



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES**

**MONITOREO DE UN SISTEMA AUTOMÁTICO PARA LA  
PRODUCCIÓN DE YOGURT EN LA PLANTA DE LÁCTEOS  
TUNSHI – ESPOCH EN TIEMPO REAL MEDIANTE  
COMUNICACIÓN INALÁMBRICA**

**Trabajo de Titulación**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA CONTROL Y REDES  
INDUSTRIALES**

**AUTOR: CARLOS GONZALO PAZMIÑO COLOMA**

**DIRECTOR: ING. EDWIN ALTAMIRANO**

Riobamba – Ecuador

2021

**©2021, Carlos Gonzalo Pazmiño Coloma**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor

Yo, Carlos Gonzalo Pazmiño Coloma declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 19 de abril de 2021



**Carlos Gonzalo Pazmiño Coloma**

**230020543-8**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

El Tribunal del Trabajo de Titulación certifica que: El trabajo de investigación: **MONITOREO DE UN SISTEMA AUTOMÁTICO PARA LA PRODUCCIÓN DE YOGURT EN LA PLANTA DE LÁCTEOS TUNSHI – ESPOCH EN TIEMPO REAL MEDIANTE COMUNICACIÓN INALÁMBRICA**, de responsabilidad del señor **Carlos Gonzalo Pazmiño Coloma**, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

	<b>FIRMA</b>	<b>FECHA</b>
Ing. José Luis Morales <b>PRESIDENTE DEL TRIBUNAL</b>	_____	19-04-2021 _____
Ing. Edwin Altamirano. <b>DIRECTOR DEL TRABAJO DE TITULACIÓN</b>	_____	19-04-2021 _____
Ing. Pablo Lozada <b>MIEMBRO DEL TRIBUNAL</b>	_____	19-04-2021 _____

## **DEDICATORIA**

El presente trabajo de titulación se lo dedico a Dios por darme salud y las fuerzas necesarias para salir adelante en todo momento.

Con mucho cariño a mis amados padres Mireya y Gonzalo quienes han depositado en mí su confianza y con sus palabras de aliento y perseverancia me han ayudado a salir adelante en el logro de esta meta que es uno de los objetivos más grandes de mi vida.

A mi amada novia Dayana por compartir y celebrar alegrías y triunfos juntos, y por acompañarme en los momentos difíciles durante el transcurso de esta etapa.

Carlos.

## **AGRADECIMIENTO**

Me llena de alegría y satisfacción el haber culminado esta valiosa etapa de mi vida como estudiante universitario, por ello expreso mi profundo agradecimiento a mis padres por brindarme su apoyo incondicional en este arduo proceso académico, quienes a pesar de la distancia siempre estuvieron velando por mi bienestar y educación.

A mi novia quien con amor me ha sabido aconsejar, ser mi guía y motivación para seguir luchando cada día hasta cumplir mis sueños y metas.

A mi primo Marco por el constante apoyo brindado en el transcurso de este proceso académico.

A la Escuela Superior Politécnica de Chimborazo, que me permitió recibir una excelente formación académica.

A mis profesores de la Facultad de Informática y Electrónica por su constancia y entrega en impartir los conocimientos adecuados que serán pilar fundamental para el desempeño de mis actividades como profesional.

A mi tutor de tesis, estimado Ing. Edwin Altamirano por su colaboración en el presente trabajo de titulación llegando a culminarse con éxito.

Al Ing. Marco Manzano, Administrador de la Planta de Lácteos Tunshi de la Facultad de Ciencias Pecuarias de la Escuela Superior Politécnica de Chimborazo, por la apertura y apoyo constante en la realización de este trabajo investigativo.

Carlos.

## TABLA DE CONTENIDO

ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE ANEXOS .....	xiv
ÍNDICE DE ABREVIATURAS .....	xv
RESUMEN .....	xvi
ABSTRACT.....	xvii
INTRODUCCIÓN .....	1
<b>I. CAPÍTULO I FUNDAMENTOS TEÓRICOS</b> -----	<b>6</b>
<b>1.1 Automatización Industrial</b> -----	<b>6</b>
<i>1.1.1 Sistema de control</i> -----	<i>6</i>
<b>1.2 Sensores.</b> -----	<b>7</b>
<i>1.2.1 Características de los sensores</i> -----	<i>8</i>
<i>1.2.2 Clasificación de sensores.</i> -----	<i>9</i>
<i>1.2.3 Sensores de temperatura.</i> -----	<i>10</i>
<i>1.2.3.1 Termistores</i> .....	<i>10</i>
<i>1.2.3.2 Termómetros infrarrojos</i> .....	<i>12</i>
<i>1.2.3.3 Termorresistencia</i> .....	<i>12</i>
<b>1.3 Controlador Lógico Programable.</b> -----	<b>14</b>
<i>1.3.1 Tipos de controladores</i> -----	<i>14</i>
<i>1.3.1.1 PLC modulares</i> .....	<i>15</i>
<i>1.3.1.2 PLC de tipo montaje en rack</i> .....	<i>15</i>
<i>1.3.1.3 PLC con panel Operador</i> .....	<i>16</i>
<i>1.3.2 Lenguajes de programación de PLC</i> -----	<i>16</i>
<i>1.3.2.1 Lista de instrucciones</i> .....	<i>17</i>
<i>1.3.3 Diagrama de bloques funcionales</i> -----	<i>18</i>
<i>1.3.4 Texto estructurado</i> -----	<i>19</i>
<i>1.3.5 SFC</i> -----	<i>19</i>
<i>1.3.6 Módulo programable LOGO! 8</i> -----	<i>20</i>
<i>1.3.7 Entorno de programación LOGO Soft Comfort V8.2</i> -----	<i>22</i>
<i>1.3.8 Módulo de ampliación LOGO AM2 RTD</i> -----	<i>22</i>
<i>1.3.9 LOGO TDE</i> -----	<i>23</i>
<b>1.4 Protocolos de comunicación</b> -----	<b>24</b>
<i>1.4.1 MODBUS TCP</i> -----	<i>24</i>
<i>1.4.2 Bluetooth.</i> -----	<i>25</i>

<b>1.5</b>	<b>Interfaz de Usuario-----</b>	<b>26</b>
<i>1.5.1</i>	<i>MIT App Inventor 2-----</i>	<i>27</i>
<i>1.5.2</i>	<i>Entorno de programación MIT App Inventor 2. -----</i>	<i>27</i>
<b>1.6</b>	<b>Productos lácteos.-----</b>	<b>29</b>
<i>1.6.1</i>	<i>Tipos de productos lácteos.-----</i>	<i>29</i>
<i>1.6.2</i>	<i>Yogurt. -----</i>	<i>30</i>
<b>II.</b>	<b>CAPÍTULO II - MARCO METODOLÓGICO -----</b>	<b>31</b>
<b>2.1</b>	<b>Tarjeta de desarrollo Arduino -----</b>	<b>31</b>
<b>2.2</b>	<b>Bluetooth HC-06 -----</b>	<b>32</b>
<b>2.3</b>	<b>Shield Ethernet W5100 -----</b>	<b>32</b>
<b>2.4</b>	<b>Logo 8-----</b>	<b>33</b>
<b>2.5</b>	<b>Diseño de la interfaz gráfica en la aplicación móvil-----</b>	<b>34</b>
<i>2.5.1</i>	<i>Inicio de sesión y entorno gráfico de la aplicación-----</i>	<i>35</i>
<i>2.5.2</i>	<i>Inserción de elementos en la aplicación móvil-----</i>	<i>37</i>
<i>2.5.3</i>	<i>Alineación e identificación de los elementos de la aplicación móvil -----</i>	<i>38</i>
<i>2.5.4</i>	<i>Inserción de componentes no visibles en la aplicación móvil -----</i>	<i>39</i>
<i>2.5.5</i>	<i>Personalización de la aplicación móvil -----</i>	<i>41</i>
<i>2.5.6</i>	<i>Esquema de programación del logo del prototipo -----</i>	<i>45</i>
<i>2.5.7</i>	<i>Comunicación Modbus en el controlador Logo -----</i>	<i>47</i>
<i>2.5.8</i>	<i>Direccionamiento de variables -----</i>	<i>50</i>
<i>2.5.9</i>	<i>Programación Logo-----</i>	<i>51</i>
<i>2.5.10</i>	<i>Programación Arduino -----</i>	<i>54</i>
<i>2.5.11</i>	<i>Comunicación Arduino con la app móvil-----</i>	<i>58</i>
<i>2.5.12</i>	<i>Lectura y visualización de datos enviados por el Arduino -----</i>	<i>61</i>
<i>2.5.13</i>	<i>Transferencia de datos mediante bluetooth desde la aplicación móvil hacia Arduino</i>	<i>62</i>
<i>2.5.14</i>	<i>Visualización de información en app móvil-----</i>	<i>63</i>
<b>III.</b>	<b>CAPÍTULO III - MARCO DE RESULTADOS-----</b>	<b>65</b>
<b>3.1</b>	<b>Medición del tiempo de respuesta del encendido y apagado del prototipo de</b>	
	<b>manera remota.-----</b>	<b>65</b>
<i>3.1.1</i>	<i>Tiempo de respuesta sin obstáculos-----</i>	<i>66</i>
<b>3.2</b>	<b>Medición de la temperatura de la leche -----</b>	<b>68</b>
<b>3.3</b>	<b>Medición del tiempo de respuesta al notificar una alarma -----</b>	<b>71</b>
	<b>CONCLUSIONES.....</b>	<b>72</b>
	<b>RECOMENDACIONES.....</b>	<b>73</b>



**BIBLIOGRAFÍA**  
**ANEXOS**

## ÍNDICE DE TABLAS

<b>Tabla 1 - 2:</b> Dirección de las variables locales del LOGO hacia direcciones Modbus del Arduino. ....	50
<b>Tabla 2 - 2:</b> Variables del LOGO y dirección Modbus asignada .....	54
<b>Tabla 3 - 2:</b> Interpretación de la información enviada desde la aplicación por el Arduino .....	57
<b>Tabla 1 - 3:</b> Toma de medidas del tiempo de repuesta del prototipo en relación a la distancia entre el dispositivo móvil y el equipo .....	65
<b>Tabla 2 - 3:</b> Error de medida real y medida ideal del prototipo .....	67
<b>Tabla 3 - 3:</b> Medidas de temperatura con el sensor pt100 y Fluke .....	69
<b>Tabla 4 - 3:</b> Tiempo de respuesta de notificación .....	71

## ÍNDICE DE FIGURAS

<b>Figura 1 - 1.</b> Estructura de un sistema automatizado.....	7
<b>Figura 2 - 1.</b> Transductor.....	8
<b>Figura 3 - 1.</b> Termocupla.....	10
<b>Figura 4 - 1.</b> Termistor.....	11
<b>Figura 5 - 1.</b> Termómetro infrarrojo.....	12
<b>Figura 6 - 1.</b> Termorresistencia.....	13
<b>Figura 7 - 1.</b> PLCs compactos.....	14
<b>Figura 8 - 1.</b> PLCs modulares.....	15
<b>Figura 9 - 1.</b> PLCs de tipo montaje en rack.....	16
<b>Figura 10 - 1.</b> OPLC Unitronics M-90.....	16
<b>Figura 11 - 1.</b> Diagrama de escalera.....	17
<b>Figura 12 - 1.</b> Lista de instrucciones.....	18
<b>Figura 13 - 1.</b> Diagrama de bloques funcionales.....	19
<b>Figura 14 - 1.</b> Texto estructurado.....	19
<b>Figura 15 - 1.</b> Diagrama Funcional Secuencial (SFC).....	20
<b>Figura 16 - 1.</b> Módulo LOGO! 8.....	21
<b>Figura 17 - 1.</b> Entorno de programación LOGO Soft Comfort V8.2.....	22
<b>Figura 18 - 1.</b> Módulo de ampliación LOGO AM2 RTD.....	23
<b>Figura 19 - 1.</b> LOGO TDE.....	23
<b>Figura 20 - 1.</b> Esquema de encapsulamiento en Modbus/TCP.....	25
<b>Figura 21 - 1.</b> Topología de una red Bluetooth.....	26
<b>Figura 22 - 1.</b> Ventana de diseño App Inventor.....	28
<b>Figura 23 - 1.</b> Ventana de bloques App Inventor.....	28
<b>Figura 1 - 2.</b> Arduino Uno.....	31
<b>Figura 2 - 2.</b> Bluetooth HC - 06.....	32
<b>Figura 3 - 2.</b> Shield Ethernet W5100.....	33
<b>Figura 4 - 2.</b> Logo 8.....	34
<b>Figura 5 - 2.</b> Página de inicio de App Inventor.....	35
<b>Figura 6 - 2.</b> Inicio de sesión de Google.....	35
<b>Figura 7 - 2.</b> Menú desplazado de “My Projects”.....	36
<b>Figura 8 - 2.</b> Pantalla principal del nuevo Proyecto.....	36
<b>Figura 9 - 2.</b> Inserción de layouts en la aplicación para insertar los elementos.....	37
<b>Figura 10 - 2.</b> Inserción de botones y labels para la conexión bluetooth.....	37
<b>Figura 11 - 2.</b> Preparación de labels para los datos de temperatura y tiempo.....	38

<b>Figura 12 - 2.</b> Propiedades modificadas de la pantalla.....	39
<b>Figura 13 - 2.</b> Resultado después de las modificaciones en las propiedades. ....	39
<b>Figura 14 - 2.</b> Inserción del componente no visible “BluetoothClient1”.....	40
<b>Figura 15 - 2.</b> Adición del reloj a la aplicación.....	40
<b>Figura 16 - 2.</b> Adición del componente de texto a voz. ....	41
<b>Figura 17 - 2.</b> Inclusión de alerta vibratoria para el usuario. ....	41
<b>Figura 18 - 2.</b> Pantalla para subir archivos de imágenes a la aplicación.....	42
<b>Figura 19 - 2.</b> Archivo disponible para usar en la aplicación.....	42
<b>Figura 20 - 2.</b> Archivo disponible para usar en la aplicación.....	43
<b>Figura 21 - 2.</b> Personalización de botones. ....	43
<b>Figura 22 - 2.</b> Edición del texto de botón “Inicio” color, texto y tamaño. ....	44
<b>Figura 23 - 2.</b> Resultado final de la aplicación.....	44
<b>Figura 24 - 2.</b> Programación del LOGO que controla del prototipo. ....	45
<b>Figura 25 - 2.</b> Bloque de función de la temperatura real.....	45
<b>Figura 26 - 2.</b> Bloques de entrada encargada del inicio y paro. ....	46
<b>Figura 27 - 2.</b> Bloque de temperatura deseada que el sistema necesita. ....	46
<b>Figura 28 - 2.</b> Bloque de función del tiempo transcurrido del proceso. ....	46
<b>Figura 29 - 2.</b> Creación de un proyecto de red en LOGO Soft Comfort. ....	47
<b>Figura 30 - 2.</b> Primera configuración del logo agregado ....	47
<b>Figura 31 - 2.</b> Diagrama en blanco luego de crear el proyecto de LOGO. ....	48
<b>Figura 32 - 2.</b> Pantalla que muestra a los dispositivos del proyecto de LOGO. ....	48
<b>Figura 33 - 2.</b> Pantalla de configuración del dispositivo Modbus.....	48
<b>Figura 34 - 2.</b> Conexión Modbus del LOGO y Arduino en el programa LOGO Soft. ....	49
<b>Figura 35 - 2.</b> Tabla de direccionamiento de memorias Modbus.....	49
<b>Figura 36 - 2.</b> Tabla de direccionamiento variables. ....	50
<b>Figura 37 - 2.</b> Identificación de la ubicación de las variables de red. ....	51
<b>Figura 38 - 2.</b> Identificación de la variable de inicio. ....	51
<b>Figura 39 - 2.</b> Identificación de la variable de paro. ....	52
<b>Figura 40 - 2.</b> Identificación de la variable temperatura actual.....	52
<b>Figura 41 - 2.</b> Identificación de la variable temperatura setpoint.....	52
<b>Figura 42 - 2.</b> Identificación de la variable de tiempo transcurrido. ....	53
<b>Figura 43 - 2.</b> Identificación de la variable de tiempo transcurrido. ....	53
<b>Figura 44 - 2.</b> Inicialización de variables y configuración inicial de red Arduino.....	55
<b>Figura 45 - 2.</b> Inicialización de la comunicación serial y ethernet.....	55
<b>Figura 46 - 2.</b> Código de lectura de información enviada desde la aplicación móvil. ....	56
<b>Figura 47 - 2.</b> Procesamiento de la información obtenida mediante condicionales. ....	56
<b>Figura 48 - 2.</b> Creación de la trama de información enviada desde Arduino a la aplicación.....	58

<b>Figura 49 - 2.</b>	Menú de opciones bluetooth en App Inventor. ....	58
<b>Figura 50 - 2.</b>	Configuración inicial de la comunicación bluetooth.....	59
<b>Figura 51 - 2.</b>	Ubicación y declaración de variables.....	59
<b>Figura 52 - 2.</b>	Ubicación de los bloques que se utilizarán en el programa.....	60
<b>Figura 53 - 2.</b>	Código de lectura y pre procesamiento de datos. ....	61
<b>Figura 54 - 2.</b>	Ubicación de los bloques requerido para procesar caracteres y listas.....	61
<b>Figura 55 - 2.</b>	Visualización de la información en los labels. ....	62
<b>Figura 56 - 2.</b>	Procesamiento de la variable temperatura.....	62
<b>Figura 57 - 2.</b>	Caracteres enviados cuando se pulsan y se sueltan los botones.....	63
<b>Figura 58 - 2.</b>	Disponibilidad de herramientas para notificaciones vibratorias y de voz. ....	63
<b>Figura 59 - 2.</b>	Código del primer caso de notificación.....	64
<b>Figura 60 - 2.</b>	Código del segundo caso de notificación.....	64
<b>Figura 1 - 3.</b>	Comparación app móvil e interfaz del sistema.....	68
<b>Figura 2 - 3.</b>	Toma de medida manual de temperatura .....	68

## **ÍNDICE DE ANEXOS**

**ANEXO A:** ELABORACIÓN DE YOGURT

**ANEXO B:** TOMA DE MEDICIONES

**ANEXO C:** TABLERO DE CONTROL ELÉCTRICO

**ANEXO D:** PROGRAMACIÓN DEL SISTEMA DE MONITOREO

**ANEXO E:** ESQUEMA ELÉCTRICO DEL PROTOTIPO

**ANEXO F:** PÁGINA DE REVISIÓN DEL TRABAJO ESCRITO DE TITULACIÓN.

## ÍNDICE DE ABREVIATURAS

S Segundos

Ms Milisegundos

°C Grados Celsius

RTD Resistance Temperature Detector

## RESUMEN

El objetivo del presente trabajo fue monitorear un Sistema Automático para la producción de yogurt en la Planta de Lácteos Tunshi - ESPOCH en tiempo real mediante comunicación inalámbrica, cuyos elementos principales son: el módulo LOGO V8, el controlador Arduino, el módulo bluetooth y la plataforma Android. La metodología utilizada para el desarrollo del trabajo comprende en adquirir las variables de estado del proceso como temperatura y tiempo mediante el módulo LOGO V8, una vez adquiridas las variables, estas son enviadas al controlador Arduino mediante el protocolo de comunicación Modbus TCP, posteriormente se establece la comunicación entre el Arduino y la aplicación mediante el módulo Bluetooth utilizando protocolo serial. Se implementó un programa de Arduino que permite gestionar y procesar las variables, su función principal es construir una trama con los datos de interés y de ésta manera evitar pérdidas de información durante su transmisión, se diseñó una aplicación móvil utilizando la plataforma MIT App Inventor 2, la cual permite al usuario conectarse mediante bluetooth desde el teléfono al sistema, una vez establecida la comunicación el usuario puede visualizar los datos del proceso: temperatura actual, temperatura de set point y tiempo transcurrido, el control del arranque y parada del proceso se realiza mediante los botones de inicio y paro, además el sistema cuenta con notificaciones de voz para emitir las alertas. Se realizó las pruebas de funcionamiento y se determinó que la velocidad de repuesta es directamente proporcional a la distancia entre el sistema y el usuario, obteniendo niveles óptimos de respuesta hasta los 22 metros de distancia en 0.7 segundos, el mensaje de alerta al llegar al set point en 0.603 segundos y la temperatura de enfriamiento para el proceso de fermentación en la elaboración de yogurt en 0.597 segundos, además se determinó que los obstáculos presentes entre el sistema y el usuario afectan a la velocidad de respuesta.

**Palabras clave:** <PRODUCCIÓN>, <YOGURT>, <LOGO V8 (SOFTWARE)>, <ARDUINO (SOFTWARE)>, <BLUETOOTH>, <MODBUS TCP>, <MIT APP INVENTOR (APLICACIÓN WEB)>.



Firmado electrónicamente por:  
**JHONATAN RODRIGO  
FARREÑO UQUILLAS**



08-04-2021

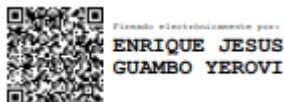
0980-DBRAI-UTP-2021



## ABSTRACT

The aim of this research was to monitor an automatic system for the yogurt production in Tunshi Dairy Plant - ESPOCH in real time through wireless communication, whose main elements are: the LOGO V8 module, the Arduino controller, the bluetooth module and the Android platform. the methodology used for the work development includes acquiring the process condition variables such as temperature and time by means of the LOGO V8 module. Once the variables are acquired, they are sent to the Arduino controller through the Modbus TCP communication protocol. Subsequently, the communication between the arduino and the application is established through the bluetooth module using serial protocol. An Arduino program was implemented to manage and process the variables, its main role is to create a frame with data of interest and thus avoid loss of information during transmission. a mobile application was designed using the MIT App Inventor 2 platform, which allows the user to connect via bluetooth from the phone to the system. once the communication is established, the user can visualize the process data: actual temperature, set point temperature and elapsed time. While the start and stop control of the process is done through the start and stop buttons, moreover, the system has voice notifications to issue alerts. The operating tests were carried out and it was determined that the reaction speed is directly proportional to the distance between the system and the user, obtaining optimal levels of response up to 22 meters away in 0.7 seconds, the alert message when reaching the set point in 0.603 seconds and the cooling temperature for the fermentation process in the production of yogurt in 0.597 seconds. Besides, it was determined that the impediments present between the system and the user affect the speed of response.

Keywords: <PRODUCTION>, <YOGURT>, <LOGO V8 (SOFTWARE) >, <ARDUINO (SOFTWARE) >, <BLUETOOTH>, <MODBUS TCP>, <MIT APP INVENTOR (WEB APPLICATION) >.



## **INTRODUCCIÓN**

El yogurt es un alimento lácteo reconocido como uno de los alimentos más saludables, el yogurt es un producto muy popular por su alto contenido de vitaminas, su producción conlleva una serie de pasos, en donde el control de temperatura es un factor determinante en la calidad del producto final.

Frente a las necesidades productivas de la empresa láctea, se busca implementar sistemas automatizados cuyo objetivo es satisfacer las necesidades en cuanto a calidad y optimización de tiempo. Por esta razón se decide implementar el monitoreo de un sistema automático para la producción de yogurt en la Planta de Lácteos Tunshi – ESPOCH en tiempo real mediante comunicación inalámbrica.

Mediante esta implementación se busca mejorar el proceso productivo mediante una aplicación móvil que le permite al usuario controlar y monitorear el sistema a distancia, evitando de esta manera la necesidad de trasladarse hacia el tablero de control para conocer el estado de las variables del proceso como: temperatura y tiempo. El sistema permite conocer esta información en el dispositivo móvil del operador dentro de un rango de 22 metros.

## ANTECEDENTES

Los derivados lácteos han sido consumidos por el ser humano desde los tiempos de las antiguas tribus nómadas del Neolítico, uno de los principales derivados es el yogurt, el cual remonta su origen a Turquía, Península Balcánica y Asia Central, su nombre proviene del rumano “iaurt”, además se estima que su aparición se dio antes de la agricultura y de forma espontánea por la fermentación en la transportación en bolsas de piel de cabra, donde se convirtió en alimento básico de dichas tribus por su conservación y facilidad de transporte.

Poco a poco se fue extendiendo la obtención y producción de Yogurt, llegó a Norteamérica, posteriormente México, Centroamérica y Sudamérica, hasta llegar a Ecuador en los 90 con una de las primeras franquicias como Parmalat y Toni.

Actualmente, de acuerdo con Euromonitor International, en el año 2012, el volumen de ventas minoristas de yogurt en América Latina ha crecido 6,2%, más rápido que todos los tipos de snacks dulces y salados. En países como Colombia, Ecuador y Perú, presentaron buen ritmo de crecimiento en dicho año, con incrementos en volumen de 8,7%, 5,7% y 5,9% respectivamente.

La Planta de Lácteos Tunshi de la Facultad de Ciencias Pecuarias de la Escuela Superior Politécnica de Chimborazo entre sus actividades de producción, se ha convertido en el principal proveedor de derivados lácteos para consumo estudiantil, docente y demás personal de la institución. Actualmente posee una producción de 120 litros de leche pasteurizada, 25 litros de yogurt y 20 quesos como promedio al día, la misma que es posible en parte gracias a equipos donados por el Gobierno de Japón, y a profesionales y estudiantes que allí laboran.

La innovación y exclusividad es clave para mantener un producto atractivo ante la demanda, características o funciones que faciliten su uso o aplicación dan un “extra o plus” que lo diferencia del resto de la competencia.

Es por esto que se plantea el “Monitoreo de un sistema automático para la producción de yogurt en la planta de lácteos Tunshi – ESPOCH en tiempo real mediante comunicación inalámbrica”, lo cual busca agilizar el proceso, haciéndolo más versátil llevándolo hacia la plataforma de código abierto, basados en Android y Arduino.

## **FORMULACIÓN DEL PROBLEMA**

¿Es viable el Monitoreo de un Sistema Automático para la producción de yogurt en la Planta de Lácteos Tunshi – ESPOCH en tiempo real mediante comunicación inalámbrica?

