



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL  
CARRERA DE INGENIERÍA INDUSTRIAL EN PROCESOS DE  
AUTOMATIZACIÓN**

**TEMA**

---

**“DESARROLLO DE UN SCADA PARA UNA PLANTA SIMULADA DE  
PRODUCCIÓN DE VIDRIO TEMPLADO”**

---

Proyecto de Trabajo de Graduación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero Industrial en Procesos de Automatización.

**SUBLÍNEA DE INVESTIGACIÓN:** Sistemas de control automatizados e instrumentación virtual para procesos industriales de baja y alta potencia.

**AUTOR:** Henry Gonzalo Alvarado Jaya

**TUTOR:** Ing. Patricio Germán Encalada Ruiz Mg.

Ambato – Ecuador

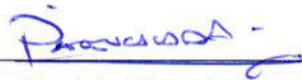
Enero – 2019

## CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: "DESARROLLO DE UN SCADA PARA UNA PLANTA SIMULADA DE PRODUCCIÓN DE VIDRIO TEMPLADO", elaborado por el señor Henry Gonzalo Alvarado Jaya, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos y méritos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación del Instructivo de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato enero, 2019

EL TUTOR



---

Ing. Mg. Patricio Germán Encalada Ruiz

## AUTORÍA

El presente Proyecto de Investigación titulado: "DESARROLLO DE UN SACADA PARA UNA PLANTA SIMULADA DE PRODUCCIÓN DE VIDRIO TEMPLADO", es de mi autoría, siendo el mismo personal, original y único, por tal razón el contenido, comentarios, conclusiones, recomendaciones y efectos legales son de exclusiva responsabilidad del autor.

Ambato enero, 2019

## AUTOR



---

Henry Gonzalo Alvarado Jaya

CC: 150090031-9

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato enero, 2019



---

Henry Gonzalo Alvarado Jaya

CC: 1500900319

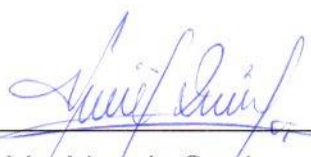
## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes calificadores, revisó y aprobó el Informe Final del Proyecto de Investigación titulado "DESARROLLO DE UN SCADA PARA UNA PLANTA SIMULADA DE PRODUCCIÓN DE VIDRIO TEMPLADO", presentado por el señor Henry Gonzalo Alvarado Jaya de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



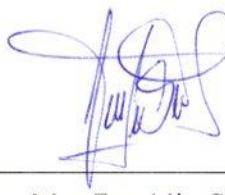
---

Ing. Mg. Elsa Pilar Urrutia Urrutia  
PRESIDENTA DEL TRIBUNAL



---

Ing. Mg. Marcelo García  
DOCENTE CALIFICADOR



---

Ing. Mg. Franklin Salazar  
DOCENTE CALIFICADOR

## **DEDICATORIA:**

*Dedico de manera muy especial a mi padre Gonzalo Alvarado y a mi madre Doria Jaya pues ellos fueron el principal cimiento para la construcción de mi vida profesional, sentó en mí las bases de responsabilidad y deseos de superación, pues en ellos tengo el espejo en el cual me quiero reflejar pues sus virtudes infinitas y su gran corazón me llevan cada día a admirarlos.*

*Henry Gonzalo Alvarado Jaya*

## **AGRADECIMIENTO:**

*A mis padres por apoyarme y brindarme su sabiduría y amor. A la Universidad Técnica de Ambato, especialmente a las autoridades y docentes que conforma la Facultad de Ingeniería en Sistemas Electrónica e Industrial y particularmente al Ing. Patricio Encalada por impartir sus enseñanzas y guiarme en el desarrollo de esta investigación. A la Organización de Mantenimiento Aprobada por la apertura brindada para realizar el presente proyecto de investigación.*

*Henry Gonzalo Alvarado Jaya*

## ÍNDICE DE CONTENIDO

CERTIFICACIÓN DEL TUTOR .....	I
AUTORÍA.....	II
DERECHOS DE AUTOR .....	III
APROBACION DE LA COMISIÓN CALIFICADORA .....	IV
DEDICATORIA: .....	V
AGRADECIMIENTO: .....	VI
RESUMEN.....	XII
ABSTRACT .....	XIII
GLOSARIO DE TÉRMINOS Y ACRÓNIMOS.....	XIV
INTRODUCCIÓN .....	XV
CAPÍTULO I.....	1
EL PROBLEMA .....	1
1.1. Tema de Investigación: .....	1
1.2. Planteamiento del problema.....	1
1.2.1. Contextualización.....	1
1.3. Delimitación.....	2
1.3.1. Delimitación de contenido .....	2
1.3.2. Delimitación espacial:.....	2
1.3.3. Delimitación temporal:.....	2
1.4. Justificación.....	3
1.5. Objetivos .....	4
1.5.1. Objetivo General .....	4
1.5.2. Objetivos Específicos.....	4
CAPÍTULO II .....	5
MARCO TEÓRICO.....	5
2.1. Antecedentes Investigativos.....	5



2.2.	Fundamentación Teórica.....	6
2.2.1.	Automatización .....	6
2.2.2.	Pirámide de la Automatización .....	8
2.2.3.	Sistema SCADA .....	10
2.2.4.	Redes de comunicación Industrial .....	12
2.3.	Propuesta de Solución .....	22
CAPÍTULO III .....		23
METODOLOGÍA .....		23
3.1.	Modalidad de Investigación .....	23
3.1.1.	Investigación Bibliográfica .....	23
3.1.2.	Investigación Aplicada.....	23
3.2.	Población y Muestra.....	23
3.3.	Recolección de información.....	23
3.4.	Procesamiento y análisis de datos .....	24
3.5.	Desarrollo del Proyecto.....	24
CAPÍTULO IV .....		26
DESARROLLO DE LA PROPUESTA.....		26
4.1.	Análisis de factibilidad.....	26
4.1.1.	Factibilidad técnica .....	26
4.1.2.	Factibilidad económica .....	26
4.2.	Análisis de requerimientos.....	26
4.2.1.	Software de Simulación .....	26
4.2.2.	Dispositivo de control .....	27
4.2.3.	Lenguaje de programación.....	29
4.2.4.	Sistemas de interfaz Hombre - Máquina.....	30
4.3.	Diseño tridimensional de la línea de producción de vidrio templado.....	31
4.3.1.	Proceso de producción de vidrio templado .....	31
4.3.2.	Simulación del proceso de producción de vidrio templado .....	33
4.4.	Comunicación software de simulación – dispositivo de control.....	42
4.5.	Desarrollo de interfaz Hombre-Máquina .....	46
4.5.1.	Consideraciones en el desarrollo de la interfaz Hombre-Máquina ....	46

4.5.2.	Navegación de HMI .....	53
4.6.	Controlador .....	58
4.7.	Pruebas y resultados .....	63
4.7.1.	Pruebas y resultados en la estación de corte .....	63
4.7.2.	Pruebas y resultados en la estación de pulido .....	66
4.7.3.	Pruebas y resultados en la estación de lavado de vidrio .....	68
4.7.4.	Pruebas y resultados de la estación de templado. ....	72
CAPÍTULO V .....		75
CONCLUSIONES Y RECOMENDACIONES .....		75
5.1.	Conclusiones .....	75
5.2.	Recomendaciones.....	76
BIBLIOGRAFÍA .....		77
ANEXOS .....		80

## ÍNDICE DE TABLAS

Tabla 1.	Trama del protocolo Modbus RTU.....	16
Tabla 2.	Relacion del modelo OSI con el Estandar TCP/IP [9].....	17
Tabla 3.	Elementos de la trama Modbus TCP.....	18
Tabla 4.	Número de dirección de referencia. ....	19
Tabla 5.	Funciones del protocolo MODBUS.....	19
Tabla 6.	Elementos que comprende GEDIS. ....	21
Tabla 7.	Programas de simulación .....	27
Tabla 8.	Características de los dispositivos para adquisición de datos.....	28
Tabla 9.	Características del lenguaje de programación.....	29
Tabla 10.	Características de NETBEANS .....	30
Tabla 11.	Características de la pantalla.....	30
Tabla 12.	Características del vidrio templado y termoendurecido.....	32
Tabla 13.	Comparativa entre hornos de templado. ....	40

Tabla 14. Código de conexión al cliente (Raspberry Pi) .....	43
Tabla 15. Texto empleado en el HMI. ....	46
Tabla 16. Uso del color empleados para el HMI GEDIS [29]. ....	47
Tabla 17. Simbolos y colores de alarmas según su prioridad .....	47
Tabla 18. Comando e ingreso de datos. ....	47
Tabla 19. Relaciones de acceso.....	63
Tabla 20. Direcciones del proceso de Corte.....	64
Tabla 21. Direcciones del proceso de Pulido. ....	67
Tabla 22. Direcciones del proceso de Lavado de vidrio. ....	70
Tabla 23. Direcciones del proceso de Templado. ....	74

## ÍNDICE DE FIGURAS

Figura 1. Automatización como convergencia de tecnologías [11]. ....	8
Figura 2. Pirámide de la Automatización [13]. ....	9
Figura 3. Esquema de un sistema SCADA .....	11
Figura 4. Protocolo MODBUS RTU según el modelo OSI [21]. ....	15
Figura 5. Estructura de la trama Modbus TCP.....	18
Figura 6. Características del protocolo MODBUS .....	20
Figura 7. Diagrama de bloques de los procesos inmersos en la producción de vidrio templado.....	31
Figura 8. Proceso de producción de vidrio templado.....	33
Figura 9. Mesa automática de corte [23]. ....	34
Figura 10. Módulo para el corte de vidrio propio de Factory I/O.....	34
Figura 11. Rectificadora de vidrio CMS BREMBANA FUTURE P [23]. ....	35
Figura 12. Módulo para el pulido de los bordes de las piezas de vidrio. ....	36
Figura 13. Máquina de lavado y secado de vidrio [24]. ....	37
Figura 14. Módulo de lavado de piezas de vidrio proporcionado por Factory I/O. ...	37
Figura 15. Diagrama P&ID del lazo de control de nivel del tanque de agua.....	38
Figura 16. Horno de templado [26]. ....	39
Figura 17. Horno de templado para el simulador.....	39

Figura 18. Diseño de la línea de producción de vidrio templado.....	41
Figura 19. Comunicación entre el simulador, emulador y elemento de monitoreo. .	43
Figura 20. Opciones de Conexión Factory I/O. ....	45
Figura 21. Direcciones de sensores y actuadores presentes en el sistema de producción de vidrio. ....	45
Figura 22. Plantilla de la pantalla de inicio de sesión.....	48
Figura 23. Ventana principal.....	49
Figura 24. Plantilla de la pantalla de procesos.....	49
Figura 25. Vista final de la pantalla de procesos. ....	50
Figura 26. Plantilla de la pantalla de alarmas no acusadas. ....	50
Figura 27. Vista final de pantalla de alarmas.....	51
Figura 28. Plantilla de la pantalla de registro histórico de alarmas. ....	51
Figura 29. Vista final de la pantalla de registro de alarmas.....	52
Figura 30. Plantilla de la pantalla de gestión de usuarios. ....	52
Figura 31. Vista final de la pantalla de gestión de usuarios.....	53
Figura 32. Pantalla de conexión.....	54
Figura 33. Ventana de proceso de corte. ....	54
Figura 34. Ventana del proceso de Pulido. ....	55
Figura 35. Ventana del proceso de lavado. ....	56
Figura 36. Ventana del proceso de templado. ....	57
Figura 37. Ventana de información para administrador.....	58
Figura 38. Diagrama de clases del algoritmo de control.....	59
Figura 39. Relaciones de acceso en el sistema de control.....	62
Figura 40. Diagrama secuencial del proceso de corte.....	65
Figura 41. Diagrama secuencial del proceso de pulido.....	67
Figura 42. Gráfica del comportamiento de la válvula de control.....	68
Figura 43. Sintonización del PID de planta en Matlab. ....	69
Figura 44. Respuesta del nivel del tanque reservorio. ....	70
Figura 45. Diagrama secuencial del proceso de Lavado de vidrio. ....	71
Figura 46. Curva de calentamiento del vidrio templado.....	73
Figura 47. Diagrama secuencial del proceso de Templado.....	73

## RESUMEN

Hoy en día, el monitoreo y control de los procesos juega un papel muy importante en el desarrollo de las industrias, de tal manera que genera beneficios para la productividad, seguridad e innovación, sin embargo, el costo que presentan los elementos empleados en los sistemas de supervisión, control y adquisición de datos (SCADA) son elevados. El presente proyecto de investigación tiene por objetivo desarrollar un sistema SCADA para una planta simulada de producción de vidrio templado, aprovechando de los recursos de la instrumentación virtual y del hardware de bajo costo, está comprendido de 3 etapas como son; diseño tridimensional de la línea de producción de vidrio templado, comunicación software de simulación – dispositivo de control y desarrollo de interfaz Hombre – Máquina.

El diseño de la línea de producción de vidrio templado se consideró los procesos estándares de corte, pulido, lavado y templado para la producción de vidrio, los mismos que fueron replicados en el software de simulación Factory I/O. Software que presenta una serie de beneficios para la implementación de sistemas de producción y control de dispositivos externos en tiempo real.

El dispositivo encargado de representar la función de controlador similar a un controlador lógico programable es la raspberry pi 3, la cual, por sus características de fácil modificación, pueden ser fácilmente adaptadas a la robustez del ambiente industrial. Mediante el empleo de la comunicación MODBUS TCP/IP se envía y recibe información entre el Factory I/O y el raspberry. La interfaz hombre – máquina está desarrollado bajo el lenguaje Java que corre en cualquier sistema operativo. Además, como dispositivo de monitoreo y entrada de comandos se empleó una pantalla táctil compatible con raspberry.

Con el desarrollo e implementación de un SCADA para una planta simulada de producción de vidrio templado bajo hardware libre (emulación de un PLC) se pretende estimular a estudiantes, docentes y personas interesadas en la instrumentación virtual, la generación de sistemas SCADA didácticos asequibles y de bajo costo.

## **ABSTRACT**

Nowadays, monitoring and control of processes play an important role in the development of industries, in such a way that it brings about benefits for productivity, safety and innovation, however, the cost of the elements used in the SCADA systems are high. The aim of this inquiry project is to work up a SCADA system for a simulated tempered glass production plant, taking advantage of the resources of virtual instrumentation and low-cost hardware, comprised of 3 stages: three-dimensional design of the tempered glass production line, simulation software communication - control device and man-machine interface development.

In the design of the tempered glass production line is considered the standard processes of cutting, polishing, washing and tempering for the production of glass, the same ones that are replicated in the Factory I/O simulation software. Software which presents a series of benefits for the implementation of production systems and control of external devices in real time.

The gadget liable for representing the controller function even to a programmable logic controller is the raspberry pi 3, which by using the MODBUS TCP / IP communication can broadcast and receive information between the Factory I / O and the raspberry. The man - machine interface is developed under the Java language, which runs on any operating system. In addition, a touch screen compatible with raspberry is used as a monitoring and command input device.

Through the development and implementation of a SCADA for a simulated production plant of tempered glass under free hardware (emulation of a PLC) is intended to stimulate students, teachers and people interested in virtual instrumentation, the generation of affordable and low cost didactic SCADA systems.

## GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

**SCADA:** Sistema de Supervisión, control y adquisición de datos.

**PLC:** Controlador lógico Programable.

**HMI:** Interfaz Hombre-Máquina.

**Modelo OSI:** Modelo de interconexión de sistemas abiertos.

**CRC:** Verificación de redundancia cíclica

**AS-I:** Bus de Sensores y Actuadores, estándar internacional IEC62026-2 y europeo EN 50295.

**CAN:** *Controller Area Network*

**SDS:** Bus para la integración de sensores y actuadores.

**Profibus:** Estándar de comunicaciones para buses de campo.

**HART:** Protocolo abierto de uso común en los sistemas de control, que se emplea para la configuración remota y supervisión de datos con instrumentos de campo.

**FIP:** *Factory Instrumentation Protocol*

**RTU:** Unidad Terminal Remota

**TCP/IP:** Protocolo de control de transmisión y protocolo de Internet.

## INTRODUCCIÓN

El presente proyecto se encuentra estructurado por cinco capítulos, detallados de la siguiente manera:

En el Capítulo I, se presentan las razones de importancia para el desarrollo de esta investigación partiendo de un análisis macro, meso y micro hasta el planteamiento del justificativo que sustente el desarrollo del mismo y los objetivos a satisfacer.

En el Capítulo II, se puntualiza una breve introducción acerca los sistemas SCADA y la automatización, donde se describe rápidamente parte de su historia y el avance tecnológico que estos han tenido hasta la actualidad. A demás, en este capítulo se plantea y describe de forma general todo lo relevante a los sistemas SCADA, así como también establecer un conocimiento base para el diseño, simulación y control del sistema SCADA a desarrollar.

El Capítulo III, está constituido por la información de la metodología empleada para desarrollar la totalidad del proyecto.

En el Capítulo IV, se describe el proceso llevado a cabo para el diseño, simulación y control del SCADA de la plata simulada de producción de vidrio templado, partiendo desde la descripción del hardware y software empleados en el desarrollo, los métodos utilizados en el diseño y simulación de la línea de producción, y además se redacta sistemáticamente los procedimientos inmersos en la elaboración de la interfaz Hombre-Máquina (HMI) del sistema SCADA.

En el Capítulo V, se cita las conclusiones y recomendaciones que se derivan del desarrollo del proyecto de investigación propuesto. Finalmente, se presentan la lista de referencias consultadas y anexos



# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1. Tema de Investigación:**

“DESARROLLO DE UN SCADA PARA UNA PLANTA SIMULADA DE PRODUCCIÓN DE VIDRIO TEMPLADO”

### **1.2. Planteamiento del problema**

#### **1.2.1. Contextualización**

Existe una vasta variedad de plantas industriales a nivel mundial encargadas de los distintos procesos necesarios para la satisfacción de necesidades del actual estilo de vida, por consiguiente, la automatización de estas plantas constituye un aspecto prioritario [1]. En el paso del tiempo las empresas productoras de vidrio alrededor del mundo han tenido que ir a la vanguardia de la tecnología actualizando su maquinaria, lo cual ha conllevado a la implementación de nuevas tecnologías que permita una capacidad y eficiencia acorde a las exigencias de la industria contemporánea [2]. Sin embargo, el estudio de la automatización de plantas industriales se halla limitado por la dificultad de aplicación en modelos físicos mediante los cuales se pueda probar los sistemas diseñados, por ende, la simulación ha representado una alternativa óptima para la implementación mediante un entorno virtual, con el cual se comprueba el adecuado funcionamiento del sistema [2].

En Ecuador la aplicación de sistemas automatizados es un campo que va en aumento debido a las prestaciones de los mismos, a esta tendencia se suman las empresas dedicadas a la producción de vidrio templado, las cuales enfrentan una competencia constante [2], por esta razón el estudio de sistemas automáticos constituye una

herramienta importante a considerar para su aplicación posterior a este tipo de industrias [4].

La Facultad de Ingeniería en Sistemas, Electrónica e Industrial, busca soluciones a la realidad actual del avance tecnológico de la industria en el país, por tanto, cuenta con equipos necesarios para el desarrollo de sistemas productivos automatizados mediante los cuales se han propuesto e implementado algunos trabajos investigativos, sin embargo, al día de hoy no se ha explorado en la automatización de entornos industriales simulados.

### **1.3. Delimitación**

#### **1.3.1. Delimitación de contenido**

**Área académica:** Electrónica

**Línea de investigación:** Automatización

**Sublínea de investigación:** Sistemas de control automatizados e instrumentación virtual para procesos industriales de baja y alta potencia.

#### **1.3.2. Delimitación espacial:**

El presente proyecto de investigación se desarrollará en las instalaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

#### **1.3.3. Delimitación temporal:**

El presente proyecto de investigación se realizará previo a la aprobación del perfil por parte del Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial hasta la culminación de un periodo académico.

#### 1.4. Justificación

El presente trabajo es de *interés*, ya que explora nuevas alternativas para el estudio y desarrollo de aplicaciones en el campo de la automatización industrial, mediante la cual se puede aprovechar las capacidades brindadas en el mismo para el diseño de sistemas automatizados previo a su implementación en la realidad, o en su defecto de forma didáctica para el aprendizaje del desarrollo de los mismos.

La aplicación de un sistema SCADA sobre una simulación de una planta de producción implica una gran *importancia* en el desarrollo de programas de control automático para la industria, generando nuevo conocimiento relevante para la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, en apoyo al área de la automatización industrial.

El trabajo a realizarse es *novedoso*, puesto que se incursionará en el uso de herramientas relativamente nuevas en el ámbito de la simulación de sistemas automatizados, siendo por consiguiente una nueva manera de realizar aplicaciones de sistemas SCADA bajo un entorno virtual tridimensional.

El aspecto considerado para la realización de un SCADA para una planta simulada, *beneficiará* a estudiantes y docentes de la Carrera de Ingeniería Industrial en Procesos de Automatización y áreas afines, debido a su naturaleza didáctica y versátil que permite diversas aplicaciones en el ámbito de la enseñanza, investigación y desarrollo de temáticas similares.

El desarrollo del SCADA aplicado a un proceso de producción de vidrio templado, causará un *impacto* al ser uno de los pocos ejemplos existentes de proyectos similares, aportando nuevas opciones para el desarrollo de estos sistemas, y en su defecto evitar la realización de prototipos físicos para la automatización de procesos industriales.

El presente proyecto de investigación es *factible*, visto que se cuenta con el conocimiento previo necesario para el desarrollo del mismo, asimismo se dispone de una amplia cantidad de información de referencia, y asequibilidad de recursos.

## **1.5. Objetivos**

### **1.5.1. Objetivo General**

Desarrollar un SCADA para una planta simulada de producción de vidrio templado.

### **1.5.2. Objetivos Específicos**

- Diseñar la distribución tridimensional de la línea de producción de vidrio templado.
- Establecer la instrumentación del proceso simulado.
- Establecer la comunicación entre el proceso simulado y el controlador bajo el protocolo MODBUS TCP/IP.
- Realizar la interfaz hombre-máquina para el monitoreo de la línea de producción.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. Antecedentes Investigativos

Mediante la investigación realizada en el repositorio de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, se determinó que existen trabajos investigativos desarrollados en el año 2014, tales como “Sistema SCADA para el proceso de paletizado L4 de envases de cristal en la Empresa Cristalería del Ecuador S. A. Cridesa de Guayaquil”, el cual se realizó mediante controladores lógicos programables y el software *FactoryTalk View Machine Edition* [5]; “Sistema SCADA para monitoreo y control del aire acondicionado en la sala de recepción satelital (MSA) del Instituto Espacial Ecuatoriano, Estación Cotopaxi”, en el cual se plantea una solución la deficiente condición ambiental para los equipos de la Sala de Recepción Satelital (MSA) a través de un SCADA encargado del funcionamiento óptimo del aire acondicionado [6]. Si bien estos proyectos son aplicaciones de sistemas SCADA, no guardan relación directa con la simulación de plantas industriales, por tanto, en los repositorios de la Universidad del Azuay, Facultad de Ciencia y Tecnología, Escuela de Ingeniería Electrónica, existe un trabajo bajo la temática “Implementación de los protocolos de comunicación industrial mediante Simatic S7-1200 para el laboratorio de automatización de la Universidad del Azuay” en el cual se aborda el uso del software Factory I/O para simulación de una fábrica para la selección de tres tipos de cajas de acuerdo a su tamaño y peso, empleándose como controlador un PLC Siemens S7-1200 [7].

Proyectos de titulación en ingenierías han efectuado aplicaciones en las cuales se realiza la simulación de plantas industriales, entre estos se haya “Desarrollo de Ejercicios Didácticos para Autómatas Programables usando Codesys”, este consta tres ejercicios de automatización de plantas industriales para ejercicios didácticos, desde un Control de Nivel hasta el control de un almacén automatizado [1]. El otro proyecto

se denomina “Diseño y Programación de Varias Estaciones de Trabajo Mediante el Uso de PLC y Softwares Específicos” el que consiste en simular de forma virtual mediante el uso del software Factory I/O y TIA Portal, el comportamiento de varias estaciones de trabajo automatizadas, cuyo control es llevado a cabo mediante la programación de un PLC, ambos proyectos forman parte del repositorio de la Universidad de Valladolid [3].

Además se consideraron investigaciones de maestría en los que se realiza la simulación de una planta industrial, tomándose como referencia el proyecto realizado por Daniel García Fernández de la Universidad de Valladolid “Simulación y Diagnóstico de una Instalación Industrial mediante Factory I/O y OPC” en el cual se ha creado un servidor OPC con la capacidad de acceder a un PLC Siemens S7-315 simulado y conectado con un cliente en Python, lenguaje en que se programó un interfaz gráfica con las librerías de Qt [4]. Además en la Universidad de Oporto se ha realizado la Integración vertical de un ERP en un simulador de fábrica virtual, lo cual permite a los alumnos desarrollar ambientes fabril que comunican con aplicaciones *Manufacturing Operations Management* (MOM) y *Enterprise Resource Planning* (ERP) posibilitando la enseñanza de estas tecnologías en el aula [8].

## **2.2. Fundamentación Teórica**

### **2.2.1. Automatización**

La Real Academia de Ciencias Exactas Físicas y Naturales define la Automática como la ciencia que trata de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la ejecución de una tarea, física o mental, previamente programada.

A partir de la definición, apeándose al ámbito industrial, se puede definir la automatización como el estudio y aplicación de la automática al control de procesos industriales, ya sea en lazo abierto o en lazo cerrado. Refiriéndose a la automatización de la fabricación, se puede englobar tanto la automatización de procesos de fabricación continua como la automatización de procesos de fabricación discontinuos, por lotes o de eventos discretos. Comúnmente se utiliza el término Automática o Regulación

Automática para el tratamiento del control de procesos continuos, mientras que se habla de Automatización para los procesos discontinuos [9].

En otras palabras, la automatización industrial hace referencia a la utilización de sistemas mecánicos, electrónicos o computarizados para el control de procesos y sistemas industriales, de forma que se reduzcan los requerimientos en cuanto a capacidades e intervenciones humanas. El objetivo fundamental de la automatización de un proceso es la incorporación de elementos que controlen el funcionamiento de la instalación, de la máquina o del sistema en general, con la finalidad de reducir los costos de producción, eliminar trabajos laboriosos e inseguros, mejorar el desempeño de la producción bajo demanda, ejecutar tareas imposibles de controlar intelectual o manualmente y simplificar el mantenimiento del proceso productivo integrando gestión y producción [10].

Las técnicas de control de procesos tienen por objetivo conseguir que las variables controladas sigan correctamente las consignas de mando, siendo, en la medida de lo posible, resistentes a las perturbaciones. Este objetivo, siendo de por sí complejo, no es sin embargo el único a cubrir al automatizar en general un proceso. Incluso, en muchos casos, automatizar no supone el ajuste de ningún lazo de regulación.

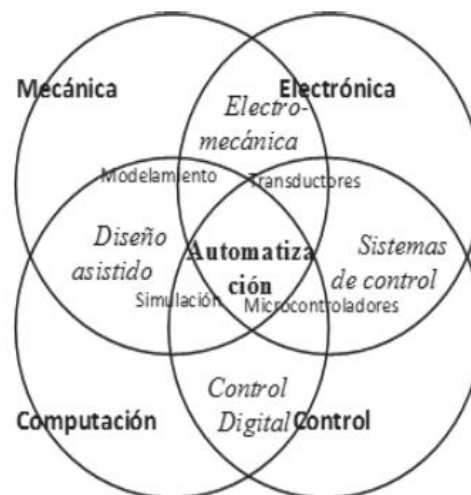
En general, la Automatización implica la utilización y el conocimiento de tres disciplinas: la mecánica, la electrónica y la informática (hardware y software), incluyendo:

- Sistemas sensoriales/instrumentación
- Sistemas de actuación
- Máquinas con control numérico
- Robots Industriales
- Sistemas de almacenamiento y transporte
- Sistemas de regulación
- Sistemas de comunicaciones

En los sistemas de fabricación modernos muchas de las operaciones están automatizadas y/o computarizadas. Sin embargo, los sistemas de fabricación incluyen personas que los hacen funcionar y que deben ser tenidos en cuenta en los procesos de automatización. En algunos sistemas altamente automatizados no hay virtual participación humana en la planta de producción [9].

