



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“TELECONTROL DE UN SISTEMA DE CONSERVACIÓN  
DE TANQUE DE LECHE”**

**TESINA DE SEMINARIO**

PREVIO A LA OBTENCIÓN DEL TÍTULO DE

**INGENIERO EN TELEMÁTICA**

Presentado por:

**HERNÁNDEZ RODRÍGUEZ CRISTHIAN XAVIER  
CEDILLO ARCE JOSSELYN MAOLY**

**GUAYAQUIL – ECUADOR**

**AÑO 2015**

## AGRADECIMIENTO

Doy gracias a Dios, quien es mi fortaleza en mi debilidad, a mis padres, por todo lo que han hecho por mí desde el día en que nací, a mis hermanos, con quienes siempre he contado, y a mi enamorada por su apoyo incondicional. Gracias a todos por creer en mí.

***Cristhian Xavier Hernández Rodríguez***

Agradezco a Dios, quien guía mi vida y en quien pongo mi futuro, a mis padres por todo el apoyo al darme la herencia más importante como es la educación, a mis hermanos y a mi compañero de proyecto por su apoyo y paciencia.

***Josselyn Maoly Cedillo Arce***

## DEDICATORIAS

Dedicado a Dios y a mi madre, porque sin ella no me encontraría aquí escribiendo este trabajo, por ser la mejor mamá del mundo y apoyarme a lo largo de toda mi vida. Por todo el sacrificio que has hecho por mí, esto es para tí. Gracias mamá.

***Cristhian Xavier Hernández Rodríguez***

A mi Divino niño, es él quien guía mi vida e ilumina siempre. Dedicado a mis padres por su apoyo y por confiar en mí, a esa persona especial que a estado conmigo a lo largo de toda mi vida universitaria y ha a sabido comprenderme RL.

***Josselyn Maoly Cedillo Arce***

## **TRIBUNAL DE SUSTENTACIÓN**

---

Msc. Marcos Millán Traverso

PROFESOR DEL SEMINARIO DE GRADUACIÓN

---

Ing. Miguel Molina Villacís.

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”. (Reglamento de exámenes y títulos profesionales de la ESPOL).

---

Cristhian Xavier Hernández Rodríguez

---

Josselyn Maoly Cedillo Arce

## RESUMEN

En este documento se presenta el prototipo de un sistema de telecontrol para un tanque de conservación de leche, utilizando software y hardware libre de bajo costo y fácil implementación.

Para el desarrollo del prototipo se utiliza un sensor ultrasónico, el cual permite a través de cálculos matemáticos obtener el volúmen de leche contenido en un tanque cilíndrico, un sensor de temperatura, el cual permite controlar el rango de temperatura al que debe mantenerse la leche para su debida conservación.

El sistema de telecontrol incorpora un servidor web, el cual permite monitorear el estado de las variables medidas por los sensores y enviar ordenes al microcontrolador para manipular el encendido o apagado de actuadores. El acceso a la interfaz web se realiza mediante una autenticación de usuario.

Para la implementación del servidor web y la interacción de los sensores y actuadores, se utiliza una tarjeta microcontroladora de la familia Intel, la cual cuenta con un puerto ethernet que permite la comunicación utilizando el protocolo TCP/IP.

Finalmente, el sistema es probado en un ambiente de laboratorio y en maqueta, creando una red LAN para realizar la transmisión y recepción de datos, además mostrando la conexión de todos los componentes para su correcto funcionamiento.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIAS .....	III
TRIBUNAL DE SUSTENTACIÓN .....	IV
DECLARACIÓN EXPRESA .....	V
RESUMEN.....	VI
ÍNDICE GENERAL.....	VIII
ABREVIATURAS .....	XIII
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABLAS .....	XVII
INTRODUCCIÓN .....	XVIII
CAPÍTULO 1 .....	1
1. GENERALIDADES.....	1
1.1. ANTECEDENTES.....	1
1.2. OBJETIVOS .....	3
1.2.1. OBJETIVO GENERAL.....	3
1.2.2. OBJETIVOS ESPECÍFICOS .....	3

1.3. ALCANCE Y LIMITACIONES .....	4
1.3.1. ALCANCE .....	5
1.3.2. LIMITACIONES.....	6
1.4. RESULTADOS ESPERADOS .....	7
1.5. APLICACIONES .....	8
CAPÍTULO 2 .....	9
2. MARCO TEÓRICO.....	9
2.1. FUNDAMENTACIÓN TEÓRICA .....	10
2.1.1. AUTOMATIZACIÓN.....	10
2.1.2. SISTEMAS EMBEBIDOS.....	10
2.1.3. SENSORES.....	12
2.1.4. ACTUADORES.....	12
2.1.5. SERVIDOR WEB .....	13
2.1.6. CAPA DE CONEXIÓN SEGURA SSL .....	14
2.1.7. CERTIFICADO CSR.....	14
2.2. ESPECIFICACIÓN HARDWARE DE LOS COMPONENTES.....	15
2.2.1. TARJETA INTEL GALILEO GEN 1 .....	15
2.2.1.1. CONECTOR ETHERNET 10/100.....	17
2.2.1.2. CONECTORES GPIO .....	17
2.2.2. MÓDULO LCD 16x2.....	18

2.2.3. SENSOR ULTRASÓNICO HC-SR04 .....	19
2.2.4. SENSOR DE TEMPERATURA LM-35 .....	20
2.2.5. ENRUTADOR .....	21
2.3. PROTOCOLOS Y CONCEPTOS DE COMUNICACIÓN USADOS.	21
2.3.1. PROTOCOLO TCP/IP.....	22
2.3.2. PROTOCOLO HTTP.....	23
2.4. ESPECIFICACION DEL SOFTWARE UTILIZADO. ....	23
2.4.1. IDE ARDUINO PARA INTEL GALILEO .....	23
2.4.2. SUBLIME TEXT .....	24
2.4.3. PHP .....	25
2.4.4. APACHE .....	26
2.4.5. MYSQL SERVER.....	26
2.4.6. OPEN SSL.....	27
CAPÍTULO 3.....	28
3. DISEÑO E IMPLEMENTACIÓN.....	28
3.1. DISEÑO GENERAL .....	28
3.1.1. ESQUEMÁTICO .....	29
3.1.2. LISTA DE RECURSOS UTILIZADOS .....	29
3.1.3. DIAGRAMA DE BLOQUES.....	31
3.1.4. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL SISTEMA.....	33

3.1.5. DIAGRAMA FUNCIONAL .....	37
3.2. IMPLEMENTACIÓN DEL HARDWARE EN LA TARJETA INTEL	
GALILEO GEN 1 .....	43
3.2.1. CREACIÓN DEL PROYECTO EN EL IDE ARDUINO .....	43
3.2.2. ASIGNACIÓN DE PINES EN LA TARJETA .....	44
3.3. CONFIGURACIÓN DEL SERVIDOR WEB EN LA TARJETA INTEL	
GALILEO GEN 1 .....	46
3.4. DESARROLLO DE LA APLICACIÓN WEB .....	48
3.5. COSTOS DE IMPLEMENTACIÓN.....	57
3.5.1. COSTO DE LA ELABORACIÓN DE LA MAQUETA.....	57
3.5.2. COSTO ESTIMADO DE IMPLEMENTACIÓN EN AMBIENTE REAL.....	58
3.6. BOSQUEJO GENERAL DE LA UBICACIÓN DEL SISTEMA.....	59
CAPÍTULO 4 .....	61
4. PRUEBAS Y RESULTADOS .....	61
4.1. PRUEBAS PREVIAS A LA IMPLEMENTACIÓN DEL PROTOTIPO.....	62
4.1.1. CONEXIÓN SSH .....	62
4.1.2. SENSOR ULTRASÓNICO HC-SR04.....	62
4.1.3. SENSOR DE TEMPERATURA LM35.....	64

4.2. SERVIDOR WEB IMPLEMENTADO EN LA PLACA INTEL GALILEO	64
4.2.1. PRUEBAS DE INTERACCIÓN CLIENTE-SERVIDOR.....	65
4.3. PRUEBAS DESPUES DE LA IMPLEMENTACIÓN DEL PROTOTIPO EN MAQUETA.....	67
4.3.1. INESTABILIDAD DEL SENSOR ULTRASÓNICO .....	67
4.4. VERIFICACIÓN DE PROCESOS .....	67

## CONCLUSIONES Y RECOMENDACIONES

## APÉNDICES

CÓDIGO FUENTE IDE ARDUINO

CÓDIGO HTML DE PÁGINA WEB

DIAGRAMA DE RECURSOS UTILIZADOS

## BIBLIOGRAFÍA

## BIBLIOGRAFÍA DE IMAGENES

## ABREVIATURAS

AC/DC	Corriente Alterna/Corriente Directa
ASCII	Código Americano Estándar para el Intercambio de Información
CSR	Solicitud de firma de certificado
GLP	Licencia Pública General de GNU
GNU	GNU No es Unix
GPIO	Entrada/Salida de Propósito General
HTTP	Protocolo de Transferencia de Hipertexto
IDE	Ambiente de Desarrollo Integrado
IP	Protocolo de Internet
LAN	Red de área local
LCD	Pantalla de cristal líquido
MAC	Control de Acceso al Medio
OSI	Interconexión de Sistemas Abiertos

PWM	modulación por ancho de pulsos
SSL	Capa de conexión segura
TCP	Protocolo de Control de Transmisión
USB	Bus universal en serie.

## ÍNDICE DE FIGURAS

Figura 2.1 – Sistema embebido [1].....	11
Figura 2.2 – Sensores [2] .....	12
Figura 2.3 – Actuadores [3] .....	13
Figura 2.4 – Tarjeta Intel Galileo [4] .....	16
Figura 2.5 – Conector RJ-45 [5] .....	17
Figura 2.6 – E/S Digitales y Analógicas [4].....	18
Figura 2.7 – Módulo LCD 16X2 [6] .....	19
Figura 2.8 – Sensor Ultrasónico [7] .....	20
Figura 2.9 – Sensor de temperatura [8].....	20
Figura 2.10 – Enrutador [9].....	21
Figura 2.11 – Modelo TCP/IP [10] .....	22
Figura 2.12 – Pantalla IDE Arduino .....	24
Figura 2.13 – Sublime text.....	25
Figura 3.1 – Esquemático de la estructura del sistema .....	29
Figura 3.2 – Diagrama de bloques del sistema. ....	31
Figura 3.3 – Diagrama de flujo de ingreso a la interfaz web.....	38
Figura 3.4 – Diagrama de flujo de la transmisión ente cliente-servidor. ...	39
Figura 3.5 – Diagrama de flujo del proceso de agitación. ....	40
Figura 3.6 – Diagrama de flujo del proceso de enfriamiento. ....	41

Figura 3.7 – Diagrama de flujo del proceso de lavado. ....	42
Figura 3.8 – IDE Arduino. ....	43
Figura 3.9 – Requerimiento HTTP[11]. ....	46
Figura 3.10 – Certificación SSL[12]. ....	47
Figura 3.11 – Funcionamiento de SSL[13]. ....	48
Figura 3.12 – Ingreso a página web. ....	49
Figura 3.13 – Ventana de inicio. ....	50
Figura 3.14 – Ventana de telemetría. ....	51
Figura 3.15 – Ventana de telecontrol. ....	52
Figura 3.16 – Ventana de administrar usuarios. ....	52
Figura 3.17 – Ventana creación de usuarios. ....	53
Figura 3.18 – Modificar usuario. ....	54
Figura 3.19 – Eliminar usuario. ....	54
Figura 3.20 – Ventana generar reportes. ....	55
Figura 3.21 – Ventana parametrizar. ....	56
Figura 3.22 – Ubicación del tanque vista lateral. ....	59
Figura 3.23 – Tanque de conservación. ....	59
Figura 3.24 – Vista superior de la ubicación del tanque. ....	60
Figura 4.1 – Código para sensor ultrasónico. ....	63

## ÍNDICE DE TABLAS

TABLA 3.1 – ASIGNACIÓN DE PINES DIGITALES.....	44
TABLA 3.2 - ASIGNACIÓN DE PINES ANALÓGICOS.....	45
TABLA 3.3 – COSTO DE ELABORACIÓN DE LA MAQUETA .....	57
TABLA 3.4 - COSTO ESTIMADO DE IMPLEMENTACIÓN EN AMBIENTE REAL. ....	58
TABLA 4.2 – VERIFICACIÓN DE PROCESOS.....	68

## INTRODUCCIÓN

El Ecuador es un país dedicado en gran parte a la actividad ganadera, la producción de leche en Ecuador mueve alrededor de 700 millones de dólares al año dentro de la cadena primaria.

El desarrollo de este prototipo se da ante el bajo nivel de tecnificación en el proceso de conservación de la leche con los que actualmente cuenta el sector agropecuario, se presenta una mejora significativa la cual beneficie a pequeños productores aumentando sus ingresos, al contar con un sistema de automatización de bajo costo, al ser implementado con tecnología OpenSource.

El documento esta dividido en 4 capítulos, para explicar de mejor manera cómo se elaboró este prototipo:

Capítulo 1, se detallan los objetivos generales y específicos, además los alcances y limitaciones de este prototipo.

Capítulo 2, se da a conocer los conceptos teóricos básicos del hardware, software y protocolos usados, necesarios para implementar el prototipo.

Capítulo 3, se explica el diseño del prototipo con un esquemático y un diagrama de bloques, además de ir detallando los pasos necesarios para su configuración, implementación y su respectivo funcionamiento.

Capítulo 4, se muestran las pruebas y los resultados obtenidos, además de elaborar las conclusiones y las recomendaciones pertinentes que se deben tomar en cuenta al momento de desarrollar este prototipo.

# **CAPÍTULO 1**

## **1. GENERALIDADES**

En este capítulo se detalla los precedentes que fueron tomados en cuenta para el diseño del prototipo, los alcances del mismo, sus limitaciones, los resultados esperados y su respectiva aplicación.

### **1.1. ANTECEDENTES**

“La producción de leche en Ecuador mueve alrededor de 700 millones de dólares al año dentro de la cadena primaria. Mientras que en toda la cadena, que incluye transporte, industrialización, comercialización, entre otros aspectos, se manejan más de 1.000 millones de dólares anuales.”[1]

En Ecuador, desde el punto de vista económico, el sector primario tiene una vital importancia. Dentro de este sector, resalta mucho la producción de leche.

Un problema común es que muchos de los productores, no poseen los recursos tecnológicos para controlar los procesos de producción o sus sistemas de conservación de leche no son de forma automática y óptima, ya sea por desconocimiento de tecnología o en mayor parte por cuestiones de costos de implementación; esto trae consigo una producción poco eficiente, insegura y dependiente de la presencia de personas en la hacienda para vigilar el correcto estado de los procesos , lo cual se traduce en pérdidas de tiempo y dinero.

En este proyecto se presenta el prototipo de un sistema tecnológico de bajo costo, que permite automatizar, monitorear y controlar el proceso de conservación de la leche en una hacienda, de forma local o remota.

## **1.2. OBJETIVOS**

### **1.2.1 OBJETIVO GENERAL**

Diseñar, implementar y probar un prototipo que permita, automatizar un sistema de conservación de leche, utilizando hardware y software libre de bajo costo y fácil implementación.

### **1.2.2 OBJETIVOS ESPECÍFICOS**

- Utilizar una placa Intel Galileo Gen1 como microcontrolador del sistema y como servidor web, demostrando así el potencial del hardware libre como un recurso de bajo costo para la implementación de soluciones tecnológicas.
- Utilizar sensores los cuales permitan, calcular el volumen de leche dentro de un tanque, y la temperatura a la cual se mantiene para su correcta conservación.
- Controlar automáticamente el encendido y apagado del sistema de enfriamiento conectado al tanque de

conservación, en función de los valores de temperatura medidos, para mantener el rango de refrigeración permitido en la leche.

- Configurar un servicio web seguro con autenticación de usuario en la placa intel galileo, para acceder a las variables medidas por los sensores y controlar los actuadores desde un navegador web.
- Controlar la activación del sistema de limpieza del tanque de conservación y el proceso de agitación de la leche contenida en el mismo, vía remota y/o local.

### **1.3. ALCANCE Y LIMITACIONES**

A continuación, detallamos las limitantes y los alcances del proyecto.