## **SISTEMATIZACIÓN DEL PROBLEMA**

- ¿Cuáles son las ventajas de usar una plataforma de código abierto como Android y Arduino?
- ¿Cuáles son los dispositivos electrónicos adecuados que permitan acoplarse al código abierto?
- ¿Cuál es la mejor forma de comunicación entre PLC y Arduino?
- ¿Cuáles son las ventajas y desventajas de usar una interfaz gráfica de usuario mediante un estándar de comunicación inalámbrica?

## **JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN**

### **JUSTIFICACIÓN TEÓRICA**

Conociendo que la innovación es fundamental para el desarrollo, se considera necesario que la Planta de Lácteos Tunshi-ESPOCH mejore la operación del sistema de producción de yogurt, haciéndolo más versátil e independiente al no requerir una supervisión por parte del operario en sitio, ya que el sistema poseerá una interfaz inalámbrica en la plataforma Android.

La aplicación dispondrá de monitorización en tiempo real del estado del sistema, tiempo y temperatura principalmente, con capacidad de detener o poner en marcha el mismo desde la comodidad de un smartphone, tablet, o cualquier dispositivo compatible con Android 6.0 o superior.

La comunicación del HMI y el sistema se realizará mediante el estándar bluetooth 4.0, disponible en la mayoría de smartphone actualmente.

El puente de conexión entre smartphone y PLC será el microcontrolador arduino, el cual intercambiará datos constantemente de proceso y comunicación y los enviará mediante bluetooth al smartphone.

La aplicación será desarrollada en MIT app Inventor 2 por ser un entorno de programación visual de diseño y creación de aplicaciones funcionales.

### **JUSTIFICACIÓN APLICATIVA**

La Planta actualmente produce un volumen de 20L de yogurt en una jornada laboral, para ello cuenta con un sistema automático, el cual es manipulado en su totalidad por el operario, todo ello se controla desde un HMI físico instalado en el sistema.

El desarrollo del presente trabajo de titulación pretende mejorar la versatilidad de uso, haciéndolo más cómodo, adaptándolo al mundo de tecnología móvil, desarrollando la comunicación entre un microcontrolador y un Programador Lógico Controlado y evitando la comunicación por cables.

Cabe recalcar que la Planta recibe constantemente la asistencia de estudiantes de la Facultad de Ciencias Pecuarias, donde se realizan prácticas estudiantiles y pre-profesionales de los mismos, donde se investiga y se lleva a cabo diferentes pruebas al producto, ya sea en sabor, consistencia y también agregando otro tipo de químicos al producto con el fin de obtener mejor calidad en el mismo.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Monitorear un Sistema Automático para la producción de yogurt en la Planta de Lácteos Tunshi  
- ESPOCH en tiempo real mediante comunicación inalámbrica.

### **OBJETIVOS ESPECÍFICOS**

- Investigar, recopilar y analizar información y compatibilidad acerca de la plataforma de código abierto Android y Arduino.
- Determinar el sistema de comunicación más adecuado entre Android y Arduino en función de las condiciones de operación del sistema.
- Diseñar una aplicación compatible con smartphone usando la plataforma Android.
- Realizar las pruebas, y correcciones de errores para la puesta a punto del sistema con el fin del mismo.

## **I. CAPÍTULO I FUNDAMENTOS TEÓRICOS**

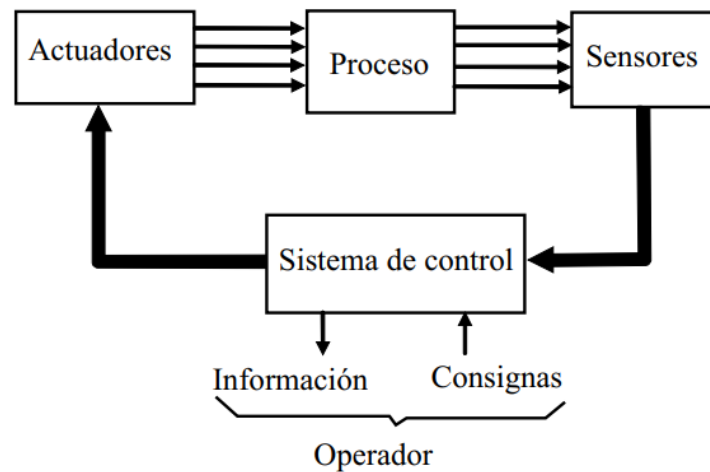
### **1.1 Automatización Industrial**

En los últimos años la tendencia de automatizar los procesos productivos ha sido progresiva. Esta tendencia es posible debido que la tecnología necesaria para su implementación es cada vez más accesible, la automatización de los procesos de producción busca mejorar la calidad y mantener ésta de manera uniforme, producir las cantidades necesarias en el instante preciso, optimizar la productividad, aminorar costos y hacer más flexible el sistema productivo. Estos objetivos se han convertido en requisitos indispensables para mantener la competitividad, por lo que el incremento del nivel de automatización de los procesos es una necesidad para sobrevivir en el mercado actual. (Llopis y Pérez, 2019, p.12)

Se pueden distinguir varios niveles en la automatización de un proceso productivo como; nivel de máquina, donde se considera la automatización de una máquina que desempeña una tarea productiva simple determinada, nivel de célula, en este nivel se considera el control automatizado de un conjunto de máquinas que trabajan de manera coordinada para ejecutar un proceso de producción más complejo, nivel de planta, en este nivel se considera el control automatizado de toda una planta de producción que trabaja de forma coordinada para cumplir objetivos de producción global de la fábrica, nivel de empresa, en este nivel se considera el conjunto de la empresa que corresponde a gestión, ventas y producción.

#### ***1.1.1 Sistema de control***

Se define como un sistema automatizado a la máquina o proceso capaz de reaccionar de manera automática (sin la intervención de un operario) ante los cambios que se producen en el mismo, ejecutando las acciones requeridas para cumplir la función para la que ha sido diseñado. La figura 1-1 muestra la estructura típica de un sistema automatizado.



**Figura 1 - 1.** Estructura de un sistema automatizado.

Fuente: (Llopis, 2019).

Como se puede observar en la figura 1-1 se trata de un sistema en lazo cerrado, donde la información sobre las variaciones ocurridas en el proceso captada por los sensores es procesada para dar lugar a las acciones necesarias, las cuales se ejecutan físicamente sobre el proceso a través de los actuadores. El sistema recibe consignas de funcionamiento por parte del operador como requerimientos de marcha, paro o requerimientos de producción, además el sistema le comunica al operador información sobre el estado del proceso, con fines de supervisión.

Se denomina automatismo al sistema en su conjunto, sin embargo, este término suele hacer referencia esencialmente al sistema de control, ya que este es el que ejecuta de manera automática las acciones sobre el proceso a partir de la información proporcionada por los sensores.

Las señales de entrada y salida pueden ser de cualquier tipo, aunque el concepto convencional de automatismo es usado para sistemas secuenciales o de eventos discretos, en los que se usan señales de tipo binario, es decir solo pueden adquirir valores de cero o uno. Este tipo de control implementa algoritmos de lógica binaria en donde se relacionan los valores que van tomando en cada instante las entradas con los valores que deben tomar las salidas, para que el sistema funcione correctamente.

Los sistemas de control trabajan también con señales de tipo continua para el control de procesos de temperatura, caudal, nivel etc. (Llopis y Pérez, 2019, p.35)

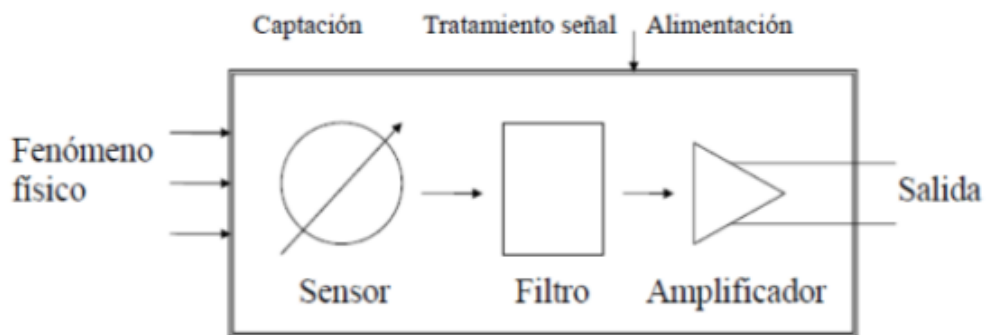
## 1.2 Sensores.

Un sensor o transductor es un dispositivo diseñado para captar un estímulo de su entorno e interpretar esa información que recibe y convertirla a una señal eléctrica, que posteriormente es procesada para generar una acción determinada en un sistema. Las características para tomar en cuenta en un sensor son: precisión, exactitud, sensibilidad, resolución, rango de funcionamiento, fiabilidad, repetitividad y rapidez de respuesta.



En aplicaciones industriales los sensores son una parte muy importante del sistema, ya que permiten garantizar el desarrollo completo del proceso que se lleva a cabo, para seleccionar el sensor adecuado se debe tomar en cuenta el tipo de material que se desea detectar, por ejemplo, si es metálico se usará un sensor inductivo, si es plástico, papel o líquido se utilizará un sensor capacitivo. Otros aspectos a considerar al momento de elegir un sensor son: la distancia operativa, datos electrónicos, forma de la carcasa y conexiones.

Para poder controlar automáticamente un proceso de producción, es indispensable disponer de información sobre el estado del proceso. Esto se puede hacer midiendo las diferentes magnitudes físicas que intervienen en el mismo. Por lo general estas magnitudes no son eléctricas, por lo que se requiere transductores para convertirlas a señales eléctricas. En la figura 2-1, se muestran los componentes de un transductor.



**Figura 2 - 1.** Transductor.

**Fuente:** (<https://bricos.com/2013/02/sensores/>).

Un transductor convierte una señal no eléctrica en una señal eléctrica, cuyos parámetros pueden ser: nivel de tensión, corriente o frecuencia, éstos contienen información sobre la magnitud medida. Debido a la necesidad de acoplar la salida de este dispositivo transductor al sistema de control, suele ser necesario adaptar etapas de filtrado y amplificación de la señal eléctrica en el circuito de acondicionamiento.

### **1.2.1 Características de los sensores**

A continuación, se presentan algunas de las características más importantes que se consideran al elegir un sensor.

**Campo de medida:** es el conjunto de valores comprendido entre los límites superior e inferior de medida, en los cuales el dispositivo es capaz de operar de manera confiable.

- **Sensibilidad:** es la variación que presenta la salida del instrumento por unidad de variación de la variable del proceso, se puede decir que es la ganancia del sensor.
- **Precisión:** es la tolerancia mínima de medida que permitirá registrar el sensor, se puede considerar también como la mínima división de escala de un instrumento.
- **Exactitud:** es la capacidad de un instrumento para presentar valores de error pequeños, esta característica se puede especificar como porcentaje del valor medido.
- **Repetibilidad:** es la máxima desviación que presentan los valores de salida obtenidos al medir por repetidas ocasiones, la misma entrada utilizando un mismo sensor dentro de un entorno con condiciones ambientales idénticas.
- **Histéresis:** se llama histéresis a la desviación presente entre la curva de calibración ascendente y descendente, debido a diferentes niveles de consumo de energía entre la carga y descarga de los instrumentos.

### *1.2.2 Clasificación de sensores.*

Los sensores pueden ser clasificados de acuerdo con un conjunto de características diferentes que no se excluyen entre sí, de la siguiente manera:

Principio físico de funcionamiento: magnéticos, capacitivos, inductivos, termorresistivos, etc.

Aporte de energía: se pueden encontrar sensores activos y pasivos, en los sensores activos como los termoelectrónicos, piezoeléctricos o fotoeléctricos, la magnitud física medida provee la energía requerida para generar la señal de salida, los sensores pasivos como los resistivos, capacitivos e inductivos requieren una fuente de alimentación externa.

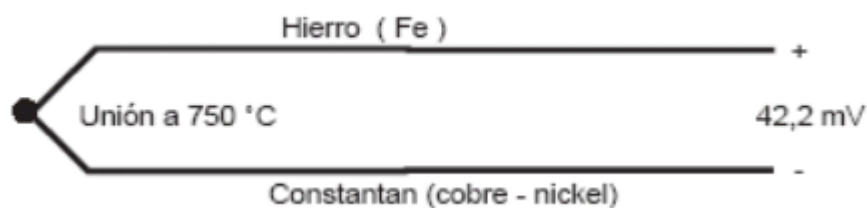
Señal eléctrica generada: pueden ser analógicos o digitales, un sensor analógico puede tomar cualquier valor dentro de un rango determinado, mientras que los sensores digitales pueden tomar únicamente un valor de verdadero o falso.

Magnitud medida: proximidad, nivel, presión, temperatura, caudal, humedad, velocidad, etc.

Rango de valores que proporciona: estos son de medida o de todo/nada. Los sensores de medida brindan a la salida diferentes valores que corresponden a cada uno de los valores de entrada un ejemplo puede ser un sensor termorresistivo, en cambio, en los sensores todo/nada la salida solo puede tomar dos estados diferentes basándose en que la magnitud esté por encima o debajo del umbral un ejemplo de ellos es un sensor de presencia/ausencia. (Fraden, 2004)

### 1.2.3 Sensores de temperatura.

Las termocuplas son los sensores para medir temperatura más usados en el ámbito industrial. Una termocupla está compuesta por dos alambres de distintos materiales unidos por un extremo, al aplicar temperatura en la unión de los materiales, un voltaje muy pequeño en el orden de los milivoltios es generado, el cual incrementa a medida que la temperatura aumenta. La figura 3-1 muestra el esquema de una termocupla común.



**Figura 3 - 1.** Termocupla.

Fuente: (Aragónés J. 2003).

Estos dispositivos suelen estar encapsulados en vainas, debido a que es necesario protegerlos de las condiciones extremas a las que suelen estar expuestos dentro del ambiente industrial, es muy común utilizar materiales como el acero inoxidable para la fabricación de la vaina, quedando por un extremo la unión y por el otro el terminal eléctrico de los cables, con la protección de una caja redonda de aluminio llamada cabezal.

Además, dependiendo de la distancia a la que se encuentre el dispositivo controlador del sensor, se debe usar cables compensados para llevar la señal sin que ésta se vea afectada para su tratamiento, suele darse el caso también en que los materiales utilizados en la termocupla como el platino puro, haga que extender la longitud de los terminales de medición sea inviable debido a los altos costos. (Aragónés, 2003)

#### 1.2.3.1 Termistores

Los termistores son dispositivos mucho más económicos que las RTD, a pesar que no son lineales presentan mayor sensibilidad, están compuestos por una mezcla sintetizada de óxidos metálicos, los termistores son básicamente semiconductores con un comportamiento similar a un resistor térmico, en el mercado se los encuentra con la denominación NTC (Negative Temperature

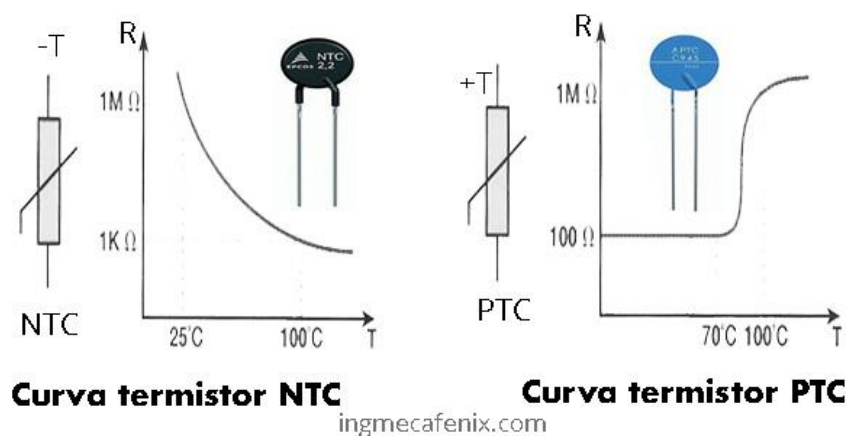
Coefficient), existen casos particulares de coeficiente positivo, cuando la resistencia incrementa con la temperatura y son denominados PTC (Positive Temperature Coefficient) como se puede observar en la figura 4-1.

En ciertos casos, la resistencia de un termistor en condiciones de temperatura ambiente puede sufrir un decremento de hasta seis por ciento por cada grado centígrado de incremento en la temperatura, su alta sensibilidad a las variaciones de temperatura, hacen que éste dispositivo sea bastante apropiado para mediciones de temperatura que requieren precisión, por lo que se lo utiliza con frecuencia en aplicaciones de control en el rango de 150 a 450 grados centígrados.

Los termistores son usados en mediciones de temperatura en sólidos, líquidos e incluso gases, debido a su tamaño reducido se los suele adaptar dentro de sondas o carcasas especiales para lograr una protección y posición adecuada, independientemente del medio en donde se vaya a desempeñar.

A diferencia de las termorresistencias y las termocuplas, el termistor presenta características de exactitud de salida o estabilidad, sin embargo, una de sus principales ventajas es la alta sensibilidad del termistor a las variaciones de temperatura.

Los termistores no son adecuados para medir temperatura dentro de rangos amplios, debido a que la variación de resistencia es muy alta para realizar una medición adecuada con un único instrumento, por lo que los alcances máximos admisibles suelen estar por los 100K, sin embargo, los termistores son especialmente útiles para medir rangos de temperatura reducidos.



**Figura 4 - 1.** Termistor.

Fuente: (<https://www.ingmecafenix.com/automatizacion/termistor-sensor-temperatura/>).

### 1.2.3.2 Termómetros infrarrojos

La característica principal de los termómetros infrarrojos es que pueden medir la temperatura de un objeto sin la necesidad de hacer contacto con el mismo, existen casos en que este tipo de medida sin contacto es crítico cuando el objeto es muy pequeño, no tiene una posición fija, cuando es inaccesible o en procesos dinámicos que necesitan una rápida respuesta.

Los termómetros infrarrojos como se muestra en la figura 5-1, determinan la temperatura que un objeto presenta en su superficie por medio de la medición de la radiación infrarroja que este emite, los rangos típicos de medida suelen estar entre los  $-50^{\circ}\text{C}$  a  $3000^{\circ}\text{C}$ , las distancias de trabajo suelen estar comprendidas entre una fracción de centímetro a varios kilómetros para aplicaciones aerotransportadas.

El principio de funcionamiento en el cual se basan los pirómetros de radiación infrarroja es que todos los cuerpos emiten radiación a longitudes de onda ubicadas en la zona infrarroja del espectro de radiación electromagnética. Los termómetros infrarrojos miden esta radiación y entregan una señal de salida calibrada en una variedad de rangos según los requisitos del proceso. (Aragóns, 2003)



**Figura 5 - 1.** Termómetro infrarrojo.

Fuente: (Fluke, 2020).

### 1.2.3.3 Termorresistencia

La termorresistencia opera bajo el principio en que a medida que varía la temperatura, su resistencia se ve modificada, y la medida de esta modificación se vincula con la variación de

temperatura. Poseen elementos sensitivos contruidos a base de conductores metálicos, que modifican su resistencia eléctrica en base a la temperatura. Esta variación en la resistencia, puede ser medida utilizando un circuito eléctrico, un componente sensitivo, una fuente de voltaje auxiliar y un dispositivo de medida.

Los dispositivos RTD más utilizados están compuestos por una resistencia de platino (Pt), denominadas también PRTD, es el elemento más estable y exacto. La relación resistencia - temperatura, correspondiente al filamento de platino es altamente reproducible, por lo que la termorresistencia fabricada a partir de este material es usado como estándar internacional de temperaturas entre los  $-260\text{ }^{\circ}\text{C}$  y los  $630\text{ }^{\circ}\text{C}$ .

Otros materiales como níquel, cobre y tungsteno también son utilizados, poseen típicamente una resistencia entre los 20 y 20000 ohmios, una de sus ventajas más importante es que sus curvas de respuesta presentan linealidad en temperaturas los  $-200\text{ }^{\circ}\text{C}$  y  $850\text{ }^{\circ}\text{C}$ .



**Figura 6 - 1.** Termorresistencia.

Fuente: (Aragónés, 2003).

Las termorresistencias industriales compuestas de platino como se muestra en la figura 6-1, que más se utilizan son calibradas con una resistencia de 100 ohmios a  $0\text{ }^{\circ}\text{C}$ , las termorresistencias compuestas a base de níquel no permiten medir temperaturas tan altas como las de platino. Los rangos de medida en las resistencias de níquel entre aproximadamente los  $-60\text{ }^{\circ}\text{C}$  y  $180\text{ }^{\circ}\text{C}$ , para los modelos estandarizados. Poseen exactitudes menores a las de platino, comúnmente se calibran a 100 ohmios en  $0^{\circ}\text{C}$ , existiendo otras calibraciones particulares.

Sin embargo, una instalación adecuada necesita cables de conexión blindados tanto para termocuplas como para termorresistencias, las interconexiones entre termorresistencias y demás instrumentos se realizan con cable común de cobre, a diferencia de las termocuplas que requieren

ser conectadas únicamente con cables especiales de compensación, implicando un costo superior. (Aragonés, 2003)

### 1.3 Controlador Lógico Programable.

Un controlador lógico programable (PLC) es una máquina programable diseñada para ser usada en un entorno industrial generalmente hostil, utiliza una memoria configurable para el almacenamiento interno de instrucciones dirigidas al usuario, para implementar soluciones particulares, como: funciones lógicas, secuencias, funciones de temporización, conteo, comparación y funciones aritméticas, con el objetivo de controlar por medio de entradas y salidas, digitales y analógicas, maquinarias o procesos. (Tiegelkamp y Jhon, 1995)

#### 1.3.1 Tipos de controladores

Los PLC compactos como se muestra en la figura 7-1 incorporan la Unidad Central de Procesamiento (CPU), la fuente de alimentación, módulos de entrada y salida en un único dispositivo, por lo general disponen de un número fijo de entradas/salidas (E/S) digitales no mayores a 30, uno o dos canales de comunicación para programar el PLC, para conectar los buses de campo y para adicionar una HMI.

Además, pueden existir entradas específicas para contadores de alta velocidad y una o dos E/S de tipo analógicas. Si se desea incrementar el número de E/S en un PLC compacto se puede conectar módulos adicionales. Los PLCs de tipo compacto se utilizan comúnmente en procesos de automatización como sustitutos de los relés.



**Figura 7 - 1.** PLCs compactos.

**Fuente:** (UNED, 2009).

### 1.3.1.1 *PLC modulares*

Los PLCs modulares como se muestran en la figura 8-1, son más potentes y poseen más funciones que los PLCs compactos, la CPU, la fuente de alimentación, el módulo de comunicaciones y demás módulos se encuentran comúnmente en dispositivos separados dispuestos en un riel de manera especial y se comunican con la CPU mediante un sistema de bus.

El número de espacios para los módulos son limitados, pero es posible aumentarlos en la mayoría de los casos. Además, los PLCs modulares pueden soportar una gran cantidad de entradas/salidas, pueden manejar programas más grandes, guardar más datos y ejecutar modos multitarea. Comúnmente se usan para tareas de control, posicionamiento, regulación, manipulación, monitorización, comunicación, procesamiento de datos, servicios-web, etc.



**Figura 8 - 1.** PLCs modulares.

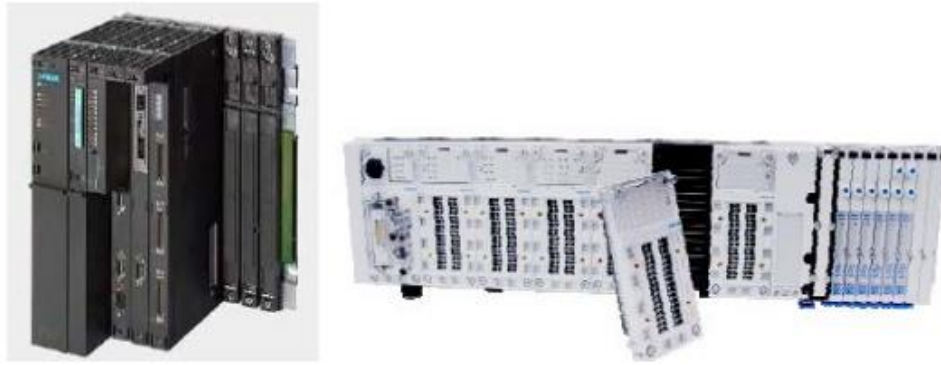
**Fuente:** (UNED, 2009).

### 1.3.1.2 *PLC de tipo montaje en rack*

Los PLCs de montaje en rack como se muestran en la figura 9-1, tienen prácticamente las mismas capacidades y funciones que los PLCs modulares, sin embargo, presentan ciertas diferencias en el bus o rack donde se ubican los módulos del PLC. El rack cuenta con ranuras para los módulos y un sistema de bus integrado para enviar y recibir información entre los diferentes módulos.

La mayoría de los módulos PLC no poseen sus propias carcasas, disponen únicamente de un panel ubicado en la parte frontal con una interfaz hombre - máquina (HMI). La principal ventaja es que la comunicación de datos entre los módulos es más veloz y el tiempo de respuesta es menor.





**Figura 9 - 1.** PLCs de tipo montaje en rack.

Fuente: (UNED, 2009).

### 1.3.1.3 *PLC con panel Operador*

El PLC con panel operador denominado OPLC posee una interfaz HMI para su funcionamiento y monitorización de las máquinas y los procesos automático, la HMI consiste principalmente en un monitor con un teclado o una pantalla táctil, el monitor puede ser de tipo gráfico o de texto como se muestra en la figura 10-1. La ventaja principal de este PLC respecto a otros es que no se requiere programar el panel de manera separada. Toda la programación se desarrolla mediante una herramienta software, lo que permite reducir los gastos del desarrollo del sistema.



**Figura 10 - 1.** OPLC Unitronics M-90.

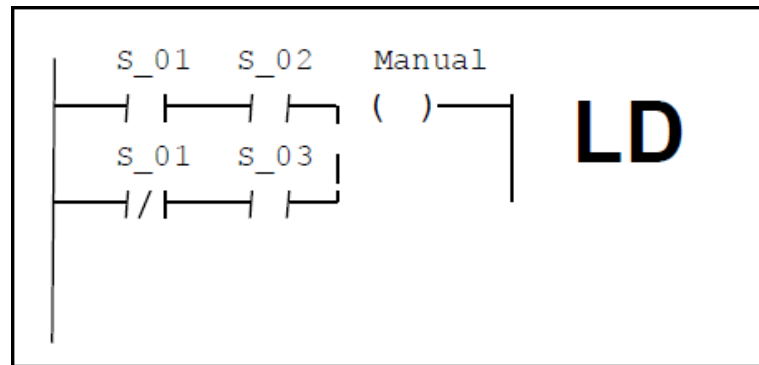
Fuente: (UNED, 2009).