### 2.2.2. Pirámide de la Automatización

En la actualidad la automatización es un enlace de varias ramas de la ingeniería, formándose de este modo un aspecto multidisciplinario, es decir existe una correlación entre distintas tecnologías que desembocan en una más de las disciplinas de la ingeniería. En la Figura 1 se muestran las diferentes ramas de la ingeniería que convergen en la automatización.

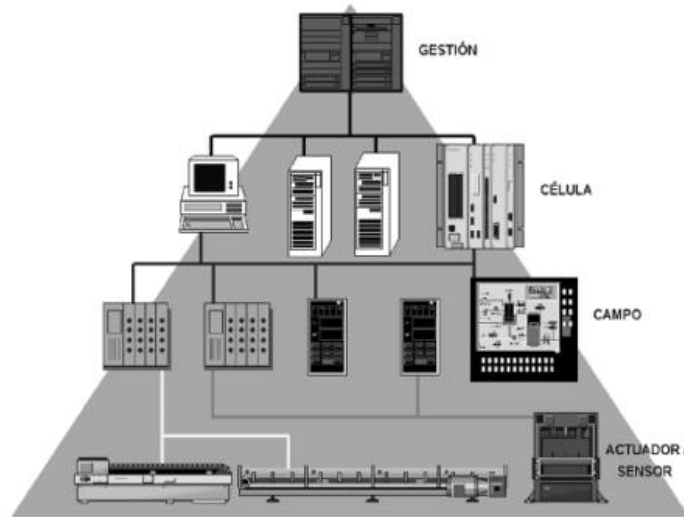


**Figura 1.** Automatización como convergencia de tecnologías [11].

Un aspecto importante de la automatización es que la misma es un proceso conformado por niveles jerárquicos, partiendo de esto se tiene el concepto denominado pirámide de la automatización. En la Figura 2 se puede apreciar la llamada pirámide de la automatización, donde los tres primeros niveles corresponden a la automatización de máquina, en estos se encuentran los elementos de hardware y software que permiten la adquisición, control y supervisión de datos, mientras en las etapas posteriores se tienen los temas de automatización organizacional, planeación, económica y



administrativa, comúnmente se sintetizan estos niveles en un solo cuarto nivel denominado de gestión [11]. Cada uno de estos niveles de la automatización se encuentran relacionados entre sí dentro del mismo nivel y a su vez a los demás niveles mediante las comunicaciones industriales [12].



**Figura 2.** Pirámide de la Automatización [13].

### **A. Nivel de Campo**

En el rango jerárquico de la automatización este es el nivel más bajo también denominado nivel de instrumentación. Está constituido básicamente por unidades de captación de señales, de actuación y de entrada/salida de datos del proceso o de un operador local [14]. En este nivel se efectúa la adquisición de las variables físicas del proceso mediante los sensores situados en él y se ejecuta una acción determinada mediante los actuadores.

Estos sensores y actuadores que interactúan directamente con el proceso productivo se denominan dispositivos de campo. Su principal característica es que trabaja con poca información y su gestión es relativamente sencilla [9].

### **B. Nivel de Control**

En este nivel están los distintos dispositivos de control y monitoreo capaces de gestionar los actuadores y sensores, tales como PLC, HMI, variadores de frecuencia, servo drive, etc. Estos dispositivos son programables y permiten que los actuadores y

sensores funcionen de forma conjunta para ser capaces de realizar el proceso industrial deseado es decir este nivel se encarga de monitorear y controlar todos los dispositivos del campo [5].

### **C. Nivel de Supervisión**

En este nivel es posible visualizar cómo se están llevando a cabo los procesos de planta, se de controlar la interacción entre los distintos dispositivos ubicados en el nivel de control. De esta forma, se puede controlar y monitorear diferentes procesos al mismo tiempo. Incluye PLCs maestros y sistemas SCADA. En este se posee una imagen virtual del proceso donde se ejecutan los mandos disponibles y visualiza posibles fallos o alteraciones del proceso a través de alarmas [5].

### **D. Nivel de Gestión**

Es el punto más alto de la pirámide, y es el encargado de realizar el control de la planta, además en este nivel puede realizar el control y monitoreo de la planta [15]. Procesa tareas de tipo corporativo que implican, generalmente, grandes cantidades de información. En otras palabras, se puede definir el nivel de gestión como la administración a nivel gerencial en el cual se engloba aspectos comerciales, calidad, planificación de la producción, costos, entre otros. En este nivel también se puede vincular nuestra planta con sistemas de control y monitoreo externos [5].

#### **2.2.3. Sistema SCADA**

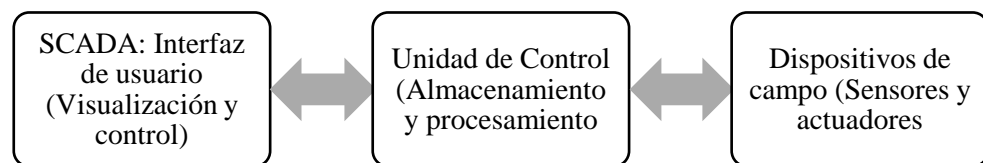
SCADA es un acrónimo de *Supervisor y Control and Data Acquisition* (Supervisión, Control y Adquisición de Datos). Los sistemas SCADA usan tecnologías de comunicación para automatizar el monitoreo y control de procesos industriales. Estos sistemas son partes integrales de la mayoría de los ambientes industriales complejos o muy geográficamente dispersos ya que pueden recoger la información de una gran cantidad de fuentes muy rápidamente y la presentan a un operador en una forma amigable. Los sistemas SCADA mejoran la eficacia del proceso de monitoreo y

control proporcionando la información oportuna para poder tomar decisiones operacionales apropiadas [16].

Los sistemas de control y adquisición de datos son aplicaciones de software especialmente diseñados para funcionar sobre ordenadores, en el control de producción proporcionando comunicación con los dispositivos de campo y controlando de forma automática desde la pantalla de un ordenador. Este también provee de toda la información que se genera en el proceso productivo a diversos usuarios [17].

El sistema permite comunicarse con los dispositivos de campo (controladores autónomos, autómatas programables, sistemas de dosificación, etc.) para controlar el proceso en forma automática desde la pantalla del ordenador, que es configurada por el usuario y puede ser modificada con facilidad. Además, provee a diversos usuarios de toda la información que se genera en el proceso productivo [18].

A continuación, en la Figura 3, se muestra el esquema básico de un SCADA conectado a un proceso automatizado, que se corresponde con los tres primeros niveles de la Pirámide de automatización [19].



**Figura 3.** Esquema de un sistema SCADA

## **A. Prestaciones de un sistema SCADA**

**1. Monitorización:** Representación de datos en tiempo real a los operadores de planta.

2. **Supervisión:** Es un proceso y herramientas de gestión para la toma de decisiones. Tiene además la capacidad de ejecutar programas que puedan supervisar y modificar el control establecido y bajo ciertas condiciones, anular o modificar tareas asociadas a los autómatas
3. **Adquisición de datos:** Por sistema de adquisición de datos se entiende el conjunto de dispositivos, líneas e interfaces que realizan la conexión entre los sensores de medición y un ordenador central que realiza el procesado y almacenamiento de la información [20].

#### 2.2.4. Redes de comunicación Industrial

Las comunicaciones de datos entre diferentes sistemas, procesos e instalaciones suponen uno de los pilares fundamentales para que esta se encuentre en un nivel de competitividad exigida en los procesos productivos actuales. En un sistema de comunicación de datos industrial es tanto más exigente cuanto más cerca del proceso nos encontramos. Cada uno de los niveles de la pirámide de la automatización cuenta con redes de comunicación cuya aplicación se basa según las siguientes características:

- **Volumen de datos:** cantidad de datos que viajan por la red en cada envío.
- **Velocidad de transmisión:** velocidad a la que viajan los datos en la red.
- **Velocidad de respuesta:** velocidad que hay entre el momento de dar la orden y la respuesta del dispositivo [16].

##### A. Bus de campo

Un bus de campo es el medio de transmisión de información en el ámbito industrial, el cual permite la comunicación entre dispositivos que forman parte de un proceso productivo. El objetivo del bus es la interconexión de dispositivos industriales y elementos de control reemplazando al bucle de corriente de 4-20mA [7].

##### B. Clasificación de los buses de campo

###### a) Buses de alta velocidad y baja funcionalidad

Este tipo de buses están orientados a la conexión de elementos sencillos como: fines carreras, relés, actuadores, fotocélulas. Usualmente son empleados en zonas específicas de una planta o máquinas que funcionan en tiempo real, suelen implementar las capas físicas y de enlace del Modelo OSI, entre estos se encuentran:

- CAN: Originalmente para vehículos.
- SDS: Integra sensores y actuadores, basado en CAN.
- AS-I: Diseñado por SIEMENS, bus serie para integrar sensores y actuadores [7].

#### **b) Buses de alta velocidad y funcionalidad media**

En estos buses se encuentra más desarrollada la capa de enlace por lo cual permite el envío de paquetes de datos de tamaño medio, en los cuales es posible incluir aspectos como configuración y calibración del dispositivo. Este bus cuenta con mayor funcionalidad y permite la gestión de elementos de campo complejos, algunos incluyen perfiles de configuración para facilitar la integración con otras marcas y también servicios de configuración desde programas basados en ordenadores, algunos ejemplos son:

- DeviceNet
- BitBus
- DIN MessBuss
- InterBus
- Modbus

#### **c) Buses de alta prestación**

Estos buses están diseñados para trabajar en todos los niveles de automatización, se basan en buses de alta velocidad, la integración tanto de funcionalidad como seguridad

se han convertido en un verdadero desafío para algunos de ellos. La capa de aplicación cuenta con varios servicios como: redes multi-maestro, maestro-esclavo<sup>13</sup>, direccionamiento unicast<sup>14</sup>, multicast<sup>15</sup>, broadcast<sup>16</sup>, comunicación de variables, seguridad en la red, autenticación, peticiones a esclavos, etc. Algunos ejemplos son:

- Profibus
- FIP
- Fiedlbus Foundation
- Modbus

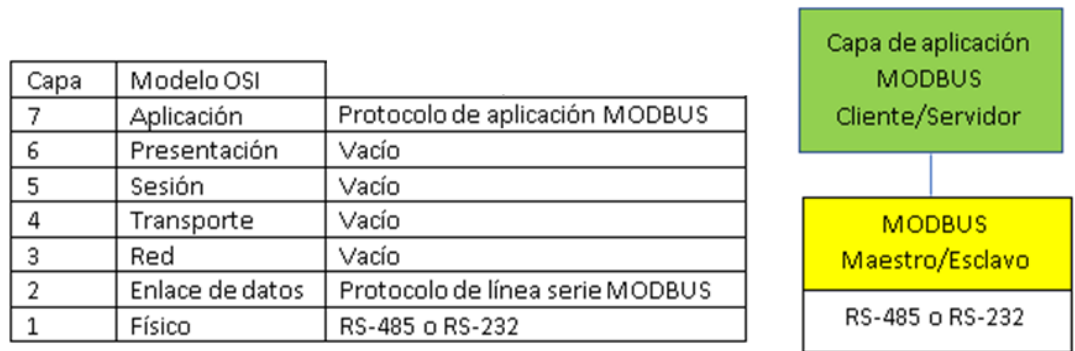
#### **d) Buses para áreas de seguridad intrínseca**

Estos buses presentan ciertas modificaciones en la capa física con el fin de cumplir con las regulaciones de seguridad intrínseca en ambientes explosivos, dicha seguridad hace referencia a la protección necesaria para que el dispositivo no tenga posibilidad de generar una explosión, algunos son:

- HART
- Profibus-PA
- FIP

#### **2.2.5. Normativa del Protocolo Modbus**

El protocolo Modbus es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, comprender mejor cómo está estructurado Modbus, se debe revisar el mencionado modelo OSI. Este modelo fue desarrollado por la Organización Internacional de Normalización y adoptado en 1983 como una referencia para el desarrollo de estándares de comunicación de datos. No intenta definir una implementación real, pero más bien sirve como ayuda estructural para entender "lo que se debe hacer" y "qué va donde", estos parámetros se contemplan en la norma ISO/IEC 7498-1 [8]. La figura 4 muestra la relación del protocolo Modbus RTU según las capas del modelo OSI.



**Figura 4.** Protocolo MODBUS RTU según el modelo OSI [21].

## Modbus RTU de acuerdo a las capas del modelo OSI

### Capa Física

Modbus RTU utiliza conexión de tipo serial para su comunicación, pudiendo implementarse las interfaces RS232 y RS485.

**RS232:** Utilizado en entornos industriales para comunicar un único equipo de campo con una unidad maestro

**RS485:** Utilizado en entornos industriales para comunicar HASTA 32 EQUIPOS de campo con una unidad

### Capa de Enlace

En la capa de enlace, se definen las identificaciones de los equipos determinados por números enteros, cada equipo cuenta con un perfil ya sea maestro o esclavo, también se define la técnica para detectar la integridad de la información enviada en la Trama entre otros aspectos. En Modbus RTU se emplea el CRC el cual es un código de detección de errores empleado para detectar cambios accidentales en los datos

### Capa de Aplicación

En la capa correspondiente a la aplicación, Modbus trabaja bajo el modelo Cliente/Servidor, siendo para el cliente desempeñado por el dispositivo maestro, y para el servidor el dispositivo esclavo.

### **Trama del protocolo Modbus RTU**

Al ser un protocolo de campo destinado al intercambio de información entre un servidor y los dispositivos esclavo, Modbus RTU sigue una estructura de trama bien definida por campos, los cuales se contemplan en la tabla 1.

**Tabla 1.** Trama del protocolo Modbus RTU.

<b>Nombre</b>	<b>Longitud (bits)</b>	<b>Función</b>
Inicio	28	Al menos 3.5 caracteres de silencio (2ms)
Dirección	8	Dirección de la estación
Función	8	Inicia código de función
Datos	n*8	Mensaje
CRC	16	Verificación de redundancia cíclica
Fin	28	Al menos 3.5 caracteres de silencio (2ms)

### **Modbus TCP/IP**

Modbus TCP/IP es la implementación del protocolo Modbus RTU mediante una interfaz TCP que se ejecuta a través de una conexión ethernet.

La estructura de mensajería Modbus es el protocolo de aplicación que define las reglas para organizar e interpretar los datos independientemente del medio de transmisión de datos. TCP / IP se refiere al Protocolo de control de transmisión y al Protocolo de red, que proporciona el medio de transmisión para la mensajería Modbus TCP / IP. Tenga en cuenta que la combinación TCP/IP es simplemente un protocolo de transporte, y no define qué significan los datos o cómo se interpretarán los datos (este es el trabajo del protocolo de la aplicación, Modbus en este caso) [9].

El modelo OSI tradicional junto con el estándar simplificado de TCP / IP, se presenta a continuación en la tabla 2.

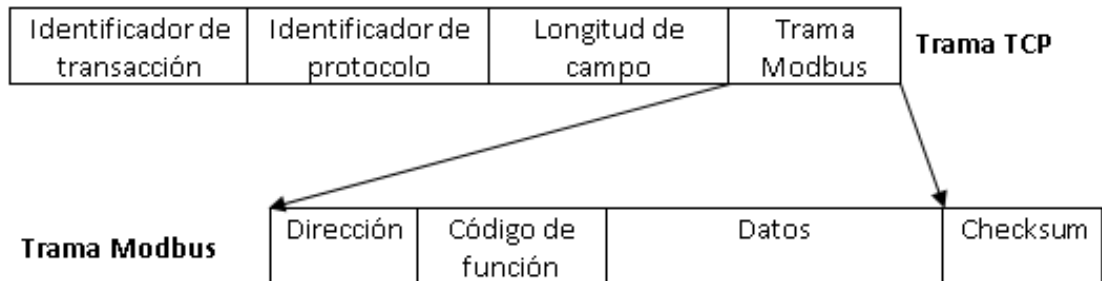


**Tabla 2.** Relacion del modelo OSI con el Estandar TCP/IP [9].

Modelo OSI			Estándar TCP/IP
7	Aplicación	Utilizado por las aplicaciones de software para preparar e interpretar los datos para su uso por las otras seis capas de OSI debajo de él. Proporciona la interfaz de la aplicación a la red.	Capa de aplicación
6	Presentación	Representación de datos, tipo de codificación y define los caracteres utilizados. Realiza negociación y conversión de datos y protocolos para garantizar que los datos se intercambien entre hosts y se puedan transportar a través de la red. También realiza compresión de datos y encriptación.	
5	Sesión	Control de marcación y sincronización de la conexión de sesión. Responsable de establecer y administrar sesiones / conexiones entre aplicaciones y red.	
4	Transporte	La secuenciación de los datos de la aplicación, controla el inicio / final de la transmisión, proporciona la detección de errores, la corrección, la recuperación de extremo a extremo y la eliminación. Proporciona control de flujo de software de datos entre redes. TCP y UDP se encuentran aquí.	Capa de transporte
3	Red	Controla el enrutamiento, la priorización, la configuración, la liberación de conexiones, el control de flujo. Establece /mantiene conexiones a través de una red y proporciona direccionamiento, enrutamiento y entrega de paquetes a los hosts.	Capa de internet, red o internetwork
2	Enlace de datos	Responsable de garantizar la entrega confiable en los niveles más bajos, incluido el marco de datos, detección y corrección de errores, control de secuencia y control de flujo. Ethernet (IEEE 802.2) y MAC se definen en este nivel.	Capa de acceso a la red o Capa de host de red.
1	Físico	Define los atributos eléctricos, mecánicos, funcionales y de procedimiento utilizado para acceder y enviar una secuencia de datos binarios en un medio físico (define el conector RJ-45 y el cable CAT5 de Ethernet)	

### Trama del protocolo Modbus TCP/IP

La trama del protocolo Modbus TCP/IP es igual a la de Modbus RTU, bajo un encapsulamiento en la trama de TCP, tal como lo muestra la figura 5.



**Figura 5.** Estructura de la trama Modbus TCP

Cada uno de los elementos de la trama comprendida en el encapsulamiento de Modbus bajo TCP/IP se detallan a continuación.

**Tabla 3.** Elementos de la trama Modbus TCP.

Nombre	Longitud (Bytes)	Función
Identificador de la transacción	2	Para la sincronización entre mensajes de servidor y cliente
Identificador del protocolo	2	0 para Modbus/TCP
Campo de longitud	2	Número de bytes en esta trama
Dirección	1	Dirección del esclavo (255 si no se usa)
Código de función	1	Códigos de función como en otras variantes
Bytes de datos	n	Datos como respuesta o comandos

Modbus TCP/IP no utiliza el campo de comprobación de errores Modbus original, ya que los métodos de suma de comprobación (Checksum) estándar de la capa de enlace TCP / IP de Ethernet se utilizan para garantizar la integridad de los datos.

## Funciones y Registros

Los registros se organizan en torno a los cuatro tipos de datos básicos mencionados anteriormente y estos datos se identifica además por el número principal de la dirección de referencia como se puede apreciar en la tabla 4.

**Tabla 4.** Número de dirección de referencia.

Referencia	Descripción
0xxxx	Leer / Escribir Salidas o Bobinas Discretas. Se usa una dirección de referencia 0x para conducir los datos de salida a un canal de salida digital.
1xxxx	Leer entradas discretas. El estado ON / OFF de una dirección de referencia 1x es controlado por el canal de entrada digital correspondiente.
3xxxx	Leer registros de entrada. Un registro de referencia 3x contiene un número de 16 bits recibido de una fuente externa, p. Ej. una señal analógica.
4xxxx	Lectura / escritura de salida o registros de mantenimiento. Un registro de 4x se usa para almacenar 16 bits de datos numéricos (binarios o decimales) o para enviar los datos de la CPU a un canal de salida.

En la tabla 5 se muestran las direcciones empleadas en el protocolo MODBUS con su respectivo código. A manera de ejemplo, el comando Leer registros de mantenimiento tiene el código de la función 0000 0011 (03H). Si el dispositivo esclavo toma la acción solicitada sin error, devuelve el mismo código en su respuesta.

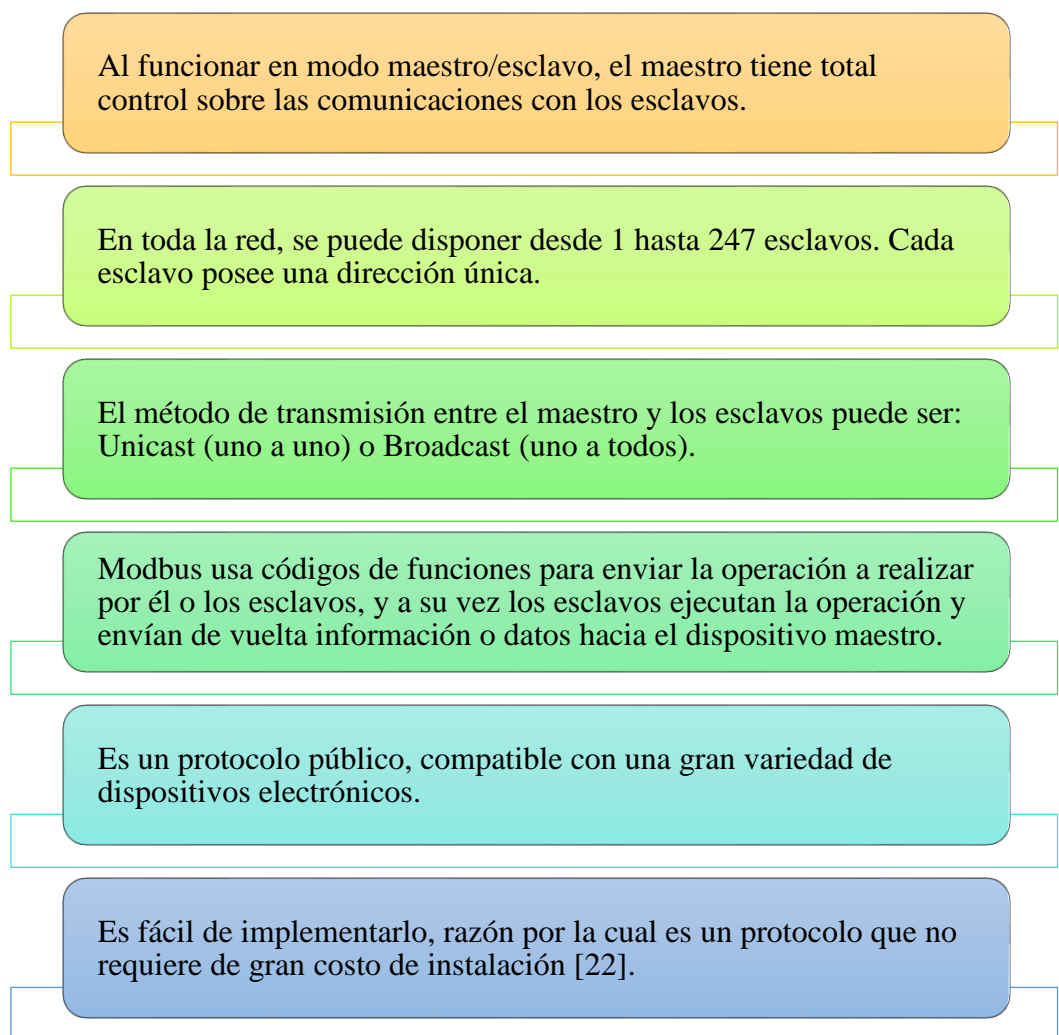
**Tabla 5.** Funciones del protocolo MODBUS.

Código	Función	Referencia
01 (01H)	Leer estado de la salida (Coil)	0xxxx
03 (03H)	Leer registros de mantenimiento (Holding Registers)	4xxxx
04 (04H)	Leer registros de entrada (Input Registers)	3xxxx
05 (05H)	Forzar única salida (Force Single Coil)	0xxxx
06 (06H)	Registro único preestablecido (Preset Single Register)	4xxxx
15 (0FH)	Forzar múltiples salidas (Force Multiple Coils)	0xxxx
16 (10H)	Registro múltiple preestablecido (Preset Multiple Registers)	4xxxx
17 (11H)	Informe de identificación de esclavo (Report Slave ID)	Oculto

### C. Características del protocolo MODBUS

Es el protocolo de comunicación más usado en muchos procesos industriales por sus grandes ventajas que presenta, es considerado un protocolo muy fiable al momento de realizar control sobre diferentes dispositivos.

En la Figura 6 se establece las características y ventajas más relevantes que presenta el uso del protocolo MODBUS.



**Figura 6.** Características del protocolo MODBUS

MODBUS al ser un protocolo implementado por la mayoría de fabricantes en sus equipos, presenta una serie de limitaciones en cuanto al tipo de funciones soportadas que impiden su uso en operaciones de supervisión y control de instalaciones de gestión de energía eléctrica [21].

### 2.2.6. Guía GEDIS

La guía ergonómica de diseño de interfaz de supervisión GEDIS ofrece un método de diseño especializado en sistemas de control supervisor industrial basado en niveles donde se van concretando los diseños de los distintos tipos de pantalla y contenidos. GEDIS ofrece una metodología normalizada en el área del diseño ergonómico de interfaz de supervisión y que comprende los aspectos de diseño de interfaz, ergonomía cognitiva, uso e interacción de la persona con el dispositivo de visualización [27].

La guía GEDIS puede convertirse en complemento para aquellos ingenieros técnicos que desarrollan interfaces de supervisión mediante los sistemas comerciales denominados de adquisición de datos y control supervisor SCADA.

**Tabla 6.** Elementos que comprende GEDIS.

Elementos de la interfaz	Definición
Arquitectura	Organización jerárquica de pantalla
Distribución de pantallas	Plantillas de los diferentes tipos de pantallas
Navegación	Modos de navegación entre pantallas
Uso del color	Asociación del color según la funcionalidad del proceso
Uso de fuentes e información textual	Tipos y tamaños de fuentes de texto
Estatus de los equipos y eventos del proceso	Uso de símbolos e iconos para representar el estado de la planta así como los cambios en la misma
Información y valores de proceso	Presentación de datos analógicos y digitales en la interfaz
Gráficos de tendencia y tablas	Presentación de valores históricos en gráficos y/o tablas
Comandos y entradas de datos	Modo de ingreso de datos en la interfaz
Alarmas	Características principales del sistema de alarmas

### **2.3. Propuesta de Solución**

A través del desarrollo e implementación de un prototipo de sistema SCADA aplicado a una planta de producción de vidrio templado simulada tridimensionalmente a escala minúscula pero con un gran potencial industrial, debido al alcance en el sin número de aplicaciones que este puede tener; se pretende realizar la instrumentación virtual del proceso productivo vinculando el entorno virtual a un controlador real mediante uso de redes de comunicación industrial, acoplándose a situaciones prácticas llevadas a cabo en la automatización industrial, y proponiendo de esta manera la creación de herramientas didácticas de bajo costo para el desarrollo de destrezas de control y monitoreo de procesos bajo estándares y normas de automatización.

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1. Modalidad de Investigación**

Para para el desarrollo de esta investigación se utilizarán los siguientes tipos y procedimientos de investigación

##### **3.1.1. Investigación Bibliográfica**

Mediante la compilación de información previa, la investigación se respaldará en la fundamentación teórica obtenida de medios tales como, libros, artículos académicos, tesis, entre otros documentos pertenecientes a distintos autores que han contribuido en el desarrollo de estudios similares, los cuales proporcionan el conocimiento adecuado para la ejecución del presente trabajo.

##### **3.1.2. Investigación Aplicada**

Es de carácter aplicativo dado que para la misma se ejecutará múltiples pruebas de funcionamiento en las distintas etapas del desarrollo, llevados a cabo de modo continuo hasta su culminación con el resultado esperado.

#### **3.2. Población y Muestra**

La población y muestra no forma parte del presente proyecto, debido a que para el mismo se realizará una investigación aplicada, a partir de información obtenida de distintas fuentes bibliográficas.

#### **3.3. Recolección de información**

Para el presente trabajo la recolección de información se realizará mediante la selección de información relevante, obtenida de fuentes como: páginas web, libros,

tesis y artículos académicos, mismos que aportaran con el conocimiento básico para el desarrollo de la investigación, clarificándose el procedimiento diseño y control de la planta de producción simulada.