### 1.3.1. ALCANCE

Este proyecto permite:

- Autenticación de usuario para el acceso al sistema a través de un navegador web, mediante una conexión segura utilizando certificado SSL.
- Monitorear la temperatura y el volúmen de la leche dentro del tanque de conservación, vía remota y/o local.
- Activar o automatizar el sistema de limpieza del tanque de conservación de leche mediante electroválvulas, vía remota o local.
- Activar o automatizar el sistema de agitación de la leche mediante un motor, vía remota o local.
- Demostrar el correcto funcionamiento del sistema mediante la creación de una maqueta, la cual permita mostrar la eficiencia del sistema.

### 1.3.2. LIMITACIONES

Las limitaciones del proyecto son:

- Los elementos actuadores para activar el sistema de limpieza del tanque de conservación, el agitación de la leche y el sistema de enfriamiento, no son reales. Estos son simulados mediante motores, electroválvula y bomba.
- Los sensores utilizados para recolección de las señales físicas no son los adecuados para trabajar en ambientes agresivos.
- La tarjeta intel galileo gen 1 no posee la protección adecuada ante el ambiente físico en donde se encuentra empotrado.
- El volumen de leche mostrado tiene un margen de error, al ser una medición que se realiza de forma indirecta.

#### **1.4. RESULTADOS ESPERADOS**

Los resultados son:

- Obtener una interfaz web con un diseño simple y de fácil uso para el usuario, además adaptable a diferentes tamaños de pantalla.
- Establecer una conexión segura al acceder de forma remota al sistema utilizando cifrado de datos.
- Monitorear la temperatura y el volumen de leche en el tanque de conservación sin problemas de comunicación.
- Habilitar los actuadores vía remota y local de manera segura.
- Demostrar el sistema mediante un prototipo para observar su correcto funcionamiento.

## 1.5. APLICACIONES

La aplicación de este proyecto está dirigida a los productores de leche en el sector primario, actualmente la mayoría de haciendas no poseen o cuentan con un bajo nivel tecnológico para la automatización y el control de los procesos de conservación. Dicho problema se da por no contar con los recursos económicos para la implementación de grandes procesos industriales o por la falta de conocimientos y capacitación.

Según el último censo agropecuario realizado en el Ecuador, son aproximadamente 298 mil productores, de los cuales, el 73% está en la sierra; el 19%, en la costa, y el 8%, en la amazonía. Aproximadamente 1,5 millones de empleos directos e indirectos genera la cadena productiva de esta actividad. [2]

Con este prototipo se ofrece una solución tecnológica, eficiente y de bajo costo que benefice a los productores de leche los cuales requieren implementar tecnología para seguir mejorando la calidad de su producto. La idea del telecontrol de procesos puede además ser ampliada para el control de otros sistemas industriales e incluso para aplicaciones domóticas, el principio de funcionamiento es el mismo.

## **CAPÍTULO 2**

### **2. MARCO TEÓRICO**

Para una mejor comprensión sobre el proyecto deben de repasarse

algunos conceptos básicos, es por esto que se ha dividido el capítulo en las siguientes secciones:

- Fundamentación teórica.
- Especificación hardware de los componentes.
- Protocolos y conceptos de comunicación usados.
- Especificación del software utilizado.

## **2.1. FUNDAMENTACIÓN TEÓRICA**

En esta sección se detallan los conceptos básicos, características y beneficios de las tecnologías utilizada en este proyecto.

### **2.1.1. AUTOMATIZACIÓN**

La automatización se refiere al conjunto de tareas realizadas con elementos tecnológicos, las cuales se ejecutan con la mínima intervención del ser humano o incluso prescindiendo totalmente.

La automatización y telemetría, se utilizan para controlar en forma remota y local, parámetros de comportamiento de diferentes procesos de fabricación, producción, conservación etc.[3]

### **2.1.2. SISTEMAS EMBEBIDOS**

Son sistemas de computación diseñados para realizar funciones dedicadas y cubrir necesidades específicas, tiene módulos estrictamente necesarios para su función, a

diferencia de un computador, la mayoría de los componentes se encuentran incluidos en la en la misma placa base. Figura 2.1.

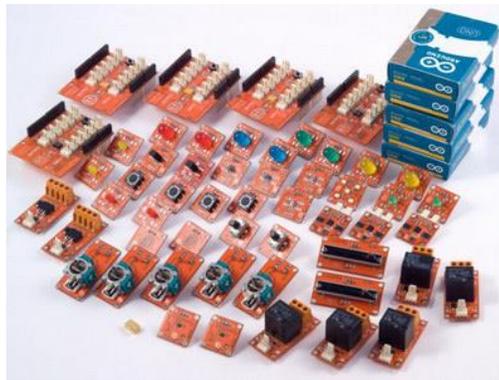


**Figura 2.1 – Sistema embebido [1]**

Existen muchos lugares donde se pueden encontrar los sistemas embebidos, la mayoría en dispositivos que realizan tareas programadas, como en una fábrica controlando el proceso de producción, existen máquinas que integran interfaz gráfica las cuales al pulsar realizan determinada orden, los sistemas embebidos se pueden encontrar en un sin fín de aplicaciones[4].

### 2.1.3. SENSORES

Un sensor o captador, es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, la cual puede cuantificarse y procesarse [5]. Figura 2.2.



**Figura 2.2 – Sensores [2]**

### 2.1.4. ACTUADORES

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica, en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control como, por ejemplo, una válvula[6]. Figura 2.3



**Figura 2.3 – Actuadores [3]**

### **2.1.5. SERVIDOR WEB**

Un servidor web o servidor HTTP, es un programa informático que brinda servicios a otros dispositivos que se comunican con el por medio de una red.

Un servidor web es un equipo donde físicamente se encuentran alojadas aplicaciones o páginas web, los clientes realizan peticiones al servidor y tienen como respuesta un código el cual suele ser compilado y ejecutado por un navegador web.

Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP

para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI [7].

#### **2.1.6. CAPA DE CONEXIÓN SEGURA SSL**

SSL es un protocolo criptográfico, permite que las aplicaciones que utilizan SSL transporten su información de forma segura , estas aplicaciones dan y reciben claves de cifrado con otras aplicaciones para poder cifrar y descifrar los datos enviados entre ellos.

Inicialmente el servidor y el cliente negocian el algoritmo criptografico a utilizar para cifrar la información, luego intercambian claves públicas y realizan autenticación basada en certificados digitales.

#### **2.1.7. CERTIFICADO CSR**

El certificado CSR es un bloque de texto cifrado generado usualmente en el servidor, contiene información como nombre o nombre de empresa, dirección, país de residencia

o el common name (dominio para el que es generado el SSL), además de estos datos también incluirá un clave pública, toda esta información será incluida en el certificado SSL.

## **2.2. ESPECIFICACIÓN HARDWARE DE LOS COMPONENTES.**

En esta sección se detallan las características básicas de cada uno de los componentes que se utilizan para implementar el proyecto.

### **2.2.1. TARJETA INTEL GALILEO GEN 1**

Es una tarjeta microcontroladora basada en el procesador Intel Quark SoC X1000, con un sistema Pentium Intel de 32 bits, es la primera placa basada en la arquitectura Intel, diseñada para ser compatible con el hardware y software de las placas arduino.

La tarjeta Intel Galileo opera con 3.3V y esta diseñada para que la tensión de sus pines E/S puedan ser de 3.3V o 5V

según se requiera a través de la configuración de un puente físico presente en la placa.

El entorno de desarrollo para esta placa es el “Arduino-IDE para Intel Galileo”, además incorpora varios puertos E/S estándar, tales como: un puerto Ethernet de 10/100 Mbps, una ranura Micro-SD, un puerto serial RS-232, un puerto Host USB, un puerto USB cliente[8].

El prototipo emplea esta placa, ya que utiliza software y hardware libre de bajo costo, además es compatible con la mayoría de las librerías empleadas por arduino, es de fácil de uso y permite conectarse a la red a través del puerto ethernet. Figura 2.4

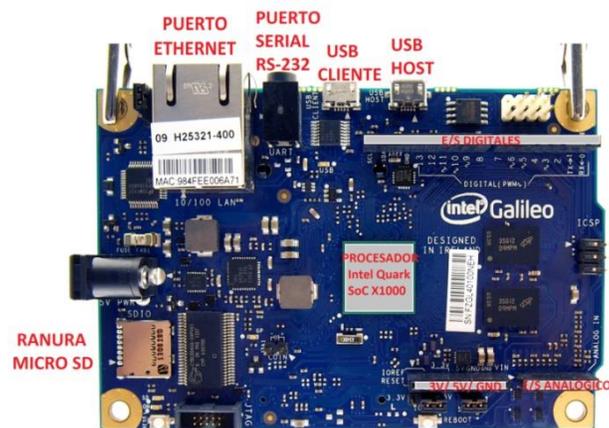


Figura 2.4 – Tarjeta Intel Galileo [4]

### 2.2.1.1. CONECTOR ETHERNET 10/100

La placa Intel Galileo posee un conector RJ45 el cual permitir conectarse a la red cableada.

Para conectarse a una red, se debe proporcionar una dirección IP y una dirección MAC[8]. Figura 2.5



Figura 2.5 – Conector RJ-45 [5]

### 2.2.1.2. CONECTORES GPIO

La placa Intel Galileo tiene 14 E/S digitales , de las cuales 6 se pueden utilizar como salidas PWM(modulación por ancho de pulso); Cada uno de los 14 pines digitales de Galileo se puede utilizar como una entrada o salida, el uso (funciones) `pinMode ()`, `digitalWrite ()`, y `digitalRead`.

Operan en 3,3 o 5 voltios. Cada pin puede proporcionar un máximo de 10 mA o recibir un máximo de 25 mA y tiene una resistencia de pull-up (desconectado por defecto) de 5,6 K a 10 kOhm.

Ademas tiene 6 entradas analógicas, a través de un AD7298 convertidor analógico-digital con una resolución de 12 bits (es decir, 4.096 valores diferentes) [8]. Figura 2.6



**Figura 2.6 – E/S Digitales y Analógicas [4]**

### **2.2.2. MÓDULO LCD 16x2**

El módulo LCD, es una pantalla de cristal líquido, que se utiliza como dispositivo de visualización gráfica, para mostrar caracteres y símbolos[9]. Figura 2.7



**Figura 2.7 – Módulo LCD 16X2 [6]**

### **2.2.3. SENSOR ULTRASÓNICO HC-SR04**

Un sensor ultrasónico emite un impulso acústico de alta frecuencia y corta duración el cual es imperceptible para el oído humano, el sensor cronometra el tiempo que el sonido tarda en llegar a un obstáculo, rebotar y volver al sensor.

Como la velocidad de propagación del sonido en el aire es un valor conocido (343.2 m/s) se puede utilizar la ecuación ( $e = v * t$ ) y así calcular la distancia recorrida por las ondas de sonido.

El prototipo emplea un sensor ultrasónico HC-SR04, para realizar las mediciones de volúmen, ya que al tratarse de un producto alimenticio no se puede entrar en contacto físico, este sensor es idial en estos casos trabaja enviando impulsos acústicos, además este módulo es compatible con la placa intel galileo y es de bajo costo [10]. Figura 2.8

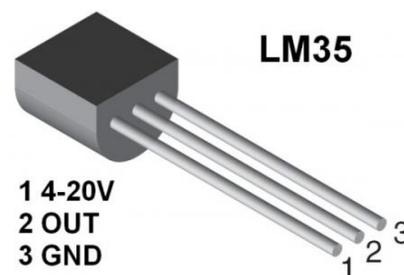


**Figura 2.8 – Sensor Ultrasónico [7]**

#### **2.2.4. SENSOR DE TEMPERATURA LM-35**

El LM35 es un sensor de temperatura, cuya tensión de salida es linealmente proporcional a temperatura en °C (grados centígrados), puede funcionar con alimentación simple o alimentación doble (+ y -).

El prototipo emplea un sensor LM35, ya que esta directamente en grados celcius, no necesita circuitos adicionales para calibrarlo extarnamente, se alimenta con baja corriente, además es de bajo costo[11]. Figura 2.9



**Figura 2.9 – Sensor de temperatura [8]**

### 2.2.5. ENRUTADOR

Un enrutador permite interconectar redes de computadoras, es el encargado de encaminar los paquetes dentro de una red, se dice que un enrutador es la puerta hacia internet.

Para realizar las pruebas del prototipo se utilizó el enrutador cisco E900 el cual funciona con la tecnología 802.11n, opera en la banda de 2,4 GHz, maneja altas velocidades, hasta 300 Mbps para lograr la mejor experiencia de red [12].

Figura 2.10



**Figura 2.10 – Enrutador [9]**

### 2.3. PROTOCOLOS Y CONCEPTOS DE COMUNICACIÓN USADOS.

En esta sección se detallan los conceptos de comunicaciones necesarios que se deben conocer antes de elaborar el proyecto.

### 2.1.1. PROTOCOLO TCP/IP

El protocolo de comunicación que utiliza la tarjeta intel galileo gen 1 para la transmisión de datos es el protocolo TCP/IP, el cual es una arquitectura de capas que determina los protocolos que se debe implementar para efectuar una comunicación entre sistemas. Actualmente es la arquitectura estándar más adoptada para la interconexión de equipos.

El modelo TCP/IP es la versión resumida del modelo OSI y se divide en cuatro capas, cada una con su conjunto de protocolos[13]. Figura 2.11



Figura 2.11 – Modelo TCP/IP [10]

### **2.1.2. PROTOCOLO HTTP**

Es un protocolo de transferencia de hipertexto, se usa en la web cuando escribimos una dirección en un navegador, este establece una conexión con el servidor web a través del protocolo HTTP, opera por petición y respuesta entre el cliente y el servidor.

Toda la información que opera en la Web mediante este protocolo es identificada mediante el URL o dirección[14].

## **2.4 ESPECIFICACION DEL SOFTWARE UTILIZADO.**

En esta sección se detallan las aplicaciones que se requirieron para la implementación y pruebas del proyecto.

### **2.4.1 IDE ARDUINO PARA INTEL GALILEO**

El entorno de desarrollo integrado “Arduino-IDE for Intel Galileo”, es de código abierto, lo cual hace que sea de fácil escritura. Este permite cargar a la placa el programa que se haya creado, por medio del puerto USB.

El entorno está escrito en Java y basado en Processing el cual es un lenguaje de programación y entorno de desarrollo integrado de código abierto, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Se distribuye bajo la licencia GNU GPL. Figura 2.12

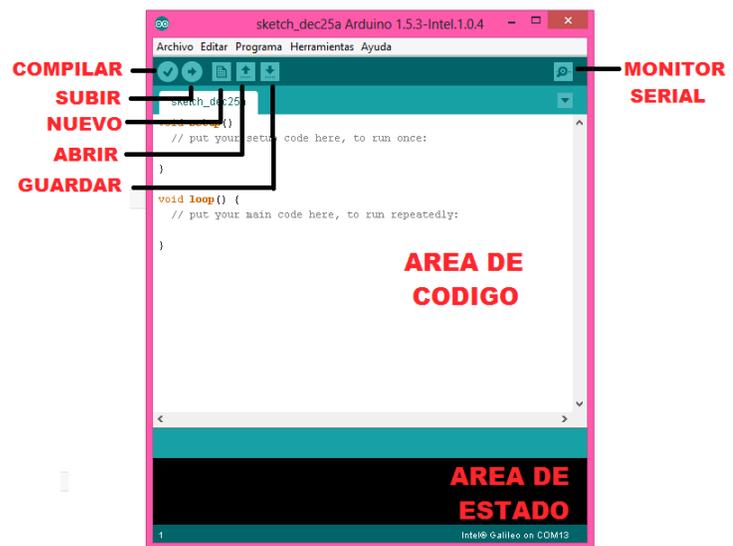


Figura 2.12 – Pantalla IDE Arduino

#### 2.4.2 SUBLIME TEXT

Es un editor de texto y editor de código fuente, muy ligero y minimalista, el código se presenta en pestañas, con coloreado de sintaxis para la mayoría de lenguajes.

En la elaboración del prototipo se utiliza este editor de texto, para crear los archivos html de la página web, además del código javascript necesario para las funciones propias de la página. Figura 2.13



Figura 2.13 – Sublime text

### 2.4.3 PHP

Es un lenguaje de código abierto muy utilizado especialmente para el desarrollo de páginas web, su código es ejecutado del lado del servidor generando HTML y enviándolo al cliente.