### 1.3.2 *Lenguajes de programación de PLC*

Con la aparición de los primeros PLCs en el año 1969, el lenguaje de programación utilizado en esa época era el llamado diagrama de escalera (Ladder Diagram –LD-) como se muestra en la

figura 11-1, muy cercano estructuralmente a los diagramas eléctricos utilizados para representar los esquemas de lógica cableada usados para controlar la mayoría de procesos en aquel entonces.

Se tenía la intención de que la transición de una tecnología a otra fuese lo más sutil posible, de tal manera que el personal de una industria con una alta cualificación en sistemas de lógica cableada, puedan ser sencillamente formados para manejar los nuevos sistemas basados en lógica programable, permitiendo a las empresas disminuir los costos durante la transición hacia un nuevo modelo en la automatización de procesos.



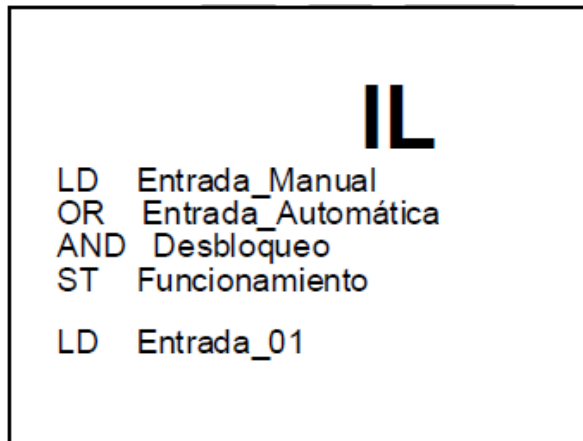
**Figura 11 - 1.** Diagrama de escalera.

**Fuente:** (Mateos, 2004).

### 1.3.2.1 Lista de instrucciones

El lenguaje de lista de instrucciones (Instruction List -IL-) es el lenguaje de programación de PLCs más versátiles de los lenguajes existentes. Es un lenguaje de comandos literales, de bajo nivel parecido al lenguaje ensamblador utilizado para la programación de microcontroladores.

Es un lenguaje originado en Alemania gracias a Siemens, desarrollado para sus primeros autómatas. Este lenguaje se basa en el uso de un mnemónico que representa la instrucción seguido del operando u operandos sobre los que se aplica. El resultado de la operación puede ser guardado sobre uno de los operandos o sobre alguno de los registros o la batería que emplea el equipo. Cada línea del programa posee una única instrucción y su ejecución se realiza de manera secuencial empezando por la primera línea de la lista como se puede observar en la figura 12-1. Cualquier programa escrito en otros lenguajes puede finalmente ser traducido a IL.



**Figura 12 - 1.** Lista de instrucciones.

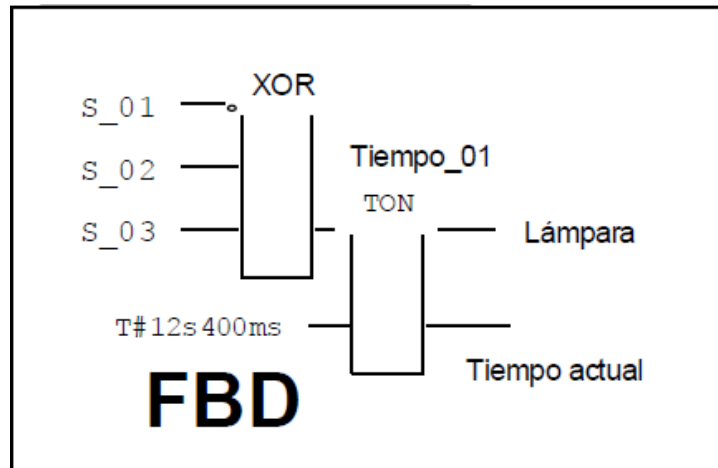
Fuente: (Mateos, 2004).

### ***1.3.3 Diagrama de bloques funcionales***

El lenguaje de los diagramas de bloques funcionales (Function Block Diagram -FBD-) es un lenguaje gráfico que aparece como una evolución de los diagramas utilizados por los ingenieros electrónicos para caracterizar los circuitos lógicos. En estos circuitos, las puertas lógicas se representan por medio del uso de símbolos estandarizados.

Un circuito electrónico puede ser almacenado en un elemento integrado el cual puede ser usado posteriormente en otro diagrama utilizando un símbolo adecuado. El lenguaje FBD como se muestra en la figura 13-1, obedece las mismas condiciones, cada operación se representa mediante un símbolo normalizado.

Un conjunto de operaciones que desempeñan una función específica pueden ser agrupadas con el fin de formar una subrutina o bloque el cual se identifica con un símbolo asociado, que posteriormente puede ser usado en otro segmento de programa tantas veces como se necesite. Sólo se requiere tener en cuenta las pautas sintácticas y semánticas a la hora de elaborar los diagramas.

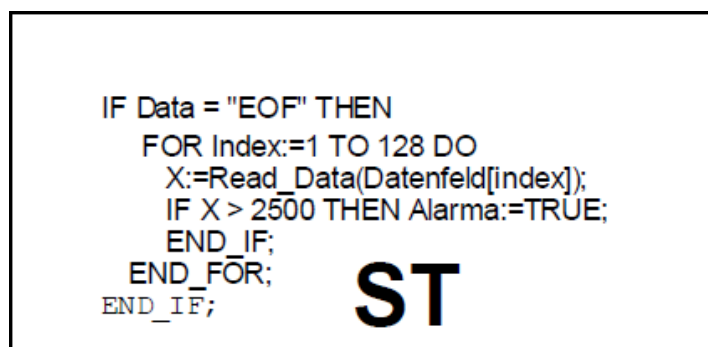


**Figura 13 - 1.** Diagrama de bloques funcionales.

Fuente: (Mateos, 2004).

### 1.3.4 Texto estructurado

El texto estructurado (Structured Text -ST-) es un lenguaje de comandos literales de alto nivel que aparece al adaptar el lenguaje Pascal utilizado en la programación de computadores a las necesidades inherentes al control de procesos. Como se muestra en la figura 14-1, es un lenguaje altamente potente, desarrollado especialmente para la representación de algoritmos de control más complejos en los cuales se requiere el uso de bucles, estructuras condicionales, funciones almacenadas, etc.



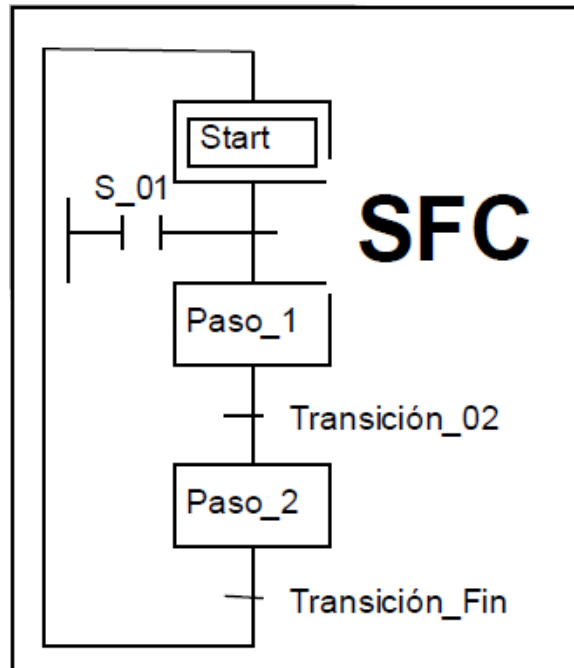
**Figura 14 - 1.** Texto estructurado.

Fuente: (Mateos, 2004).

### 1.3.5 SFC

El lenguaje de los diagramas funcionales secuenciales (Sequential Function Chart - SFC) aparece como una evolución del lenguaje de modelado de sistemas secuenciales GRAFCET (estándar

internacional IEC 848). Se debe aclarar que este último no es un lenguaje de programación (a pesar de que sea posible obtener un programa de los diagramas descritos en este tipo de lenguaje), por otro lado, SFC como se muestra en la figura 15-1, es un lenguaje de programación con todas las connotaciones que ello supone. SFC proporciona al desarrollador un potente lenguaje para la representación de automatismos secuenciales. (Mateos, 2004)



**Figura 15 - 1.** Diagrama Funcional Secuencial (SFC).

Fuente: (Mateos, 2004).

### ***1.3.6 Módulo programable LOGO! 8***

LOGO! 8 es un controlador compacto que se compone de un módulo básico y módulos de expansión que se añaden según el requerimiento de la aplicación. Este diseño modular lo hace una solución flexible, económica y sencilla de usar para tareas de procesos simples. Tiene un diseño robusto y compacto, lo que lo convierte en un dispositivo universal tanto para aplicaciones industriales como para aplicaciones de domótica. Este equipo cuenta con una amplia gama de módulos lógicos básicos, que permiten la implementación de soluciones de automatización básicas.

El módulo lógico LOGO! 8 que se observa en la figura 16-1, posee un puerto de comunicación Ethernet integrado que permite la comunicación, este puerto le permite comunicarse hasta con 8 dispositivos Ethernet adicionales como: módulos lógicos LOGO!, Simatic S7-1200, paneles Simatic HMI, etc. A partir del firmware 4, se incorpora el protocolo de comunicación Modbus

TCP/IP en su puerto integrado Ethernet, esta funcionalidad, permite su comunicación con variadores de velocidad o medidores de energía que incorporan este protocolo. Otra de las funcionalidades es el servidor web incorporado el cual permite realizar tareas de visualización y control desde un web browser. Este dispositivo puede ser maestro de hasta 8 módulos configurados como esclavos.

El módulo cuenta con una ranura para Micro-SD Card, que permite archivar datos de proceso (Datalogging), así como respaldar el programa de usuario. Una funcionalidad bastante útil, es que admite cambiar el color de fondo del display, esto permite alertar al operador sobre algún aviso o condición peligrosa de funcionamiento en el proceso, ¡esta funcionalidad se configura directamente en el software LOGO! Soft Comfort V8.

Uno de los módulos que pueden ser añadidos al dispositivo es el módulo de comunicación GSM/GPRS CMR2020, el cual permite enviar mensajes SMS de estado y alarma desde el dispositivo a un teléfono celular. Asimismo, es posible enviar mensajes SMS desde el celular para que el controlador ejecute comandos preestablecidos.

El módulo LOGO! 8 al tener una mayor velocidad y capacidad de procesamiento, no es compatible con los módulos de expansión LOGO! de versiones anteriores. Para la programación del LOGO! 8, se necesita el software LOGO! Soft Comfort V8. (Siemens, 2020)



**Figura 16 - 1.** Módulo LOGO! 8.

**Fuente:** (Siemens, 2018).

### 1.3.7 Entorno de programación LOGO Soft Comfort V8.2

LOGO! Soft Comfort V8 como muestra la figura 17-1, es un software de configuración y programación para el módulo LOGO!, este software permite la simulación del programa y la verificación de datos en línea. Permite la programación de todas las generaciones de LOGO!.

Este entorno proporciona documentación especializada con información útil durante el desarrollo del proyecto como programas de conmutación, comentarios y ajustes de parámetros. Además, se puede acceder a asistencia en línea relacionada con el contexto que incluye detalles sobre los bloques de función individuales y una explicación de los parámetros y las líneas de tiempo.

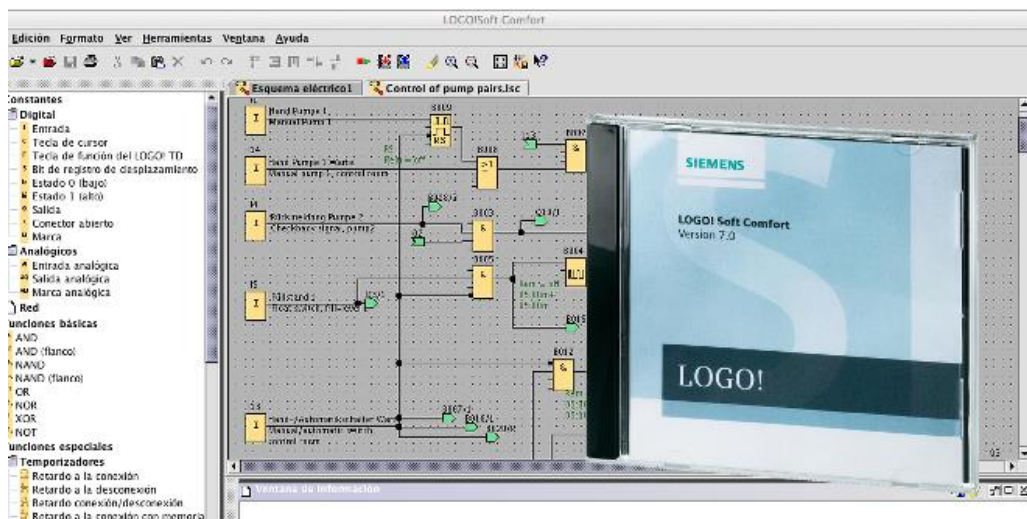


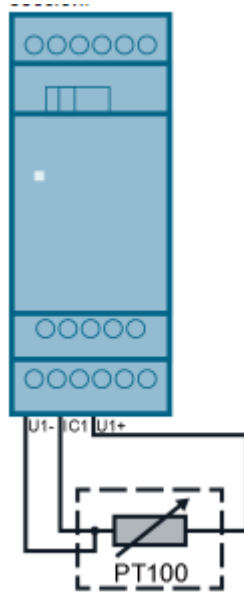
Figura 17 - 1. Entorno de programación LOGO Soft Comfort V8.2

Fuente: (Siemens, 2018).

### 1.3.8 Módulo de ampliación LOGO AM2 RTD

AM2 RTD es un módulo de expansión de LOGO con grado de protección IP20 tiene un suministro de tensión de 12/24 V DC. Posee dos entradas analógicas que pueden ser utilizadas para conectar sondas de temperatura Pt100/Pt1000 para la medición de temperatura en un rango de -50 a 200 °C. Este módulo puede ser implementado en sistemas de calefacción, ventilación y aire acondicionado.

Este módulo permite la conexión de sensores de temperatura a 2, 3 o 4 hilos, en la figura 18-1, se describe una de las configuraciones más comunes que es la conexión a 3 hilos, en este modo de conexión se compensa la resistencia de los cables de medida y no influye en el resultado de la medida. Para que este resultado sea correcto, todos los cables deben tener la misma resistencia. Para lo cual, los cables necesitan tener la misma longitud y sección.



**Figura 18 - 1.** Módulo de ampliación LOGO AM2 RTD

**Fuente:** (Siemens, 2018).

### 1.3.9 LOGO TDE

Logo TDE como se muestra en la figura 19-1, es la pantalla de texto externo de LOGO! 8, la cual permite visualizar 6 caracteres por cada 20 líneas. Debido a que LOGO! 8 puede utilizar textos de mensajes más grandes, este puede actualizar sus textos reduciendo las abreviaturas y también implementar nuevos símbolos de gran tamaño. Por lo tanto, durante la migración deberá optimizar sus mensajes.



**Figura 19 - 1.** LOGO TDE.

**Fuente:** (Siemens, 2018).



## **1.4 Protocolos de comunicación**

Un protocolo de comunicación industrial es un conjunto de reglas que permiten el intercambio de datos entre varios dispositivos dentro de una red. A medida que la tecnología avanza, los protocolos de comunicación deben tener características particulares que respondan a las necesidades de transmisión de datos en tiempo real. Los protocolos usados en la industria hoy en día, surgen de la evolución de los primeros protocolos basados en comunicación serial y de la creación de nuevos estándares que se basan en las nuevas tecnologías, como ejemplo de evolución de los protocolos antiguos cabe mencionar a DNP3, Modbus/TCP, Profinet, etc.

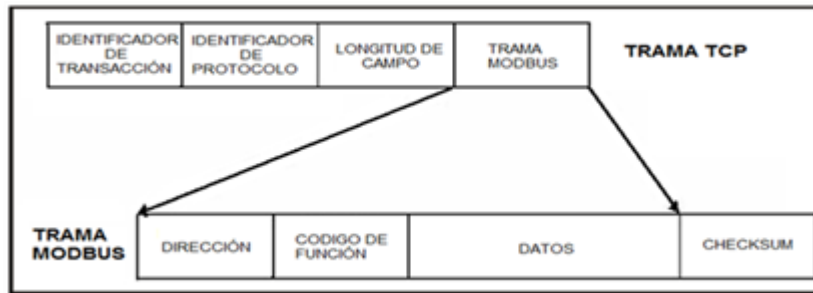
Estos protocolos se aprovechan de las ventajas en cuanto a funcionalidad y seguridad que presentan tanto Ethernet como TCP/IP para ofrecer capacidades superiores de transferencia de información en los sistemas automáticos de control.

### **1.4.1 MODBUS TCP**

Modbus/TCP es un protocolo de comunicación desarrollado para permitir la comunicación dentro de una red, de equipos industriales como: PLCs, computadores, variadores de velocidad, monitores de energía entre otros dispositivos físicos de entrada/salida. Modbus/TCP fue desarrollado por Schneider Automation como una variación de la gama de protocolos MODBUS, comúnmente usada para la supervisión y el control de equipos de automatización.

La especificación Modbus/TCP determina un estándar interoperable en el área de la automatización industrial, el cual es sencillo de implementar para cualquier dispositivo que permita conexión de sockets TCP/IP. “Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502 y normalmente usando comunicación half-duplex sobre una conexión dada.” (Olaya, López y Moreno, 2004)

Esto significa, que no se obtiene ningún beneficio en enviar solicitudes agregadas sobre una conexión única mientras está pendiente una respuesta. Modbus/TCP básicamente encapsula una trama MODBUS dentro de una trama TCP en una manera simple como se muestra en la figura 20-1.



**Figura 20 - 1.** Esquema de encapsulamiento en Modbus/TCP.

Fuente: (Olaya, López y Moreno, 2004).

El monitoreo continuo de datos también llamado “streaming data” es poco eficiente con el protocolo Modbus/TCP, debido principalmente a la sobrecarga que provoca el uso del protocolo de transporte TCP. Esta sobrecarga se debe al servicio de entrega de datos confiable y la notificación de recepción para cada paquete transmitido, de esta manera se incrementa considerablemente el tráfico en la red cuando se monitorea de manera recurrente un esclavo Modbus/TCP particular. Sin embargo, esto puede significar un problema menor a medida que Ethernet incrementa su velocidad. La solución para el monitoreo de datos continuos sobre una red Ethernet es la implementación del protocolo de transporte UDP, con la desventaja de perder confiabilidad.

#### **1.4.2 Bluetooth.**

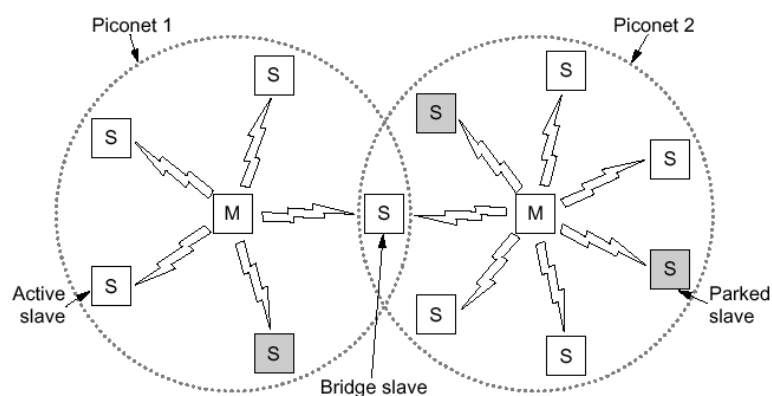
Bluetooth es una especificación industrial para las redes de área personal (PAN), también conocida como IEEE 802.15.1. Bluetooth proporciona una forma de conectar e intercambiar información entre dispositivos como: teléfonos móviles, ordenadores portátiles, PCs, impresoras, cámaras digitales y otros, a través de radiofrecuencia de corto alcance, segura, de bajo costo y disponible a nivel mundial.

La especificación del Bluetooth se diseñó inicialmente para desarrollar dispositivos interconectables de bajo consumo de energía a través de radiofrecuencias de corto alcance de 1 a 100 metros, dependiendo de la categoría del dispositivo.

Si están dentro del rango antes mencionado, dos dispositivos que cuentan con la especificación Bluetooth pueden comunicarse, a pesar de no estar en la misma habitación, sin embargo, se debe respetar una distancia máxima de 100 metros. La transferencia de datos y la voz varían en muchos

dispositivos Bluetooth dependiendo de los requisitos como de la cantidad de la energía necesaria para la transmisión, la tasa de transferencia y la distancia.

Una de las principales ventajas de utilizar esta tecnología es la posibilidad de crear PANs de una manera ad-hoc, es decir, un dispositivo que entra en el rango de otro, se conecta automáticamente y estos constituyen un PAN. En este escenario, a un dispositivo se le da el papel de maestro, mientras que a los demás se les da el papel de esclavo. A este conjunto de amo y esclavos se le da nombre de piconet. Cada piconet puede tener hasta 8 dispositivos. Cuando dos piconets se conectan, a través de un dispositivo común, se forma para llamado Scatternet, ésta tipología es mostrada en la figura 21-1.



**Figura 21 - 1.** Topología de una red Bluetooth.

Fuente: ( Siqueira, 2006).

Otra característica llamativa de la especificación del Bluetooth es la seguridad. Mediante mecanismos como el salto de frecuencia, la autenticación mediante PIN (número de identificación personal) y la encriptación de 128 bits, se garantiza una comunicación segura y sin interferencias entre los dispositivos Bluetooth. (Siqueira, 2006)

## 1.5 Interfaz de Usuario

La interfaz gráfica de usuario (GUI) es el contenido gráfico a través del cual se visualiza información del equipo en una pantalla. La mayoría de los programas tienen alguna forma de interfaz visual, que sirve tanto para mostrar información al usuario como como un mapa de navegación entre los diferentes comandos. Existen interfaces visuales mucho más complejas y diseños más intuitivos, como las de los teléfonos inteligentes, cuyo objetivo es disminuir al máximo la curva de aprendizaje. De este modo la interfaz de los sitios web también tiende a estar diseñada para que cualquier visitante pueda usarla sin requerir conocimientos específicos previos.

### ***1.5.1 MIT App Inventor 2***

MIT App Inventor es una herramienta de programación visual de arrastrar y soltar desarrollada para diseñar y crear aplicaciones móviles completamente funcionales para Android. App Inventor promueve el desarrollo de soluciones tecnológicas únicas a partir del diseño y creación de aplicaciones.

El paradigma de programación intuitiva de App Inventor y las capacidades de desarrollo incremental permiten al desarrollador centrarse en la lógica para programar una aplicación en lugar de en la sintaxis del lenguaje de codificación, fomentando la alfabetización digital para todos. Desde que se trasladó de Google al MIT, se han agregado una serie de mejoras y se están llevando a cabo proyectos de investigación. (Pokress y Veiga, 2013)

### ***1.5.2 Entorno de programación MIT App Inventor 2.***

El entorno de programación de MIT App Inventor 2, contiene dos ventanas principales que son la ventana de diseño en donde se define visualmente la interfaz y la ventana de bloques en donde se define estructuralmente los procesos que la aplicación ejecuta.

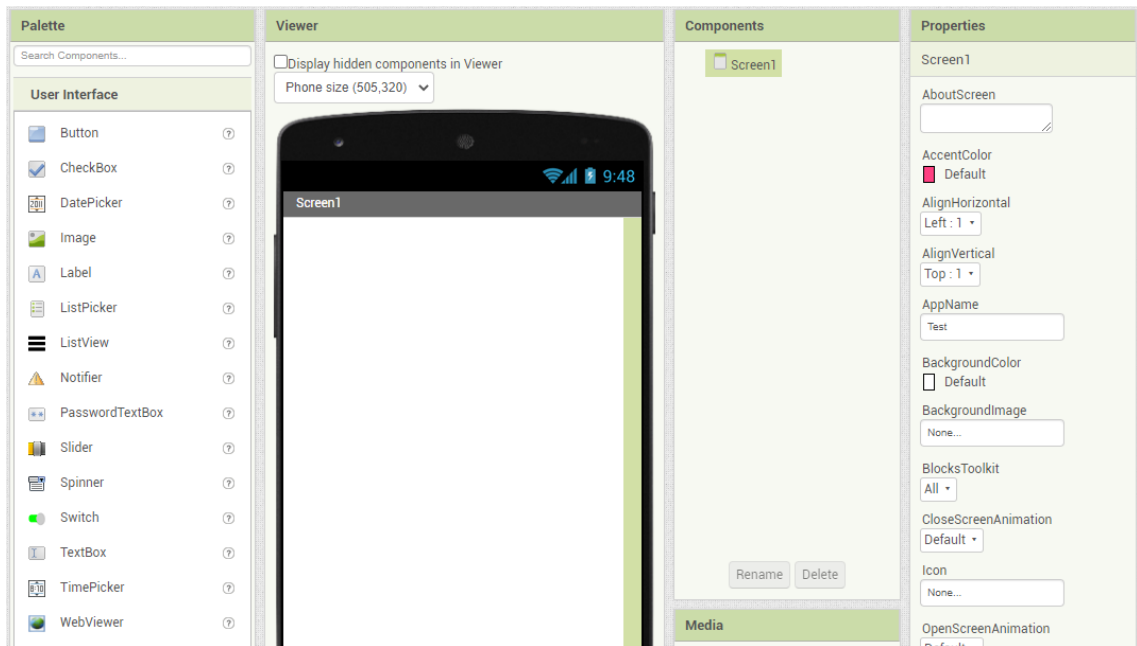
La ventana de diseño que se muestra en la figura 22-1, se encuentra dividida en cuatro secciones principales:

**Paleta:** ubicada a la izquierda de la pantalla, contiene todos los componentes que pueden ser añadidos a la interfaz como botones, imágenes, sonidos, etc.