### **3.4. Procesamiento y análisis de datos**

El procesamiento de la información será efectuado mediante un análisis crítico, seleccionándose únicamente la información notable, siguiendo de modo sistemático los siguientes pasos:

1. Recolección de información en Internet, informes, documentación de software, artículos científicos, tesis, libros.
2. Análisis y selección de la información compilada a fin de obtener fundamentos de relevantes referentes al tema.
3. Interpretación de resultados en base a observación y diagnóstico basado en aspectos teóricos de referencia.

### **3.5. Desarrollo del Proyecto**

Los aspectos básicos al momento del desarrollo del proyecto se detallan a continuación:

- Analizar la información previa referente a los sistemas SCADA y plantas de producción de vidrio templado, mediante la selección de aspectos relevantes acerca de los temas mencionados.
- Estudiar la documentación del software de simulación a ser documentado (Partes disponibles, funcionamiento, controladores).
- Establecer el tipo de conectividad mediante la cual se establecerá el control y adquisición de datos con el entorno virtual.
- Determinar el controlador físico a ser implementado para el control del proceso de la planta a ser simulada.
- Definir el lenguaje de programación a ser utilizado, así como el entorno de desarrollo, librerías y algoritmos.



- Diseñar la distribución de planta e instrumentación del proceso virtual tridimensional
- Realizar pruebas de ejecución en base a lenguaje de programación y consola, a fin de determinar el correcto funcionamiento de señales de control y lectura de variables.
- Diseñar la interfaz hombre máquina del SCADA, y efectuar las conexiones necesarias entre en entorno virtual, controlador y HMI.
- Analizar los Resultados los resultados obtenidos.
- Elaboración del informe final.

## **CAPÍTULO IV**

### **DESARROLLO DE LA PROPUESTA**

#### **4.1. Análisis de factibilidad**

##### **4.1.1. Factibilidad técnica**

El presente trabajo de investigación desde el enfoque técnico es factible su desarrollo, puesto que el investigador tiene a su disposición los materiales, equipos electrónicos e informáticos necesarios.

##### **4.1.2. Factibilidad económica**

El proyecto desarrollo de un SCADA para una planta simulada de producción de vidrio templado es factible económicamente, debido a que los gastos generados serán cubiertos por el investigador.

#### **4.2. Análisis de requerimientos**

##### **4.2.1. Software de Simulación**

Existen un sin número de programas que permiten la simulación de procesos tanto para el campo industrial como para el de servicios, sin embargo, algunos carecen de varias herramientas necesarias para reproducir un proceso que contemple todos los elementos que lo conforman.

Entre los programas de simulación más comúnmente empleados se puede mencionar:

- Factory I/O
- Easy PLC

- Flexsim

Para seleccionar el software de simulación adecuado para el desarrollo del proyecto se consideró los siguientes requerimientos: costo, factibilidad de conexión con dispositivos externos, librerías y elementos necesarios para la simulación, interfaz intuitiva. Los requerimientos mencionados se detallan en la tabla 7.

**Tabla 7.** Programas de simulación

<b>Software</b>	<b>Costo \$/año</b>	<b>Soporte para redes industriales</b>	<b>Librerías y elementos</b>	<b>Interfaz</b>
Easy PLC	500	Si (hardware adicional)	Básico	Fácil uso
Flexsim	150	No	Básico	Moderado
Factory I/O	312	Si	Intermedio	Fácil uso

El software a utilizarse en el desarrollo del proyecto es Factory I/O, debido a que presenta varias características relevantes frente a las otras alternativas, entre estas tenemos: puede conectarse con dispositivos externos (PLCs, hardware libre, etc.), también presenta una interfaz mucho más interactiva y cuenta con una librería de elementos mucho más amplia para la simulación de procesos productivos. La desventaja que presenta es en el costo debido a que es relativamente mayor a su competencia.

#### **4.2.2. Dispositivo de control**

En el desarrollo de un sistema SCADA, se debe integrar el módulo de adquisición de datos, el cual se lo realiza con dispositivos electrónicos que permitan la lectura y procesamiento de las variables presentes en el proceso, así como también el control de actuadores. En la tabla 8 se visualizan una serie de dispositivos que permiten la adquisición de datos.

**Tabla 8.** Características de los dispositivos para adquisición de datos.

<b>Dispositivo</b>	<b>Características</b>	<b>Costo</b>
Microcontrolador PIC 16fxxx	<ul style="list-style-type: none"> <li>✓ Entradas digitales</li> <li>✓ Entradas analógicas</li> <li>✓ Salidas digitales</li> <li>✓ No puede monitorear en tiempo real.</li> <li>✓ No dispone de conector ethernet.</li> </ul>	\$ 5.00 - \$ 10.00
Arduino Uno	<ul style="list-style-type: none"> <li>✓ Entradas digitales</li> <li>✓ Entradas analógicas</li> <li>✓ Salidas digitales</li> <li>✓ Salidas PWM</li> <li>✓ No dispone de conector ethernet</li> </ul>	\$15.00 - \$30.00
Raspberry Pi 3B	<ul style="list-style-type: none"> <li>✓ Entradas y Salidas Digitales (GPIO)</li> <li>✓ Monitoreo en tiempo real</li> <li>✓ Interfaz Ethernet</li> <li>✓ Conexión para pantalla</li> </ul>	\$ 60.00
PLC Siemens 1200	<ul style="list-style-type: none"> <li>✓ Entradas digitales</li> <li>✓ Entradas analógicas</li> <li>✓ Alta capacidad de procesamiento.</li> <li>✓ Interfaz Ethernet / PROFINET integrado.</li> </ul>	\$ 500.00

La alternativa con mejores prestaciones para la adquisición de datos es un PLC el cual cuenta con módulos que facilitan la lectura de sensores, además de ofrecer una interfaz muy intuitiva de programación, sin embargo, el costo es parcialmente alto, por lo que la segunda alternativa el dispositivo de hardware libre raspberry pi ofrece algunas de las características similares al PLC y su costo es en comparación al PLC relativamente bajo, por lo que se decidió utilizarlo en el proyecto.

Factory I/O ofrece una amplia lista de controladores o drivers que permiten la comunicación con dispositivos electrónicos externos, gracias a ello se puede vincular la placa raspberry Pi modelo 3B.

### 4.2.3. Lenguaje de programación

Existe una gran variedad de lenguajes de programación de alto nivel, por ejemplo: Matlab, Python, C, C++, Java, empleados en el desarrollo de aplicaciones para la industria en servidores, redes y puntos de control. El lenguaje de programación empleado para el desarrollo de los algoritmos de control para el sistema planteado es Java bajo el entorno de desarrollo NETBEANS debido a las características mostradas en la siguiente tabla 9.

**Tabla 9.** Características del lenguaje de programación.

Lenguaje de Programación	Características
Java	<ul style="list-style-type: none"><li>✓ Multiplataforma</li><li>✓ Entornos de desarrollo intuitivos y de fácil aprendizaje.</li><li>✓ Soporte para raspberry Pi</li><li>✓ Gran cantidad de información disponible para su uso</li><li>✓ Facilidad en la instalación de librerías, paquetes, etc.</li><li>✓ Creación de pantallas gráficas (animación)</li></ul>
Python	<ul style="list-style-type: none"><li>✓ Multiplataforma</li><li>✓ Intuitivo y de fácil aprendizaje</li><li>✓ Soporte para trabajar con tarjetas embebidas Raspberry Pi.</li><li>✓ Soporte por la comunidad en desarrollo de aplicaciones</li></ul>
C++	<ul style="list-style-type: none"><li>✓ Multiplataforma</li><li>✓ Entorno de desarrollo medio</li><li>✓ No presenta soporte para raspberry</li><li>✓ Complejidad en la instalación de librerías.</li></ul>

NETBEANS es un entorno de desarrollo estándar, escrito en el lenguaje de programación Java, en la tabla 10 se muestran las características del IDE NETBEANS.

**Tabla 10.** Características de NETBEANS

Entorno de desarrollo	Características
NETBEANS	<ul style="list-style-type: none"> <li>✓ Lenguaje simple estructurado en c y c++</li> <li>✓ Está orientado a la programación en objetos.</li> <li>✓ Esta distribuido (permite abrir sockets)</li> <li>✓ Es altamente confiable (robusto) en comparación con C.</li> </ul>

#### 4.2.4. Sistemas de interfaz Hombre - Máquina

El modo de supervisión en el desarrollo del sistema SCADA está formado por una interfaz hombre-máquina (HMI) que conecta una pantalla al dispositivo de control para este caso la placa raspberry Pi 3 modelo B, el panel táctil es una característica deseada ya que facilita la interacción debido a que existen pantallas que no presenta dicha característica. Las características de las pantallas se describen en la tabla 11.

**Tabla 11.** Características de la pantalla.

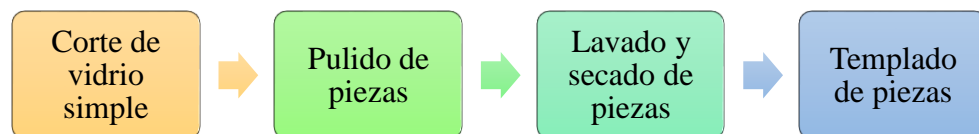
Dispositivo	Características
Pantalla Táctil HMI 7"	<ul style="list-style-type: none"> <li>✓ Resolución 800x420</li> <li>✓ Panel táctil capacitivo de vidrio templado</li> <li>✓ Compatible con mini ordenador populares como Raspberry Pi, BB Black, Banana Pi</li> </ul>
Pantalla Táctil HMI 5"	<ul style="list-style-type: none"> <li>✓ Compatible con Raspberry Pi/2 A +, B + y Arduino</li> <li>✓ Resolución 800 x 480</li> <li>✓ Panel táctil resistivo</li> <li>✓ Consumo de energía 5V410mA</li> </ul>
Pantalla Táctil HMI 3"	<ul style="list-style-type: none"> <li>✓ Compatible únicamente con Raspberry Pi 3 2 Modelo B/A/B+/A+</li> <li>✓ Resolución de 480x320</li> <li>✓ Uso de energía móvil</li> <li>✓ Tipo de pantalla táctil: Resistivo</li> </ul>

Considerando algunos aspectos relevantes para el monitoreo de procesos industriales como ejemplo el diseño de interfaces HMI, la pantalla táctil seleccionada para el presente proyecto es aquella que cuenta con un tamaño de 7" y una resolución de 800x420, además por el soporte que ofrece para un sin número de placas mini ordenador.

### 4.3. Diseño tridimensional de la línea de producción de vidrio templado.

#### 4.3.1. Proceso de producción de vidrio templado

El vidrio simple es sometido a un tratamiento térmico que radica en elevar su temperatura hasta cerca del punto de ablandamiento (650° C) y enfriarlo rápidamente. Esto hace que la superficie del vidrio se enfríe más rápido que el centro del mismo, que estará relativamente más caliente. Al enfriarse la superficie, se contrae mientras que el centro no lo hace de la misma manera. Cuando el centro se va enfriando, fuerza a la superficie y a los bordes a la compresión [22]. Los procesos inmersos en la producción de vidrio templado se muestran en el diagrama de bloques de la figura 7.



**Figura 7.** Diagrama de bloques de los procesos inmersos en la producción de vidrio templado.

- A. Corte de vidrio simple:** Proceso en el cual se toma las características del cliente como: tamaño, forma y espesor del vidrio para cortarlo en piezas.
- B. Pulido de piezas cortadas:** Las piezas previamente cortadas presentan aristas vivas que deben ser reducidas, para ello el proceso de pulido desbasta los lados de las piezas cortadas.
- C. Lavado y secado de piezas:** Proceso de limpieza de los contaminantes, como el polvo, presentes en las superficies de las piezas de vidrio a templar.
- D. Templado de Piezas:** Proceso que eleva la temperatura hasta el punto de ablandamiento y luego es enfriado, logrando así dotar a las piezas de vidrio mayor

resistencia. En la Tabla 12 se muestra la comparativa entre un vidrio templado y termoendurecido.

**Tabla 12.** Características del vidrio templado y termoendurecido.

<b>Característica</b>	<b>Templado</b>	<b>Termoendurecido</b>
Resistencia al impacto respecto al vidrio crudo.	4 veces más	2 veces más
Flexión respecto al vidrio crudo.	Igual	Igual
Forma de rotura	Pequeños fragmentos sin aristas cortantes	Pedazos grandes sin aristas cortantes
Soporta cambios de temperatura se hasta	250°C	120°C
Presenta rotura espontanea	Si	No
Se considera vidrio de seguridad	Si	No

El vidrio templado o vidrio reforzado es utilizado en la industria automovilística en la colocación de laterales y luneta en un automotor, también es empleado en puertas de chimenea, cobertores de ducha, entre otros, debido a su resistencia y al ser un vidrio que se desmenuza en tozos pequeños granulares a diferencia de los vidrios normales que se dividen en astillas con aristas vivas, gracias a esta propiedad los vidrios templados tienen menor probabilidad de generar lesiones [22]. La figura 8 muestra el diagrama de procesos de producción de vidrio templado.





**Figura 8.** Proceso de producción de vidrio templado.

### 4.3.2. Simulación del proceso de producción de vidrio templado

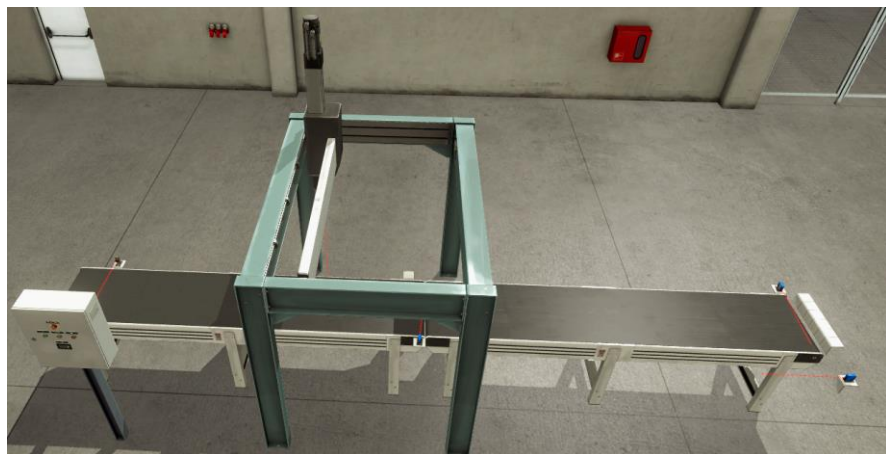
#### A. Corte de planchas de vidrio simple

La entrada al proceso de corte consiste en planchas de vidrio simple, que de acuerdo a los requerimientos del cliente (espesor, formas, etc.) son cortados en piezas. Para ello se emplea una máquina cortadora automática como la presentada en la figura 9.



**Figura 9.** Mesa automática de corte [23].

La mesa automática de corte no existe en la librería del programa de simulación Factory I/O, por lo tanto, se desarrolló un módulo que permitan realizar la misma función que realiza la mesa de corte. Este módulo está formado por un emisor de material que reemplaza el colocador de vidrio sobre la mesa, cuenta con un sensor difuso para la detección de presencia de la plancha de vidrio, también posee 2 bandas transportadoras para el avance del vidrio. El plotter es reemplazado por un sistema pick and place, el mismo que cuenta con un sensor detector de posición para el corte, como salida se obtiene las piezas de vidrio que pasan a través de un sensor detector de presencia activando el dispositivo de removido de piezas, además el módulo de corte cuenta con un panel de control (pulsadores de marcha, paro de emergencia y reajuste), en la figura 10 se puede apreciar el módulo de corte, los planos referenciales de la estación de corte se encuentran en el Anexo B.



**Figura 10.** Módulo para el corte de vidrio propio de Factory I/O.

## B. Pulido y lijado de piezas cortadas

Una vez procesado las planchas de vidrio en piezas que cuentan con aristas con vivas y no son adecuadas para la manipulación por parte del cliente, debido a que su filo puede causar corte, por lo tanto, el proceso de pulido y lijado es el encargado del tratamiento de los bordes de las piezas de vidrio cortadas. La máquina encargada de realizar el proceso es una rectificadora de plancha de vidrio CMS BREMBANA FUTURE P, la misma que se puede observar en la figura 11.



**Figura 11.** Rectificadora de vidrio CMS BREMBANA FUTURE P [23].

Es necesario la creación del módulo de pulido. El módulo de pulido está formado por 2 bandas transportadoras en configuración L por medio de un transferidor que pasa de una banda a otra, cada banda posee en su parte lateral unas cintas que simulan ser los rodillos abrasivos. Posee varios sensores para detectar la presencia del material a procesar tanto en la salida como entrada de la máquina. Los medios de control que dispone el módulo son: un panel con los pulsadores de marcha, paro de emergencia y reajuste. Una vez procesados los bordes de las piezas de vidrio estos pasan por sensor que activa un transportador para moverlas al siguiente proceso. El módulo de pulido desarrollado en el simulador se observa en la figura 12. mientras que los planos de la estación se presentan en el Anexo C.



**Figura 12.** Módulo para el pulido de los bordes de las piezas de vidrio.

### **C. Lavado de piezas pulidas**

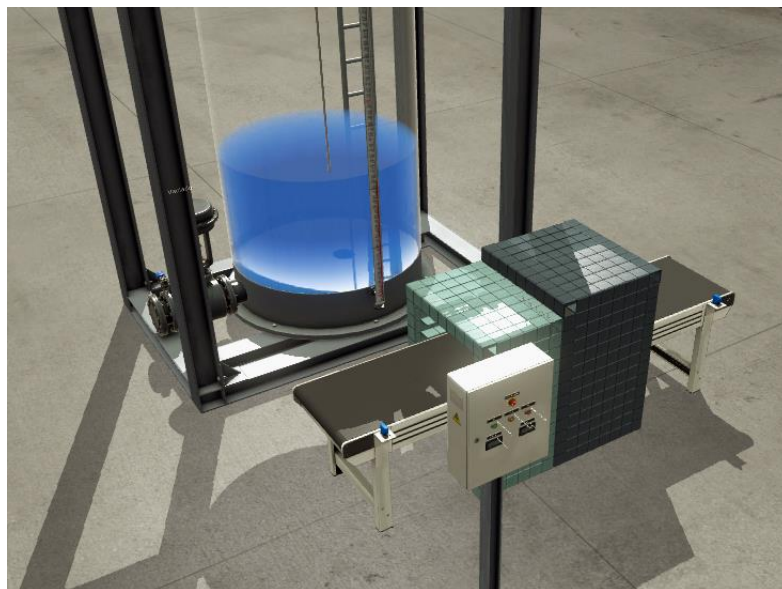
Los desechos generados en el proceso de pulido se esparcen sobre las piezas de vidrio contaminándolas, por lo cual es necesario realizar la limpieza de estos. El proceso encargado de limpiar las superficies y bordes del vidrio en el proceso de lavado y secado, el mismo que toma las piezas y mediante un sistema de rodillos se mueven mientras se rocía agua a presión, después de pasan por la sección de secado por aire comprimido. La Figura 13 muestra una máquina de lavado y secado de vidrio utilizada en este proceso.





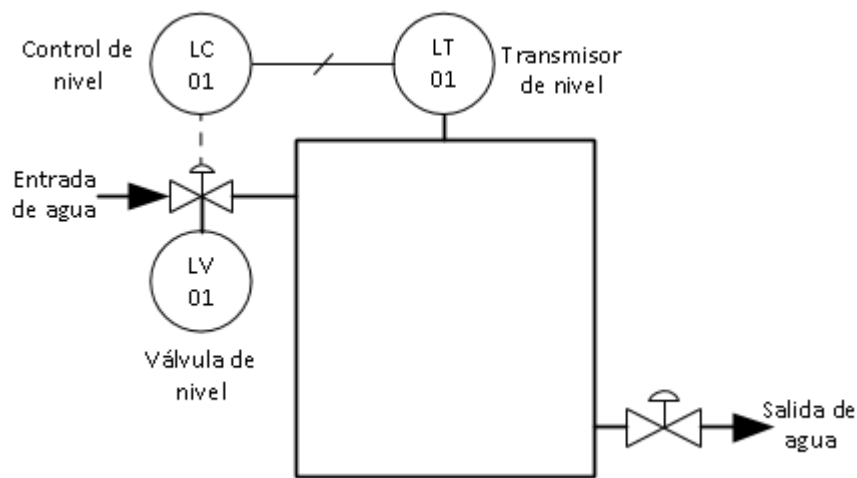
**Figura 13.** Máquina de lavado y secado de vidrio [24].

El módulo de lavado desarrollado en el programa de simulación que replique los procedimientos que realiza la máquina presentada en la figura 13 está formado por una banda transportadora con un sensor detector de presencia, la cual mueve las piezas de vidrio colocadas por el transportador hace la cámara de rociado de agua a presión y secado, además cuenta con un tanque reservorio de agua con 2 válvulas, estas son utilizadas para el llenado y vaciado del tanque. La válvula de vaciado es de activación manual mientras que la del llenado corresponde a un controlador PID. En la figura 14 se observa el módulo de lavado en el simulador. Al igual que los módulos anteriores cuenta con el panel de control con los pulsadores de arranque, paro de emergencia y reajuste (planos – Anexo D).



**Figura 14.** Módulo de lavado de piezas de vidrio proporcionado por Factory I/O.

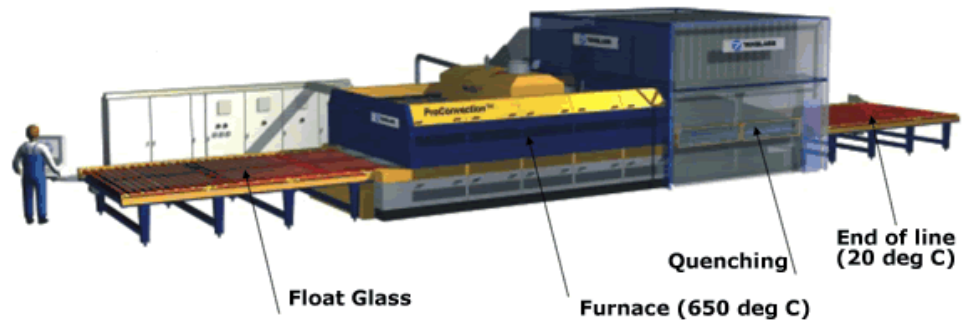
Dado que se ha considerado la implementación de un controlador PID para el nivel de agua se ha determinado el lazo de control según indica el diagrama P&ID según la figura 15, en donde se aprecia que el transmisor de nivel envía una señal con el valor actual de la variable al controlador, en este se realiza las acciones correctivas necesarias, posteriormente el controlador envía la señal de control respectiva a la válvula determinando la cantidad de agua que ingresa al tanque, estabilizándose de esta manera el nivel en un valor deseado.



**Figura 15.** Diagrama P&ID del lazo de control de nivel del tanque de agua.

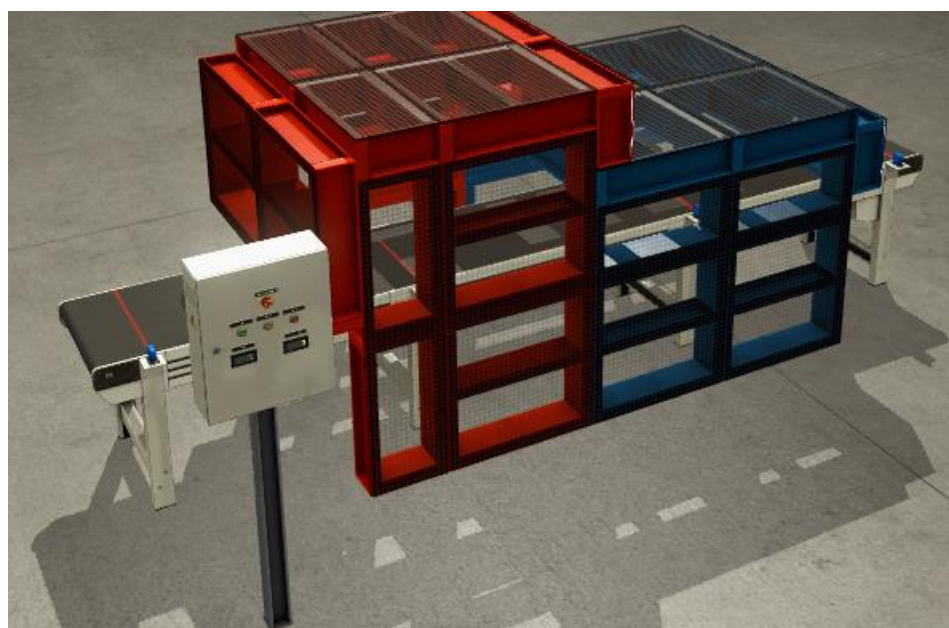
#### **D. Templado de piezas**

El proceso final en la línea de producción de vidrio templado es la etapa donde se realiza en templado de las piezas de vidrio mediante un tratamiento térmico. El proceso empieza con recibir las piezas secas de la máquina de lavado y secado y enviarlas a un horno donde permanecerán por un tiempo de templado y luego saldrán hacia la etapa de enfriado para finalmente ser empacados y enviados a sus respectivos clientes. El tiempo de templado varía de acuerdo al espesor del vidrio como menciona en [25]. En la figura 16 se puede visualizar la máquina que realiza este proceso.



**Figura 16.** Horno de templado [26].

Factory I/O cuenta con una librería que proporciona varios elementos como sensores, actuadores y máquinas. En este caso la librería carece de un horno, por lo que se decidió utilizar partes semejantes a uno para que simulen el módulo de templado de vidrio, además se incluyeron algunos sensores de posición para la etapa de templado y enfriado, finalmente se agregó el panel de control que tienen todos los módulos que conforman la línea de producción de vidrio en el simulador. En la figura 17 se puede observar el horno de templado (planos – Anexo E).



**Figura 17.** Horno de templado para el simulador.

Existen varias alternativas acopladas al funcionamiento de un horno industrial, el cual consiste en el recurso que energice dicha máquina, para ello se emplea una Tabla 13 comparativa donde se aprecian los diferentes recursos aplicados como fuente de

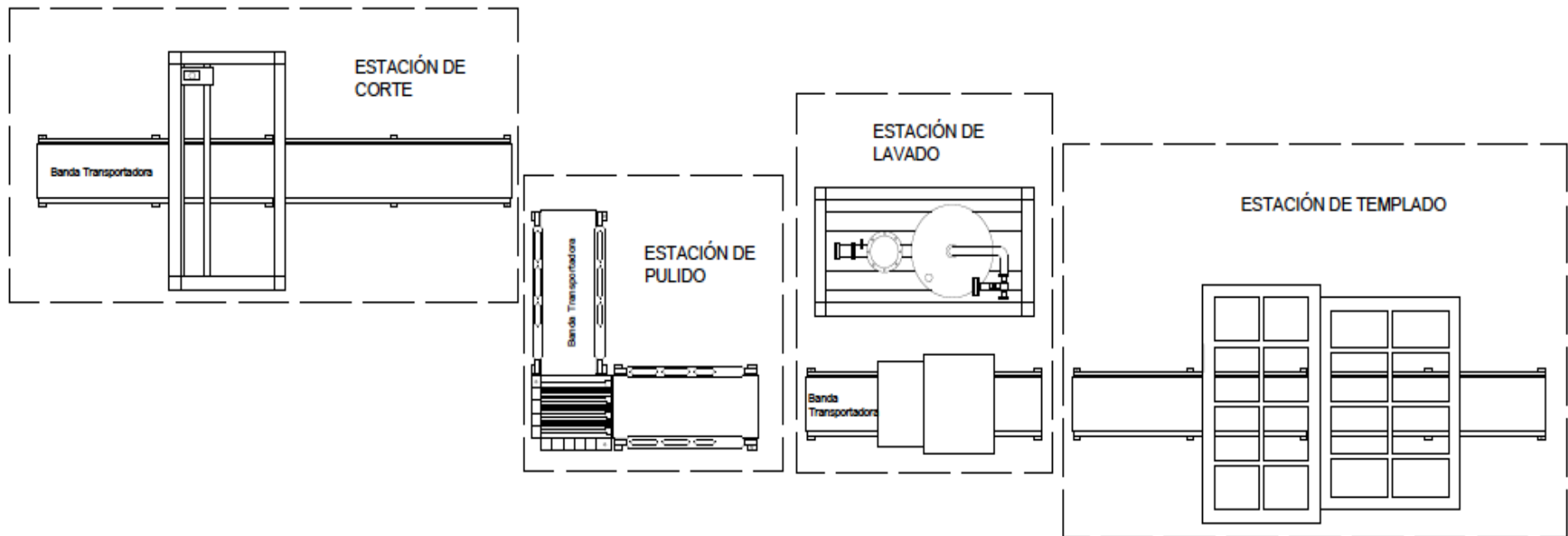
energía para la estación de templado. La alternativa empleada en el proyecto es de un horno de templado eléctrico simple.