La página web de este prototipo se desarrollo incrustando código HTML en páginas PHP, además se desarrollo scripts PHP para la recolección de datos.

#### **2.4.4 APACHE**

Es un servidor web HTTP de código abierto, es muy versatil ya que puede ser instalado en diferentes plataformas y configurado dependiendo de las necesidades propias del sistema.

El prototipo utiliza como servidor web Apache para el envío de las páginas web, trabajando en conjunto con PHP y MySql.

#### **2.4.5 MYSQL SERVER**

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario , es muy utilizado en el desarrollo de aplicaciones web y puede ser instalado en cualquier plataforma.

Para el desarrollo del prototipo se utiliza MySql como gestor de base de datos para realizar la autenticación de usuarios y la generación de reportes.

#### **2.4.6 OPEN SSL**

Es un proyecto de software libre el cual consiste en un paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, estas herramientas ayudan al sistema a implementar el Secure Sockets Layer (SSL). Para el desarrollo del prototipo se utiliza OpenSSL, ya que permite crear certificados digitales que fueron aplicados al servidor apache instalado.

## **CAPÍTULO 3**

### **3. DISEÑO E IMPLEMENTACIÓN**

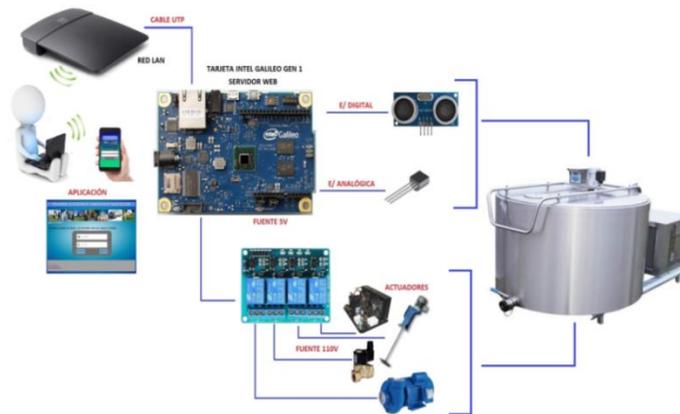
Para realizar un seguimiento detallado del diseño, implementación y ensamblaje del prototipo, el capítulo se dividió en 5 secciones:

#### **3.1. DISEÑO GENERAL**

En esta sección se muestra el esquemático, lista de recursos utilizados, diagrama de bloque, diagrama funcional y una descripción del sistema.

### 3.1.1. ESQUEMÁTICO

Se muestra un diagrama esquemático de la estructura del sistema, donde se puede observar como se interconectan los dispositivos para su correcto funcionamiento.



**Figura 3.1 – Esquemático de la estructura del sistema**

### 3.1.2. LISTA DE RECURSOS UTILIZADOS

Los materiales que se utilizaron son:

- 1 computadora.
- 1 enrutador.
- 1 tarjeta Intel Galileo Gen1.
- 1 sensor ultrasónico HC-SR04.

- Tarjeta micro SD mínimo 4GB
- 1 sensor de temperatura LM35.
- 1 cable directo para conexión ETHERNET.
- 3 Reles de 5V a 10<sup>a</sup>
- Actuadores
  - Electroválvula 110V (evacuar agua)
  - Motor 110V (proceso de agitación)
  - Ventilador (proceso de enfriamiento)
- 1 pantalla LCD 16x2

Los recursos usados de la tarjeta Intel Galileo son:

- Puerto ethernet 10/100.
- 1 pin analógico.
- 5 pines analógicos convertidos digitales.
- 13 pines digitales.
- Pines de alimentación de 5V.

- Ranura para tarjeta micro sd.

El paquete de software utilizados para elaborar el prototipo son:

- Arduino para Intel Galileo.
- Sublime text.

### 3.1.3. DIAGRAMA DE BLOQUES

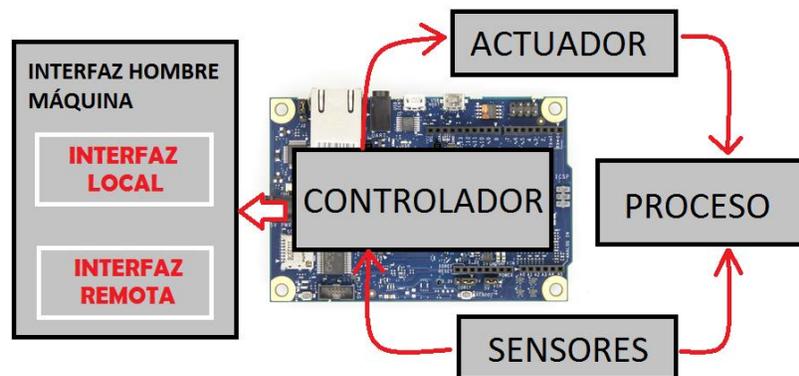


Figura 3.2 – Diagrama de bloques del sistema.

El sistema esta formado por:

- **Pulsadores:** permiten interactuar con el usuario, ingresando ordenes para activar un determinado proceso.
- **Leds:** permiten indicar al usuario, los procesos que se encuentran activos.
- **LCD:** permite mostrar al usuario de forma textual la temperatura y el volumen de leche contenida en el tanque, además presenta mensajes de errores y avisos cuando es activado determinado proceso.
- **Sensores:** los cuales recogen información de magnitudes físicas externas y las transforman en señales eléctricas para ser enviadas al controlador y así poder procesarlas.
- **Actuadores:** los cuales son activados a partir de las ordenes dadas por el controlador.

### 3.1.4. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL SISTEMA

A continuación se realiza una descripción del funcionamiento del sistema de telecontrol , para explicar de forma más detallada el funcionamiento del mismo.

- **Interfaz web:** el sistema incorpora un servidor web alojado en la tarjeta controladora Intel Galileo Gen 1, por medio de la cual se puede realizar el telecontrol del sistema enviando peticiones http desde un navegador web.
- **Visualización de Volúmen y temperatura:** es posible observar a través de la interfaz web y/o la pantalla LCD, los valores de la temperatura y el volúmen del fluido contenido en el tanque.
- **Proceso de agitación:** este proceso mejora la consistencia de la leche, aumenta su blancura, ayuda a la transferencia del calor, evita la formación de grasa; el agitación mejora la calidad del producto final. En la industria alimenticia se recomienda el uso de tanques con fondo redondeado, con el fin de eliminar los bordes

rectos o regiones en las cuales no penetrarían las corrientes del fluido.

En el prototipo presentado el proceso de agitación puede ser de modo automático o manual.

Modo automático: el sistema está programado para que el tiempo de reposo previo al agitación sea de cinco minutos, así mismo el motor agita la leche durante cinco minutos, este proceso se activará siempre y cuando el sensor ultrasónico detecte que existe leche en el tanque.

Modo manual: Desde la interfaz web el sistema permite cambiar el tiempo predeterminado de reposo previo al agitación, por el tiempo que se considere necesario dependiendo de parámetros como volumen de leche en el tanque, el motor agitará la leche durante cinco minutos.

- **Proceso de enfriamiento:** una vez extraída la leche de las glándulas mamarias de la vaca, se deposita en el tanque de refrigeración, es importante mantener la leche a una temperatura inferior a 8 °C si va a ser recogida en el día o a 5 °C si va a ser recogida al día siguiente, con

el objetivo de evitar la multiplicación de microorganismos mesófilos acidificantes (bacterias lácticas). En el prototipo presentado, el sistema de enfriamiento es de forma automática, es decir el tanque encendera o apagará el compresor para mantener la leche a una temperatura entre 3 °C a 5 °C, este proceso se activara siempre y cuando el sensor ultrasónico detecte que exista leche en el tanque.

- **Proceso de lavado:** El tanque refrigerante puede ser también una fuente de contaminación, para tratar de minimizarla hay que utilizar equipos con un diseño correcto que se puedan limpiar y desinfectar eficazmente, es importante que el tanque de conservación se limpie antes de volver a utilizarlo. En el prototipo presentado el sistema de limpieza puede ser automático o manual.

Modo automático: El sistema es capaz de autolimpiarse, una vez que el sensor ultrasónico no detecte nivel de leche, se encendera la bomba para extraer agua y lavarse.

Modo manual: de forma local, presionando el botón ubicado en el tablero eléctrico, siempre y cuando no exista leche en el tanque, caso contrario se mostrara el mensaje “Aun hay leche”; de forma remota, por medio de la interfaz web, dando clic en el botón lavar tanque, siempre y cuando no exista leche en el tanque caso contrario se mostrara el mensaje “Aun hay leche”.

- **Paro de Emergencia:** En todo sistema de automatización debe existir una forma de detener instantáneamente la marcha de la máquina por razones de seguridad; para ello se implementó en el tablero eléctrico un pulsador que permite realizar esta acción y otro que pone nuevamente en marcha a la máquina; este último pulsador no activa el funcionamiento de forma instantánea, sino que emite una alarma que indica que en unos segundos operará de manera normal para mayor seguridad de quien esté manipulando la máquina en ese momento.

### **3.1.5. DIAGRAMA FUNCIONAL**

Para mostrar la forma de operación del sistema de telecontrol, se estructuró un diagrama funcional dividido en cinco partes:

En la figura 3.3 se muestra el diagrama funcional de la interacción entre la página web y el servidor web alojado en la tarjeta intel galileo gen 1.

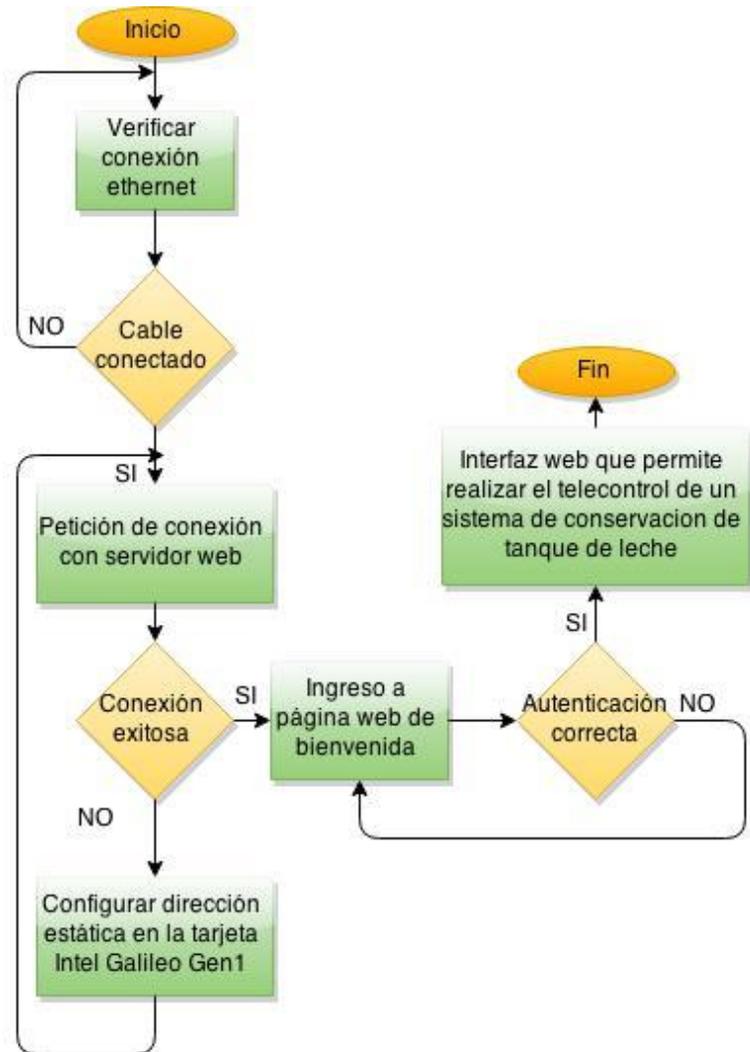
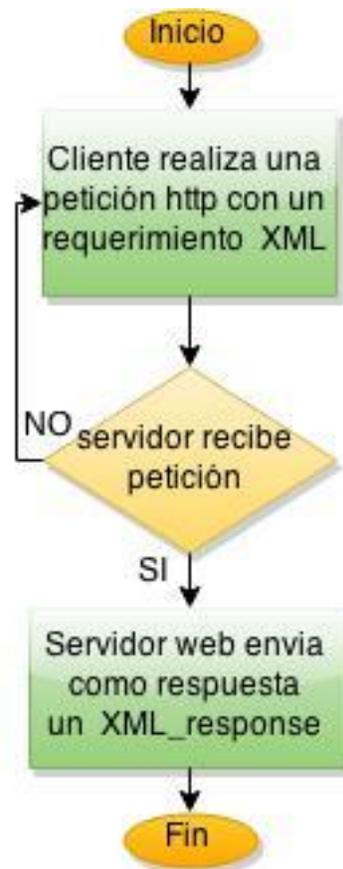


Figura 3.3 – Diagrama de flujo de ingreso a la interfaz web.

En la figura 3.4 se muestra el diagrama funcional de la transmisión bidireccional de valores de entrada/salida en la comunicación cliente-servidor.



**Figura 3.4 – Diagrama de flujo de la transmisión ente cliente-servidor.**

En la figura 3.5 muestra el diagrama funcional del proceso de agitación.

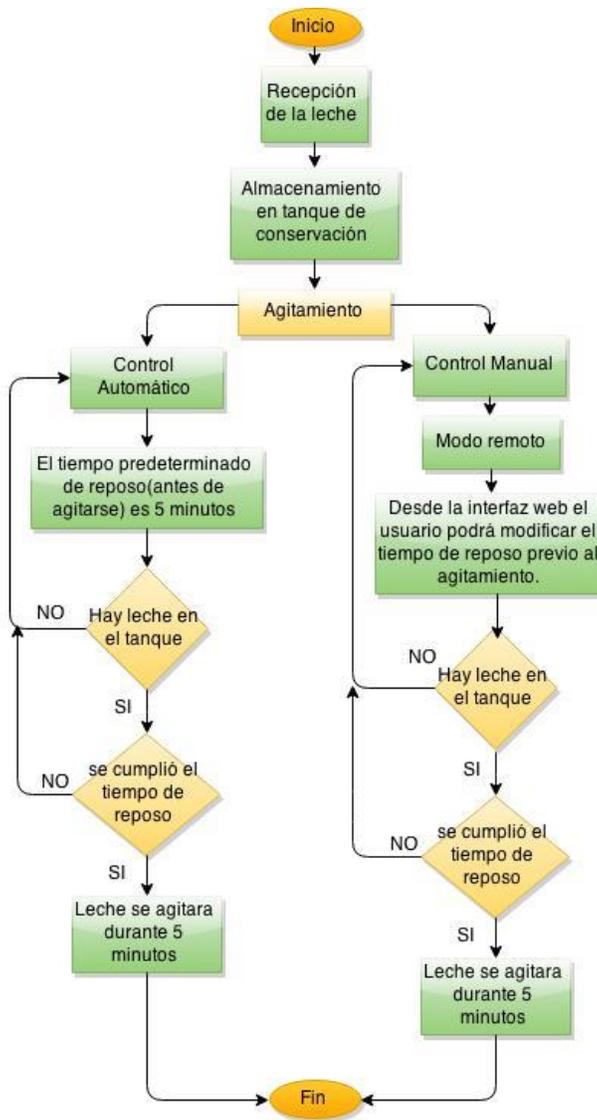


Figura 3.5 – Diagrama de flujo del proceso de agitación.

En la figura 3.6 se muestra el diagrama funcional del proceso de enfriamiento de la leche.



**Figura 3.6 – Diagrama de flujo del proceso de enfriamiento.**

En la figura 3.7 se muestra el diagrama funcional del proceso de lavado el tanque.

