama Visual del Trabajo de Suficiencia Profesional.....	28
Figura 9. Diagrama de Flujo del Procedimiento de conexión del Docente .....	30
Figura 10. Diagrama de flujo de la conexión del estudiante .....	30
Figura 11. Visualización de los formatos de la cámara empleando el comando imaqtool .....	31
Figura 12. Resultados de la conexión Google Meet y Anydesk de la Interfaz Remota .....	31
Figura 13. Diagrama de Funcionamiento de la Interfaz Remota .....	33
Figura 14. Página principal de la interfaz Remota en Matlab (iniciar sesión) .....	34
Figura 15. Ingreso de Usuario.....	35
Figura 16. Autenticación del estudiante realizado de forma exitosa .....	35
Figura 17. Sistema de Agregación de Estudiantes y Docentes .....	36
Figura 18. Mensaje de Alerta por no darle acepto haber leído los términos y condiciones.....	37
Figura 19. Términos y Condiciones de la Interfaz Remota .....	37
Figura 20. Almacenamiento Correcto de los Datos del Usuario .....	38
Figura 21. Importar imagen para la visualización del usuario en pantalla .....	38
Figura 22. Inicio de sesión del usuario Gerson Daniel Diestra Ninahuilca.....	39
Figura 23. Inicialización de la interfaz remota.....	40
Figura 24. Montaje de las cámaras web en el laboratorio de Control y Automatización .....	43
Figura 25. Interfaz Remota Etapa final .....	43
Figura 26. Diagrama de flujo del funcionamiento de la tarjeta de adquisición electrónica .....	45
Figura 27. Fuente de alimentación .....	47

Figura 28. Circuito de manejo de las entradas y salidas del PIC 16F877A .....	47
Figura 29. Circuito de manejo del LCD a través del microcontrolador PIC 16F628A .....	48
Figura 30. Circuito de comunicación USB-UART con el circuito integrado CH340G .....	48
Figura 31. Etapa de entrada configuración de resistencias .....	49
Figura 32. Etapa de potencia para el manejo de las salidas hacia el PLC .....	49
Figura 33. Diseño de la tarjeta de adquisición electrónica .....	50
Figura 34. Desarrollo del circuito impreso en la placa PCB .....	50
Figura 35. Versión final del desarrollo de la tarjeta de adquisición electrónica .....	51
Figura 36. Conexión y pruebas con el PLC en el laboratorio de Control y Automatización .....	52
Figura 37. Diagrama de Flujo de la Programación en el PIC 16F877A .....	54
Figura 38. Diagrama de Flujo de la Programación en el PIC 16F628A .....	55
Figura 39. Carga del código de programación al PIC 16F877A .....	56
Figura 40. Visualización del encendido del puerto del PLC, recibido por el puerto serial .....	56
Figura 41. Visualización del apagado del puerto del PLC, recibido por el puerto serial .....	57
Figura 42. Demostración de la activación del Sistema .....	57
Figura 43. Demostración de la desactivación del Sistema .....	58
Figura 44. Prueba de Transmisión de datos a 9600 Baudios en el osciloscopio ..	58
Figura 45. Reconocimiento de los Módulos Lucas Nülle GmbH .....	59
Figura 46. Estructura de la Red Profibus .....	59
Figura 47. Lista de variables utilizadas para la simulación .....	60
Figura 48. Programa Principal (Main) .....	61
Figura 49. Esquema del funcionamiento de la Estación de Separación .....	62
Figura 50. Estación de Separación .....	63
Figura 51. Esquema del funcionamiento de la Estación de Montaje .....	64
Figura 52. Estación de Montaje .....	64
Figura 53. Esquema del funcionamiento de la Estación de Verificación .....	65
Figura 54. Estación de Verificación .....	66
Figura 55. Esquema del funcionamiento de la Estación de Manipulación .....	67
Figura 56. Estación de Manipulación .....	67

Figura 57. Flujograma del funcionamiento de la programación de la pantalla HMI	68
.....	68
Figura 58. Lista de variables utilizadas para la conexión con el PLC	68
Figura 59. Diseño de la Interfaz de la pantalla HMI	69
Figura 60. Visualización de la Pantalla HMI OP177B con el programa cargado	69
Figura 61. Bloques de la Estación Automatizada de Procesos Industriales	70
Figura 62. Inicio de la Simulación	71
Figura 63. Detección del sensor derecho de la Estación de Separación	71
Figura 64. Falta piezas en la Estación de Separación	72
Figura 65. Detección del sensor derecho de la Estación de Montaje	72
Figura 66. Falta piezas en la Estación de Montaje	73
Figura 67. Detección del sensor derecho de la Estación de Verificación	73
Figura 68. Detección del sensor derecho de la Estación de Manipulación	74
Figura 69. El pivote desciende Estación de Manipulación	75
Figura 70. El pivote asciende Estación de Manipulación	75
Figura 71. El pivote a 0° grados Estación de Manipulación	76
Figura 72. El pivote a 0° grados desciende Estación de Manipulación	76
Figura 73. El pivote a 0° grados asciende Estación de Manipulación	77
Figura 74. Parada del Proceso	78
Figura 75. Diagrama de Gantt	78

## LISTA DE TABLAS

Tabla 1. Comparación de Microcontroladores .....	46
Tabla 2. Presupuesto para la elaboración del prototipo .....	53
Tabla 3. Cursos de mayor relevancia para la elaboración del proyecto .....	81

## RESUMEN

El presente trabajo de suficiencia profesional para obtener el título profesional de Ingeniero Electrónico y de Telecomunicaciones, titulado “Diseño e implementación de una interfaz remota para la educación a distancia de la estación automatizada de procesos industriales en la Universidad Nacional Tecnológica de Lima Sur, 2020” consistió en el desarrollo de una interfaz remota que incluye un software de enlace remoto desarrollado con Tia Portal, Matlab, Google Meet y Anydesk, una tarjeta de adquisición y potencia para activar y monitorear los puertos de entrada y salida del controlador lógico programable así como cuatro cámaras webs que muestran en tiempo real el funcionamiento del sistema automatizado de procesos industriales del laboratorio de Control y Automatización de la Untels.

Se realizó la tarjeta de adquisición y potencia la cual consta de dos microcontroladores PIC 16F877A y 16F628A que manejan las conexiones con el PLC a través de un circuito resistivo con transistores de potencia que activan y desactivan los puertos de entrada del PLC, también adquieren el estado de sus salidas y para finalizar muestra una visualización de las acciones a realizar por parte de la interfaz gráfica a través de una pantalla LCD 16x2. La programación de ambos microcontroladores fue desarrollada en el entorno MPLAB para obtener una mejor eficiencia del código de programación.

Se realizó la implementación de cuatro cámaras web en la interfaz para obtener una apreciación en tiempo real del sistema automatizado de procesos industriales. Se realizó la programación en el software Tia portal V13 como ejemplo para desarrollar ejercicios con el sistema automatizado de procesos industriales. Se realizó la interfaz en el software Matlab haciendo uso de sus comandos de comunicación serial para establecer la conexión con la tarjeta de adquisición y potencia. La interfaz cuenta con funciones para activar y desactivar las cámaras de acuerdo a la necesidad del estudiante y de un sistema de autenticación para limitar el acceso a un grupo determinado de estudiantes, a su vez esta interfaz tiene la opción de inicializar el puerto serial para así activar y desactivar los puertos de salida y monitorear los puertos de entrada del PIC y estos a su vez del PLC. Dando por concluida una conexión exitosa entre los equipos de laboratorio y los estudiantes.

## INTRODUCCIÓN

El Ministerio de Educación y la Superintendencia Nacional de Educación Superior Universitaria a través de la Resolución Vice Ministerial R.V.M.N°085-2020-MINEDU, del 01 Abril del 2020, suspendió a nivel nacional las clases presenciales de forma indefinida debido a la pandemia por el COVID-19, por lo anterior, las universidades tanto privadas como públicas requieren desarrollar simuladores virtuales que permitan el uso de los equipos de laboratorio, para así compensar las faltas de prácticas con equipos que son importantes para el desarrollo de las competencias profesionales. Según la Asociación de Institutos Superiores Tecnológicos y Escuelas Superiores del Perú afirma que más de 65 mil jóvenes dejaron de estudiar y cerca de 170 mil que se encuentran en el riesgo de abandonar los estudios en los próximos meses, una de las causas es de que algunas carreras necesitan ser presenciales por ser más prácticas que teóricas como es el caso de las ingenierías, es por esta última que se plantea el siguiente trabajo de suficiencia profesional con la finalidad de reducir la brecha entre el estudiante y las partes prácticas del desarrollo de su carrera profesional, diseñar una interfaz remota que permita al estudiante desenvolverse de manera no presencial con los equipos físicos de laboratorio poniendo a prueba sus destrezas para desarrollarse profesionalmente.

El sistema de interfaz remota permite al estudiante tener el control del proceso manejar tanto el sistema mecánico como son los motores, electroválvulas, y conocer el estado lógico de los sensores tipo inductivo, capacitivo, fotoeléctrico y magnético con la finalidad de poder desarrollar sus prácticas como lo venían haciendo en ciclos presenciales. La comunicación a distancia nos la dará las herramientas de Google Meet y Anydesk que son software que se usan actualmente para las clases virtuales y son de uso libre y sin paga.

El presente trabajo de suficiencia profesional propone el capítulo uno llamado marco teórico que contiene las bases teóricas y la definición de términos básicos. En el capítulo dos llamado metodología del desarrollo del trabajo profesional contiene la delimitación temporal y espacial del trabajo, la determinación y análisis del problema, el modelo de solución propuesto, así como los resultados que demuestran el éxito del presente trabajo de suficiencia profesional.

## OBJETIVOS

### a. General

- Diseñar e implementar una interfaz remota para la educación a distancia de la estación automatizada de procesos industriales en la Universidad Nacional Tecnológica de Lima Sur, 2020.

### b. Específicos

- Diseñar e Implementar una tarjeta de adquisición y potencia para activar y monitorear los puertos de entrada y salida del controlador lógico programable que gobierna al sistema automatizado de procesos industriales.
- Integrar cuatro cámaras web en la Interfaz en Matlab para obtener una apreciación en tiempo real del sistema automatizado de procesos industriales.
- Desarrollar la Interfaz en Matlab para el manejo de la tarjeta de adquisición y potencia también la programación en Tia Portal y establecer la conexión remota a través del software Google Meet y Anydesk.
- Validar el funcionamiento del sistema de interfaz remota para la educación a distancia de la estación automatizada de procesos industriales de la Untels.



# CAPÍTULO I:

## MARCO TEÓRICO

### 1.1 Bases Teóricas

Está compuesto por el marco teórico general, el cual está comprendido por los términos generales de una interfaz remota, educación a distancia y la estación de automatización de procesos industriales. El marco teórico específico está compuesto por artículos de investigación relacionados al presente trabajo.

#### 1.1.1 Interfaz Remota

Para Scolari (2018), la interfaz puede ser considerada como: “el lugar donde se produce la interacción, una zona de frontera entre el mundo real y el virtual o mejor un entorno de traducción entre los usuarios, los diseñadores y los artefactos tecnológicos”. (p.26)

Para Gómez (2010), los programas que sirven para establecer una conexión remota se conectan a un servidor para administrarlo, ejecutando programas en máquinas más potentes, etc. Se suelen utilizar para administración a distancia y así evitar desplazamiento. (p.65)

En consecuencia, de acuerdo a lo mencionado anteriormente en la fusión de ambos términos obtenemos la definición de interfaz remota, es la interfaz que nos permite conectarnos de forma remota a un equipo físico ubicado y manejado de manera no presencial de un lugar determinado.

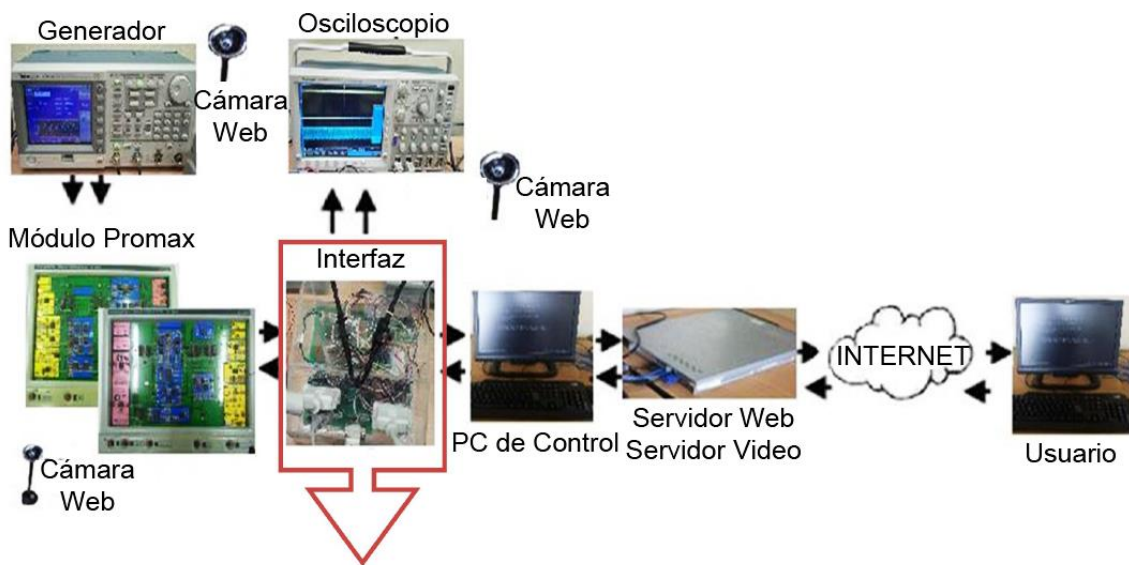
Para Ramírez (2010) una interfaz remota permite que los estudiantes puedan controlar los instrumentos desde un lugar fuera del campus. (p.10)

De acuerdo a Ramírez (2010) en su investigación realizada en la Pontificia Universidad Católica del Perú, la estructura de una interfaz remota está compuesta como se especifica a continuación:

- Tarjeta de adquisición de datos que sirve como medio de comunicación entre los módulos de laboratorio Promax EC-696, el osciloscopio y la computadora.
- La computadora de escritorio se emplea para la visualización de la selección de los parámetros del osciloscopio como también del módulo.
- Configuración servidor web para la visualización de las cámaras web a través del Internet.

- Desarrollo de la Interfaz en el software Matlab para el control de los módulos educativos Promax.

De acuerdo a la Figura 1 se muestra un generador de dos canales que alimenta al módulo emisor; este módulo tiene dos entradas con conectores coaxiales. Por ellas se puede conectar las dos entradas provenientes de los dos canales del generador de señales. Los módulos cuentan con puntos de medición fijos que permiten ver el comportamiento de la señal en su trayectoria. Mediante estos, las señales analógicas pueden ser vistas utilizando un osciloscopio, observando de esta manera las transformaciones que alteran la señal a lo largo de su recorrido.



**INTERFAZ:**

Cable de comunicación: módulo - interfaz, interfaz – PC de control

Herramientas que permitan el control desde la PC: Tarjetas de circuitos eléctricos

Figura 1. Modelo para el laboratorio remoto de comunicaciones

Fuente: <http://hdl.handle.net/20.500.12404/517>

Juárez et al (2016) aplicó una interfaz de comunicación remota entre un sistema clasificador de ondas cerebrales y un robot móvil (p. 1). Este trabajo propone llevar a cabo un control remoto de un robot móvil por medio de un clasificador de ondas cerebrales, obteniendo comandos de movimiento a través de una red VPN al ser esta una tecnología libre lo convierte en una herramienta más robusta y confiable.

Tovar (2003) desarrolló una aplicación local en lenguaje de programación JAVA, que permite manejar la interfaz Science Workshop 750 de Pasco y adquirir

los datos obtenidos por los sensores digitales conectados a ella aplicado en el Laboratorio de Física Básica de la Universidad Nacional sede Manizales, Colombia. (p.3)

Edel (2015) implementó una interfaz web para laboratorios Remotos en dispositivos móviles (p.6). Con el fin de que los estudiantes puedan tener el acceso a los laboratorios de forma remota y obtener experiencia práctica.

### 1.1.2 Educación a Distancia

Para Garduño (2005), “la educación a distancia resulta un campo apropiado para comprender los desafíos actuales de las entidades educativas frente a la sociedad de la información y el conocimiento, así como ante las tecnologías y nuevos escenarios que reclama la sociedad actual. Por lo anterior, conviene contemplar que la educación a distancia es una expresión imprecisa a la que se pueden atribuir (y de hecho se atribuyen) muy diversos sentidos. El estudio de la teoría de la educación a distancia, los elementos que se precisan para llegar al aprendizaje de esta naturaleza y la utilidad de las TIC representan fenómenos de análisis necesario para comprender con certeza las posibilidades que puede tener la educación a distancia en la educación del nuevo siglo”. (Garduño, 2005, p.3)

En la figura 2 se representa un esquema de plataforma de educación a distancia la cual consta de alumnos y tutores que hacen uso de la red para compartir información a través de internet.

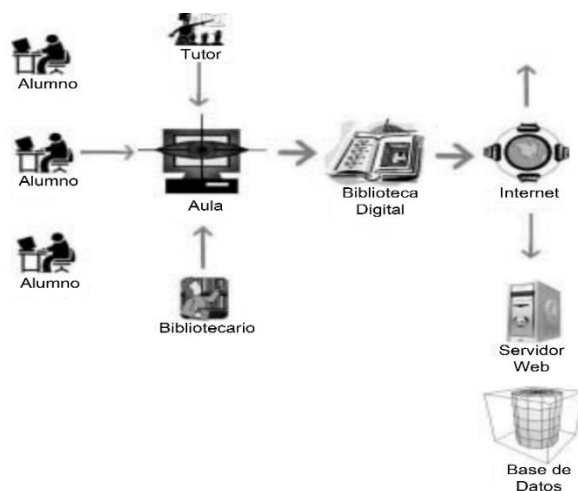


Figura 2. Plataforma de educación a Distancia  
Fuente: Garduño V.(2005). Enseñanza virtual sobre la organización de recursos

Para Roesler et al (2015), “en el Perú, la educación a distancia se ha ido posicionando como una modalidad de estudios cuyas estrategias metodológicas

han ido migrando también a la modalidad presencial. Su validez y competencia, dentro del sistema educativo peruano no ha estado alejado del debate, sin embargo como modalidad de estudios se encuentra fundamentada en el Artículo 27° de la vigente Ley General de Educación N°28044 y en la autonomía universitaria que sustentaba el artículo 4 de la extinta ley universitaria N° 23733 y que fue derogada para dar paso a la actual Ley universitaria N° 30220 vigente desde el año 2014 y que en su artículo 47 sostiene la vigencia de la educación a distancia”. (p.63)

Según la Resolución Viceministerial N°105-2020-Minedu expone lo siguiente: Disponer que, en el marco de lo establecido en el artículo 14 del Decreto Supremo N° 094-2020-PCM, el personal de las universidades públicas y privadas puedan ingresar a los locales de las sedes y filiales de las mismas, con el fin de que, de manera excepcional, realicen determinadas actividades que faciliten la continuidad y mejora en la calidad y oportunidad del servicio educativo que se viene prestando de manera no presencial, entre las cuales se encuentran las siguientes:

- a. Recojo de libros, cuadernos de trabajo, guías, archivos informáticos y materiales educativos en general.
- b. Mantenimiento, reparación y mejora de las plataformas informáticas y tecnologías de la información de la universidad, que tengan por finalidad garantizar la conectividad de los docentes y los estudiantes, así como el acceso de los mismos a los diversos servicios de la universidad.
- c. Mantenimiento del equipamiento de laboratorios o instalaciones que pudieran sufrir deterioro o pérdida por causa de la suspensión de actividades presenciales, dispuesta por las disposiciones vigentes.
- d. Desarrollar las acciones que resulten necesarias para atender las investigaciones y actividades de los laboratorios que venían desarrollándose, vinculadas a la prestación del servicio educativo, a fin de evitar que los mismos se perjudiquen.
- e. Otras, que la universidad considere necesarias para garantizar la continuidad, calidad y oportunidad del servicio educativo no presencial, siempre que cumplan con las características de ser necesarias y, que no puedan ser realizadas de manera remota.

### 1.1.3 Estación de Automatización de Procesos Industriales

“El termino automatización cubre la realización automatizada de operaciones comerciales e industriales. Casi todas las fases del trabajo productivo se incluyen en este campo. Desarrollada en el área de la manufactura, el alcance de la automatización se ha extendido para incluir objetivos tan diversos como la agricultura, construcción de buques, transporte y educación. A través del uso de ordenadores, la automatización se ha aplicado a tareas administrativas, cálculos matemáticos, asociados con investigaciones científicas, control de proyectiles, clasificación y evaluación de datos, incluso en la compleja región del diagnóstico médico”. (Seippel, 2003, p.328)

Para Martínez (2010) “un proceso esta automatizado cuando funciona solo, es decir, sin intervención del ser humano”. (p.205)

En la figura 3 se representa un ejemplo de un sistema automatizado, mediante un programa realizado en FBD. Las cargas conectadas a las salidas del PLC no deben sobrepasar los valores soportados por las características de los elementos internos que activan o desactivan dichas salidas. Para todo el conexasionado habrá que tener en cuenta los datos que facilitan los fabricantes.

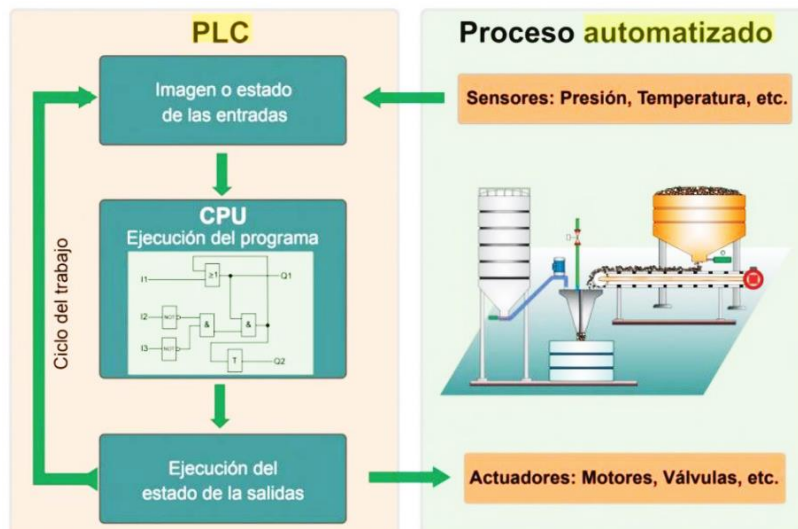


Figura 3. Proceso Automatizado

Fuente: Nuevo G. (2020). Montaje y mantenimiento eléctrico-electrónico

“Los tipos de procesos industriales de acuerdo a su fabricación permiten transformar las materias en productos acabados. El control de estos procesos automáticos se puede realizar mediante la tecnología de lazo abierto o lazo cerrado y su aplicación industrial en función de la evolución del proceso con respecto al tiempo, se clasifica en tres tipos:

- **Proceso Discreto:** Se basa en la entrada de materia que, a través de varias fases secuenciales de actuación, se obtiene un producto final, es decir, a lo largo de un sistema de fabricación discreto compuesto por varias fases, el producto recibe procesos distintos de cada fase, sin embargo, el proceso es siempre el mismo para todos los productos.
- **Proceso Continuo:** Los procesos continuos se basan en la constante entrada de elementos y salida de productos del sistema, regulándose los procesos que actúan sobre el elemento en función de las características de este y el producto final deseado. Es decir, en los procesos continuos el elemento de entrada puede ser de características diferentes y será el proceso el encargado de ajustar dicho elemento al producto final.
- **Proceso por Lote:** Cuando los elementos de entrada en el proceso son un conjunto de piezas distintas que a lo largo del proceso conformaran un único producto, se conoce por proceso por lote o discontinuo. Para la obtención del producto es necesaria la aplicación de distintas operaciones sobre las materias de entrada.

Los procesos discretos y por lotes se usan principalmente en PLC, mientras que los procesos continuos están íntimamente ligados a la automatización industrial de mecanismos de producción y control". (Entrena, 2015, p.21)

En la figura 4 se representa un sistema de envasado de refrescos, es un ejemplo de proceso continuo donde la materia prima es una botella vacía que primero se etiqueta, segundo se llena el refresco y tercero se coloca el tapón, quedando así el producto final.

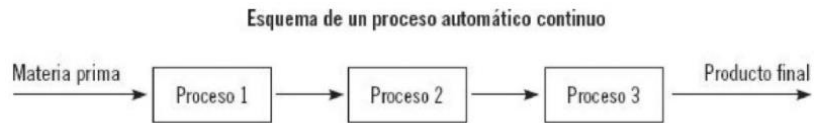


Figura 4. Proceso Automatizado continuo para el etiquetado de botellas  
 Fuente: Entrena G. (2015). Instalación de Equipos y Elementos de Sistemas de Automatización Industrial

“Web-Labai permite a los alumnos interactuar y realizar, desde internet, prácticas de automatización y control de procesos. Consta de un sistema Hardware-Software que permite monitorizar y controlar variables, remotamente desde una página Web, de la maqueta de un proceso industrial del departamento de Ingeniería Electrónica de la Universidad Politécnica “Antonio José de Sucre”, Venezuela. La interacción de un computador servidor, un Controlador Lógico Programable y una cámara IP permite el control y monitorización remota de una maqueta que simula un proceso de fabricación de alimentos”. (De la Cruz et al, 2010, p.1)

En la figura 5 se muestra la estructura general del proceso a nivel lógico, la interconexión entre hardware que son tanto el PLC como la cámara IP y la maqueta de un proceso industrial con el software que contiene a la aplicación servidor-cliente.

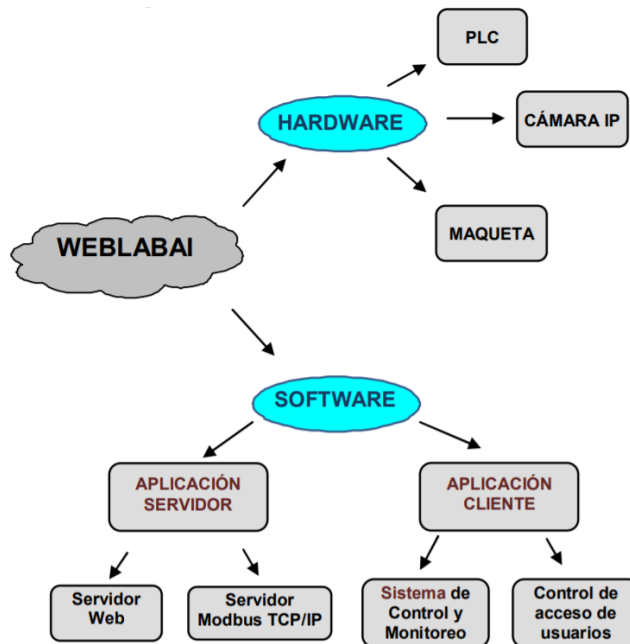


Figura 5. Esquema General del Proceso  
Fuente: De la Cruz et al (2010). Web-Labai: Laboratorio Remoto de Automatización

En la figura 6 se representa la interfaz gráfica que fue desarrollada en java, de esta manera el alumno accede a la plataforma, a través de cualquier parte del internet.

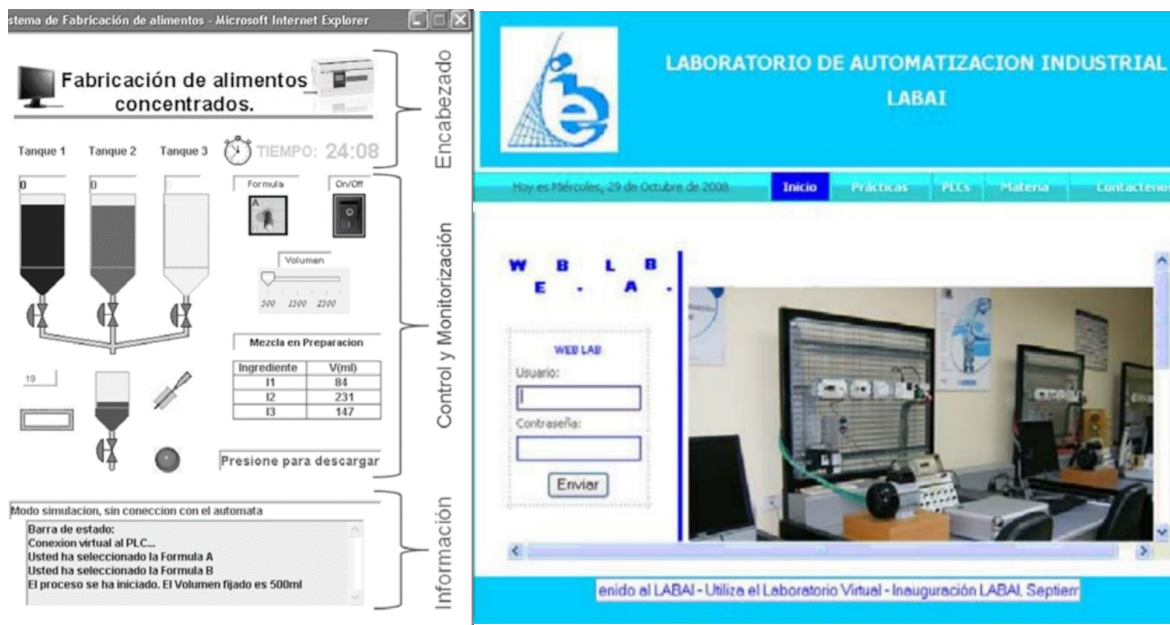


Figura 6. Ventana de Control y Monitorización  
Fuente: De la Cruz et al (2010). Web-Labai: Laboratorio Remoto de Automatización



### 1.1.4 Marco Teórico Específico

Se revisaron publicaciones en Revistas y Journals clasificados dentro de los cuartiles Q1 y Q2 de acuerdo al Ranking Scimago, los cuales se resumen a continuación:

Tobarra et al (2015), en su investigación *“Creación de Laboratorios Remotos personalizados usando la Deconstrucción”* publicado en el Journal IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, el cual trata sobre el empleo de laboratorios remotos personalizados que adquieren gran importancia para la comunidad docente en la educación a distancia, debido a que los profesores tienen la necesidad de utilizar equipos reales a través del Internet, que se adapta a las aptitudes de los estudiantes. Describió cómo se ha llevado a cabo la adaptación de laboratorios científicos reales basados en OpenSocial para crear laboratorios como servicio, que permiten a los profesores crear escenarios versátiles en función del proceso de aprendizaje/enseñanza. Esto es posible gracias a su deconstrucción, que puede verse como un conjunto de servicios ubicados en un contenedor específico de la Web. Concluyó que la creación de laboratorios remotos personalizados es de verdadero interés, debido al hecho de que los conferenciantes podrán adaptarlos al proceso de aprendizaje/enseñanza. Esto es incluso más notable en la educación a distancia y, lo que es más, puede mejorar las habilidades científicas de los jóvenes y aumentan su voluntad de convertirse en científicos cuando llegan a la edad adulta. Por estas razones, este trabajo ofrece varios laboratorios de energía renovable personalizados, que se basan sobre el concepto de deconstrucción.

Por lo indicado en la investigación de Tobarra et al, él usó laboratorios científicos reales y los convirtió en laboratorios remotos con la finalidad de que los estudiantes puedan mejorar sus habilidades científicas con la educación a distancia.