**Tabla 13.** Comparativa entre hornos de templado.

<b>Tipo de horno de templado</b>	<b>Características</b>	<b>Consideraciones de aplicación en la simulación</b>
Horno de templado eléctrico con sistema de control avanzado	<ul style="list-style-type: none"> <li>✓ Controlador avanzado con Interfaz Hombre-Máquina.</li> <li>✓ Controlador PID</li> </ul>	<ul style="list-style-type: none"> <li>✓ Es necesario el modelamiento del sistema.</li> <li>✓ Manejo de variables analógicas [27].</li> </ul>
Horno de templado eléctrico con termostato con sistema de control simple (interno)	<ul style="list-style-type: none"> <li>✓ Controlador basado en autómata programable., con sistema de control interno para el mantenimiento uniforme de la temperatura</li> </ul>	<ul style="list-style-type: none"> <li>✓ Manejo de señales básicas de control (Marcha, Paro).</li> </ul>
Horno de templado a gas	<ul style="list-style-type: none"> <li>✓ Control basado en dispositivos de seguridad (termostatos)</li> </ul>	<ul style="list-style-type: none"> <li>✓ Manejo de señales básicas de control (Marcha, Paro).</li> <li>✓ Dependencia del nivel de combustible [28].</li> </ul>

Todos los módulos desarrollados en el programa Factory I/O trabajan en secuencia, esto quiere decir que mediante un sensor de presencia que activa el dispositivo removedor, el cual tiene por objetivo mover las piezas de un módulo a otro para su siguiente procesamiento. Además, calcula el tamaño de lote a producir para que todas las máquinas realicen el mismo número de unidades procesada por la etapa inicial de corte. En la Figura 18 se puede observar el diseño de la línea de producción de vidrio y en el Anexo F se muestra la réplica de la línea desarrollada en el programa de simulación Factory I/O.





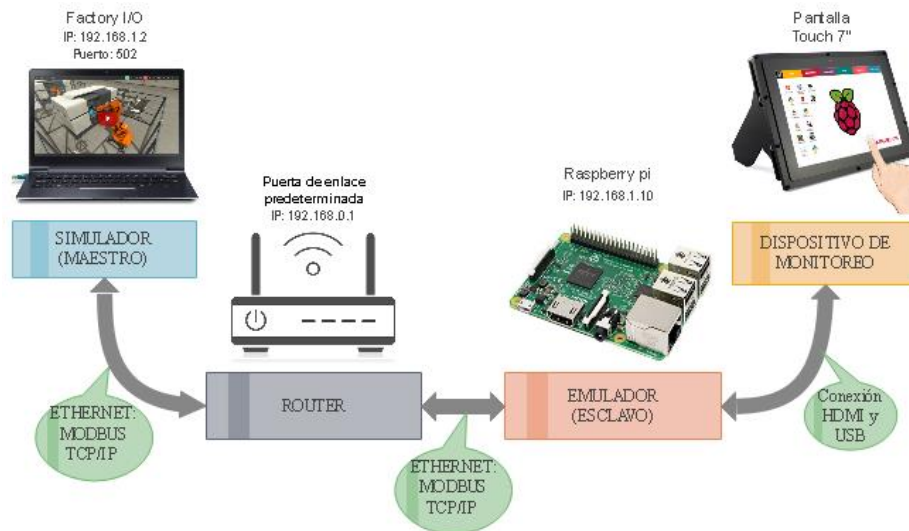
**Figura 18.** Diseño de la línea de producción de vidrio templado.

La simulación de cualquier sistema de producción permite mejorar el diseño y desempeño del mismo, haciéndolo más eficiente y eficaz, debido a que se determinan las causas que dificultan la ejecución del sistema como pueden ser: demoras en un proceso, máquinas con tiempo improductivo, paradas indeseadas, entre otros.

Para el caso de estudio la simulación del proceso de producción de vidrio templado presenta la ventaja de conocer las posibles causas de fallos en la línea antes de ser implementada, ya que se pueden evitar problemas y errores frecuentes en la implementación de un proceso. La simulación frente al sistema real de producción de vidrio templado tiene el beneficio costo debido a que disminuye los costos asociados a la experimentación del sistema real. Además, la simulación constituye una herramienta básica en el proceso de aprendizaje práctico, ya que es accesible para la aplicación de casos prácticos apegados a la realidad.

#### **4.4. Comunicación software de simulación – dispositivo de control**

Una vez finalizada la etapa de creación del escenario o réplica del proceso de producción de vidrio templado con sus respectivas secciones, máquinas y sensores presentes en cada estación se procede a realizar la comunicación entre el simulador, dispositivo de control y elementos de monitoreo, para ello se empleó como emulador de dispositivo de control una Raspberry Pi 3, la cual en configuración de cliente se comunica mediante una conexión ethernet bajo el protocolo Modbus TCP/IP al programa de simulación Factory I/O en configuración de servidor mediante un router empleado como puerta de enlace predeterminada, la dirección IP con la cual se comunica corresponde a la dirección que proporciona el software Factory I/O utilizando la tarjeta de red ethernet del ordenador, al mismo tiempo se asigna el número de puerto de la conexión TCP/IP.



**Figura 19.** Comunicación entre el simulador, emulador y elemento de monitoreo.

En la figura 19 se puede observar cómo está establecida la comunicación implementando el protocolo MODBUS TCP/IP, donde se presenta al programa de simulación Factory I/O como el servidor conectado de forma física a una puerta de enlace predeterminada (ROUTER), y mediante el uso de la librería easyMODBUS – TCP/IP se conecta la placa raspberry Pi como cliente a la puerta de enlace de forma inalámbrica, el medio físico para esta conexión es la conexión ethernet a través de cable UTP y conectores RJ45, posteriormente se emplea como medio de monitoreo una pantalla táctil en comunicación con la raspberry pi mediante conexión HDMI para la visualización de imagen y USB para la capacidad táctil. En la tabla 14 se observa el código empleado en la comunicación entre el servidor y el cliente.

**Tabla 14.** Código de conexión al cliente (Raspberry Pi)

<pre>import de.re.easymodbus.modbusclient.*; ModbusClient Cliente = new ModbusClient("127.0.0.1",502);</pre>
<p>Donde:</p> <ul style="list-style-type: none"> <li>✓ Cliente es el nombre del objeto que se ha creado.</li> <li>✓ "127.0.0.1", representa la dirección IP del servidor MODBUS TCP (Dada por el simulador y es un dato de tipo String.</li> <li>✓ 502, es el número de puerto de comunicación dada por el simulador (Servidor)</li> </ul>

#### 4.4.1 Librería EasyMODBUS

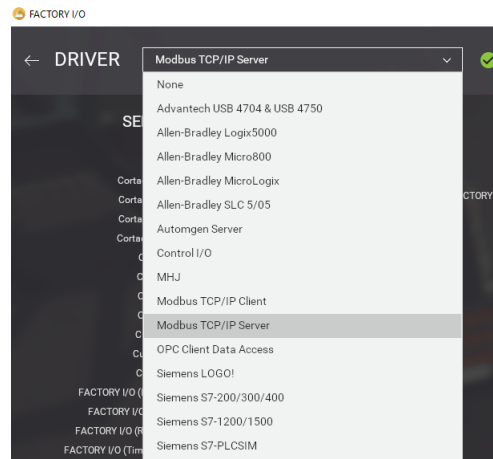
La aplicación del protocolo modbus utiliza la biblioteca externa EasyModbus. La biblioteca está desarrollada por la compañía alemana Ross Rossmann Engineering Solutions con licencia pública abierta. Es compatible con .NET y JAVA. Y Python. Incluye métodos y componentes para la comunicación entre Modbus RTU y Modbus TCP. También incluye herramientas de desarrollo de software como Modbus Server Simulator. La biblioteca proporciona acceso rápido y seguro desde PC, sistemas embebidos, PLC u otros componentes de automatización industrial. Solo se necesitan unas pocas líneas de código para leer o escribir datos desde o hacia el sistema. Las funciones que presenta la librería se describen a continuación:

- Lectura de bobinas (coils)
- Leer entradas discretas (discrete inputs)
- Lectura de registros (Holding registers)
- Lectura de registros de entrada (input registers)
- Escritura de una sola bobina
- Escritura de un solo registro
- Escritura de multiples bobinas
- Escritura de multiples registros
- Leer/escribir multiples registros

La característica seleccionada de la librería easyMODBUS es MODBUS TCP/IP Cliente, debido a la capacidad y compatibilidad del simulador (Factory I/O) y de la placa raspberry Pi. El modo de conexión realizado designa al programa simulador Factory I/O como el servidor (SERVER), mientras que el raspberry es el cliente (CLIENT), puesto que cuenta con los siguientes parámetros:

- Permite escribir y leer señales digitales analógicas (Requerido para programar el proceso)
- El modo servidor solo permite visualizar notificaciones cuando existen cambios en entradas o salidas, o en la conexión de clientes

En el Anexo A se presenta todos los métodos disponibles para configuración del servidor que posee la librería easyMODBUS. La figura 20 muestra las opciones para la comunicación con un dispositivo externo empleando los drivers que vienen incluidos en el software Factory I/O.



**Figura 20.** Opciones de Conexión Factory I/O.

(192.168.1.6:502) Slave ID:1			
Input 0	Coil 0	Input 0	Cut_ent
Cut_start	Coil 1	Input 1	Cut_sal
Cut_Stop	Coil 2	Input 2	Pul_ent
Cut_emg	Coil 3	Input 3	Pul_B1
Cut_reset	Coil 4	Input 4	Pul_B2
Cut_s_ent	Coil 5	Input 5	Rodillos1-1
Cut_s_pos	Coil 6	Input 6	Rodillos1-2
Cut_s_sal	Coil 7	Input 7	Pul_Gir_Avance
Pul_start	Coil 8	Input 8	Pul_Gir
Pul_stop	Coil 9	Input 9	Rodillos2-1
Pul_emg	Coil 10	Input 10	Rodillos2-2
Pul_reset	Coil 11	Input 11	Pul_sal
Pul_s_ent	Coil 12	Input 12	Lav_ent
Pul_s_1	Coil 13	Input 13	Lav_sal
Pul_s_2	Coil 14	Input 14	Hor_ent
Pul_s_sal	Coil 15	Input 15	Hor_sal
Lav_start	Holding Reg 0	Input 16	Cut_B1
Lav_stop	Holding Reg 1	Input 17	Cut_B2
Lav_emg	Holding Reg 2	Input 18	CortadoraX
Lav_reset	Holding Reg 3	Input 19	CortadoraY
Lav_s_ent	Holding Reg 4	Input 20	CortadoraZ
Hor_stop	Holding Reg 7	Input 23	Vaciado
Hor_emg	Holding Reg 8	Input 24	Niv_setpoint_dis
Hor_reset	Holding Reg 9	Input 25	Niv_dis

**Figura 21.** Direcciones de sensores y actuadores presentes en el sistema de producción de vidrio.

En la figura 21 se muestra el listado de todas las entradas y salidas empleadas en la réplica del proceso de producción de vidrio diseñado en el programa Factory I/O.

#### **4.5. Desarrollo de interfaz Hombre-Máquina**

Las pantallas de visualización desarrolladas para el monitoreo del proceso de producción de vidrio templado están codificadas bajo el uso de las herramientas de NETBEANS – JAVA en la creación de interfaces gráficas.

##### **4.5.1. Consideraciones en el desarrollo de la interfaz Hombre-Máquina**

El desarrollo de interfaces gráficas para la interacción hombre-máquina es una tarea que se basa en estándares para su consolidación, debido a esto, se empleó la guía GEDIS como instructivo para la creación de las pantallas, el cual se basa en la norma ISA 101 [29], y se derivan los siguientes códigos de colores para cada parámetro perteneciente al HMI (fondo, equipos, alarmas, etc.) En el Anexo G se muestran las tablas con la codificación de colores tomados de la guía GEDIS.









El texto utilizado en la aplicación de la interfaz se define según se muestra en la tabla 15.

**Tabla 15.** Texto empleado en el HMI.

Fuente	Ms Sans Serif
Tamaño en títulos	24
Tamaño en general	12




La guía GEDIS contiene un listado estandarizado de colores para ser empleados de manera óptima en la pantalla de visualización, para este caso se han implementado los colores listados según indica la tabla 16.

**Tabla 16.** Uso del color empleados para el HMI GEDIS [29].

Colores de fondo				
Item	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Sinópticos de Área y Subárea		Arena	208/204/191	31/37/188
Detalle Máquinas		Verde Oscuro	120/80/90	64/128/128
Menus y Analógicas		Gris Plata	160/0/224	238/238/238
Tablas en Sinópticos		Gris Plata	160/0/224	238/238/238
Estatus de Equipos de Proceso				
Equipo Parado		Blanco	160/0/240	255/255/255
Equipo Trabajando		Verde	80/240/53	0/113/0
Materiales de proceso				
Agua		Verde	80/240/53	0/113/0
Señales analógicas				
Temperatura		Marrón	0/240/46	98/0/0


Para la visualización de alarmas se empleo el uso de color recomendado por la guía GEDIS, así como el uso de simbología recomendado por la norma ISA 101 [30].






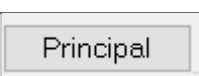
**Tabla 17.** Simbolos y colores de alarmas según su prioridad

Prioridad	Símbolo
Crítica	
De advertencia	
Mensaje general	

Las funciones ligadas a los botones de comando se encuentran detallados según muestra la Tabla 18.

**Tabla 18.** Comando e ingreso de datos.

Gráfico	Nombre	Funciones
	Aviso	Aviso de alarmas activas en el proceso.
	Casa	Accede a la pantalla de Inicio o Home.
	Abrir sesión	Acceder al sistema.

	Cerrar sesión	Salir del sistema.
	Salir	Salir de la aplicación (permitido solo para el administrador)
	Arranque	Inicia el proceso
	Paro	Para el proceso
	Reiniciar	Reinicia el proceso
	Botones de navegación	Permiten la navegación entre las ventanas del HMI

#### A. Ventana Principal

La estructura de la ventana principal se observa en la Figura 22, cuenta con el título, ingreso de datos, comandos operativos (pantalla de inicio de sesión). La Figura 23 presenta el aspecto final de la ventana principal.



**Figura 22.** Plantilla de la pantalla de inicio de sesión

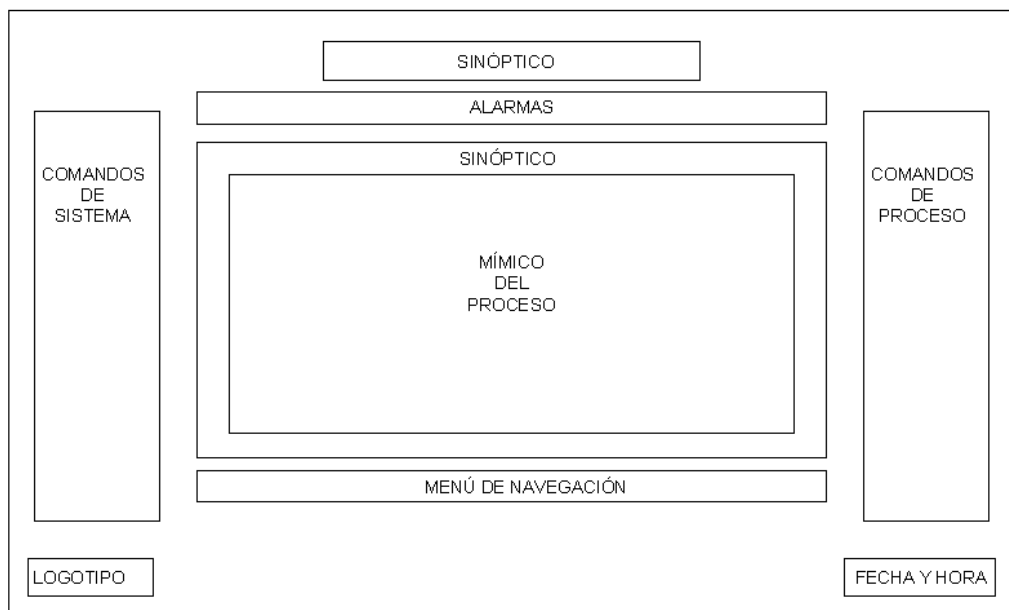




**Figura 23.** Ventana principal.

## B. Pantalla de procesos

En la figura 24 y 25 se observa la estructura de la pantalla para el control de procesos y la vista final de la pantalla de procesos.



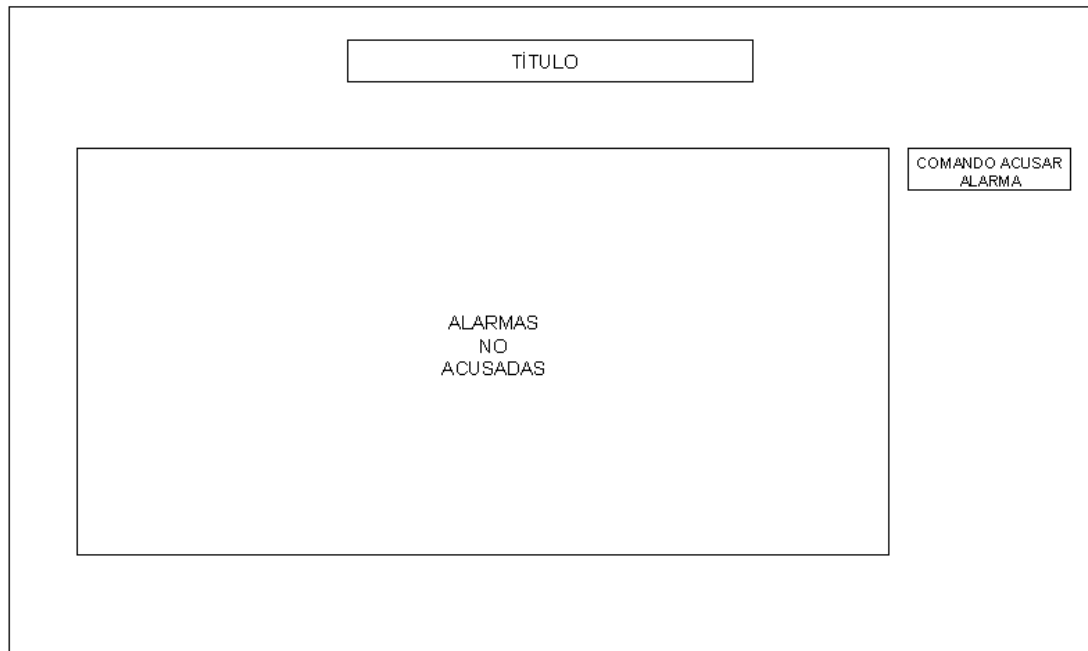
**Figura 24.** Plantilla de la pantalla de procesos.



**Figura 25.** Vista final de la pantalla de procesos.

### C. Pantalla de alarmas no acusadas

La pantalla cuenta con los parámetros de: Título, Registro de alarmas no acusadas y un comando para acusar alarmas, la figura 26 muestra esta estructura.



**Figura 26.** Plantilla de la pantalla de alarmas no acusadas.

AVISOS NO ACUSADOS

Activacion	Alarma	Usuario	Observacion
10-07-2018 22:53:47	Se ha pulsado paro ...	Default	
11-05-2018 03:21:28	Se ha pulsado paro ...	Default	
11-05-2018 03:18:57	Se ha pulsado paro ...	Admin	
11-05-2018 02:47:18	Se ha reiniciado la si...	Admin	
11-05-2018 02:12:48	Se ha reiniciado la si...	Admin	
11-05-2018 01:51:50	Se ha reiniciado la si...	Admin	
11-05-2018 01:49:11	Se ha reiniciado la si...	Admin	
11-05-2018 01:35:11	Se ha reiniciado la si...	Admin	
11-05-2018 00:06:04	Se ha reiniciado la si...	Admin	
10-05-2018 23:52:23	Se ha reiniciado la si...	Admin	
10-05-2018 23:36:01	Se ha reiniciado la si...	Admin	
10-05-2018 23:09:01	Se ha reiniciado la si...	Admin	
10-05-2018 21:58:16	Se ha reiniciado la si...	Admin	
10-05-2018 21:44:46	Se ha reiniciado la si...	Admin	
10-05-2018 21:43:00	Se ha reiniciado la si...	Default	

Acusar Alarma

**Figura 27.** Vista final de pantalla de alarmas.

#### D. Pantalla de registro histórico de alarmas

Posee una estructura simple, puesto que su función principal es dotar de información al operador solo la situación de las alarmas. La estructura está formada por el título y un sinóptico de área para el registro de alarmas. En Figura 28 se puede apreciar la estructura y en la Figura 29 el resultado final de la pantalla de registro.



**Figura 28.** Plantilla de la pantalla de registro histórico de alarmas.

X

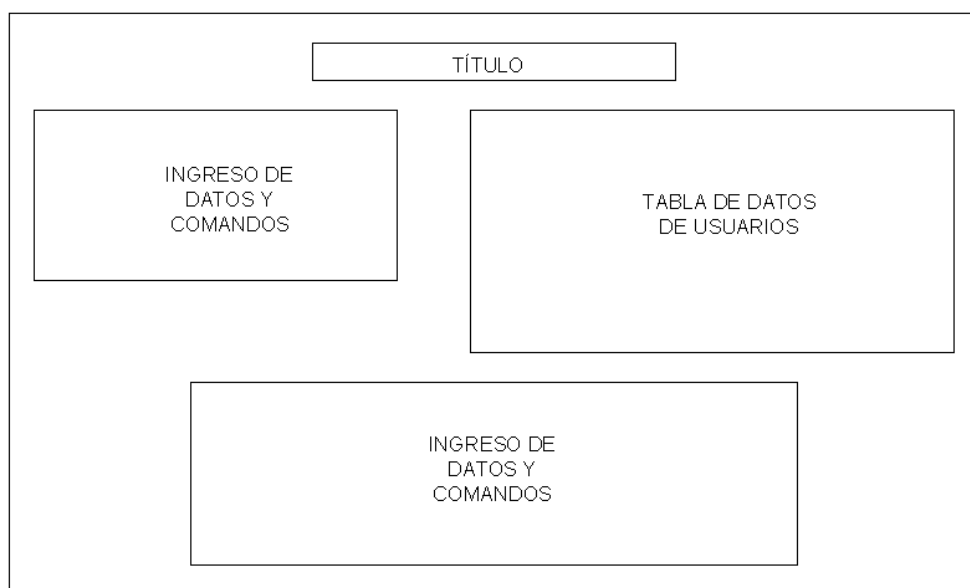
HISTÓRICO DE ALARMAS

Activacion	Alarma	Usuario	Observacion
18-07-2018 18:12:39	Error de conexión	Default	18-07-2018 18:12:44
18-07-2018 18:08:19	Error de conexión	admin	18-07-2018 18:08:20
18-07-2018 18:07:19	Error de conexión	Default	18-07-2018 18:07:22
18-07-2018 17:54:52	Error de conexión	admin	18-07-2018 17:54:53
18-07-2018 17:54:25	Error de conexión	Default	18-07-2018 17:54:28
18-07-2018 14:10:22	Error de conexión	Default	18-07-2018 14:10:23
18-07-2018 14:08:43	Error de conexión	Default	18-07-2018 14:08:46
18-07-2018 12:04:23	Error de conexión	admin	18-07-2018 12:04:25
18-07-2018 12:00:14	Error de conexión	Default	18-07-2018 12:03:54
18-07-2018 11:39:29	Error de conexión	Default	18-07-2018 11:39:57
18-07-2018 11:34:11	Error de conexión	Default	18-07-2018 11:34:13
18-07-2018 11:33:09	Error de conexión	Default	18-07-2018 11:33:14
18-07-2018 11:24:41	Error de conexión	admin	18-07-2018 11:24:42
18-07-2018 11:24:25	Error de conexión	admin	18-07-2018 11:24:27
18-07-2018 11:24:17	Error de conexión	admin	18-07-2018 11:24:18

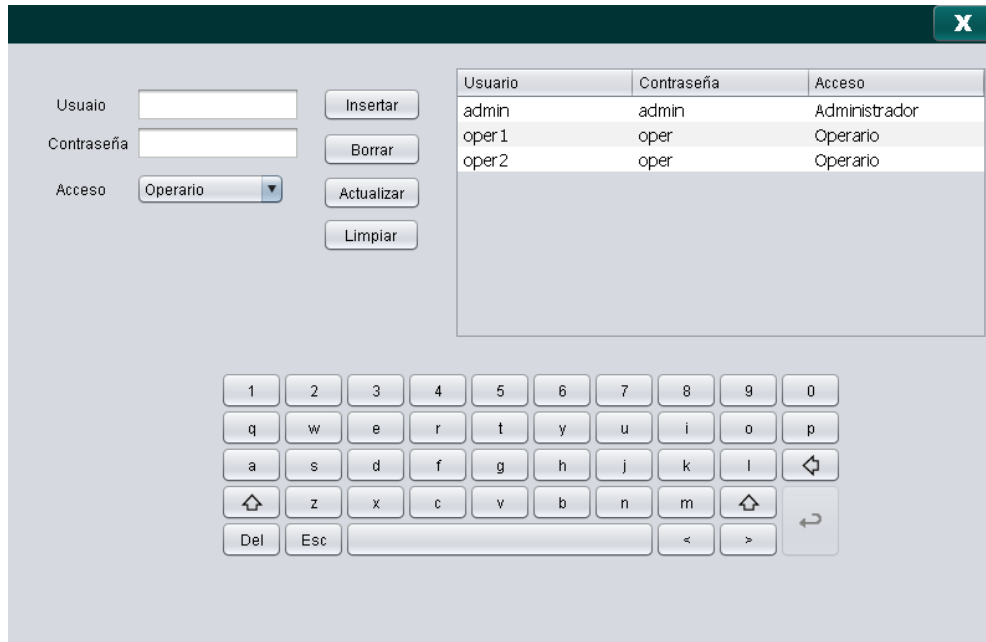
**Figura 29.** Vista final de la pantalla de registro de alarmas.

### E. Pantalla de gestión de usuarios

Está compuesta por los siguientes ítems: título, sinópticos para el ingreso de datos y comandos del administrador, para visualizar los operadores creados, y por último un sinóptico de área donde se ubicó un teclado para el ingreso de datos y comandos (Figura 30).



**Figura 30.** Plantilla de la pantalla de gestión de usuarios.

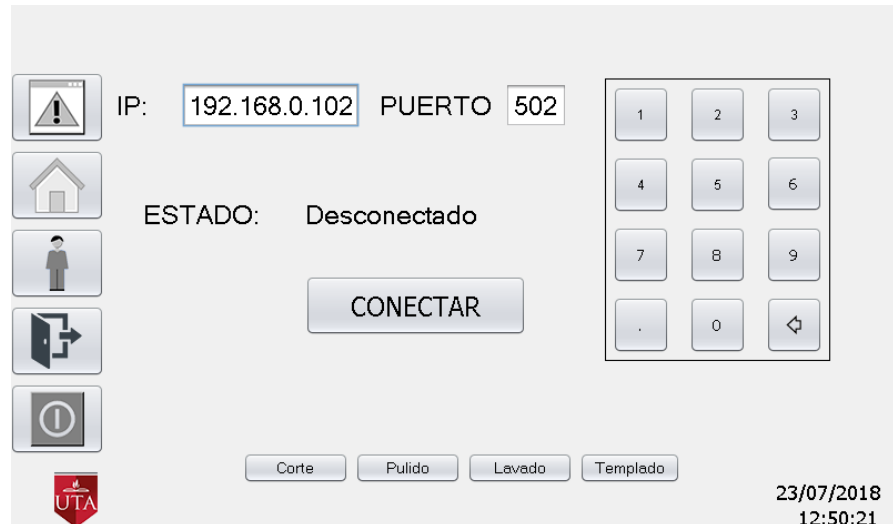


**Figura 31.** Vista final de la pantalla de gestión de usuarios.

#### 4.5.2. Navegación de HMI

La pantalla de arranque tiene por objetivo controlar el acceso del personal a las operaciones tanto de monitoreo como de control, la interfaz inicia totalmente bloqueada permitiéndose únicamente el acceso por parte de los usuarios registrados (Figura 32).