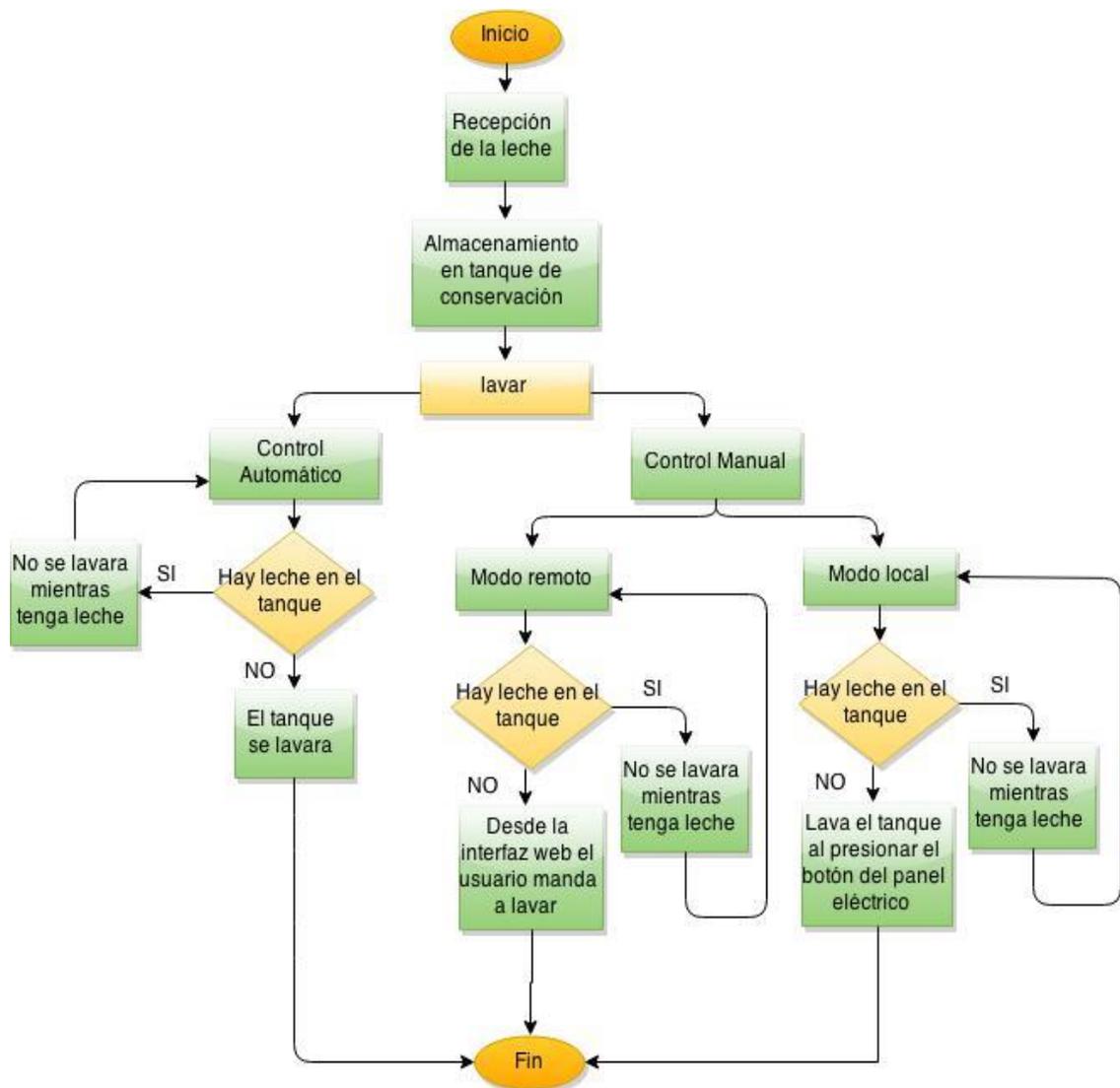


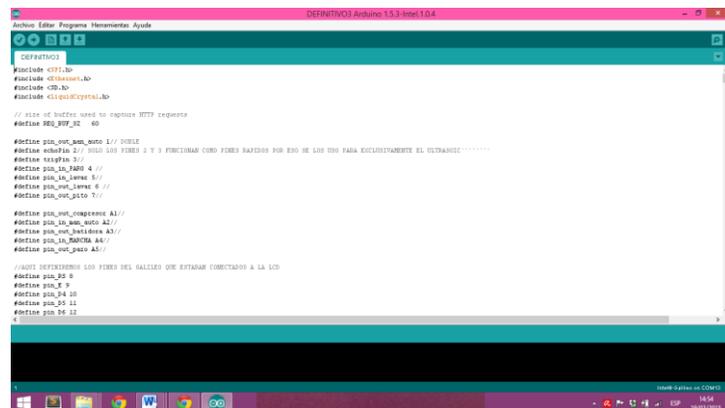
Figura 3.7 – Diagrama de flujo del proceso de lavado.

## 3.2. IMPLEMENTACIÓN DEL HARDWARE EN LA TARJETA INTEL GALILEO GEN 1

### 3.2.1. CREACIÓN DEL PROYECTO EN EL IDE ARDUINO

El software empleado para el desarrollo del proyecto, es el IDE de arduino versión propia para el intel galileo gen 1, la cual puede ser descargada desde la página oficial de arduino.

Para crear un nuevo proyecto basta con iniciar el software, a continuación se muestra la interfaz del IDE.



```
Archivo  Editor  Programa  Herramientas  Ayuda
DEFINITIVO2_Arduino_1.5.2-Intel_1.0.4

DEFINITIONS
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#include <LiquidCrystal.h>

// Para el buzzer used no capture MP3 requests
#include Buzzer.h

#include pin_mcp_9600_1 // D0012
#include echo.h // 0001 L00 P00S 2 Y 3 FUNCIONAN COMO PINES PARIDOS POR ESO SE LOS USO PARA EXCLUSIVAMENTE EL ULTRASONIC*****
#include trigto_3 //
#include pin_in_P000 4 //
#include pin_in_0000 5 //
#include pin_mcp_0000 6 //
#include pin_mcp_0100 7 //

#include pin_mcp_000000 A1 //
#include pin_in_000_0000 A2 //
#include pin_mcp_000000 A3 //
#include pin_in_000000 A4 //
#include pin_mcp_0000 A5 //

//MUCHO DEFINIRON LOS PINES DEL GALILEO QUE ESTARIAN CONECTADOS A LA L00
#include pin_00 8
#include pin_0 9
#include pin_04 10
#include pin_00 11
#include pin_04 12
4
```

Figura 3.8 – IDE Arduino.

Las librerías usadas para el desarrollo del proyecto son:

**SD:** La librería SD permite la lectura y escritura en tarjetas SD, el directorio de trabajo es siempre la raíz de la tarjeta SD[15].

**LiquidCrystal:** Esta librería permite que una placa Arduino pueda comunicarse y controlar una pantalla LCD. La biblioteca trabaja con ya sea en modo de 4 u 8 bits (es decir, por medio de 4 u 8 líneas de datos, además de los rs, habilitar y, opcionalmente, las líneas de control rw) [15].

### 3.2.2. ASIGNACIÓN DE PINES EN LA TARJETA

La tarjeta Intel Galileo cuenta con catorce pines digitales y seis pines analógicos.

La asignación física de los pines digitales es la siguiente.

*Tabla 3.1 – Asignación de pines digitales*

<b>Pin 0</b>	<b>Sin utilizar</b>
<b>Pin 1</b>	Led modo manual o automático
<b>Pin 2</b>	Pin echo de sensor ultrasónico
<b>Pin 3</b>	Pin trigger de sensor ultrasónico

<b>Pin 4</b>	Botón de paro
<b>Pin 5</b>	Botón de lavar
<b>Pin 6</b>	Led lavar tanque
<b>Pin 7</b>	Sirena
<b>Pin 8</b>	LCD
<b>Pin 9</b>	LCD
<b>Pin 10</b>	LCD
<b>Pin 11</b>	LCD
<b>Pin 12</b>	LCD
<b>Pin 13</b>	LCD

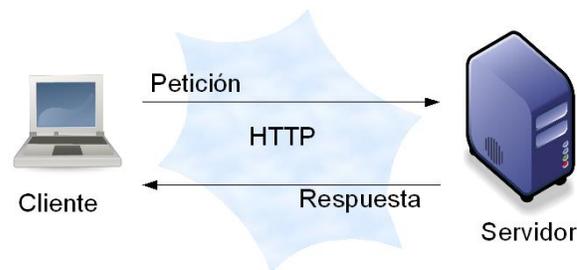
Desde el pin A1 analógico al pin A5 analógico, fueron programados para ser utilizados como pines digitales ante la necesidad de pines para conectar los componentes, únicamente el pin A0 fue utilizado como pin analógico donde se encuentra el sensor de temperatura, la asignación física de los pines analógicos se detalla a continuación.

*Tabla 3.2 - Asignación de pines analógicos.*

<b>Pin A0</b>	<b>Sensor temperatura</b>
<b>Pin A1</b>	Compresor de enfriamiento.
<b>Pin A2</b>	Botón modo manual o automático
<b>Pin A3</b>	Led de Agitamiento
<b>Pin A4</b>	Botón de marcha
<b>Pin A5</b>	Led de paro

### 3.3. CONFIGURACIÓN DEL SERVIDOR WEB EN LA TARJETA INTEL GALILEO GEN 1

La tarjeta Intel Galileo Gen 1 posee un sistema operativo basado en Linux, gracias a esto es posible la instalación de un servidor web, para el envío y recepción de peticiones http a través de una conexión ethernet que maneja una velocidad de 10/100Mb.



**Figura 3.9 – Requerimiento HTTP[11].**

Para que el controlador detecte las peticiones HTTP entrantes, se utilizó el servidor web apache por su versatilidad.

Para el procesamiento de las peticiones se utilizan scripts PHP, los cuales interactúan con el sistema operativo del servidor(Linux), realizando la activación de los actuadores del sistema y la interacción con las variables procesadas por el programa, el cual

esta escrito en el entorno de desarrollo Arduino IDE, cargado en la placa.

Para lograr que el servidor web instalado en la placa de desarrollo ofrezca una conexión segura ante las peticiones de un cliente, se utilizó openssl.



**Figura 3.10 – Certificación SSL[12].**

En términos generales, fue necesario seguir los siguientes pasos:

- Instalar openssl.
- Crear una llave privada.
- Crear un CSR (Certificate Signing Request).
- Generar un certificado SSL.
- Configurar el certificado SSL en Apache.

De esta manera, cuando un cliente intente establecer una conexión https, el servidor le enviará una clave pública, la cual permitirá que toda petición del cliente sea cifrada. El servidor posee una clave privada, la cual es la única capaz de descifrar el mensaje enviado por el cliente.

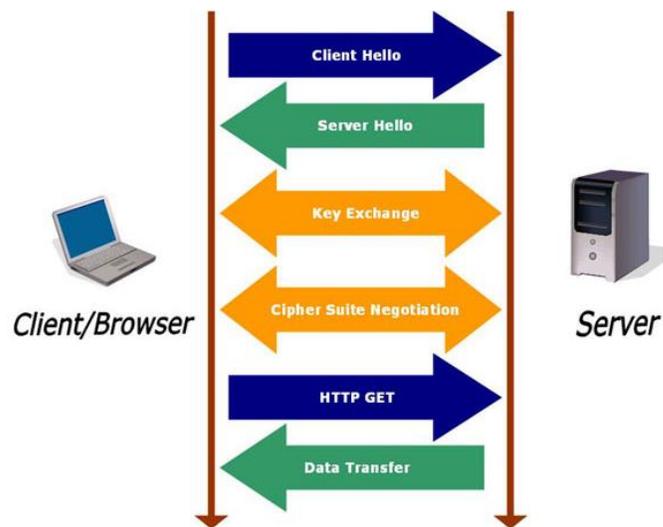


Figura 3.11 – Funcionamiento de SSL[13].

### 3.4. DESARROLLO DE LA APLICACIÓN WEB

El telecontrol de un sistema de conservación de tanque de leche, cuenta con una interfaz web desarrollada con HTML, PHP y CSS; dicha página es de tipo responsive, es decir se adapta a cualquier tipo de dispositivo (teléfonos inteligentes, tabletas, ordenadores

portátil u ordenadores de escritorio), para que los usuarios visualicen la página web con la misma satisfacción que si estuvieran accediendo a través de un ordenador.

La página web muestra una interfaz de bienvenida, donde el usuario deberá autenticarse para acceder a las opciones que ofrece el portal web.

El sistema incorpora dos niveles de acceso, los cuales tienen determinados privilegios dependiendo del tipo de usuario, los cuales son de tipo administrador o de tipo usuario.



**Figura 3.12 – Ingreso a página web.**

Luego del proceso de autenticación , la página web muestra un menú con las etiquetas inicio, telemetría, telecontrol, administrar usuarios, generar reportes, parametrizar y cerrar sesión.

En la ventana de inicio la página muestra una breve descripción del sistema.



**Figura 3.13 – Ventana de inicio.**

La ventana de telemetría permite, observar los valores de temperatura y volúmen al cual se encuentra la leche dentro del tanque.



**Figura 3.14 – Ventana de telemetría.**

En la ventana de telecontrol, el sistema permite reprogramar el tiempo de reposo previo al proceso agitación, durante el cual encenderá el motor para agitar la leche por cinco minutos, también ofrece la posibilidad de lavar el tanque, siempre y cuando no exista leche en el mismo.



Figura 3.15 – Ventana de telecontrol.

En la ventana administrar usuarios, la cual solo está disponible para el usuario tipo administrador, es posible crear, modificar o eliminar usuarios.



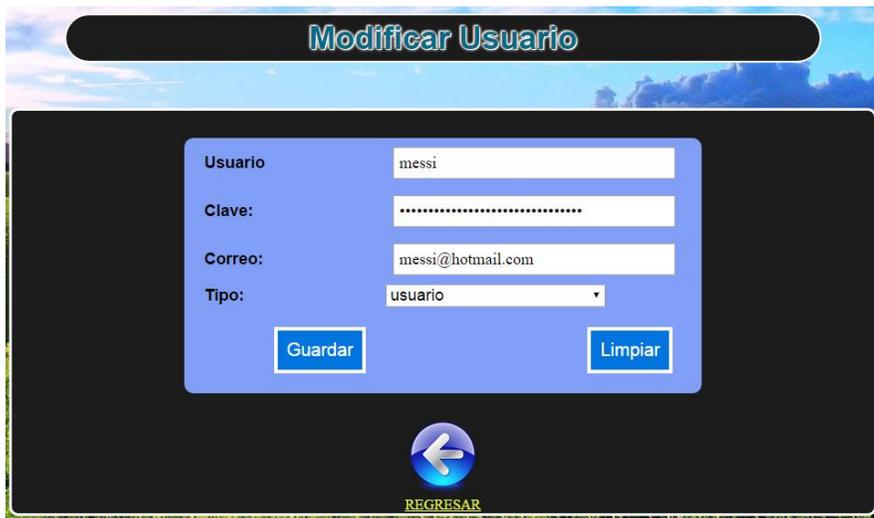
Figura 3.16 – Ventana de administrar usuarios.

Crear usuario, solo el usuario tipo Administrador tiene los permisos para crear nuevas cuentas, deberá ingresar todos los datos requeridos para crear un nuevo usuario, además de ingresar el tipo de usuario el cual se asignará a esta nueva cuenta.



**Figura 3.17 – Ventana creación de usuarios.**

Modificar usuario, solo el usuario tipo Administrador tiene los permisos para modificar las cuentas creadas, actualizar datos, reasignar clave o cambiar el nivel de usuario.



The screenshot shows a web form titled "Modificar Usuario" with a blue header. The form fields are as follows:

Usuario	<input type="text" value="messi"/>
Clave:	<input type="password" value="....."/>
Correo:	<input type="text" value="messi@hotmail.com"/>
Tipo:	<input type="text" value="usuario"/>

Below the fields are two buttons: "Guardar" and "Limpiar". At the bottom center is a blue circular button with a white left-pointing arrow and the text "REGRESAR" below it.

**Figura 3.18 – Modificar usuario.**

Eliminar usuario, solo el usuario tipo Administrador tiene los permisos para eliminar una cuenta de usuario.



**Figura 3.19 – Eliminar usuario.**

En la ventana generar reportes, el usuario tipo Administrador tiene la posibilidad de generar reportes en formato PDF sobre los usuarios que ingresaron al sistema, además de una lista de todos los usuarios que están registrados en la base de datos.



**Figura 3.20 – Ventana generar reportes.**

Ventana parametrizar, solo el usuario tipo administrador tendrá todos los privilegios, es decir por medio de la interfaz web el administrador podrá realizar cambios, los cuales pueden alterar el funcionamiento normal del sistema, es posible realizar cambios muy importantes para el funcionamiento, como ingresar la dirección IP con la cual trabajará la tarjeta intel galileo, especificar los parámetros del alto del tanque de conservación y radio del mismo, especificar el rango de temperatura a la cual debe mantenerse la leche dentro del tanque para su debida

conservación y configurar el sistema de alerta vía correo electrónico la cual permite enviar un correo a la dirección especificada por el administrador cada vez que el tanque este lleno.