Jernigan et al (2009), en su investigación *“Implementar una experiencia de laboratorio remoto en un programa de grado de ingeniería conjunta: Levitación aerodinámica de una pelota de playa”* publicado en el Journal IEEE Transactions on Education, en esta investigación detalla la implementación de un laboratorio remoto en un curso de sistemas de control a distancia utilizando hardware y software fácilmente disponibles. El experimento físico consiste en una pelota de playa y un soplador de CC; el objetivo del control es hacer que la altura de la pelota

de playa levitada aerodinámicamente siga una trayectoria de referencia manipulando el voltaje del soplador. MATLAB/Simulink junto con xPC Target sirven como plataforma de control, mientras que Microsoft NetMeeting y el equipo estándar de videoconferencia por Internet se utiliza para interconectar a los estudiantes de enseñanza a distancia con el equipo de laboratorio. Tanto los estudiantes locales del campus de la Universidad Estatal de Carolina del Norte en Raleigh como los estudiantes a distancia de la Universidad de Carolina del Norte en Asheville completaron el experimento de laboratorio. Se concluyó que esta tecnología permitió a los estudiantes a distancia implementar, ejecutar y evaluar el rendimiento de sus diseños de control en tiempo real. En el caso del laboratorio remoto, se requería que un operador local posicione la bola en la boquilla del soplador antes de cada arranque del mismo. La interacción personal con el operador local y el profesor del curso es un componente fundamental de la educación a distancia; no se recomienda sacrificar la interacción personal en áreas de los laboratorios totalmente automatizados. El experimento de levitación de la bola proporcionó una experiencia atractiva tanto para los estudiantes locales como para los de educación a distancia. Los estudiantes se sintieron simultáneamente intrigados y desafiados por la compleja naturaleza del sistema, que estimuló muchos pensamientos. Las encuestas mostraron que los estudiantes a distancia veían el proyecto de diseño favorablemente; en algunos aspectos más favorablemente que sus homólogos locales. Las calificaciones de los cursos, incluidas las del proyecto de diseño, también eran similares entre los grupos locales y los de distancia. Como se ha revelado en los estudios, un inconveniente del método es que la baja resolución del vídeo dificultaba la visualización clara de los gráficos en tiempo real transmitidos desde la pantalla de la computadora local. Los métodos descritos en el presente documento pueden extenderse fácilmente a otros proyectos de laboratorio a distancia y a otros cursos. Los autores han utilizado el mismo equipo informático polivalente (computadoras personales, tarjetas de adquisición de datos, fuentes de alimentación, etc.) para controlar los proyectos de diseño con péndulos invertidos y balanceo de balas. También se podrían utilizar otras aplicaciones informáticas (conexión de escritorio remoto, red de computación virtual, etc).

Por lo indicado en la investigación de Jernigan et al, él usó Matlab como plataforma de control y NetMeeting para la comunicación remota con la finalidad de que los estudiantes puedan implementar, ejecutar y evaluar el rendimiento de sus diseños en tiempo real.

Gardel V. (2010), en su investigación *“El laboratorio de automatización a distancia utilizando un Grupo de máquinas virtuales”* publicado en el Journal IEEE Transactions on Industrial Electronics, en esta investigación se presentó un laboratorio remoto para la automatización industrial que comprende diferentes fabricantes de controladores lógicos programables (PLC). Esta instalación proporciona un entorno para que los usuarios remotos aprendan muchos temas de automatización mientras utilizan diferentes PLC junto con varios tipos de sensores, actuadores y redes de comunicación industrial. El laboratorio remoto tiene cámaras de protocolo de Internet que proporcionan al usuario remoto una visión del entorno real de cada banco de PLC. La arquitectura del Laboratorio de Automatización Remota se puede aplicar a otras áreas del aprendizaje electrónico. Concluyó que la arquitectura desarrollada permite un uso ininterrumpido de cualquier laboratorio de automatización. Este reduce la depreciación del equipo y aumenta el rendimiento del laboratorio lo que genera un importante ahorro de costes, ya que no es necesario adquirir un banco de PLC para cada estudiante de un curso. Además, abre el abanico de posibles usuarios que pueden estudiar/practicar en el trabajo o en casa, reduciendo la necesidad de asistir al curso físicamente.

Por lo indicado en la investigación de Gardel V., se usó laboratorios de automatización a distancia, usando un grupo de máquinas virtuales y cámaras de protocolo de internet que proporcionan una visión del entorno más real y con esto logró aumentar el rendimiento del laboratorio lo cual generó ahorro en costes.

## 1.2 Definición de términos básicos

- *Anydesk*: Software de comunicación remota que provee acceso bidireccional entre computadoras personales
- *Baudios*: Es la velocidad con la cual se transmiten los bits y se expresa como bits por segundo.
- *Cable MPI*: Multi Point Interface, Interfaz multipunto.
- *Cámara web*: Es un dispositivo capaz de registrar imágenes en movimiento y en tiempo real.
- *Comunicación serial*: Es el proceso de envío de datos de un bit a la vez.
- *Eagle*: Software de Diseño de tarjetas Electrónicas para la Fabricación en Serie.
- *Electroválvulas*: Válvula Electromecánica que se activa por la excitación de su bobina.
- *FBD*: Diagrama de bloques funcionales.
- *Google Meet*: Es una aplicación de videoconferencia de Google, que permite conectarte y colaborar de forma segura desde cualquier lugar.
- *Lcd 16x2*: Liquid Cristal Display, pantalla de cristal líquido de 2 filas y 16 columnas.
- *Matlab*: Es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.
- *Microcontrolador*: Es un circuito integrado Programable.
- *PCB*: Placa de circuito impreso.
- *PIC 16F628A y 16F877A*: Son Circuitos Integrados programables de 18 y 40 pines de conexión respectivamente, estos pertenecen a la familia de microcontroladores Microchip.
- *PLC*: Controlador Lógico Programable.
- *Profibus DP*: es un estándar de red digital de campo abierto que se encarga de la comunicación entre los sensores de campo y el sistema de control.
- *Puerto COM*: es un puerto de comunicación serial que permite enlazar periféricos y la computadora.
- *Sensores Capacitivos*: Sensor que reaccionan ante metales y no metales cambiando su estado lógico de 0 a 1.

- *Sensores Fotoeléctricos*: Sensor que responde al cambio en la intensidad de luz cambiando su estado lógico de 0 a 1.
- *Sensores Inductivos*: Sensor que detecta materiales ferrosos cambiando su estado lógico de 0 a 1.
- *Sensores Magnéticos*: Sensor que responde ante un imán cambiando su estado lógico de 0 a 1.
- *Tabla ASCII*: American Standard Code for Information Interchange, Código Estándar Americano para el Intercambio de Información.
- *Tia Portal*: Totally Integrated Automation Portal, Portal de Automatización Totalmente Integrado.

## **CAPÍTULO II: METODOLOGÍA DE DESARROLLO DEL TRABAJO PROFESIONAL**

En este capítulo tratará sobre la delimitación temporal y espacial del presente trabajo de suficiencia profesional además se presenta la determinación y análisis del problema y posteriormente se trabajó el modelo de solución propuesto.

### **2.1 Delimitación temporal y espacial del trabajo**

Delimitación Temporal: El presente trabajo de suficiencia profesional se realizó en el periodo comprendido entre el mes de Marzo del 2020 a Septiembre del 2020.

Delimitación Espacial: El presente trabajo de suficiencia profesional se realizó en el laboratorio de Control y Automatización en la Universidad Nacional Tecnológica de Lima Sur, en Villa el Salvador, Lima, Perú.

### **2.2 Determinación y análisis del problema**

En la actualidad debido a la pandemia los estudiantes han sido los más afectados debido a que ahora solo llevan clases teóricas y nada de prácticas, porque no pueden tener acceso a los laboratorios de la UNTELS debido a la pandemia. Esto a futuro podría afectar enormemente a los nuevos profesionales que vienen estudiando las diferentes carreras tecnológicas que impliquen emplear laboratorios de forma presencial. Incluso al momento de buscar un empleo el egresado tendrá menos oportunidades de otros egresados que si tuvieron y utilizaron sus equipos. Eso implicaría una gran desventaja para los estudiantes que viene cursando actualmente en la universidad.

Un punto importante es el gran contagio que se viene dando a nivel mundial es por ello más, que se evita todo tipo de clases presenciales. Debido a esto los estudiantes estarían expuestos a los contagios no solo ellos sino también los docentes e incluso el personal administrativo se vería afectado. Generando así una gran cantidad de pérdida de vidas Humanas. E incluso en el ámbito de la educación peruana se ve muy afectada la educación del País en todos los niveles de educación, primaria, secundaria, universitaria y postgrado. Siendo un poco más específico, en los ciclos presenciales no se llegaba a terminar los laboratorios debido a que eran muchos los estudiantes que integraban a un grupo y eran pocos los que llegaban a utilizar la estación automatizada de procesos industriales.

La pantalla HMI casi no se usaba debido a que en los cursos se llegaba a lo mucho a controlar los PLC. Esto es importante porque actualmente todas las

industrias poseen pantallas HMI para controlar sus procesos y si el estudiante no sabe cómo operarlo o manejarlo estaría en desventaja frente a la competencia.

### 2.3 Modelo de solución propuesto

De acuerdo a como se muestra la figura 7 en la primera etapa de la validación de la conexión remota a distancia de Docentes y Estudiantes trata sobre la arquitectura de la conexión y las herramientas de software como Google Meet y Anydesk que son necesarias para establecer la comunicación. Como segunda etapa se elaboró el diseño y desarrollo de una Interfaz remota en Matlab que permite establecer la conexión entre los docentes y estudiantes con la estación de automatización de procesos industriales, los cuales consisten en la programación de las cuatro cámaras web y la interfaz como medio de enlace. Como tercera etapa sobre el desarrollo y prueba de la tarjeta de adquisición electrónica para el control del sistema automatizado consiste en la implementación del desarrollo de la placa electrónica con los microcontroladores PIC 16F628A y 16F877A que son capaces de comunicarse con el PLC S7-300 a través de sus puertos de entrada y salida mediante el conector DB37 y a la computadora por el puerto USB. Como última etapa, pero no menos importante será la validación y ejecución del control del sistema Automatizado de Procesos Industriales compuesto por cuatro Módulos de Trabajo que son los siguientes: Sistema de Separación, Sistema de Montaje, Sistema de Verificación, Sistema de Manipulación y por último una pantalla HMI, OP177B, todo el sistema en conjunto es gobernado por el PLC y fue programado en TIA portal con un ejemplo didáctico para el aprendizaje y correcto funcionamiento de todo el sistema.

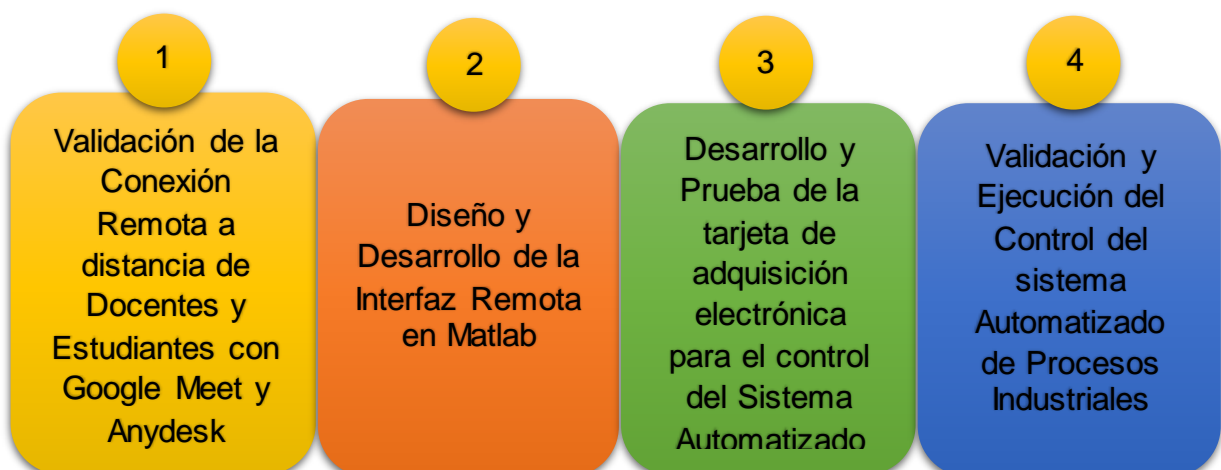


Figura 7. Etapas de la Elaboración del Proyecto  
Fuente: Elaboración Propia

En la figura 8 se muestra el diagrama visual del trabajo de suficiencia profesional en la cual hace referencia a los módulos utilizados en la elaboración del trabajo, el mismo está conformado por las cuatro cámaras web, los cuatro módulos automatizados de procesos industriales, adicionando a ello la pantalla HMI y todo ello controlado por el PLC S7-300 en conjunto con la interfaz remota logran el manejo total del sistema. La interfaz en Matlab se utiliza como medio de comunicación entre la persona y el sistema. Esta herramienta puede ser manejada a través de Google Meet y Anydesk para un control remoto o a distancia. Poniendo énfasis en que las cámaras mostraran en tiempo real el funcionamiento del proceso programado.

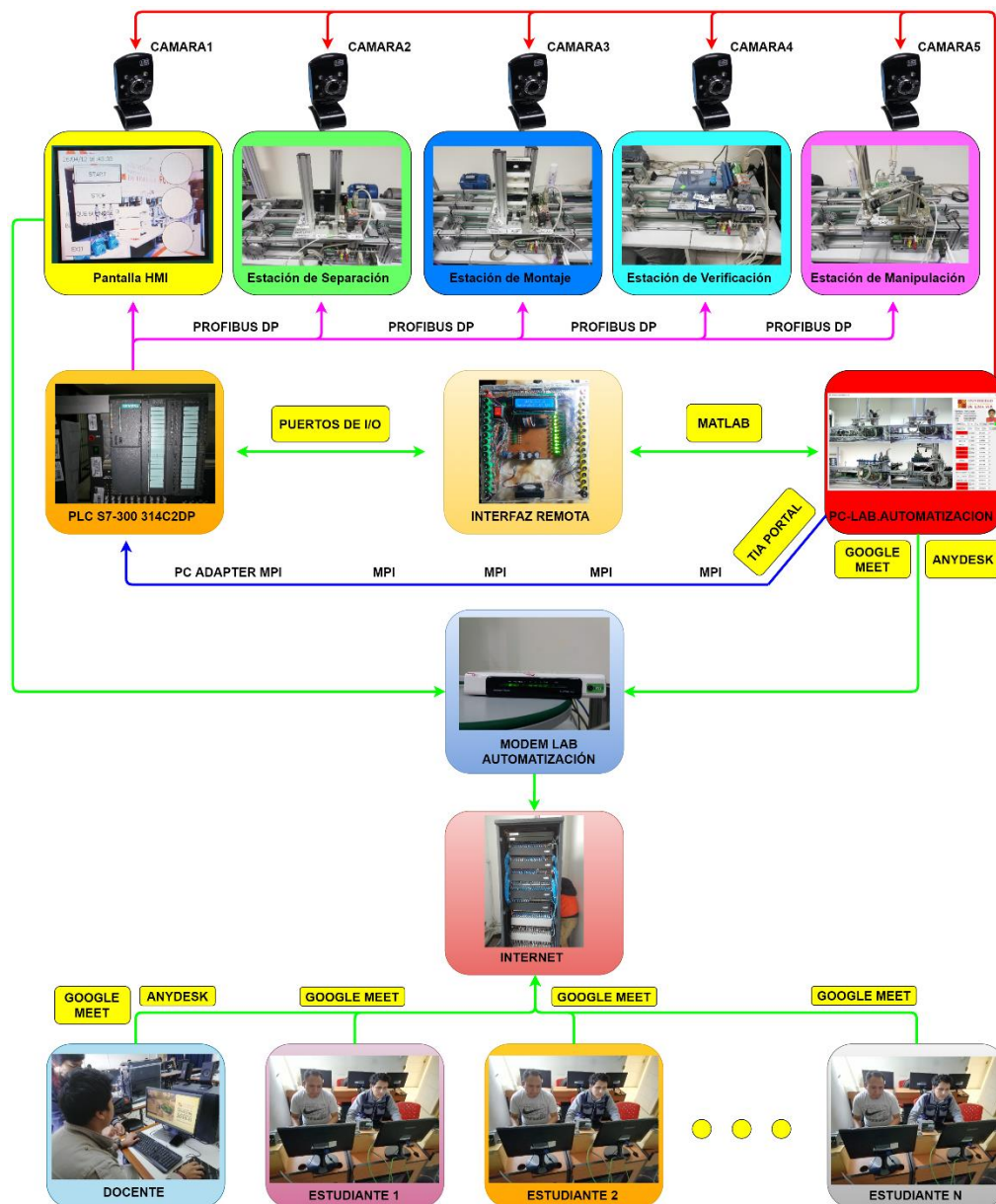


Figura 8. Diagrama Visual del Trabajo de Suficiencia Profesional  
Fuente: Elaboración Propia



### **2.3.1 Validación de la Conexión Remota a Distancia de Docentes y Estudiantes con Google Meet y Anydesk**

En este apartado se explica el proceso aplicado para establecer la comunicación de forma correcta entre los Docentes y Estudiantes del curso para evitar problemas que puedan suceder en el desarrollo de mismo. Se indica lo siguiente, para que la conexión pueda tener éxito tanto los docentes como estudiantes deben contar con una computadora con conexión a internet, no necesariamente potente, puede ser también una computadora básica y con un ancho de banda mínimo recomendado de dos megas de velocidad para no tener inconvenientes en la visualización del mismo.

En la figura 9 muestra el diagrama de flujo que debe seguir el docente para lograr el objetivo, conectarse a la interfaz remota y realizar en paralelo la programación en Tia portal que será necesaria para simular el proceso de funcionamiento del sistema automatizado.

Lo primero que debe realizar el docente es iniciar sesión y crear una reunión en Google Meet, copiar esa dirección y compartirla con los estudiantes del curso y con el asistente de laboratorio, con la finalidad de que todos estén incluidos en la misma sala en Google Meet. Es por esta sesión que se transmitirá el sonido que realiza el sistema automatizado en el funcionamiento del proceso, haciendo que el laboratorio se convierta en un entorno más real y además el docente tendrá que compartir su pantalla para que todos los estudiantes puedan ver el funcionamiento del sistema en sí. Luego una vez iniciado Google Meet deberá iniciar sesión en Anydesk, este software si tendrá que descargarlo de su página oficial, una vez descargado al abrirlo mostrará dos parámetros muy importantes una es la propia dirección de la estación de trabajo y un espacio en blanco donde se pondrá la dirección del puesto de trabajo al cual se quiera conectarse y realizar la comunicación Remota a través de internet.

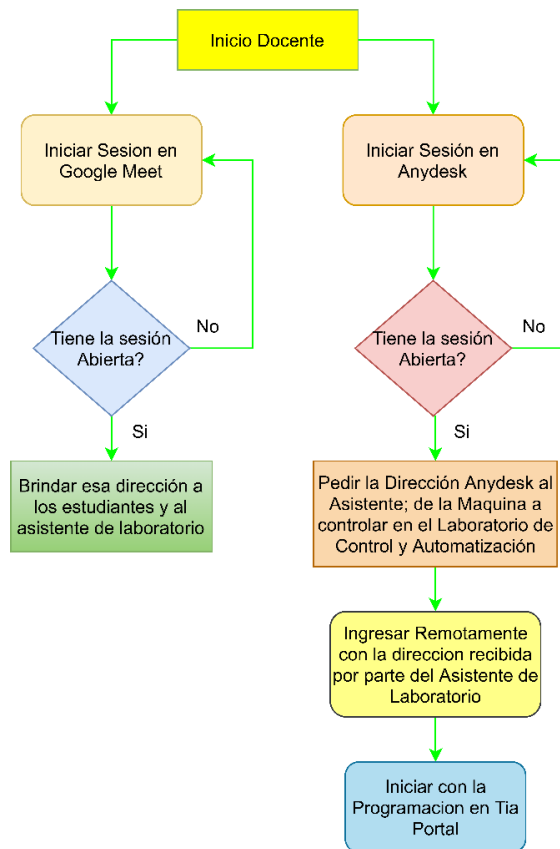


Figura 9. Diagrama de Flujo del Procedimiento de conexión del Docente  
Fuente: Elaboración Propia

En la figura 10 se muestra el diagrama de flujo de la conexión del estudiante. El estudiante inicia solicitando la dirección de la reunión en Google Meet al docente, si en caso no pueda comunicarse con el docente o compañero de clase podría consultarle al asistente de laboratorio para que le brinde dicha dirección. Una vez ingresado a la reunión solo prestar la debida atención mientras el docente explica la solución y manejo del problema.

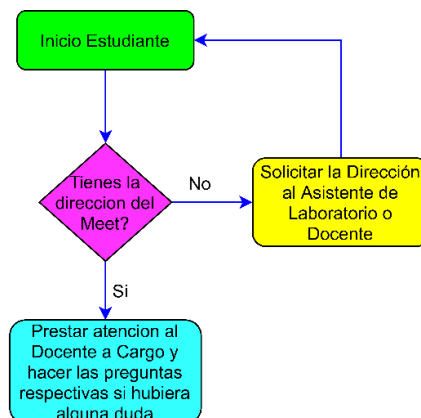


Figura 10. Diagrama de flujo de la conexión del estudiante  
Fuente: Elaboración Propia

Se empleo el uso de Anydesk porque es con esta herramienta que se puede visualizar en tiempo real y controlar de forma remota a una resolución de 720x480 pixeles eso es más que necesario para transmitir video en tiempo real.

La calidad de la visualización también dependerá de la resolución de las cámaras, eso es un punto importante en la proposición de que cámaras comprar si es una cámara HD o FHD.

En la figura 11 se aprecia los formatos que admite nuestra cámara web las cuales pueden ser configuradas de acuerdo al formato que admiten. Para averiguar dichos formatos aplicamos el comando `>>imaqtool` en la ventana de comandos de Matlab como se muestra a continuación.

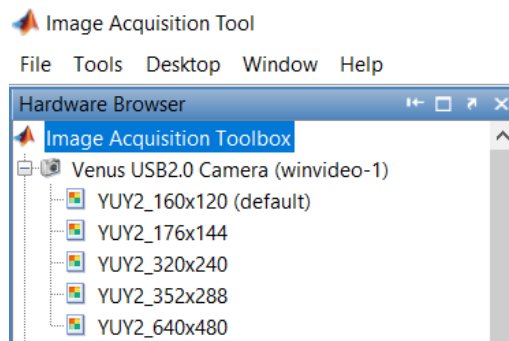


Figura 11. Visualización de los formatos de la cámara empleando el comando `imaqtool`  
Fuente: Elaboración propia utilizando el software Matlab

En la figura 12 se muestra el resultado de la conexión Google Meet y Anydesk de la Interfaz Remota, la cual muestra en su imagen la transmisión en tiempo real del funcionamiento del Módulo de automatización de procesos industriales. En la pantalla se muestra la estación de Manipulación.

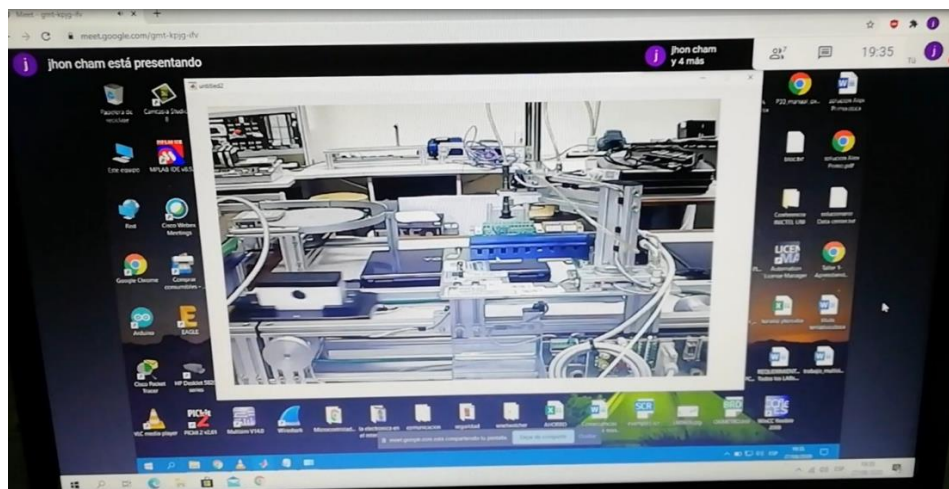


Figura 12. Resultados de la conexión Google Meet y Anydesk de la Interfaz Remota  
Fuente: Elaboración Propia

### **2.3.2 Diseño y Desarrollo de la Interfaz Remota en Matlab**

En la figura 13 se muestra el diagrama de funcionamiento de la interfaz remota, al iniciar la interfaz pedirá que ingrese su usuario y contraseña, las personas quienes iniciarán la interfaz serán el administrador, el docente o estudiante que anteriormente fueron registrados ya sea por el administrador o docente. En el caso del docente tendrá la opción de agregar estudiantes a su clase, los cuales pueden registrarse ingresando sus datos como nombres, apellidos, código, foto y otros datos adicionales que servirá como medio probatorio para la presentación de sus informes, con esto demostrarán que fueron ellos los que hicieron el trabajo y así se pueda evitar plagio en la presentación de los trabajos o si fuera el caso que ellos eviten compartir sus trabajos con otros estudiantes. Una vez iniciado la interfaz remota se procede a sincronizar la interfaz con el usuario para esto el estudiante debe ingresar su usuario y contraseña si este no fuera correcto tiene que contactarse con el docente o administrador, una vez sincronizado el usuario, ingresa a la plataforma de inicio en el cual debe seleccionar las cámaras que sean necesarias para visualizar la estación automatizada en tiempo real. También, se debe seleccionar la velocidad a la cual trabaja el sistema con Matlab y por último, seleccionar la dirección que posee el dispositivo. Una vez realizado todo estos pasos se inicia el puerto serial y tiene la opción para pausar la comunicación con la tarjeta electrónica en caso sea necesario, es una opción adicional que brinda comodidad a la hora de hacer simulaciones en tiempo real ya sea pausar el proceso, para realizar una reprogramación en la ejecución de las tareas a realizar de la estación automatizadas de procesos industriales.

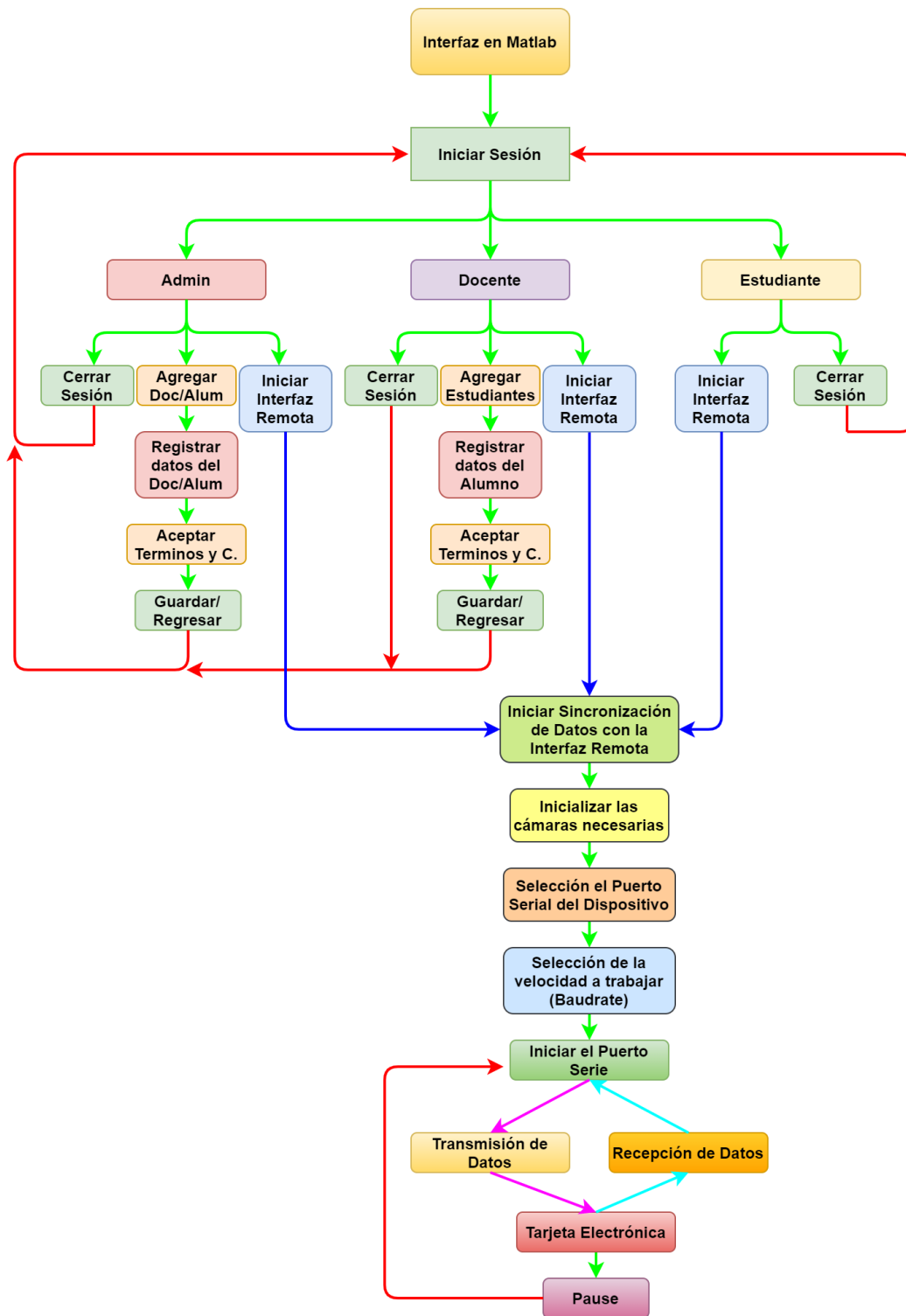


Figura 13. Diagrama de Funcionamiento de la Interfaz Remota  
Fuente: Elaboración Propia

En la figura 14 se muestra la página principal de la interfaz remota en Matlab, donde el estudiante o docente ingresa su usuario y contraseña previamente registrado en la interfaz ya sea por el administrador o el docente, en caso sea un nuevo docente necesariamente debe ponerse en coordinación con el administrador

y pueda registrarse, una vez registrado el docente tendrá la capacidad de agregar a sus estudiantes sin la necesidad de contactarse con el administrador.



Figura 14. Página principal de la interfaz Remota en Matlab (iniciar sesión)  
Fuente: Elaboración propia utilizando el software Matlab

El desarrollo de la interface contiene un sistema de identificación y registro para que los estudiantes puedan registrarse y hagan uso de la plataforma como se muestra en la figura 15. Si se observa la contraseña se puede notar que solo se observa unos cuadrados, eso es porque la fuente en si encripta la contraseña para que esta no sea vista para los demás que están visualizando la pantalla ya sea porque se esté compartiendo pantalla con la clase, eso es para el caso que algún docente o estudiante intente ingresar a la interfaz y así no sea víctima de usurpación por parte de otros estudiantes. En el caso de los estudiantes su usuario será determinado por su código y en el caso del docente puede utilizar su DNI como usuario para su registro.



Figura 15. Ingreso de Usuario  
Fuente: Elaboración propia utilizando el software Matlab

Para que el estudiante pueda iniciar sesión se necesita que el docente a cargo del curso pueda registrarlo como es el caso de la figura 16, caso contrario no podrá utilizar la interfaz remota. Es necesario que el estudiante tenga una foto digital para su registro en formato jpg. Algo similar a como se presenta en la figura 16.



Figura 16. Autenticación del estudiante realizado de forma exitosa  
Fuente: Elaboración propia utilizando el software Matlab



El sistema de agregación de estudiantes y docentes consta de varios parámetros tales como nombres, apellidos, foto, correo, curso, docente, código y contraseña como se muestra en la figura 17. Debe tomarse en cuenta que el docente solo puede agregar a estudiantes mas no a profesores, en el caso de añadir profesores solo el administrador podría hacerlo, debido a que son los docentes los únicos que puedes agregar estudiantes. Para definir los cursos existe una lista predefinida que muestra los cursos de control II, Automatización de procesos Industriales y Automatismo Programable, en esta opción el estudiante solo seleccionaría el curso en el cual este matriculado.