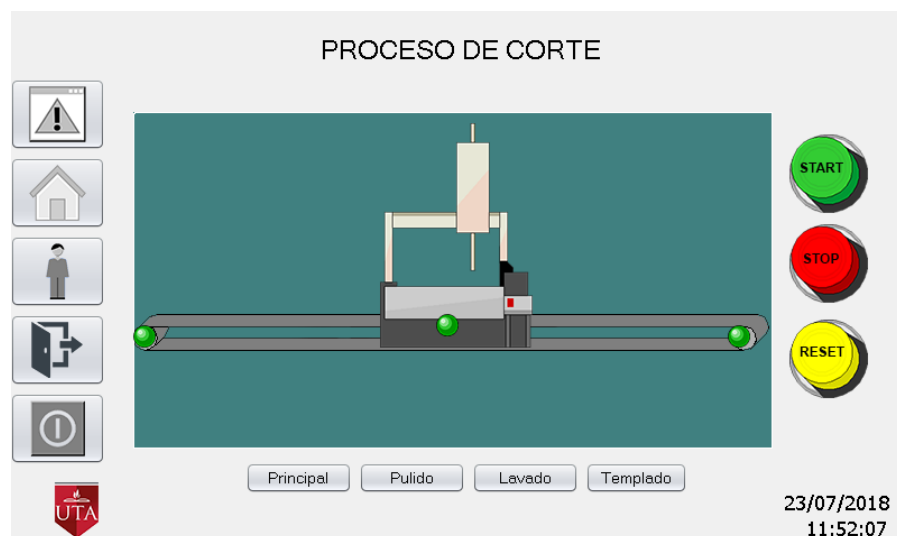
Los usuarios quienes estén registrados como operadores tienen únicamente el acceso a las ventanas de bienvenida y de los procesos, para ello solo se activan los botones 1, 2, 3, 4, mientras que los demás permanecen inactivos incluido el botón de salida. Para el usuario que es administrador, activan todos los campos y opciones de la página de arranque, además incrementa las ventas a las que tienen acceso como por ejemplo la ventana de gestión de usuarios y configuración de conectividad. En la Figura 32 se muestra la pantalla de conexión del usuario registrado como administrador.



**Figura 32.** Pantalla de conexión.

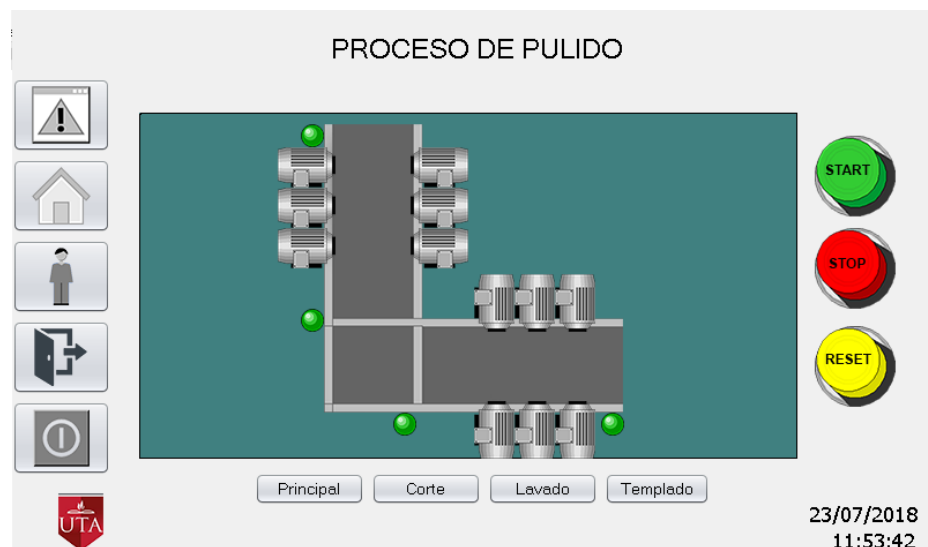
**a) Interfaz de cada bloque del proceso**

**A. Proceso de Corte:** La ventana que sirve como medio de monitoreo y control para el proceso se visualiza en la figura 33, el cual contiene los botones de arranque, paro y parada de emergencia que funciona de la misma forma que los empleados en la simulación, además no presenta el botón de reajuste, debido a que esta función solo se puede realizar desde la máquina. Cuenta con algunos sensores digitales representados por los círculos de color verde, que se encienden al ser activados y la máquina cambia de color [29].



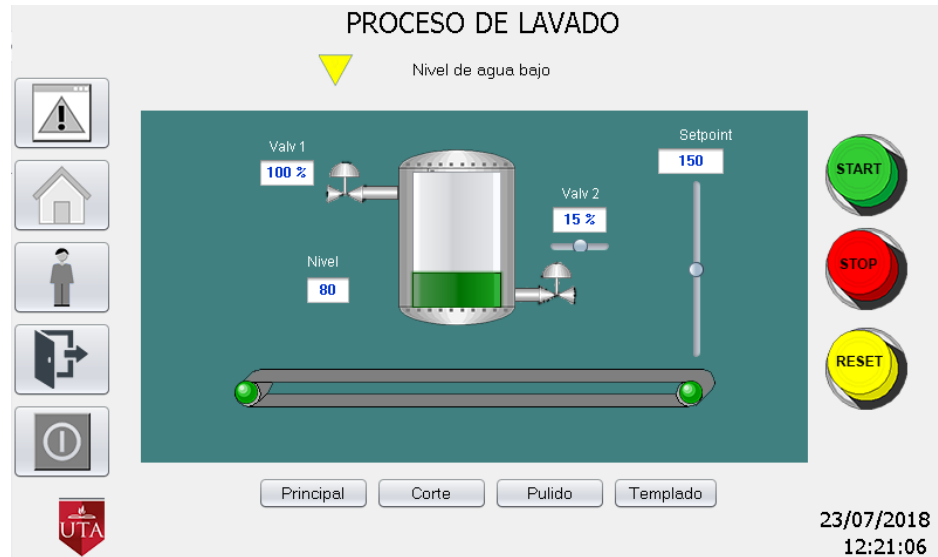
**Figura 33.** Ventana de proceso de corte.

**B. Proceso de Pulido:** En la Figura 34 se puede apreciar la ventana de monitoreo y control del proceso de pulido, en el cual se observan características similares a las que presenta el mismo proceso en la simulación. La ventana cuenta con 4 círculos pequeños que simulan los sensores presentes en el proceso los cuales se activan al ingresar el producto a pulir, en este caso se activan los motores dispuestos paralelamente.



**Figura 34.** Ventana del proceso de Pulido.

**C. Proceso de Lavado:** La ventana cuenta con algunos indicadores que cumplen la función de visualizar los datos correspondientes al nivel y caudal del tanque que tiene el proceso, cuenta con 2 válvulas de apertura para reducir o aumentar el caudal y además también presenta un regulador tipo slider para configuración del punto de ajuste del tanque (SETPOINT). Al igual que las ventanas de los procesos ya mencionados cuenta con sus respectivos botones de control y también con los sensores de activación de la máquina como se puede apreciar en la figura 35.



**Figura 35.** Ventana del proceso de lavado.

**D. Proceso de Horneado o templado:** El horno ventana del proceso de templado la cual se puede apreciar en la figura 36, consta de 4 etapas: ingreso de materia prima, calentamiento del vidrio, enfriado por aire, y salida de producto.

- a. El ciclo empieza al encender la máquina, además la banda transportadora inicia su operación cuando cumpla las condiciones de: calibración de la temperatura del horno y la posición correcta del objeto.
- b. Dentro del horno hay un sensor de difuso que detecta el objeto en posición, aquí espera un tiempo determinado en el cual el vidrio se encuentra a la temperatura [referencia libro].
- c. Posterior al calentamiento, se enfría mediante aire comprimido para lo cual el objeto debe llegar a la posición del sensor donde se activarán los ventiladores que enfriarán el producto hasta los 20° C, finalmente un sensor emite la señal para la salida del producto del sistema.

Un punto importante a considerar es que el proceso de calentamiento y enfriado no existen en Factory I/O, por lo tanto, se establecen mediante programación en java en la raspberry pi para dar la apariencia de funcionamiento.



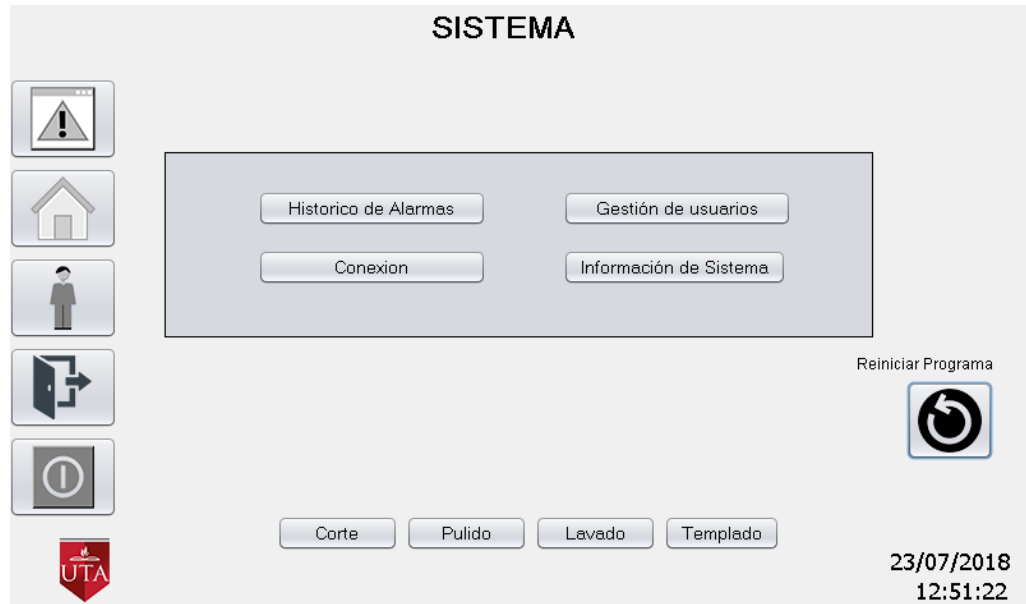


**Figura 36.** Ventana del proceso de templado.

## b) Pantalla del administrador

La interfaz gráfica mostrada en la figura 37, la cual tiene acceso únicamente el o los administradores presenta un menú de opciones pertenecientes al sistema controlado, entre estos tenemos:

- Información sobre el proyecto: Despliega una ventana con información en texto que incluye (Nombre del proyecto; Autor; Última actualización y descripción breve del proyecto).
- Información sobre el sistema: Incluye información Sobre el tipo de controlador, el tipo de pantalla (dispositivo de entrada y salida pantalla touch, tipo de conectividad).
- Administración de usuarios; Muestra una ventana donde se ingresan, modifican y eliminan los usuarios que tienen acceso al sistema.
- Historial de alarmas que muestra un registro histórico de alarmas que se han presentado en el sistema



**Figura 37.** Ventana de información para administrador.

#### **4.6. Controlador**

El lenguaje de programación bajo el cual se codificó los algoritmos de control del sistema simulado de producción de vidrio, el cual ya fue mencionado, es Java con el IDE (entorno de desarrollo) NETBEANS. El siguiente diagrama de clases mostrado en la figura 38 presenta la estructura del funcionamiento del controlador del sistema. El diagrama dado la amplia extensión de parámetros y métodos incluidos en la programación a sido simplificado para denotar únicamente la relación existente, entre las clases de programación.

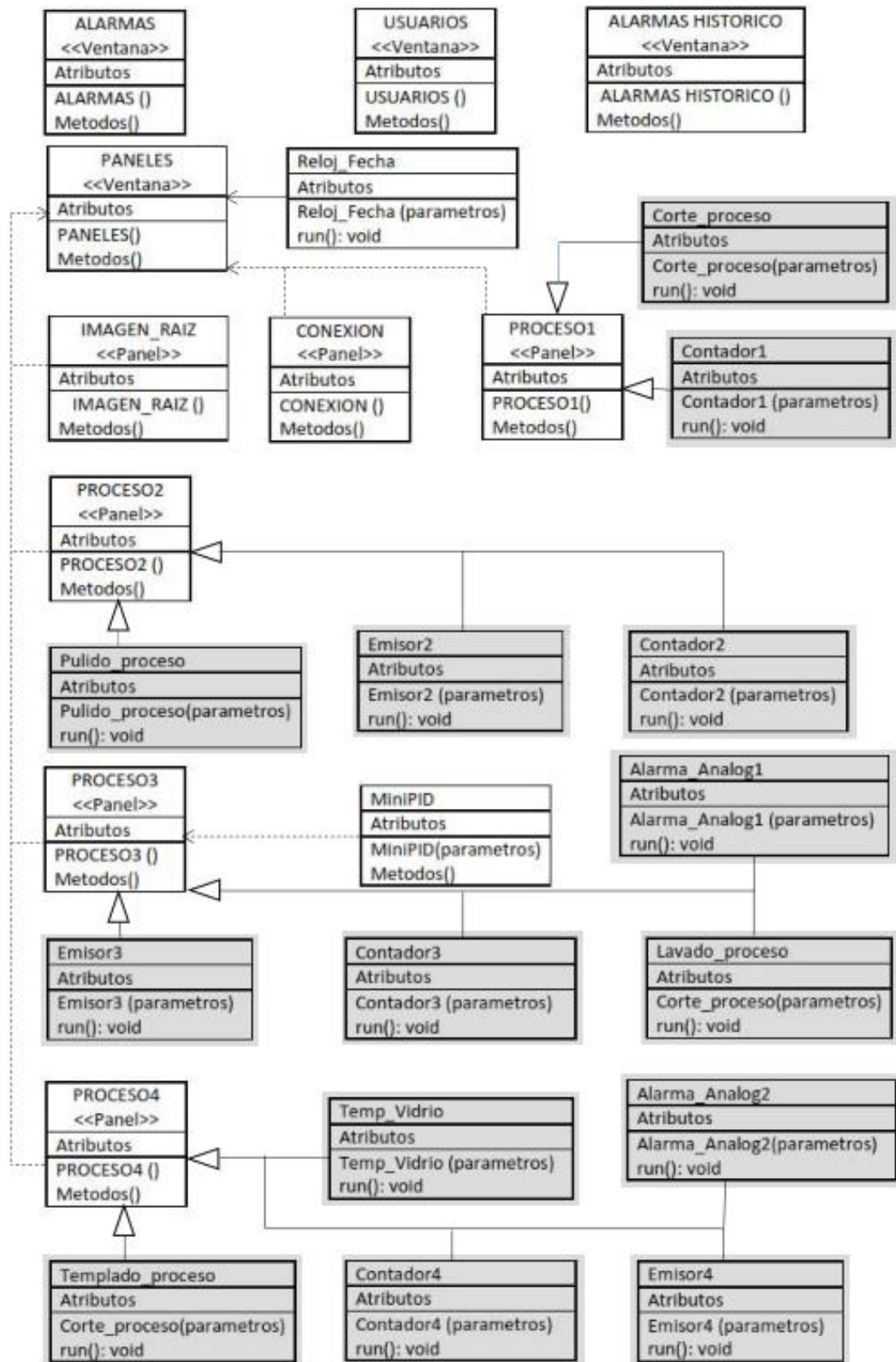


Figura 38. Diagrama de clases del algoritmo de control.

- A. ALARMAS (Formulario):** Formulario de consulta en base de datos SQLITE, donde se despliegan las alarmas y avisos que no han sido atendidas. En esta clase se implementa la librería sliteJDBC para su funcionamiento.
- B. ALARMAS HISTÓRICO (Formulario):** Formulario de consulta en base de datos SQLITE, en la cual se despliegan todas las alarmas y avisos registrados de forma cronológica. En esta clase se implementa la librería sliteJDBC para su funcionamiento.
- C. USUARIOS (Formulario):** Formulario de gestión de la base de datos de usuarios, permite la modificación, creación y eliminación de usuarios. De igual forma a los anteriores formularios funciona mediante la librería sliteJDBC. El ingreso de datos se realiza mediante un teclado virtual compuesto por botones.
- D. PANELES (Formulario):** Ventana principal del HMI, se mantiene como panel por defecto la interfaz de inicio de sesión, con la implementación de la base de datos en slite, y como medio para el ingreso de datos se usó un teclado virtual. Esta ventana gestiona todos los cambios de interfaces (navegación) y contiene el menú de navegación y los comandos de sistema.
- E. IMAGEN\_RAIZ:** Panel con comandos e indicaciones mostrados según el nivel de permiso del usuario que ha iniciado sesión.
- F. CONEXION:** Panel de prueba de conexión, el cual contiene un teclado numérico virtual para el ingreso de datos. En este panel se almacena un archivo de texto el cual contiene la última IP ingresada, para evitar el continuo ingreso de este parámetro en próximas ejecuciones.
- G. PROCESO 1 (Panel):** Correspondiente al proceso de corte, se implementan los comandos, alarmas, y sinóptico del proceso. En este se ejecutan dos hilos mediante el cual funciona el HMI.

- a. **Corte\_proceso (Thread):** Clase que contiene el algoritmo de control del proceso de corte. Esta clase implementa el método run() para crear un hilo de ejecución.
- b. **Contador 1 (Thread):** Esta clase corresponde a un hilo ejecutable, el cual se encarga de contar las unidades producidas en la estación 1, a su vez emite la materia prima al inicio del proceso y al salir la unidad del mismo. El contador de esta clase es indispensable para el segundo proceso, ya que el número de unidades producidas en el proceso de corte será el mismo para los posteriores procesos.

**H. PROCESO 2 (Panel):** Correspondiente al proceso de pulido, se implementa los comandos, alarmas, y sinóptico del proceso. En este se ejecutan hilos del cual depende la lógica del proceso de pulido.

- a. **Pulido\_proceso (Thread):** Clase (hilo) de ejecución, el cual contiene la lógica del proceso de pulido. Contempla la lectura y escritura de variables, así como el funcionamiento de elementos gráficos del sinóptico
- b. **Contador 2(Thread):** Clase encargada del conteo de unidades producidas en el proceso de pulido.
- c. **Emisor2 (Thread):** Clase encargada del ingreso y remoción de materia prima. Esta clase depende del hilo, contador1 para su funcionamiento.

**I. PROCESO 3 (Panel):**

- a. Lavado\_proceso (Thread)
- b. **MiniPID:** Controlador PID contenido en una clase, el cual mediante el paso de parámetros emite la señal de control para la válvula de llenado del tanque de agua de la simulación.
- c. **Contador 3(Thread):** Clase encargada del conteo de unidades del proceso de lavado
- d. **Emisor3 (Thread):** Clase encargada de la colocación y remoción de materia prima del lavado

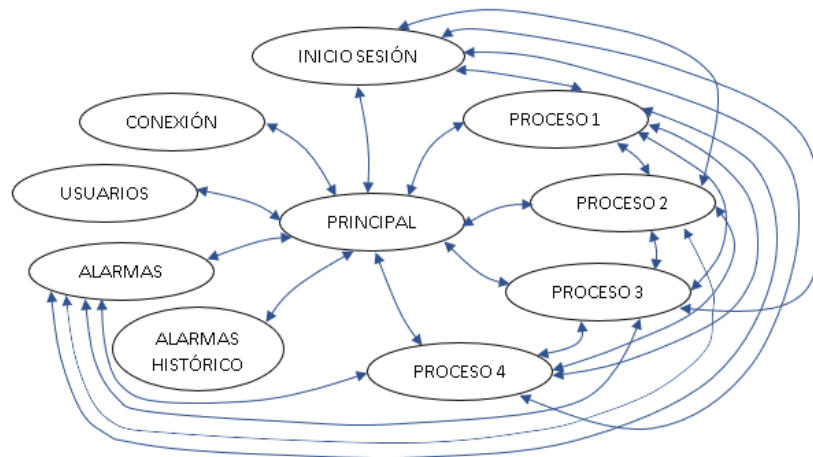
- e. **Alarma\_Analog1:** Clase en la que se definen los valores muy alto alto, bajo y muy bajo correspondiente a la variable analógica de nivel

**J. PROCESO 4 (Panel):**

- a. **Templado\_proceso (Thread):** Clase que contiene la lógica del proceso.
- b. **Contador 4(Thread):** Clase que cuenta las unidades producidas en el proceso de templado.
- c. **Emisor4 (Thread):** Clase encargada de colocar y retirar la materia prima del proceso. Esta depende del contador del proceso anterior.
- d. **Temp\_Vidrio:** Clase que simula mediante una función en el tiempo, la trasferencia de calor en el proceso de calentamiento.
- e. **Alarma\_Analog2:** Clase en la que se definen los valores muy alto alto, bajo y muy bajo correspondiente a la variable analógica de temperatura

**K. Diagrama de navegación**

Tanto las ventanas como los paneles presentan un despliegue de forma específica, con combinaciones las cuales dependen según la gráfica mostrada en la figura 39. La navegación en todos los casos es bidireccional, es decir se puede retornar a una ventana anterior a la actual, además algunas funciones son desactivadas para el usuario con acceso de tipo operario.



**Figura 39.** Relaciones de acceso en el sistema de control.

La Tabla 19 muestra la descripción de acceso y navegación de cada uno de los parámetros mostrados en la figura 39.

**Tabla 19.** Relaciones de acceso

<b>Ítem</b>	<b>Descripción</b>
INICIO SESIÓN	Acceso únicamente a la ventana principal
PRINCIPAL	Acceso a las pantallas de conexión, gestión de usuarios, alarmas no acusadas, registro histórico de alarmas, y cada uno de los procesos, en todos estos casos es posible retornar a la pantalla principal
PROCESO 1	Navegación a todos los demás procesos, a la pantalla principal, alarmas e inicio de sesión.
PROCESO 2	Se puede navegar a todos los demás procesos, a la pantalla principal, alarmas e inicio de sesión.
PROCESO 3	Se puede navegar a todos los demás procesos, a la pantalla principal, alarmas e inicio de sesión
PROCESO 4	Se puede navegar a todos los demás procesos, a la pantalla principal, alarmas e inicio de sesión.

## **4.7. Pruebas y resultados**

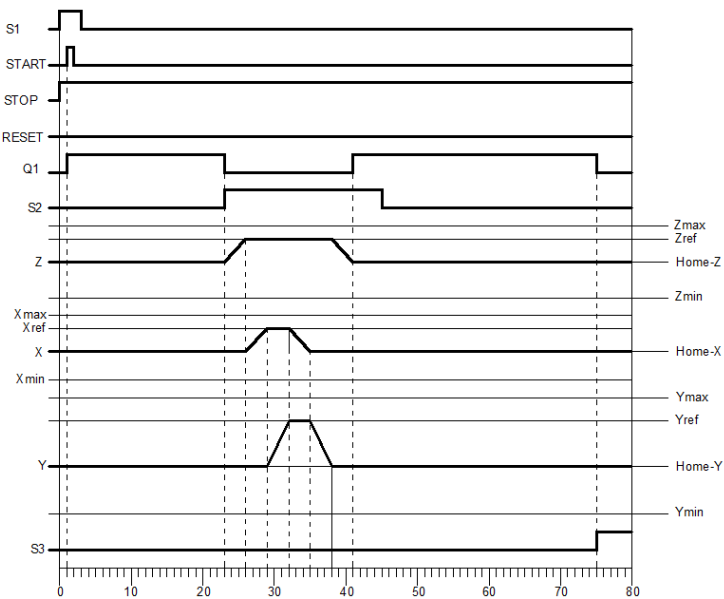
### **4.7.1. Pruebas y resultados en la estación de corte**

El diagrama secuencial mostrado en la figura 40 muestra el funcionamiento de la estación de corte de vidrio, y en la Tabla 20 se observan las direcciones establecidas para cada elemento (comandos, actuadores, sensores) perteneciente a la estación.

**Tabla 20.** Direcciones del proceso de Corte.

Direcciones	Detalle
S1	Sensor de presencia ubicado a la entrada del proceso
START	Pulsador de Macha
STOP	Pulsador de Paro
RESET	Pulsador de Reajuste
Q1	Motor de la banda
S2	Sensor de presencia ubicado en posición del plotter de corte
Zmax.	Posición máxima que se puede alcanzar en el eje Z
Zref.	Posición deseada en el eje Z
Home-Z	Posición inicial del eje Z
Z	Motor que mueve el eje Z del plotter
Zmin.	Posición mínima que se puede alcanzar en el eje Z
Xmax.	Posición máxima que se puede alcanzar en el eje X
Xref.	Posición deseada en el eje X
Home-X	Posición inicial del eje X
X	Motor que mueve el eje X del plotter
Xmin.	Posición mínima que se puede alcanzar en el eje X
Ymax.	Posición máxima que se puede alcanzar en el eje Y
Yref.	Posición deseada en el eje Y
Home-Y	Posición inicial del eje Y
Y	Motor que mueve el eje Y del plotter
Ymin.	Posición mínima que se puede alcanzar en el eje Y
S3	Sensor de presencia ubicado a la salida del proceso.





**Figura 40.** Diagrama secuencial del proceso de corte

### A. Descripción del funcionamiento

- La estación de corte se encuentra previamente cargada con el material en la entrada de la misma, por lo tanto, el sensor S1 se encuentra activado.
- Inicia el desplazamiento del material por la banda transportadora al pulsar el comando N/A de Marcha (START) mientras el sensor 1 está activo, el cual activan el motor (M1) de la banda.
- El pulsador de Paro (STOP) permanece Normalmente Cerrado
- El pulsador de Reajuste (RESET) Permanece Normalmente Abierto
- Al llegar el material a la posición de corte el sensor S2 se activa y la banda transportadora se detiene.
- Al activarse el sensor S2, el motor del eje Z del plotter dirige el mismo a su posición de referencia Zref.
- Al llegar el eje Z a su posición de referencia, el motor del eje X dirige este a su posición de referencia Xref.
- Al llegar el eje X a su posición de referencia, el motor del eje Y dirige este a su posición de referencia Yref.
- Al llegar el eje Y a su posición de referencia, el motor del eje X dirige este a su posición inicial.

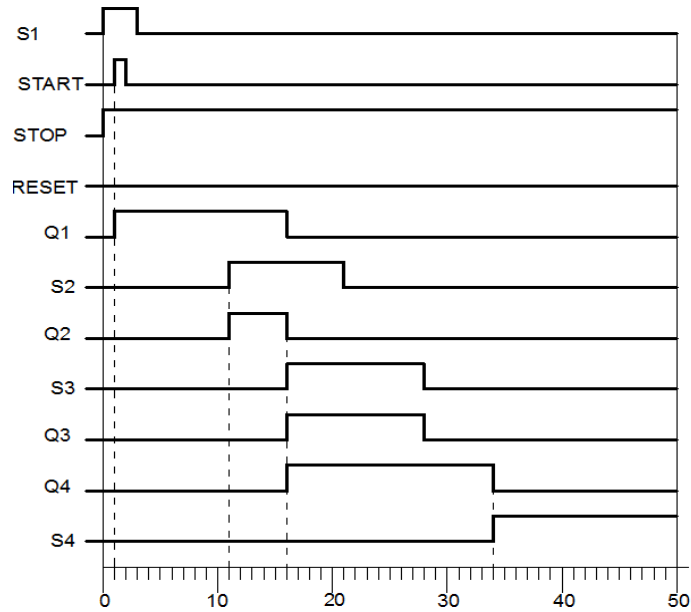
- Al llegar el eje X a su posición de inicial, el motor del eje Y dirige este a su posición de inicial.
- Al llegar el eje Y a su posición de inicial, el motor del eje Z dirige este a su posición de inicial.
- Al llegar el eje Z a su posición de inicial, el motor de la cinta transportadora continua su movimiento.
- Al retirar el objeto del alcance del sensor S2 la señal de este se desactiva.
- Al llegar el material a su posición final donde se encuentra el sensor S3, este se activa, mientras que la cinta transportadora se detiene.
- El sensor S3 permanece activado tanto como se tarde en retirar el material

#### **4.7.2. Pruebas y resultados en la estación de pulido**

El funcionamiento de la estación se describe a continuación:

- La estación se encuentra previamente cargada con el material en la entrada de la misma, por tanto, el sensor S1 se encuentra activado.
- Al pulsar el pulsador N/A de Marcha (START) mientras el sensor 1 está activado el motor (M1) de la cinta transportadora se activa.
- El pulsador de Paro (Stop) permanece Normalmente Cerrado
- El pulsador de Reajuste (RESET) Permanece Normalmente Abierto
- El sensor 2 se activa al ingresar la materia prima al transferidor hasta que sale del mismo.
- Al activarse el sensor 2 se activa el motor de entrada al transferidor
- El sensor 3 se activa al llegar la materia prima al final del transferidor hasta que la misma sale de este.
- El motor de entrada al transferidor se desactiva al activarse el sensor 3 al igual que el motor de la banda 1, mientras que al mismo tiempo se activa el motor de salida del transferidor hasta que la materia prima abandone el mismo.
- La banda 2 se activa al mismo tiempo que el motor de salida del transferidor hasta que el sensor 4 es activado.
- El sensor 4 se mantiene activado mientras la materia prima no se retire del final del proceso.