The image shows a web-based administrative interface with a dark blue header and four main configuration panels, each with a 'Guardar cambios' button.

- ADMINISTRAR** (Header)
- ASIGNAR NUEVA DIRECCION IP**
  - Ingrese una direccion IP: 192 168 1 46
  - Mascara de red: 255 255 255 0
  - Puerta de enlace: 192 168 1 1
  - Guardar cambios
- REDIMENSIONAR PARAMETROS DEL TANQUE CILINDRICO**
  - Diametro: 26.67 (cm)
  - Ingrese Altura: 31.55 (cm)
  - Guardar cambios
- PARAMETROS DE TEMPERATURA**
  - Ingrese el rango temperatura:
  - Rango min: 25.00 (°C)
  - Rango max: 27.00 (°C)
  - Guardar cambios
- SISTEMA DE ALERTAS**
  - Ingrese correo electronico al cual se enviaran las alertas:
  - Correo: exheman@gmail.com
  - Guardar cambios

Figura 3.21 – Ventana parametrizar.

### 3.5. COSTOS DE IMPLEMENTACIÓN

#### 3.5.1. COSTO DE LA ELABORACIÓN DE LA MAQUETA

Materiales para la elaboración física de la maqueta.

Tabla 3.3 – Costo de elaboración de la maqueta

Descripción	Cantidad	Precio unitario (USD)	Precio Total (USD)
Recipiente contenedor de leche	1	\$ 10,00	\$ 10,00
Recipiente contenedor de agua para lavado	1	\$ 8,00	\$ 8,00
Tubo para transporte de aire refrigerante	1	\$ 5,00	\$ 5,00
Bandeja para desalojo de agua	1	\$ 2,00	\$ 2,00
Bancos de soporte para recipientes	2	\$ 3,00	\$ 6,00
Bomba de agua de 25w	1	\$ 25,00	\$ 25,00
Motor de 110 VAC	1	\$ 5,00	\$ 5,00
Ventilador usado como refrigerante	1	\$ 25,00	\$ 25,00
Sensor ultrasónico hc-sr04	1	\$ 10,00	\$ 10,00
Sensor de temperatura LM35	1	\$ 3,00	\$ 3,00
Electroválvula simple de 110VAC	1	\$ 5,00	\$ 5,00
Manguera para desagüe de lavadora	1 m	\$ 10,00 por metro	\$ 10,00
Ducha	1	\$ 15,00	\$ 15,00
<b>Materiales para sistema de control</b>			
Caja de control (plástica)	1	\$ 60,00	\$ 60,00
Cable flexible 14	40 m	\$ 0,35 por metro	\$ 14,00
Cable flexible 22	40 m	\$ 0,35 por metro	\$ 14,00
Cable directo para conexión Ethernet	5 m	\$ 1,00 por metro	\$ 5,00
Intel Galileo Gen1	1	\$ 110,00	\$ 110,00
Pantalla LCD monocromática de 16x2	1	\$ 20,00	\$ 20,00
Relés de 5V	3	\$ 1,00	\$ 3,00
Transistores bc337	3	\$ 0,50	\$ 1,50
Pulsadores	4	\$ 0,30	\$ 1,20
Diodos LED	7	\$ 0,25	\$ 1,75
Buzzer	1	\$ 1,00	\$ 1,00
Riel DIN	1 m	\$ 1,25 por metro	\$ 1,25
Borneras de DIN	50	\$ 0,30	\$ 15,00
Canaleta para tablero eléctrico	1 m	\$ 1,50 por metro	\$ 1,50
<b>Costo total de elaboración de la maqueta:</b>			<b>\$ 378,20</b>

### 3.5.2. COSTO ESTIMADO DE IMPLEMENTACIÓN EN AMBIENTE REAL

*Tabla 3.4 - Costo estimado de implementación en ambiente real.*

Descripción	Cantidad	Precio unitario (USD)	Precio Total (USD)
Costo del sistema electro-mecánico			
Tanque de leche, sistema de agitación, sensor de nivel, sistema de enfriamiento y sistema de lavado	-----	\$ 8000,00	\$ 8000,00
Materiales para sistema de control			
Caja de control (metálica)	1	\$ 50,00	\$ 50,00
Cable flexible 14	40 m	\$ 0,35 por metro	\$ 14,00
Cable flexible 22	40 m	\$ 0,35 por metro	\$ 14,00
Cable directo para conexión Ethernet	5 m	\$ 1,00 por metro	\$ 5,00
Intel Galileo Gen1	1	\$ 110,00	\$ 110,00
Pantalla LCD monocromática de 16x2	1	\$ 20,00	\$ 20,00
Relés de 5V de estado sólido	3	\$ 20,00	\$ 60,00
Transistores bc337	3	\$ 0,50	\$ 1,50
Pulsadores de botón rasante	4	\$ 5,00	\$ 20,00
Luces LED de 110 VAC	7	\$ 5,00	\$ 35,00
Buzzer	1	\$ 1,00	\$ 1,00
Riel DIN	1 m	\$ 1,25 por metro	\$ 1,25
Borneras de DIN	50	\$ 0,30	\$ 15,00
Canaleta para tablero eléctrico	1 m	\$ 1,50 por metro	\$ 1,50
Diseño e Instalación	-----	-----	\$ 500,00
Página web para telecontrol	-----	-----	\$ 1151,75
Costo total del sistema electro-mecánico			\$ 8000,00
Costo total del sistema de control			\$ 2000,00
Costo total del sistema de Implementación			\$ 10000,00

### 3.6. BOSQUEJO GENERAL DE LA UBICACIÓN DEL SISTEMA

A continuación se muestra una distribución espacial recomendada, de la ubicación del sistema de telecontrol.



Figura 3.22 – Ubicación del tanque vista lateral..

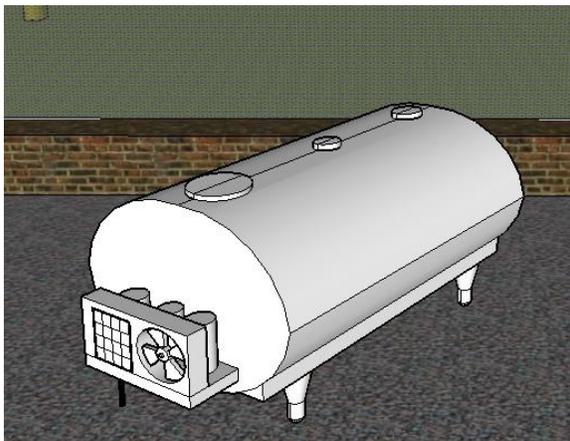


Figura 3.23 – Tanque de conservación.



**Figura 3.24 – Vista superior de la ubicación del tanque.**

## **CAPÍTULO 4**

### **4. PRUEBAS Y RESULTADOS**

A continuación se detallarán los procesos de prueba que se tuvieron que realizar antes y después de la implementación del prototipo para verificar su correcto funcionamiento y validar los resultados correspondientes .

## **4.1. PRUEBAS PREVIAS A LA IMPLEMENTACIÓN DEL PROTOTIPO**

### **4.1.1. CONEXIÓN SSH**

La placa Intel Galileo debe tener correctamente instalado el sistema operativo basado en Linux. Para verificar que la instalación del mismo fue exitosa, se accedió al dispositivo a través del puerto 22 mediante una conexión SSH.

### **4.1.2. SENSOR ULTRASÓNICO HC-SR04**

Para probar el funcionamiento del sensor ultrasónico, se procedió a utilizar dentro del IDE de Arduino, el código del dicho sensor el cual se encuentra en la página oficial de arduino o en cualquier página sobre el mismo.[16]

```

#define PIN_TRIG 12
#define PIN_ECO 13
void setup() {
  // Inicializacion de la comunicacion serial
  Serial.begin (9600);
  // Inicializacion de pines digitales
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECO, INPUT);
}
void loop() {
  long duracion, distancia; // Variables
  /* Hacer el disparo */
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIG, HIGH); // Flanco ascendente
  delayMicroseconds(10); // Duracion del pulso
  digitalWrite(PIN_TRIG, LOW); // Flanco descendente

  /* Recepcion del eco de respuesta */
  duracion = pulseIn(PIN_ECO, HIGH);

  /* Calculo de la distancia efectiva */
  distancia = (duracion/2) / 29;

  /* Imprimir resultados a la terminal serial */
  if (distancia >= 500 || distancia <= 0){
    Serial.println("Fuera de rango");
  }
  else {
    Serial.print(distancia);
    Serial.println(" cm");
  }

  // Retardo para disminuir la frecuencia de las lecturas
  delay(500);
}

```

**Figura 4.1 – Código para sensor ultrasónico.**

Luego de las pruebas respectivas, el programa no funcionó, dado que el IDE de arduino 1.5.3 para intel galileo no tiene implementada la función PulseIn, la cual es pieza clave para el funcionamiento de este sensor.

Este inconveniente fue solucionado creando una función similar, a la cual se denominó PulseIn2, cuyo código fuente proviene de usuarios de la comunidad Intel[17] y de esta manera se logró que el sensor mida distancias de forma correcta.

#### **4.1.3. SENSOR DE TEMPERATURA LM35**

Las pruebas realizadas con este sensor, fueron exitosas al exponerlo a climas calientes y fríos.

Se utilizó código fácil de implementar gracias a la función AnalogRead que provee el IDE Arduino 1.5.3.

## **4.2. SERVIDOR WEB IMPLEMENTADO EN LA PLACA INTEL GALILEO**

Para levantar un servicio web dentro de la placa Intel Galileo, se instalaron los paquetes de Apache, PHP y MySQL server respectivamente a través de una conexión SSH. Para probar el correcto funcionamiento del servicio, se ingresó un navegador web,

enviando una petición a la dirección ip del servidor luego de haber agregado previamente un documento de texto escrito en PHP.

#### **4.2.1. PRUEBAS DE INTERACCIÓN CLIENTE-SERVIDOR**

Al realizar la interacción entre la página web y la placa intel galileo se presentaron las siguientes dificultades:

- Mostrar en el portal web los valores medidos por los sensores.
- Enviar peticiones de activación de salidas a la placa Intel Galileo.

En términos generales, la forma en la que se obtienen los valores de las variables físicas del sistema es la siguiente:

El cliente envía un requerimiento XML constantemente al servidor, en el cual se ejecuta un script PHP que accede a archivos de texto plano en un directorio específico del sistema operativo; estos archivos son creados y actualizados desde el código escrito en arduino que fue cargado a la placa de desarrollo y contienen los valores

numéricos de las variables requeridas tales como el volumen de leche dentro del tanque y su temperatura. Una vez extraída la información de los archivos mencionados, se genera un archivo XML como respuesta al requerimiento del cliente.

En términos generales, la forma en la que se activan salidas de la placa de desarrollo para encender actuadores es la siguiente:

Cuando el cliente envía una petición de activación, se ejecuta un script PHP en el lado del servidor, el cual sobrescribe el archivo donde se encuentra el valor actual de la gpio con 1 o 0 respectivamente de acuerdo a la acción requerida.

### **4.3. PRUEBAS DESPUES DE LA IMPLEMENTACIÓN DEL PROTOTIPO EN MAQUETA**

#### **4.3.1. INESTABILIDAD DEL SENSOR ULTRASÓNICO**

Al fijar el sensor ultrasónico en el tanque y ejecutar el sistema se encontró que las mediciones de distancia que realizaba dicho sensor, variaban a cada instante producto de ruido inmerso en la señal; este inconveniente se solucionó, calculando la media aritmética de cada 10 mediciones, logrando así una mayor estabilidad de la distancia medida por el sensor para cálculos posteriores.

#### **4.4. VERIFICACIÓN DE PROCESOS**

En la siguiente tabla se detalla cada uno de los procesos verificados los cuales garantizan el correcto funcionamiento del sistema.

*Tabla 4.2 – Verificación de procesos.*

PROCESO	ESTADO
Enciende el sistema	
Muestra la temperatura	
Muestra el volumen	
Funciona botón de marcha	
Funciona botón de paro	
Cambio de Manual a automatico	
Cambio de automatico a manual	
Boton de lavar	
Proceso de agitamiento	
Proceso de enfriamiento	
Sistema de alerta (sirena)	
Autenticacion de usuarios	
Creación de nuevos usuarios	
Administración remota	
Telecontrol del sistema	

## **CONCLUSIONES Y RECOMENDACIONES**

### **CONCLUSIONES**

1. Se diseñó un prototipo de un sistema de automatización de un tanque de conservación de leche.
2. Se implementó el prototipo diseñado en una maqueta para demostrar su correcto funcionamiento.
3. Se logró monitorear variables del sistema mediante un navegador web.
4. Se logró habilitar actuadores del sistema mediante peticiones realizadas a través de la web.
5. Se realizó un control de acceso al sistema en la web mediante el registro de usuarios de tipo administrador e invitado.

6. Se realizó una sección de administración en el sitio web que permite la parametrización de variables del sistema.
7. Se logró realizar una conexión segura a través de certificados SSL.
8. Se demostró que es posible implementar soluciones tecnológicas de bajo costo a través del uso de hardware y software libre.

## **RECOMENDACIONES**

1. Utilizar un recubrimiento que proteja la placa controladora.
2. Utilizar sensores con recubrimientos más robustos en caso de querer implementar el sistema en ambientes agresivos.
3. Configurar el enrutador para habilitar la conexión al servidor web desde una red externa.
4. Implementar un sistema generador de energía que sirva como respaldo ante un corte del servicio eléctrico. El sistema deberá enviar una notificación por correo electrónico y/o vía sms cuando este suceso ocurra.
5. Implementar una red de sensores que permitan dar aviso al usuario en caso que un actuador no halla respondido a sus señales de habilitación ya sea por daño o por desperfecto mecánico. Esta notificación puede ser enviada vía sms y/o correo electrónico.