**Visor:** simula una pantalla de un dispositivo móvil, aquí se añaden los diferentes componentes y se da forma al aspecto que tendrá la interfaz de la aplicación que se desarrolla.

**Componentes:** esta sección recoge en una lista aquellos elementos que finalmente forman parte de la aplicación.

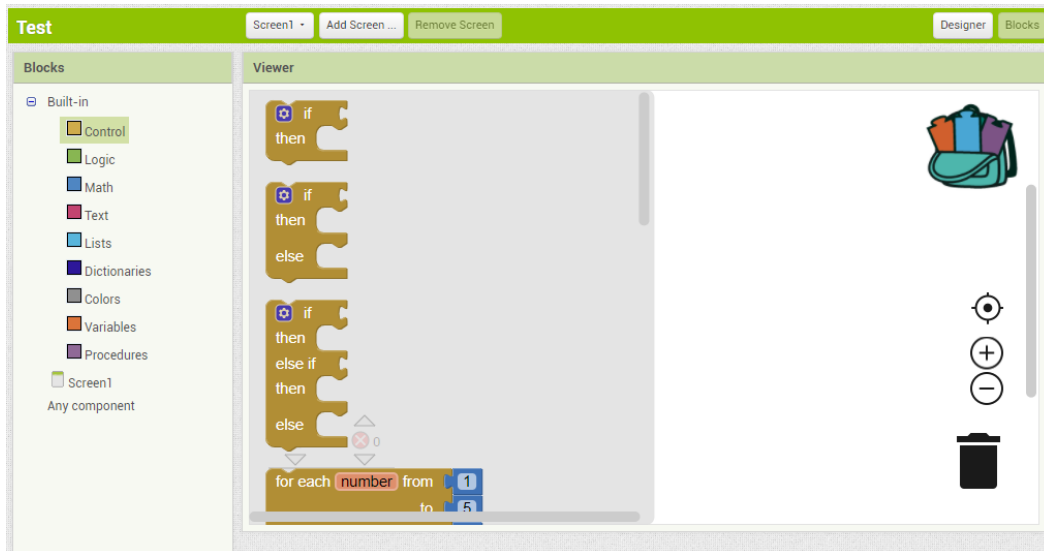
**Propiedades:** esta sección permite modificar las propiedades de los elementos que componen la aplicación.



**Figura 22 - 1.** Ventana de diseño App Inventor.

**Fuente:** (Pokress y Veiga, 2013).

La ventana de bloques como se muestra en la figura 23-1, permite el desarrollo de la programación que define el comportamiento de los componentes seleccionados, mediante la unión de bloques que permitirán que se ejecuten los diferentes eventos.



**Figura 23 - 1.** Ventana de bloques App Inventor.

**Fuente:** ( Pokress y Veiga, 2013).

## **1.6 Productos lácteos.**

El grupo de productos lácteos incluye alimentos como la leche y sus derivados procesados. Las plantas industriales que procesan este tipo de alimentos pertenecen a la industria láctea y se caracterizan por el manejo de un producto altamente perecedero, como la leche, que requiere una correcta vigilancia y análisis durante todos los pasos de la cadena de frío hasta que llega al consumidor.

Las propiedades físicas y químicas de los lácteos se analizan en muchos casos de manera similar que, en la leche, es decir, se utilizan por ejemplo lactómetros para medir la densidad específica. Sin embargo, la elaboración de los lácteos es diferente dependiendo el proceso al que se someta; por ejemplo, los yogures requieren un proceso de fermentación láctica, por otro lado, la elaboración de mantequilla requiere un proceso mecánico de concentración de su contenido graso.

En ciertos casos se requieren procesos combinados como en la elaboración de quesos, en donde se debe realizar un proceso de fermentación seguido de maduración. Estos procesos modifican la concentración y la composición inicial de ciertos macronutrientes y micronutrientes, dependiendo del lácteo en cuestión.

### ***1.6.1 Tipos de productos lácteos.***

Los procesadores de leche producen una amplia variedad de productos lácteos que se detallan a continuación:

La leche líquida es el producto lácteo más consumido, producido y comercializado. La leche líquida incluye productos como la leche desnatada, la leche pasteurizada, la leche normalizada, la leche reconstituida, la leche de prolongada conservación (UHT) y la leche enriquecida. El consumo de leche líquida en forma cruda se ve reduciendo cada vez más en todo el mundo.

Las leches fermentadas se usan frecuentemente para elaborar otros productos lácteos. Se consigue a partir de la fermentación de la leche mediante el uso de microorganismos adecuados para alcanzar un nivel deseado de acidez, como en el caso del yogurt.

Los quesos se obtienen a partir de la coagulación de la proteína láctica también llamada caseína, que se separa del suero. Existen muchas variedades de queso, las distintas características de los quesos derivan de las diferencias en la composición de la leche y los tipos de esta, los procesos de elaboración y los microorganismos utilizados.

La mantequilla es un producto graso derivado de la leche, se obtiene del batido de la leche o nata; en varios países en desarrollo, la mantequilla tradicionalmente se obtiene batiendo la leche entera agria. Las leches condensadas y evaporadas se obtienen de la eliminación parcial del agua de la leche entera o desnatada. La elaboración consiste en aplicar tratamiento térmico para asegurar la estabilidad e inocuidad bacteriológica de la leche. La leche en polvo se consigue a partir de la deshidratación de la leche y su presentación más común es en forma de polvo o gránulos. (Bonet, 2005)

### ***1.6.2 Yogurt.***

El yogurt es un alimento funcional, un derivado lácteo obtenido por procesos de fermentación de bacterias ácido lácticas de la leche. Desde tiempos antiguos son ampliamente conocidos los efectos que tiene en la salud humana el yogurt, entre ellos están: prevención de cáncer de colon, disminución de niveles colesterol en la sangre, mejoramiento de la flora intestinal, efectos positivos en el sistema inmune y prevención de helicobacter pylori, entre otros. Las bacterias encargadas de estos efectos son las bacterias ácido-lácticas-probióticas como Bifidobacterias, Estreptococos y principalmente Lactobacilos. (Huertas, 2012)

## II. CAPÍTULO II - MARCO METODOLÓGICO

### 2.1 Tarjeta de desarrollo Arduino

La tarjeta electrónica Arduino UNO es un dispositivo que consta de entradas y salidas, analógicas y digitales, comunicación serial, I2C, etc. Con las que se puede contar para realizar cualquier proyecto debido a su variedad de herramientas, la información que se encuentra en la comunidad de Arduino al existir librerías para trabajar con el PLC, tiene suficiente velocidad de respuesta para trabajar con el Modbus, su facilidad al momento de programar y el bajo costo con el que se puede adquirir a diferencia de otras placas de desarrollo de hardware hace factible su implementación en este prototipo.



**Figura 1 - 2.** Arduino Uno

**Fuente:** Nylamp, 2020, <https://bit.ly/3kxOHgi>

Las características técnicas del dispositivo se presentan a continuación en la tabla 1-2.

**Tabla 1-2:** Datos técnicos de la tarjeta Arduino UNO

PARÁMETRO	VALOR	RANGO
Entradas analógicas	6 AI	0 V a +5 V
Entradas / Salidas digitales	14 DI/O	+5 VDC
Salidas PWM	6 PWM	+5 VDC
Alimentación	USB	5 VDC

**Fuente:** <http://www.ni.com/pdf/manuals/375296c.pdf>

**Realizado por:** Pazmiño C. 2020

Arduino hará la parte de intermediario recibiendo la información enviada desde LOGO y la enviará a la aplicación móvil mediante el bluetooth.



## 2.2 Bluetooth HC-06

Es un dispositivo que permite la comunicación serial inalámbrica con cualquier teléfono móvil, tablet u ordenador. Este dispositivo es muy usado para transmisión de datos en proyectos electrónicos de forma inalámbrica.



**Figura 2 - 2.** Bluetooth HC - 06

**Fuente:** Nylamp, 2020, <https://bit.ly/3kxOHgi>

Las características técnicas del dispositivo se presentan a continuación en la tabla 2-2.

**Tabla 2-2:** Datos técnicos del Bluetooth HC-06

PARÁMETRO	VALOR
Protocolo Bluetooth	Protocolo estándar Bluetooth V2.0
Nivel de potencia	Class2(+6dBm)
Banda de frecuencia	2.40GHz—2.48GHz, ISM Band
Sensibilidad de recepción	-85dBm
Voltaje de operación	+3.3V a +6V
Temperatura de operación	-20°C a +55°C
Corriente de operación	40mA

**Fuente:** <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet>

**Realizado por:** Pazmiño C. 2020

El módulo HC-06 ayudará en la comunicación con el dispositivo móvil enviando la información a este y recibiendo los datos que procesa el Arduino otorgados por el LOGO 8 encargado del proceso de producción del yogurt.

## 2.3 Shield Ethernet W5100

Las shields de Arduino ayudan en la fácil integración de herramientas con las placas Arduino UNO, MEGA, etc. Además, que con las librerías desarrolladas por la comunidad permite la fácil

integración de una comunicación Ethernet ya sea de manera local o una conexión global a través de internet.



**Figura 3 - 2.** Shield Ethernet W5100

**Fuente:** Nylamp, 2020, <https://bit.ly/307rw18>

Las características técnicas del dispositivo se presentan a continuación en la tabla 3-2.

**Tabla 3-2:** Datos técnicos del Shield Ethernet w5100

PARÁMETRO	VALOR
Microcontrolador	ATmega328
Voltaje de operación	5VDC
Entradas / Salidas digitales	14 DI/O
Entradas analógicas	6 IA

**Fuente:** <http://www.farnell.com/datasheets/1638960.pdf>

**Realizado por:** Pazmiño C. 2020

La shield permitirá la comunicación MODBUS TCP entre el Arduino UNO y el LOGO 8, extrayendo y enviando información según la programación realizada en el Arduino.

## 2.4 Logo 8

Se lo considera como un PLC para aplicaciones pequeñas de la marca Siemens por las grandes prestaciones que tiene, pero se ve limitado por los módulos de ampliación que se puede adicionar y la memoria limitada que se puede utilizar para la programación. Éste se programa en el software propiciado por la marca, llamado Logo Soft Comfort.



**Figura 4 - 2.** Logo 8

**Fuente:** Siemens, 2020, <https://bit.ly/3j836PN>

Las características técnicas del dispositivo se presentan a continuación en la tabla 4-2.

**Tabla 4-2:** Datos técnicos del Logo 8

PARÁMETRO	VALOR
Alimentación	12/24VDC
Entradas digitales	8 DI
Entradas analógicas integradas	4 AI (0-10VDC)
Salidas digitales tipo relé	4 DO RLY

**Fuente:** Siemens, 2020

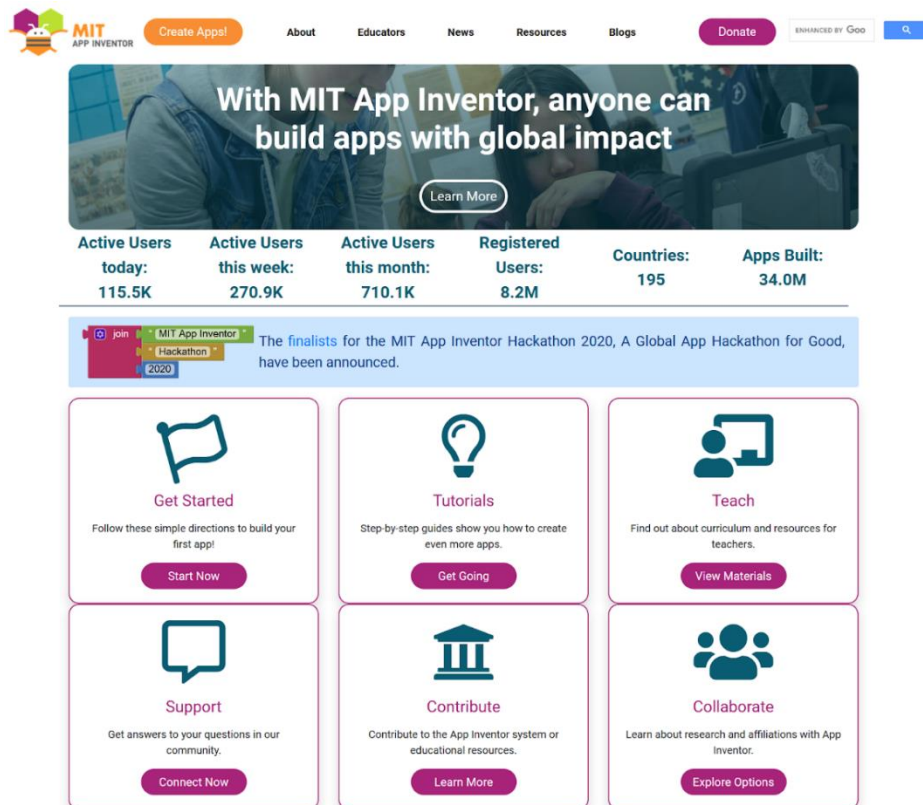
**Realizado por:** Pazmiño C. 2020

Logo es la pieza central del sistema de producción de yogurt, es el controlador que maneja los sensores, pulsadores (entradas) y el motor, elementos de alarma y señalización (salidas). El dispositivo se encarga de la toma de decisiones acorde a su programación cargada y diseñada para este fin, donde se extraerá la información requerida para este proyecto.

## 2.5 Diseño de la interfaz gráfica en la aplicación móvil

El diseño de la aplicación se realizó con “MIT App Inventor 2”, por la facilidad de programación y al entorno amigable que propicia para la creación de una aplicación, sin tener el conocimiento de un lenguaje de programación destinado a ese propósito.

MIT App inventor es una aplicación web que no requiere de una instalación previa basta con abrir entrar a su página web como se muestra en la figura 5 - 2 y presionar el botón “Create Apps”.

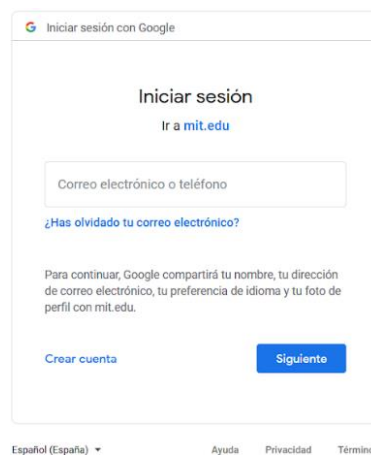


**Figura 5 - 2.** Página de inicio de App Inventor

Realizado por: Pazmiño C. 2020

### 2.5.1 Inicio de sesión y entorno gráfico de la aplicación

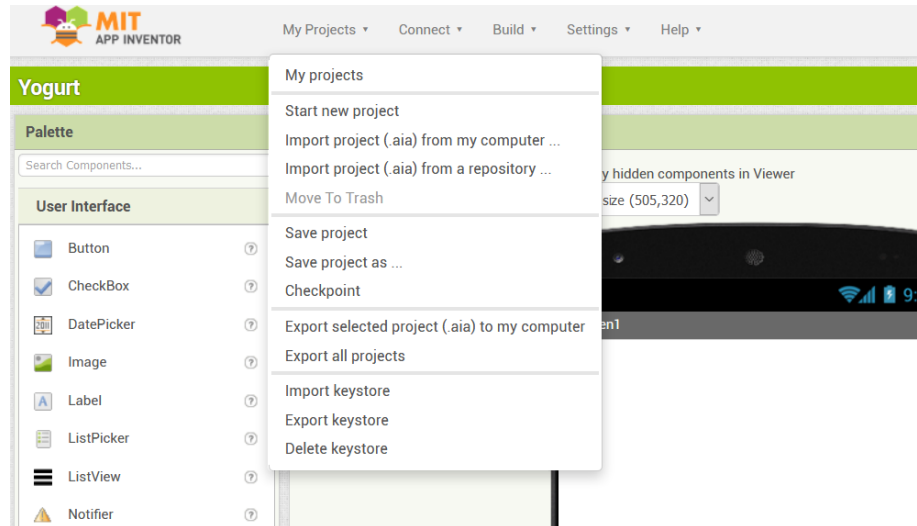
Aparece la página de inicio de sesión de Google como se muestra en la figura 6 -2, en la que se debe ingresar el correo electrónico de Gmail, seguidamente la contraseña para comenzar a programar la aplicación móvil con App inventor.



**Figura 6 - 2.** Inicio de sesión de Google

Realizado por: Pazmiño C. 2020

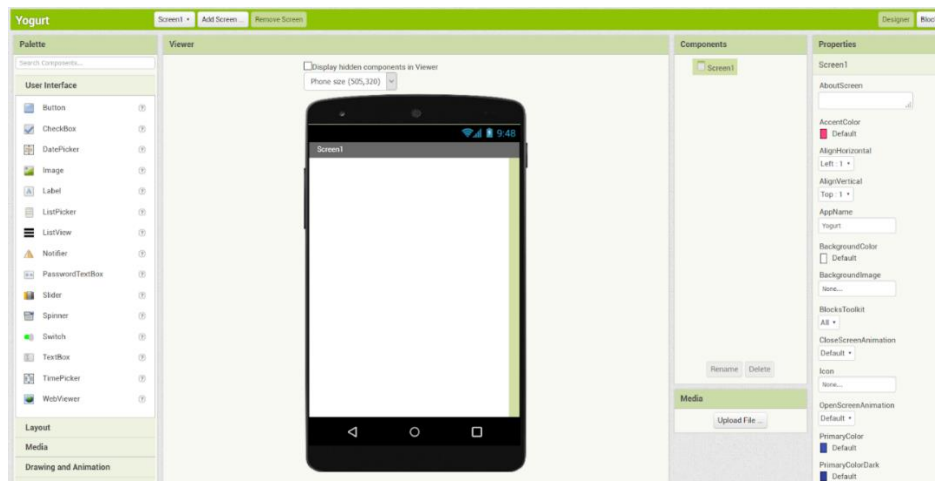
Iniciada la sesión con Google se procede a crear un nuevo proyecto pulsando “My projects” en la barra de opciones junto al logo de MIT APP INVENTOR de la parte superior de la pantalla posteriormente aparecerá un menú, y se selecciona la opción “Start new project” como se muestra en la figura 7 - 2.



**Figura 7 - 2.** Menú desplazado de “My Projects”

**Realizado por:** Pazmiño C. 2020

En la figura 8 - 2 se muestra la pantalla que se despliega para comenzar con el diseño de la aplicación, simplemente es una pantalla en blanco dentro de un dispositivo móvil, donde se pondrá los elementos que irán en la interfaz y servirá para que el usuario pueda usarla con mucha facilidad.

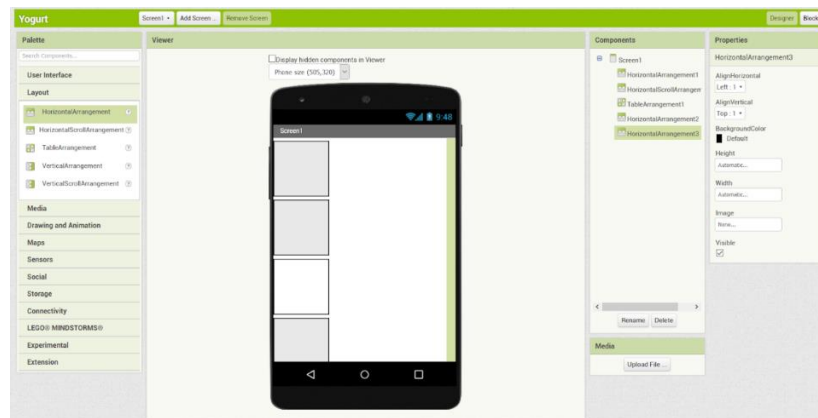


**Figura 8 - 2.** Pantalla principal del nuevo Proyecto

**Realizado por:** Pazmiño C. 2020

## 2.5.2 Inserción de elementos en la aplicación móvil

El primer paso para comenzar con el diseño de la aplicación móvil es insertar los layouts que son los que albergarán los elementos como: botones, labels, sliders, etc. Estos permiten organizar en la pantalla estos elementos y también permite decidir la orden de inserción de los elementos sea de izquierda a derecha o de arriba hacia abajo de acuerdo con la necesidad de la aplicación. Véase la figura 9 - 2.

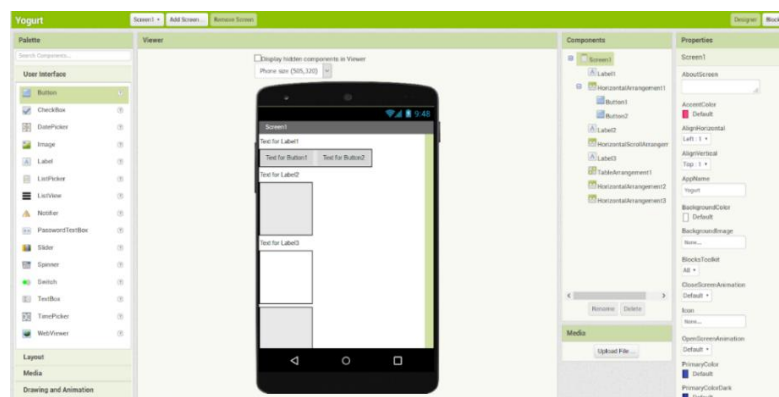


**Figura 9 - 2.** Inserción de layouts en la aplicación para insertar los elementos.

Realizado por: Pazmiño C. 2020

Insertamos los elementos que se requiere para que el usuario la use, en este caso se insertó algunos label que servirán para mostrar los títulos que identificarán las secciones pertenecientes a conexión con dispositivos bluetooth, inicio y paro del proceso y visualización de parámetros medidos por el sistema.

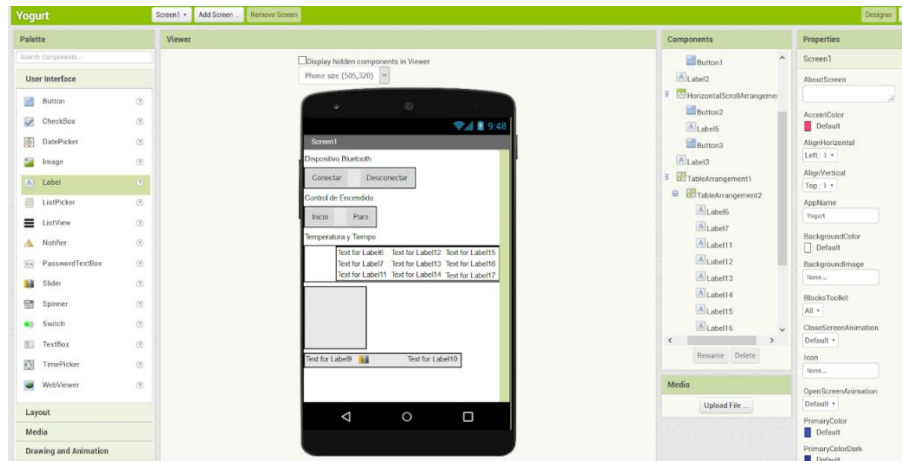
En la figura 10 - 2 se visualiza la introducción de los botones destinados a establecer la conexión con el dispositivo bluetooth y la desconexión de este, estos se encuentran sin identificación momentáneamente porque solo se priorizará el diseño posteriormente se rotulará cada elemento.



**Figura 10 - 2.** Inserción de botones y labels para la conexión bluetooth

Realizado por: Pazmiño C. 2020

En la figura 11 - 2 se muestra la sección de la conexión bluetooth con los botones y labels ya identificados, estos se lo realizan en el apartado “Properties” en la opción “Text”, se cambió en nombre de “Text for Botton1” por “Conectar”. En esta figura también se puede apreciar que se añadieron más labels destinados a mostrar los parámetros medidos por el sistema que son temperatura actual, temperatura del setpoint y tiempo transcurrido.



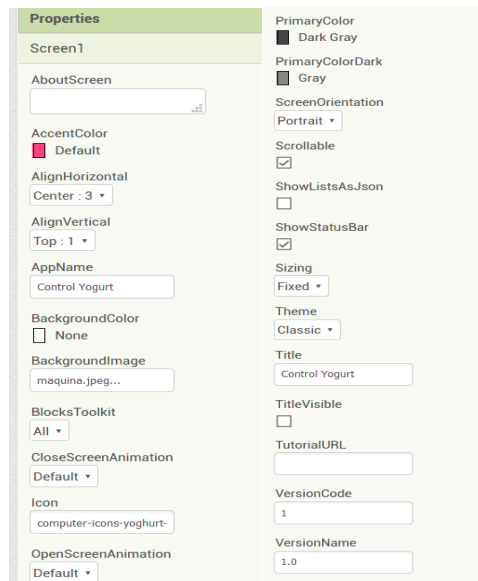
**Figura 11 - 2.** Preparación de labels para los datos de temperatura y tiempo.

**Realizado por:** Pazmiño C. 2020

### 2.5.3 Alineación e identificación de los elementos de la aplicación móvil

El orden y la identificación de cada elemento es importante, por lo que se procedió a nombrar cada label y cada botón de la aplicación, ya que esto también servirá para saber a qué elemento nos estamos dirigiendo dentro del entorno de programación.

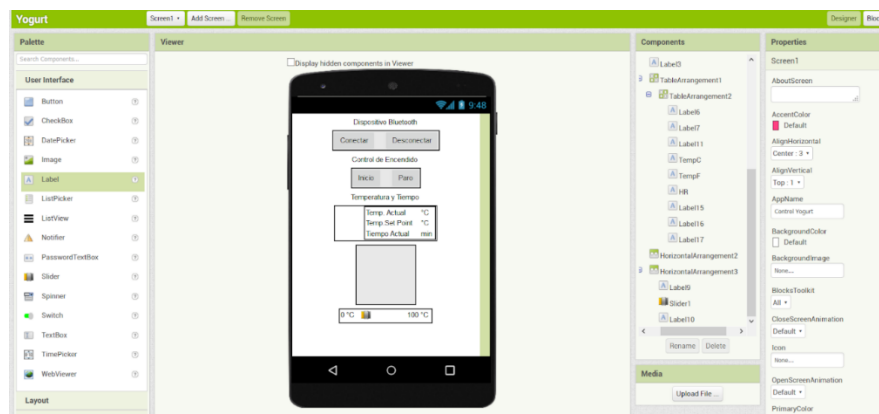
En el apartado de “Properties”, seleccionando la pantalla principal de la aplicación llamada “Screen1”, en la sección de componentes, se modificará la alineación de los elementos y también se ocultará el título de la pantalla con ese nombre para mejorar la visibilidad. Véase la figura 12 - 2.