El diagrama secuencial mostrado en la figura 42 describe el comportamiento de los elementos que forman el sistema, además en la tabla 21 se describen las direcciones establecidas.



**Figura 41.** Diagrama secuencial del proceso de pulido

**Tabla 21.** Direcciones del proceso de Pulido.

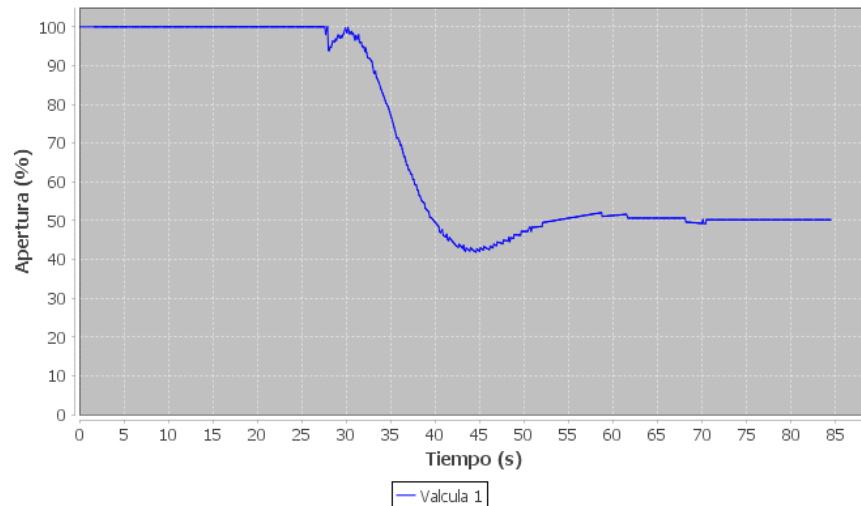
Direcciones	Detalle
S1	Sensor de presencia ubicado a la entrada del proceso
START	Pulsador de Macha
STOP	Pulsador de Paro
RESET	Pulsador de Reajuste
Q1	Motor de la banda
S2	Sensor de presencia ubicado en posición de entrada al transferidor
Q2	Motor de entrada al transferidor
S3	Sensor de presencia ubicado en el final del transferidor
Q3	Motor de salida del transferidor
Q4	Motor de la banda 2
S4	Sensor de presencia ubicado a la salida del proceso

### 4.7.3. Pruebas y resultados en la estación de lavado de vidrio

De acuerdo a la simplicidad en el funcionamiento de una lavadora de vidrio en la producción de vidrio templado, se ha implementado un controlador proporcional integral derivativo (PID) en el tanque reservorio de agua, el cual presenta las siguientes características:

- Capacidad 300 litros
- Radio tubería de descarga: 0.125 m
- Caudal máximo de entrada: 0.25 m<sup>3</sup>/s
- Caudal máximo de salida: 0.3543 m<sup>3</sup>/s

En la Figura 42 se observa la gráfica del comportamiento que tiene la válvula de control, este comportamiento radica en una tendencia cambiante de acuerdo al porcentaje de apertura dado, por ejemplo, se observa que al cambiar de una apertura de 100% a una 50% la gráfica no refleja un cambio lineal-uniforme, debido a que corresponde a una señal de control dada por el controlador.



**Figura 42.** Gráfica del comportamiento de la válvula de control.

El llenado del tanque empieza en  $t = 8$  seg como se puede apreciar en la Figura 42 y durante el lapso de tiempo 8 a 50 seg se tiene una señal creciente debido a que el nivel va aumentando de 0 a el máximo permitido de 500 litros, a partir del  $t = 60$  seg el nivel del tanque se mantiene estable a la misma magnitud.

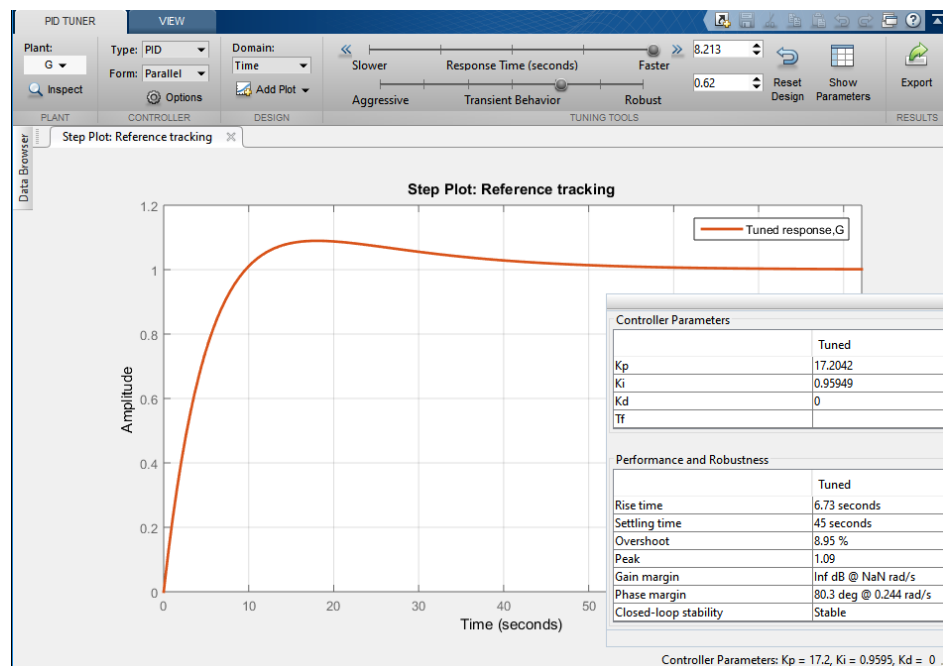
La sincronización del controlador PID para el llenado del tanque reservorio, parte de la recolección de datos dada una entrada tipo Escalón, estos datos son:

- Entrada = Apertura de la válvula de entrada al 50%
- Salida= Lectura del sensor de nivel

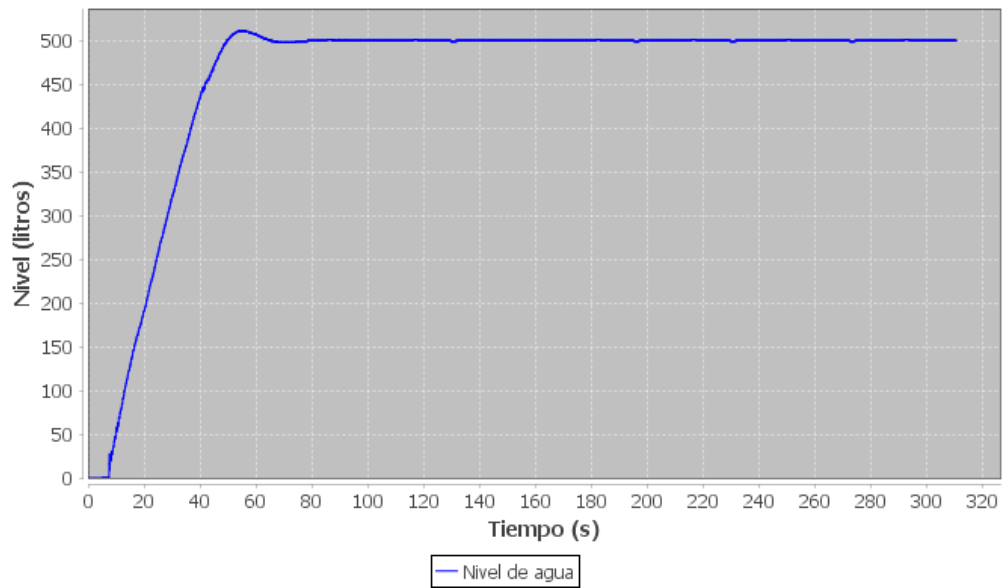
Mediante una función codificada en Java se almacenaron los datos de porcentaje de apertura de la válvula de entrada de agua y el nivel del tanque reservorio en un archivo de texto para su posterior uso. Mediante la herramienta *ident* de Matlab se obtuvo la función de transferencia de la planta.

$$G(s) = \frac{0.01382}{s+0.01382} \quad (1)$$

Mediante la herramienta PID tuner de Matlab se realizó la obtención de las ganancias requeridas para obtener una respuesta óptima en el sistema de control. La respuesta obtenida de la aplicación se puede apreciar en la figura 44.



**Figura 43.** Sintonización del PID de planta en Matlab.

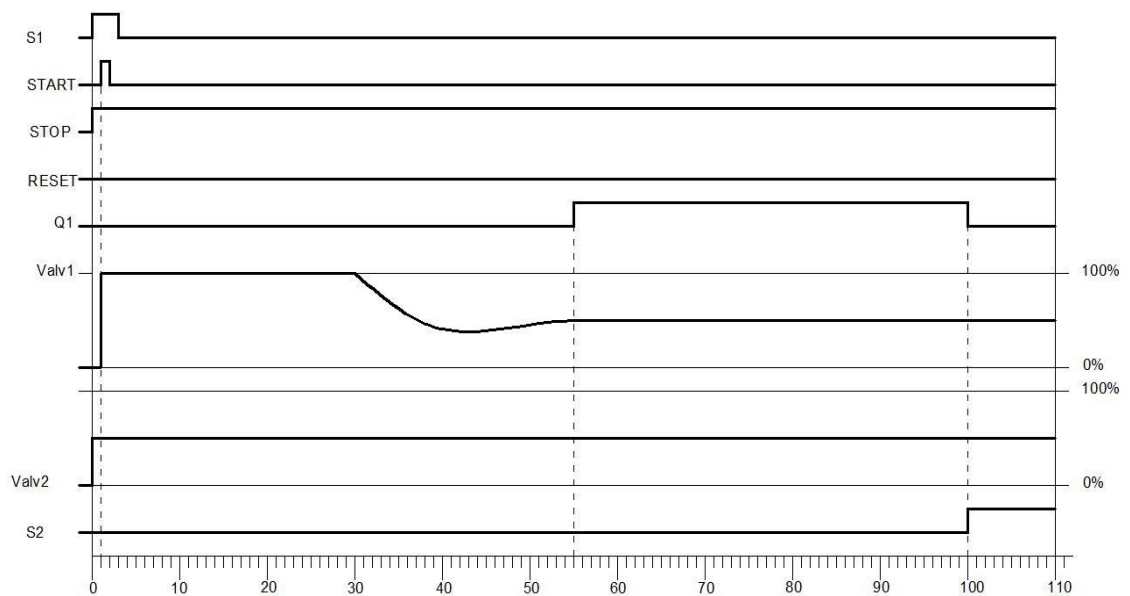


**Figura 44.** Respuesta del nivel del tanque reservorio.

Las direcciones establecidas en el proceso de lavado se visualizan en la Tabla 22. El diagrama secuencial del proceso se presenta en la figura 45.

**Tabla 22.** Direcciones del proceso de Lavado de vidrio.

Direcciones	Detalle
S1	Sensor de presencia ubicado a la entrada del proceso
START	Pulsador de Macha
STOP	Pulsador de Paro
RESET	Pulsador de Reajuste
Q1	Motor de la banda
Valv1	Válvula 1
Valv2	Válvula 2
S2	Sensor de presencia ubicado en el final del transferidor
Q3	Motor de salida del transferidor
Q4	Motor de la banda 2
S4	Sensor de presencia ubicado a la salida del proceso



**Figura 45.** Diagrama secuencial del proceso de Lavado de vidrio.

#### **A. Descripción del funcionamiento**

- La estación se encuentra previamente cargada con el material en la entrada de la misma, por tanto, el sensor S1 se encuentra activado.
- Al pulsar el pulsador N/A de Marcha (START) el tanque empieza a llenarse hasta el SETPOINT configurado (medio tanque =150 litros).
- El pulsador de Paro (STOP) permanece Normalmente Cerrado.
- El pulsador de Reajuste (RESET) Permanece Normalmente Abierto (NA).
- La válvula 1 se abre según la señal proporcionada.
- La válvula 2 permanece abierta en un 50% (constante).
- Al llegar al valor deseado del nivel la banda 1 se activa hasta llegar al final del proceso activándose el sensor de presencia 2 (S2).
- El sensor 4 permanece activado hasta que se retire la materia prima del final del proceso.

#### 4.7.4. Pruebas y resultados de la estación de templado.

Teniendo como base la transferencia de calor producida cuando un vidrio a temperatura ambiente se coloca en un horno caliente, la Tasa de energía térmica que absorbe el vidrio es proporcional a la diferencia de temperatura entre el horno y el vidrio. A medida que el vidrio se calienta, la diferencia de temperatura disminuye y la velocidad de transferencia de energía se reduce. Esto produce una curva de calentamiento que es exponencial.

La expresión matemática empleada es:

$$y = T_{max}(1 - e^{-k}) \quad (2)$$

Donde  $y$  es la temperatura actual del vidrio,  $T_{max}$  es la temperatura del horno  $k$  es la constante de tiempo. Siendo  $k$ :

$$k = \frac{t}{d \cdot 17,6} \quad (3)$$

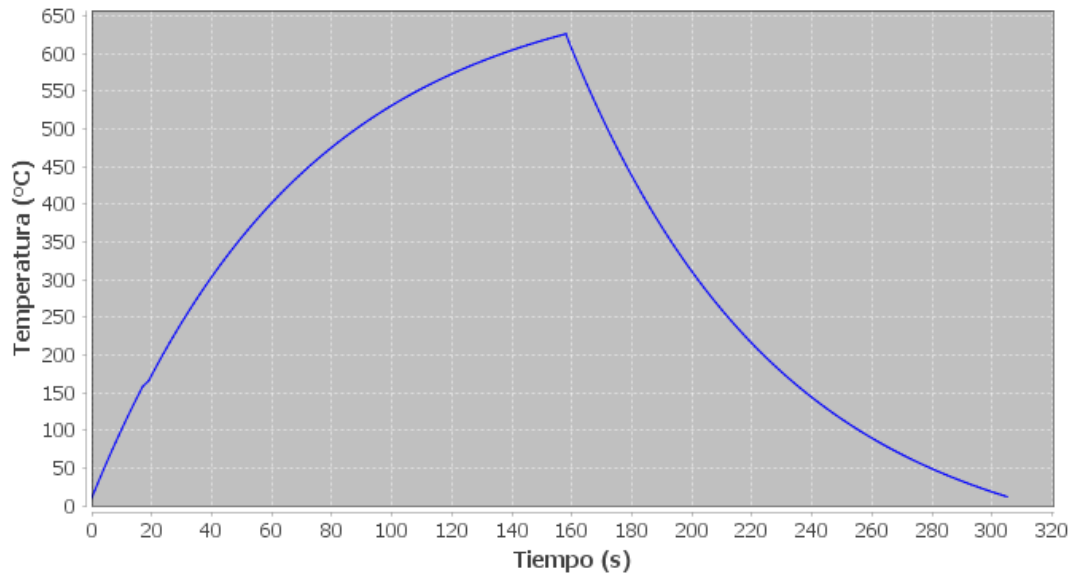
Donde  $d$  es espesor del vidrio, Según *Glass tempering Handbook* [25], la temperatura estándar de salida del vidrio corresponde a 625°, y para un horno caliente a 700°, se requiere una constante de tiempo  $k$  de aproximadamente 2.22, sin importar el espesor del vidrio, por lo tanto, para un vidrio de 4 mm de espesor se tiene que:

$$\text{Tiempo en el Horno} = 2.224 \times 17.6$$

$$\text{Tiempo en el Horno} = 156 \text{ segundos}$$

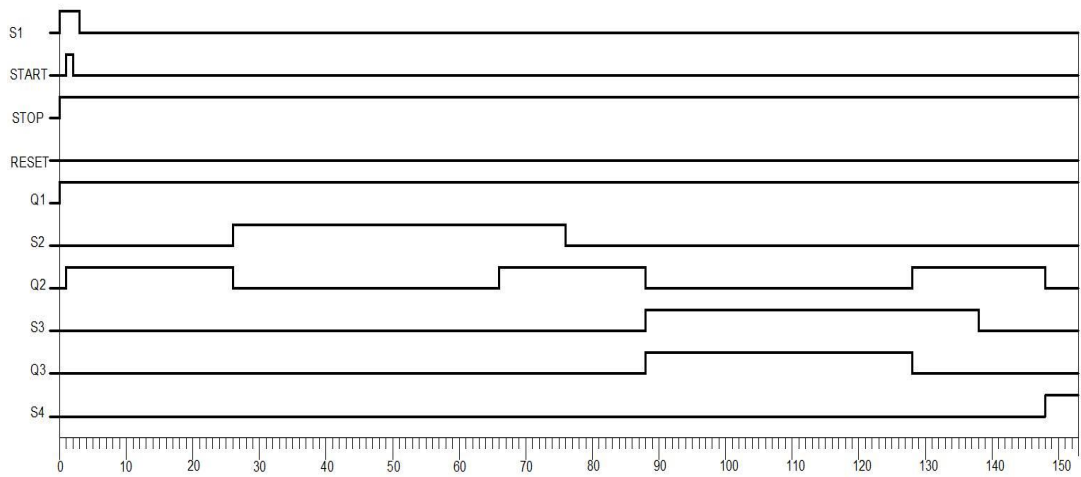
Una vez obtenido el tiempo de calentamiento, dato con el cual se ha programado un temporizador para que cuando ingresa el vidrio al horno, este espere el tiempo de calentamiento calculado, luego el vidrio es enviado a la cabina de enfriado con el mismo tiempo del horno. La figura 46 muestra la curva característica de este proceso.





**Figura 46.** Curva de calentamiento del vidrio templado

Las direcciones establecidas para la estación en el proceso de templado se describen en la tabla 23 y en la figura 47.



**Figura 47.** Diagrama secuencial del proceso de Templado.

**Tabla 23.** Direcciones del proceso de Templado.

Direcciones	Detalle
S1	Sensor de presencia ubicado a la entrada del proceso
START	Pulsador de Macha
STOP	Pulsador de Paro
RESET	Pulsador de Reajuste
Q1	Horno activado (Previamente encendido/precalentado)
S2	Sensor de presencia ubicado en la zona de calentamiento
Q2	Motor de la banda transportadora
S3	Sensor de presencia ubicado en la zona de enfriamiento
Q3	Motores de los ventiladores de la zona de enfriamiento
S4	Sensor de presencia ubicado a la salida del proceso

#### **A. Descripción del funcionamiento**

- La estación se encuentra previamente cargada con el material en la entrada de la misma, por tanto, el sensor S1 se encuentra activado.
- Se considera que el horno se encuentra previamente precalentado (encendido).
- Al pulsar el pulsador N/A de Marcha (START) la banda se activa hasta que la materia prima llega a la posición de calentamiento.
- El sensor 2 se activa al llegar la materia prima a la zona de calentamiento, y la banda se desactiva tanto según el tiempo necesario para calentar el vidrio a 625° (156 segundos reales, considerado 40 con motivos prácticos)
- Transcurrido el tiempo la banda se reactiva, y el sensor 2 se desactiva al dejar de detectar la presencia de la materia prima
- Al activarse el sensor 3 (S3) la banda se detiene en la zona de enfriamiento, donde se activa la señal que enciende los ventiladores durante el tiempo necesario para enfriar el vidrio (156 segundos reales.)
- Transcurrido el tiempo la banda se reactiva, y el sensor 3 se desactiva al dejar de detectar la presencia de la materia prima.
- Al activarse el sensor 4 (S4) la banda se desactiva. El sensor 4 permanece activado hasta que el producto terminado sea retirado.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

- La distribución de una planta de vidrio templado corresponde a una configuración lineal, donde el producto pasa a través de varias estaciones; el software de simulación ha permitido representar cada una de ellas, así como también ubicarlas según la distribución de la planta, facilitando la visualización del proceso desde varios puntos de vista a través de la navegación en entorno virtual construido.
- Se ha utilizado la librería de sensores y actuadores disponibles en Factory I/O, los cuales mediante el uso de direcciones específicas de entrada y salida basadas en protocolos industriales hacen posible su lectura desde dispositivos externos. El uso de tarjetas de desarrollo que cuentan con opciones de conectividad para la adquisición de datos facilita la comunicación bajo protocolos industriales en aplicaciones de bajo costo.
- El uso de protocolos abiertos como MODBUS facilita el desarrollo de aplicaciones de automatización, ya que se dispone de una amplia gama de dispositivos industriales con los cuales se puede establecer una comunicación. Empleando la compatibilidad del simulador con el protocolo, se ha aprovechado el conector ethernet de la raspberry pi y la computadora, los cuales se comunican a través de un ROUTER, mediante la configuración Maestro (Factory I/O), esclavo (Raspberry pi).
- Mediante el uso de herramientas disponibles en matlab para la identificación de la función de transferencia del tanque de agua empleado en la estación de lavado, se consiguió la sintonización óptima del controlador PID, en este caso el error en estado estacionario presenta una tendencia a cero y respetando una tolerancia de  $\pm 1\%$ , según la grafica obtenia en la prueba realizada se obtuvo

una respuesta sobreamortiguada, por consiguiendo la aplicación del controlador al tener una oscilación muy baja cuida y alarga la vida de una valvula proporcional.

- El desarrollo de la aplicación mediante el uso del lenguaje Java permitió un fácil desarrollo de interfaces gráficas, implementación de librerías, y escritura de algoritmos, así como la portabilidad del proyecto entre distintas plataformas y sistemas operativos tanto para su ejecución como para su modificación. Aplicando normas de diseño de HMI como las comprendidas en la guía GEDIS se ha realizado las interfaces para la interacción con el proceso, contando con una fácil visualización, navegación y uso de comandos
- Desarrollar este tipo de prototipos que emulan a un sistema SCADA aplicando normas industriales, permite diseñar de mejor manera posibles aplicaciones en ambientes reales, como es el caso de la producción de vidrio templado, presentándose la posibilidad de servir como precedente para las fabricas del país y la zona 3.

## **5.2. Recomendaciones**

- Utilizar parámetros de diseño basados en estándares para que los procesos cumplan con los requerimientos tecnológicos de la industria. (Guía GEDIS)
- Es importante que el software de desarrollo presente facilidades de modificación, o migración a otras plataformas, en especial, la creación de interfaces gráficas.
- Para investigaciones futuras, se puede considerar la comunicación tanto entre tarjetas de desarrollo y dispositivos industriales.
- Trabajar en el apartado de ciberseguridad, para dotarle al sistema robustez frente a las amenazas de ataque y robo de información.

## BIBLIOGRAFÍA

- [1] G. Lopez de Ayala, “Desarrollo de Ejercicios Didácticos para Autómatas Programables usando Codesys,” Universidad de Sevilla, 2017.
- [2] S. Lluglla, “Evaluación del rendimiento de las máquinas y eficiencia del personal en el proceso de vidrio de línea blanca de la empresa FAIRIS C.A.,” Universidad Técnica de Ambato, 2013.
- [3] V. del Barrio, “Diseño y programación de varias estaciones de trabajo mediante el uso de PLC y softwares específicos,” Universidad de Valladolid, 2017.
- [4] D. García, “Simulación y Diagnóstico de una Instalación Industrial mediante Factory I / O y OPC,” Universidad de Valladolid, 2017.
- [5] M. García, “Sistema SCADA para el proceso de paletizado L4 de envases de cristal en la empresa Cristalería del Ecuador S.A Cridesa de Guayaquil,” Universidad Técnica de Ambato, 2014.
- [6] D. Lopez, “Sistema SCADA para monitoreo y control del aire acondicionado en la sala de recepción satelital (MSA) del Instituto Espacial Ecuatoriano, estación Cotopaxi,” Universidad Técnica de Ambato, 2014.
- [7] D. Sarmiento, “Implementación de los protocolos de comunicación industrial mediante Simatic S7-1200 para el laboratorio de automatización de la Universidad del Azuay,” Universidad del Azuay, 2016.
- [8] M. Miragaia, “Integración vertical de un ERP en el entorno de fabricación virtual,” Universidad de Oporto, 2015.
- [9] E. G. A. Barrientos, *Sistemas de Producción Automatizados*, 1st ed. Madrid: Dextra Editorial, 2014.
- [10] D. Sanchez, *Introducción a la síntesis y programación de automatismos secuenciales*, 1st ed. Cádiz: Servicio de Publicaciones de la Universidad de Cádiz, 2013.
- [11] L. E. Contreras Bravo, J. A. Tristancho Ortiz, and L. F. Vargas Tamayo, “Automatización en nivel de control de planta mediante el uso de herramientas libres y computación,” *Univ. Dist. Francisco José Caldas*, vol. 6, pp. 45–52, 2015.
- [12] L. Contreras and J. Tristancho, “Diseño de guías de laboratorio para desarrollar

- habilidades profesionales en la asignatura Automatización del programa de ingeniería industrial,” *Acad. y Virtualidad*, vol. 8, no. 2, pp. 112–122, 2015.
- [13] M. J. Molina, “Automatización del sistema de control de la máquina empaquetadora de blisters en la farmacéutica PROPHAR,” Universidad Técnica de Ambato, 2015.
- [14] N. Oliva, M. A. Castro Gil, and G. Díaz Orueta, *Redes de Comunicaciones Industriales*, 1st ed. Madrid: UNED - Universidad Nacional de Educación a Distancia, 2013.
- [15] H. Cevallos and J. Gualacio, “Implementación de un sistema de control y monitoreo a distancia en procesos de ensamblaje con robot industrial kawasaki RS003 por medio de la red GSM en el laboratorio de Automatización Industrial de la Facultad de Mecánica,” Escuela Superior Politécnica de Chimborazo, 2017.
- [16] M. Ruiz, “Diseño de un módulo SCADA de enseñanza práctica con comunicacion PROFIBUS y PLC’S S7-1200 para el laboratorio de la Escuela de Ingeniería Electrónica de la UNP,” Universidad Nacional de Piura, 2016.
- [17] L. Chico, “Sistema inalámbrico para control y monitoreo de la planta de tratamiento de agua potable del cantón Baños,” Universidad Técnica de Ambato, 2015.
- [18] E. Pérez López, “SCADA systems in the industrial automation,” *Tecnol. en Marcha*, vol. 28, no. 4, pp. 3–14, 2015.
- [19] P. F. De Carrera, “Diseñador BPM para el sistema IDbox,” Universidad de Cantabria, 2017.
- [20] D. Ortiz, “Sistema automatizado para el control de flujo de trigo en las tolvas de la empresa Molinos Miraflores S.A,” Universidad Técnica de Ambato, 2014.
- [21] J. H. Feria, J. R. Camargo, and E. G. Muskus, “Diseño e implementación de un algoritmo para translación de protocolo entre las normas IEC60870-5-104 y MODBUS TCP / IP,” *Ingenium*, vol. 16, pp. 118–125, 2015.
- [22] Pearson, “Manual del Vidrio Plano,” in *Cámara del vidrio plano y sus manufacturas de la República Argentina*, TOD Produc., Argentina: CAVIPLAN, 2009, pp. 1–96.
- [23] Cms-GlassTechnology, “Brembana futura p.” .
- [24] benetechmachinery, “Glass Washing Machine.” .