Figura 17. Sistema de Agregación de Estudiantes y Docentes  
Fuente: Elaboración propia utilizando el software Matlab

En la figura 18 se muestra una ventana que contiene un mensaje de alerta, justamente por no aceptar los términos y condiciones. Para subsanar este error el estudiante tendría que darle clic en “ok” y seleccionar la casilla donde indica aceptar términos y condiciones. Posterior a ello, se mostraría un aviso importante, el cual es necesario que el estudiante lo lea luego debe dar clic en “ok” y finaliza guardando los datos registrados por el estudiante.





Figura 18. Mensaje de Alerta por no darle acepto haber leído los términos y condiciones  
Fuente: Elaboración propia utilizando el software Matlab

Para que el estudiante y docente pueda hacer uso de la interfaz remota requiere de aceptar los términos y condiciones del programa como se muestra en la figura 19. Para finalizar una vez terminada la lectura debe dar clic en “ok” para continuar con el procedimiento.



Figura 19. Términos y Condiciones de la Interfaz Remota  
Fuente: Elaboración propia utilizando el software Matlab

Una vez registrado se muestra el mensaje “Se almacenó correctamente los datos del estudiante/docente” como se observa en la figura 20. Este mensaje es

para verificar que los datos del estudiante o docente se guardaron de manera exitosa.



Figura 20. Almacenamiento Correcto de los Datos del Usuario  
Fuente: Elaboración propia utilizando el software Matlab

Para el registro del docente o estudiante se requiere cargar una foto digital, para esto se presiona el botón “foto”, ubicado en la interface de registro de datos. El formato de imagen con el que trabaja la interfaz es el formato .jpg, como se muestra en la figura 21. Donde se muestra la ventana de importar imagen se selecciona la imagen y se procede a abrir dicha imagen, con este paso estaría cargado la imagen completamente a la interfaz.

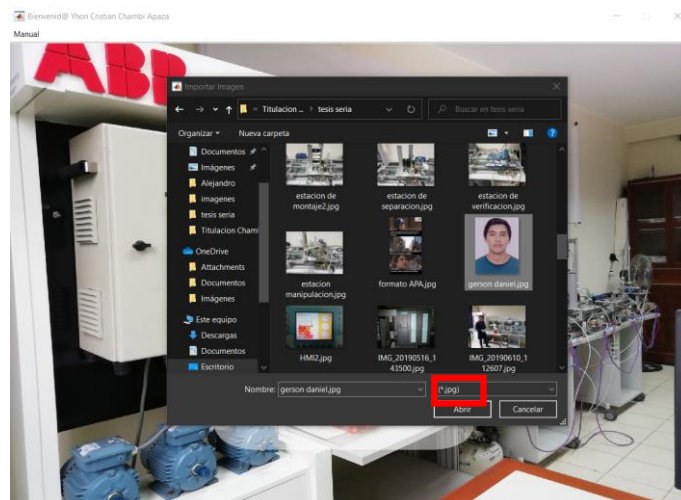


Figura 21. Importar imagen para la visualización del usuario en pantalla  
Fuente: Elaboración propia utilizando el software Matlab

Una vez guardado los datos del estudiante se presiona la opción regresar, dando lugar a la apertura de la interfaz inicial donde se registra sesión nuevamente y al finalizar se prueba con la cuenta creada, como se muestra en la figura 22. También hay que acotar que cuando el estudiante este registrado se habilitará una barra en la parte superior de la interfaz, haciendo mención a los manuales que dicho sea de paso son necesarios para comprender el funcionamiento de cada uno de los módulos de la estación automatizada de procesos industriales.



Figura 22. Inicio de sesión del usuario Gerson Daniel Diestra Ninahuilca  
Fuente: Elaboración propia utilizando el software Matlab

En la figura 23 se muestra la interfaz remota elaborada en Matlab la cual esta subdividida en 5 partes, estas son las siguientes: Datos del estudiante(rojo), inicialización de las cámaras(verde), inicialización del puerto serial(azul), activación de salidas y censado de las entradas de la tarjeta electrónica(amarillo) y por último la sección de visualización de las cámaras web(lila). Cabe acotar que las opciones que se muestran en la interfaz; por ejemplo, start, stop, válvula, pivote, etc, son editables debido a que los problemas para cada caso pueden variar de acuerdo a lo que designe el docente a cargo del curso.



Figura 23. Inicialización de la interfaz remota  
Fuente: Elaboración propia utilizando el software Matlab

### Inicialización de las Cámaras web en la Interfaz en Matlab

Para el reconocimiento de las cámaras web en Matlab se requiere instalar el packet tool denominado Image Acquisition Toolbox Support Package for OS Generic Video Interface, sobre todo para que reconozca los comandos que se utilizaron en la inicialización de cada una de las cámaras web. Dichas cámaras se inicializan teniendo como resolución estándar la calidad de 640x480 pixeles debido a que son cámaras HD. Para su visualización dentro de una ventana denominada axes se hace un escalamiento en las dimensiones del video para que se muestre exactamente en esa ventana.

La interfaz cuenta con cuatro menús de selección, cada una para la elección de una sola cámara web, las opciones que maneja cada una de ellas son: “HMI” que hace referencia a la pantalla HMI OP177B; “separación” que hace referencia a la estación de separación; “montaje” que hace referencia a la estación de montaje; “verificación” que hace referencia a la estación de verificación y por último “manipulación” que hace referencia a la estación de manipulación, para así encender las cámaras a utilizar las cuales se pueden seleccionar de acuerdo a las necesidades del estudiante.



Los pasos que debe seguir el estudiante para que la interfaz remota logre inicializar el puerto serial del microcontrolador PIC 16F877A se detalla a continuación. Para esto, primero que nada se debe verificar la dirección del puerto serial ingresando a la opción de administrador de dispositivos del ordenador. Una vez ahí se selecciona la parte denominada Puertos (COM y LPT). Se debe verificar que en su descripción diga USB-SERIAL CH340 (COM4). Con esto da por entendido que el dispositivo fue reconocido exitosamente y ahora se puede seleccionar el puerto COM para inicializar la comunicación entre Matlab y la tarjeta electrónica. Cabe aclarar que el mismo dispositivo puede tener otra dirección COM en otro ordenador siendo para este caso en particular el COM4, incluso pueden llegar a tener otros puertos COM abiertos sobre todo porque se utiliza también el puerto COM del MPI que sirve para subir la programación al PLC, es por eso que se debe verificar todo eso mucho antes. Una vez resuelto el tema de la dirección COM se procede a la selección de la velocidad a la cual el dispositivo se conectará al ordenador, esta tarjeta electrónica fue configurada previamente en su programación para trabajar a una velocidad de 115200 baudios, los cuales pueden ser modificados para trabajar a una velocidad de 9600 baudios por tanto no generaría inconveniente alguno, a excepción de que la comunicación sea un poco más lenta en el segundo caso. Una vez seleccionado la velocidad de 115200 baudios en la interfaz remota, se procede a iniciar el puerto con el comando fopen el cual inicializa el puerto de comunicaciones serial. Posterior a ello se espera un segundo para que el puerto serial pueda inicializar sin ningún inconveniente y pueda comunicarse de forma exitosa.

Para continuar en la explicación, se necesita saber el tema de la dirección que lleva el dispositivo internamente en la programación. La interfaz en Matlab posee diez direcciones las cuales se pueden utilizar de manera individual, va desde la dirección cero a la dirección nueve, para este caso se tomó la dirección dos, esa sería la dirección del dispositivo en la interfaz. Para obtener una comunicación exitosa se necesita seleccionar la dirección correcta con la que se programó el dispositivo, si en caso se equivocara en la dirección del dispositivo no hará nada la interfaz debido a que no se seleccionó de manera correcta la dirección.

Para el caso, cuando se activa la casilla o también denominado checkbox lo primero que realiza la interfaz es imprimir y enviar a través del puerto serial la dirección del dispositivo, para este caso la dirección es igual a dos. El tipo de dato

con el que se trabaja en Matlab para el envío de datos por el puerto serial es como se muestra a continuación, el término “%s” da entender que el valor que envía a la tarjeta electrónica será de tipo string también llamado cadena de caracteres. Como segundo valor a enviar será el de la dirección del puerto. La asignación que se hizo a cada una de las salidas es correspondiente al orden de las letras del abecedario. Por ejemplo, la dirección “Q0.0” fue asignada a la letra “a” en minúscula, la cual es importante debido a que para desactivar esa misma dirección se utiliza la letra “A” en mayúscula. Y así sucesivamente para cada una de las direcciones hasta la dirección “Q1.6” la cual tiene asignado la letra “o” para encender y la letra “O” para apagar el puerto.

La forma en como la interfaz remota pregunta al microcontrolador PIC 16F877A el estado de cada una de sus entradas es enviando en el primer carácter la dirección del dispositivo, para a continuación enviar la letra “z”, dando a entender al microcontrolador que desea saber el estado de cada una de sus entradas. Luego de esto se realiza la lectura inmediata del puerto serial, debido a que la respuesta es enviada inmediatamente por el microcontrolador PIC y es almacenada en quince variables, donde la primera variable es la letra “a” y así sucesivamente hasta la letra “o”, una vez almacenada toda esa información es procesada por una función condicional para determinar si debe prender o apagar el fondo de color amarillo en la interfaz, de acuerdo al estado lógico que habrá censado el microcontrolador con respecto a cada una de sus entradas.

El código para accionar el color del encendido de los bloques en la interfaz en Matlab, indican que dicha entrada se encuentra en un estado lógico alto o bajo según la determinación de la tarjeta de Adquisición Electrónica. Para ello se empleó una condición; si el valor adquirido es igual al valor “1” del código ascii, que equivale a 49 en número decimal, cambiará a un fondo amarillo indicando que su estado lógico es alto; en caso contrario lo cambiará a fondo blanco, el cual demuestra que su estado lógico es bajo.

En la figura 24 se muestra el montaje de las cámaras web que se encuentran enmarcadas de color rojo separadas una distancia entre cada una de ellas para obtener un mejor ángulo al momento de enfocar cada uno de los módulos automatizados de procesos industriales. Se puede ver que las cámaras se encuentran conectadas al ordenador principal donde se encuentra alojado la

interfaz remota la cual hará uso de las mismas para visualizar los módulos automatizados en tiempo real.

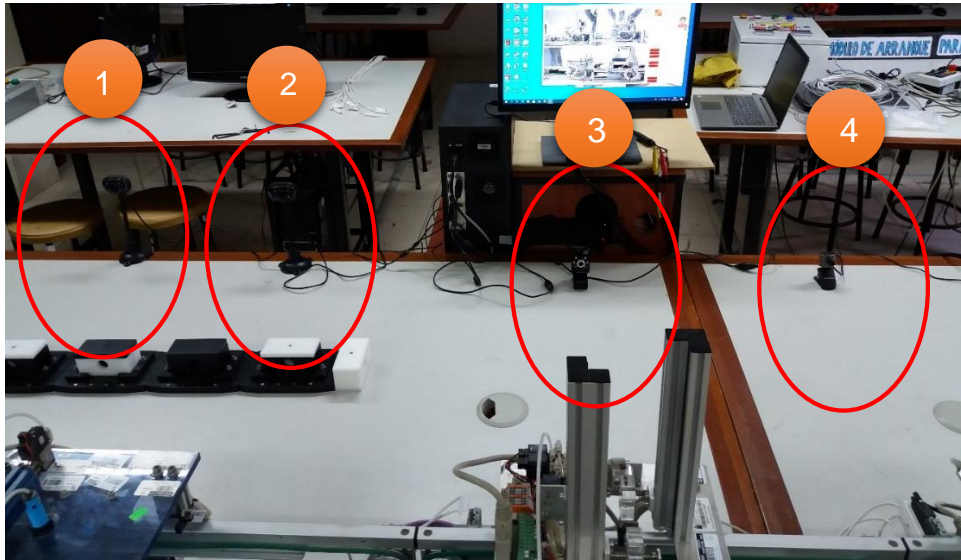


Figura 24. Montaje de las cámaras web en el laboratorio de Control y Automatización  
Fuente: Elaboración Propia

En la figura 25 se aprecia la etapa final cuando se activan las 4 cámaras web colocadas en cada una de las estaciones automatizadas de procesos industriales. Además, se muestra perfectamente la identificación del usuario que hace uso de la interfaz remota la cual da entender que ya se programó en el software TIA Portal y está haciendo uso de la interfaz remota. Activando las diferentes salidas. Como se muestra en imagen, las estaciones están habilitadas porque se visualiza que los indicadores se encuentran encendidos de cada uno de los sensores.

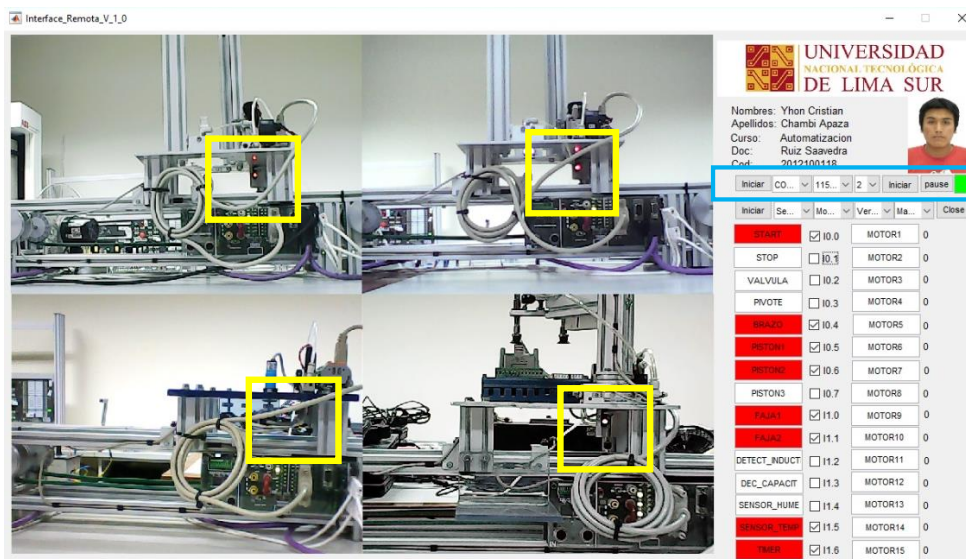


Figura 25. Interfaz Remota Etapa final  
Fuente: Elaboración propia utilizando el software Matlab

### **2.3.3 Desarrollo y prueba de la tarjeta de Adquisición Electrónica para el Control del Sistema Automatizado**

En esta etapa del desarrollo de la tarjeta de adquisición electrónica se detalla tanto el funcionamiento como el acabado y la elaboración física del módulo, para ello implica el manejo de varios software tanto de simulación como de diseño para validar el funcionamiento antes de realizarlo, además de la programación necesaria para el funcionamiento del mismo. En la figura 26 se muestra la explicación del funcionamiento, lo que hace inicialmente es esperar, que valor le envía la interfaz remota para hacer una determinada acción, posterior a ello verifica si la dirección con la que envió la interfaz coincide con la dirección del dispositivo, si ese no es el caso descarta esa orden y nuevamente se pone en modo escucha para recibir otro valor, si en caso este coincide con la dirección decodifica el segundo valor y clasifica que termino fue el que se recibió de acuerdo a eso toma una decisión, si en caso es el valor "z" lo que hace es leer los quince estados de las entradas del microcontrolador y posterior a ello enviárselos a la interface en Matlab en forma de texto plano con la finalidad de que este pueda entenderlo rápidamente y pueda tomar las decisiones necesarias para mostrarlas al usuario final. Luego determina si el valor recibido representa un valor mayor a 80 si fuera mayor entendería que debería preguntar qué puerto es el que se va encender y si por el contrario el valor recibido es menor a 80 debería preguntar qué puerto es el que se va a apagar, tanto al valor numérico como los caracteres hacen referencia a la tabla ascii. Ahora si en caso el valor recibido es "a" por ejemplo, lo que hace es activar cambiando el estado lógico de 0 a 1 la salida "Q0.0" del microcontrolador, considerando que este es el nombre de un puerto del PIC 16F877A en paralelo a esto el microcontrolador PIC 16F628A quien gobierna al LCD 16x2 muestra este cambio de estado. Similarmente ocurre cuando el microcontrolador recibe la letra "A", cabe resaltar que las variables en mayúscula representan el apagado del mismo puerto y las variables en minúscula representan la activación del puerto del PIC. Como son quince las salidas que maneja el microcontrolador, existe una asignación de los puertos con las variables, por ejemplo el encendido va desde la letra "a" hasta la "o" y en el caso del apagado de los mismos va desde la "A" hasta la "O". Hay que aclarar que la comunicación se realiza a una velocidad de 115200 baudios, lo que significa que cada bit enviado por el puerto serial tiene una demora de 8.68 microsegundos, lo cual es muy favorable para que el dispositivo pueda realizar la



acción lo más antes posible y no se sienta el retardo como si lo haría, si se pondría a funcionar a una velocidad menor como el caso de 9600 baudios que posee una demora de 104 microsegundos por cada bit enviado por el puerto serial, concluimos que el primer valor calculado es mucho menor que el segundo, es por ese motivo que se empleó esa velocidad para la transferencia de información entre la tarjeta de adquisición electrónica y la interfaz remota.

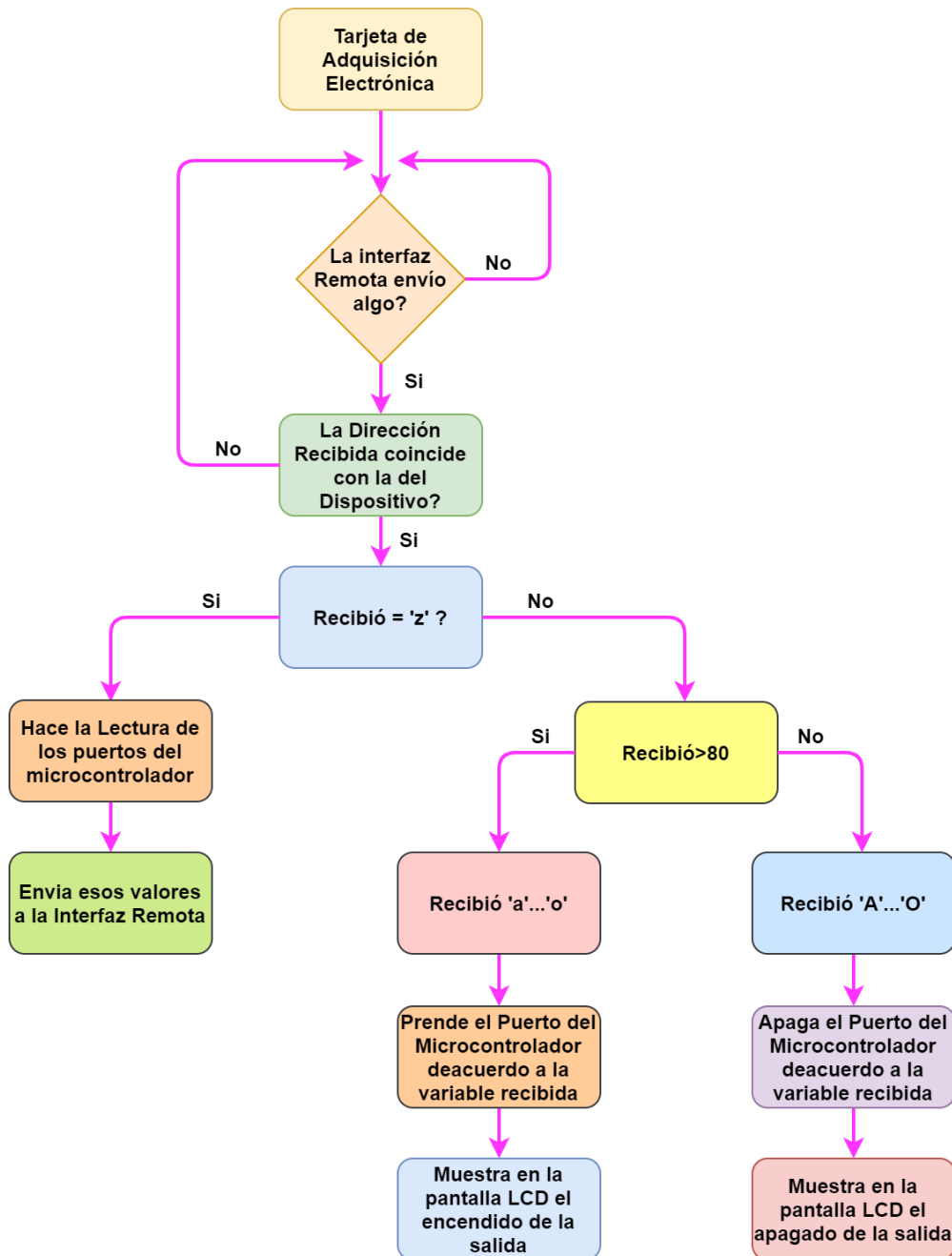


Figura 26. Diagrama de flujo del funcionamiento de la tarjeta de adquisición electrónica  
Fuente: Elaboración Propia

Para el diseño de la tarjeta de adquisición electrónica se pudo desarrollar con diversos microcontroladores, de los cuales se optó por el microcontrolador PIC debido al bajo costo, prestaciones en cuanto a velocidad debido a que este soporta hasta 20Mhz esto se puede apreciar en la tabla 1. Eso implica un menor tiempo en la ejecución de cada una de las instrucciones, para lo cual se trabajó con el software MPLAB para tener una mayor exactitud en cuanto a los tiempos y sincronizar de forma correcta la tarjeta de adquisición electrónica con la interfaz remota en Matlab.

Tabla 1.Comparación de Microcontroladores

	Atmega 16	PIC 16F877a
RAM	1024	368
ROM	16KBytes	14.3KBytes
EEPROM	512Bytes	256Bytes
Puertos I/O	32	33
Oscilador	interno 8MHz externo16Mhz	Externo 20MHz
USART	SI	SI
I2C	SI	SI
SPI	SI	SI
ADC	8 canales 10 bits	8 canales 10 bits
Precio	s/ 15.4	s/ 14

Fuente: Propia

Para trabajar con una alta velocidad de transferencia de datos como lo es la velocidad de 115200 baudios se necesita tener una frecuencia de reloj elevada es por eso más, que se usó el PIC en lugar del atmega debido que puede soportar hasta un oscilador de cristal de hasta 20MHz lo cual de acuerdo al fabricante posee un tiempo de instrucción de hasta 200ns lo cual es muy conveniente debido a que el tamaño de un bit transmitido a una velocidad de 115200 es como hacer 43 instrucciones dentro de la programación eso daría tiempo para poder procesar esa información y determinar una acción. En cuando al tamaño de las memorias tanto como son la RAM y la ROM no son tan necesarias que sean grandes para desenvolverse de forma correcta lo que en si son necesarias es la cantidad de los puertos de entradas y salidas debido a que son estas las que realizarán la comunicación con el PLC.

En la figura 27, se muestra la fuente de Alimentación la cual está compuesta por el integrado LM7805, posee una alimentación de entrada de 24V la cual es regulada a una tensión constante de 5V. Para obtener una salida de voltaje más estable se utilizaron filtros de 2200uF el cual se aplicó tanto a la entrada como a la salida y para evitar el recalentamiento del circuito integrado se implementó un disipador de calor TO-220, además de la adición de un sistema de ventilación externo.

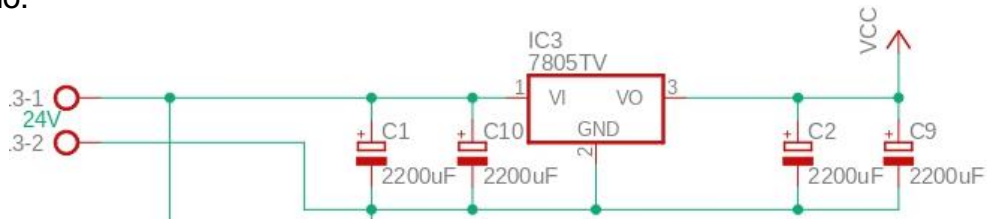


Figura 27. Fuente de alimentación  
Fuente: Elaboración propia utilizando el software Eagle

En la figura 28 se observa el microcontrolador PIC 16F877A el cual controla los puertos de entrada y salida a través de una configuración de resistencias y transistores de potencia (2N2222 y BD140), trabaja a una frecuencia de 20MHz con un cristal de cuarzo externo, funciona a una tensión de 5V DC. Para el funcionamiento del microcontrolador se requiere que el pin 1 este derivado a VCC para su correcto funcionamiento.

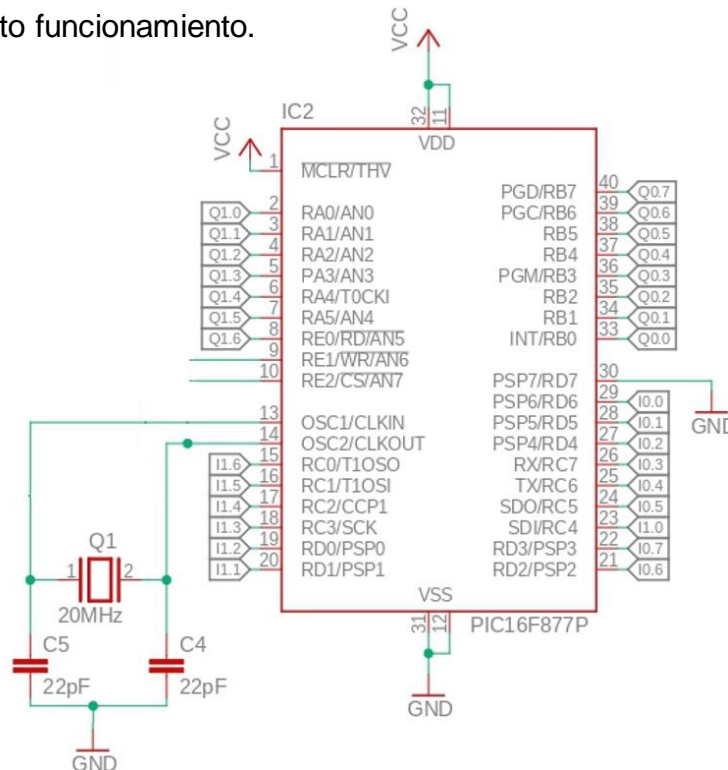


Figura 28. Circuito de manejo de las entradas y salidas del PIC 16F877A  
Fuente: Elaboración propia utilizando el software Eagle

En la figura 29 se visualiza el funcionamiento de la pantalla LCD 16x2 el cual está gobernado por el microcontrolador PIC 16F628A, necesita de una fuente de alimentación de 5V para funcionar, trabaja con la oscilación del cristal de 20 MHz del PIC 16F877A, necesita de un potenciómetro para regular el contraste del display y además se requiere alimentar el led de iluminación en pantalla para visualizar de forma clara lo que se muestra en ella.

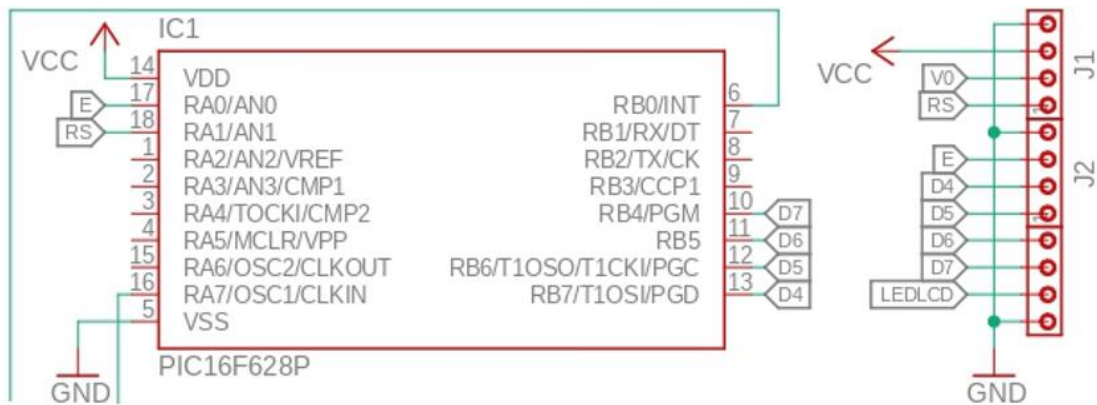


Figura 29. Circuito de manejo del LCD a través del microcontrolador PIC 16F628A  
Fuente: Elaboración propia utilizando el software Eagle

En la figura 30 se muestra el integrado CH340G el cual es el encargado de comunicar la interfaz remota con la tarjeta de adquisición electrónica. Además, trabaja necesariamente con un cristal externo de 12 MHz, este dispositivo transfiere la señal TTL por el USB y viceversa.

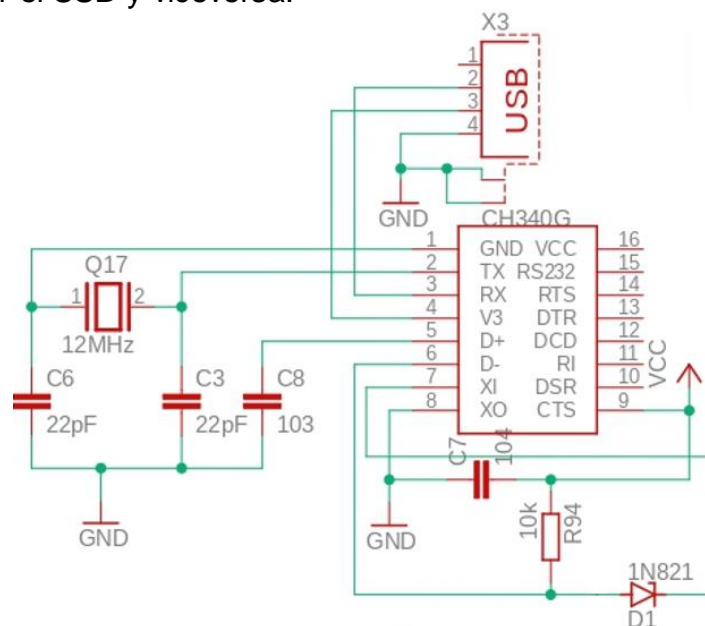


Figura 30. Circuito de comunicación USB-UART con el circuito integrado CH340G  
Fuente: Elaboración propia utilizando el software Eagle



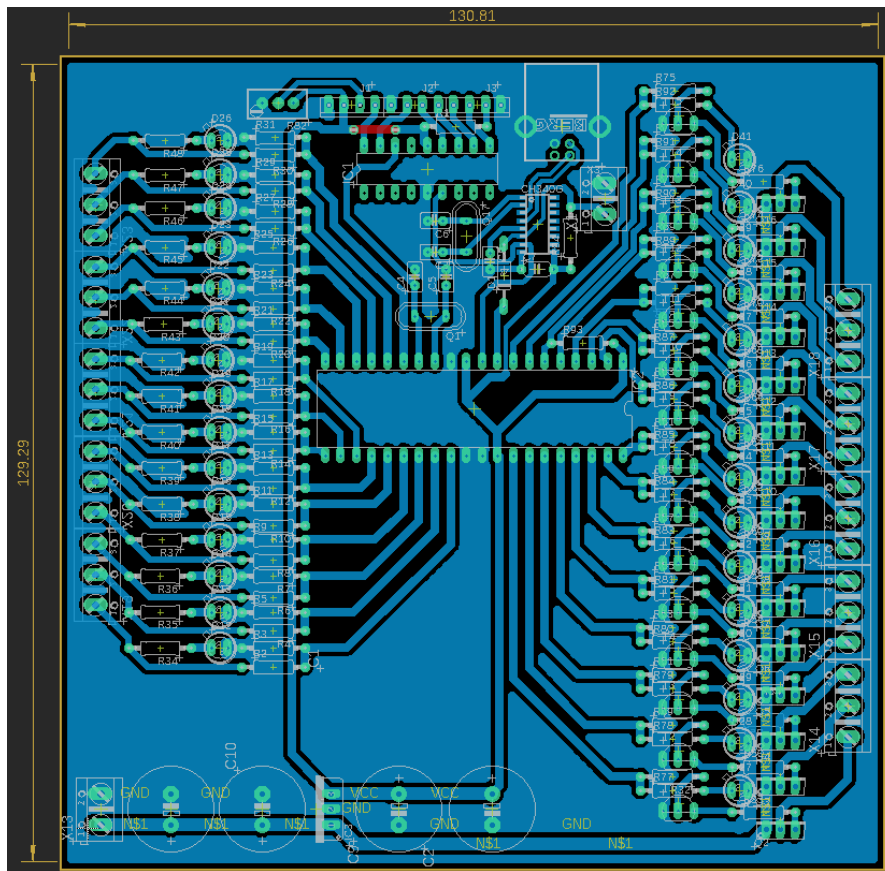


Figura 33. Diseño de la tarjeta de adquisición electrónica  
Fuente: Elaboración propia utilizando el software Eagle

En la figura 34 se muestra el desarrollo del circuito impreso en físico realizado en placa PCB, el cual posee una dimensión de 130mmx130mm. Para la elaboración de dicha placa se requirió emplear insumos químicos para disolver partes de la capa de cobre.