**Figura 12 - 2.** Propiedades modificadas de la pantalla.

Realizado por: Pazmiño C. 2020

El resultado de la modificación de las propiedades de la pantalla y la rotulación de los elementos se pueden observar en la figura 13 - 2.



**Figura 13 - 2.** Resultado después de las modificaciones en las propiedades.

Realizado por: Pazmiño C. 2020

#### 2.5.4 Inserción de componentes no visibles en la aplicación móvil

La interfaz ya se encuentra diseñada, ahora se añadirá los elementos que facilitarán la programación de la comunicación y procesamiento de datos internos de la aplicación, estos se encuentran en la parte derecha de la pantalla, en el apartado de “Palette” en la sección de “Connectivity”.



Ahí se encontrará “BluetoothClient” y se arrastra hacia la pantalla. Esto se muestra en la figura 14 - 2, se habrá añadido correctamente cuando aparezca debajo de la pantalla de diseño de la aplicación en la sección de “Non-visible components”.

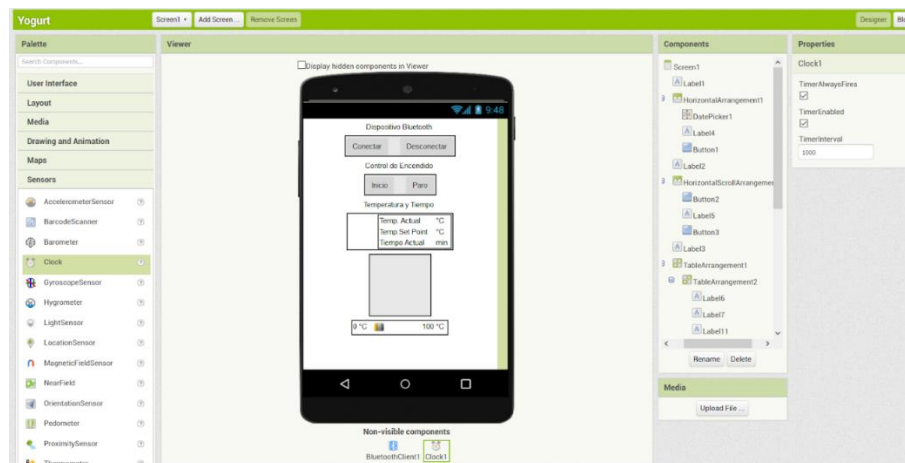
En esta sección “Non-visible components” como su nombre lo indica son componentes que no se ven en la aplicación, pero formarán parte de la programación.



**Figura 14 - 2.** Inserción del componente no visible “BluetoothClient1”.

Realizado por: Pazmiño C. 2020

Otro componente no visible que permite realizar peticiones de manera cíclica y que la información se envíe periódicamente en intervalos de tiempo muy cortos, se encuentra en la parte derecha de la pantalla en la sección de “Sensors” con el nombre de Clock como se muestra en la figura 15 - 2. Este se añade de la misma forma que se añaden estos tipos de componentes.



**Figura 15 - 2.** Adición del reloj a la aplicación

Realizado por: Pazmiño C. 2020

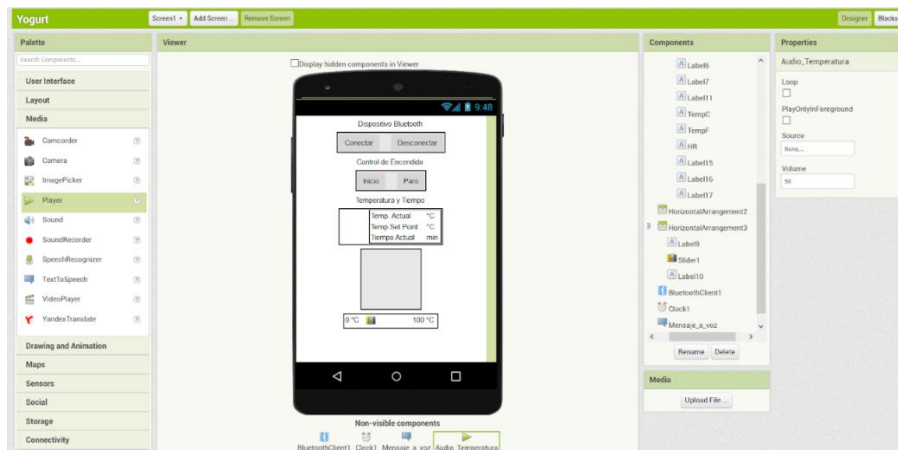
El componente de texto a voz ubicado en la sección de “Media” con el nombre de “TextToSpeech”, el cual hace la parte de notificar al usuario de la aplicación con un mensaje escrito llevado a voz y se añade como se muestra en la figura 16 - 2.



**Figura 16 - 2.** Adición del componente de texto a voz.

Realizado por: Pazmiño C. 2020

Para que la notificación no sea solo sonora sino también que sea vibratoria se debe añadir otro de estos componentes como se muestra en la figura 17 - 2, encontrada en la misma sección de “Media” con el nombre de “Player” y solo basta con arrastrar a la pantalla para poder añadir a la programación.



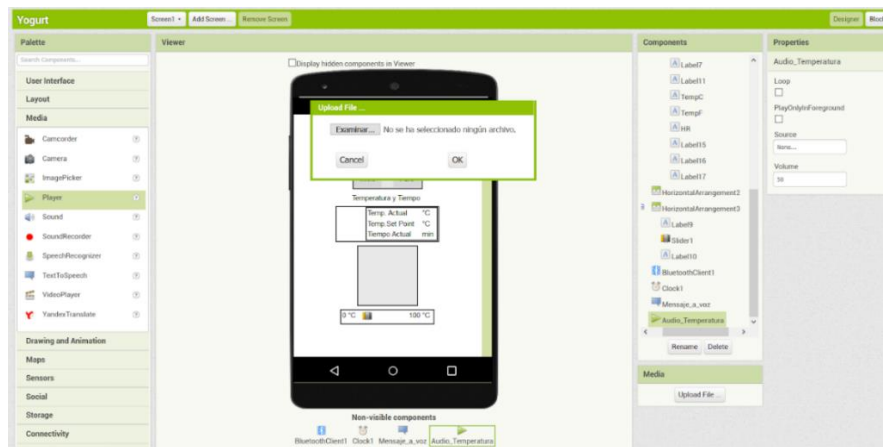
**Figura 17 - 2.** Inclusión de alerta vibratoria para el usuario.

Realizado por: Pazmiño C. 2020

### 2.5.5 Personalización de la aplicación móvil

La personalización de la aplicación depende de los recursos gráficos como imágenes o fotos, recursos que se agregan en el apartado de “Media” a la derecha de la pantalla bajo el apartado de

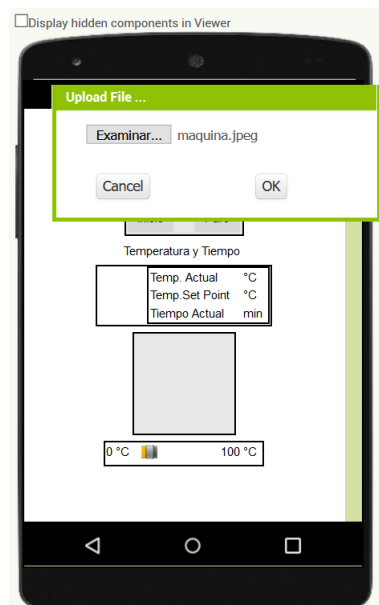
“Components”. Ahí se encuentra un botón “Upload File ...” que al presionar muestra una ventana como se muestra en la figura 18 - 2, para añadir un archivo local para la aplicación se hace clic en “Examinar ...” a continuación mostrará el explorador de archivos para seleccionar la imagen.



**Figura 18 - 2.** Pantalla para subir archivos de imágenes a la aplicación.

**Realizado por:** Pazmiño C. 2020

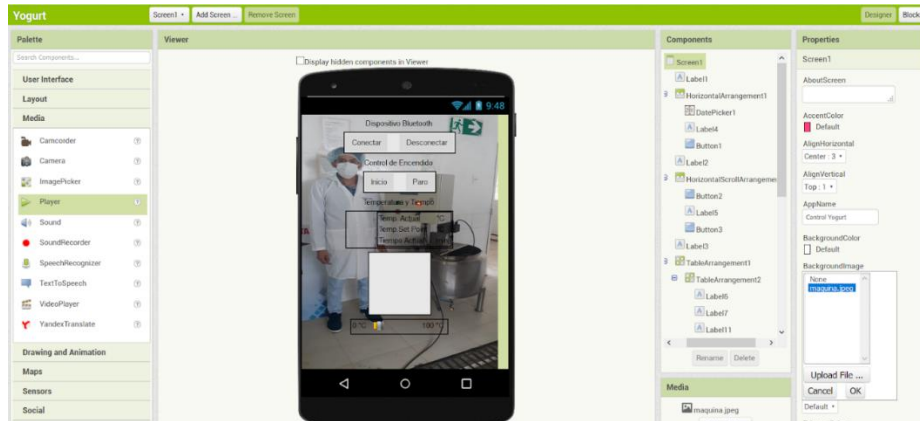
Una vez seleccionado el archivo aparecerá en la ventana “Upload File ...” junto con la extensión de la imagen, solo basta con presionar “OK”, para que el archivo esté disponible para su uso como se muestra en la figura 19 - 2.



**Figura 19 - 2.** Archivo disponible para usar en la aplicación.

**Realizado por:** Pazmiño C. 2020

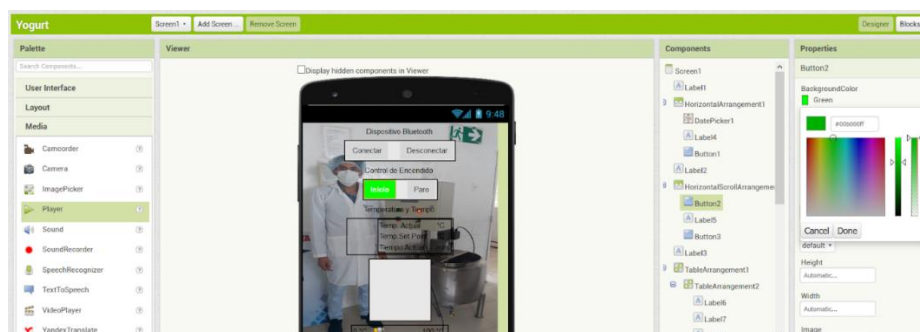
Se optó por cambiar el fondo de la pantalla acorde al trabajo realizado con la imagen que se subió a la aplicación, como se observa en la figura 20 – 2. Para realizar este proceso hay que ubicarse en el apartado de “Properties” en la opción de “BackgroundImage”, muestra la lista de imágenes disponibles para usar, solo se selecciona el archivo y se pulsa aceptar, finalmente, aparece en la pantalla principal de la aplicación.



**Figura 20 - 2.** Archivo disponible para usar en la aplicación.

**Realizado por:** Pazmiño C. 2020

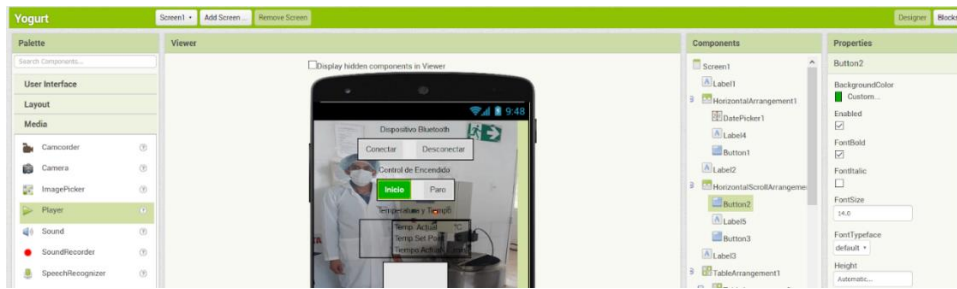
Para personalizar la aplicación y mejorar la apariencia visual se cambió el color de botones como se muestra en la figura 21 - 2. Se selecciona el elemento a cambiar la apariencia y se elige la opción “Background Color” donde se despliega una paleta de colores para escoger el color acorde a la necesidad de la aplicación.



**Figura 21 - 2.** Personalización de botones.

**Realizado por:** Pazmiño C. 2020

Las opciones personalizables son el color, la fuente y el tamaño del texto, esto sirve para que el botón esté claramente identificado y reconocido como se muestra en la figura 22 - 2.



**Figura 22 - 2.** Edición del texto de botón “Inicio” color, texto y tamaño.

Realizado por: Pazmiño C. 2020

Finalmente, la aplicación móvil tiene una apariencia como se muestra en la figura 23 - 2, editando cada elemento para lograr identificarlos y que los datos obtenidos por la aplicación se distingan claramente.

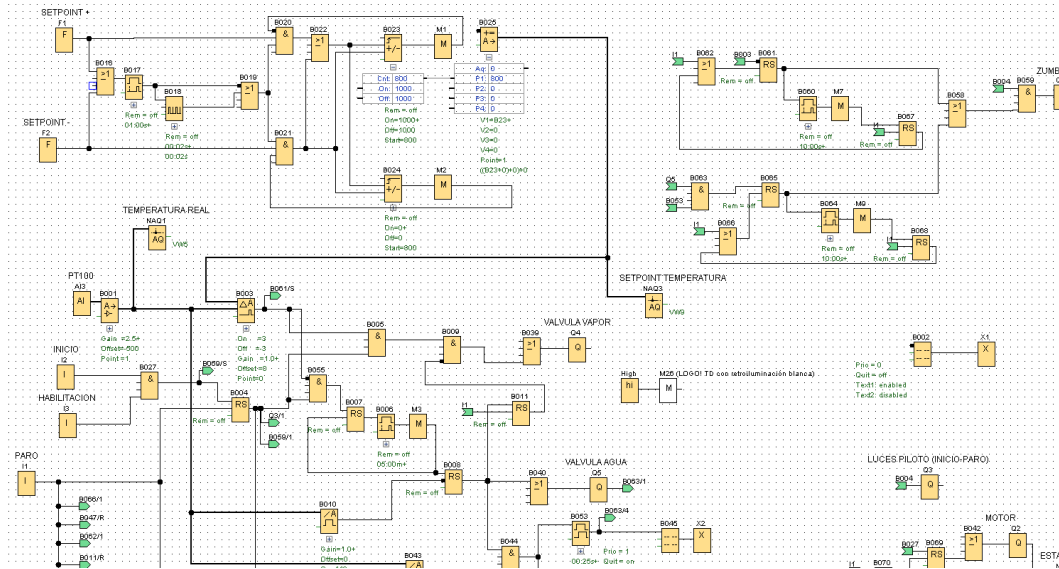


**Figura 23 - 2.** Resultado final de la aplicación

Realizado por: Pazmiño C. 2020

### 2.5.6 Esquema de programación del logo del prototipo

La programación del Logo que controla el sistema como se muestra en la figura 24 - 2, contiene las variables que se requiere la visualización en la aplicación móvil.

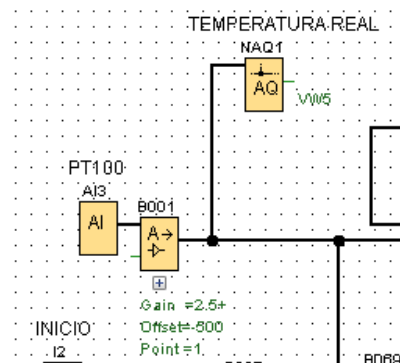


**Figura 24 - 2.** Programación del LOGO que controla del prototipo.

Realizado por: Pazmiño C. 2020

Analizado el programa es necesario obtener las variables que se traslada a la aplicación móvil como:

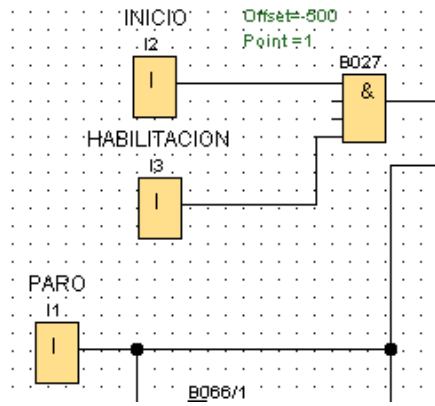
**Temperatura real.** - Esta variable muestra la temperatura actual de la leche, la cual es requerida conocer para la toma de decisiones del operador



**Figura 25 - 2.** Bloque de función de la temperatura real.

Realizado por: Pazmiño C. 2020

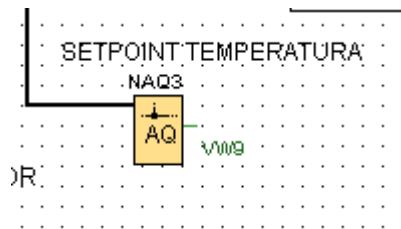
**Inicio y paro.** - Son las variables que controlan el arranque y la detención del sistema. Véase la figura 26 - 2.



**Figura 26 - 2.** Bloques de entrada encargada del inicio y paro.

Realizado por: Pazmiño C. 2020

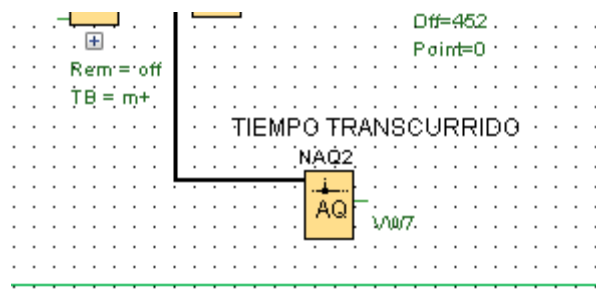
**Setpoint temperatura.** - La temperatura a la que debe llegar el sistema de acuerdo con cada etapa del proceso. Véase la figura 27 - 2.



**Figura 27 - 2.** Bloque de temperatura deseada que el sistema necesita.

Realizado por: Pazmiño C. 2020

**Tiempo transcurrido.** - El tiempo transcurrido una vez que el proceso se inició y la variable se muestra en la figura 28 - 2.

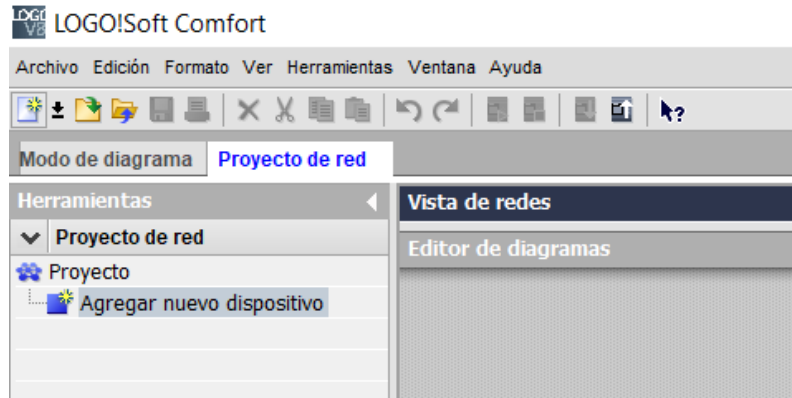


**Figura 28 - 2.** Bloque de función del tiempo transcurrido del proceso.

Realizado por: Pazmiño C. 2020

### 2.5.7 Comunicación Modbus en el controlador Logo

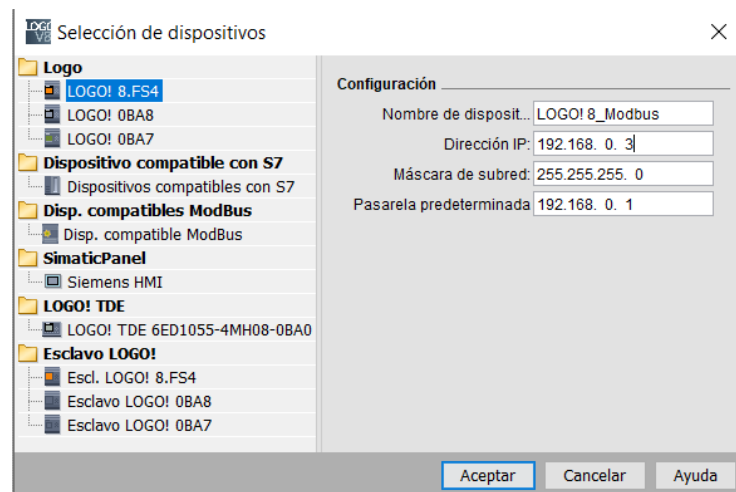
El programa que se encuentra dentro del LOGO, fue realizado como un esquema eléctrico, para realizar la comunicación vía MODBUS con el Arduino se requiere crear un proyecto de red como se muestra en la figura 29 - 2.



**Figura 29 - 2.** Creación de un proyecto de red en LOGO Soft Comfort.

Realizado por: Pazmiño C. 2020

En la figura 30 - 2, pide los datos iniciales del LOGO como nombre, IP, etc. Una vez terminada la configuración se guarda dando clic en “Aceptar”.

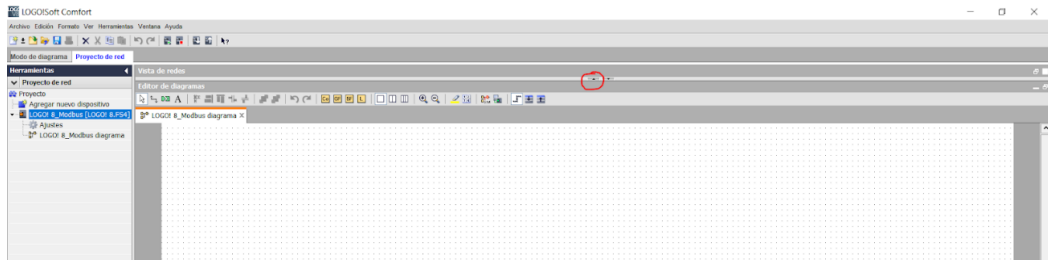


**Figura 30 - 2.** Primera configuración del logo agregado

Realizado por: Pazmiño C. 2020

Cuando el proyecto es creado, muestra la pantalla principal con un diagrama de bloque en blanco listo para programar, como se muestra en la figura 31 – 2. Para ver las opciones de comunicación se debe presionar el botón marcado mostrado.

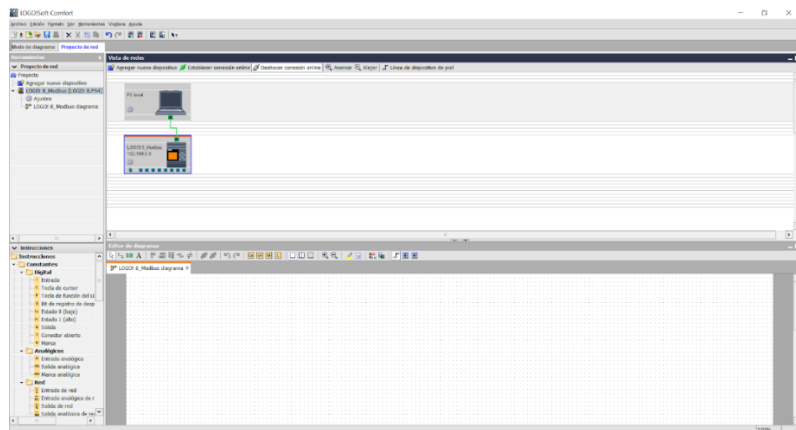




**Figura 31 - 2.** Diagrama en blanco luego de crear el proyecto de LOGO.

Realizado por: Pazmiño C. 2020

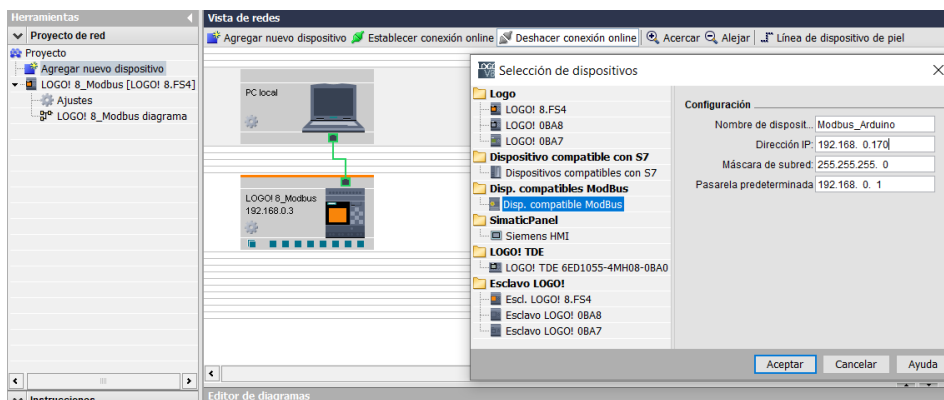
La pantalla que muestra el proyecto es un ordenador conectado al LOGO en un espacio donde se puede añadir los dispositivos con los que se puede comunicar como se muestra en la figura 32 - 2.



**Figura 32 - 2.** Pantalla que muestra a los dispositivos del proyecto de LOGO.

Realizado por: Pazmiño C. 2020

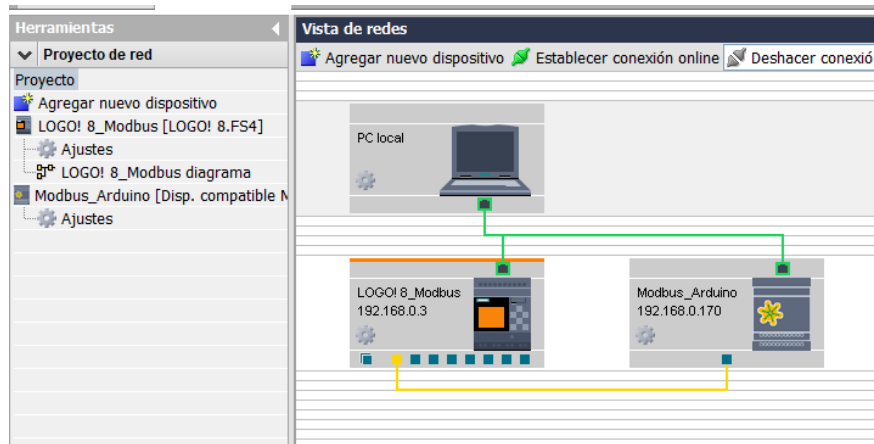
En esta pantalla, se debe hacer clic en la opción “Agregar nuevo dispositivo”, posteriormente muestra una lista donde se debe seleccionar “Disp. compatible Modbus” y llenar los campos como el nombre, IP, etc. Como se muestra en la figura 33 - 2.