# APÉNDICES

## CÓDIGO FUENTE IDE ARDUINO

```
//#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#include <LiquidCrystal.h>

// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 60

#define pin_out_man_auto 1// DOBLE
#define echoPin 2// SOLO LOS PINES 2 Y 3 FUNCIONAN COMO PINES RAPIDOS
POR ESO SE LOS USO PARA EXCLUSIVAMENTE EL ULTRASOIC*****
#define trigPin 3//
#define pin_in_PARO 4 //
#define pin_in_lavar 5//
#define pin_out_lavar 6 //
#define pin_out_pito 7//

#define pin_out_compresor A1//
#define pin_in_man_auto A2//
#define pin_out_batidora A3//
#define pin_in_MARCHA A4//
#define pin_out_paro A5//

//AQUI DEFINIREMOS LOS PINES DEL GALILEO QUE ESTARAN CONECTADOS A
LA LCD
#define pin_RS 8
#define pin_E 9
#define pin_D4 10
#define pin_D5 11
#define pin_D6 12
#define pin_D7 13

LiquidCrystal lcd(pin_RS, pin_E, pin_D4, pin_D5, pin_D6 , pin_D7);

// MAC address from Ethernet shield sticker under board
byte mac[] = { 0x98, 0x4F, 0xEE, 0x01, 0x57, 0xBB };
EthernetServer server(80); // create a server at port 80
File webFile; // the web page file on the SD card
```

```

char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as null terminated
string
char req_index = 0;           // index into HTTP_req buffer

int tiempo_mensajes_pantalla=3000; //AQUI SE CONFIGURA EL TIEMPO EN ms EN
QUE SE MUESTRA UN MENSAJE (POR LO GENERAL DE ERROR) EN LA
PANTALLA

//VARIABLES PARA EL CÁLCULO DE VOLUMEN
float altura_tanque=29.3; //ALTURA DEL TANQUE EN CM
float distancia=0; //VARIABLE QUE CONTIENE LA DISTANCIA QUE MIDE EL
ULTRASONICO
float diam_tanque=30.2; //RADIO DEL TANQUE EN CM
float nivel_leche=0; //NIVEL DE LECHE EN EL TANQUE EN CM
float nivel_para_lavado=1; //NIVEL DE LECHE A PARTIR DEL CUAL SE PUEDE
MANDAR A LAVAR EL TANQUE;
float volumen_leche=0;
int cont=0;
int tam_array=20;
float valores_dist [20];
float error_vol=0.3;
float vol_temp=0;

//VARIABLES PARA EL LAVADO
int hay_leche=0;
int lavar_auto=1;
int lavando=0;
int recien_lavado=1;
int tiempo_lavado_tanque=10000; //tiempo en ms de lavado de tanque

//VARIABLES PARA BATIDO AUTOMATICO
long previousMillis = 0;
int tiempo_on_batidora = 10000; //BATIDORA PRENDIDA 10S
int tiempo_off_batidora = 60000; //BATIDORA APAGADA INICIALMENTE 10S

//VARIABLES PARA CONTROL DE TEMPERATURA
float lim_inf_temp=25; //TEMP MINIMA DEL TANQUE
float lim_sup_temp=27; //TEMP MAXIMA DEL TANQUE
float temperatura=0; //VARIABLE QUE CONTIENE LA TEMPERATURA QUE MIDE EL
SENSOR DE TEMP LM35
int tempPin = 0; // VARIABLE QUE CONTIENE EL PIN ANALOGICO POR DONDE
INGRESARA LA SEÑAL DE TEMP DEL LM35

//VARIABLE PARA EL STOP Y MARCHA DE LA MÁQUINA
volatile int state = LOW;

//VARIABLES PARA BASE DE DATOS
char Ruta[10] = {'d', 'a', 't', 'a', 's', '.', 't', 'x', 't', '\0'};
int num_users=2;

```

```

String users[12];
int ip_1=192;
int ip_2=168;
int ip_3=1;
int ip_4=41;
String dir_correo_dest="mmillan@espol.edu.ec";
String dir_correo_emisor="cxhernan@gmail.com";
String pass_correo_emisor="NereyCris.1";

IPAddress ip(ip_1,ip_2,ip_3,ip_4); // IP address, may need to change depending on
network

void setup() {

    Serial.begin(9600);
    lcd.begin(16, 2);

    pinMode(trigPin, OUTPUT_FAST);
    pinMode(echoPin, INPUT_FAST);

    pinMode(pin_in_lavar, INPUT);
    pinMode(pin_in_man_auto, INPUT);
    pinMode(pin_in_PARO, INPUT);
    pinMode(pin_in_MARCHA, INPUT);

    pinMode(pin_out_compresor, OUTPUT);
    pinMode(pin_out_batidora, OUTPUT);
    pinMode(pin_out_lavar, OUTPUT);
    pinMode(pin_out_pito, OUTPUT);
    pinMode(pin_out_man_auto,OUTPUT);
    pinMode(pin_out_paro,OUTPUT);

    attachInterrupt(pin_in_PARO, interrup_paro, CHANGE); //DEFINIENDO LA
INTERRUPCION DEL PARO DE MÁQUINA

    // initialize SD card
    //Serial.println("Initializing SD card...");
    if (!SD.begin(4)) {
        Serial.println("ERROR - SD card initialization failed!");
        return; // init failed
    }
    leer_datos();

    //guardar_datos();

    //ALARMA_CORREO(dir_correo_emisor,pass_correo_emisor,dir_correo_dest,"TANQUE
DE LECHE HACIENDA","VACIADO DEL TANQUE DE LECHE","Se informa que se ha
vaciado el tanque de leche de la hacienda LA MIRA.");

```

```
ALARMA_CORREO(dir_correo_emisor,pass_correo_emisor,dir_correo_dest,"CEDILLO_
HERNANDEZ","SENT BY INTEL GALILEO","Ing. buenas madrugadas, si le llego esto es
porque nos funcinó lo del correo electronico :)");
```

```
    //Serial.println("SUCCESS - SD card initialized.");
    // check for index.htm file
    if (!SD.exists("index.htm")) {
        // Serial.println("ERROR - Can't find index.htm file!");
        return; // can't find index file
    }
    //Serial.println("SUCCESS - Found index.htm file.");
```

```
    Ethernet.begin(mac, ip); // initialize Ethernet device
    server.begin(); // start to listen for clients
    //digitalWrite(pin_out_compresor,HIGH);
    //digitalWrite(pin_out_batidora,HIGH);
    digitalWrite(pin_out_man_auto,HIGH);// EL SISTEMA ARRANCA CON LAVADO
AUTOMATICO
```

```
}
```

```
void loop() {
```

```
    //EL CIRCUITO SOLO PRESTA ATENCION AL BOTON DE LAVADO SI EL
AUTOMATICO ESTA DESACTIVADO
```

```
    //digitalWrite(7, state);
    if(state==HIGH)
        parar_maquina();
```

```
    get_volumen_leche();
    get_temperature();
```

```
    //digitalWrite(pin_out_man_auto,LOW);
    //delay(500);
    //digitalWrite(pin_out_man_auto,LOW);
    //delay(500);
```

```
    //enciende leds de aviso de que el lavado esta automatico o manual
```

```
    /*if(lavar_auto==1)
    {
        digitalWrite(pin_out_man_auto,HIGH);
    }
    else
    {
        digitalWrite(pin_out_man_auto,LOW);
    }*/
```

```

// controlar_temp(lim_inf_temp,lim_sup_temp);

//LA SIGUIENTE FUNCION ENCIENDE O APAGA EL COMPRESOR DEPENDIENDO
DE LOS LIMITES DE TEMPERATURA PERMITIDOS EN LA LECHE.

//lcd.clear();
lcd.setCursor(0,0);
lcd.print("VOL: ");
lcd.setCursor(0,1);
lcd.print("TEM:");

lcd.setCursor(5,0);
lcd.print(volumen_leche);
lcd.print(" litros");
lcd.setCursor(4,1);
lcd.print(temperatura);
lcd.print(" celsius");

if(volumen_leche!=0)//ACTUALIZA UNA VARIABLE Q PERMITE VER SI HAY O NO
LECHE EN EL TANQUE
{
    hay_leche=1;
    controlar_temp(lim_inf_temp,lim_sup_temp);
}
else
{
    hay_leche=0;
    digitalWrite(pin_out_compresor,LOW);
}

if(hay_leche==1 && recien_lavado==1)// ENCIENDE EL COMPRESOR LUEGO DE
HABERSE LAVADO EL TANQUE Y LLENARSE DE LECHE (esto depend de la temp de
la leche q se meta)
{
    controlar_temp(lim_inf_temp,lim_sup_temp);
}

if(hay_leche==1)//ACTUALIZO EL INDICADOR DE QUE EL TANQUE FUE LLENO
DESPUES DE HABER SIDO LAVADO
    recien_lavado=0;

    escuchar_boton_fisico_lavar(pin_in_lavar,pin_out_lavar);//ESTA PENDIENTE DEL
BOTON PARA MANDAR A LAVAR
    escuchar_boton_fisico_man_auto();//ESTA PENDIENTE DEL BOTON PARA CAMBIO
DE LAVADO MANUAL O AUTOMATICO

```

```

if(lavar_auto==1)//LAVADO EN AUTOMATICO
{
  if(hay_leche==0)
  {
    if(recien_lavado==0)
      lavar_tanque();
  }
}
else//LAVADO EN MODO MANUAL
{
  if(hay_leche==0)
  {
    if(lavando==1)
    {
      lavar_tanque();
      /*lcd.setCursor(0,1);
      lcd.print("HOLA!!");
      delay(3000);*/
    }
  }
}
}

```

```

//BATIDO AUTOMATICO DE LA LECHE
batir(); //esta funcion verifica internamente si es el momento de batir o no.

```

```

//CODIGO PARA CONEXION ETHERNET

```

```

EthernetClient client = server.available(); // try to get client

```

```

if (client) { // got client?
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) { // client data available to read
      char c = client.read(); // read 1 byte (character) from client
      // limit the size of the stored received HTTP request
      // buffer first part of HTTP request in HTTP_req array (string)
      // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
      if (req_index < (REQ_BUF_SZ - 1)) {
        HTTP_req[req_index] = c; // save HTTP request character
        req_index++;
      }
      // last line of client request is blank and ends with \n
      // respond to client only after last line received
      if (c == '\n' && currentLineIsBlank) {
        // send a standard http response header
        client.println("HTTP/1.1 200 OK");
        // remainder of header follows below, depending on if
        // web page or XML page is requested
        // Ajax request - send XML file

```

```

if (StrContains(HTTP_req, "ajax_inputs")) {
    // send rest of HTTP header
    client.println("Content-Type: text/xml");
    client.println("Connection: keep-alive");
    client.println();
    SetEstados();
    // send XML file containing input states
    XML_response(client);
}
else { // web page request
    // send rest of HTTP header
    client.println("Content-Type: text/html");
    client.println("Connection: keep-alive");
    client.println();
    // send web page
    if (StrContains(HTTP_req, "GET / ") || StrContains(HTTP_req, "GET
/index.htm")) {
        webFile = SD.open("index.htm");    // open web page file
    }
    else if (StrContains(HTTP_req, "GET /menu.htm")) {
        webFile = SD.open("menu.htm");    // open web page file
    }
    else if (StrContains(HTTP_req, "GET /telemetria.htm")) {
        webFile = SD.open("telemetria.htm");    // open web page file
    }
    else if (StrContains(HTTP_req, "GET /telecontrol.htm")) {
        webFile = SD.open("telecontrol.htm");    // open web page file
    }
    /*else if (StrContains(HTTP_req, "GET /banner1.jpg")) {
        webFile = SD.open("banner1.jpg");
    }
    else if (StrContains(HTTP_req, "GET /clave2.png")) {
        webFile = SD.open("clave2.png");
    }
    else if (StrContains(HTTP_req, "GET /homo.jpg")) {
        webFile = SD.open("homo.jpg");
    }
    else if (StrContains(HTTP_req, "GET /lavar.jpg")) {
        webFile = SD.open("lavar.jpg");
    }
    else if (StrContains(HTTP_req, "GET /tanque.jpg")) {
        webFile = SD.open("tanque.jpg");
    }
    else if (StrContains(HTTP_req, "GET /temperatura.jpg")) {
        webFile = SD.open("temperatura.jpg");
    }
    else if (StrContains(HTTP_req, "GET /usuario1.png")) {
        webFile = SD.open("usuario1.png");
    }
}

```

```

else if (StrContains(HTTP_req, "GET /usuario2.png")) {
    webFile = SD.open("usuario2.png");
}
else if (StrContains(HTTP_req, "GET /volumen.jpg")) {
    webFile = SD.open("volumen.jpg");
}*/
if (webFile) {
    while(webFile.available()) {
        client.write(webFile.read()); // send web page to client
    }
    webFile.close();
}
}
// display received HTTP request on serial port
//Serial.print(HTTP_req);
// reset buffer index and all buffer elements to 0
req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
// every line of text received from the client ends with \r\n
if (c == '\n') {
    // last character on line of received text
    // starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // a text character was received from client
    currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the data
client.stop(); // close the connection

} // end if (client)
}

//***** ESTA MARAVILLOSA FUNCION ENVIA UN EMAIL CON
TODOS LOS PARAMETROS QUE SE LE DE*****
void ALARMA_CORREO(String dir_correo_emisor,String pass_correo_emisor,String
dir_correo_dest, String name_from, String asunto,String mensaje)
{
    int i=1;

    //primero borra el archivo anterior con un comando linux
    system("rm /media/realroot/correo.py");

    File dataFile = SD.open("correo.py", FILE_WRITE);

```

```

if (dataFile) {
    dataFile.println("import smtplib");
    dataFile.print("fromaddr = ");
    dataFile.print(dir_correo_emisor);
    dataFile.println("");
    dataFile.print("toaddrs = ");
    dataFile.print(dir_correo_dest);
    dataFile.println("");
    dataFile.print("msg = ");
    dataFile.print("From: ");
    dataFile.print(name_from);
    dataFile.print("\r\n'+");
    dataFile.print("To: ");
    dataFile.print(dir_correo_dest);
    dataFile.print("\r\n'+");
    dataFile.print("Subject: ");
    dataFile.print(asunto);
    dataFile.print("\r\n'+");
    dataFile.print("\r\n'+");
    dataFile.print("");
    dataFile.print(mensaje);
    dataFile.println("");
    dataFile.print("username = ");
    dataFile.print(dir_correo_emisor);
    dataFile.println("");
    dataFile.print("password = ");
    dataFile.print(pass_correo_emisor);
    dataFile.println("");
    dataFile.println("server = smtplib.SMTP('smtp.gmail.com:587')");
    dataFile.println("server.ehlo()");
    dataFile.println("server.starttls()");
    dataFile.println("server.login(username,password)");
    dataFile.println("server.sendmail(fromaddr, toaddrs, msg)");
    dataFile.println("server.quit()");

    // Cerramos el archivo.
    dataFile.close();

    system("python /media/realroot/correo.py");
    // Avisamos de que se ha podido escribir correctamente.
    Serial.println("impresion correcta");

    // Si no pudimos escribir en el fichero avisamos por el puerto serie.
}
else{
    Serial.println("Error al escribir en valorpot.txt");
}
}
}

```

```
//***** ESTA FUNCIÓN GUARDA LOS DATOS EN LA "BASE", ES DECIR  
EN EL ARCHIVO TXT CON VARIABLES DEL SISTEMA*****
```

```
void guardar_datos()
```

```
{
```

```
    int i=1;
```

```
    //primero borra el archivo anterior con un comando linux  
    system("rm /media/realroot/datos.txt");
```

```
    File dataFile = SD.open(Ruta, FILE_WRITE);
```

```
    if (dataFile) {
```

```
        dataFile.println(num_users);
```

```
        for(i=1;i<=(4*num_users);i++)
```

```
        {
```

```
            dataFile.println(users[i-1]);
```

```
        }
```

```
        dataFile.println(diam_tanque);
```

```
        dataFile.println(altura_tanque);
```

```
        dataFile.println(lim_inf_temp);
```

```
        dataFile.println(lim_sup_temp);
```

```
        dataFile.println(ip_1);
```

```
        dataFile.println(ip_2);
```

```
        dataFile.println(ip_3);
```

```
        dataFile.println(ip_4);
```

```
        dataFile.println(dir_correo_dest);
```

```
        dataFile.println("0");
```

```
        // Cerramos el archivo.
```

```
        dataFile.close();
```

```
        // Avisamos de que se ha podido escribir correctamente.
```

```
        Serial.println("impresion correcta");
```

```
        // Si no pudimos escribir en el fichero avisamos por el puerto serie.
```

```
    }
```

```
    else{
```

```
        Serial.println("Error al escribir en valorpot.txt");
```

```
    }
```

```
}
```

```
//*****ESTA FUNCIÓN LEE EL ARCHIVO DE DATOS Y ACTUALIZA EL  
VALOR DE LAS VARIABLES DEL SISTEMA.. ES UNA LLAMADA A LA BASE DE  
DATOS...*****
```

```
void leer_datos()
```

```
{
```

```
    int i=1;
```

```
    num_users=ReadFile(0,Ruta).toInt();
```

```
    for(i=1;i<=(4*num_users);i++)
```

```
    {
```

```

    users[i-1]=ReadFile(i,Ruta);
}
diam_tanque=str_to_float(ReadFile(num_users*4+1,Ruta));
altura_tanque=str_to_float(ReadFile(num_users*4+2,Ruta));
lim_inf_temp=str_to_float(ReadFile(num_users*4+3,Ruta));
lim_sup_temp=str_to_float(ReadFile(num_users*4+4,Ruta));
ip_1=ReadFile(num_users*4+5,Ruta).toInt();
ip_2=ReadFile(num_users*4+6,Ruta).toInt();
ip_3=ReadFile(num_users*4+7,Ruta).toInt();
ip_4=ReadFile(num_users*4+8,Ruta).toInt();
dir_correo_emisor=ReadFile(num_users*4+9,Ruta);

Serial.println(num_users);
for(i=1;i<=(4*num_users);i++)
{
    Serial.println(users[i-1]);
}
Serial.println(diam_tanque);
Serial.println(altura_tanque);
Serial.println(lim_inf_temp);
Serial.println(lim_sup_temp);
Serial.println(ip_1);
Serial.println(ip_2);
Serial.println(ip_3);
Serial.println(ip_4);
Serial.println(dir_correo_dest);
}

//*****ESTA FUNCION RECIBE UN STRING NUMERICO ENTERO O CON
DECIMALES Y BOTA EL MISMO NUMERO EN FORMATO FLOAT
float str_to_float(String numero)
{
    String cad_entero;
    String cad_decimal;
    float entero;
    int indice_punto_dec=0;
    float decimal;
    float numero_flotante;

    //CALCULO LA POSICIÓN DEL PUNTO DECIMAL EN LA CADENA
    indice_punto_dec=numero.indexOf('.');

    //EN CASO DE SI HALLA PUNTO DECIMAL EN LA CADENA, ENTRA AL IF, CASO
    CONTRARIO LA CADENA ES SOLO UN NÚMERO ENTERO
    if(numero.indexOf('.')!=-1)
    {
        cad_entero=numero.substring(0,numero.indexOf('.')); // EXTRAE DE LA CADENA, LA
        PARTE ENTERA DEL NÚMERO Y LA GUARDA EN OTRA CADENA
    }
}