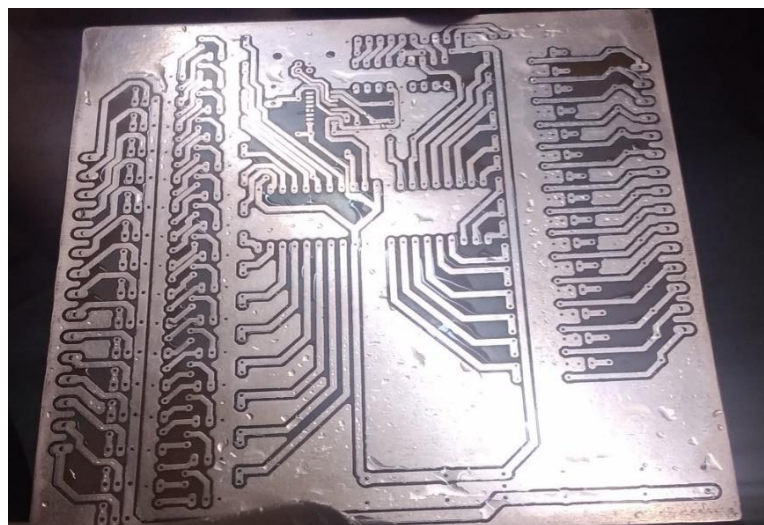


Figura 34. Desarrollo del circuito impreso en la placa PCB  
Fuente: Elaboración Propia



En la figura 35 se muestra la tarjeta de adquisición electrónica en su fase completa la cual quedó de acuerdo a como se diseñó, teniendo en consideración todos sus puntos de conexión. Además de eso se muestra un interruptor para prender el dispositivo, con su respectiva conexión USB conectada como corresponde, además se muestra la visualización en la pantalla LCD y las salidas activadas los cuales se demuestran con los leds prendidos. El módulo también puede ser conectado a través de sus conexiones bananas al PLC. La alimentación del mismo es a través de las borneras rojo (+) y negro (-) a una tensión de 24V, los cuales pueden ser alimentados de los módulos de PLC que se encuentran en el laboratorio de Control y Automatización.

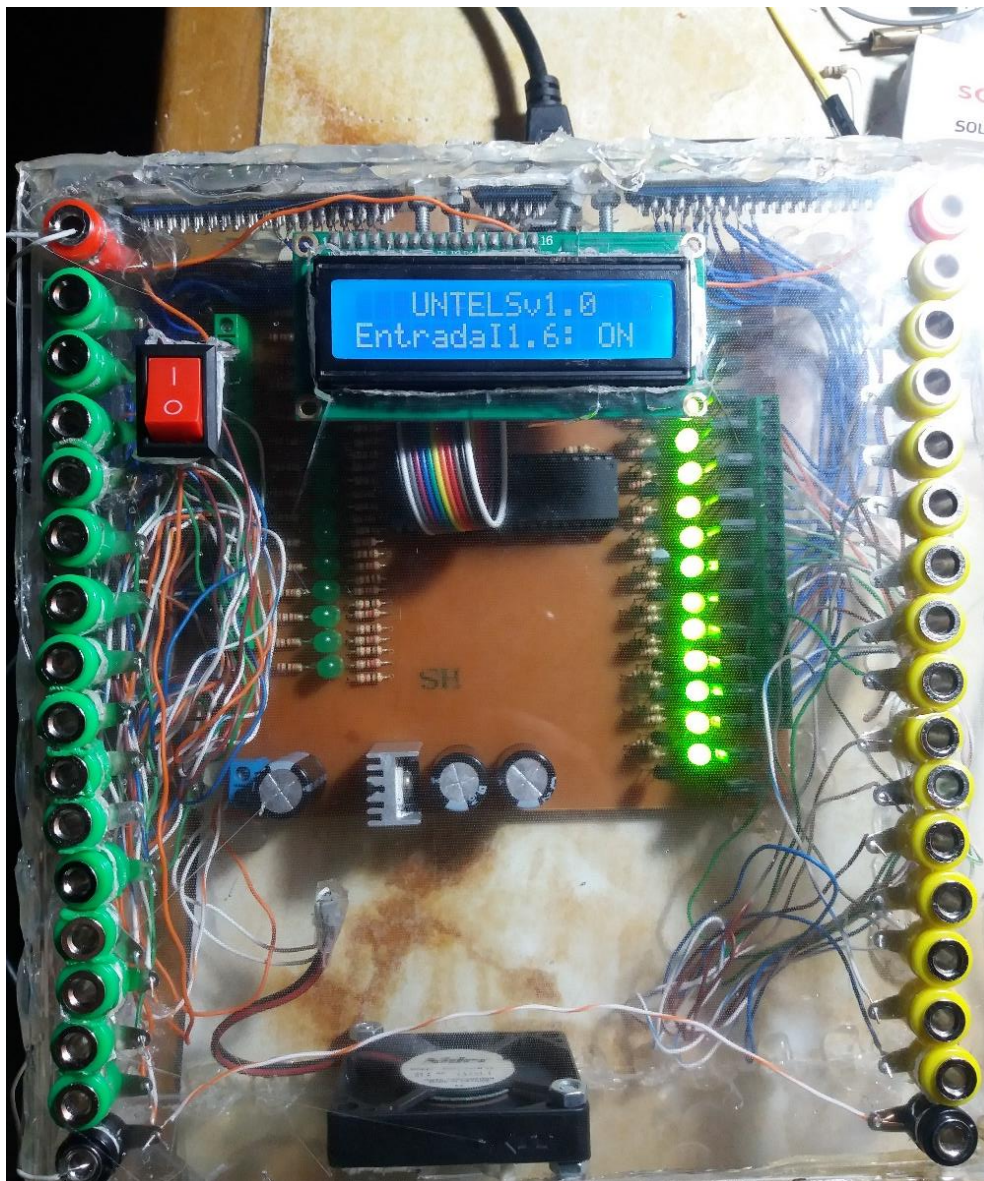


Figura 35. Versión final del desarrollo de la tarjeta de adquisición electrónica  
Fuente: Elaboración Propia

En la figura 36 se muestra la conexión del PLC con la tarjeta de adquisición electrónica a través de los cables DB37, un cable designado para las entradas y otro para las salidas, la alimentación de energía que requiere la tarjeta es suministrada por el módulo de PLC y no necesita de alimentación externa para funcionar. En la imagen también se aprecia como las salidas de la tarjeta de adquisición electrónica son reflejadas en el encendido de los indicadores en las entradas del PLC que gobierna a los demás módulos a través de la comunicación profibus.

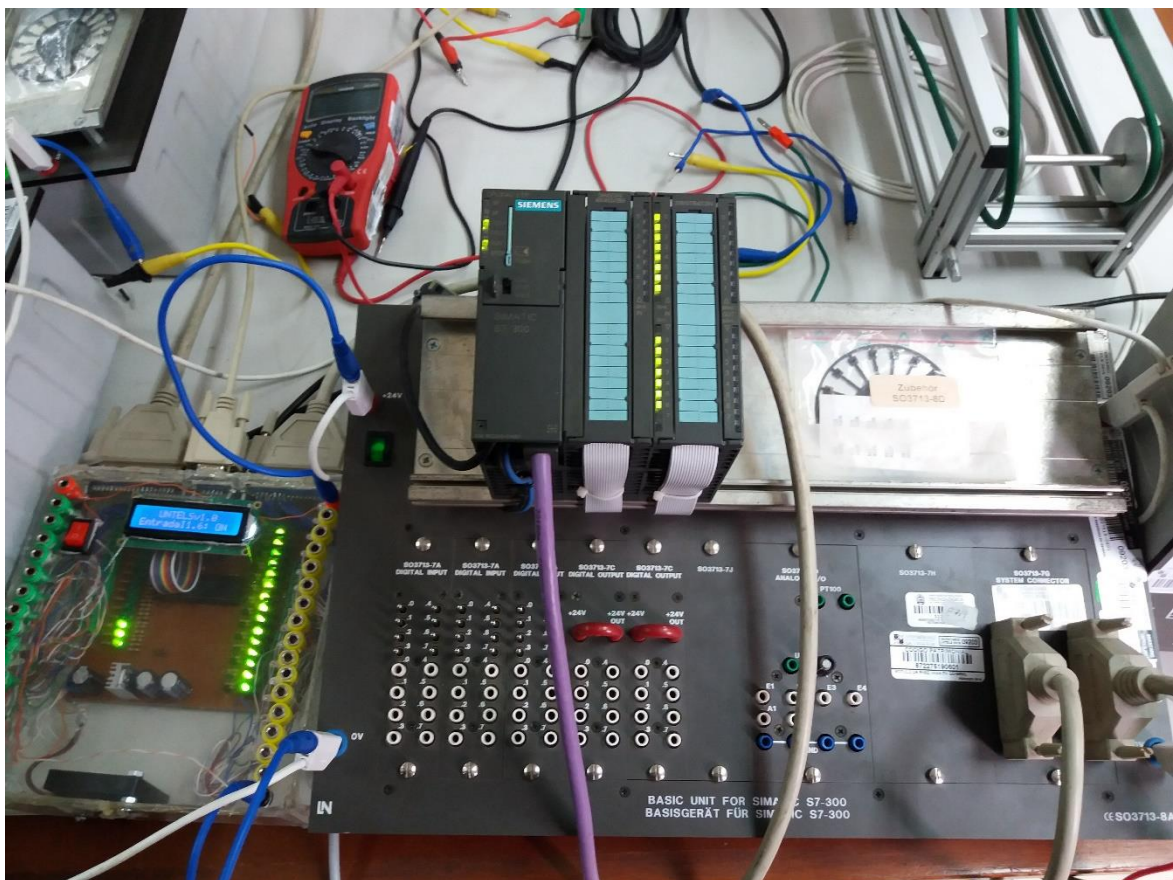


Figura 36. Conexión y pruebas con el PLC en el laboratorio de Control y Automatización  
Fuente: Elaboración Propia



## Presupuesto para la elaboración del primer prototipo

En la tabla 2 se muestra los componentes comprados de forma presencial en las tiendas ubicadas en el centro de Lima (Paruro) las cuales tuvieron un precio al por menor y son referenciales debido a que están sujetas al cambio del dólar.

Tabla 2. Presupuesto para la elaboración del prototipo

Dispositivo	Cantidad	Unidad	Precio Unitario	Precio Total
MICROCONTROLADOR PIC 16F877A	1	UND	S/14.00	14
MICROCONTROLADOR PIC 16F628A	1	UND	S/8.00	8
PANTALLA LCD 16x2	1	UND	S/8.00	8
BORNERAS DE 3 CONTACTOS	10	UND	S/0.60	6
BORNERAS DE 2 CONTACTOS	2	UND	S/0.40	0.8
CONVERTIDOR USB-SERIAL CH340G	1	UND	S/5	5
ESPADINES	1	UND	S/1	1
INDICADORES LED	30	UND	S/0.20	6
TRANSISTOR BJT PNP DB140	15	UND	S/0.50	7.5
TRANSISTOR BJT NPN 2N2222	15	UND	S/0.20	3
CAPACITORES 2200uF 25V	3	UND	S/0.80	2.4
REGULADOR DE TENSION LM7805	1	UND	S/1.50	1.5
CONECTORES DB37	2	UND	S/8.50	17
CONECTOR DB9	1	UND	S/1	1
PLACA PCB	1	UND	S/2	2
CABLES HEMBRA-MACHO	12	UND	S/0.20	2.4
CABLE	15	MTS	S/0.20	3
CONECTOR USB TIPO B HEMBRA	1	UND	S/1	1
CAMARA WEB USB 2.0	5	UND	S/50	250
BORNERAS BANANA HEMBRA	34	UND	S/0.50	17
VENTILADOR DE 24V 0.7A	1	UND	S/5	5
POTENCIOMETRO DE PRECISION 5K	1	UND	S/1	1
CABLE USB 5MTS	1	UND	S/10	10
ACRILICO PARA LA CUBIERTA 40cmx40cm	1	UND	S/10	10
ESTAÑO PARA SOLDADURA	6	MTS	S/0.50	3
DISIPADOR TO-220	1	UND	S/1.50	1.5
OSCILADOR DE CRISTAL	2	UND	S/1	2
SOCKET DE 40 Y 18 PINES	2	UND	S/0.50	1
CONDENSADORES 22pF	4	UND	S/0.10	0.4
INTERRUPTOR	1	UND	S/1	1
RESISTENCIAS 220, 1K, 10K	75	UND	S/0.10	7.5
CONECTOR ETHERNET RJ45	2	UND	S/0.50	1
CABLE UTP CAT5E	5	MTS	S/1	5
<b>TOTAL</b>				<b>405</b>

Fuente: Propia

## Programación de los Microcontroladores PIC 16F877A y 16F628A

En la figura 37 se muestra la forma en cómo funciona la programación en el PIC 16F877A lo primero que hace la programación es configurar los puertos necesarios como entradas y salidas, posterior a ello se define y asigna una dirección al dispositivo e inmediatamente después se pone en modo escucha por el puerto serie esperando a recibir algo, si ese algo que recibió y este es igual a su dirección asignada pasa a la siguiente condición que es determinar si el segundo carácter es “z” si lo fuera, automáticamente lee todos los puertos del microcontrolador y envía toda esa información hacia la interfaz en Matlab a través del puerto serial. Y si el carácter recibido no fuera “z”, pasa a otra condición indicando si este es superior al valor de 80 si así lo fuera, se redirige hacia la función prender, la cual encendería el puerto correspondiente a la dirección recibida. En caso fuera inferior apagaría el puerto.

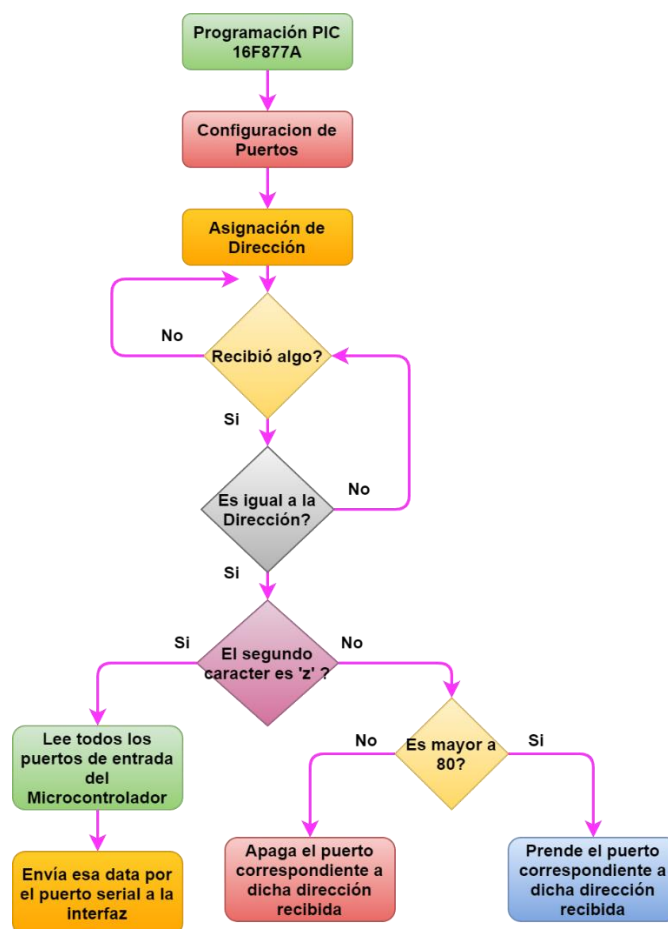


Figura 37. Diagrama de Flujo de la Programación en el PIC 16F877A  
Fuente: Elaboración Propia

En la figura 38 se muestra la forma en cómo funciona la programación del PIC 16F628A la cual es la encargada de controlar la pantalla LCD 16x2. Lo primero que se realiza es configurar los puertos del microcontrolador, luego asigna una dirección al igual que en el PIC16F877A, posterior a ello inicializa la pantalla LCD a través de una secuencia de bits y muestra en ella la frase “UNTELSv1.0” para luego ponerse en modo escucha a la espera del envío de un carácter, si el carácter recibido es igual a la dirección del PIC, pasa a una segunda condición la cual pregunta si ese carácter es mayor a 80 si así lo fuera muestra en pantalla el encendido del puerto correspondiente y si no; muestra en pantalla el apagado del puerto.

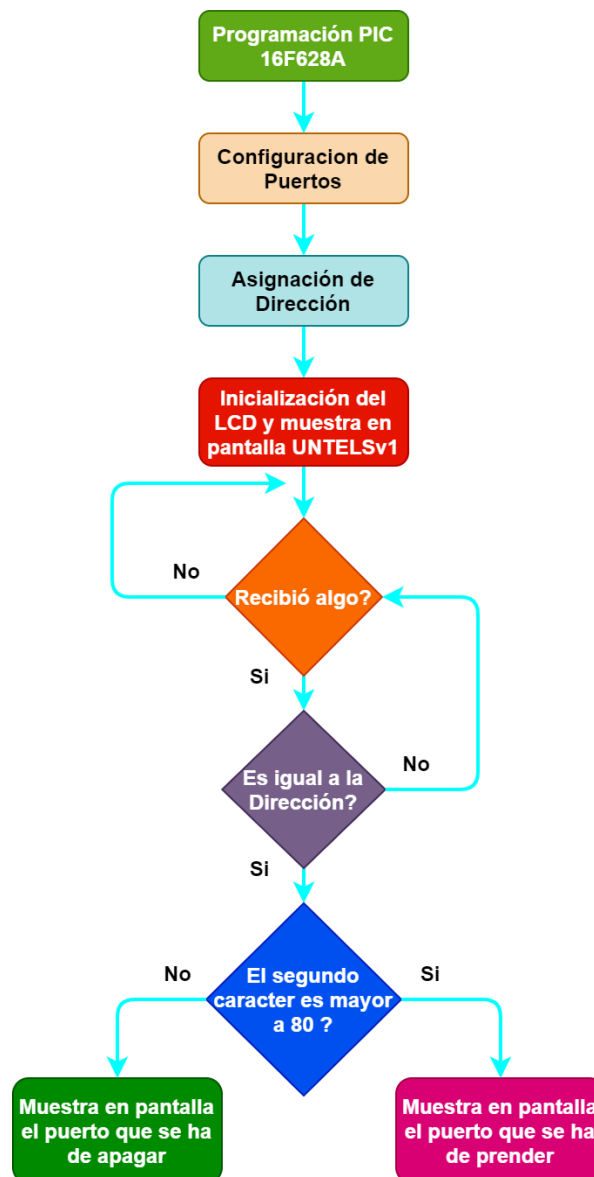


Figura 38. Diagrama de Flujo de la Programación en el PIC 16F628A  
Fuente: Elaboración Propia

En la figura 39 se muestra el programador pickit 2, justamente cuando se carga la programación al PIC 16F877A de forma exitosa.

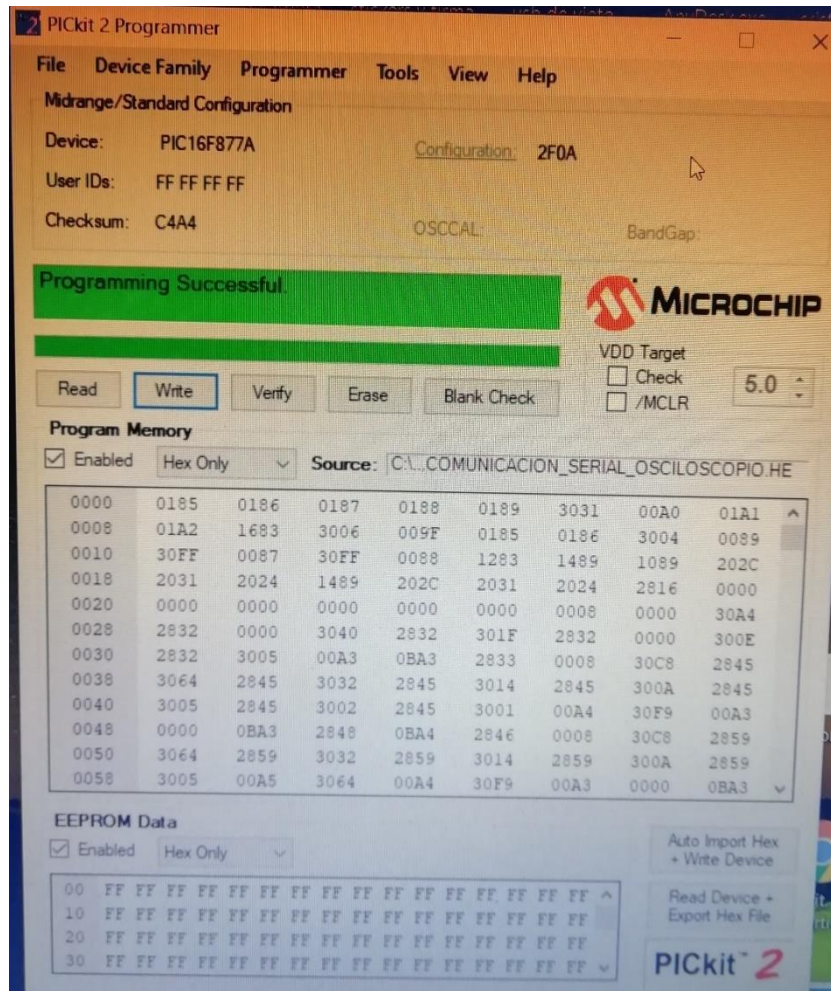


Figura 39. Carga del código de programación al PIC 16F877A  
Fuente: Elaboración propia utilizando el software PICKit2

En la figura 40 se muestra el encendido del led, el cual demuestra que se recibió la letra "a" por el puerto serial, además de visualizarse en pantalla la activación del puerto del PLC.



Figura 40. Visualización del encendido del puerto del PLC, recibido por el puerto serial  
Fuente: Elaboración Propia

En la figura 41 se muestra el apagado del led, el cual demuestra que se recibió la letra "A" por el puerto serial, además de visualizarse en pantalla la desactivación del puerto del PLC.



Figura 41. Visualización del apagado del puerto del PLC, recibido por el puerto serial  
Fuente: Elaboración Propia

En la figura 42 se muestra la activación de las salidas, corroborando la presencia de los 24 voltios que debe enviar al PLC.

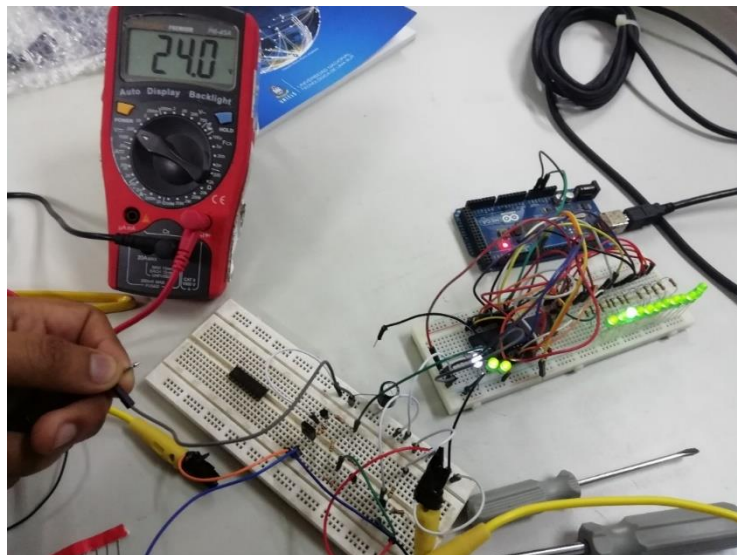


Figura 42. Demostración de la activación del Sistema  
Fuente: Elaboración Propia

En la figura 43 se muestra la desactivación de las salidas corroborando la ausencia de tensión, indicando los 0 voltios que debe enviar al PLC.



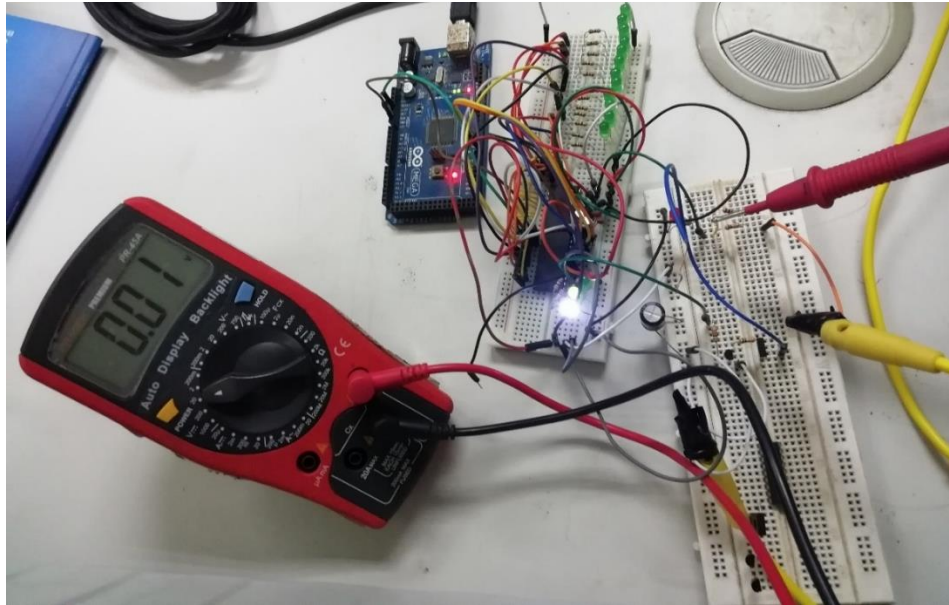


Figura 43. Demostración de la desactivación del Sistema  
Fuente: Elaboración Propia

En la figura 44 se muestra la comunicación serial transmitida a 9600 Baudios, en la cual se puede apreciar la precisión con la que se envía la información a aproximadamente 104 microsegundos por cada bit de dato. Para esto se empleó el osciloscopio del laboratorio de Control y Automatización de la Universidad Nacional Tecnológica de Lima Sur.

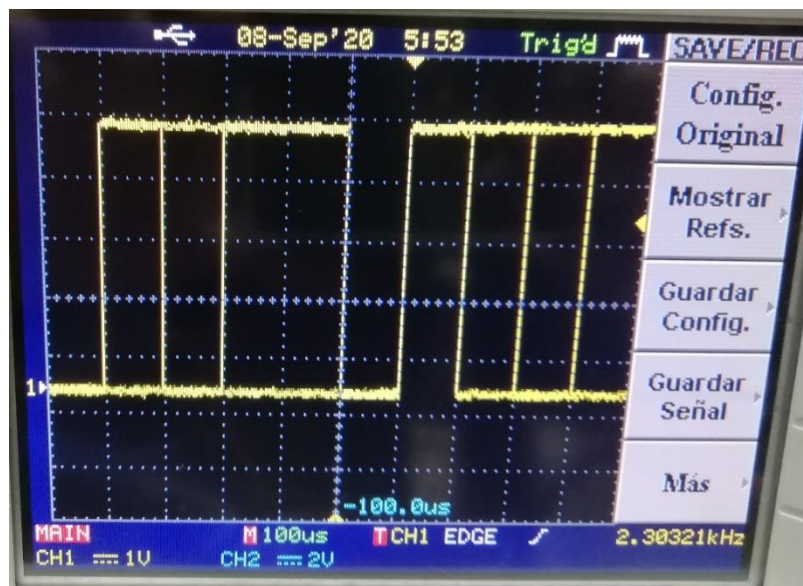


Figura 44. Prueba de Transmisión de datos a 9600 Baudios en el osciloscopio  
Fuente: Elaboración Propia

### 2.3.4 Validación y ejecución del Control del Sistema Automatizado de Procesos Industriales

En la figura 45 se muestra el reconocimiento de los módulos Lucas Nülle en el software TIA Portal; en caso no se cuente con esa opción como se muestra en la figura se debe proceder a descargar los controladores de la página del fabricante y se añade a través de la opción “administrador de archivos de descripción de dispositivos” del TIA Portal, presente en la barra de menú del software.



Figura 45. Reconocimiento de los Módulos Lucas Nülle GmbH  
Fuente: Elaboración propia utilizando el software TIA Portal

En la figura 46 se muestra la estructura de la Red Profibus DP en la que el PLC S7-300 314C-2-DP trabaja como maestro profibus y posee la dirección 2 mientras que la pantalla HMI posee la dirección 1 y las demás estaciones de la dirección 3 a la 6.

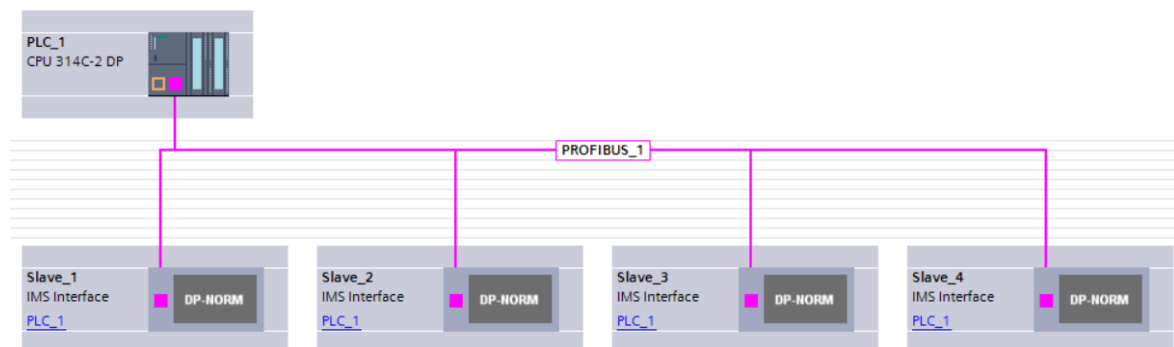


Figura 46. Estructura de la Red Profibus  
Fuente: Elaboración propia utilizando el software TIA Portal

En la figura 47 se muestra la lista de variables utilizadas para la simulación en TIA Portal donde se puede apreciar todas las variables, tipos de variables y cada una de sus direcciones.