**Figura 33 - 2.** Pantalla de configuración del dispositivo Modbus.

Realizado por: Pazmiño C. 2020

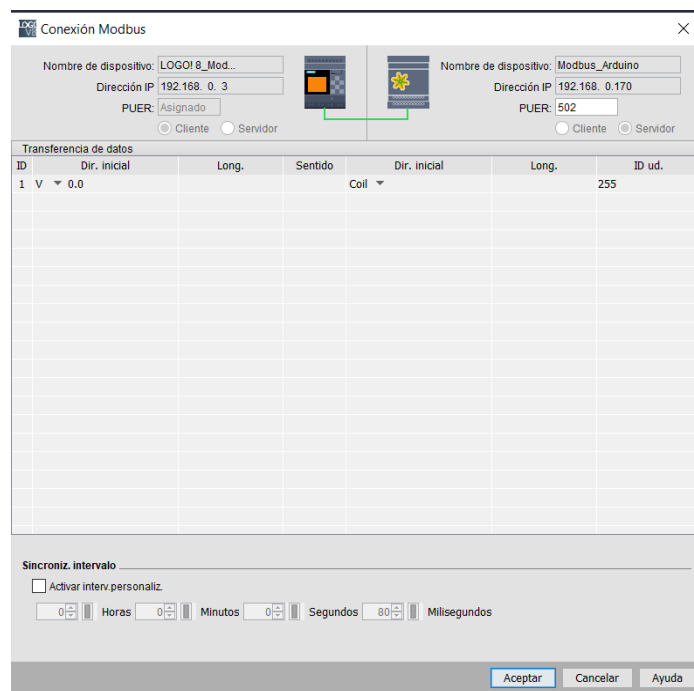
La conexión Modbus se realiza uniendo las partes inferiores de cada dispositivo, como se muestra en la figura 34 - 2, de esa forma se realiza la comunicación.



**Figura 34 - 2.** Conexión Modbus del LOGO y Arduino en el programa LOGO Soft.

Realizado por: Pazmiño C. 2020

Para poder acceder a las configuraciones de los espacios de memoria asignados para la comunicación, se debe realizar doble clic en la línea amarilla a continuación se muestra la tabla para direccionar las memorias mostrada en la figura 35 - 2.



**Figura 35 - 2.** Tabla de direccionamiento de memorias Modbus.

Realizado por: Pazmiño C. 2020

### 2.5.8 Direccionamiento de variables

La tabla de direccionamiento de Modbus contiene la “ID” esta identifica la variable, “Dir. Inicial” es la memoria local del LOGO en la que se recibe o se envía la información al Arduino, “Long.” es la longitud de la información en dato tipo palabra, “Sentido” este apartado indica si el LOGO envía o recibe la información, “Dir. Inicial” del dispositivo Modbus a conectarse en este caso el Arduino. Véase la figura 36 - 2.

**Figura 36 - 2.** Tabla de direccionamiento variables.

Realizado por: Pazmiño C. 2020

En la tabla 1-2 se muestra las variables locales del LOGO hacia direcciones Modbus del Arduino.

**Tabla 1 - 2:** Dirección de las variables locales del LOGO hacia direcciones Modbus del Arduino.

ID	Dir. Inicial	Long.	Sentido	Dir. Inicial	Long.	ID ud.
1	VW 11	1 WORD	< -	IR 121	1 WORD	255
2	VW 13	1 WORD	< -	IR 123	1 WORD	255
3	VW 17	1 WORD	< -	IR 125	1 WORD	255
4	VW 19	1 WORD	< -	IR 127	1 WORD	255
5	VW 5	1 WORD	- >	HR 101	1 WORD	255
6	VW 15	1 WORD	- >	HR 106	1 WORD	255
7	VW 9	1 WORD	- >	HR 111	1 WORD	255
8	VW 21	1 WORD	- >	HR 113	1 WORD	255
9	VW 23	1 WORD	- >	HR 115	1 WORD	255

Fuente: Autor.

Realizado por: Pazmiño, Carlos, 2020.

### 2.5.9 Programación Logo

El primer paso para la asignación de variables es direccionar las entradas y salidas que se requiere para la comunicación. En el apartado de Red y dependiendo del caso, se selecciona Entrada o Salida analógica de red como se muestra en la figura 37 - 2.

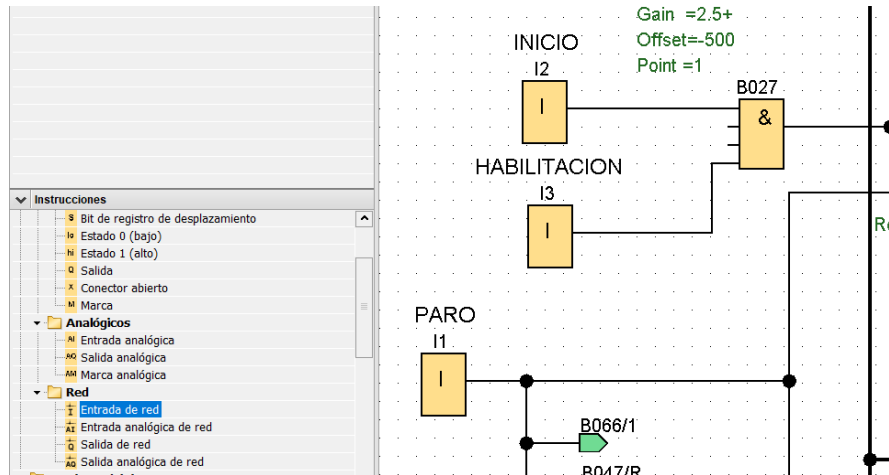


Figura 37 - 2. Identificación de la ubicación de las variables de red.

Realizado por: Pazmiño C. 2020

Las variables que llevan a la aplicación se listan a continuación:

**Inicio.** - Dará el pulso de arranque para el inicio remoto del sistema. Se encuentra direccionado a la Entrada analógica de red VW11 como se muestra en la figura 38 - 2.

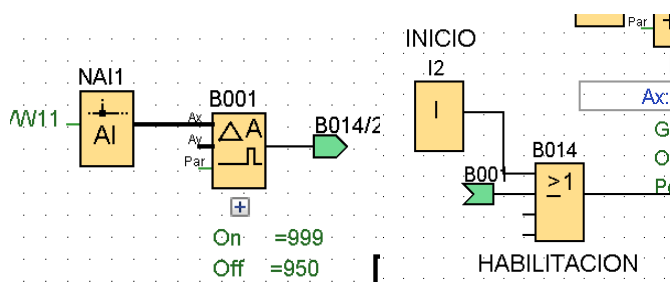
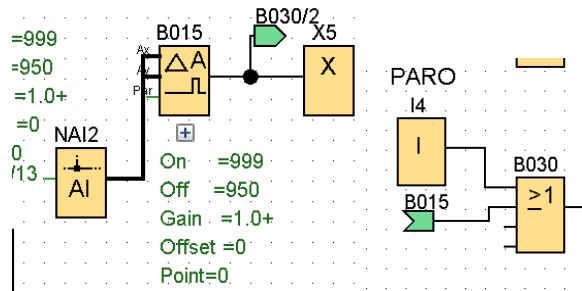


Figura 38 - 2. Identificación de la variable de inicio.

Realizado por: Pazmiño C. 2020

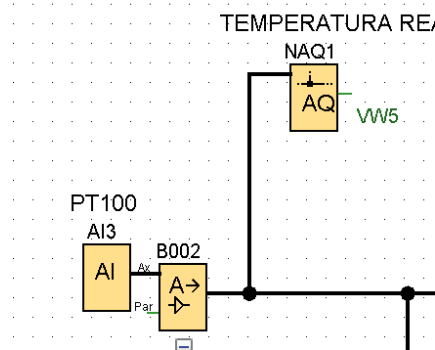
**Paro.** - Terminará el proceso en caso de que sea necesario, sea por una falla o para finalizar todo el proceso. Está asignada en la entrada analógica de red VW13, como se observa en la figura 39 - 2.



**Figura 39 - 2.** Identificación de la variable de paro.

Realizado por: Pazmiño C. 2020

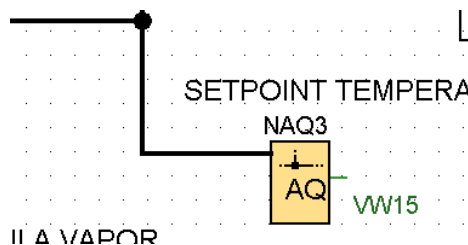
**Temperatura actual.** - Enviada por la salida analógica de red VW5, está la temperatura en la que se encuentra el sistema, mostrada en la figura 40 -2.



**Figura 40 - 2.** Identificación de la variable temperatura actual.

Realizado por: Pazmiño C. 2020

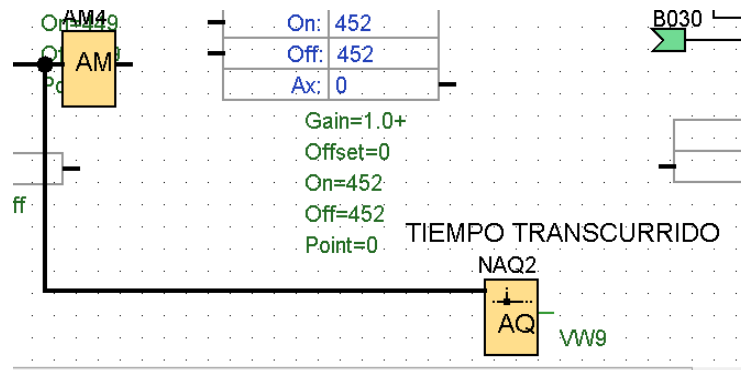
**Temperatura setpoint.** - Es la temperatura a la que se desea que el sistema alcance, se envía por la salida analógica de red VW15 mostrada en la figura 41 - 2.



**Figura 41 - 2.** Identificación de la variable temperatura setpoint.

Realizado por: Pazmiño C. 2020

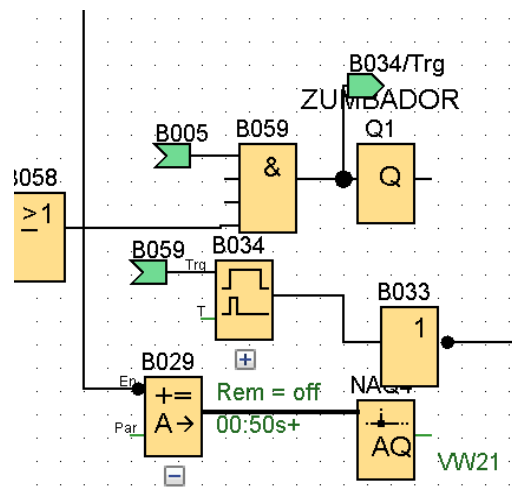
**Tiempo transcurrido.** - Es el tiempo que le toma al sistema en cumplir su ciclo. Esta será enviada por la salida analógica de red VW9. Véase la figura 42 - 2.



**Figura 42 - 2.** Identificación de la variable de tiempo transcurrido.

Realizado por: Pazmiño C. 2020

**Zumbador.** - El zumbador es la señal que emite el sistema cuando necesita que el operador atienda al sistema y esta variable se enviará por la salida analógica de red VW21 como se muestra en la figura 43 - 2.



**Figura 43 - 2.** Identificación de la variable de tiempo transcurrido.

Realizado por: Pazmiño C. 2020

El resumen del direccionamiento de las variables del LOGO con las memorias o espacios de memoria Modbus se ven reflejados en la tabla 2 - 2.

**Tabla 2 - 2:** Variables del LOGO y dirección Modbus asignada

VARIABLES LOGO	MEMORIAS LOGO	MEMORIAS MODBUS
Inicio	VW11	IR 121
Paro	VW13	IR 123
Temperatura actual	VW5	HR 101
Temperatura setpoint	VW15	HR 106
Tiempo transcurrido	VW9	HR 111
Zumbador	VW21	HR 113

**Fuente:** Autor.

**Realizado por:** Pazmiño, Carlos, 2020.

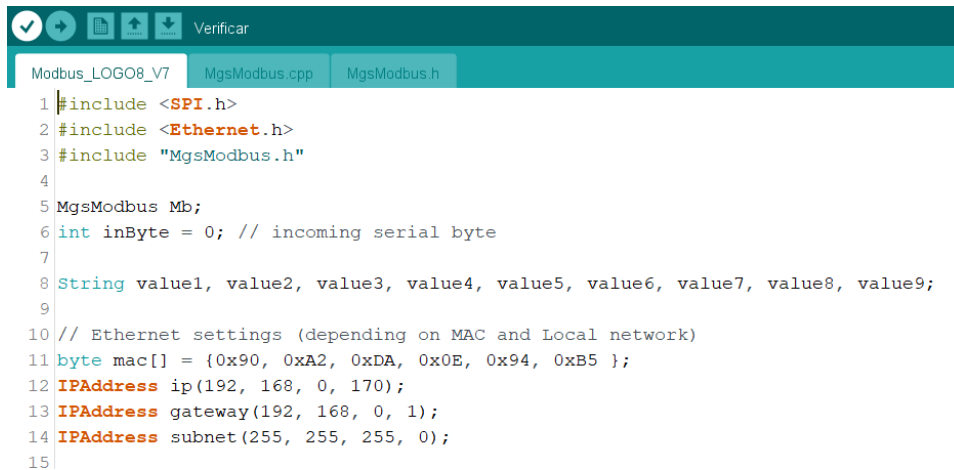
### **2.5.10 Programación Arduino**

En el programa de Arduino existen librerías que ayudan a una mejor optimización de código, para eso se utilizarán 2 librerías distribuidas por el repositorio de la marca, las cuales son “SPI.h” y “Ethernet.h”, estas son las que establecerán la comunicación vía Ethernet.

La última librería es la “Mgs Modbus.h”, esta es compartida por la comunidad de Arduino y se utiliza para comunicar vía Modbus TCP IP al Arduino con el LOGO. A continuación, se crea la variable “Mb”, que usa los atributos de la librería.

Las siguientes dos variables que se crean a continuación son las que se utilizan para guardar los datos obtenidos y enviados desde el LOGO, como son “value1”, “value2”, ..., “value9”.

Finalmente se debe declarar los parámetros para establecer la comunicación como la dirección MAC de la shield de Arduino, la dirección IP, la máscara de subred y la puerta de enlace predeterminado como se muestra en la figura 44 - 2.

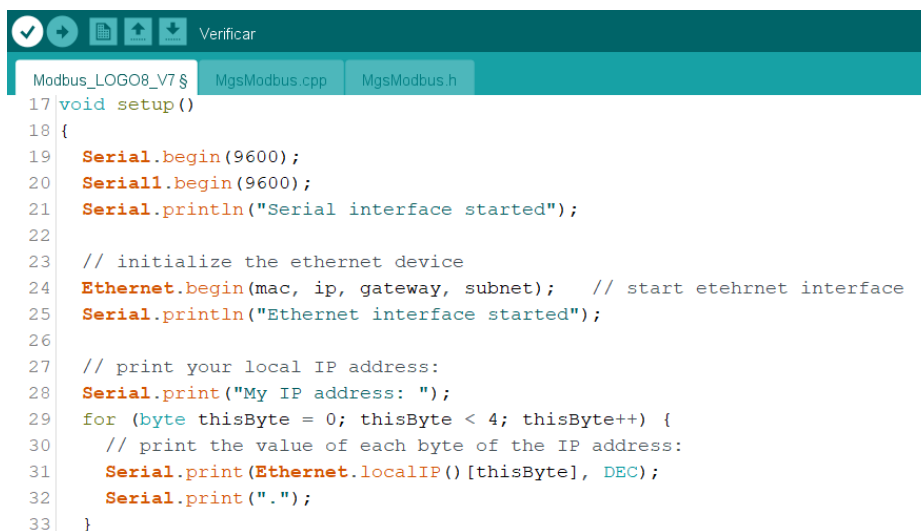


```
1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include "MgsModbus.h"
4
5 MgsModbus Mb;
6 int inByte = 0; // incoming serial byte
7
8 String value1, value2, value3, value4, value5, value6, value7, value8, value9;
9
10 // Ethernet settings (depending on MAC and Local network)
11 byte mac[] = {0x90, 0xA2, 0xDA, 0x0E, 0x94, 0xB5 };
12 IPAddress ip(192, 168, 0, 170);
13 IPAddress gateway(192, 168, 0, 1);
14 IPAddress subnet(255, 255, 255, 0);
15
```

**Figura 44 - 2.** Inicialización de variables y configuración inicial de red Arduino.

Realizado por: Pazmiño C. 2020

En la configuración de `setup ()` de Arduino se inicializa los puertos seriales 0 y 1, a 9600 baudios donde el puerto 0 será para visualizar los datos en el terminal serial de Arduino, mientras que el puerto 1 le pertenece al canal que usará el bluetooth para comunicarse con la aplicación móvil. La configuración de la shield ethernet y nuestra la IP que tiene asignado el Arduino se visualizará por unos momentos en el monitor serial, este código se muestra en la figura 45 - 2.



```
17 void setup()
18 {
19   Serial.begin(9600);
20   Serial1.begin(9600);
21   Serial.println("Serial interface started");
22
23   // initialize the ethernet device
24   Ethernet.begin(mac, ip, gateway, subnet); // start ethernet interface
25   Serial.println("Ethernet interface started");
26
27   // print your local IP address:
28   Serial.print("My IP address: ");
29   for (byte thisByte = 0; thisByte < 4; thisByte++) {
30     // print the value of each byte of the IP address:
31     Serial.print(Ethernet.localIP()[thisByte], DEC);
32     Serial.print(".");
33   }

```

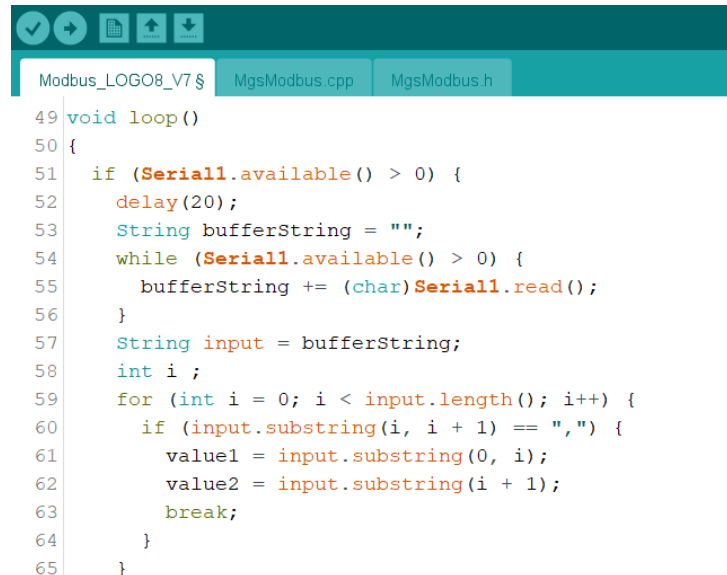
**Figura 45 - 2.** Inicialización de la comunicación serial y ethernet.

Realizado por: Pazmiño C. 2020

El código del void loop inicia preguntando si existe algún dato entrante por el puerto serial 1, este es el dato que envía la aplicación móvil hacia el Arduino mediante bluetooth, si este llega a ser más de un carácter lo convierte en una cadena de caracteres o String.



Una vez obtenida la cadena de caracteres se ve de la siguiente forma “a,”, el siguiente paso es contar el número de caracteres y separar del signo de puntuación coma “,”. Con esto se busca trabajar solo con la letra para comparar y ejecutar el proceso correspondiente como se muestra en la figura 46 - 2.



```
Modbus_LOGO8_V7 $ MgsModbus.cpp MgsModbus.h
49 void loop()
50 {
51   if (Serial1.available() > 0) {
52     delay(20);
53     String bufferString = "";
54     while (Serial1.available() > 0) {
55       bufferString += (char)Serial1.read();
56     }
57     String input = bufferString;
58     int i ;
59     for (int i = 0; i < input.length(); i++) {
60       if (input.substring(i, i + 1) == ",") {
61         value1 = input.substring(0, i);
62         value2 = input.substring(i + 1);
63         break;
64       }
65     }
66   }
```

**Figura 46 - 2.** Código de lectura de información enviada desde la aplicación móvil.

Realizado por: Pazmiño C. 2020

La información que llega desde la aplicación después de ser procesado debe ser interpretada para la toma de decisiones según el caracter que llega desde la app como se observa en la figura 47 - 2.



```
Modbus_LOGO8_V7 $ MgsModbus.cpp MgsModbus.h
65 }
66 if (value1 == "a") {
67   Mb.MbData[120] = 1000;
68 } if (value1 == "b") {
69   Mb.MbData[120] = 950;
70 } if (value1 == "c") {
71   Mb.MbData[122] = 1000;
72 } if (value1 == "d") {
73   Mb.MbData[122] = 950;
74 } if (value1 == "e") {
75   Mb.MbData[124] = 1000;
76 } if (value1 == "f") {
77   Mb.MbData[124] = 950;
78 } if (value1 == "g") {
79   Mb.MbData[126] = 1000;
80 } if (value1 == "h") {
81   Mb.MbData[126] = 950;
82 }
83 Serial.println(value1);
84 }
```

**Figura 47 - 2.** Procesamiento de la información obtenida mediante condicionales.

Realizado por: Pazmiño C. 2020

Los datos que llega de las acciones realizadas en la aplicación son interpretados como se muestra en la siguiente tabla 3 -2

**Tabla 3 - 2:** Interpretación de la información enviada desde la aplicación por el Arduino

<b>Acción desde la aplicación</b>	<b>Caracter enviado desde la aplicación</b>	<b>Datos enviados desde el Arduino al LOGO</b>	<b>Memoria de Modbus utilizada</b>
Pulsar el botón de inicio	A	1000	IR 121
Soltar el botón de inicio	B	950	IR 121
Pulsar el botón de paro	C	1000	IR 123
Soltar el botón de paro	D	950	IR 123

**Fuente:** Autor.

**Realizado por:** Pazmiño, Carlos, 2020.

Para acceder a las memorias Modbus del LOGO desde el Arduino se debe desplazar una memoria para que coincida la información, por ejemplo: si se desea acceder a la memoria 121 del LOGO desde el Arduino se debe seleccionar la memoria 120 como se muestra en la figura 46 - 2.

La comunicación Modbus inicia con la línea 86 “Mb.MbsRun();” línea de comando propia de la librería. A continuación, se envían los datos mediante el puerto serial al que está conectado el bluetooth para visualizar en la aplicación móvil.

La trama de datos comienza con un caracter especial “#,” el cual se utiliza para saber el inicio de la trama enviada hacia la aplicación, y esta pueda procesarla. Cada dato es enviado, la parte entera separada de la parte decimal por un punto, debido a que no puede ser enviado directamente el dato de tipo real.

Los datos que requiere la aplicación son la temperatura actual, temperatura setpoint y el tiempo transcurrido; cada uno de ellos está separado con una coma “,” como se muestra en la figura 48 – 2 los cuales serán procesados posteriormente por la aplicación móvil.

```

Modbus_LOGO8_V7 $ MgsModbus.cpp MgsModbus.h
86 Mb.MbsRun();
87 Serial.print("#,");
88 Serial.print(Mb.MbData[100]/10,1);
89 Serial.print(".");
90 Serial.print(Mb.MbData[100]-Mb.MbData[100]/10*10);
91 Serial.print(",");
92 Serial.print(Mb.MbData[105]/10,1);
93 Serial.print(",");
94 Serial.print(Mb.MbData[110]/10);
95 Serial.print(",");
96 Serial.print(Mb.MbData[112]);
97 Serial.print(",");
98 Serial.println(Mb.MbData[114]);
99 Serial.print("#,");
100 Serial.print(Mb.MbData[100]/10,1);
101 Serial.print(".");
102 Serial.print(Mb.MbData[100]-Mb.MbData[100]/10*10);
103 Serial.print(",");
104 Serial.print(Mb.MbData[105]/10,1);
105 Serial.print(".");
106 Serial.print(Mb.MbData[105]-Mb.MbData[105]/10*10);
107 Serial.print(",");
108 Serial.print(Mb.MbData[110]);
109 Serial.print(",");
110 Serial.print(Mb.MbData[112]);
111 Serial.print(",");
112 Serial.println(Mb.MbData[114]);
113 Serial.println();
114 Serial.println();

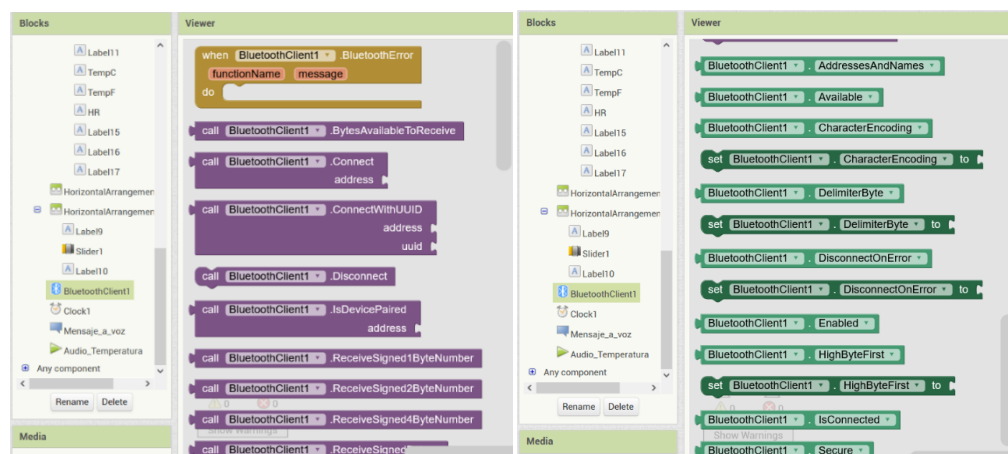
```

**Figura 48 - 2.** Creación de la trama de información enviada desde Arduino a la aplicación.

Realizado por: Pazmiño C. 2020

### 2.5.11 Comunicación Arduino con la app móvil

Dentro de App Inventor se debe acceder a las configuraciones de bluetooth, en el apartado de “Blocks” seleccionando “BluetoothClient1” aparecerá las opciones que se puede utilizar como se muestra en la figura 49 - 2.