- [25] J. Barr, "Glass Basic Properties," in *The Glass Tempering Handbook Understanding the Glass Tempering Process*, p. 6.
- [26] bpsystems, "Glass Toughening Line Control System." .
- [27] LANDGLASS, "LD-A AirStream Flat Glass Tempering Furnace." .
- [28] LANDGLASS, "Gas Heating Tempering Furnaces." .
- [29] P. Ponsa and A. Granollers, "Diseño de pantalla," in *Diseño Industrial*, 2007, pp. 1–10.
- [30] Adroit Technologies, "Situation Awareness - ISA 101 High Performance HMI," 2018. [Online]. Available: <http://www.adroit-europe.com/HpHmi>.

# **ANEXOS**



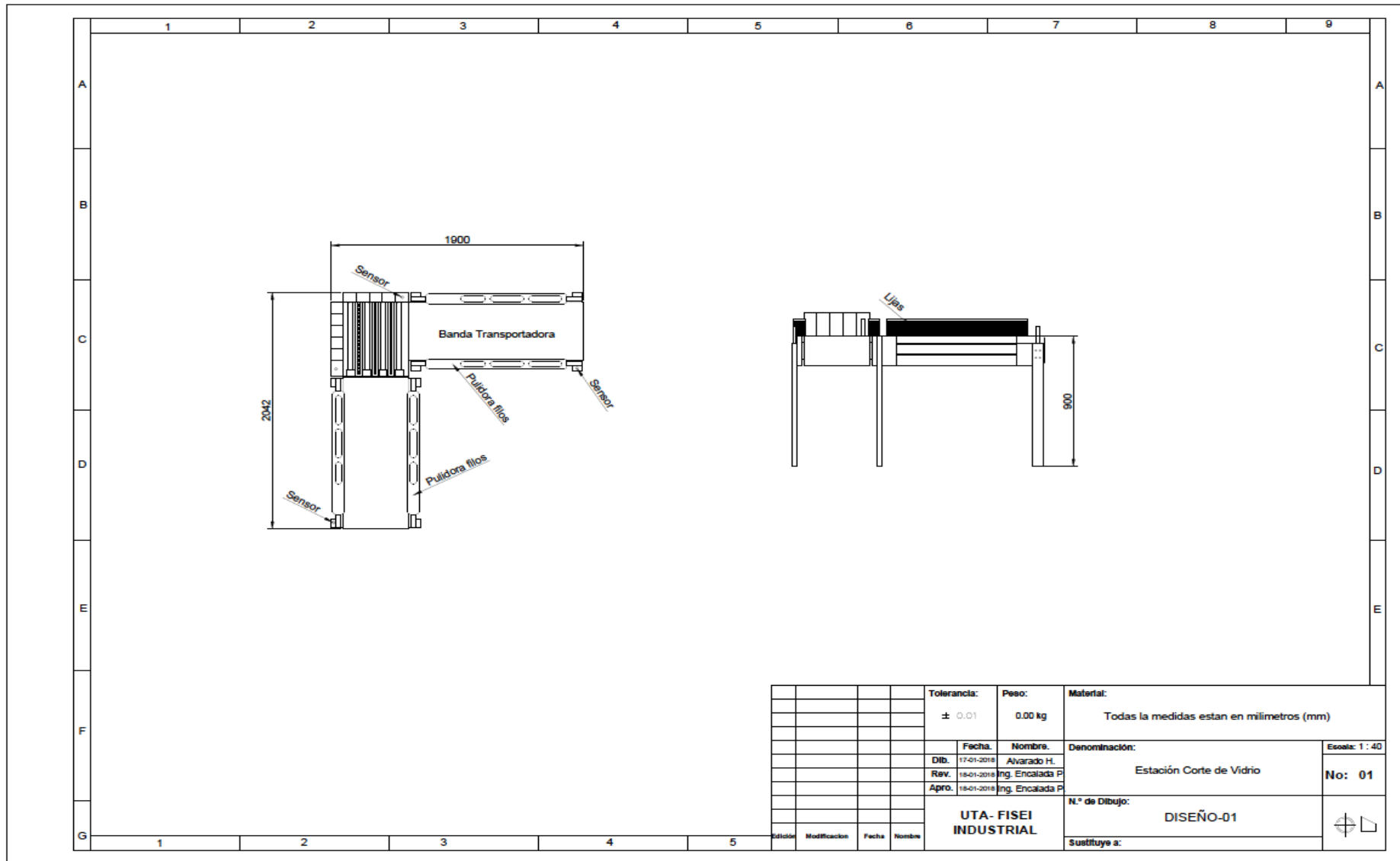
## Anexo A. Métodos disponibles para cliente


Método	Descripción
ModbusClient()	
ModbusClient(string ipAddress, int port)	
void Connect()	Se conecta al servidor Modbus-TCP o al esclavo Modbus-RTU
void Connect(string ipAddress, int port)	
boolean[] ReadDiscreteInputs(int startingAddress, int quantity)	Leer entradas discretas del servidor
boolean[] ReadCoils(int startingAddress, int quantity)	Leer bobinas del servidor
int[] ReadHoldingRegisters(int startingAddress, int quantity)	Leer registros de retención del servidor
int[] ReadInputRegisters(int startingAddress, int quantity)	Leer registros de entrada del servidor
void WriteSingleCoil(int startingAddress, boolean value)	Escribir una única bobina en el servidor
void WriteSingleRegister(int startingAddress, int value)	Escribir un único registro en el servidor
void WriteMultipleCoils(int startingAddress, boolean[] values)	Escribir múltiples bobinas al Servidor
void WriteMultipleRegisters(int startingAddress, int[] values)	Escribir múltiples registros en el servidor
int[] ReadWriteMultipleRegisters(int startingAddressRead, int quantityRead, int startingAddressWrite, int quantityWrite)	
void Disconnect()	
static float ConvertRegistersToFloat(int[] registers)	

static float ConvertRegistersToFloat(int[] registers, RegisterOrder registerOrder)	
static Int ConvertRegistersToDouble(int[] registers)	
static float ConvertRegistersToDouble(int[] registers, RegisterOrder registerOrder)	
static int[] ConvertFloatToTwoRegisters(float floatValue)	
static int[] ConvertFloatToTwoRegisters(float floatValue, RegisterOrder registerOrder)	
static int[] ConvertDoubleToTwoRegisters(Int32 doubleValue)	
static int[] ConvertDoubleToTwoRegisters(Int32 doubleValue, RegisterOrder registerOrder)	
boolean isConnected()	Devuelve el estado de la conexión TCP. VERDADERO si está conectado, FALSO si está desconectado
String getIpAddress()	
void setIpAddress(String ipAddress)	
int getPort()	
void setPort(int port)	
boolean getUDPFlag()	
void setUDPFlag(boolean udpFlag)	
int getConnectionTimeout()	
void setConnectionTimeout(int connectionTimeout)	
void addReceiveDataChangedListener(ReceiveDataChangedListener toAdd)	
void addSendDataChangedListener(SendDataChangedListener toAdd)	



**Anexo C. Plano de la estación de Pulido**

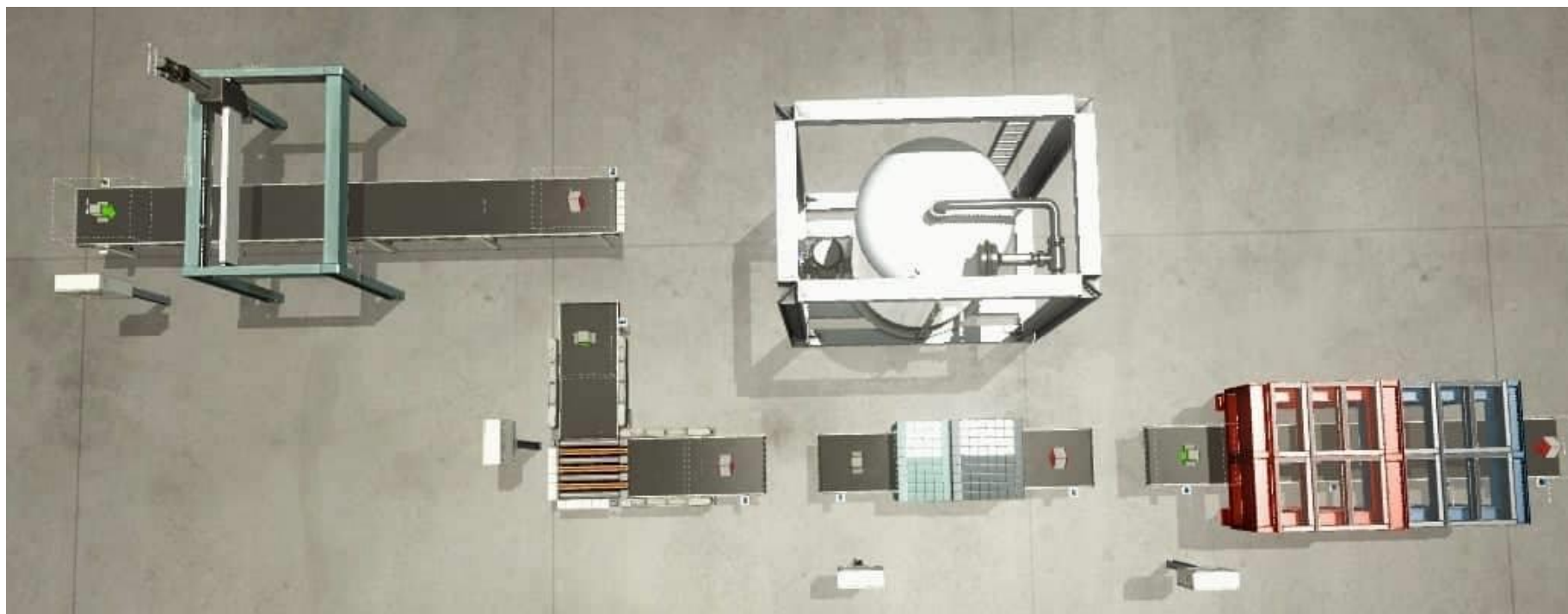


				Tolerancia:	Peso:	Material:	
				± 0.01	0.00 kg	Todas la medidas estan en milímetros (mm)	
						Fecha:	Nombre:
						17-01-2019	Avarado H.
						Rev.	18-01-2019
						18-01-2019	Ing. Encalada P
						Apro.	18-01-2019
						18-01-2019	Ing. Encalada P
						Denominación:	
						Estación Corte de Vidrio	
						N.º de Dibujo:	
						DISEÑO-01	
						Sustituye a:	
						Escala: 1 : 40	
						No: 01	
						UTA- FISEI INDUSTRIAL	
							





**Anexo F.** Réplica realizada en Factory I/O de la línea de producción de vidrio templado.



## Anexo G. Estándares de colores guía GEDIS

### 4. Colores de fondo

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Sinópticos de Área y Subárea		Arena	208/204/191	31/37/188
Detalle Máquinas		Verde Oscuro	120/80/90	64/128/128
Menus y Analógicas		Gris Plata	160/0/224	238/238/238
Tablas en Sinópticos		Gris Plata	160/0/224	238/238/238

### Estatus de Equipos de Proceso

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Equipo Parado		Blanco	160/0/240	255/255/255
Equipo Trabajando		Verde	80/240/53	0/113/0

### Alarmas

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Alarma Crítica		Rojo	0/240/120	255/0/0
Alarma de Advertencia		Amarillo	40/240/120	255/255/0
Mensaje General		Azul Claro	120/240/120	0/255/255

### 5. Materiales de Proceso

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Gas con Material		Amarillo Claro	40/240/180	255/255/128
Aceite		Café Oscuro	80/240/190	148/255/148
Agua		Verde	80/240/53	0/113/0
Aire		Azul Claro	120/240/120	0/255/255

### 6. Señales Analógicas

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Temperatura		Marrón	0/240/46	98/0/0
Presión, Depresión		Azul Rey	160/240/120	0/0/255
Potencia		Violeta	200/240/60	128/0/128
Caudal		Azul Marino	140/240/60	0/64/128
Velocidad		Verde Oscuro	80/240/53	0/113/0
Otras		Negro	160/0/0	0/0/0/

### 7. Items Varios

Ítem	Color	Descripción	Matiz/Sat/Lum	Rojo/Verde/Azul
Código Equipos Normal		Negro	160/0/0	0/0/0/
Botón Confirmar Alarmas		Amarillo Claro	40/240/180	255/255/128
Títulos de Pantallas		Azul Marino	140/240/60	0/64/128
Texto Fallas Críticas		Rojo	0/240/120	255/0/0
Texto Advertencias		Amarillo	40/240/120	255/255/0
Texto General		Azul Marino	140/240/60	0/64/128



## ANEXO H. Programación de la ventana principal del HMI

```
package INTERFAZ;

import CLASES.Reloj_Fecha;
import java.awt.AWTException;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Robot;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Calendar;
import javax.swing.JOptionPane;

/**
 *
 * @author Administrador
 */
public class PANELES extends javax.swing.JFrame implements ActionListener,
    ItemListener{
    public static String usuario="Default";
    Connection c = null;
    Statement stmt = null;
    ResultSet rs =null;
    PreparedStatement pst=null;
    public String txtip;
```

```

private Robot robot;
public IMAGEN_RAIZ p0 =new IMAGEN_RAIZ();
    public CONEXION cx =new CONEXION();
    public PROCESO1 p1 =new PROCESO1();
    public PROCESO2 p2 =new PROCESO2();
    public PROCESO3 p3 =new PROCESO3();
    public PROCESO4 p4 =new PROCESO4();
    public SISTEMA ipr=new SISTEMA();

public PANELES() {
    this.setUndecorated(true);
    this.getContentPane().setBackground(new Color(240,240,240));
    initComponents();
    this.PANEL_P.setBackground(new Color(240,240,240));
    this.Pan1.setBackground(new Color(240,240,240));
    this.Pan2.setBackground(new Color(240,240,240));
    this.Pan3.setBackground(new Color(240,240,240));
    this.PanNav.setBackground(new Color(240,240,240));
    this.PanNav.setVisible(false);
    this.cmd_home.setVisible(false);
    this.cmd_login.setVisible(false);
    this.cmd_logout.setVisible(false);
    this.cmd_alarmas.setVisible(false);
    this.b_exit.setVisible(false);
    c=javaconnect.ConnecrDb();
    this.setLocationRelativeTo(null);
    Reloj_Fecha hilo=new Reloj_Fecha(txtHora,txtFecha);
    hilo.start();
    //Definir ActionListener en los botones de teclado virtual
    tecla_1.addActionListener(this);
    tecla_2.addActionListener(this);
    tecla_3.addActionListener(this);
    tecla_4.addActionListener(this);

```

```
tecla_5.addActionListener(this);
tecla_6.addActionListener(this);
tecla_7.addActionListener(this);
tecla_8.addActionListener(this);
tecla_9.addActionListener(this);
tecla_0.addActionListener(this);
tecla_q.addActionListener(this);
tecla_w.addActionListener(this);
tecla_e.addActionListener(this);
tecla_r.addActionListener(this);
tecla_t.addActionListener(this);
tecla_y.addActionListener(this);
tecla_u.addActionListener(this);
tecla_i.addActionListener(this);
tecla_o.addActionListener(this);
tecla_p.addActionListener(this);
tecla_a.addActionListener(this);
tecla_s.addActionListener(this);
tecla_d.addActionListener(this);
tecla_f.addActionListener(this);
tecla_g.addActionListener(this);
tecla_h.addActionListener(this);
tecla_j.addActionListener(this);
tecla_k.addActionListener(this);
tecla_l.addActionListener(this);
tecla_z.addActionListener(this);
tecla_x.addActionListener(this);
tecla_c.addActionListener(this);
tecla_v.addActionListener(this);
tecla_b.addActionListener(this);
tecla_n.addActionListener(this);
tecla_m.addActionListener(this);
tecla_bsp.addActionListener(this);
```

```

    tecla_izq.addActionListener(this);
    tecla_der.addActionListener(this);
    tecla_esp.addActionListener(this);
    tecla_spr.addActionListener(this);
    tecla_shift.addItemListener(this);
    tecla_shift2.addItemListener(this);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    PANEL_P = new javax.swing.JPanel();
    PAN_HOME = new java.awt.Panel();
    jLabel1 = new javax.swing.JLabel();
    Pan1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    txt_usuario = new javax.swing.JTextField();
    txt_pass = new javax.swing.JPasswordField();
    clean1 = new javax.swing.JButton();
    clean = new javax.swing.JButton();
    Pan2 = new javax.swing.JPanel();
    tecla_1 = new javax.swing.JButton();
    tecla_2 = new javax.swing.JButton();
    tecla_3 = new javax.swing.JButton();
    tecla_4 = new javax.swing.JButton();
    tecla_6 = new javax.swing.JButton();

```

```
tecla_5 = new javax.swing.JButton();
tecla_7 = new javax.swing.JButton();
tecla_8 = new javax.swing.JButton();
tecla_9 = new javax.swing.JButton();
tecla_0 = new javax.swing.JButton();
tecla_p = new javax.swing.JButton();
tecla_o = new javax.swing.JButton();
tecla_i = new javax.swing.JButton();
tecla_u = new javax.swing.JButton();
tecla_y = new javax.swing.JButton();
tecla_t = new javax.swing.JButton();
tecla_r = new javax.swing.JButton();
tecla_e = new javax.swing.JButton();
tecla_w = new javax.swing.JButton();
tecla_q = new javax.swing.JButton();
tecla_a = new javax.swing.JButton();
tecla_s = new javax.swing.JButton();
tecla_shift = new javax.swing.JToggleButton();
tecla_spr = new javax.swing.JButton();
tecla_esp = new javax.swing.JButton();
tecla_z = new javax.swing.JButton();
tecla_x = new javax.swing.JButton();
tecla_d = new javax.swing.JButton();
tecla_f = new javax.swing.JButton();
tecla_c = new javax.swing.JButton();
tecla_v = new javax.swing.JButton();
tecla_g = new javax.swing.JButton();
tecla_h = new javax.swing.JButton();
tecla_b = new javax.swing.JButton();
tecla_n = new javax.swing.JButton();
tecla_j = new javax.swing.JButton();
tecla_k = new javax.swing.JButton();
tecla_l = new javax.swing.JButton();
```

```

tecla_bsp = new javax.swing.JButton();
tecla_enter = new javax.swing.JButton();
tecla_shift2 = new javax.swing.JToggleButton();
tecla_m = new javax.swing.JButton();
tecla_izq = new javax.swing.JButton();
tecla_der = new javax.swing.JButton();
Pan3 = new javax.swing.JPanel();
icono = new javax.swing.JLabel();
estado = new javax.swing.JLabel();
cmd_login = new javax.swing.JButton();
b_exit = new javax.swing.JButton();
cmd_alarmas = new javax.swing.JButton();
PanNav = new javax.swing.JPanel();
b_prin = new javax.swing.JButton();
b_corte = new javax.swing.JButton();
b_pul = new javax.swing.JButton();
b_lav = new javax.swing.JButton();
b_temp = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
txtFecha = new javax.swing.JLabel();
txtHora = new javax.swing.JLabel();
cmd_home = new javax.swing.JButton();
cmd_logout = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
setSize(new java.awt.Dimension(800, 480));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

PANEL_P.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1,
1));
PANEL_P.setPreferredSize(new java.awt.Dimension(687, 400));

```

```

PAN_HOME.setBackground(new java.awt.Color(240, 240, 240));
PAN_HOME.setForeground(new java.awt.Color(255, 255, 255));
PAN_HOME.setMinimumSize(new java.awt.Dimension(542, 211));
PAN_HOME.setPreferredSize(new java.awt.Dimension(542, 362));
PAN_HOME.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Microsoft Sans Serif", 1, 24)); // NOI18N
jLabel1.setText("INICIO DE SESIÓN");
PAN_HOME.add(jLabel1,                                     new
    org.netbeans.lib.awtextra.AbsoluteConstraints(220, 10, -1, -1));

Pan1.setBorder(javax.swing.BorderFactory.createLineBorder(new
    java.awt.Color(0, 0, 0)));

jLabel2.setFont(new java.awt.Font("MS Reference Sans Serif", 0, 14)); //
    NOI18N
jLabel2.setText("Usuario");

jLabel3.setFont(new java.awt.Font("MS Reference Sans Serif", 0, 14)); //
    NOI18N
jLabel3.setText("Contraseña");

clean1.setText("X");
clean1.setFocusable(false);
clean1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        clean1ActionPerformed(evt);
    }
});

clean.setText("X");
clean.setFocusable(false);
clean.addActionListener(new java.awt.event.ActionListener() {

```

```

public void actionPerformed(java.awt.event.ActionEvent evt) {
    cleanActionPerformed(evt);
}
});

javax.swing.GroupLayout Pan1Layout = new javax.swing.GroupLayout(Pan1);
Pan1.setLayout(Pan1Layout);
Pan1Layout.setHorizontalGroup(

    Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(Pan1Layout.createSequentialGroup()
            .addGap(24, 24, 24)

        .addGroup(Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(Pan1Layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(17, 17, 17)
                .addComponent(txt_usuario,
javax.swing.GroupLayout.PREFERRED_SIZE,                204,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(Pan1Layout.createSequentialGroup()
                .addComponent(jLabel3)
                .addGap(19, 19, 19)
                .addComponent(txt_pass,
javax.swing.GroupLayout.PREFERRED_SIZE,                204,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(36, 36, 36)

        .addGroup(Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```



```

        .addComponent(clean1,
javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(clean,
javax.swing.GroupLayout.Alignment.TRAILING))
        .addContainerGap(25, Short.MAX_VALUE))
);
Pan1Layout.setVerticalGroup(

Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
Pan1Layout.createSequentialGroup()
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
.addGroup(Pan1Layout.createSequentialGroup()
.addGap(10, 10, 10)
.addComponent(jLabel2))
.addComponent(txt_usuario,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(clean1, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
.addGroup(Pan1Layout.createSequentialGroup()
.addGap(20, 20, 20)
.addComponent(jLabel3))
.addGroup(Pan1Layout.createSequentialGroup()
.addGap(18, 18, 18)

```

```

        .addComponent(clean,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan1Layout.createSequentialGroup()
        .addGap(20, 20, 20)
        .addComponent(txt_pass,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()
);

```

```

PAN_HOME.add(Pan1,                                new
    org.netbeans.lib.awtextra.AbsoluteConstraints(100, 110, 430, 100));

```

```

Pan2.setBorder(javax.swing.BorderFactory.createLineBorder(new
    java.awt.Color(0, 0, 0)));

```

```

tecla_1.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_1.setText("1");
tecla_1.setFocusable(false);

```

```

tecla_2.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_2.setText("2");
tecla_2.setFocusable(false);

```

```

tecla_3.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_3.setText("3");
tecla_3.setFocusable(false);

```

```

tecla_4.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_4.setText("4");
tecla_4.setFocusable(false);

```

```
tecla_6.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_6.setText("6");
tecla_6.setFocusable(false);
```

```
tecla_5.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_5.setText("5");
tecla_5.setFocusable(false);
```

```
tecla_7.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_7.setText("7");
tecla_7.setFocusable(false);
```

```
tecla_8.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_8.setText("8");
tecla_8.setFocusable(false);
```

```
tecla_9.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_9.setText("9");
tecla_9.setFocusable(false);
```

```
tecla_0.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_0.setText("0");
tecla_0.setFocusable(false);
```

```
tecla_p.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_p.setText("p");
tecla_p.setFocusable(false);
```

```
tecla_o.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_o.setText("o");
tecla_o.setFocusable(false);
```

```
tecla_i.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
```

```
tecla_i.setText("i");  
tecla_i.setFocusable(false);
```

```
tecla_u.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_u.setText("u");  
tecla_u.setFocusable(false);
```

```
tecla_y.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_y.setText("y");  
tecla_y.setFocusable(false);
```

```
tecla_t.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_t.setText("t");  
tecla_t.setFocusable(false);
```

```
tecla_r.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_r.setText("r");  
tecla_r.setFocusable(false);
```

```
tecla_e.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_e.setText("e");  
tecla_e.setFocusable(false);
```

```
tecla_w.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_w.setText("w");  
tecla_w.setFocusable(false);
```

```
tecla_q.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_q.setText("q");  
tecla_q.setFocusable(false);
```

```
tecla_a.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N  
tecla_a.setText("a");
```

```
tecla_a.setFocusable(false);
tecla_a.setMaximumSize(new java.awt.Dimension(39, 39));
tecla_a.setMinimumSize(new java.awt.Dimension(39, 39));
tecla_a.setPreferredSize(new java.awt.Dimension(39, 39));

tecla_s.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_s.setText("s");
tecla_s.setFocusable(false);

tecla_shift.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 11)); // NOI18N
tecla_shift.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/shift.gif"))); //
    NOI18N
tecla_shift.setFocusable(false);

tecla_spr.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_spr.setText("Del");
tecla_spr.setFocusable(false);

tecla_esp.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 11)); // NOI18N
tecla_esp.setFocusable(false);

tecla_z.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_z.setText("z");
tecla_z.setFocusable(false);

tecla_x.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_x.setText("x");
tecla_x.setFocusable(false);

tecla_d.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_d.setText("d");
tecla_d.setFocusable(false);
```

```
tecla_f.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_f.setText("f");
tecla_f.setFocusable(false);
```

```
tecla_c.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_c.setText("c");
tecla_c.setFocusable(false);
```

```
tecla_v.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_v.setText("v");
tecla_v.setFocusable(false);
```

```
tecla_g.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_g.setText("g");
tecla_g.setFocusable(false);
```

```
tecla_h.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_h.setText("h");
tecla_h.setFocusable(false);
```

```
tecla_b.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_b.setText("b");
tecla_b.setFocusable(false);
```

```
tecla_n.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_n.setText("n");
tecla_n.setFocusable(false);
```

```
tecla_j.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_j.setText("j");
tecla_j.setFocusable(false);
```

```

tecla_k.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_k.setText("k");
tecla_k.setFocusable(false);

tecla_l.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_l.setText("l");
tecla_l.setFocusable(false);

tecla_bsp.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 11)); // NOI18N
tecla_bsp.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/backspace.gif")));
    // NOI18N
tecla_bsp.setFocusable(false);

tecla_enter.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 11)); //
    NOI18N
tecla_enter.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/enter.gif"))); //
    NOI18N
tecla_enter.setFocusable(false);
tecla_enter.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tecla_enterActionPerformed(evt);
    }
});

tecla_shift2.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 11)); //
    NOI18N
tecla_shift2.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/shift.gif"))); //
    NOI18N
tecla_shift2.setFocusable(false);

```

```

tecla_m.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_m.setText("m");
tecla_m.setFocusable(false);

tecla_izq.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_izq.setText("<");
tecla_izq.setFocusable(false);

tecla_der.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
tecla_der.setText(">");
tecla_der.setFocusable(false);

javax.swing.GroupLayout Pan2Layout = new javax.swing.GroupLayout(Pan2);
Pan2.setLayout(Pan2Layout);
Pan2Layout.setHorizontalGroup(

    Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(Pan2Layout.createSequentialGroup()

            .addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(Pan2Layout.createSequentialGroup()
                    .addComponent(tecla_1,
javax.swing.GroupLayout.PREFERRED_SIZE,                50,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, 0)
                    .addComponent(tecla_2,
javax.swing.GroupLayout.PREFERRED_SIZE,                50,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, 0, 0)

```



```

        .addComponent(tecla_3,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_4,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_5,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_6,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_7,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_8,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_9,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_0,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()

```

```

        .addComponent(tecla_q,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_w,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_e,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_r,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_t,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_y,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_u,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_i,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)

```

```

        .addComponent(tecla_o,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_p,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()
        .addComponent(tecla_a,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_s,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_d,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_f,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_g,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_h,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)

```

```

        .addComponent(tecla_j,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_k,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_l,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_bsp,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addComponent(tecla_shift,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_spr,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addGroup(Pan2Layout.createSequentialGroup()
        .addComponent(tecla_z,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)

```

```

        .addComponent(tecla_x,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_c,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_v,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_b,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_n,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(tecla_esp,
javax.swing.GroupLayout.PREFERRED_SIZE,           300,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addComponent(tecla_m,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_izq,
javax.swing.GroupLayout.PREFERRED_SIZE,           50,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
    .addComponent(tecla_shift2,
javax.swing.GroupLayout.PREFERRED_SIZE,          50,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(tecla_der,
javax.swing.GroupLayout.PREFERRED_SIZE,          50,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(tecla_enter,
javax.swing.GroupLayout.PREFERRED_SIZE,          50,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGap(0, 0, Short.MAX_VALUE)
);
Pan2Layout.setVerticalGroup(

```