1	ESTADO	Tabla de variables e..	Int	%MW10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Tag_1	Tabla de variables e..	Bool	%Q2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Tag_2	Tabla de variables e..	Bool	%Q2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Tag_3	Tabla de variables e..	Bool	%Q3.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	START	Tabla de variables e..	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Tag_5	Tabla de variables e..	Bool	%I4.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Tag_6	Tabla de variables e..	Bool	%I4.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Tag_7	Tabla de variables e..	Bool	%M0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Tag_8	Tabla de variables e..	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Tag_9	Tabla de variables e..	Bool	%M0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	STOP	Tabla de variabl...	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Tag_11	Tabla de variables e..	Bool	%M0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Tag_12	Tabla de variables e..	Bool	%M0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	ESTADO1	Tabla de variables e..	Int	%MW12		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Tag_14	Tabla de variables e..	Bool	%M1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Tag_15	Tabla de variables e..	Bool	%M1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	Tag_16	Tabla de variables e..	Bool	%M1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Tag_17	Tabla de variables e..	Bool	%Q4.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Tag_18	Tabla de variables e..	Bool	%Q4.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	Tag_19	Tabla de variables e..	Bool	%Q5.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	Tag_20	Tabla de variables e..	Bool	%I6.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	Tag_21	Tabla de variables e..	Bool	%I6.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	Tag_22	Tabla de variables e..	Bool	%Q4.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	ESTADO2	Tabla de variables e..	Int	%MW14		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	Tag_24	Tabla de variables e..	Bool	%Q7.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	Tag_26	Tabla de variables e..	Bool	%I8.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	Tag_27	Tabla de variables e..	Bool	%I8.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	Tag_28	Tabla de variables e..	Bool	%M2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	Tag_25	Tabla de variables e..	Bool	%Q6.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	BUENOS	Tabla de variables e..	Int	%MW16		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
31	MALOS	Tabla de variables e..	Int	%MW18		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	Tag_4	Tabla de variables e..	Bool	%I7.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	Tag_10	Tabla de variables e..	Bool	%I7.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	Tag_13	Tabla de variables e..	Bool	%M2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	MALOS1	Tabla de variables e..	Int	%MW20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	Tag_23	Tabla de variables e..	Bool	%M2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	Tag_29	Tabla de variables e..	Bool	%M2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38	Tag_30	Tabla de variables e..	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39	Tag_31	Tabla de variables e..	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40	Tag_32	Tabla de variables e..	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
41	ESTADO3	Tabla de variables e..	Int	%MW22		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
42	Tag_33	Tabla de variables e..	Bool	%Q9.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
43	Tag_34	Tabla de variables e..	Bool	%Q8.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
44	Tag_35	Tabla de variables e..	Bool	%I10.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
45	Tag_36	Tabla de variables e..	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
46	Tag_37	Tabla de variables e..	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
47	Tag_38	Tabla de variables e..	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
48	Tag_39	Tabla de variables e..	Bool	%Q8.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
49	Tag_40	Tabla de variables e..	Bool	%Q8.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
50	Tag_41	Tabla de variables e..	Bool	%Q8.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
51	Tag_42	Tabla de variables e..	Bool	%I10.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
52	Tag_43	Tabla de variables e..	Bool	%M3.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
53	Tag_44	Tabla de variables e..	Bool	%M3.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
54	Tag_45	Tabla de variables e..	Bool	%I9.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
55	Tag_46	Tabla de variables e..	Bool	%I9.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
56	Tag_47	Tabla de variables e..	Bool	%M3.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
57	Tag_48	Tabla de variables e..	Bool	%M3.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
58	Tag_49	Tabla de variables e..	Bool	%M3.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
59	Tag_50	Tabla de variables e..	Bool	%M3.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
60	Tag_51	Tabla de variables e..	Bool	%M3.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
61	TOTAL	Tabla de variables e..	Int	%MW24		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 47. Lista de variables utilizadas para la simulación  
Fuente: Elaboración propia utilizando el software TIA Portal



En la figura 48 se muestra la función principal del ejemplo propuesto en el software TIA Portal, el cual está compuesto por bloques de funciones denominados Subsistema\_Separación, Subsistema\_Montaje, Subsistema\_Verificación y Subsistema\_Manipulación.



Figura 48. Programa Principal (Main)  
Fuente: Elaboración propia utilizando el software TIA Portal

### Programación del Sistema de Separación

En la figura 49 se muestra el diagrama de funcionamiento de la estación de separación de forma automática. Inicialmente comienza con todos sus actuadores apagados espera a que el operador indique el inicio del ciclo presionando el botón "start", la faja se prende y empieza a moverse; posterior a ello debe detectar el sensor izquierdo de la subestación para determinar que el carro se encuentra ahí y luego colocar la pieza a ensamblar, cuando ya lo detecta pasa al estado 2 y activa

el tope, después espera a detectar el sensor de la derecha para así apagar la faja y activar la separación; enseguida ha de pasar un tiempo de 200ms para recién apagar la separación, ese tiempo de espera es importante para que la pieza caiga correctamente. Si ese tiempo fuera muy corto el actuador no podría responder tan rápido y no dejaría caer dicha pieza y así se generaría un error en el proceso, posterior a ello se apaga el tope y la faja inicia nuevamente el movimiento. En todo momento el sistema está al pendiente si el operador en algún momento presiona el botón de parada siendo este el caso volvería al estado 0 donde todos los actuadores se encuentran apagados y están a la espera de que el operador presione “start” y así pasar al siguiente estado.

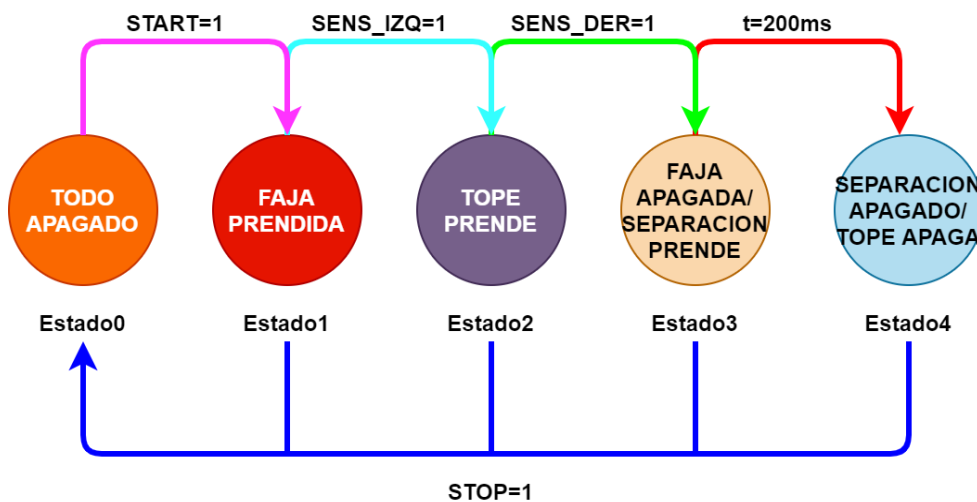


Figura 49. Esquema del funcionamiento de la Estación de Separación  
Fuente: Elaboración Propia

En la figura 50 se muestra la subestación de Separación la cual podemos notar que se encuentra conectada al PLC a través del bus profibus, además de ello cuenta con un cable que conecta los actuadores de la parte superior con la tarjeta principal del módulo de la subestación, también cuenta con dos borneras de conexión a 24V los cuales son compartidos por los módulos de PLC que están en el laboratorio de Control y Automatización.

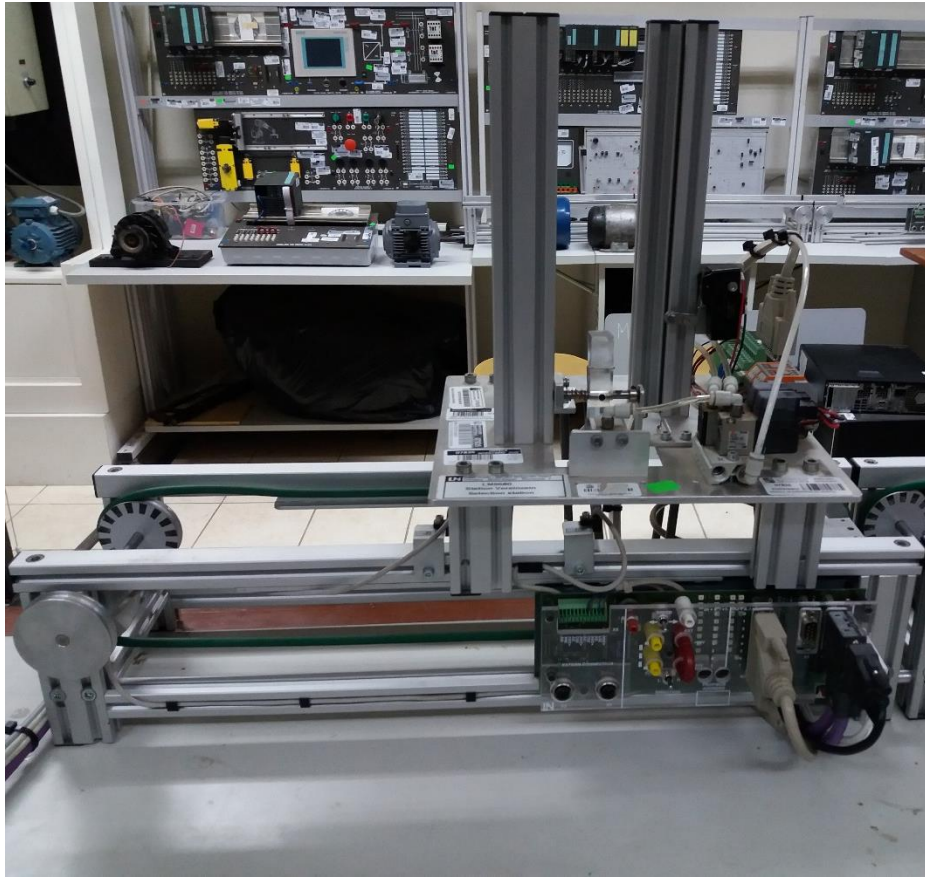


Figura 50. Estación de Separación  
Fuente: Elaboración Propia

### **Programación del Sistema de Montaje**

En la figura 51 se muestra el diagrama de funcionamiento de la estación de Montaje de forma automática. Inicialmente comienza con todos sus actuadores apagados espera a que el operador indique el inicio del ciclo presionando el botón “start”, la faja se prende y empieza a moverse; posterior a ello debe detectar el sensor izquierdo de la subestación para determinar que el carro se encuentra ahí y luego colocar la pieza a ensamblar, cuando ya lo detecta pasa al estado 2 y activa el tope, después espera a detectar el sensor de la derecha para así apagar la faja y activar el Montaje; enseguida ha de pasar un tiempo de 200ms para recién apagar el montaje, ese tiempo de espera es importante para que la pieza caiga correctamente. Si ese tiempo fuera muy corto el actuador no podría responder tan rápido y no dejaría caer dicha pieza y así se generaría un error en el proceso, posterior a ello se apaga el tope y la faja inicia nuevamente el movimiento. En todo momento el sistema está al pendiente si el operador en algún momento presiona

el botón de parada siendo este el caso volvería al estado 0 donde todos los actuadores se encuentran apagados y están a la espera de que el operador presione “start” y así pasar al siguiente estado.

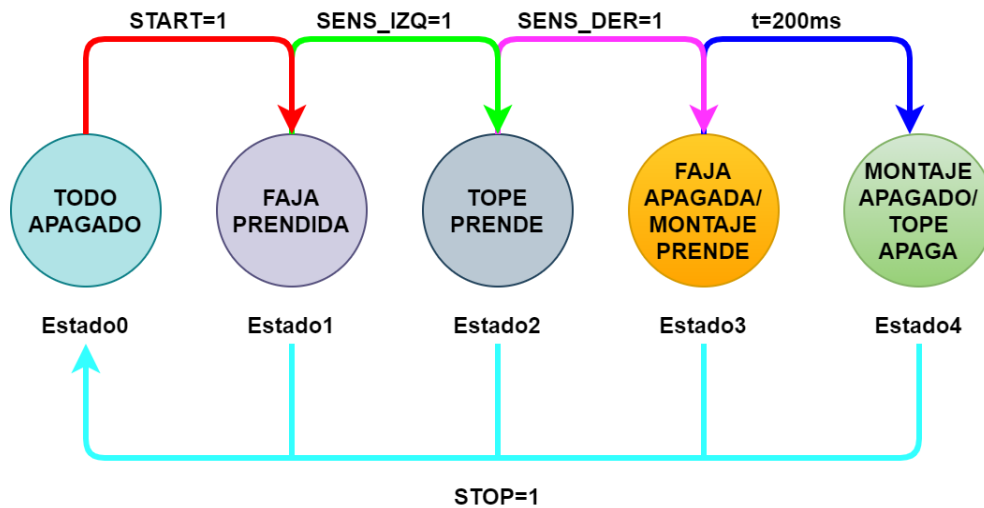


Figura 51. Esquema del funcionamiento de la Estación de Montaje  
Fuente: Elaboración Propia

En la figura 52 se muestra la subestación de Montaje la cual podemos notar que se encuentra conectada al PLC a través del bus profibus, además de ello cuenta con un cable que conecta los actuadores de la parte superior con la tarjeta principal del módulo de la subestación, también cuenta con dos borneras de conexión a 24V los cuales son compartidos por los módulos de PLC que están en el laboratorio de Control y Automatización.



Figura 52. Estación de Montaje  
Fuente: Elaboración Propia

## Programación del Sistema de Verificación

En la figura 53 se muestra el diagrama de funcionamiento de la estación de Verificación de forma automática. Inicialmente comienza con todos sus actuadores apagados espera a que el operador indique el inicio del ciclo presionando el botón “start”, la faja se prende y empieza a moverse; posterior a ello debe detectar el sensor izquierdo de la subestación para determinar que el carro se encuentra ahí y así se apagan tanto la faja como el tope. A través de sus sensores laterales y superiores verifica que el bloque este correctamente ensamblado, teniendo como consigna el mismo color tanto en el cuerpo como en la tapa. Si el color es el mismo la variable “bueno” incrementa en una unidad al igual que la variable total; de lo contrario la variable “malo” incrementaría en una unidad.

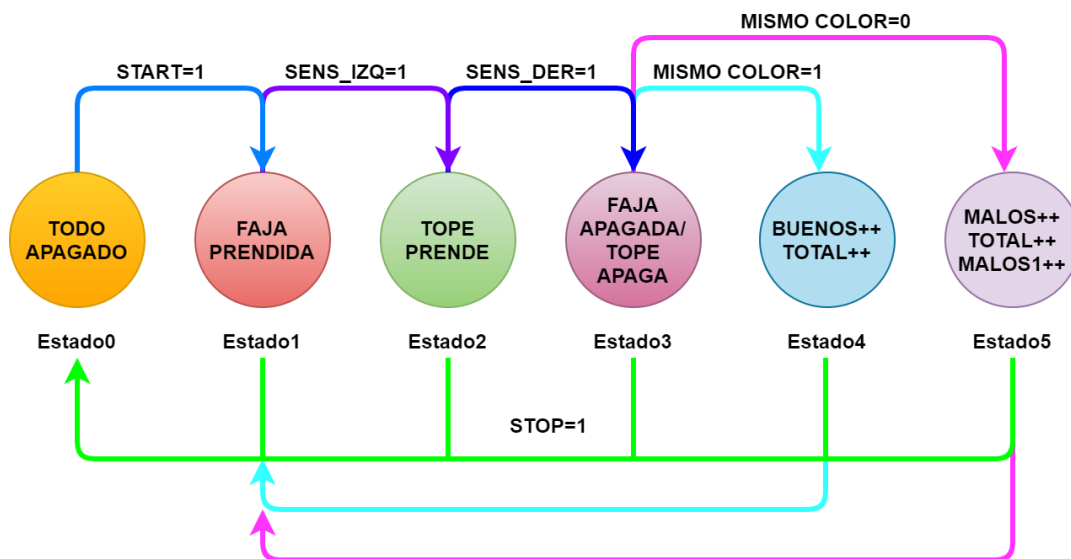


Figura 53. Esquema del funcionamiento de la Estación de Verificación  
Fuente: Elaboración Propia

En la figura 54 se muestra la estación de verificación con cada uno de sus sensores de posición izquierda y derecha, también los sensores que se encuentran en la parte lateral y superior. Estos sensores realizan la verificación de los colores del cuerpo y la tapa del bloque ensamblado, haciendo una desigualdad si el bloque está bien en caso sea del mismo color; de lo contrario la estación de manipulación se encargaría de retirarlo del proceso de ensamblaje. Uno de los puntos importantes de la estación de verificación es que cuentan con dos borneras de conexión a 24V los cuales son compartidos por los módulos de PLC que se encuentran en el laboratorio de Control y Automatización.

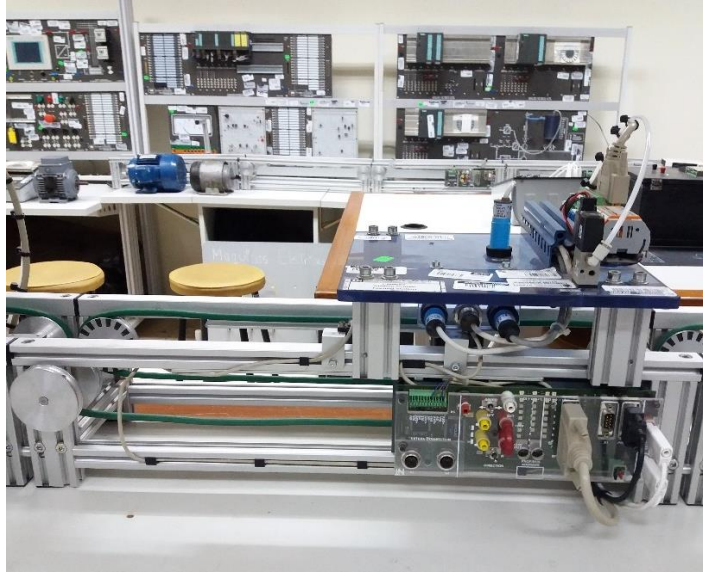


Figura 54. Estación de Verificación  
Fuente: Elaboración Propia

### **Programación del Sistema de Manipulación**

En la figura 55 se muestra el esquema de funcionamiento de la estación de manipulación, la cual inicialmente tiene sus actuadores apagados, para eso espera a que el operario presione la opción “start” que activa la faja y posiciona al pivote a  $90^\circ$  después espera a detectar el sensor de la posición izquierda para luego preguntar en memoria si la variable “malos1” es cero o no; según eso pasa al siguiente estado, es ahí cuando el tope se enciende y espera a que el sensor de la derecha detecte la posición para apagar la faja, apaga el tope y baja el aspirador, después de eso debe pasar 1.2 segundos para que el generador de vacío se encienda, después espera 0.5 segundos para subir el aspirador luego censa cuando el aspirador se encuentra arriba para posterior a ello el pivote se traslade a la posición  $0^\circ$ , cuando esa posición se realiza baja el aspirador pasando un tiempo de 1.2 segundos nuevamente, luego el generador de vacío se apaga y el aspirador sube y por último cuando el aspirador llegue a la posición de arriba se hará el descuento de la variable “malos1”. Y el proceso volverá al estado 1 donde se volverá a repetir el proceso hasta que el operario pare el proceso a través de la opción “stop”.



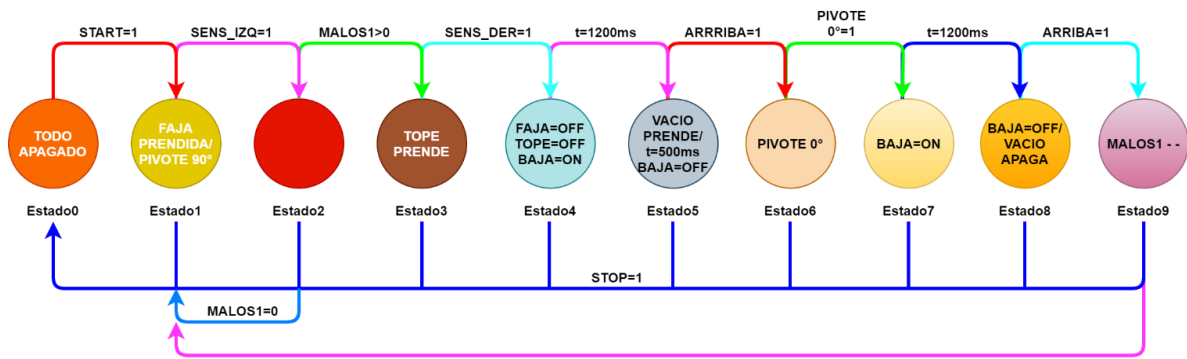


Figura 55. Esquema del funcionamiento de la Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 56 se muestra la estación de manipulación la cual cuenta con un brazo que tiene dos gomas de jebe que las usa para atrapar la tapa de los bloques empleando una válvula de vacío para sujetarla fuertemente y así no se caiga. Posee sensores de posición en la parte izquierda y derecha además del tope para que el carrito se quede en la posición correcta para que el brazo pueda actuar sobre el bloque.

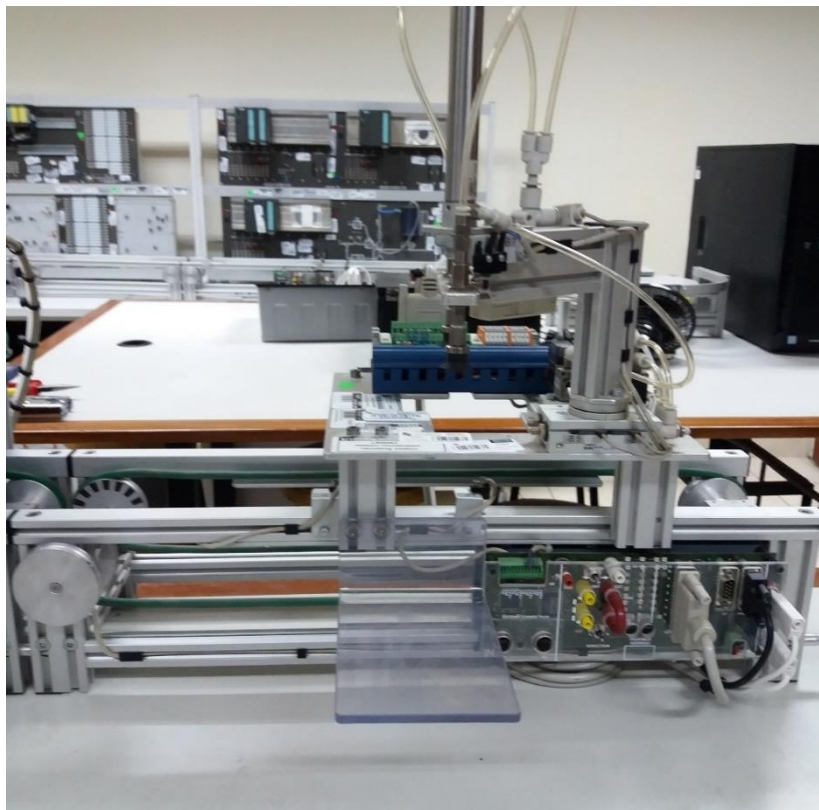


Figura 56. Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 57 se muestra el flujograma del funcionamiento de la programación de la pantalla HMI. Pregunta inicialmente si se presionó la opción “start” si fuera el caso se prende el foco verde ubicado en pantalla, si en caso se presiona “stop” se prender el foco rojo, y si en caso detecto bloques buenos, aumenta la cantidad de bloques buenos y también aumenta la cantidad total de bloques, y si por el contrario detecto bloques malos aumenta la cantidad de bloques malos y también aumenta la cantidad total de bloques.

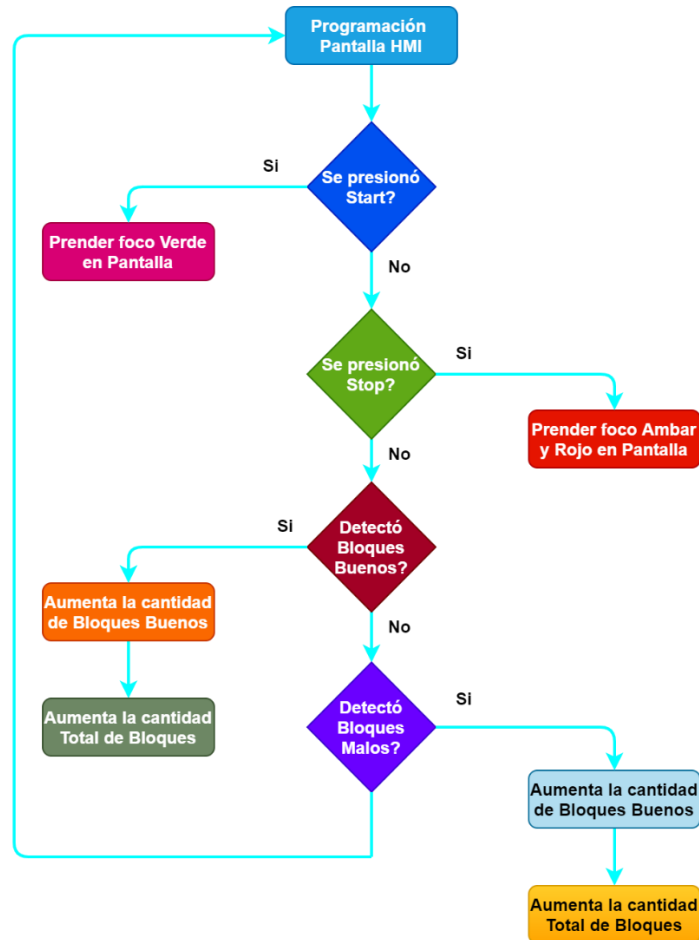


Figura 57. Flujograma del funcionamiento de la programación de la pantalla HMI  
Fuente: Elaboración Propia

En la figura 58 se muestra la lista de variables utilizadas para el control desde la pantalla HMI OP177B de la Universidad Nacional Tecnológica de Lima Sur.

Nombre	Conexión	Tipo de datos	Dirección	Elementos de ...	Ciclo de adquis...
BUENOS	Conexión_1	Int	MW 16	1	100 ms
MALOS	Conexión_1	Int	MW 18	1	100 ms
START_HMI	Conexión_1	Bool	M 0.6	1	100 ms
STOP_HMI	Conexión_1	Bool	M 0.7	1	100 ms
TOTAL	Conexión_1	Int	MW 24	1	100 ms

Figura 58. Lista de variables utilizadas para la conexión con el PLC  
Fuente: Elaboración propia utilizando el software WinCC Flexible



En la figura 59 se muestra la elaboración del programa con sus botones e indicadores, además de cuadros donde se muestran la cantidad de bloques buenos y malos los cuales serán determinados por la estación de verificación distinguiendo que bloques son los que han sido bien elaborados y tienen el mismo color en sus cuerpos y tapas.

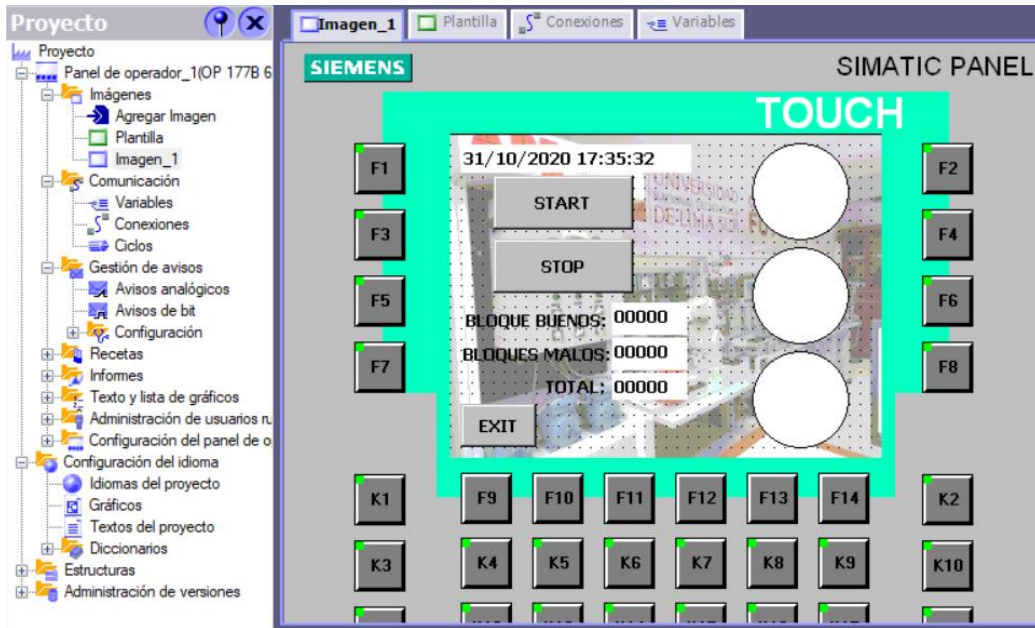


Figura 59. Diseño de la Interfaz de la pantalla HMI  
Fuente: Elaboración propia utilizando el software WinCC Flexible

En la figura 60 se muestra la pantalla HMI OP177B con el programa cargado desde el software WinCC. Además de eso en la programación las variables “malo” y “malo\_1” van aumentando como se muestra en la interfaz donde dice bloques malos cantidad 2. Como última cantidad a mostrar es el total de bloques procesados.

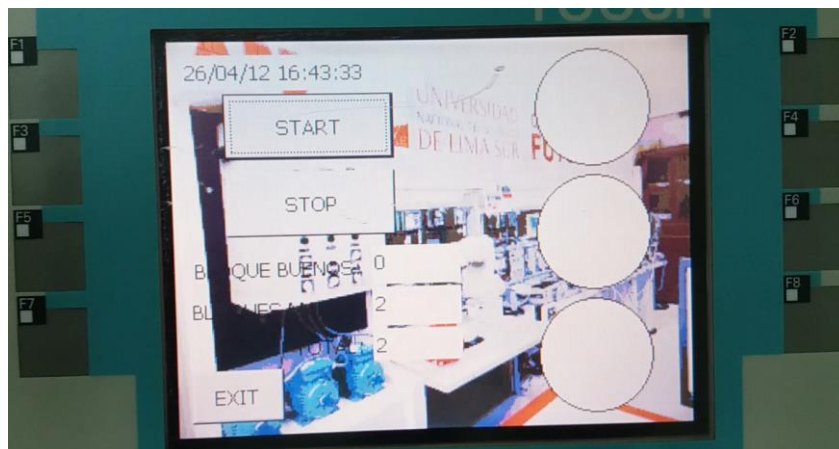


Figura 60. Visualización de la Pantalla HMI OP177B con el programa cargado  
Fuente: Elaboración Propia

En la figura 61 se muestra las posibilidades en las cuales se puede presentar el bloque del proceso a ensamblar. Los bloques bien elaborados tienen el mismo color en sus cuerpos y tapas y los mal elaborados poseen diferentes colores en sus cuerpos y tapas.



Figura 61. Bloques de la Estación Automatizada de Procesos Industriales  
Fuente: Elaboración Propia

### **Validación del Sistema desarrollado desde un lugar Remoto.**

En la figura 62 se muestra cuando el estudiante inicializa el proceso accionando el funcionamiento automático, presionando la opción "START". Previamente inicializó las cámaras y sincronizó el puerto serial e inicio sesión en la Interfaz Remota. El funcionamiento automático hace mención a que no necesitar de un operario para que realice o haga el movimiento de cada uno de los equipos, sino que estos mismos desarrollen una tarea repetitivamente de forma automática.