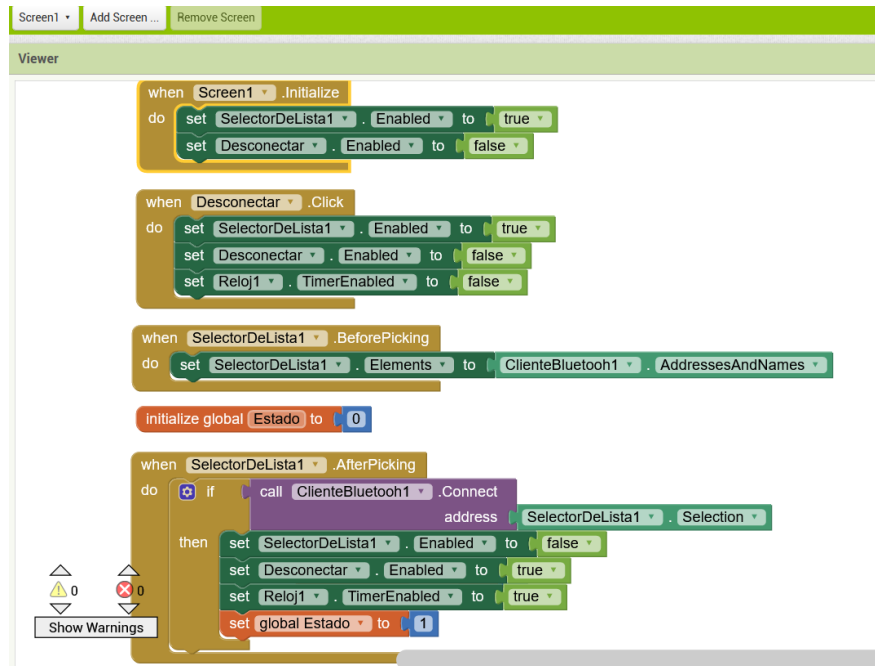


**Figura 49 - 2.** Menú de opciones bluetooth en App Inventor.

Realizado por: Pazmiño C. 2020

La programación de la aplicación inicia desde la carga de la pantalla principal, en esta se habilita la lista de dispositivos bluetooth vinculados al dispositivo móvil y deshabilita el botón de desconectar. El botón desconectar deshabilita el reloj que repite cíclicamente el código.

La lista de selección pregunta si está encendido el bluetooth y si lo está incluye en su lista los dispositivos vinculados al celular para habilitar el botón de desconectar e inicializar el reloj como se muestra en la figura 50 - 2.



**Figura 50 - 2.** Configuración inicial de la comunicación bluetooth.

Realizado por: Pazmiño C. 2020

La programación continúa con la declaración de las variables globales para guardar los datos obtenidos desde el Arduino para visualizar los datos, estos se encuentran en el apartado de “Blocks”, “Built-in” y “Variables”. Véase la figura 51 - 2.



**Figura 51 - 2.** Ubicación y declaración de variables.

Realizado por: Pazmiño C. 2020

Los bloques que se van a usar son: “Clock1” qué es el temporizador, en la parte de “Control” los condicionales “if” y “else”, también los conectores lógicos “and”, “or”, los condicionales: mayor que “>”, menor que “<”, mayor o igual “>=”, menor o igual “<=”, igual que “==” y diferente de “!=”. Los bloques mencionados que se utilizarán, se muestran en la figura 51 - 2.



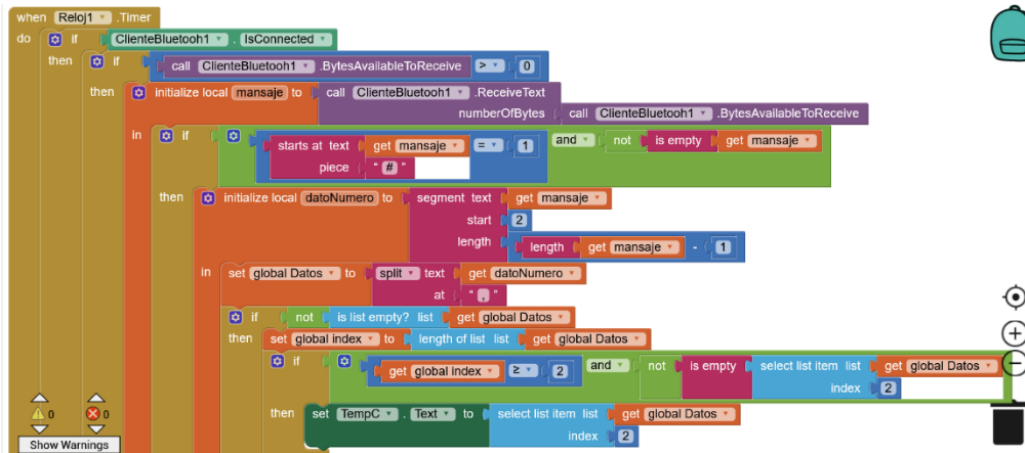
**Figura 52 - 2.** Ubicación de los bloques que se utilizarán en el programa.

Realizado por: Pazmiño C. 2020

Por lo tanto, el código empieza con el timer “Clock1” que crea el ciclo loop repetitivo, seguido de la condición que pregunta si la conexión con el dispositivo bluetooth está efectuada, a continuación, se asegura que la cadena de caracteres está llegando.

Cuando las condiciones están verificadas se guarda en una variable el mensaje que le envía el Arduino, después para que asegurarse que el mensaje llegue bien y claro se detecta el inicio del mensaje con el caracter “#”.

Cuando la información está verificada, elimina el primer caracter y guarda en otra variable el excedente del mensaje, seguidamente, se almacena en una lista los datos de interés, luego de haberlos aislados aplicando un split, detecta un caracter, en este caso es una coma “,” y cada dato separado por esta lo guarda en la lista mencionada se observan en la figura 53 - 2. los bloques que se utilizarán se encuentran en la figura 54 - 2



**Figura 53 - 2.** Código de lectura y pre procesamiento de datos.

Realizado por: Pazmiño C. 2020

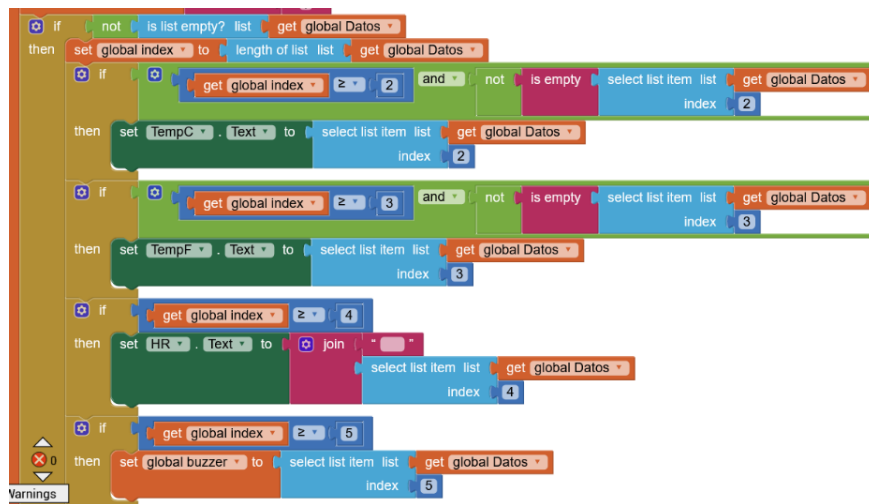
### 2.5.12 Lectura y visualización de datos enviados por el Arduino



**Figura 54 - 2.** Ubicación de los bloques requerido para procesar caracteres y listas

Realizado por: Pazmiño C. 2020

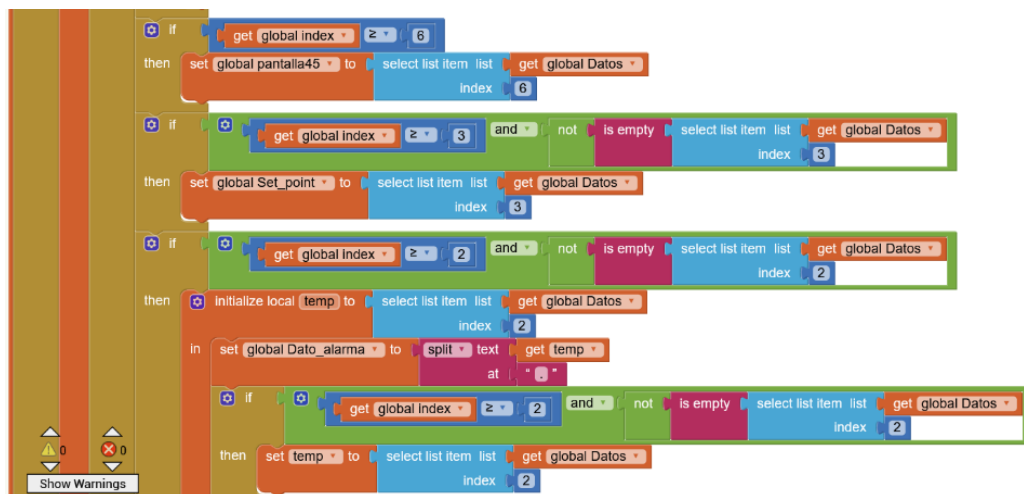
Cuando la información está pre procesada se debe preguntar si la variable se encuentra vacía, si no es así almacena en otra variable el número de elementos que existen dentro de la lista, y con este número se accede a la posición en la que se encuentra los datos que se desea visualizar en la aplicación como se muestra en la figura 55 - 2.



**Figura 55 - 2.** Visualización de la información en los labels.

Realizado por: Pazmiño C. 2020

Existen variables adicionales en el caso de enviar más información desde el Arduino a la aplicación, sin necesidad de modificar mayormente el código y finalmente se procesa el dato de la temperatura para mostrar en el slider y para emitir las alarmas a manera de notificaciones haciendo lectura de esta variable como se muestra en la figura 56 - 2.

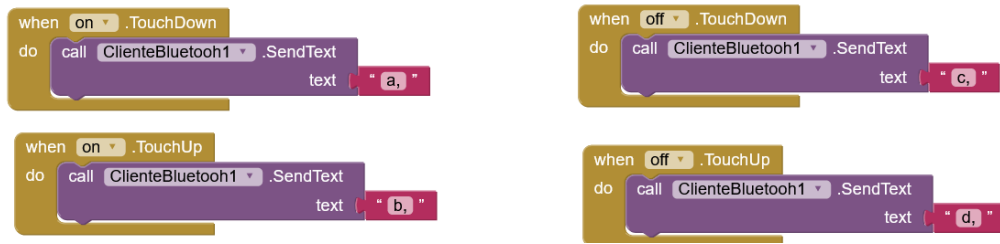


**Figura 56 - 2.** Procesamiento de la variable temperatura.

Realizado por: Pazmiño C. 2020

### 2.5.13 Transferencia de datos mediante bluetooth desde la aplicación móvil hacia Arduino

La información que envían cuando se pulsan los botones se muestran en la siguiente figura 57 - 2 y las acciones se reunieron en la tabla 3 - 2.

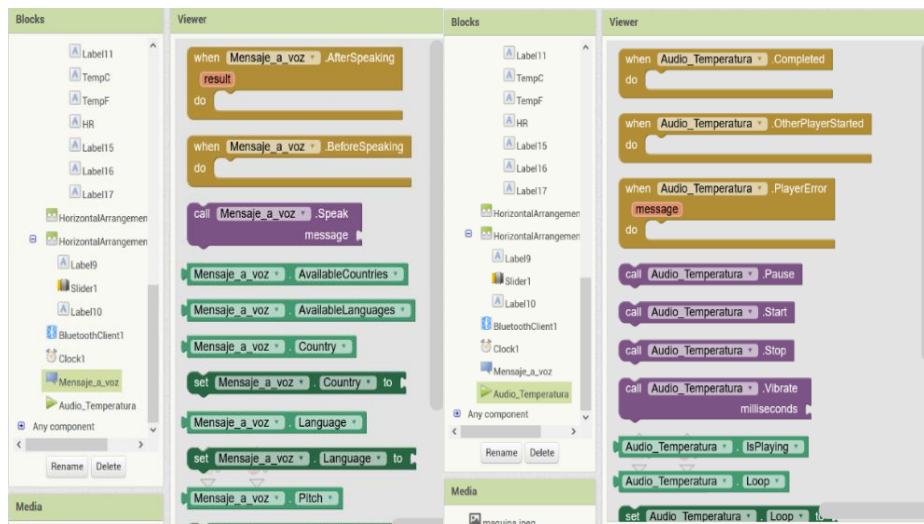


**Figura 57 - 2.** Caracteres enviados cuando se pulsán y se sueltan los botones.

Realizado por: Pazmiño C. 2020

### 2.5.14 Visualización de información en app móvil

Los bloques para notificar y crear las alertas vibratorias junto con la información que se encuentran dentro del LOGO, se acceden en los ítems de “Mensaje\_de\_voz” y “Audio\_Temperatura” como se observa en la figura 58 - 2.



**Figura 58 - 2.** Disponibilidad de herramientas para notificaciones vibratorias y de voz.

Realizado por: Pazmiño C. 2020

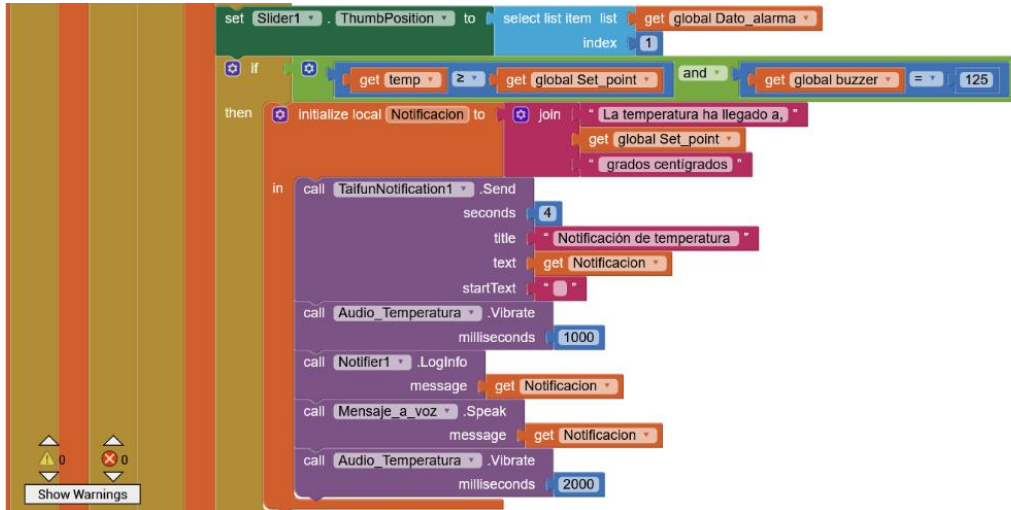
Las notificaciones se emiten en dos casos:

1. Cuando la temperatura actual del sensor llega a la configurada por el usuario.
2. Cuando luego del proceso anterior al bajar la temperatura llega a 45°.

Para el primer caso de notificación compara que la temperatura actual y la temperatura configurada sean iguales, además de analizar la variable global “buzzer” sea igual a 125. Cuando pasa la comparación de información se crea una cadena de caracteres “La temperatura ha llegado



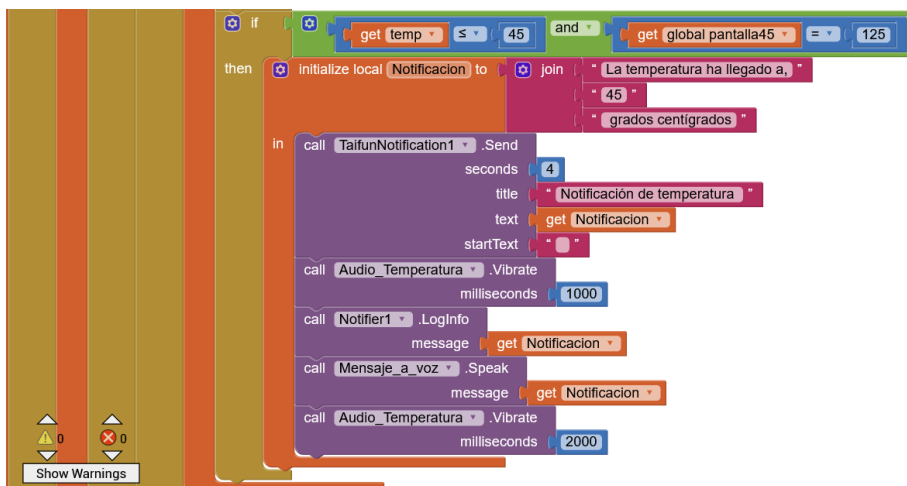
a,” acompañado de la temperatura configurada, y agregando las palabras “grados centígrados” con esto se pretende crear la frase que dirá la notificación con el bloque de “Mensaje\_a\_voz” y que vibre durante 3 segundos con el bloque “Audio\_Temperatura”, como se muestra en el código de la figura 59 - 2.



**Figura 59 - 2.** Código del primer caso de notificación.

**Realizado por:** Pazmiño C. 2020

El segundo caso se encarga de notificar cuando la temperatura baja a 45 °C luego de haber llegado a la temperatura configurada y la variable global “pantalla45” que se conecta con la memoria Modbus de Arduino es igual a 125. La alarma se dispara con un mensaje transformado a voz de “La temperatura ha llegado a 45 grados centígrados” paralelamente con la vibración del dispositivo por 3 segundos como se muestra en la figura 60 - 2.



**Figura 60 - 2.** Código del segundo caso de notificación.

**Realizado por:** Pazmiño C. 2020

### III. CAPÍTULO III - MARCO DE RESULTADOS

En el presente capítulo se presenta los resultados de las pruebas realizadas en el prototipo con el fin de demostrar su confiabilidad al momento de ser utilizado por los operadores del mismo, siendo estas pruebas las siguientes:

- Medición del tiempo de respuesta del encendido y apagado del prototipo de manera remota.
- Medición de la temperatura de la leche.
- Medición de los tiempos de respuesta mediante una alarma cuando la temperatura de la leche llega a su punto ideal para la pasteurización y posterior enfriado para su fermentación.

#### 3.1 Medición del tiempo de respuesta del encendido y apagado del prototipo de manera remota.

Los retardos de encendido y apagados remotos del prototipo para la realización de yogurt ayudan a conocer el tiempo que tarda en responder desde el instante en el que se presiona el botón desde la aplicación móvil, hasta el accionamiento del equipo, sea ante una emergencia para apagarlo o simplemente inicial el proceso de realización del yogurt.

Idealmente los retardos de respuesta deben ser los mismos, independientemente de la distancia a la que se encuentra el operador, pero en realidad este se ve afectado no solo por la distancia sino también por los obstáculos que puedan estar entre el dispositivo móvil y el equipo, esta comparativa se hace en la tabla 1 - 3.

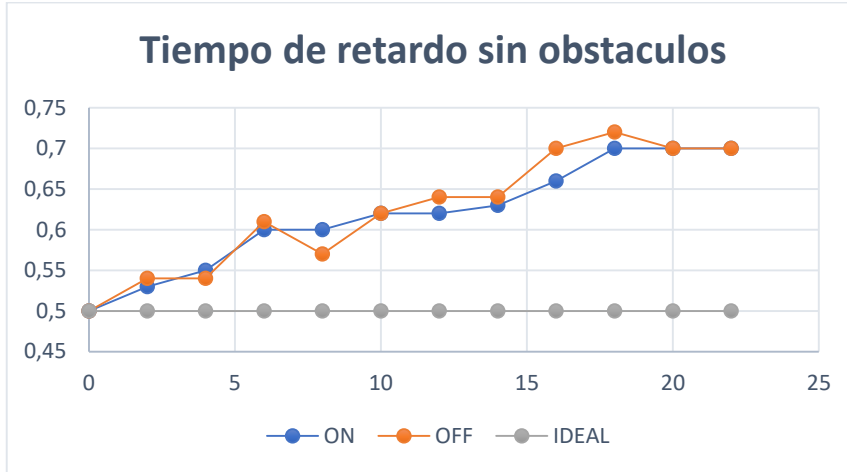
**Tabla 1 - 3:** Toma de medidas del tiempo de repuesta del prototipo en relación a la distancia entre el dispositivo móvil y el equipo

<b>Distancia (m)</b>	<b>ON (s)</b>	<b>OFF(s)</b>	<b>IDEAL(s)</b>
<b>0</b>	0.5	0.5	0.5
<b>2</b>	0.53	0.54	0.5
<b>4</b>	0.55	0.54	0.5
<b>6</b>	0.6	0.61	0.5
<b>8</b>	0.6	0.57	0.5
<b>10</b>	0.62	0.62	0.5
<b>12</b>	0.62	0.64	0.5
<b>14</b>	0.63	0.64	0.5
<b>16</b>	0.66	0.7	0.5
<b>18</b>	0.7	0.72	0.5
<b>20</b>	0.7	0.7	0.5

22	0.7	0.7	0.5
----	-----	-----	-----

Realizado por: Pazmiño C. 2020

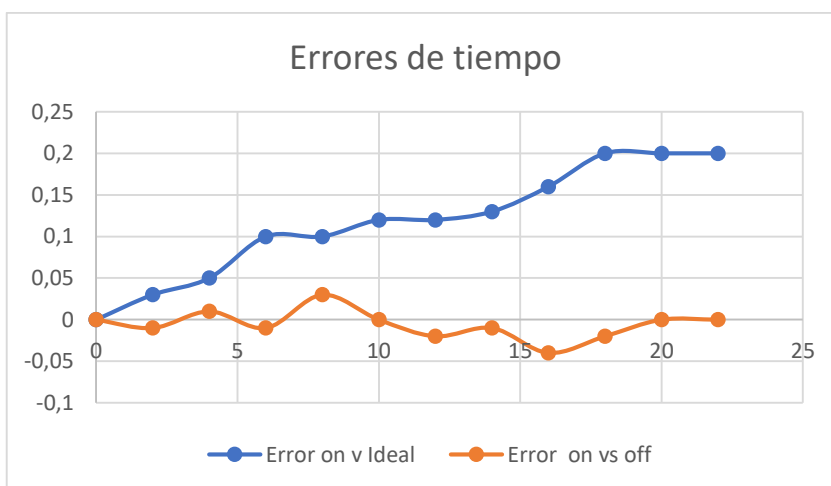
### 3.1.1 Tiempo de respuesta sin obstáculos



**Gráfico 1 - 3.** Tiempo de respuesta sin obstáculos.

Realizado por: Pazmiño C. 2020

En la gráfica 1 - 3 se observa cómo es directamente proporcional el tiempo de reacción del sistema con la distancia a la que se encuentra, este resultado sería un inconveniente si el proceso requiere de acciones inmediatas de parte del operador, debido a que el sistema no necesita de este tipo de acciones, sino simplemente monitoreo del estado del producto, se podría considerar que el sistema trabaja en tiempo real en esta aplicación, como se observa en la gráfica 2 – 3.



**Gráfico 2 - 3.** Errores entre tiempo real y tiempo ideal

Realizado por: Pazmiño C. 2020

La gráfica 2 - 3 se puede observar que el tiempo de encendido ideal vs el tiempo medido, así mismo como existe una ligera variación entre el tiempo de encendido y apagado, estos valores se ven reflejados en la tabla 2 – 3.

**Tabla 2 - 3:** Error de medida real y medida ideal del prototipo

<b>Error on vs Ideal</b>	<b>Error on vs off</b>
<b>0 s</b>	0 s
<b>0.03 s</b>	-0.01 s
<b>0.05 s</b>	0.01 s
<b>0.1 s</b>	-0.01 s
<b>0.1 s</b>	0.03 s
<b>0.12 s</b>	0 s
<b>0.12 s</b>	-0.02 s
<b>0.13 s</b>	-0.01 s
<b>0.16 s</b>	-0.04 s
<b>0.2 s</b>	-0.02 s
<b>0.2 s</b>	0 s
<b>0.2 s</b>	0 s
<b>PROMEDIO</b>	<b>PROMEDIO</b>
<b>0.1175 s</b>	<b>-0,00583333 s</b>

**Realizado por:** Pazmiño C. 2020

El error promedio entre el tiempo de encendido ideal y el real es 0.1175s. Que en relación con el tiempo de espera del sistema para cambiar de etapa del proceso de elaboración de yogurt que varía entre 30 minutos y 3 horas. El prototipo no requiere un monitoreo constante, pero si saber el estado de este durante el proceso con un tiempo de respuesta inmediato.

El error entre el tiempo de encendido y apagado no es grande por lo que se puede concluir que el sistema responde de forma similar al conmutar de un estado a otro. Estos resultados que se obtienen del prototipo son visibles en la pantalla del dispositivo móvil como se muestra en la figura 1 – 3.



**Figura 1 - 3.** Comparación app móvil e interfaz del sistema

**Realizado por:** Pazmiño C. 2020

Los datos pueden ser visibles tanto en la pantalla del prototipo como en la aplicación móvil permitiendo un permanente monitoreo e intervenir durante el proceso.

### **3.2 Medición de la temperatura de la leche**

La medición de la temperatura de la leche dentro del prototipo, se realizó mediante un termómetro de mercurio que lo realiza manualmente el operador, para saber el momento del ciclo para la elaboración del yogurt, como se muestra en la figura 2 - 3.