```

Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
Pan2Layout.createSequentialGroup()
    .addGap(0, 0, Short.MAX_VALUE)

```

```

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
    .addComponent(tecla_1,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(tecla_2,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(tecla_3,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(tecla_4,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_5,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_6,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_7,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_8,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_9,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_0,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addComponent(tecla_q,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_w,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_e, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(tecla_r, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_t, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_y,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_u,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_i, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_o,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_p,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addComponent(tecla_a, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_s, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_d,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_f, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_g,
javax.swing.GroupLayout.PREFERRED_SIZE,                30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

        .addComponent(tecla_h,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_j, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_k,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_l, javax.swing.GroupLayout.PREFERRED_SIZE,
30, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_bsp,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addGroup(Pan2Layout.createSequentialGroup()
        .addComponent(tecla_shift,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_spr,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()

.addGroup(Pan2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
        .addComponent(tecla_z,
javax.swing.GroupLayout.PREFERRED_SIZE,          30,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(tecla_x,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_c,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_v,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_b,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(tecla_n,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(tecla_esp,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()
        .addComponent(tecla_m,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(tecla_izq,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(Pan2Layout.createSequentialGroup()
        .addComponent(tecla_shift2,
javax.swing.GroupLayout.PREFERRED_SIZE,           30,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)

```

```

        .addComponent(tecla_der,
javafx.swing.GroupLayout.PREFERRED_SIZE,          30,
javafx.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(tecla_enter,
javafx.swing.GroupLayout.PREFERRED_SIZE,          60,
javafx.swing.GroupLayout.PREFERRED_SIZE)))
);

PAN_HOME.add(Pan2, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
220, 502, -1));

icono.setIcon(new
javafx.swing.ImageIcon(getClass().getResource("/simbolos/Al_az.png")); //
NOI18N

estado.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
estado.setText("Usuario y/o contraseña incorrectos");

javafx.swing.GroupLayout Pan3Layout = new javafx.swing.GroupLayout(Pan3);
Pan3.setLayout(Pan3Layout);
Pan3Layout.setHorizontalGroup(

Pan3Layout.createParallelGroup(javafx.swing.GroupLayout.Alignment.LEA
DING)
.addGroup(Pan3Layout.createSequentialGroup()
.addGap(20, 20, 20)
.addComponent(icono)
.addGap(0, 0, 0)
.addComponent(estado, javafx.swing.GroupLayout.PREFERRED_SIZE,
230, javafx.swing.GroupLayout.PREFERRED_SIZE))
);
Pan3Layout.setVerticalGroup(

```

```

Pan3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
.addGroup(Pan3Layout.createSequentialGroup()
.addGap(10, 10, 10)

.addGroup(Pan3Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
.addComponent(icono, javax.swing.GroupLayout.DEFAULT_SIZE, 30,
Short.MAX_VALUE)
.addComponent(estado, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(10, 10, 10))
);

PAN_HOME.add(Pan3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 50, 280, 50));

javax.swing.GroupLayout PANEL_PLayout = new
javax.swing.GroupLayout(PANEL_P);
PANEL_P.setLayout(PANEL_PLayout);
PANEL_PLayout.setHorizontalGroup(

PANEL_PLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
.addGroup(PANEL_PLayout.createSequentialGroup()
.addComponent(PAN_HOME,
javax.swing.GroupLayout.DEFAULT_SIZE, 698, Short.MAX_VALUE)
.addContainerGap())
);
PANEL_PLayout.setVerticalGroup(

```

```

    PANEL_PLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
    LEADING)
    .addGroup(PANEL_PLayout.createSequentialGroup()
        .addComponent(PAN_HOME,
            javax.swing.GroupLayout.DEFAULT_SIZE, 378, Short.MAX_VALUE)
        .addContainerGap())
    );

    getContentPane().add(PANEL_P,                                     new
        org.netbeans.lib.awtextra.AbsoluteConstraints(90, 0, 710, 390));

    cmd_login.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/simbolos/6.PNG")); //
        NOI18N
    cmd_login.setEnabled(false);
    cmd_login.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cmd_loginActionPerformed(evt);
        }
    });
    getContentPane().add(cmd_login,                                 new
        org.netbeans.lib.awtextra.AbsoluteConstraints(0, 200, 85, 64));

    b_exit.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/simbolos/4.PNG")); //
        NOI18N
    b_exit.setEnabled(false);
    b_exit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            b_exitActionPerformed(evt);
        }
    });

```

```

getContentPane().add(b_exit,                                     new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 340, 85, 64));

cmd_alarmas.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/2.PNG"))); //
    NOI18N
cmd_alarmas.setEnabled(false);
cmd_alarmas.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmd_alarmasActionPerformed(evt);
    }
});
getContentPane().add(cmd_alarmas,                               new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 60, 85, 64));

b_prin.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
b_prin.setText("Principal");
b_prin.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        b_prinActionPerformed(evt);
    }
});

b_corte.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
b_corte.setText("Corte");
b_corte.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        b_corteActionPerformed(evt);
    }
});

b_pul.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
b_pul.setText("Pulido");

```

```

b_pul.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        b_pulActionPerformed(evt);
    }
});

```

```

b_lav.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
b_lav.setText("Lavado");
b_lav.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        b_lavActionPerformed(evt);
    }
});

```

```

b_temp.setFont(new java.awt.Font("Microsoft Sans Serif", 0, 14)); // NOI18N
b_temp.setText("Templado");
b_temp.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        b_tempActionPerformed(evt);
    }
});

```

```

javax.swing.GroupLayout      PanNavLayout      =      new
    javax.swing.GroupLayout(PanNav);
PanNav.setLayout(PanNavLayout);
PanNavLayout.setHorizontalGroup(

    PanNavLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            PanNavLayout.createSequentialGroup()
                .addGap(11, Short.MAX_VALUE)

```

```

        .addComponent(b_prin, javax.swing.GroupLayout.PREFERRED_SIZE,
95, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(b_corte, javax.swing.GroupLayout.PREFERRED_SIZE,
95, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(b_pul, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(b_lav, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(b_temp)
        .addContainerGap()
);
PanNavLayout.setVerticalGroup(

PanNavLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
PanNavLayout.createSequentialGroup()
        .addGap(0, 25, Short.MAX_VALUE)

        .addGroup(PanNavLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)

```



```

        .addComponent(b_corte)
        .addComponent(b_prin)
        .addComponent(b_pul)
        .addComponent(b_lav)
        .addComponent(b_temp)))
);

getContentPane().add(PanNav,                                     new
    org.netbeans.lib.awtextra.AbsoluteConstraints(90, 380, 520, 50));

jLabel4.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/uta.png"))); //
    NOI18N
getContentPane().add(jLabel4,                                   new
    org.netbeans.lib.awtextra.AbsoluteConstraints(40, 420, -1, -1));

txtFecha.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
txtFecha.setText("00/00/0000");
getContentPane().add(txtFecha,                                  new
    org.netbeans.lib.awtextra.AbsoluteConstraints(690, 430, -1, 17));

txtHora.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
txtHora.setText("00:00:00");
getContentPane().add(txtHora,                                   new
    org.netbeans.lib.awtextra.AbsoluteConstraints(710, 450, -1, -1));

cmd_home.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/Home_Disabled.p
    ng"))); // NOI18N
cmd_home.setEnabled(false);
cmd_home.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmd_homeActionPerformed(evt);
    }
});

```

```

    }
});
getContentPane().add(cmd_home,                                new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 130, 85, 64));

cmd_logout.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/simbolos/icon_038490_256
    .png"))); // NOI18N
cmd_logout.setBorder(null);
cmd_logout.setEnabled(false);
cmd_logout.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cmd_logoutActionPerformed(evt);
    }
});
getContentPane().add(cmd_logout,                                new
    org.netbeans.lib.awtextra.AbsoluteConstraints(0, 270, 85, 64));

pack();
} // </editor-fold>
//Inicio de sesión de usuarios
private void tecla_enterActionPerformed(java.awt.event.ActionEvent evt) {
    String sql = "select * from Usuarios Where username=? and clave=?";
    IMAGEN_RAIZ p0 =new IMAGEN_RAIZ();
    try{
        pst=c.prepareStatement(sql);
        pst.setString(1, txt_usuario.getText());
        pst.setString(2, txt_pass.getText());
        rs=pst.executeQuery();
        if(rs.next()){
            String acc=rs.getString("acceso");
            usuario=rs.getString("username");
            if( "Administrador".equals(acc)){

```

```

cmd_alarmas.setEnabled(true);
cmd_home.setEnabled(true);
cmd_login.setEnabled(true);
cmd_logout.setEnabled(true);
b_exit.setEnabled(true);
PanNav.setVisible(true);
b_prin.setVisible(false);
this.cmd_home.setVisible(true);
this.cmd_login.setVisible(true);
this.cmd_logout.setVisible(true);
this.cmd_alarmas.setVisible(true);
this.b_exit.setVisible(true);
JOptionPane.showMessageDialog(null, "Sesion Iniciada: Administrador");
icono.setIcon(null);
estado.setText("Sesion iniciada: Administrador");
}
if("Operario".equals(acc)){
cmd_alarmas.setEnabled(true);
cmd_home.setEnabled(true);
cmd_login.setEnabled(true);
cmd_logout.setEnabled(true);
b_exit.setEnabled(false);
this.cmd_home.setVisible(true);
this.cmd_login.setVisible(true);
this.cmd_logout.setVisible(true);
this.cmd_alarmas.setVisible(true);
this.b_exit.setVisible(true);
JOptionPane.showMessageDialog(null, "Sesion Iniciada: Operario");
icono.setIcon(null);
estado.setText("Sesion iniciada: Operario");
}
p0.setSize(687,400);
p0.setLocation(0,0);

```

```

        PANEL_P.removeAll();
        PANEL_P.add(p0, BorderLayout.CENTER);
        PANEL_P.revalidate();
        PANEL_P.repaint();
    }
    else{
        JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrecto");
        icono.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/simbolos/Al_am.png")));
        estado.setText("Usuario y/o contraseña incorrectos");
    }
    pst.close();
    rs.close();
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
}

private void cleanActionPerformed(java.awt.event.ActionEvent evt) {
    txt_usuario.setText("");
}

private void cleanActionPerformed(java.awt.event.ActionEvent evt) {
    txt_pass.setText("");
}

private void cmd_logoutActionPerformed(java.awt.event.ActionEvent evt) {
    cmd_alarmas.setEnabled(false);
    cmd_home.setEnabled(false);
    cmd_login.setEnabled(false);
    cmd_logout.setEnabled(false);
    b_exit.setEnabled(false);
}

```

```

this.PanNav.setVisible(false);
this.cmd_home.setVisible(false);
this.cmd_login.setVisible(false);
this.cmd_logout.setVisible(false);
this.cmd_alarmas.setVisible(false);
this.b_exit.setVisible(false);    PAN_HOME.setSize(687,400);
txt_usuario.setText("");
txt_pass.setText("");
PAN_HOME.setLocation(0,0);
PANEL_P.removeAll();
PANEL_P.add(PAN_HOME, BorderLayout.CENTER);
PANEL_P.revalidate();
PANEL_P.repaint();
JOptionPane.showMessageDialog(null, "Sesion Finalizada");
}

private void cmd_homeActionPerformed(java.awt.event.ActionEvent evt) {
    PAN_HOME.setSize(687,400);
    PAN_HOME.setLocation(0,0);
    PANEL_P.removeAll();
    PANEL_P.add(p0, BorderLayout.CENTER);
    PANEL_P.revalidate();
    PANEL_P.repaint();
}

private void cmd_alarmasActionPerformed(java.awt.event.ActionEvent evt) {
    ALARMAS al=new ALARMAS();
    al.setVisible(true);
}

private void b_exitActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0); //Cerrar sistema
}

```

```
private void cmd_loginActionPerformed(java.awt.event.ActionEvent evt) {  
    PAN_HOME.setSize(687,400);  
    PAN_HOME.setLocation(0,0);  
    PANEL_P.removeAll();  
    PANEL_P.add(PAN_HOME, BorderLayout.CENTER);  
    PANEL_P.revalidate();  
    PANEL_P.repaint();  
}
```

```
private void b_prinActionPerformed(java.awt.event.ActionEvent evt) {  
    IMAGEN_RAIZ p0 =new IMAGEN_RAIZ();  
    b_prin.setVisible(false);  
    p0.setSize(687,400);  
    p0.setLocation(0,0);  
    PANEL_P.removeAll();  
    PANEL_P.add(p0, BorderLayout.CENTER);  
    PANEL_P.revalidate();  
    PANEL_P.repaint();  
    b_corte.setVisible(true);  
    b_pul.setVisible(true);  
    b_lav.setVisible(true);  
    b_temp.setVisible(true);  
}
```

```
private void b_corteActionPerformed(java.awt.event.ActionEvent evt) {  
    PROCESO1 p1 =new PROCESO1();  
    p1.setSize(687,400);  
    p1.setLocation(0,0);  
    PANEL_P.removeAll();  
    PANEL_P.add(p1, BorderLayout.CENTER);  
    PANEL_P.revalidate();  
    PANEL_P.repaint();  
}
```

```

    b_prin.setVisible(true);
    b_pul.setVisible(true);
    b_lav.setVisible(true);
    b_temp.setVisible(true);
    b_corte.setVisible(false);
}

private void b_pulActionPerformed(java.awt.event.ActionEvent evt) {
    PROCESO2 p2 =new PROCESO2();
    p2.setSize(687,400);
    p2.setLocation(0,0);
    PANEL_P.removeAll();
    PANEL_P.add(p2, BorderLayout.CENTER);
    PANEL_P.revalidate();
    PANEL_P.repaint();
    b_prin.setVisible(true);
    b_corte.setVisible(true);
    b_lav.setVisible(true);
    b_temp.setVisible(true);
    b_pul.setVisible(false);
}

private void b_lavActionPerformed(java.awt.event.ActionEvent evt) {
    PROCESO3 p3 =new PROCESO3();
    p3.setSize(687,400);
    p3.setLocation(0,0);
    PANEL_P.removeAll();
    PANEL_P.add(p3, BorderLayout.CENTER);
    PANEL_P.revalidate();
    PANEL_P.repaint();
    b_prin.setVisible(true);
    b_corte.setVisible(true);
}

```

```

    b_pul.setVisible(true);
    b_temp.setVisible(true);
    b_lav.setVisible(false);
}

```

```

private void b_tempActionPerformed(java.awt.event.ActionEvent evt) {
    PROCESO4 p4 =new PROCESO4();
    p4.setSize(687,400);
    p4.setLocation(0,0);
    PANEL_P.removeAll();
    PANEL_P.add(p4, BorderLayout.CENTER);
    PANEL_P.revalidate();
    PANEL_P.repaint();
    b_prin.setVisible(true);
    b_corte.setVisible(true);
    b_pul.setVisible(true);
    b_temp.setVisible(true);
    b_lav.setVisible(false);
    b_prin.setVisible(true);
    b_corte.setVisible(true);
    b_pul.setVisible(true);
    b_lav.setVisible(true);
    b_temp.setVisible(false);
}

```

```

/**
 * @param args the command line arguments
 */
//Definir pulsaciones de teclas mediante teclado virtual y clase java Robot
public void actionPerformed(ActionEvent e) {
    try {
        robot = new Robot();
        if (e.getSource() == tecla_1)

```



```
robot.keyPress(KeyEvent.VK_1);
robot.keyRelease(KeyEvent.VK_1);
if (e.getSource() == tecla_2)
    robot.keyPress(KeyEvent.VK_2);
    robot.keyRelease(KeyEvent.VK_2);
if (e.getSource() == tecla_3)
    robot.keyPress(KeyEvent.VK_3);
    robot.keyRelease(KeyEvent.VK_3);
if (e.getSource() == tecla_4)
    robot.keyPress(KeyEvent.VK_4);
    robot.keyRelease(KeyEvent.VK_4);
if (e.getSource() == tecla_5)
    robot.keyPress(KeyEvent.VK_5);
    robot.keyRelease(KeyEvent.VK_5);
if (e.getSource() == tecla_6)
    robot.keyPress(KeyEvent.VK_6);
    robot.keyRelease(KeyEvent.VK_6);
if (e.getSource() == tecla_7)
    robot.keyPress(KeyEvent.VK_7);
    robot.keyRelease(KeyEvent.VK_7);
if (e.getSource() == tecla_8)
    robot.keyPress(KeyEvent.VK_8);
    robot.keyRelease(KeyEvent.VK_8);
if (e.getSource() == tecla_9)
    robot.keyPress(KeyEvent.VK_9);
    robot.keyRelease(KeyEvent.VK_9);
if (e.getSource() == tecla_0)
    robot.keyPress(KeyEvent.VK_0);
    robot.keyRelease(KeyEvent.VK_0);
if (e.getSource() == tecla_q)
    robot.keyPress(KeyEvent.VK_Q);
    robot.keyRelease(KeyEvent.VK_Q);
if (e.getSource() == tecla_w)
```

```

    robot.keyPress(KeyEvent.VK_W);
    robot.keyRelease(KeyEvent.VK_W);
if (e.getSource() == tecla_e)
    robot.keyPress(KeyEvent.VK_E);
    robot.keyRelease(KeyEvent.VK_E);
if (e.getSource() == tecla_r)
    robot.keyPress(KeyEvent.VK_R);
    robot.keyRelease(KeyEvent.VK_R);
if (e.getSource() == tecla_t)
    robot.keyPress(KeyEvent.VK_T);
    robot.keyRelease(KeyEvent.VK_T);
if (e.getSource() == tecla_y)
    robot.keyPress(KeyEvent.VK_Y);
    robot.keyRelease(KeyEvent.VK_Y);
if (e.getSource() == tecla_u)
    robot.keyPress(KeyEvent.VK_U);
    robot.keyRelease(KeyEvent.VK_U);
if (e.getSource() == tecla_i)
    robot.keyPress(KeyEvent.VK_I);
    robot.keyRelease(KeyEvent.VK_I);
if (e.getSource() == tecla_o)
    robot.keyPress(KeyEvent.VK_O);
    robot.keyRelease(KeyEvent.VK_O);
if (e.getSource() == tecla_p)
    robot.keyPress(KeyEvent.VK_P);
    robot.keyRelease(KeyEvent.VK_P);
if (e.getSource() == tecla_a)
    robot.keyPress(KeyEvent.VK_A);
    robot.keyRelease(KeyEvent.VK_A);
if (e.getSource() == tecla_s)
    robot.keyPress(KeyEvent.VK_S);
    robot.keyRelease(KeyEvent.VK_S);
if (e.getSource() == tecla_d)

```

```
robot.keyPress(KeyEvent.VK_D);
robot.keyRelease(KeyEvent.VK_D);
if (e.getSource() == tecla_f)
    robot.keyPress(KeyEvent.VK_F);
    robot.keyRelease(KeyEvent.VK_F);
if (e.getSource() == tecla_g)
    robot.keyPress(KeyEvent.VK_G);
    robot.keyRelease(KeyEvent.VK_G);
if (e.getSource() == tecla_h)
    robot.keyPress(KeyEvent.VK_H);
    robot.keyRelease(KeyEvent.VK_H);
if (e.getSource() == tecla_j)
    robot.keyPress(KeyEvent.VK_J);
    robot.keyRelease(KeyEvent.VK_J);
if (e.getSource() == tecla_k)
    robot.keyPress(KeyEvent.VK_K);
    robot.keyRelease(KeyEvent.VK_K);
if (e.getSource() == tecla_l)
    robot.keyPress(KeyEvent.VK_L);
    robot.keyRelease(KeyEvent.VK_L);
if (e.getSource() == tecla_z)
    robot.keyPress(KeyEvent.VK_Z);
    robot.keyRelease(KeyEvent.VK_Z);
if (e.getSource() == tecla_x)
    robot.keyPress(KeyEvent.VK_X);
    robot.keyRelease(KeyEvent.VK_X);
if (e.getSource() == tecla_c)
    robot.keyPress(KeyEvent.VK_C);
    robot.keyRelease(KeyEvent.VK_C);
if (e.getSource() == tecla_v)
    robot.keyPress(KeyEvent.VK_V);
    robot.keyRelease(KeyEvent.VK_V);
if (e.getSource() == tecla_b)
```

```

        robot.keyPress(KeyEvent.VK_B);
        robot.keyRelease(KeyEvent.VK_B);
    if (e.getSource() == tecla_n)
        robot.keyPress(KeyEvent.VK_N);
        robot.keyRelease(KeyEvent.VK_N);
    if (e.getSource() == tecla_m)
        robot.keyPress(KeyEvent.VK_M);
        robot.keyRelease(KeyEvent.VK_M);
    if (e.getSource() == tecla_esp)
        robot.keyPress(KeyEvent.VK_SPACE);
        robot.keyRelease(KeyEvent.VK_SPACE);
    if (e.getSource() == tecla_izq)
        robot.keyPress(KeyEvent.VK_LEFT);
        robot.keyRelease(KeyEvent.VK_LEFT);
    if (e.getSource() == tecla_der)
        robot.keyPress(KeyEvent.VK_RIGHT);
        robot.keyRelease(KeyEvent.VK_RIGHT);
    if (e.getSource() == tecla_spr)
        robot.keyPress(KeyEvent.VK_DELETE);
        robot.keyRelease(KeyEvent.VK_DELETE);
    if (e.getSource() == tecla_bsp)
        robot.keyPress(KeyEvent.VK_BACK_SPACE);
        robot.keyRelease(KeyEvent.VK_BACK_SPACE);

    } catch (AWTException e1) {
        e1.printStackTrace();
    }
}

public void itemStateChanged(ItemEvent ev) {
    try {
        robot = new Robot();
        if (ev.getStateChange() == ItemEvent.SELECTED && ev.getSource() ==
            tecla_shift) {

```

```

        convertirMayuscula();
        robot.keyPress(KeyEvent.VK_SHIFT);
    }
else if (ev.getStateChange() == ItemEvent.DESELECTED && ev.getSource()
    == tecla_shift) {
    convertirMinuscula();
    robot.keyRelease(KeyEvent.VK_SHIFT);
}

if (ev.getStateChange() == ItemEvent.SELECTED && ev.getSource() ==
    tecla_shift2) {
    convertirMayuscula();
    robot.keyPress(KeyEvent.VK_SHIFT);
}
else if (ev.getStateChange() == ItemEvent.DESELECTED && ev.getSource()
    == tecla_shift2) {
    convertirMinuscula();
    robot.keyRelease(KeyEvent.VK_SHIFT);
}

} catch (AWTException e1) {
    e1.printStackTrace();
}
}

public void convertirMayuscula() {
    tecla_1.setText("!");
    tecla_2.setText("@");
    tecla_3.setText("#");
    tecla_4.setText("$");
    tecla_5.setText("%");
    tecla_6.setText("^");
    tecla_7.setText("&");
    tecla_8.setText("*");
}

```

```
tecla_9.setText("(");
tecla_0.setText("0");
tecla_q.setText("Q");
tecla_w.setText("W");
tecla_e.setText("E");
tecla_r.setText("R");
tecla_t.setText("T");
tecla_y.setText("Y");
tecla_u.setText("U");
tecla_i.setText("I");
tecla_o.setText("O");
tecla_p.setText("P");
tecla_a.setText("A");
tecla_s.setText("S");
tecla_d.setText("D");
tecla_f.setText("F");
tecla_g.setText("G");
tecla_h.setText("H");
tecla_j.setText("J");
tecla_k.setText("K");
tecla_l.setText("L");
tecla_z.setText("Z");
tecla_x.setText("X");
tecla_c.setText("C");
tecla_v.setText("V");
tecla_b.setText("B");
tecla_n.setText("N");
tecla_m.setText("M");
    }
public void convertirMinuscula() {
    tecla_1.setText("1");
    tecla_2.setText("2");
    tecla_3.setText("3");
```

```
tecla_4.setText("4");
tecla_5.setText("5");
tecla_6.setText("6");
tecla_7.setText("7");
tecla_8.setText("8");
tecla_9.setText("9");
tecla_0.setText("0");
tecla_q.setText("q");
tecla_w.setText("w");
tecla_e.setText("e");
tecla_r.setText("r");
tecla_t.setText("t");
tecla_y.setText("y");
tecla_u.setText("u");
tecla_i.setText("i");
tecla_o.setText("o");
tecla_p.setText("p");
tecla_a.setText("a");
tecla_s.setText("s");
tecla_d.setText("d");
tecla_f.setText("f");
tecla_g.setText("g");
tecla_h.setText("h");
tecla_j.setText("j");
tecla_k.setText("k");
tecla_l.setText("l");
tecla_z.setText("z");
tecla_x.setText("x");
tecla_c.setText("c");
tecla_v.setText("v");
tecla_b.setText("b");
tecla_n.setText("n");
tecla_m.setText("m");
```

```

    }
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
    (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
    and feel.
    *
    * For details see
    * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(PANELES.class.getName()).log(java.util.
        logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(PANELES.class.getName()).log(java.util.
        logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(PANELES.class.getName()).log(java.util.
        logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```



```

        java.util.logging.Logger.getLogger(PANELES.class.getName()).log(java.util.
        logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {

        new PANELES().setVisible(true);
    }
});
}

// Variables declaration - do not modify
public static javax.swing.JPanel PANEL_P;
private java.awt.Panel PAN_HOME;
private javax.swing.JPanel Pan1;
private javax.swing.JPanel Pan2;
private javax.swing.JPanel Pan3;
private javax.swing.JPanel PanNav;
private javax.swing.JButton b_corte;
private javax.swing.JButton b_exit;
private javax.swing.JButton b_lav;
private javax.swing.JButton b_prin;
private javax.swing.JButton b_pul;
private javax.swing.JButton b_temp;
private javax.swing.JButton clean;
private javax.swing.JButton clean1;
public static javax.swing.JButton cmd_alarmas;
private javax.swing.JButton cmd_home;
private javax.swing.JButton cmd_login;

```

```
private javax.swing.JButton cmd_logout;
private javax.swing.JLabel estado;
private javax.swing.JLabel icono;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JButton tecla_0;
private javax.swing.JButton tecla_1;
private javax.swing.JButton tecla_2;
private javax.swing.JButton tecla_3;
private javax.swing.JButton tecla_4;
private javax.swing.JButton tecla_5;
private javax.swing.JButton tecla_6;
private javax.swing.JButton tecla_7;
private javax.swing.JButton tecla_8;
private javax.swing.JButton tecla_9;
private javax.swing.JButton tecla_a;
private javax.swing.JButton tecla_b;
private javax.swing.JButton tecla_bsp;
private javax.swing.JButton tecla_c;
private javax.swing.JButton tecla_d;
private javax.swing.JButton tecla_der;
private javax.swing.JButton tecla_e;
private javax.swing.JButton tecla_enter;
private javax.swing.JButton tecla_esp;
private javax.swing.JButton tecla_f;
private javax.swing.JButton tecla_g;
private javax.swing.JButton tecla_h;
private javax.swing.JButton tecla_i;
private javax.swing.JButton tecla_izq;
private javax.swing.JButton tecla_j;
private javax.swing.JButton tecla_k;
```

```
private javax.swing.JButton tecla_l;  
private javax.swing.JButton tecla_m;  
private javax.swing.JButton tecla_n;  
private javax.swing.JButton tecla_o;  
private javax.swing.JButton tecla_p;  
private javax.swing.JButton tecla_q;  
private javax.swing.JButton tecla_r;  
private javax.swing.JButton tecla_s;  
private javax.swing.JToggleButton tecla_shift;  
private javax.swing.JToggleButton tecla_shift2;  
private javax.swing.JButton tecla_spr;  
private javax.swing.JButton tecla_t;  
private javax.swing.JButton tecla_u;  
private javax.swing.JButton tecla_v;  
private javax.swing.JButton tecla_w;  
private javax.swing.JButton tecla_x;  
private javax.swing.JButton tecla_y;  
private javax.swing.JButton tecla_z;  
private javax.swing.JLabel txtFecha;  
private javax.swing.JLabel txtHora;  
public static javax.swing.JPasswordField txt_pass;  
public static javax.swing.JTextField txt_usuario;  
// End of variables declaration  
}
```