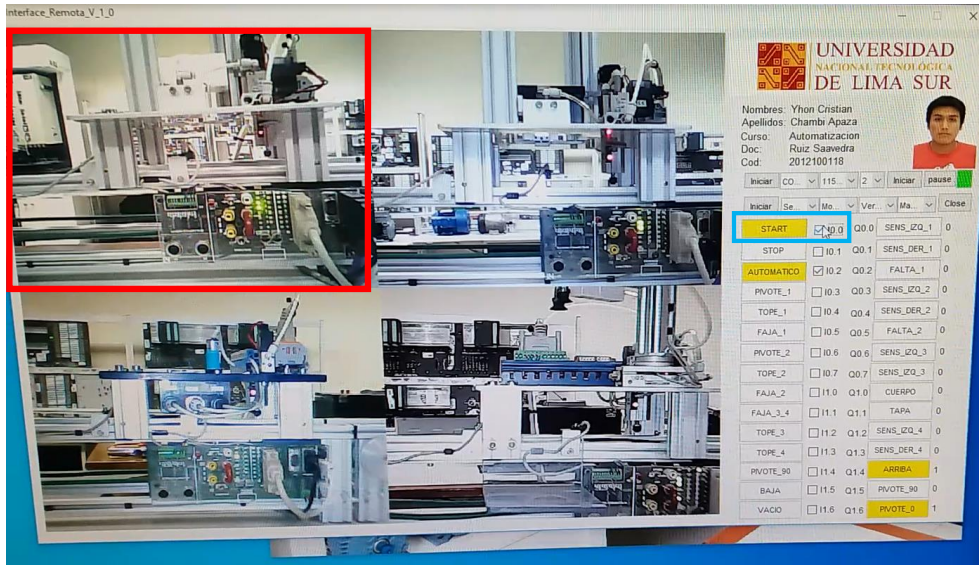


Figura 62. Inicio de la Simulación  
Fuente: Elaboración Propia

En la figura 63 se logra apreciar la detección del sensor izquierdo a consecuencia de este se activa el tope y finalmente detecta el sensor de la derecha donde se ve en la imagen que lo detecta y automáticamente el tope se apaga debido a que ya hizo su función.



Figura 63. Detección del sensor derecho de la Estación de Separación  
Fuente: Elaboración Propia

En la figura 64 se muestra cómo se activó la separación y ya el bloque cayó en el carro donde se ensamblará las piezas. Posterior a ello el sensor que lleva en la parte superior de la estación de separación, indica que ya no hay piezas para seguir fabricando y muestra en la interfaz la activación de la opción "Falta\_1", que justamente hace referencia a la falta de piezas en la estación de separación.



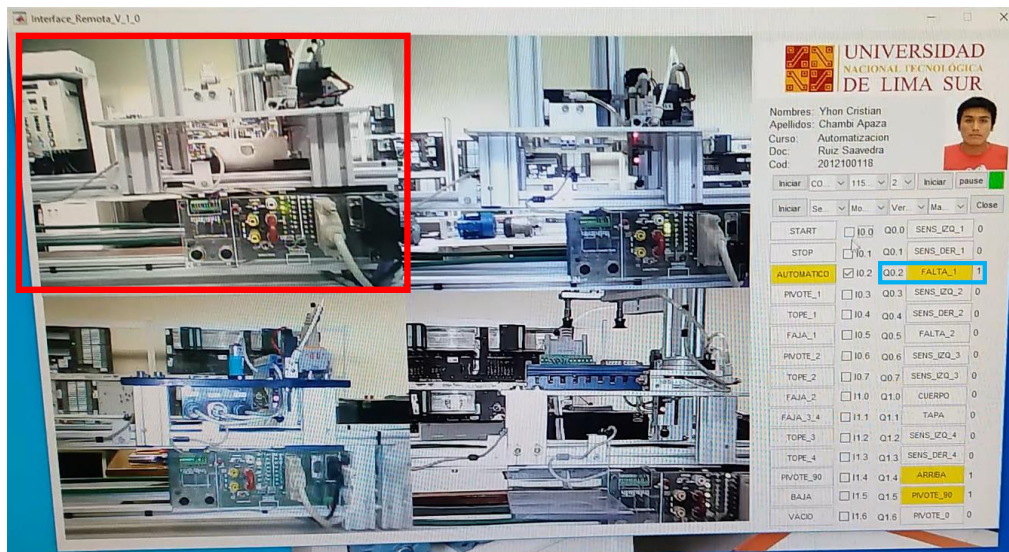


Figura 64. Falta piezas en la Estación de Separación  
 Fuente: Elaboración Propia

En la figura 65 se logra apreciar la detección del sensor izquierdo en la estación de Montaje a consecuencia de este se activa el tope, para cuando llegue al sensor de la derecha pare la faja de la estación de Montaje y desactive el tope, tal cual como se muestra en la imagen con la opción "SENS\_DER\_2" activada cambiando de color blanco a amarillo, el cual indica que el sensor lo detectó.



Figura 65. Detección del sensor derecho de la Estación de Montaje  
 Fuente: Elaboración Propia

En la figura 66 se muestra como luego de dejar caer la tapa sobre el carro, ya se tiene un producto que está constituido por un cuerpo y una tapa, ahora también se muestra que la interfaz, tanto la estación de separación como la

estación de montaje les hacen falta piezas para seguir fabricando el ensamblado de los bloques.

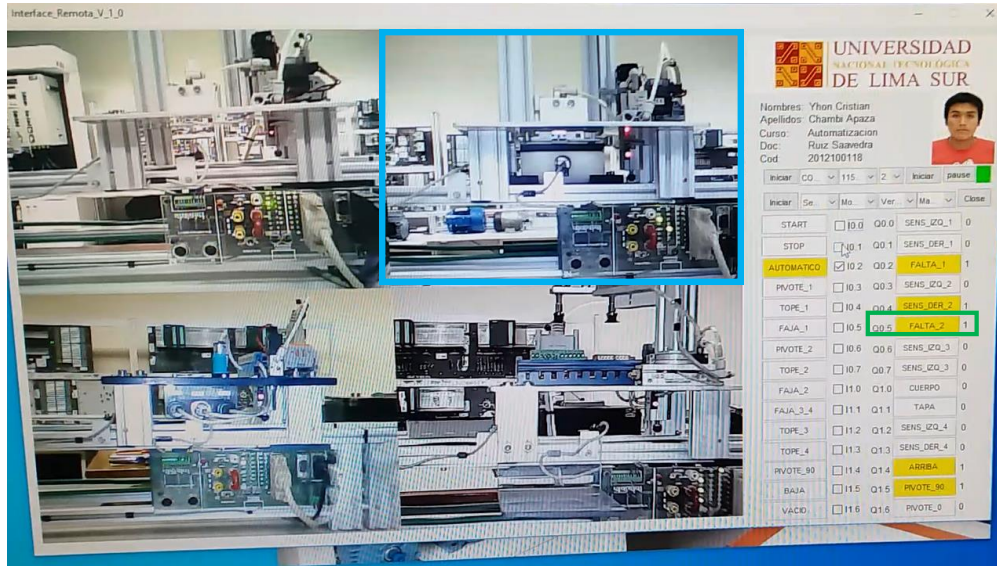


Figura 66. Falta piezas en la Estación de Montaje  
Fuente: Elaboración Propia

En la figura 67 se muestra como el bloque paso a la estación de verificación y es ahí donde el sensor izquierdo activó el tope y segundos después detectó el sensor de la derecha desactivando así la faja y el tope, además de eso se hizo la detección del cuerpo del bloque fabricado en este caso el cuerpo es blanco por lo que reconoce como uno lógico y la tapa como es negro lo reconoce como 0 lógico, independientemente tanto la variable “malo” y “malo\_1” aumenta en una unidad el valor de las variables que se muestran en la pantalla HMI. Viendo en esta la cantidad de bloques buenos y malos generados en el momento del ensamblaje.



Figura 67. Detección del sensor derecho de la Estación de Verificación  
Fuente: Elaboración Propia



En la figura 68 se muestran varias opciones. Antes de llegar a la estación de manipulación primero lo detecta el sensor de la izquierda en este caso pregunta si es un bloque bueno o malo, si es malo como es el caso debido a que los colores que presenta no son iguales lo que hace inmediatamente es activar el tope, cuando llega al sensor de la izquierda como se muestra en la imagen para la faja y el tope se desactiva, el pivote ya se encuentra a 90°, además el sensor denominado arriba se encuentra activado.

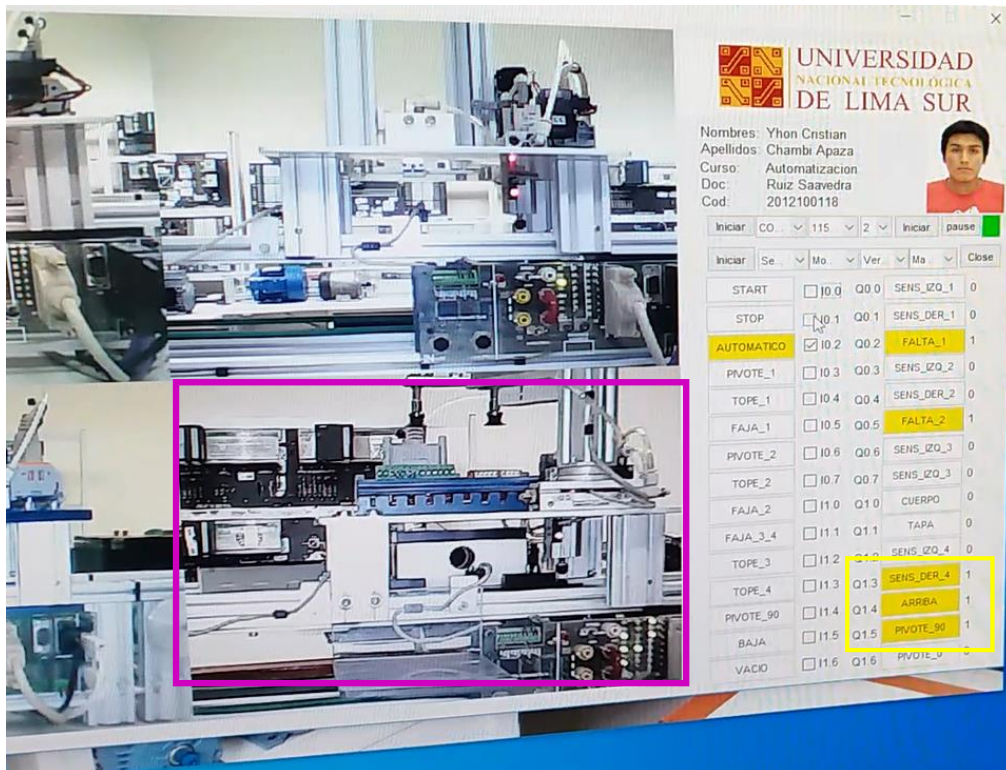


Figura 68. Detección del sensor derecho de la Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 69 se muestra como el pivote desciende para cargar la tapa, como no posee un final de carrera se le asignó un tiempo de 1200ms como tiempo promedio que le toma para llegar hasta la posición inferior, posterior a ello activa el generador de vacío, esto lo que hace es absorber la tapa con el aire para que esta no se caiga cuando el pivote vuelva a subir y sea trasladado hacia el deposito de bloques malos.



Figura 69. El pivote desciende Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 70 se muestra como el pivote absorbe el bloque y lo sube para trasladarlo al depósito de bloques malos. La opción arriba lo detecta y lo muestra en la imagen.

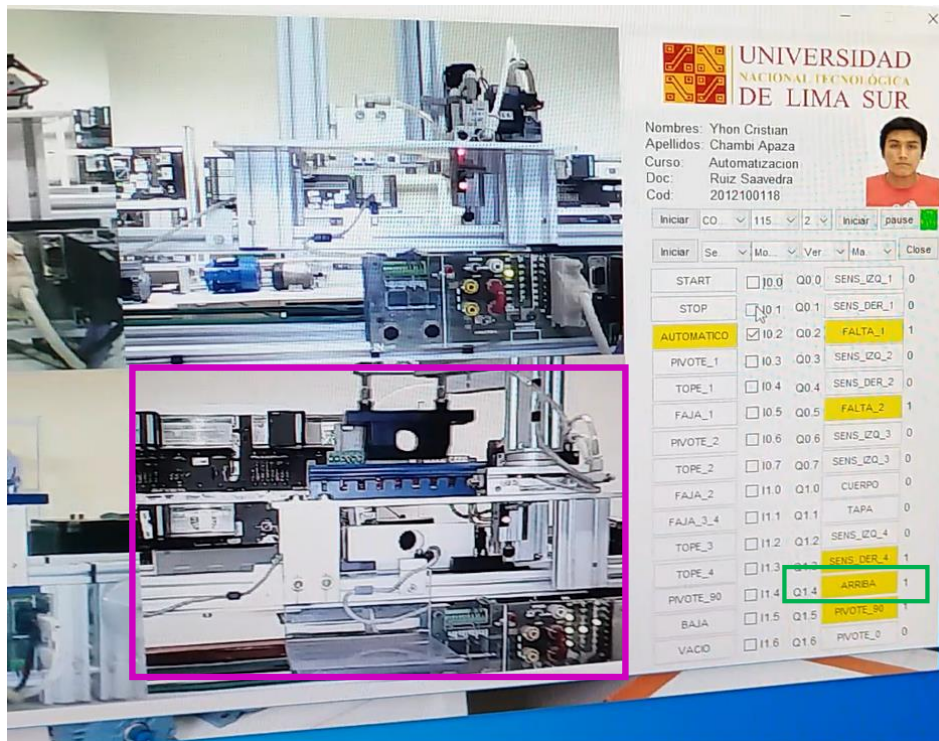


Figura 70. El pivote asciende Estación de Manipulación  
Fuente: Elaboración Propia



En la figura 71 se muestra cuando el pivote se va a la posición de 0° para dejar la tapa del bloque mal ensamblado.

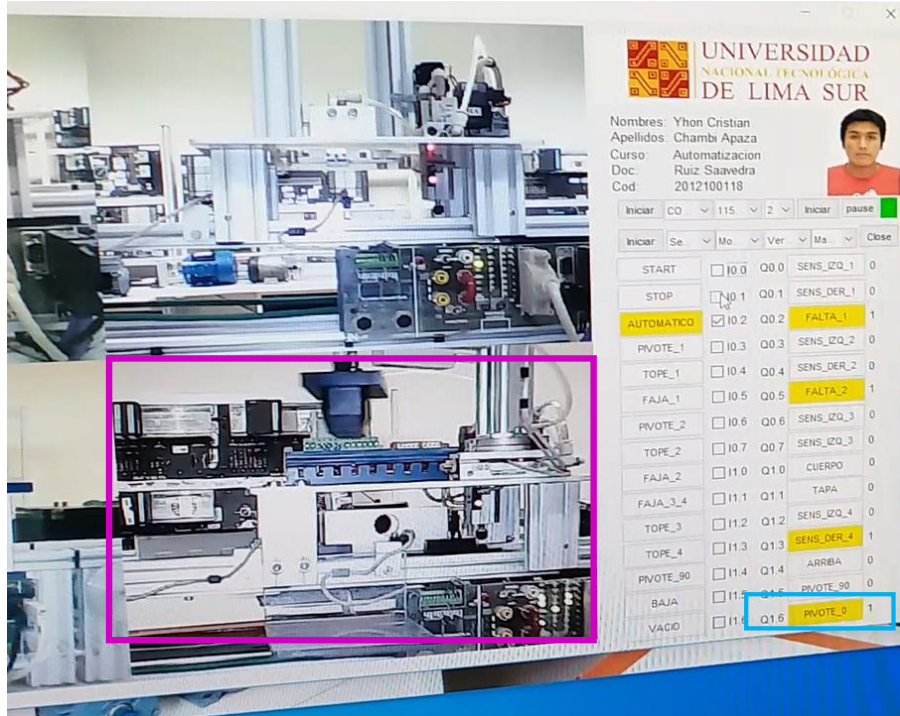


Figura 71. El pivote a 0° grados Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 72 se muestra como el pivote descende para dejar el bloque mal ensamblado, se puede apreciar también como la opción “arriba” se encuentra desactivada.

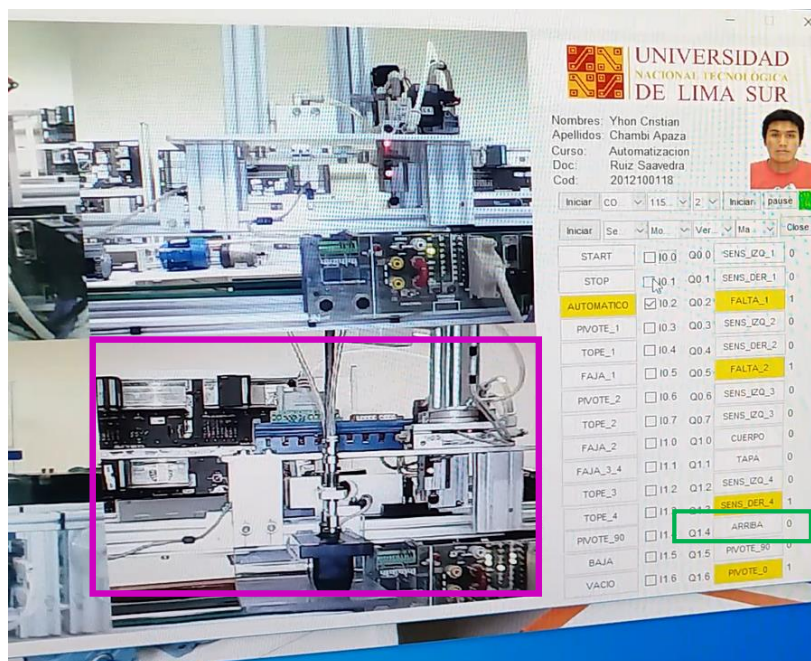


Figura 72. El pivote a 0° grados descende Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 73 se muestra cuando la válvula generadora de vacío se desactiva y posterior a ello, el pivote asciende hasta llegar a la parte de arriba. Como se muestra en la imagen al llegar a la parte superior la opción “arriba” lo detecta y lo muestra cambiando a color amarillo.

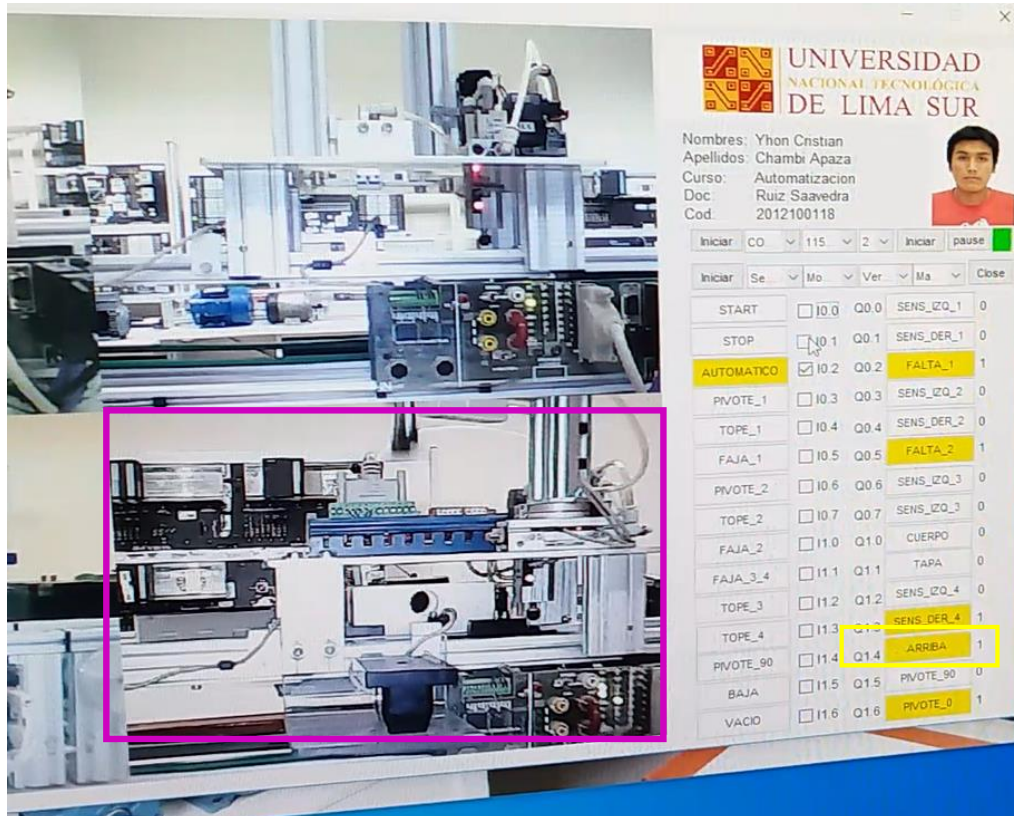


Figura 73. El pivote a 0° grados asciende Estación de Manipulación  
Fuente: Elaboración Propia

En la figura 74 se muestra la parada del proceso para continuar con el accionamiento manual de cada uno de los módulos. En este caso solo sería desactivar la opción de automático y proceder a emplear cada una de las opciones siguientes que se muestran en la interfaz, como la opción “pivote\_1”, “tope\_1”, “faja\_1”, etc. Cada una de estas opciones hacen que los módulos realicen esa determinada acción de activación o desactivación para cumplir con el proceso. La funcionalidad es la misma que la automática, sino que en este caso el operario tendrá el control de todo el sistema accionando cada una de sus partes.



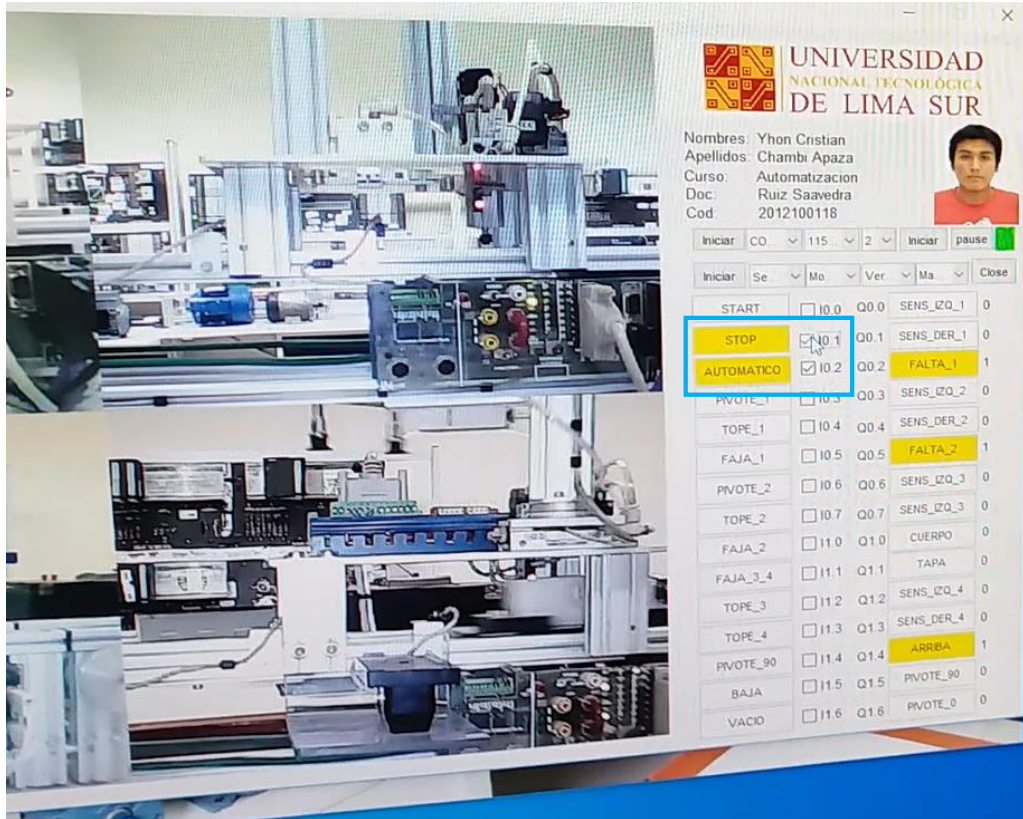


Figura 74. Parada del Proceso  
Fuente: Elaboración Propia

En la figura 75 se muestra el diagrama de Gantt en el tiempo, el inicio de la elaboración del trabajo es del 30 de Marzo al 03 de Septiembre del 2020 teniendo como tareas principales el diseño e implementación de la Tarjeta Electrónica, el Desarrollo de la Interfaz en Matlab y la Validación del funcionamiento del Sistema, teniendo en total una duración de 110 días.

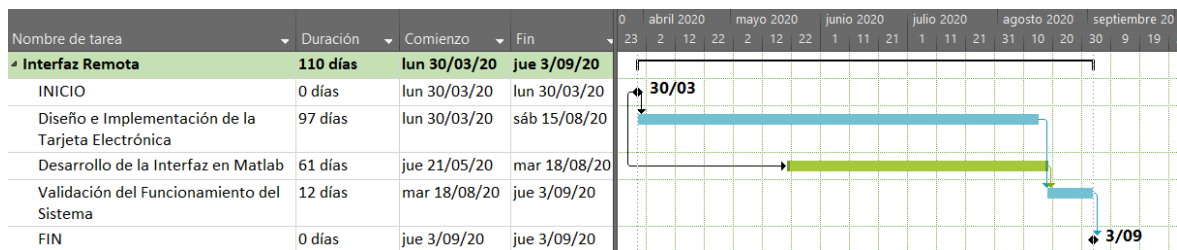


Figura 75. Diagrama de Gantt  
Fuente: Elaboración propia utilizando el software Microsoft Project

## **Análisis del aporte y contribución en base a la experiencia en las labores desarrolladas en el presente año.**

El puesto laboral que desempeñando en la Universidad Nacional Tecnología de Lima Sur es el de Asistente del Laboratorio de Control y Automatización, siendo las actividades propias del puesto detalladas a continuación:

- Resguardar, distribuir y controlar los instrumentos, equipos, materiales, herramientas, manuales y otros que se requieran para el desarrollo de las prácticas.
- Mantener en buen estado y operativos los Instrumentos, equipos y herramientas.
- Llevar al día el registro de equipos, instrumentos, herramientas y mobiliario del Laboratorio y colaborar con el inventario de las mismas.
- Mantener informado al Jefe de Laboratorio sobre altas, bajas y devoluciones de los bienes asignados, así como cualquier acción en perjuicio de los bienes de los laboratorios cometida por alumnos, profesores, personal administrativo o personal de servicio.
- Asesorar al usuario en las técnicas sobre el uso de equipo, herramientas y materiales.
- Vigilar que el laboratorio este siempre en condiciones de conservación e integridad de los Instrumentos, equipos, mobiliario, manuales y herramientas de operación y que cuente con las medidas de higiene y seguridad.
- Elaborar manuales para prácticas de laboratorio de los equipos que no lo poseen.
- Coordinar y apoyar a la Oficina de Planificación y presupuesto en cuanto a la elaboración de términos de referencia de requerimiento de equipos para los laboratorios.
- Actualizar el Software de los equipos de Cómputo de los laboratorios de forma periódica.

En base a las actividades antes mencionadas, se ha percibido problemas que son perjudiciales para los estudiantes que hacen uso del Laboratorio de Control y Automatización.

Debido al aislamiento social por el COVID-19 los estudiantes de la carrera de Ingeniería Electrónica y Telecomunicaciones realizan clases virtuales y no cuentan con laboratorios prácticos que son fundamentales para aplicar los

conocimientos teóricos, sobre todo los estudiantes que cursan los últimos ciclos; sin la posibilidad de realizar experiencias de laboratorio, la formación de los estudiantes queda restringida a los conocimientos conceptuales, en perjuicio de la sociedad y el país. Por eso, el desarrollo del presente trabajo de suficiencia profesional “Diseño e implementación de una interfaz remota para la educación a distancia de la estación automatizada de procesos industriales en la Universidad Nacional Tecnológica de Lima Sur, 2020” tiene una significancia de alcance nacional y será aplicado para mejorar el sistema de educación universitario a distancia aplicado en el ciclo 2020-II en la Universidad Nacional Tecnológica de Lima Sur.

Los cursos que se emplearon para el desarrollo de este proyecto se muestran en la tabla 3. El curso de programación para la ingeniería nos permitió desarrollar la programación en Matlab que es un software muy potente en el campo de la ingeniería, además de otros cursos adicionales como telecomunicaciones I, II, III estos cursos sirvieron para entender el funcionamiento de los diferentes dispositivos y sobre todo la parte teórica que fue de gran importancia para entender y diseñar el circuito de la tarjeta de adquisición electrónica. También se emplearon los cursos de Electrónica básica, intermedia y avanzada para el desarrollo y diseño de la configuración de resistencias y posición de transistores bjt para el control de la etapa de potencia y visualización del encendido de los indicadores led que se encuentran en el sistema. También se emplearon los cursos de automatización industrial, control II y Seminario de Tópicos Especiales para el desarrollo y programación de la estación automatizada de procesos industriales y también para el desarrollo del manejo a través de la interfaz en Matlab además del manejo de las cámaras Web.

Tabla 3. Cursos de mayor relevancia para la elaboración del proyecto

Descripción	Cursos
Se empleó para la elaboración de la Interfaz en Matlab	Programación para la Ingeniería
Entendimiento del concepto de la dirección de los dispositivos en la red y la forma en cómo se comunican entre sí	Telecomunicaciones I, II, III
Sirvió para el diseño de la etapa de potencia de la tarjeta de adquisición Electrónica	Electrónica Básica/Intermedia y Avanzada
Sirvió para la programación en los PLC	Automatización Industrial
Sirvió para entender el principio de control de los PLC	Control I, II
Sirvió para la programación tanto de la pantalla HMI y PLC	Seminario de Tópicos Especiales

Fuente: Propia

## 2.4 Resultados

La cantidad de estudiantes beneficiados en promedio son 120, debido a las prestaciones del dispositivo se salvaguardan las prácticas de laboratorio de los estudiantes del ciclo 2020-II de la carrera profesional de Ingeniería Electrónica y Telecomunicaciones de la Universidad Nacional Tecnológica de Lima Sur. Así podrán tener una fuerte competitividad a la hora de postular a un puesto de trabajo y estará en ventaja frente a sus competidores. A su vez el estudiante puede desenvolverse eficientemente en la industria y en consecuencia, podrá alcanzar un nivel de desarrollo personal y profesional.

Unos de los beneficios que conlleva el desarrollo de la Interfaz Remota es el de evitar el contagio y la propagación del COVID-19 reduciendo a 0% el índice de contagios, de esta forma se evitaría la propagación de la enfermedad y los estudiantes salvaguardarían de manera integra a su familia sin necesidad de asistir de manera presencial al laboratorio.

La interfaz remota también aporta en cuando al horario en el cual puede ser accedido por los estudiantes del curso, teniendo como hora de inicio las 8:00am hasta las 6:00pm en el cual se dispondría del laboratorio para hacer pruebas y trabajos dejados por el docente del curso, teniendo un total de 9 horas continuas.

Además, gracias a la interfaz remota cada estudiante puede hacer uso del sistema automatizado de procesos industriales de forma individual así todos podrían tener acceso a los equipos de laboratorio, anteriormente eran muy pocos los estudiantes que tenían acceso al equipo debido a que formaban grupos para hacer trabajos en los cuales solo el 20% tenían acceso a los equipos, ahora tendrán un acceso del 100% del alumnado, debido a la disponibilidad del laboratorio de control y automatización.

Además, se aprovecha eficientemente el uso de la pantalla HMI debido a que está integrado en el proyecto. Esta directamente conectado y listo para trabajar solo necesita de realizar la programación en el software WinCC Flexible para cargarlo. Anteriormente los estudiantes tenían problemas a la hora de subir sus programaciones porque tenían que configurarlo, ahora lo podrán hacer de forma remota sin necesidad de preocuparse por la parte física.



## CONCLUSIONES

- Se logró diseñar e Implementar la tarjeta de adquisición Electrónica y de potencia para activar y monitorear los puertos de entrada y salida del controlador lógico programable que gobierna al sistema automatizado de procesos industriales, mediante la detección de señales digitales que proceden del PLC y un circuito de potencia, configuración de transistores bjt y resistencias configuradas en división de tensión.
- Se integró cuatro cámaras web en la Interfaz en Matlab para obtener una apreciación en tiempo real del sistema automatizado de procesos industriales, estas cámaras una resolución de 640x480 pixeles garantizando así una resolución óptima para la visualización de cada uno de los equipos.
- Se desarrolló la Interfaz en Matlab para el manejo de la tarjeta de adquisición Electrónica y de potencia de manera que sea escalable para conectar otras tarjetas de adquisición con otra dirección, también la programación en Tia Portal y establecer la conexión remota a través del software Google Meet y Anydesk.
- Se validó exitosamente el funcionamiento del sistema de interfaz remota para la educación a distancia de la estación automatizada de procesos industriales en la Universidad Nacional Tecnológica de Lima Sur. En las pruebas que se realizó al dispositivo, se logró satisfactoriamente el registro de 5 estudiantes con los cuales se realizó las pruebas desde lugares remotos estableciendo la comunicación correctamente.