```

```
cad_decimal=numero.substring(numero.indexOf('.')+1,numero.length());//EXTRAE DE LA CADENA, LA PARTE DECIMAL DEL NÚMERO Y LA GUARDA EN OTRA CADENA
```

```
entero=(float)(cad_entero.toInt());//CONVIERTE LA CADENA A UN ENTERO Y LE HACE CAST PARA SER GUARDADO COMO FLOAT
```

```
decimal=(float)(cad_decimal.toInt());//CONVIERTE LA CADENA A UN ENTERO Y LE HACE CAST PARA SER GUARDADO COMO FLOAT
```

```
numero_flotante=entero+decimal/((float)pow(10,cad_decimal.length()));//SE CALCULA EL NÚMERO FLOAT RESULTANTE CON LA PARTE ENTERA Y LA PARTE DECIMAL
```

```
}  
else //EN CASO DE QUE LA CADENA NO TENGA UN . SIGNIFICA QUE ES SOLO ENTERA Y SE HACE EL MISMO PROCESO PERO SOLO CON LA PARTE ENTERA
```

```
{  
cad_entero=numero.substring(0,numero.indexOf('.'));  
entero=(float)(cad_entero.toInt());  
numero_flotante=entero;  
}
```

```
return numero_flotante;
```

```
}
```

```
//***** ESTA FUNCION ARROJA COMO STRING LA LINEA QUE SE REQUIERA DE UN ARCHIVO TXT, CUENTA DE LINEA 0 EN ADELANTE*****
```

```
String ReadFile(int Linea,char Ruta[]){
```

```
int Lin=0;
```

```
String Resultado;
```

```
File myFile;
```

```
byte Bin;
```

```
myFile = SD.open(Ruta);
```

```
if (myFile) {
```

```
while (myFile.available()) {
```

```
Bin=myFile.read();
```

```
if (Bin==13){
```

```
Lin++;
```

```
myFile.read();
```

```
}
```

```
else
```

```
{
```

```
if (Lin==Linea){
```

```
Resultado=Resultado+(char(Bin));
```

```
}
```

```
if (Lin>Linea){
```

```
myFile.close();
```

```
return Resultado;
```

```
}
```

```
}
```

```

    }
    myFile.close();
  }
  else {
    return"NOFILE!!";
  }
}

```

//\*\*\*\*\*ESTA ES LA INTERRUPCION QUE SE ACTIVARA AL PRESIONAR EL PARO DE MÁQUINA, SOLO ACTIVA UNA BANDERA INDICADORA DE QUE ESO OCURRIO\*\*\*\*\*

```

void interrup_paro()
{

```

```

    //digitalWrite(pin_out_compresor,LOW);
    if(state==LOW)
    {
        digitalWrite(pin_out_batidora,LOW);
        digitalWrite(pin_out_lavar, LOW);

        digitalWrite(pin_out_paro,HIGH);
    }

```

```

    state = HIGH;

```

```

    /*lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" MAQUINA EN");
    lcd.setCursor(0,1);
    lcd.print("ESTADO DE: PARO");
    digitalWrite(pin_out_pito,HIGH);
    /*do{

```

```

    }while(digitalRead(pin_in_PARO)==HIGH);
    esta_en_paro=HIGH; //VERIFICA SI SE PRESIONO BOTON DE PARO Y SI ES ASI
    DETIENE LA MAQUINA E INICIA LA MARCHA DESDE EL PRINCIPIO CUANDO SE
    APLASTE MARCHA
    parar_maquina();*/
}

```

//\*\*\*\*\*ESTA FUNCIÓN VERIFICA SI ESTA ACTIVA LA BANDERA DE STOP Y DETIENE LA MAQUINA HASTA QUE SE PRESIONE EL BOTON DE MARCHA\*\*\*\*\*

```

void parar_maquina()
{

```

```

    //if(esta_en_paro = 1)
    //{
    //DESACTIVAMOS LAS SALIDAS (EL SISTEMA SE PARA)
    digitalWrite(pin_out_compresor,LOW);

```

```

digitalWrite(pin_out_batidora,LOW);
digitalWrite(pin_out_lavar, LOW);

digitalWrite(pin_out_paro,HIGH);

digitalWrite(pin_out_pito,HIGH);
delay(200);
digitalWrite(pin_out_pito,LOW);

lcd.clear();
lcd.setCursor(0,0);
lcd.print(" MAQUINA EN");
lcd.setCursor(0,1);
lcd.print("ESTADO DE: PARO");

do{

}while(digitalRead(pin_in_MARCHA)==LOW);
do{

}while(digitalRead(pin_in_MARCHA)==HIGH);
digitalWrite(pin_out_paro,LOW);
int i=0;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("ARRANQUE");
lcd.setCursor(0,1);
lcd.print("EN PROCESO!!");
for(i=1;i<=10;i++)
{
digitalWrite(pin_out_pito,HIGH);
delay(100);
if(i==10)
delay(2000);
digitalWrite(pin_out_pito,LOW);
delay(2000-200*i);
}
state = LOW;
lcd.clear();
//}
}

//*****ESTA FUNCIÓN PERMITE ENCENDER Y APAGAR EL
COMPRESOR PARA MANTENER LA LECHE EN UN RANGO DE TEMP
ASIGNADO*****
void controlar_temp(float lim_inf_temp,float lim_sup_temp)
{
if(temperatura>lim_sup_temp)
{

```

```

    digitalWrite(pin_out_compresor,HIGH);
}
else if(temperatura<lim_inf_temp)
{
    digitalWrite(pin_out_compresor,LOW);
}
}

```

```

//*****ESTA FUNCIÓN PERMITE QUE EL TANQUE
SE LAVE Y ACTIVA O DESACTIVA SALIDAS PERTINENTES A
ELLO*****

```

```

void lavar_tanque()
{
    // DESACTIVAR COMPRESOR
    digitalWrite(pin_out_compresor,LOW);

    //SE CALCULA EL TIEMPO DE LAVADO Y SE ESPERA A QUE LAVE
    lavando=1;
    digitalWrite(pin_out_lavar,HIGH);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("LAVANDO");
    lcd.setCursor(0,1);
    lcd.print("TANQUE!!");
    delay(tiempo_lavado_tanque);

    // EL TANQUE QUEDA EN ESTADO DE RECIEN LAVADO
    recien_lavado=1;
    lavando=0;
    hay_leche=0;
    lcd.clear();
    digitalWrite(pin_out_lavar,LOW);
}

```

```

//***** EL SIGUIENTE MÉTODO PERMITE
REALIZAR LA HOMOGENIZACIÓN DE LA
LECHE*****

```

```

void batir()
{
    unsigned long currentMillis = millis();

    if(currentMillis - previousMillis > (long)tiempo_off_batidora)
    {
        if(hay_leche==1)
        {
            digitalWrite(pin_out_batidora, HIGH);
            lcd.setCursor(0,0);
            lcd.print("EN PROCESO DE:");
            lcd.setCursor(0,1);

```

```

    lcd.print("HOMOGENIZACION");
    delay(tiempo_on_batidora);
    lcd.clear();
    digitalWrite(pin_out_batidora, LOW);
    previousMillis = currentMillis+(long)(tiempo_on_batidora);
  }
}
}

//*****PERMITE INTERACTUARR CON
BOTON FISICO DE LAVAR TANQUE*****
void escuchar_boton_fisico_lavar(boolean pin_in,boolean pin_out)
{
  if(digitalRead(pin_in)==HIGH)
  {
    do{
    }while(digitalRead(pin_in)==HIGH);
    if(lavar_auto==0)
    {
      if(hay_leche==0)
        lavando=1;
      else
      {
        //mensaje que aun hay leche en tanque
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("AUN HAY");
        lcd.setCursor(0,1);
        lcd.print("LECHE");
        digitalWrite(pin_out_pito,HIGH);
        delay(tiempo_mensajes_pantalla);
        lcd.clear();
        digitalWrite(pin_out_pito,LOW);
      }
    }
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("DEBE ACTIVAR EL");
    lcd.setCursor(0,1);
    lcd.print("MODO MANUAL");
    digitalWrite(pin_out_pito,HIGH);
    delay(tiempo_mensajes_pantalla);
    digitalWrite(pin_out_pito,LOW);
    lcd.clear();
  }
}
}
}

```

```

//*****FUNCION QUE ESTA PENDIENTE DE QUE
CAMBIEN MODO MANUAL AUTOMATICO DE LAVADO*****
void escuchar_boton_fisico_man_auto()
{
  if(digitalRead(pin_in_man_auto)==HIGH)
  {
    do{
      }while(digitalRead(pin_in_man_auto)==HIGH);
    if(lavar_auto==1)
    {
      lavar_auto=0;
      digitalWrite(pin_out_man_auto,LOW);
    }
    else
    {
      digitalWrite(pin_out_man_auto,HIGH);
      lavar_auto=1;
    }
  }
}

```

```

//*****FUNCIÓN QUE CALCULA EL NIVEL DE
LECHE EN EL TANQUE RESPECTIVO SEGÚN LO QUE BOTA EL
ULTRASONICO*****
void get_nivel_leche()
{
  float duration=0;

  digitalWrite(trigPin, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = (float)pulseIn2(echoPin, HIGH);
  valores_dist[cont] = ((duration/29)/2)+1;

  //AQUI PROMEDIARE 10 VALORES QUE ME DE EL SENSOR PARA QUE NO
  VARIEN TANTO

  if(cont==tam_array-1)
  {
    distancia=0;
    int i=0,j=0;
    for (i=0;i<tam_array;i++) {
      distancia=distancia+valores_dist[i];
    }
    distancia=distancia/tam_array;
    cont=-1;//pa q en la siguiet vuelta se inicialice en

```

```

    for (j=0;j<tam_array;j++) {
        valores_dist[j]=0;
    }
}
else if(distancia==0)
{
    distancia=valores_dist[0];
}

cont++;
nivel_leche=altura_tanque-distancia;

}

//*****FUNCIÓN QUE CALCULA EL VOLUMEN
DE LECHE EN EL TANQUE RESPECTIVO SEGÚN EL NIVEL DE LECHE PRESENTE
EN EL TANQUE*****
void get_volumen_leche()
{
    get_nivel_leche();
    volumen_leche= 22.698-0.7821*distancia; //ECUACION SACADA EN EXCEL
    DESPUES DE INTERPOLAR VARIOS PUNTOS

    if(volumen_leche<=0.6)
        volumen_leche=0;
    //VALIDACION DE ADMISION DE ERROR EN VOLUMEN

    /*if(vol_temp==0)
        vol_temp=volumen_leche;
    else
    {
        if(abs(volumen_leche-vol_temp)>error_vol)
            volumen_leche=vol_temp;
    }*/

    /*volumen_leche=PI*(radio_tanque)*(radio_tanque)*nivel_leche;// volumen en cm
    cubicos
    volumen_leche=volumen_leche/1000; //volumen en litros. */
}

//Funcion que emula PulseIn, calcula el ancho de un pulso, en este caso se usa para
estimar el tiempo de echo del ultrasonico.
unsigned long pulseIn2(uint8_t pin, uint8_t state) {
    unsigned long pulseWidth = 0;
    unsigned long loopCount = 0;
    unsigned long loopMax = 5000000;
    // While the pin is *not* in the target state we make sure the timeout hasn't been
    reached.
    while ((digitalRead(pin)) != state) {

```

```

        if (loopCount++ == loopMax) {
            return 0;
        }
    }
    // When the pin *is* in the target state we bump the counter while still keeping track of
    the timeout.
    while ((digitalRead(pin)) == state) {
        if (loopCount++ == loopMax) {
            return 0;
        }
        pulseWidth++;
    }
    // Return the pulse time in microsecond!
    return pulseWidth * 1.18; // Calculated the pulseWidth //this is the value to be tweaked
}

```

```

//*****FUNCIÓN QUE CALCULA LA TEMPERATURA
MEDIDA POR EL SENSOR LM35*****
void get_temperature()
{
    temperatura = analogRead (tempPin);
    temperatura = temperatura * 0.48828125;
}

```

```

//***** SI EL NAVEGADOR ENVIA PETICION PARA CAMBIAR ESTADO
DELOS LED, CAMBIA LA VARIABLE LED_state Y DE PASO ACTIVA O DESACTIVA
LA SEÑAL DE SALIDA DEL GALILEO*****
void SetEstados(void)
{
    //ESTADO DE VARIABLE LAVAR_AUTO
    if (StrContains(HTTP_req, "LAVAR_AUTO=1")) {
        lavar_auto = 1;
        digitalWrite(pin_out_man_auto,HIGH);
        //digitalWrite(pin_out_batidora,HIGH);
    }
    else if (StrContains(HTTP_req, "LAVAR_AUTO=0")) {
        lavar_auto = 0;
        digitalWrite(pin_out_man_auto,LOW);
        //digitalWrite(pin_out_batidora,LOW);
    }
}

```

```

//ESTADO DE LA VARIABLE QUE VERIFICA SI SE ESTA O NO LAVANDO EL
TANQUE
if (StrContains(HTTP_req, "LAVAR=1")) {
    lavando = 1;
    //digitalWrite(pin_out_lavar,HIGH);
}

```

```

    }
    else if (StrContains(HTTP_req, "LAVAR=0")) {
        lavando = 0;
        //digitalWrite(pin_out_lavar, LOW);
    }

    if (StrContains(HTTP_req, "BATIR")) {
        tiempo_off_batidora=60*1000*get_tiempo_batido(HTTP_req).toInt();//aqui se
        procesa la peticion del tiempo de batido y se refresca esa variable en el galileo.
    }

}

//***** ESTA FUNCIÓN RECIBE LA PETICIÓN HTTP DEL
// NAVEGADOR Y DESCOMPONE LA MISMA PARA SACAR DE ALLI EL TIEMPO OFF
// DE BATIR DE TANQUE*****
String get_tiempo_batido(char *HTTP_req)
{
    int i=0;
    int len = strlen(HTTP_req);
    lcd.clear();

    for(i=0;i<len;i++)
    {
        // en este if se pregunta si se encontro en el http request la palabra batir
        if(HTTP_req[i]=='B' && HTTP_req[i+1]=='A' && HTTP_req[i+2]=='T' &&
        HTTP_req[i+3]=='I' && HTTP_req[i+4]=='R' )
        {
            //en este if se pregunta de cuantas cifras es el numero junto a la palabra batir
            if(HTTP_req[i+5]=='1')
                return (String)(HTTP_req[i+6]);
            else if(HTTP_req[i+5]=='2')
            {
                return ((String)HTTP_req[i+6]+(String)HTTP_req[i+7]);
            }
        }
    }
}

//***** LEE LOS VALORES O ESTADOS DE LOS SENSORES, SWITCHES Y
// SALIDAS Y CREA UN XML DONDE ESCRIBE ESOS DATOS Y LOS ENVIA CUANDO
// EL NAVEGADOR LO SOLICITA*****
void XML_response(EthernetClient cl)
{
    int analog_val; // stores value read from analog inputs
    int count; // used by 'for' loop

    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");

```

//LEE DATOS DE LOS SENSORES PARA QUE SE ESCRIBAN EN EL XML... La etiqueta es solo un nombre para que el html la interprete, le pude llamar JUANITO jajaja)

```
cl.print("<xml_vol>");
cl.print(volumen_leche);
cl.println("</xml_vol>");
```

```
cl.print("<xml_temp>");
cl.print(temperatura);
cl.println("</xml_temp>");
```