**Figura 2 - 3.** Toma de medida manual de temperatura

**Realizado por:** Pazmiño C. 2020

Para el prototipo se realizó una serie de pruebas de temperatura con el sensor pt100 y el sensor digital Fluke, para medir la temperatura de la leche, señal que es llevada al módulo del LOGO!8 y sirve para comparar con la medida realizada por el termómetro de mercurio, cuyos datos se presentan en la tabla 3 – 3. dedicado a la toma de temperatura.

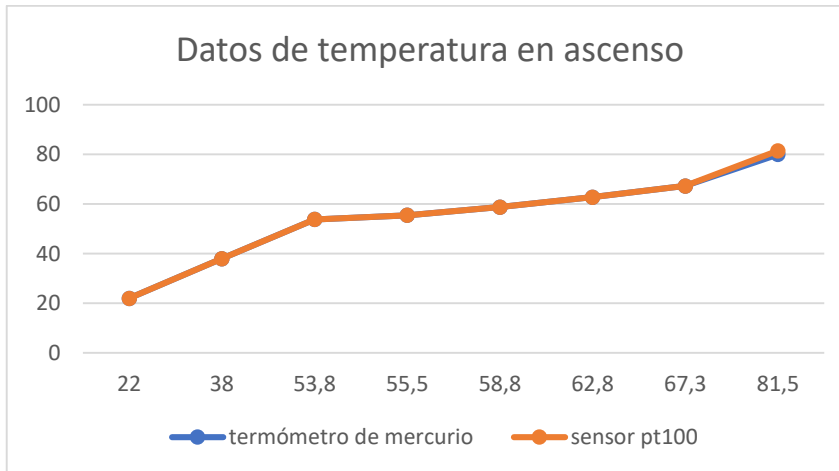
**Tabla 3 - 3:** Medidas de temperatura con el sensor pt100 y Fluke

<b>Estado de la Máquina</b>	<b>Termómetro de mercurio</b>	<b>Sensor pt100</b>	<b>Pirómetro Fluke</b>
<b>Temperatura inicial de la leche</b>	22 °C	22 °C	22 °C
<b>Temperatura con agitador encendido</b>	38 °C	38 °C	37.2 °C
	53.8 °C	53.8 °C	47.5 °C
	55.5 °C	55.5 °C	53.2 °C
	58.8 °C	58.8 °C	56 °C
	62.8 °C	62.8 °C	62.8 °C
	67.3 °C	67.3 °C	71 °C
<b>Temperatura deseada</b>	80 °C	81 °C	82 °C
<b>Temperatura en reposo en descenso</b>	62.5 °C	62.5 °C	62 °C
	57 °C	57 °C	56 °C
	53 °C	53 °C	53 °C
	49 °C	49 °C	48 °C

**Realizado por:** Pazmiño C. 2020

La medida de temperatura se adquirió al momento en que está en ascenso como se muestra en la gráfica 3 - 3. En el proceso se adicionó una medida de un pirómetro digital para comparar en sus diferentes estados. Estas medidas se tomaron de forma controlada y continua.

**Gráfico 3 - 3.** Toma de datos temperatura en ascenso.



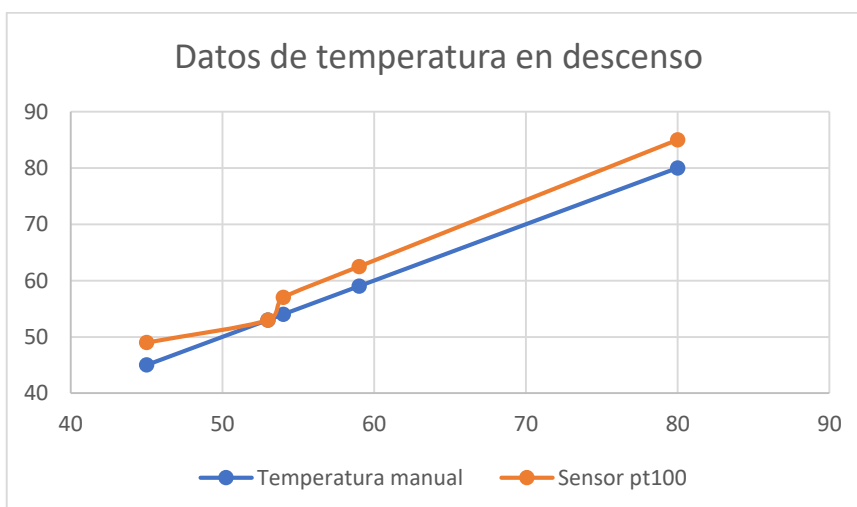
**Realizado por:** Pazmiño C. 2020

En la gráfica se distingue que la temperatura adquirida de mediante el sensor pt100 está conectado al módulo de ampliación LOGO AM2 RTD, varía durante el proceso al momento de llegar a la temperatura deseada, por lo que se concluye que los valores coinciden en su gran mayoría con las mediciones manuales con el termómetro de mercurio.

Por este motivo no es necesario realizar una tabla comparativa del error ya que las medidas tomadas de forma manual coinciden con las adquiridas con el LOGO con su módulo de aplicación.

También se tomaron las medidas cuando la temperatura desciende, debido a que en ocasiones el sensor no tiene una respuesta lineal, y la histéresis hace que los resultados varíen como se muestra en gráfica 4-3.

**Gráfico 4 - 3.** Toma de datos temperatura en descenso.



**Realizado por:** Pazmiño C. 2020

La medición de temperatura con el sensor varía levemente con la manual debido a este efecto de histéresis, pero su respuesta es casi lineal por lo que se puede deducir que las medidas obtenidas son confiables y tienen un error del 6.25% al momento de dejar al proceso en reposo.

### 3.3 Medición del tiempo de respuesta al notificar una alarma

La importancia de saber los tiempos de respuesta durante el proceso de la pasteurización es primordial debido a que el prototipo necesita de la intervención del operador de la máquina. Los datos del tiempo de respuesta se ven reflejados en la tabla

**Tabla 4 - 3:** Tiempo de respuesta de notificación

TIEMPO (S)	TEMPERATURA	
	80°C	45°C
PRUEBA 1	0,6 s	0,58 s
PRUEBA 2	0,5 s	0,7 s
PRUEBA 3	0,6 s	0,53 s
PRUEBA 4	0,65 s	0,6 s
PRUEBA 5	0,55 s	0,65 s
PRUEBA 6	0,62 s	0,7 s
PRUEBA 7	0,64 s	0,5 s
PRUEBA 8	0,7 s	0,6 s
PRUEBA 9	0,58 s	0,59 s
PRUEBA 10	0,59 s	0,52 s
<b>PROMEDIO</b>	<b>0,603 s</b>	<b>0,597 s</b>

**Realizado por:** Pazmiño C. 2020

Se puede observar que cuando el prototipo llega a los 80°C la notificación tiene un tiempo de respuesta de 0.603 segundos en promedio, debido a que este tiempo se encuentra en el rango de los milisegundos es un tiempo de respuesta inmediato durante el proceso, de igual forma al momento que el sistema llega a 45°C el tiempo de respuesta promedio es 0.597 s.

Los tiempos de respuesta de las notificaciones no tienen una variación considerable por lo que están trabajando dentro del mismo rango de tiempo y tienen un error de 0.006 s.



## CONCLUSIONES

Se investigó y se analizó las características y funcionalidad de las plataformas de código abierto Android y Arduino que se utilizó para la implementación del prototipo siendo compatibles para el monitoreo de la producción de yogurt.

Se determinó que el sistema de comunicación más adecuado entre Android y Arduino es el protocolo serial implementado mediante bluetooth, debido a que cumple con los requerimientos de alcance y rapidez de respuesta que requiere el prototipo para envío y recepción de datos en tiempo de respuesta inmediato.

Se diseñó una aplicación compatible con Android, la cual permite visualizar los datos del proceso que requiere la elaboración del yogurt como: temperatura hasta llegar al setpoint y tiempo transcurrido, además permite controlar el arranque y parada del motor de la máquina.

Se realizó las pruebas de funcionamiento y se determinó que la velocidad de repuesta es directamente proporcional a la distancia entre el prototipo y el usuario, obteniendo niveles óptimos de respuesta hasta los 22 metros de distancia en 0.7 segundos y el mensaje de alerta cuando ha llegado a set point en 0.603 segundos y la temperatura de enfriamiento para el proceso de fermentación en la elaboración de yogurt en 0.597 segundos.

## **RECOMENDACIONES**

Para una mejor comunicación entre dispositivos Arduino y LOGO!8 mediante protocolo Modbus RTU se debe considerar la versión, ya que el LOGO!8.2 es la única compatible con este protocolo.

Para que se pueda realizar el monitoreo en el proceso de la elaboración del yogurt la aplicación debe estar instalada en el dispositivo móvil y activada la conexión Bluetooth para la interacción entre el prototipo y la persona encargada.

Se recomienda utilizar el prototipo dentro de un rango de 22 metros, para evitar pérdidas de información de la medición de temperatura y evitar retardos en la comunicación entre el dispositivo Android y el prototipo.

El prototipo debe estar ubicado en un lugar con la menor cantidad de obstáculos para poder monitorear de mejor manera el proceso que requiere la elaboración del yogurt y así no afecte el desempeño del mismo.

Se recomienda que el dispositivo Android donde se va a recibir las notificaciones tenga una configuración de volumen adecuado para que sean escuchadas y atendidas por la persona encargada del proceso de elaboración de yogurt.

## **BIBLIOGRAFÍA**

“¿Qué es un Proceso Industrial?”. Ingeniería Industrial [en línea], 2019. [Consulta: 5 septiembre 2020]. Disponible en: <https://www.ingenieriaindustrialonline.com/procesos-industriales/que-es-un-proceso-industrial/>.

“Introducción a las comunicaciones a través de la red eléctrica”. CCM [en línea], 2013. [Consulta: 7 septiembre 2020]. Disponible en: <https://es.ccm.net/contents/122-introduccion-a-las-comunicaciones-a-traves-de-la-red-electrica>.

**WILSON**. “Procesos Agroindustriales: PRODUCCION DEL YOGUR EN EL ECUADOR”. Procesos Agroindustriales [en línea], 2012. [Consulta: 11 septiembre 2020]. Disponible en: <http://wilsonproces.blogspot.com/2012/10/produccion-del-yogur-en-el-ecuador.html>.

**SANCHIS LLOPIS, Roberto; ROMERO PÉREZ, Julio**. Automatización industrial. 2019.

“¿Qué son los Sensores Industriales?”. Bricos [en línea], 2013. [Consulta: 5 octubre 2020]. Disponible en: <https://bricos.com/2013/02/sensores/>.

**FRADEN, Jacob**. Handbook of modern sensors: physics, designs, and applications. Springer Science & Business Media, 2004.

**ARAGONÉS, Jesús Bausà, et al**. Sensores de temperatura. Recuperado el, 2003, vol. 3, p. 2003-04.

**MANDADO PÉREZ, Enrique; MARCOS ACEVEDO, Jorge; FERNÁNDEZ SILVA, Celso**. Automatas programables y sistemas de automatizacion/PLC and Automation Systems. Marcombo, 2009.

**TIEGELKAMP, Michael; KARL-HEINZ, John**. IEC 61131-3: Programming industrial automation systems. Berlin: Springer, 1995.

**MATEOS, F**. Automatas programables. Visión general. Oviedo: Universidad de Oviedo, 2004.

Catálogo Siemens 2020 (SIEMENS, Catálogo. Sistema de automatización, S7-200. Junio de, 2004.)

“Controladores Industriales Inteligentes”. [en línea], [Consulta: 3 octubre 2020]. Disponible en: [http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion\\_de\\_referencia\\_ISE6\\_1\\_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE6_1_1.pdf)

“LOGO! Software”. Siemens Global Website [en línea], 2019. [Consulta: 7 octubre 2020]. Disponible en:

<https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-software.html>.

“Protocolos de comunicación industriales”. Logicbus [en línea], 2019. [Consulta: 11 octubre 2020]. Disponible en: <https://www.logicbus.com.mx/blog/protocolos-de-comunicacion-industriales/>.

**OLAYA, Andrés F. Ruiz; LÓPEZ, Asfur Barandica; MORENO, Fabio G. Guerrero.** Implementación de una red MODBUS/TCP. Ingeniería y Competitividad, 2004, vol. 6, no 2, p. 35-44.

**DE SIQUEIRA, Thiago Senador.** Bluetooth—Características, protocolos e funcionamiento. 2006.

“¿Qué es Interfaz de usuario?” [en línea], 2017. [Consulta: 13 octubre 2020]. Disponible en: <https://www.workana.com/i/glosario/interfaz-de-usuario/>.

**CRAWFORD POKRESS, Shaileen; DOMINGUEZ VEIGA, José.** MIT App Inventor: Enabling personal mobile computing. [blog]. [Consulta: 14 octubre 2020]. Disponible en: <https://arxiv.org/abs/1310.2830>

“INVENTOR MIT App”. M. I. T. App Inventor. [en línea]. 2017. [Consulta: 13 octubre 2020]. Disponible en: <http://appinventor.mit.edu/explore/>

“Lácteo”. Wikipedia, la enciclopedia libre [en línea], 2020. [Consulta: 18 octubre 2020]. Disponible en: <https://es.wikipedia.org/w/index.php?title=L%C3%A1cteo&oldid=130610038>.

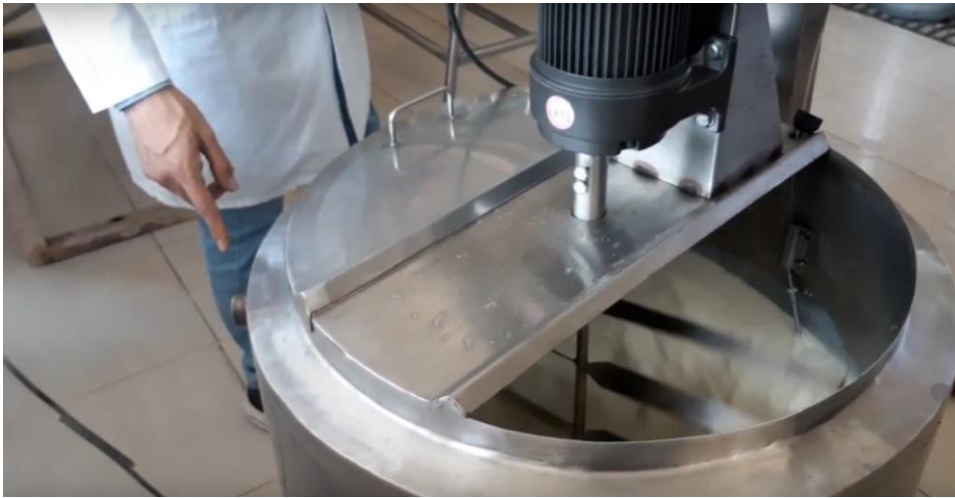
“Producción y productos lácteos: Tipos y características”. [en línea], 2012. [Consulta: 19 octubre 2020]. Disponible en: <http://www.fao.org/dairy-production-products/products/tipos-y-caracteristicas/es/>.

BONET, B., et al. Leche, Nata, Mantequilla y otros productos lácteos. 2005.

**HUERTAS, Ricardo A.** Yogur en la salud humana. Revista Lasallista de investigación, 2012, vol. 9, no 2, p. 162-177.

## ANEXOS

### ANEXO A: ELABORACIÓN DE YOGURT



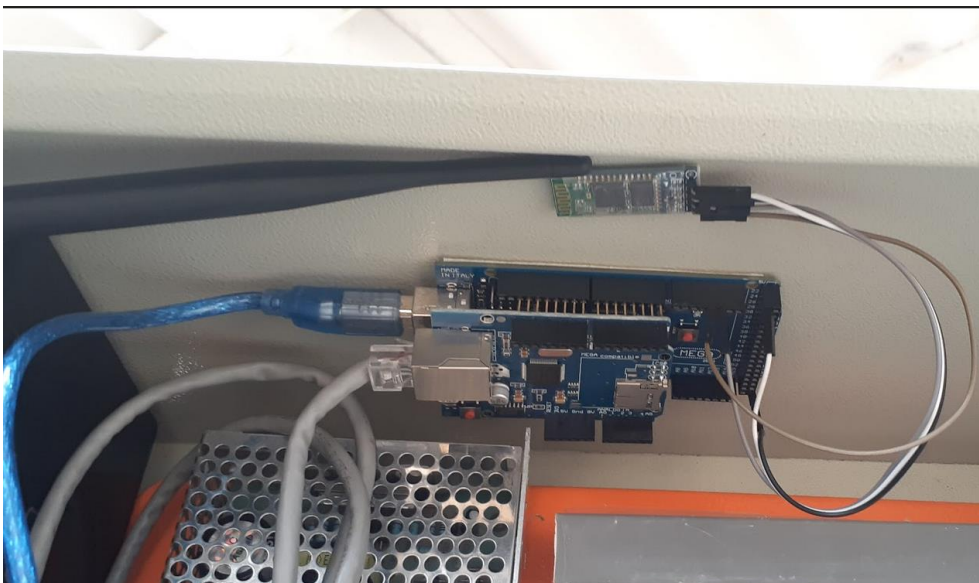
**ANEXO B: TOMA DE MEDICIONES**





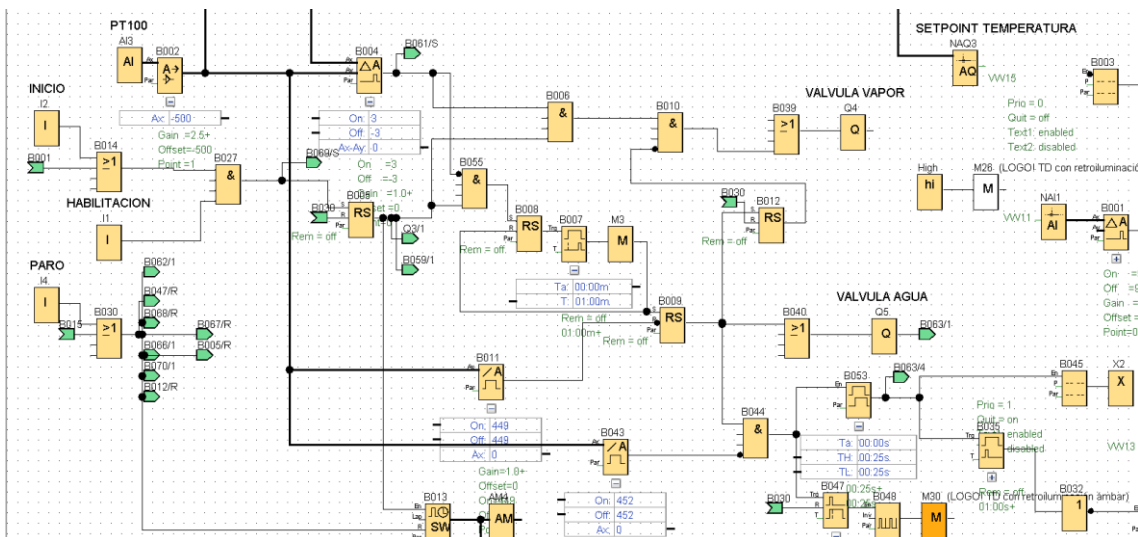
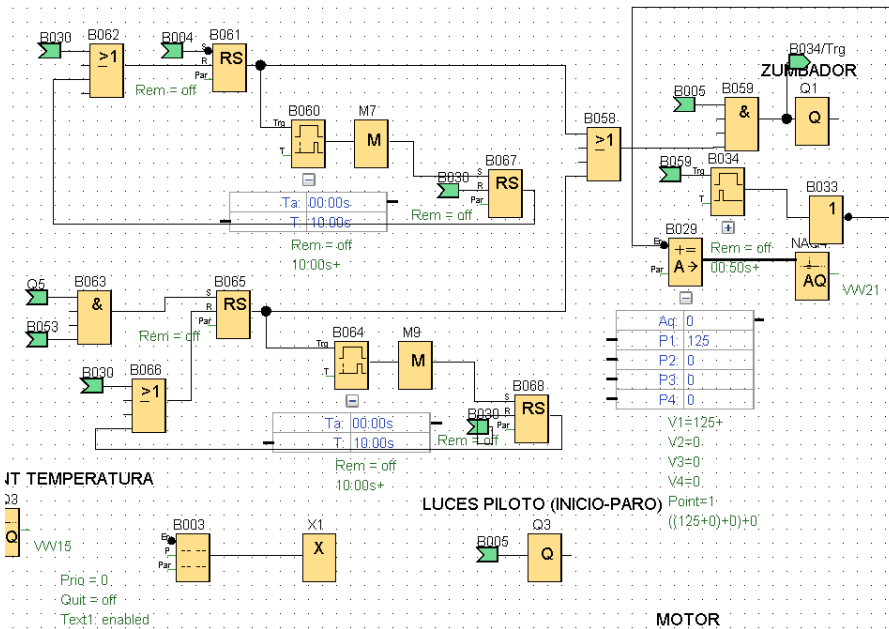
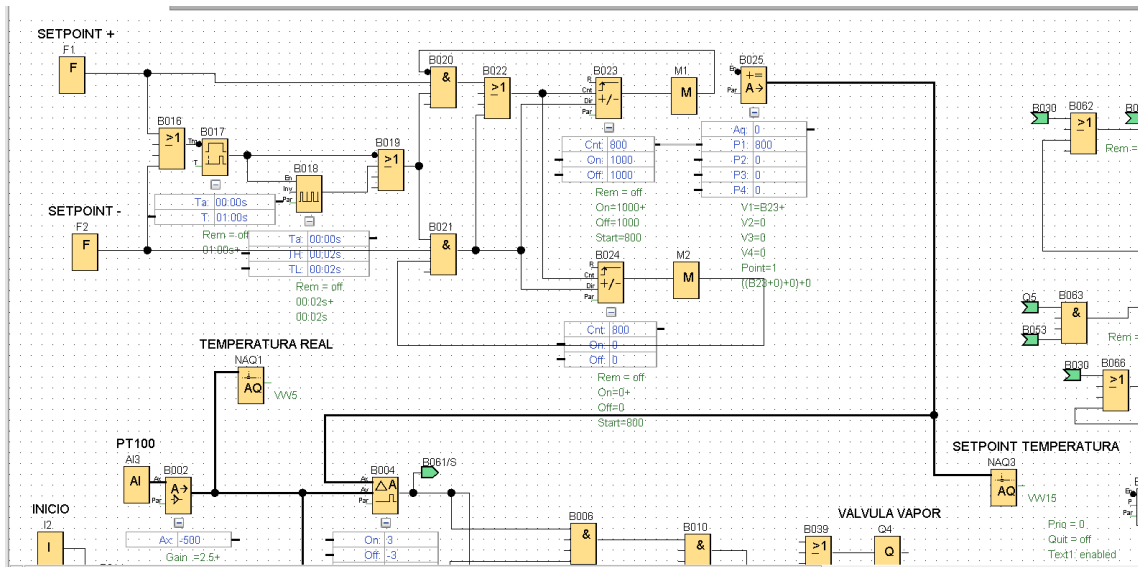


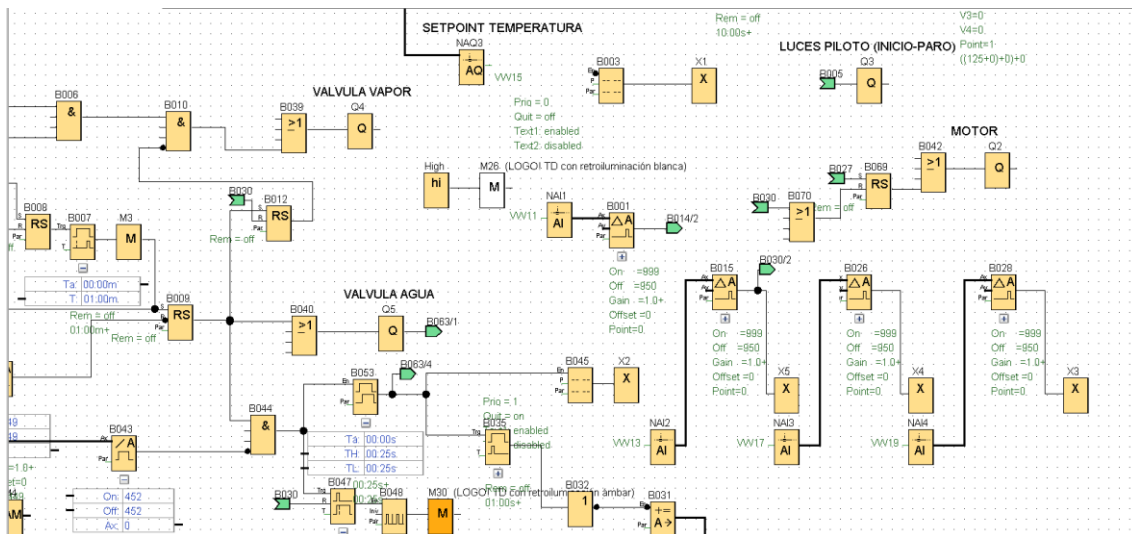
## ANEXO C: TABLERO DE CONTROL ELÉCTRICO





# ANEXO D: PROGRAMACIÓN DEL SISTEMA DE MONITOREO



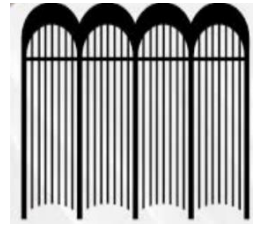




**ANEXO F: PÁGINA DE REVISIÓN DEL TRABAJO ESCRITO DE TITULACIÓN.**



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS**  
**PARA EL APRENDIZAJE Y LA INVESTIGACIÓN**



**UNIDAD DE PROCESOS TÉCNICOS**

Fecha de entrega: 05/04/2021

<b>INFORMACION DEL AUTOR/A(S)</b>
Carlos Gonzalo Pazmiño Coloma
<b>INFORMACIÓN INSTITUCIONAL</b>
FACULTAD: Facultad de Informática y Electrónica
CARRERA: Ingeniería en Electrónica, Control y Redes Industriales
TÍTULO A OPTAR: Ingeniero en Electrónica, Control y Redes Industriales
f. ANALISTA DE BIBLIOTECA RESPONSABLE