## RECOMENDACIONES

Como trabajos futuros se propone implementar la etapa del manejo de señales analógicas para controlar los PLC en otras aplicaciones presentes en la industria.

Además, se propone adicionar cámaras que permitan controlar sistemas complejos de tipo industrial. Se propone también añadir tarjetas de adquisición electrónica y así controlar los distintos módulos existentes en el laboratorio de control y automatización.

Para el control de la tarjeta de adquisición electrónica se recomienda el software de ingeniería LabView, este software también te permite desarrollar una interfaz de usuario y a la vez permite establecer la comunicación serial entre la interfaz y la tarjeta de adquisición electrónica.

La ventaja presente en el trabajo de suficiencia profesional es el de poder expandir este módulo hacia los demás laboratorios de electrónica como también a los de mecánica para que los estudiantes también puedan acceder a sus laboratorios de forma remota.

## BIBLIOGRAFÍA

- De la Cruz et al (2010). *Web-Labai Laboratorio Remoto de Automatización Industrial*. DOI: 10.4995/RIAI.2010.01.10
- Edel R. (2015). *Interfaz Web para Laboratorios Remotos en Dispositivos Móviles* (trabajo de diploma). Universidad Central "Marta Abreu" de las Villas. Santa Clara, Cuba. Recuperada de: <https://dspace.uclv.edu.cu/>
- Entrena G. (2015). *Instalación de Equipos y elementos de sistemas de automatización industrial*. Antequera, Málaga: IC Editorial S.A.
- Gardel V. et al. (2010). *El laboratorio de automatización a distancia utilizando un Grupo de máquinas virtuales*. DOI: 10.1109/TIE.2010.2041131
- Garduño V. (2005). *Enseñanza virtual sobre la organización de recursos informativos digitales*. México: Universidad Nacional Autónoma de México.
- Gómez J. (2010). *Servicios en Red*. Madrid, España: Editex S.A.
- Jernigan et al. (2009). *Implementar una experiencia de laboratorio remoto en un programa de grado de ingeniería conjunta: Levitación aerodinámica de una pelota de playa*. DOI: 10.1109/TE.2008.924217
- Juárez M. et al (2016). *Interfaz de Comunicación Remota entre un Sistema clasificador de Ondas Cerebrales y un Robot Móvil*. Revista de Aplicaciones de la Ingeniería, 3(8). Recuperada de: <https://www.ecorfan.org/>
- Martínez A. (2010). *Automatización y Telecontrol de Sistemas de Riego*. Barcelona, España: Marcombo, S.A. Recuperada de: <https://books.google.com.pe/>
- Nuevo G. (2020). *Montaje y Mantenimiento Eléctrico-Electrónico*. Madrid, España: Paraninfo S.A.
- Ramírez P. (2010). *Diseño e Implementación de una Interfaz para el Control de un módulo educativo utilizando una PC para un laboratorio Remoto-Virtual de Comunicaciones* (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima. Recuperada de: <http://hdl.handle.net/20.500.12404/517>
- Resolución Viceministerial N°105-202, "Regulan el ingreso del personal a sedes y filiales de los Centros de Educación Técnico - Productiva, Institutos y Escuelas de Educación Superior, Universidades públicas y privadas; para realizar, de manera excepcional, actividades que faciliten continuidad y mejora del servicio educativo que se viene prestando de manera no presencial". Recuperada de:

<https://busquedas.elperuano.pe/normaslegales/regulan-el-ingreso-del-personal-a-sedes-y-filiales-de-los-ce-resolucion-vice-ministerial-n-105-2020-minedu-1868135-1/>

Roesler et al (2015). *Los problemas de la evaluación de la educación a distancia en América Latina y el Caribe*. Recuperada de: <https://virtualeduca.org/>

Scolari C. (2018). *Las leyes de la Interfaz*. Barcelona, España: Gedisa, S.A.

Seippel. R. (2003). *Fundamentos de Electricidad*. Barcelona, España: Reverté S.A.

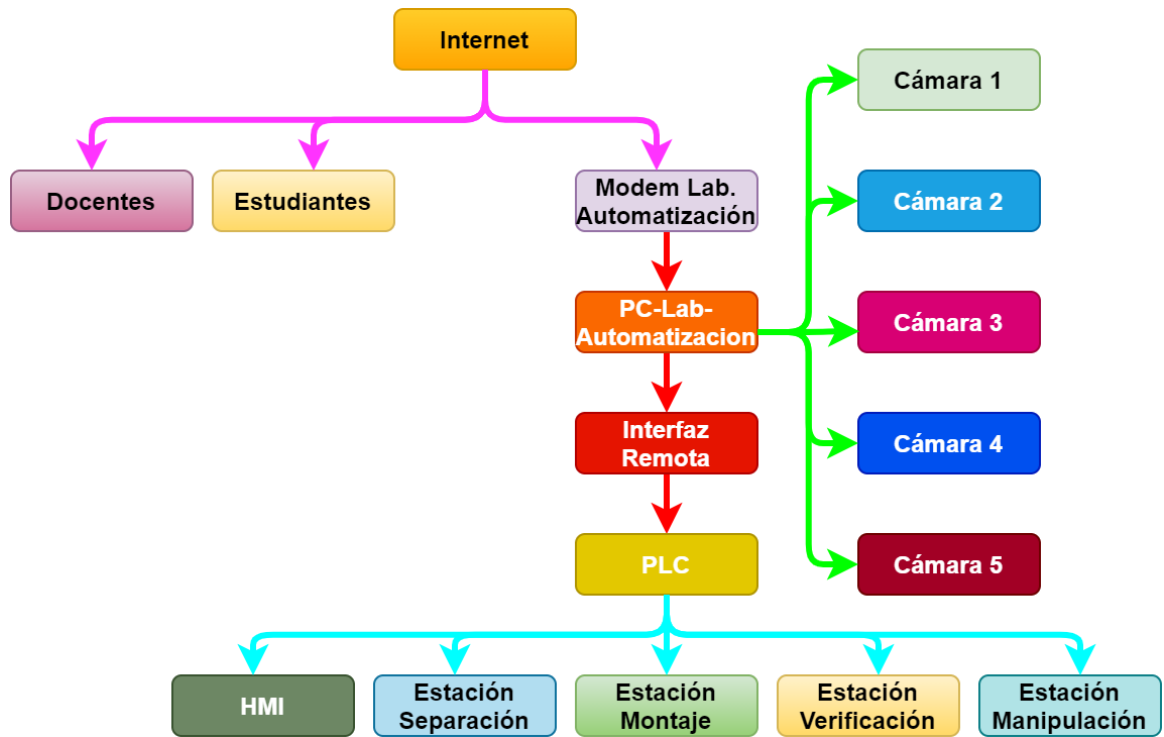
Tobarra et al. (2015). *Creación de Laboratorios Remotos personalizados usando la Deconstrucción*. DOI: 10.1109/RITA.2015.2418011

Tovar et al (2003). *Laboratorio Remotos de Física Básica*. Revista Colombiana de Física, 35(4). Rescatado de: <http://www.revcolfis.org/>

## **ANEXOS**

## Anexo A

### Diagrama de flujo del Trabajo de Suficiencia Profesional



## Anexo B

### Programación de la Inicialización de las Cámaras Web en Matlab

```
function iniciar_Callback(hObject, eventdata, handles)
% hObject handle to iniciar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global obj1 obj2 obj3 obj4
axes(handles.axes1);
obj1=videoinput('winvideo',primero,'YUY2_640x480');%640x480
resolution1=get(obj1,'VideoResolution');
bandas1= get(obj1,'NumberOfBands');
hImage1=image(zeros(resolution1(2),resolution1(1),bandas1));
preview(obj1,hImage1);
axes(handles.axes2);
obj2=videoinput('winvideo',segundo,'YUY2_640x480');
resolution2=get(obj2,'VideoResolution');
bandas2= get(obj2,'NumberOfBands');
hImage2=image(zeros(resolution2(2),resolution2(1),bandas2));
preview(obj2,hImage2);
axes(handles.axes3);
obj3=videoinput('winvideo',tercero,'YUY2_640x480');
resolution3=get(obj3,'VideoResolution');
bandas3= get(obj3,'NumberOfBands');
hImage3=image(zeros(resolution3(2),resolution3(1),bandas3));
preview(obj3,hImage3);
axes(handles.axes6);
obj4=videoinput('winvideo',cuarto,'YUY2_640x480');%640x480
resolution4=get(obj4,'VideoResolution');
bandas4= get(obj4,'NumberOfBands');
hImage4=image(zeros(resolution4(2),resolution4(1),bandas4));
preview(obj4,hImage4);
```



## Anexo C

### Inicialización del puerto Serie y demás código para el funcionamiento de la Interfaz Remota en Matlab

```
function Iniciar_Callback(hObject, eventdata, handles)
% hObject   handle to Iniciar (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
global pic1 uno1a uno1A uno1b uno1B uno1c uno1C uno1d uno1D uno1e uno1E uno1f uno1F
uno1g uno1G uno1h uno1H
seleccion=get(handles.popupmenu8,'Value');
npuerto=lista1(seleccion);
puerto=get(handles.popupmenu9,'Value');
if puerto==2
    puerto1=9600;
end
if puerto==3
    puerto1=115200;
end
delete(instrfind({'Port'},{npuerto}));
pic1=serial(npuerto,'BaudRate',puerto1);
fopen(pic1);
pause(1);
bucle=1;
set(handles.edit32,'BackgroundColor','green');
FS=1000;%250
pause2=0;
tic;
while 1
    if pause2==0
        if toc > (1/FS)
            tic;
            if uno1a==1
                fprintf(pic1,'%s',direccion);
                fprintf(pic1,'%s','a');
                uno1a=0;
            end
            if uno1A==1
                fprintf(pic1,'%s',direccion);
                fprintf(pic1,'%s','A');
                uno1A=0;
            end
            if uno1b==1
                fprintf(pic1,'%s',direccion);
                fprintf(pic1,'%s','b');
                uno1b=0;
            end
            if uno1B==1
                fprintf(pic1,'%s',direccion);
                fprintf(pic1,'%s','B');
                uno1B=0;
            end
        end
    end
end
```

```

end
if uno1c==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','c');
    uno1c=0;
end
if uno1C==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','C');
    uno1C=0;
end
if uno1d==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','d');
    uno1d=0;
end
if uno1D==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','D');
    uno1D=0;
end
if uno1e==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','e');
    uno1e=0;
end
if uno1E==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','E');
    uno1E=0;
end
if uno1f==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','f');
    uno1f=0;
end
if uno1F==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','F');
    uno1F=0;
end
if uno1g==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','g');
    uno1g=0;
end
if uno1G==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','G');
    uno1G=0;
end
if uno1h==1

```

```

    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'h');
    uno1h=0;
end
if uno1H==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'H');
    uno1H=0;
end
if uno1i==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'i');
    uno1i=0;
end
if uno1l==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'l');
    uno1l=0;
end
if uno1j==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'j');
    uno1j=0;
end
if uno1J==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'J');
    uno1J=0;
end
if uno1k==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'k');
    uno1k=0;
end
if uno1K==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'K');
    uno1K=0;
end
if uno1l==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'l');
    uno1l=0;
end
if uno1L==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'L');
    uno1L=0;
end
if uno1m==1
    fprintf(pic1, '%s', direccion);
    fprintf(pic1, '%s', 'm');

```

```

    uno1m=0;
end
if uno1M==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','M');
    uno1M=0;
end
if uno1n==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','n');
    uno1n=0;
end
if uno1N==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','N');
    uno1N=0;
end
if uno1o==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','o');
    uno1o=0;
end
if uno1O==1
    fprintf(pic1,'%s',direccion);
    fprintf(pic1,'%s','O');
    uno1O=0;
end
fprintf(pic1,'%s',direccion);
fprintf(pic1,'%s','z');
a=fscanf(pic1,'%s');
set(handles.text1,'String',a);
b=fscanf(pic1,'%s');
set(handles.text2,'String',b);
c=fscanf(pic1,'%s');
set(handles.text3,'String',c);
d=fscanf(pic1,'%s');
set(handles.text4,'String',d);
e=fscanf(pic1,'%s');
set(handles.text5,'String',e);
f=fscanf(pic1,'%s');
set(handles.text6,'String',f);
g=fscanf(pic1,'%s');
set(handles.text7,'String',g);
h=fscanf(pic1,'%s');
set(handles.text8,'String',h);
i=fscanf(pic1,'%s');
set(handles.text9,'String',i);
j=fscanf(pic1,'%s');
set(handles.text10,'String',j);
k=fscanf(pic1,'%s');
set(handles.text11,'String',k);
l=fscanf(pic1,'%s');

```

```

set(handles.text12,'String',l);
m=fscanf(pic1,'%s');
set(handles.text13,'String',m);
n=fscanf(pic1,'%s');
set(handles.text14,'String',n);
o=fscanf(pic1,'%s');
set(handles.text15,'String',o);
if a==49
    set(handles.edit3,'BackgroundColor','yellow');
else
    set(handles.edit3,'BackgroundColor','white');
end
if b==49
    set(handles.edit5,'BackgroundColor','yellow');
else
    set(handles.edit5,'BackgroundColor','white');
end
if c==49
    set(handles.edit7,'BackgroundColor','yellow');
else
    set(handles.edit7,'BackgroundColor','white');
end
if d==49
    set(handles.edit9,'BackgroundColor','yellow');
else
    set(handles.edit9,'BackgroundColor','white');
end
if e==49
    set(handles.edit11,'BackgroundColor','yellow');
else
    set(handles.edit11,'BackgroundColor','white');
end
if f==49
    set(handles.edit13,'BackgroundColor','yellow');
else
    set(handles.edit13,'BackgroundColor','white');
end
if g==49
    set(handles.edit15,'BackgroundColor','yellow');
else
    set(handles.edit15,'BackgroundColor','white');
end
if h==49
    set(handles.edit17,'BackgroundColor','yellow');
else
    set(handles.edit17,'BackgroundColor','white');
end
if i==49
    set(handles.edit19,'BackgroundColor','yellow');
else
    set(handles.edit19,'BackgroundColor','white');
end
end

```

```
if j==49
    set(handles.edit21,'BackgroundColor','yellow');
else
    set(handles.edit21,'BackgroundColor','white');
end
if k==49
    set(handles.edit23,'BackgroundColor','yellow');
else
    set(handles.edit23,'BackgroundColor','white');
end
if l==49
    set(handles.edit25,'BackgroundColor','yellow');
else
    set(handles.edit25,'BackgroundColor','white');
end
if m==49
    set(handles.edit27,'BackgroundColor','yellow');
else
    set(handles.edit27,'BackgroundColor','white');
end
if n==49
    set(handles.edit29,'BackgroundColor','yellow');
else
    set(handles.edit29,'BackgroundColor','white');
end
if o==49
    set(handles.edit31,'BackgroundColor','yellow');
else
    set(handles.edit31,'BackgroundColor','white');
end
end
Interface_Remota_V_1_0;
else
    break;
end
end
```



# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input  
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM),  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin  
PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during Sleep via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™  
(Master mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) – 8 bits wide with  
external  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{CS}$  controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital  
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference  
(VREF) module
  - Programmable input multiplexing from device  
inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash  
program memory typical
- 1,000,000 erase/write cycle Data EEPROM  
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)  
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

- Low-power, high-speed Flash/EEPROM  
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I <sup>2</sup> C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



## Anexo E

### Programación en MPLAB del Microcontrolador PIC 16F877A

```
LIST          P=16F877A
INCLUDE      <P16F877A.INC>
_CONFIG     _FOSC_HS & _WDT_OFF & _PWRTE_OFF & _BOREN_OFF & _LVP_OFF &
_CPD_OFF & _WRT_OFF & _DEBUG_OFF & _CP_OFF
```

```
CBLOCK 0x20
DIRECCION
VALOR_DIR
VALOR_ESTADO
ENDC
```

```
#DEFINE TX      PORTE,1
#DEFINE RX      PORTE,2
#DEFINE Q00     PORTB,0
#DEFINE Q01     PORTB,1
#DEFINE Q02     PORTB,2
#DEFINE Q03     PORTB,3
#DEFINE Q04     PORTB,4
#DEFINE Q05     PORTB,5
#DEFINE Q06     PORTB,6
#DEFINE Q07     PORTB,7
#DEFINE Q10     PORTA,0
#DEFINE Q11     PORTA,1
#DEFINE Q12     PORTA,2
#DEFINE Q13     PORTA,3
#DEFINE Q14     PORTA,4
#DEFINE Q15     PORTA,5
#DEFINE Q16     PORTE,0
#DEFINE I00     PORTD,6
#DEFINE I01     PORTD,5
#DEFINE I02     PORTD,4
#DEFINE I03     PORTC,7
#DEFINE I04     PORTC,6
#DEFINE I05     PORTC,5
#DEFINE I06     PORTD,2
#DEFINE I07     PORTD,3
#DEFINE I10     PORTC,4
#DEFINE I11     PORTD,1
#DEFINE I12     PORTD,0
#DEFINE I13     PORTC,3
#DEFINE I14     PORTC,2
#DEFINE I15     PORTC,1
#DEFINE I16     PORTC,0
#DEFINE I17     PORTD,7
```

```
ORG          0x00
CLRF         PORTA
CLRF         PORTB
CLRF         PORTC
CLRF         PORTD
CLRF         PORTE
MOVLW .50
MOVWF DIRECCION
CLRF         VALOR_DIR
CLRF         VALOR_ESTADO
BSF         STATUS,RP0
MOVLW .6
MOVWF ADCON1
CLRF         TRISA
CLRF         TRISB
MOVLW .4
```

```

MOVWF TRISE
MOVLW      .255
MOVWF     TRISC
MOVLW      .255
MOVWF     TRISD
BCF       STATUS,RP0
BSF       TX

INICIO
BTFSC     RX
GOTO      INICIO
BTFSC     RX
GOTO      INICIO
CALL      TIEMPO
CALL      TIEMPO_MEDIO
BTFSC     RX
GOTO      SET_BITD0
RETORNO_BITD0
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD1
RETORNO_BITD1
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD2
RETORNO_BITD2
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD3
RETORNO_BITD3
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD4
RETORNO_BITD4
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD5
RETORNO_BITD5
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD6
RETORNO_BITD6
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITD7
RETORNO_BITD7
CALL      TIEMPO
MOVF      DIRECCION,W
SUBWF     VALOR_DIR,W
BTFSC     STATUS,Z
GOTO      CONTINUA_MOS
CALL      TIEMPO_ESPERA
CLRF     VALOR_DIR
GOTO      INICIO
CONTINUA_MOS
BTFSC     RX
GOTO      CONTINUA_MOS
BTFSC     RX
GOTO      CONTINUA_MOS
CALL      TIEMPO
CALL      TIEMPO_MEDIO
BTFSC     RX
GOTO      SET_BITE0
RETORNO_BITE0
CALL      TIEMPO
BTFSC     RX
GOTO      SET_BITE1

```

```

RETORNO_BITE1
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE2
RETORNO_BITE2
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE3
RETORNO_BITE3
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE4
RETORNO_BITE4
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE5
RETORNO_BITE5
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE6
RETORNO_BITE6
  CALL      TIEMPO
  BTFSC    RX
  GOTO     SET_BITE7
RETORNO_BITE7
  CALL      TIEMPO
  MOVLW   .122
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    ESTADO_ENTRADA_I00
  MOVLW   .80
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,C
  GOTO    PRENDER
  GOTO    APAGAR

```

```

PRENDER
  MOVLW   .97
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q00
  MOVLW   .98
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q01
  MOVLW   .99
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q02
  MOVLW   .100
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q03
  MOVLW   .101
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q04
  MOVLW   .102
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q05
  MOVLW   .103
  SUBWF   VALOR_ESTADO,W
  BTFSC   STATUS,Z
  GOTO    PRENDER_Q06
  MOVLW   .104
  SUBWF   VALOR_ESTADO,W

```

```

BTFSC      STATUS,Z
GOTO       PRENDER_Q07
MOVLW     .105
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q10
MOVLW     .106
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q11
MOVLW     .107
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q12
MOVLW     .108
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q13
MOVLW     .109
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q14
MOVLW     .110
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q15
MOVLW     .111
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      PRENDER_Q16
GOTO      FINAL

```

#### APAGAR

```

MOVLW     .65
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q00
MOVLW     .66
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q01
MOVLW     .67
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q02
MOVLW     .68
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q03
MOVLW     .69
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q04
MOVLW     .70
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q05
MOVLW     .71
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q06
MOVLW     .72
SUBWF     VALOR_ESTADO,W
BTFSC     STATUS,Z
GOTO      APAGAR_Q07
MOVLW     .73
SUBWF     VALOR_ESTADO,W

```

```

    BTFSC      STATUS,Z
    GOTO       APAGAR_Q10
    MOVLW     .74
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q11
    MOVLW     .75
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q12
    MOVLW     .76
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q13
    MOVLW     .77
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q14
    MOVLW     .78
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q15
    MOVLW     .79
    SUBWF     VALOR_ESTADO,W
    BTFSC     STATUS,Z
    GOTO      APAGAR_Q16
    GOTO      FINAL

PRENDER_Q00
    BSF       Q00
    GOTO      FINAL
APAGAR_Q00
    BCF       Q00
    GOTO      FINAL
PRENDER_Q01
    BSF       Q01
    GOTO      FINAL
APAGAR_Q01
    BCF       Q01
    GOTO      FINAL
PRENDER_Q02
    BSF       Q02
    GOTO      FINAL
APAGAR_Q02
    BCF       Q02
    GOTO      FINAL
PRENDER_Q03
    BSF       Q03
    GOTO      FINAL
APAGAR_Q03
    BCF       Q03
    GOTO      FINAL
PRENDER_Q04
    BSF       Q04
    GOTO      FINAL
APAGAR_Q04
    BCF       Q04
    GOTO      FINAL
PRENDER_Q05
    BSF       Q05
    GOTO      FINAL
APAGAR_Q05
    BCF       Q05
    GOTO      FINAL
PRENDER_Q06
    BSF       Q06
    GOTO      FINAL

```

APAGAR_Q06		
BCF	Q06	
GOTO	FINAL	
PRENDER_Q07		
BSF	Q07	
GOTO	FINAL	
APAGAR_Q07		
BCF	Q07	
GOTO	FINAL	
PRENDER_Q10		
BSF	Q10	
GOTO	FINAL	
APAGAR_Q10		
BCF	Q10	
GOTO	FINAL	
PRENDER_Q11		
BSF	Q11	
GOTO	FINAL	
APAGAR_Q11		
BCF	Q11	
GOTO	FINAL	
PRENDER_Q12		
BSF	Q12	
GOTO	FINAL	
APAGAR_Q12		
BCF	Q12	
GOTO	FINAL	
PRENDER_Q13		
BSF	Q13	
GOTO	FINAL	
APAGAR_Q13		
BCF	Q13	
GOTO	FINAL	
PRENDER_Q14		
BSF	Q14	
GOTO	FINAL	
APAGAR_Q14		
BCF	Q14	
GOTO	FINAL	
PRENDER_Q15		
BSF	Q15	
GOTO	FINAL	
APAGAR_Q15		
BCF	Q15	
GOTO	FINAL	
PRENDER_Q16		
BSF	Q16	
GOTO	FINAL	
APAGAR_Q16		
BCF	Q16	
GOTO	FINAL	
FINAL		
CLRF	VALOR_DIR	
CLRF	VALOR_ESTADO	
GOTO	INICIO	
ESTADO_ENTRADA_I00		
BTFSC	I00	
GOTO	ES_UNO_I00	
GOTO	ES_CERO_I00	
ESTADO_ENTRADA_I01		
CALL	RETORNO_DE_CARRO	
CALL	NUEVA_LINEA	
BTFSC	I01	
GOTO	ES_UNO_I01	
GOTO	ES_CERO_I01	



ESTADO_ENTRADA_I02	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I02
	GOTO	ES_UNO_I02
	GOTO	ES_CERO_I02
ESTADO_ENTRADA_I03	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I03
	GOTO	ES_UNO_I03
	GOTO	ES_CERO_I03
ESTADO_ENTRADA_I04	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I04
	GOTO	ES_UNO_I04
	GOTO	ES_CERO_I04
ESTADO_ENTRADA_I05	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I05
	GOTO	ES_UNO_I05
	GOTO	ES_CERO_I05
ESTADO_ENTRADA_I06	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I06
	GOTO	ES_UNO_I06
	GOTO	ES_CERO_I06
ESTADO_ENTRADA_I07	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I07
	GOTO	ES_UNO_I07
	GOTO	ES_CERO_I07
ESTADO_ENTRADA_I10	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I10
	GOTO	ES_UNO_I10
	GOTO	ES_CERO_I10
ESTADO_ENTRADA_I11	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I11
	GOTO	ES_UNO_I11
	GOTO	ES_CERO_I11
ESTADO_ENTRADA_I12	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I12
	GOTO	ES_UNO_I12
	GOTO	ES_CERO_I12
ESTADO_ENTRADA_I13	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I13
	GOTO	ES_UNO_I13
	GOTO	ES_CERO_I13
ESTADO_ENTRADA_I14	CALL	RETORNO_DE_CARRO
	CALL	NUEVA_LINEA
	BTFSC	I14
	GOTO	ES_UNO_I14
	GOTO	ES_CERO_I14
ESTADO_ENTRADA_I15		

CALL	RETORNO_DE_CARRO
CALL	NUEVA_LINEA
BTFSC	I15
GOTO	ES_UNO_I15
GOTO	ES_CERO_I15
ESTADO_ENTRADA_I16	
CALL	RETORNO_DE_CARRO
CALL	NUEVA_LINEA
BTFSC	I16
GOTO	ES_UNO_I16
GOTO	ES_CERO_I16
ESTADO_ENTRADA_FINAL	
CALL	RETORNO_DE_CARRO
CALL	NUEVA_LINEA
CLRF	VALOR_DIR
CLRF	VALOR_ESTADO
GOTO	INICIO
ES_UNO_I00	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I01
ES_CERO_I00	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I01
ES_UNO_I01	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I02
ES_CERO_I01	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I02
ES_UNO_I02	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I03
ES_CERO_I02	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I03
ES_UNO_I03	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I04
ES_CERO_I03	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I04
ES_UNO_I04	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I05
ES_CERO_I04	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I05
ES_UNO_I05	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I06
ES_CERO_I05	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I06
ES_UNO_I06	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I07
ES_CERO_I06	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I07
ES_UNO_I07	
CALL	TRANSMITIR_UNO
GOTO	ESTADO_ENTRADA_I10
ES_CERO_I07	
CALL	TRANSMITIR_CERO
GOTO	ESTADO_ENTRADA_I10
ES_UNO_I10	

```

CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I11
ES_CERO_I10
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I11
ES_UNO_I11
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I12
ES_CERO_I11
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I12
ES_UNO_I12
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I13
ES_CERO_I12
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I13
ES_UNO_I13
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I14
ES_CERO_I13
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I14
ES_UNO_I14
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I15
ES_CERO_I14
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I15
ES_UNO_I15
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_I16
ES_CERO_I15
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_I16
ES_UNO_I16
CALL      TRANSMITIR_UNO
GOTO     ESTADO_ENTRADA_FINAL
ES_CERO_I16
CALL      TRANSMITIR_CERO
GOTO     ESTADO_ENTRADA_FINAL

```

```

SET_BITD0
MOVLW    .1
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD0
SET_BITD1
MOVLW    .2
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD1
SET_BITD2
MOVLW    .4
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD2
SET_BITD3
MOVLW    .8
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD3
SET_BITD4
MOVLW    .16
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD4
SET_BITD5
MOVLW    .32
ADDWF   VALOR_DIR,F
GOTO    RETORNO_BITD5
SET_BITD6

```

```

        MOVLW        .64
        ADDWF VALOR_DIR,F
        GOTO        RETORNO_BITD6
SET_BITD7
        MOVLW        .128
        ADDWF VALOR_DIR,F
        GOTO        RETORNO_BITD7
SET_BITE0
        MOVLW        .1
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE0
SET_BITE1
        MOVLW        .2
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE1
SET_BITE2
        MOVLW        .4
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE2
SET_BITE3
        MOVLW        .8
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE3
SET_BITE4
        MOVLW        .16
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE4
SET_BITE5
        MOVLW        .32
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE5
SET_BITE6
        MOVLW        .64
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE6
SET_BITE7
        MOVLW        .128
        ADDWF VALOR_ESTADO,F
        GOTO        RETORNO_BITE7

TRANSMITIR_UNO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BSF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BSF          TX
        CALL         TIEMPO_ENVIO
        BSF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BSF          TX
        CALL         TIEMPO_ENVIO
        RETURN

TRANSMITIR_CERO
        BCF          TX
        CALL         TIEMPO_ENVIO
        BCF          TX

```

```
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
RETURN
```

#### RETORNO\_DE\_CARRO

```
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
RETURN
```

#### NUEVA\_LINEA

```
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BCF       TX
CALL      TIEMPO_ENV IO
BSF       TX
CALL      TIEMPO_ENV IO
RETURN
INCLUDE <RETARDOS.INC>
```

END



# MICROCHIP PIC16F627A/628A/648A

## 18-pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

### High-Performance RISC CPU:

- Operating speeds from DC – 20 MHz
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- 35 single-word instructions:
  - All instructions single cycle except branches

### Special Microcontroller Features:

- Internal and external oscillator options:
  - Precision internal 4 MHz oscillator factory calibrated to  $\pm 1\%$
  - Low-power internal 48 kHz oscillator
  - External Oscillator support for crystals and resonators
- Power-saving Sleep mode
- Programmable weak pull-ups on PORTB
- Multiplexed Master Clear/Input-pin
- Watchdog Timer with independent oscillator for reliable operation
- Low-voltage programming
- In-Circuit Serial Programming™ (via two pins)
- Programmable code protection
- Brown-out Reset
- Power-on Reset
- Power-up Timer and Oscillator Start-up Timer
- Wide operating voltage range (2.0-5.5V)
- Industrial and extended temperature range
- High-Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - 40 year data retention

### Low-Power Features:

- Standby Current:
  - 100 nA @ 2.0V, typical
- Operating Current:
  - 12  $\mu$ A @ 32 kHz, 2.0V, typical
  - 120  $\mu$ A @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1  $\mu$ A @ 2.0V, typical
- Timer1 Oscillator Current:
  - 1.2  $\mu$ A @ 32 kHz, 2.0V, typical
- Dual-speed Internal Oscillator:
  - Run-time selectable between 4 MHz and 48 kHz
  - 4  $\mu$ s wake-up from Sleep, 3.0V, typical

### Peripheral Features:

- 16 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Selectable internal or external reference
  - Comparator outputs are externally accessible
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Timer1: 16-bit timer/counter with external crystal/clock capability
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
  - 16-bit Capture/Compare
  - 10-bit PWM
- Addressable Universal Synchronous/Asynchronous Receiver/Transmitter USART/SCI

Device	Program Memory	Data Memory		I/O	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1