//INFORMAR AL NAVEGADOR EL ESTADO DE SI ESTA EN AUTOLAVADO O MANUAL DE TANQUE Y TAMBIEN EL SI EL TANQUE SE ESTA LAVANDO O NO.

```
cl.print("<xml_lavar_auto>");
cl.print(lavar_auto);
cl.println("</xml_lavar_auto>");
```

```
cl.print("<xml_lavando>");
cl.print(lavando);
cl.println("</xml_lavando>");
```

//INFORMAR AL NAVEGADOR SI HAY O NO LECHE EN EL TANQUE

```
cl.print("<xml_hay_leche>");
cl.print(hay_leche);
cl.println("</xml_hay_leche>");
```

//INFORMAR AL NAVEGADOR EL TIEMPO DE OFF DE BATIDORA ACTUAL EN MINUTOS

```
int tiempo=tiempo_off_batidora/(1000*60);
cl.print("<xml_tiempo_bat>");
cl.print(tiempo);
cl.println("</xml_tiempo_bat>");
```

```
cl.print("</inputs>");
}
```

// sets every element of str to 0 (clears array)

void StrClear(char \*str, char length)

```
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}
```

// searches for the string sfind in the string str

```
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;

    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }

    return 0;
}
```

# CÓDIGO HTML DE PÁGINA WEB

## Index.html

```
<!DOCTYPE HTML>
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <title>Control de acceso</title>
  <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
  <style type="text/css">
    body{
      background: radial-gradient(#fff,#0489B1, #08088A);
    }
    .contenedor{
      width: 80%;
      max-width: 1000px;
      margin:0 auto;
      overflow: hidden;
    }
    header{
      background:#5882FA;
      margin:20px 0;
      color: #fff;
      border-style: outset;
      text-align: center;
      font-size: 1.5em;
      font-family: sans-serif;
    }
    .contenedor h1{
      color: #fff;
      font-size: 150%;
      margin:0px;
      text-shadow: black 0.1em 0.1em 0.2em
    }
    .contenedor h2{
      color: #0B0B3B;
      font-family: cursive;
      font-size: 120%;
      margin-bottom: 0px;
    }
    .contenedor img{
      width: 100%;
      height: auto;
    }
    .main{
      width:100%;
      padding: 10px;
      box-sizing:border-box;
    }
    #div_ingreso {
```

```

        background-color:#6E6E6E;
        padding-left:50px;
        padding-right:50px;
        padding-top:10px;
        padding-bottom:20px;
        width: 350px;
        left: 53%;
        position: absolute;
        margin-top:10px;
        margin-left: -260px;
        -webkit-border-radius: 30px;
    }
    footer{
        margin-top: 250px;
        box-sizing:border-box;
        width: 100%;
        padding: 5px;
        color: #FF4000;
        background: #A4A4A4;
    }
    .cajadetexto {
        font-family: serif;
        font-weight:500;/*grosor de la letra*/
        font-size: 18px;
        padding: 13px 13px 13px 54px;
        width:100%;
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        -ms-box-sizing: border-box;
        box-sizing: border-box;
    }

    .cajadetexto:focus{
        border: 3px solid #3498db;
        color: #3498db;
        padding: 13px 13px 13px 54px;
    }

    #usuario{
        background-image: url(usuario1.png);
        background-size: 30px 30px;
        background-position: 11px 8px;
        background-repeat: no-repeat;
    }
    #clave{
        background-image: url(clave1.png);
        background-size: 30px 30px;
        background-position: 11px 8px;
        background-repeat: no-repeat;
    }
    #btn_ingresar{
        cursor: pointer;
        float: left;
        background-color: #0174DF;

```

```

        border: #fff solid 4px;
        font-size: 20px;
        color: white;
    }
    #btn_limpiar{
        cursor: pointer;
        float: right;
        background-color: #0174DF;
        border: #fff solid 4px;
        font-size: 20px;
        color: white;
    }
    #btn_ingresar:hover {
        width: 120px;
        font-family: fantasy;
        background-color:#fff;
        border: #0174DF solid 4px;
        font-size: 20px;
        color: #0174DF;
    }
    #btn_limpiar:hover{
        width: 120px;
        font-family: fantasy;
        background-color:#fff;
        border: #0174DF solid 4px;
        font-size: 20px;
        color: #0174DF;
    }
    @media only screen and (max-width: 500px) {
        body{
            background: linear-gradient(#04B431,#5882FA);
        }
        #div_ingreso{
            left: 3%;
            margin-right: 3%;
            width: 88%;
            margin-left: 0;
            padding-left: 3%;
            padding-right: 3%;}
    }
</style>
<script type="text/javascript">
    function validar(){
        if (document.form.usuario.value == "admin" &
document.form.clave.value=="123")
        {
            window.open("menu.htm");
            document.form.usuario.value="";
            document.form.clave.value="";}
        else{
            alert('¡Usuario o Clave incorrectas, ingrese nuevamente!');
            document.form.usuario.value="";
            document.form.clave.value="";

```

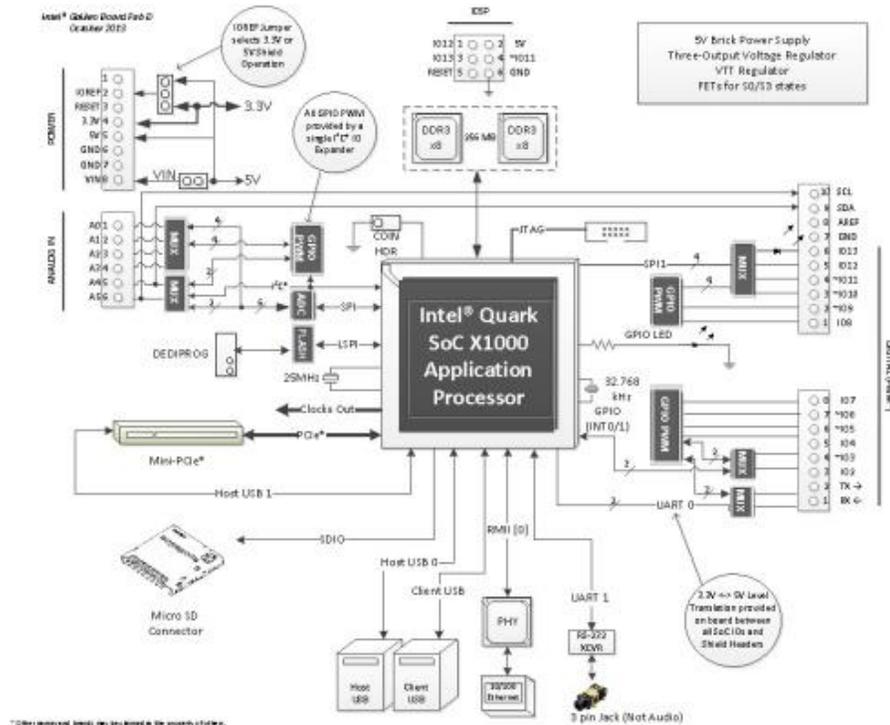
```

        document.form.usuario.focus();
    }
    function inhabilitar(){
        alert ("Esta función está inhabilitada.\n\nPerdonen las molestias.")
        return false
    }
    document.onclick="inhabilitar" ;
</script>
</head>
<body>
    <marquee bgcolor="#A4A4A4" width="100%" scrolldelay="10" direction="left"
loop="infinite">TELECONTROL DE UN SISTEMA DE CONSERVACIÓN DE TANQUE DE
LECHE</marquee>
    <div class="contenedor">
        <header>TELECONTROL DE UN SISTEMA DE CONSERVACIÓN DE TANQUE
DE LECHE</header>
        <section class="main">
            
            <h1>BIENVENIDOS:</h1>
            <h2>Introduzca un nombre de usuario y una contraseña valida para
ingresar al sistema:</h2>
            <div id="div_ingreso">
                <form name="form" class="form_ingreso" method="post" action="#">
                    <p><input type="text" name="usuario" id="usuario" class="cajadetexto"
placeholder="USUARIO"/></p>
                    <p><input type="password" name="clave" id="clave" class="cajadetexto"
placeholder="CLAVE" /></p>
                    <div>
                        <input type="submit" value="Ingresar"
id="btn_ingresar" value="Aceptar" onclick="validar()"/>
                        <input type="reset" value="Limpiar" id="btn_limpiar"/>
                    </div>
                </form>
            </div>
        </section>
        <footer>
            <p>
                Elaborado por:</br>
                <a href="https://www.facebook.com/maolycedillo">Josselyn
Cedillo A.</a></br>
                <a
href="https://www.facebook.com/cristhian.hernandez.908?fref=ts">Cristhian Hernández R.</a>
            </p>
        </footer>
    </div>
</body>
</html>

```

# DIAGRAMA DE RECURSOS UTILIZADOS

## TARJETA INTEL GALILEO GEN 1



## SENSOR ULTRASÓNICO HC-SR04

### Parámetros eléctricos

Dimensiones del circuito	43 x 20 x 17 mm
Voltaje de trabajo	DC 5v
Corriente de trabajo	15 mA
Frecuencia de trabajo	40 Hz
Rango máximo	4.5 m
Rango mínimo	1.7 cm
Duración mínima del pulso de disparo (nivel TTL)	10 μS
Duración del pulso eco de salida (nivel TTL)	100-25000 μS

Tiempo mínimo de espera entre una medida y el inicio de otra	20 S
--	------

Pines de conexión

	<p>VCC  Trig (Disparo del ultrasonido)  Echo (Recepción del ultrasonido)  GND</p> <p>Distancia = {(Tiempo entre Trig y el Echo) * (V.Sonido 340 m/s)}/2</p>
--	---

**SENSOR ANALÓGICO LM 35**

Parámetros eléctricos

Voltaje de funcionamiento	5V (DC)
Corriente estática	<2 mA
Trabajo actual	15mA
Frecuencia de trabajo	40KHz
Ángulo Eficaz	<15 °
Distancia de detección	2 cm - 450 cm
Resolución	0,3 cm
Medición de ángulo	30 °
Disparo de la señal de entrada	TTL pulso 10µs
Echo señal de salida	señal PWL de TTL



## BIBLIOGRAFÍA

- [1] El Telégrafo, Producción lechera mueve \$ 700 millones al año, <http://www.telegrafo.com.ec/economia/item/produccion-lechera-mueve-700-millones-al-ano.html>, fecha de publicación septiembre 2013.
- [2] Explored, El precio oficial del litro de leche ayudó al sector a mejorar sus condiciones, <http://www.explored.com.ec/noticias-ecuador/el-precio-oficial-del-litro-de-leche-ayudo-al-sector-a-mejorar-sus-condiciones-489803.html>, fecha de publicación julio 2011.
- [3] Monografias.com, Automatización industrial, <http://www.monografias.com/trabajos76/automatizacion/automatizacion.shtml>, fecha de publicación diciembre de 2009.
- [4] IdosE Ingeniería, Informática y Electrónica, Sistema embebido, <http://www.idose.es/sistemas-embebidos>, fecha de consulta diciembre 2014.
- [5] José Luis Molina Marticorena, Sensores, [http://www.profesormolina.com.ar/tecnologia/sens\\_transduct/que\\_es.htm](http://www.profesormolina.com.ar/tecnologia/sens_transduct/que_es.htm), fecha de consulta diciembre 2014.

[6] Wikipedia , Actuadores , <http://es.wikipedia.org/wiki/Actuador>, fecha de publicación noviembre 2014.

[7] Monografias.com , Servidores web, <http://www.monografias.com/trabajos75/servidores-web/servidores-web2.shtml>, fecha de publicación octubre de 2009.

[8] Arduino, Intel Galileo, <http://arduino.cc/en/ArduinoCertified/IntelGalileo>, fecha de consulta octubre 2014.

[9] Arduino, LiquidCrystal - Hello World , <http://arduino.cc/en/Tutorial/LiquidCrystal>, fecha de consulta octubre 2014.

[10] El cajón de Ardu, Sensor Ultrasónico HC-SR04, <http://elcajondeardu.blogspot.com/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html>, fecha de publicación marzo del 2014.

[11] Wikipedia, Lm35, <http://es.wikipedia.org/wiki/LM35>, fecha de consulta octubre 2014.

[12] Linksys, Router cisco e900, <http://www.linksys.com/es-eu/products/routers/E900>, fecha de consulta enero 2015.

- [13] Wikipedia, Protocolo TCP/IP, <http://www.monografias.com/trabajos/protocolotcpip/protocolotcpip.shtml>, fecha de publicación noviembre de 1999.
- [14] Definición abc, Definición de HTTP, <http://www.definicionabc.com/tecnologia/http.php#ixzz3S4mgAZWu>, fecha de consulta enero 2015.
- [15] Arduino, Libraries, <http://arduino.cc/en/reference/libraries>, fecha de consulta diciembre 2014.
- [16] TecBolivia, Sensor Ultrasónico HC-SR04 con Arduino, <http://tecbolivia.com/index.php/descargas/category/2-tutorial-de-arduino-intermedio>, fecha de consulta noviembre 2014.
- [17] Intel Corporation, Función pulseIn, <https://communities.intel.com/thread/45708>, fecha de publicación julio 2014.
- [18] Starting Electronics, Basic Arduino Web Server, <http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/basic-web-server/>, fecha de publicación 2013.

[19] Starting Electronics , SD Card Web Server I/O,  
<http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/SD-card-IO/>, fecha de publicación abril 2013.

## BIBLIOGRAFÍA DE IMAGENES

- [1] Instituto nacional de tecnología industrial, Simposio Argentino de Sistemas Embebidos - SASE2012”, [http://www.inti.gob.ar/noticias/fotos\\_notas/simposio.jpg](http://www.inti.gob.ar/noticias/fotos_notas/simposio.jpg) , fecha de publicación agosto 2012.
- [2] Droide Comunidad – Robótica Educativa-Programación-3D , Sensores, <http://droidecomunidad.com/wp-content/uploads/2013/05/Tinkerkit-aula.jpg>, fecha de publicación mayo 2013.
- [3] LEKU-ONA , Actuadores, [http://www.leku-ona.com/wp-content/uploads/2012/12/naumatica\\_05.jpg](http://www.leku-ona.com/wp-content/uploads/2012/12/naumatica_05.jpg), fecha de publicación diciembre 2012.
- [4] Arduino, Intel galileo, [http://arduino.cc/en/uploads/ArduinoCertified/IntelGalileo\\_fabD\\_Front.jpg](http://arduino.cc/en/uploads/ArduinoCertified/IntelGalileo_fabD_Front.jpg), fecha de consulta octubre 2014.
- [5] Instructables, Intel Galileo”, <http://cdn.instructables.com/F24/N76Z/I2TG1JRR/F24N76ZI2TG1JRR.LARGE.jpg>, fecha de publicación enero 2014.
- [6] Bluetip electrónica-robótica, Modulo LCD, <http://www.bluetip.com.uy/image/cache/data/Componentes/LCD/lcd03-16x2b-500x500.png>, fecha de consulta enero 2015.

- [7] Fritzing, Sensor ultrasónico hc-sr04, <http://fritzing.org/media/fritzing-repo/projects/h/hc-sr04-project/images/HC-SR04-2.jpg>, fecha de consulta enero 2015.
- [8] Rastating, Sensor analógico lm35, [http://blog.rastating.com/content/images/2014/Feb/LM35\\_500x500\\_1\\_.jpg](http://blog.rastating.com/content/images/2014/Feb/LM35_500x500_1_.jpg), fecha de publicación febrero 2014.
- [9] Linksys, Router cisco e900, [http://www.linksys.com/videos/gallery/631/802/E900\\_right-angle-850x425.png](http://www.linksys.com/videos/gallery/631/802/E900_right-angle-850x425.png), fecha de consulta enero 2015.
- [10] Conecta Magallanes, Redes Computacionales, <http://conectamagallanes.cl/wp-content/uploads/2014/10/red.png>, fecha de publicación octubre 2014.
- [11] rocixcode , Códigos y significado de los Errores Http, <https://rocixcode.files.wordpress.com/2011/05/cliente-servidor.png>, fecha de publicación mayo 2011.
- [12] 1and1, GeoTrust dedicated SSL certificate, <http://www.1and1.com/details-ssl-certificate>, fecha de consulta febrero 2015.

[13] Trustwave, How SSL Works,  
<https://ssl.trustwave.com/support/support-how-ssl-works.php>, fecha de  
consulta febrero 2015.