## Anexo G

### Programación en MPLAB del microcontrolador PIC 16F628A para el Manejo de la pantalla LCD 16x2

```
LIST          P=16F628A
INCLUDE      <P16F628A.INC>
__CONFIG    _EXTCLK_OSC & _WDTE_OFF & _PWRTE_OFF & _MCLRE_OFF & _BOREN_OFF &
_LVP_OFF & _CPD_OFF & _CP_OFF

CBLOCK 0x20
DIRECCION
VALOR_DIR
VALOR_ESTADO
ENDC

#DEFINE     RS          PORTA,1
#DEFINE     ENABLE     PORTA,0
#DEFINE     D7         PORTB,4
#DEFINE     D6         PORTB,5
#DEFINE     D5         PORTB,6
#DEFINE     D4         PORTB,7
#DEFINE     RX         PORTB,0

ORG         0x00
MOVLW      .7
MOVWF      CMCON
CLRF      PORTA
CLRF      PORTB
MOVLW     .50
MOVWF     DIRECCION
CLRF     VALOR_DIR
CLRF     VALOR_ESTADO
BSF     STATUS,RP0
CLRF     TRISA
MOVLW   .1
MOVWF   TRISB
BCF     STATUS,RP0
CALL   Retardo_1s

INICIO_LCD
BCF     D7
BCF     D6
BSF     D5
BCF     D4
BSF     ENABLE
CALL   TIEMPO_LCD
BCF     ENABLE
BCF     D7
BCF     D6
BSF     D5
BCF     D4
BSF     ENABLE
CALL   TIEMPO_LCD
BCF     ENABLE
BSF     D7
BCF     D6
BCF     D5
BCF     D4
BSF     ENABLE
CALL   TIEMPO_LCD
BCF     ENABLE
BCF     D7
BCF     D6
```

BCF	D5
BCF	D4
BSF	ENABLE
CALL	TIEMPO_LCD
BCF	ENABLE
BSF	D7
BSF	D6
BCF	D5
BCF	D4
BSF	ENABLE
CALL	TIEMPO_LCD
BCF	ENABLE
BCF	D7
BCF	D6
BCF	D5
BCF	D4
BSF	ENABLE
CALL	TIEMPO_LCD
BCF	ENABLE
BCF	D7
BCF	D6
BCF	D5
BSF	D4
BSF	ENABLE
CALL	TIEMPO_LCD
BCF	ENABLE
BCF	D7
BCF	D6
BCF	D5
BCF	D4
CALL	Retardo_5ms
CALL	Retardo_2ms
CALL	Retardo_1ms
CALL	POSICION_03
CALL	LETRA_U
CALL	LETRA_N
CALL	LETRA_T
CALL	LETRA_E
CALL	LETRA_L
CALL	LETRA_S
CALL	LETRA_v
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_0
INICIO	
BTFSC	RX
GOTO	INICIO
BTFSC	RX
GOTO	INICIO
CALL	TIEMPO
CALL	TIEMPO_MEDIO
BTFSC	RX
GOTO	SET_BITD0
RETORNO_BITD0	
CALL	TIEMPO
BTFSC	RX
GOTO	SET_BITD1
RETORNO_BITD1	
CALL	TIEMPO
BTFSC	RX
GOTO	SET_BITD2
RETORNO_BITD2	
CALL	TIEMPO
BTFSC	RX
GOTO	SET_BITD3
RETORNO_BITD3	

	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITD4
RETORNO_BITD4		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITD5
RETORNO_BITD5		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITD6
RETORNO_BITD6		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITD7
RETORNO_BITD7		
	CALL	TIEMPO
	MOVF	DIRECCION,W
	SUBWF	VALOR_DIR,W
	BTFSC	STATUS,Z
	GOTO	CONTINUA MOS
	CALL	TIEMPO_ESPERA
	CLRF	VALOR_DIR
	GOTO	INICIO
CONTINUA MOS		
	BTFSC	RX
	GOTO	CONTINUA MOS
	BTFSC	RX
	GOTO	CONTINUA MOS
	CALL	TIEMPO
	CALL	TIEMPO_MEDIO
	BTFSC	RX
	GOTO	SET_BITE0
RETORNO_BITE0		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE1
RETORNO_BITE1		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE2
RETORNO_BITE2		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE3
RETORNO_BITE3		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE4
RETORNO_BITE4		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE5
RETORNO_BITE5		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE6
RETORNO_BITE6		
	CALL	TIEMPO
	BTFSC	RX
	GOTO	SET_BITE7
RETORNO_BITE7		
	CALL	TIEMPO
	MOVLW	.80
	SUBWF	VALOR_ESTADO,W
	BTFSC	STATUS,C
	GOTO	PRENDER

```

GOTO          APAGAR

PRENDER
MOVLW .97
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q00
MOVLW .98
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q01
MOVLW .99
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q02
MOVLW .100
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q03
MOVLW .101
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q04
MOVLW .102
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q05
MOVLW .103
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q06
MOVLW .104
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q07
MOVLW .105
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q10
MOVLW .106
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q11
MOVLW .107
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q12
MOVLW .108
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q13
MOVLW .109
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q14
MOVLW .110
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q15
MOVLW .111
SUBWF VALOR_ESTADO,W
BTFSC      STATUS,Z
GOTO      PRENDER_Q16
GOTO      FINAL

APAGAR
MOVLW .65

```

```

SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q00
MOVLW .66
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q01
MOVLW .67
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q02
MOVLW .68
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q03
MOVLW .69
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q04
MOVLW .70
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q05
MOVLW .71
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q06
MOVLW .72
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q07
MOVLW .73
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q10
MOVLW .74
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q11
MOVLW .75
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q12
MOVLW .76
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q13
MOVLW .77
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q14
MOVLW .78
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q15
MOVLW .79
SUBWF VALOR_ESTADO,W
BTFSC STATUS,Z
GOTO APAGAR_Q16
GOTO FINAL

```

ENTRADA

```

CALL POSICION_10
CALL LETRA_E
CALL LETRA_n
CALL LETRA_t
CALL LETRA_r

```

CALL	LETRA_a
CALL	LETRA_d
CALL	LETRA_a
CALL	LETRA_l
RETURN	
PRENDER_Q00	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_0
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPA CIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q00	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_0
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q01	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_1
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPA CIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q01	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_1
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q02	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_2
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPA CIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q02	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_2
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q03	
CALL	ENTRADA

CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_3
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q03	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_3
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q04	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_4
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q04	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_4
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q05	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_5
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q05	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_5
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q06	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_6
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N



GOTO	FINAL
APAGAR_Q06	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_6
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q07	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_7
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q07	
CALL	ENTRADA
CALL	LETRA_0
CALL	LETRA_PUNTO
CALL	LETRA_7
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q10	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_0
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q10	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_0
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q11	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_1
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q11	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_1
CALL	LETRA_DOS_PUNTOS

CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q12	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_2
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q12	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_2
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q13	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_3
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q13	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_3
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q14	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_4
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_ESPACIO
CALL	LETRA_O
CALL	LETRA_N
GOTO	FINAL
APAGAR_Q14	
CALL	ENTRADA
CALL	LETRA_1
CALL	LETRA_PUNTO
CALL	LETRA_4
CALL	LETRA_DOS_PUNTOS
CALL	LETRA_O
CALL	LETRA_F
CALL	LETRA_F
GOTO	FINAL
PRENDER_Q15	
CALL	ENTRADA
CALL	LETRA_1

```

CALL      LETRA_PUNTO
CALL      LETRA_5
CALL      LETRA_DOS_PUNTOS
CALL      LETRA_ESPACIO
CALL      LETRA_O
CALL      LETRA_N
GOTO      FINAL
APAGAR_Q15
CALL      ENTRADA
CALL      LETRA_1
CALL      LETRA_PUNTO
CALL      LETRA_5
CALL      LETRA_DOS_PUNTOS
CALL      LETRA_O
CALL      LETRA_F
CALL      LETRA_F
GOTO      FINAL
PRENDER_Q16
CALL      ENTRADA
CALL      LETRA_1
CALL      LETRA_PUNTO
CALL      LETRA_6
CALL      LETRA_DOS_PUNTOS
CALL      LETRA_ESPACIO
CALL      LETRA_O
CALL      LETRA_N
GOTO      FINAL
APAGAR_Q16
CALL      ENTRADA
CALL      LETRA_1
CALL      LETRA_PUNTO
CALL      LETRA_6
CALL      LETRA_DOS_PUNTOS
CALL      LETRA_O
CALL      LETRA_F
CALL      LETRA_F
GOTO      FINAL
FINAL
CLRF      VALOR_DIR
CLRF      VALOR_ESTADO
GOTO      INICIO
SET_BITD0
MOVLW    .1
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD0
SET_BITD1
MOVLW    .2
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD1
SET_BITD2
MOVLW    .4
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD2
SET_BITD3
MOVLW    .8
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD3
SET_BITD4
MOVLW    .16
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD4
SET_BITD5
MOVLW    .32
ADDWF    VALOR_DIR,F
GOTO     RETORNO_BITD5

```

```

SET_BITD6
    MOVLW    .64
    ADDWF   VALOR_DIR,F
    GOTO    RETORNO_BITD6
SET_BITD7
    MOVLW    .128
    ADDWF   VALOR_DIR,F
    GOTO    RETORNO_BITD7
SET_BITE0
    MOVLW    .1
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE0
SET_BITE1
    MOVLW    .2
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE1
SET_BITE2
    MOVLW    .4
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE2
SET_BITE3
    MOVLW    .8
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE3
SET_BITE4
    MOVLW    .16
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE4
SET_BITE5
    MOVLW    .32
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE5
SET_BITE6
    MOVLW    .64
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE6
SET_BITE7
    MOVLW    .128
    ADDWF   VALOR_ESTADO,F
    GOTO    RETORNO_BITE7

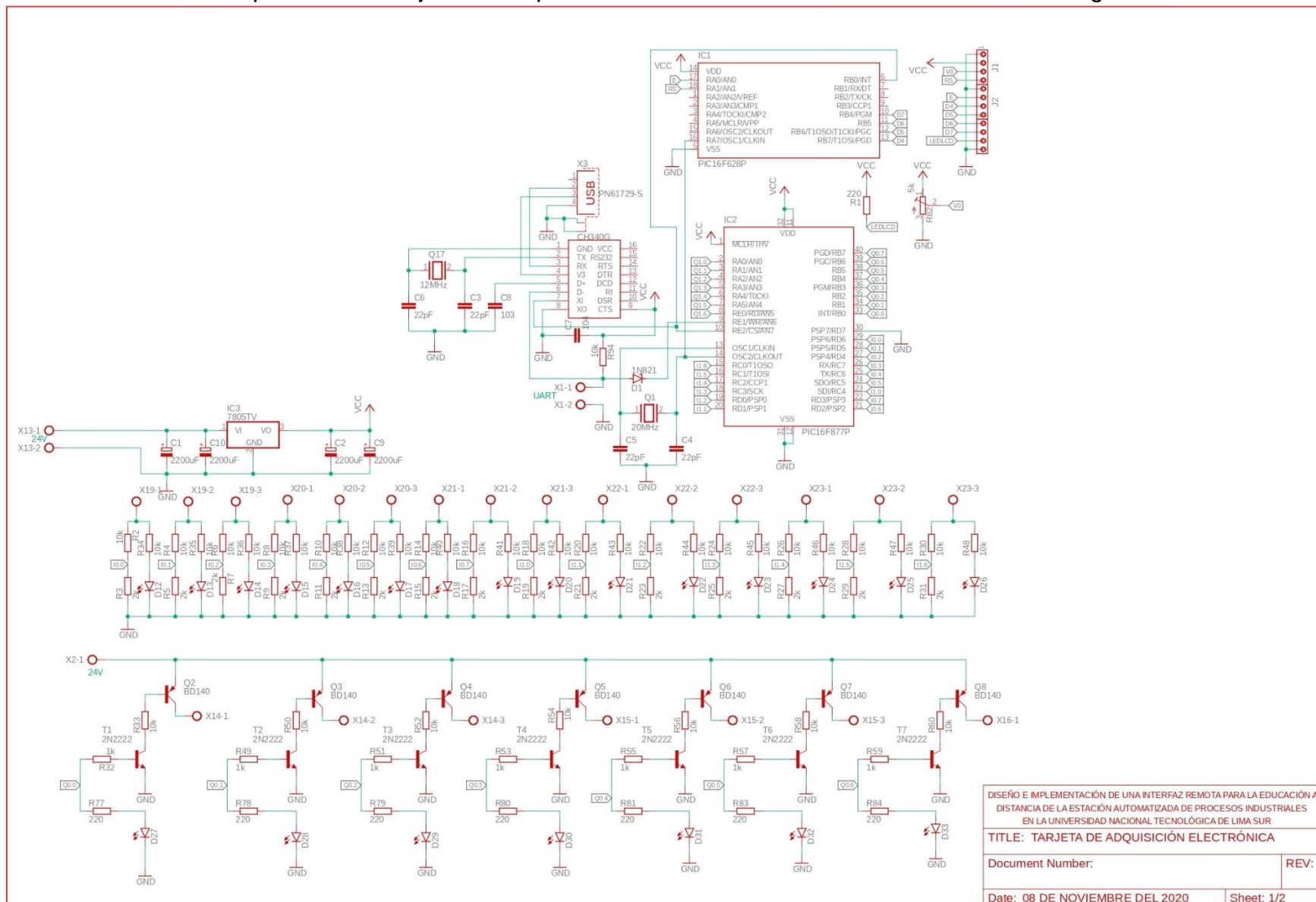
    INCLUDE <RETARDOS.INC>

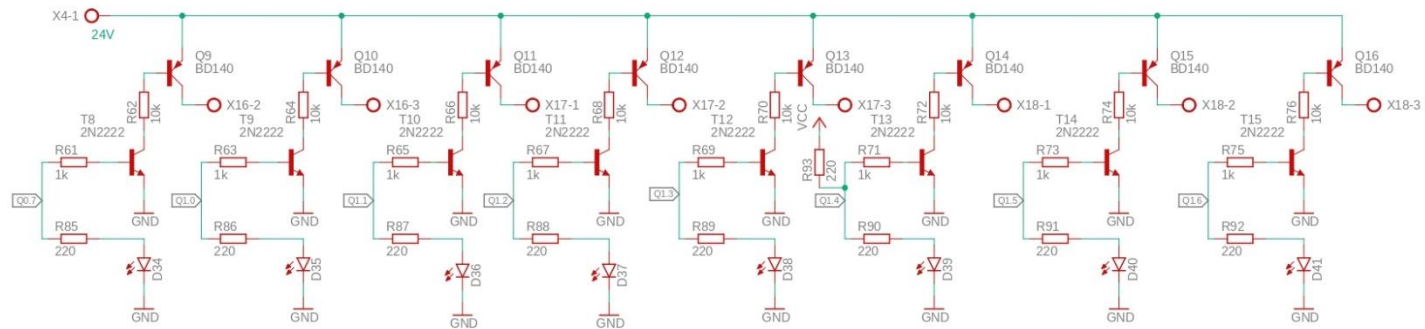
    END

```

## Anexo H

### Esquema de la Tarjeta de Adquisición Electrónica desarrollada en el Software Eagle





DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ REMOTA PARA LA EDUCACIÓN A DISTANCIA DE LA ESTACIÓN AUTOMATIZADA DE PROCESOS INDUSTRIALES EN LA UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

TITLE: TARJETA DE ADQUISICIÓN ELECTRÓNICA

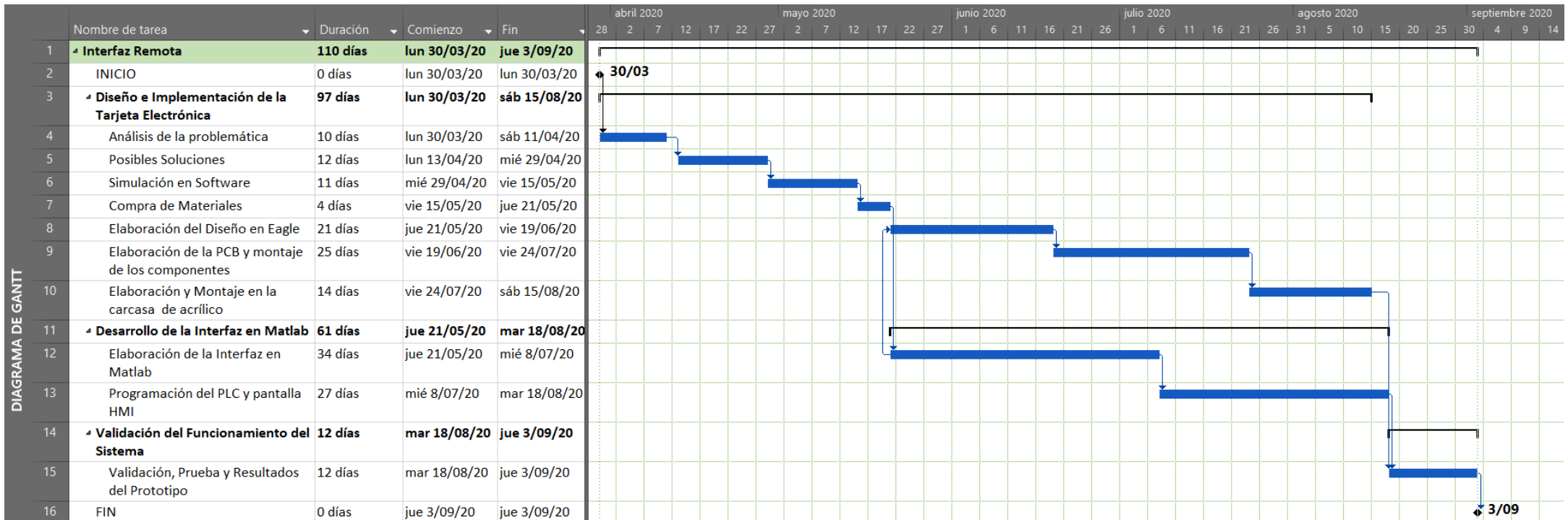
Document Number: \_\_\_\_\_ REV: \_\_\_\_\_

Date: 08 DE NOVIEMBRE DEL 2020 Sheet: 2/2

# Anexo I

## Diagrama de Gantt

Fecha de Inicialización del Proyecto 30 de Marzo al 03 de Septiembre del 2020





## USB to serial chip CH340

English DataSheet

Version: 1D

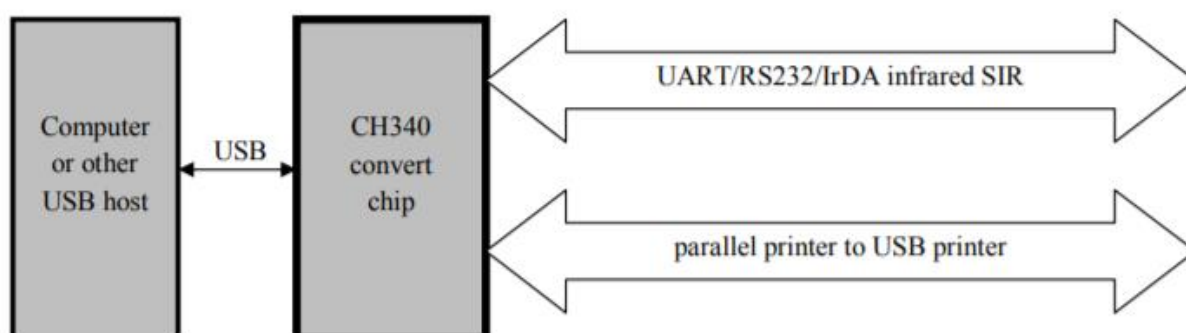
<http://wch.cn>

### 1. Introduction

CH340 is a USB bus convert chip and it can realize USB convert to serial interface, USB convert to IrDA infrared or USB convert to printer interface.

In serial interface mode, CH340 supplies common MODEM liaison signal, used to enlarge asynchronous serial interface of computer or upgrade the common serial device to USB bus directly. More detail about USB convert to printer interface please referring to the second manual CH340DS2.

In infrared mode, add infrared transceiver to CH340 can compose USB infrared adapter, realize SIR infrared communication.



### 2. Features

- Full speed USB device interface, conforms to USB Specification Version 2.0, only needs crystal and capacitance external.
- Emulate standard serial interface, used to upgrade the former peripheral device, or add excess serial interface through USB.
- Totally compatible with serial application program in computer endpoint Windows operation system.
- Hardware full duplex serial interface, set transceiver buffer, supports communication baud rate varies from 50bps to 2Mbps.
- Supports common MODEM liaison signal RTS, DTR, DCD, RI, DSR and CTS.
- Through adding level converter equipment to supply RS232, RS485, RS422 and other interface.
- Supports IrDA criterion SIR infrared communication, supports baud rate varies from 2400bps to 115200bps.
- For it is through USB converts to serial interface, only compatible with application layer not totally.
- Software compatible with CH341, using drive of CH341 directly.
- Support 5V and 3.3V source voltage.
- Supply SSOP-20 package without lead, compatible with RoHS.

## Anexo K

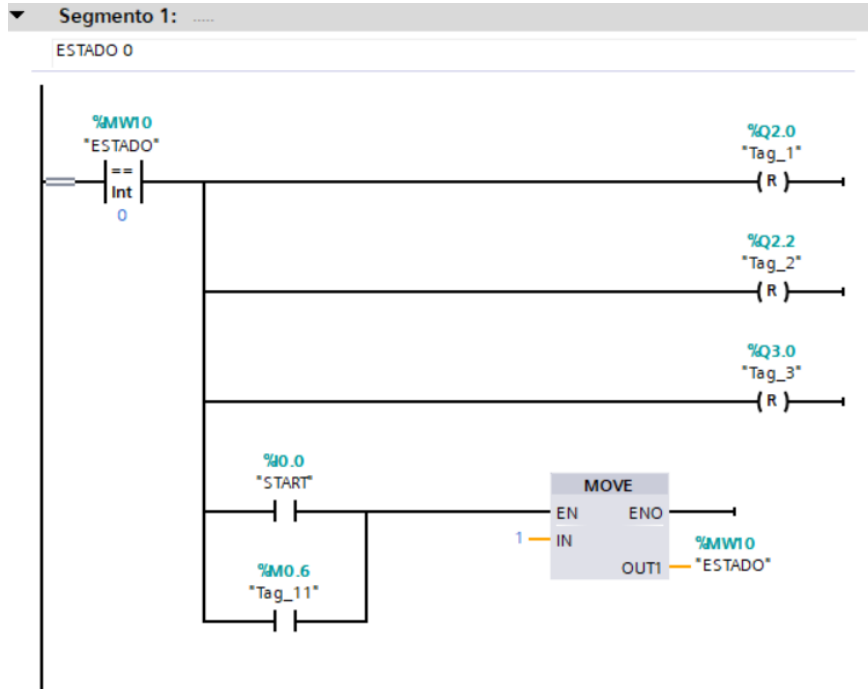
### Tabla ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

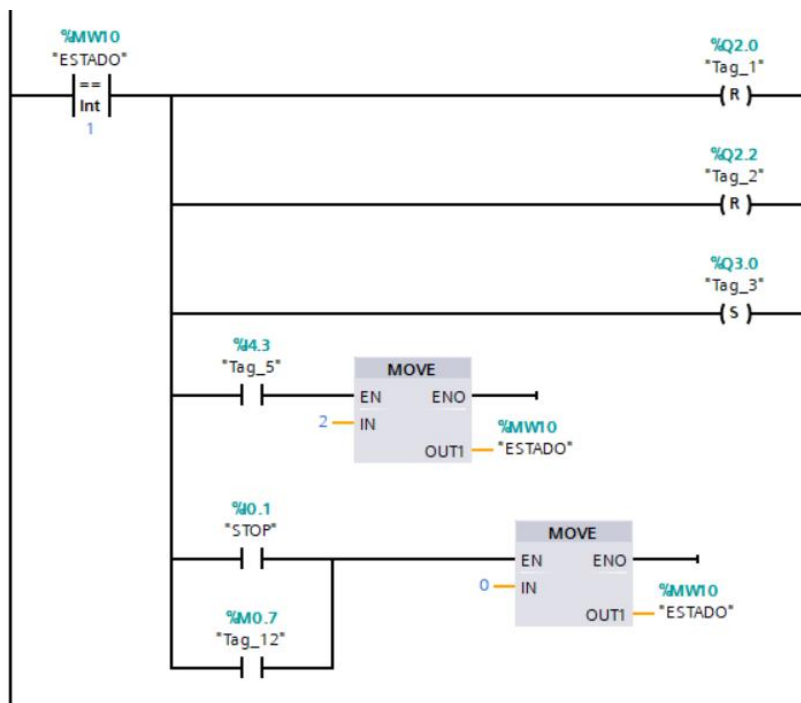
## Anexo L

Programación en TIA Portal de la Estación Automatizada de Procesos Industriales  
de la Universidad Nacional Tecnológica de Lima Sur

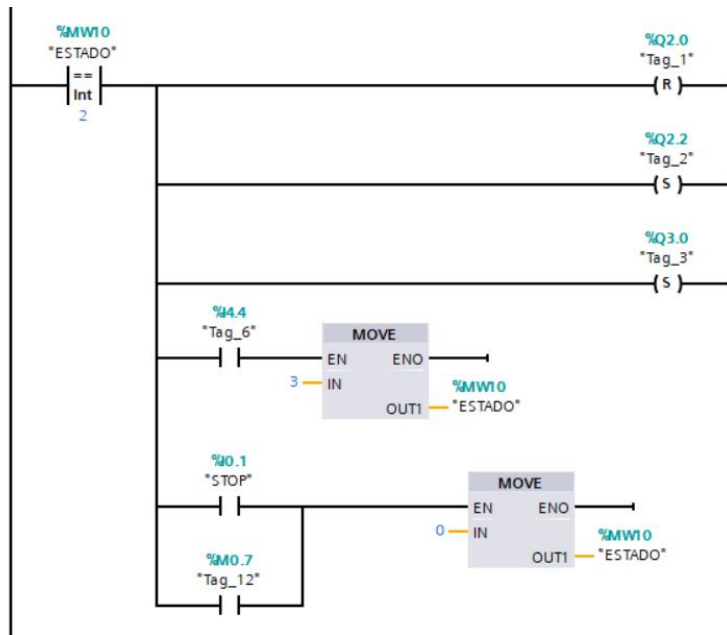
### Programación del Sistema de Separación



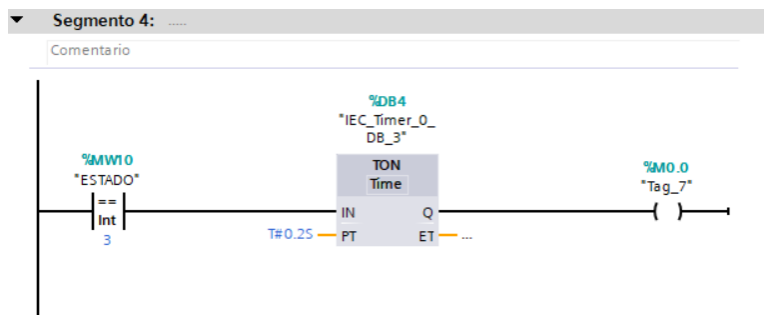
### Segmento 1



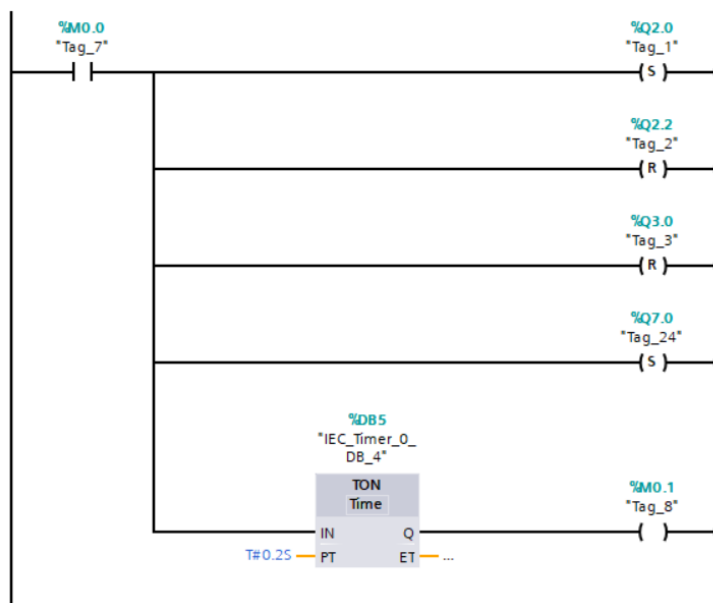
### Segmento 2



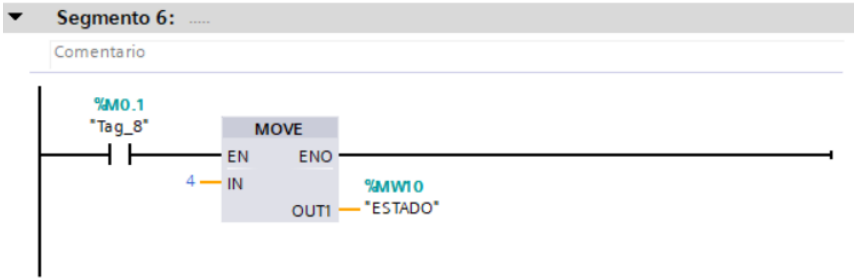
### Segmento 3



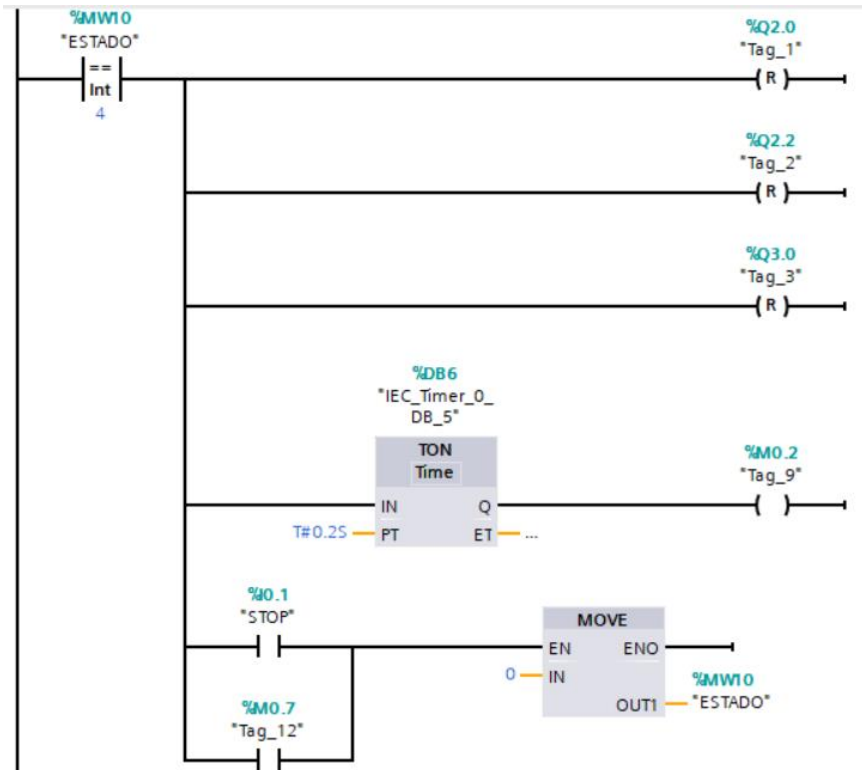
### Segmento 4



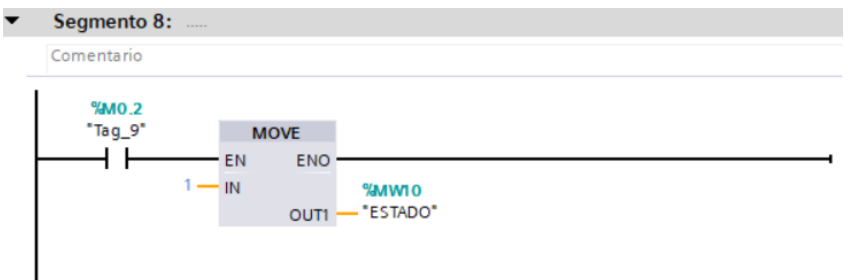
### Segmento 5



Segmento 6



Segmento 7



Segmento 8