



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electrónica y Computación

“TELECONTROL DE BIODIGESTOR PARA GENERACIÓN DE
GAS”

TESINA DE SEMINARIO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN TELEMÁTICA

Presentado por:

Pablo Andrés Martillo Campoverde

Ronald Eduardo Orellana Verdezoto

Guayaquil – Ecuador

2015

AGRADECIMIENTO

A Dios por haber guiado todos y cada de uno de los pasos que he dado a largo de mi preparación. A mis padres, por ser los pilares en mi vida y las personas a las que quiero siempre hacer sentir orgullosos. A mi hermano por todos sus consejos y enseñanzas. A la empresa LA LLAVE S.A., por su apoyo y gestión con los equipos.

Pablo Andrés Martillo Campoverde

A Dios por estar siempre conmigo y darme la fortaleza y sabiduría necesaria para seguir adelante en este sendero de aprendizaje. A mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación, forjando la persona que soy y apoyarme incondicionalmente en todo momento. A mi hermana por creer siempre y haber depositado su confianza en mí.

Ronald Eduardo Orellana Verdezoto

DEDICATORIA

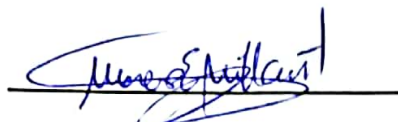
A mis familias, todos han estado siempre pendiente de mí y me han apoyado en diversas formas para poder alcanzar esta meta. A mi novia Adriana, quién me ha acompañado toda mi vida universitaria, brindándome su apoyo día a día y en los momentos más difíciles.

Pablo Andrés Martillo Campoverde

Mis padres y hermana, a ellos les dedico todo el esfuerzo y sacrificio entregado para conseguir esta meta, esperando siempre llenarlos de orgullo y satisfacción. A mi novia Karem, quien ha sido mi motivación y mi apoyo en esta última etapa universitaria. A todas aquellas personas que de uno u otra manera han sido parte de esta etapa de mi vida.

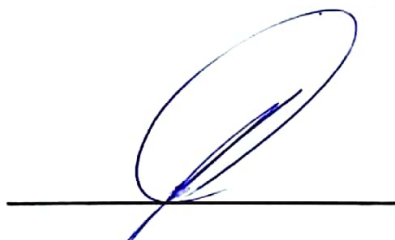
Ronald Eduardo Orellana Verdezoto

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in blue ink, appearing to read 'Marcos Millán T.', is written over a solid horizontal black line.

M.Sc. Marcos Millán T.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in blue ink, appearing to read 'José Menéndez', is written over a solid horizontal black line.

M.Sc. José Menéndez

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

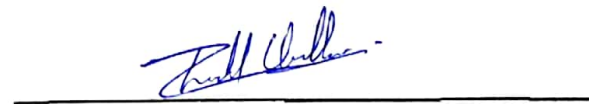
DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.”

(Reglamento de Graduación de la ESPOL)

A handwritten signature in blue ink, appearing to read 'P. Martillo', is written above a solid horizontal line.

Pablo Andrés Martillo Campoverde

A handwritten signature in blue ink, appearing to read 'Ronald Orellana', is written above a solid horizontal line.

Ronald Eduardo Orellana Verdezoto

RESUMEN

El principal objetivo de este trabajo, ha sido la elaboración de un prototipo de sistema para controlar remotamente el proceso de un biodigestor para la generación de gas metano, partiendo de la transformación natural de los desechos orgánicos durante su descomposición, la misma que sucede en el interior del biodigestor. Para la realización del mismo se emplearon diferentes tecnologías, con la finalidad de monitorear todas las variables que se presentarán en el desarrollo, y de esta manera controlar cada una de las etapas del mismo, y así poder optimizar los recursos empleados para la producción de gas.

Para la implementación de este modelo se utilizaron sensores electrónicos de PH, presión, temperatura y de nivel, que examinan las variables del proceso; los datos obtenidos se procesan en un controlador lógico programable, para interactuar mediante electroválvulas y motores, con el procedimiento.

El proceso se controla remotamente mediante mensajes de texto y una aplicación web, la misma que permitirá una mayor facilidad para el usuario, ya que podrá manipular desde cualquier sitio el biodigestor.

ÍNDICE GENERAL

RESUMEN.....	viii
ÍNDICE GENERAL	ix
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE TABLAS.....	xv
INTRODUCCIÓN.....	xvi
CAPÍTULO 1.....	1
1. ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1. OBJETIVO GENERAL.....	1
1.2. OBJETIVOS ESPECÍFICOS.....	1
1.3. INFORMACIÓN GENERAL	2
1.4. IMPACTO AMBIENTAL	5
1.5. IMPACTO SOCIOECONÓMICO.....	6
1.6. IMPACTO TECNOLÓGICO	8
CAPÍTULO 2.....	10
2. BIODIGESTOR.....	10
2.1. CONCEPTO DE BIODIGESTOR.....	10
2.2. CLASIFICACIÓN DE BIODIGESTOR.....	12
2.2.1. CONTINUIDAD DE CARGA DE MATERIA	12
2.2.1.1. DISCONTINUOS.....	13
2.2.1.2. SEMI CONTINUO	13

2.2.1.3. CONTINUOS	14
2.2.2. ESTRUCTURA.....	14
2.2.3. FORMA	15
2.2.4. ALMACENAMIENTO DE GAS	15
2.2.4.1. DOMO FIJO	16
2.2.4.2. DOMO MÓVIL.....	16
2.2.4.3. DEPÓSITO FLOTANTE	16
2.2.4.4. GASÓMETRO DE CAUCHO O PLÁSTICO	16
2.3. CARACTERÍSTICAS Y PRINCIPIO DE FUNCIONAMIENTO.....	17
CAPÍTULO 3.....	19
3. DISEÑO DEL SISTEMA DE TELECONTROL DEL BIODIGESTOR	19
3.1. CONTROLADOR.....	20
3.1.1. PLC	20
3.1.2. DISEÑO DE AUTOMATIZACIÓN	25
3.1.2.1. SISTEMA DE TANQUES	25
3.1.2.2. ETAPA DE CARGA.....	27
3.1.2.3. ETAPA DE DIGESTIÓN.....	33
3.2. SENSORES.....	35
3.2.1. SENSOR DE NIVEL	36
3.2.2. SENSOR DE TEMPERATURA.....	37
3.2.3. SENSOR DE PH	38
3.2.4. SENSOR DE PRESIÓN	39

3.3. ACTUADORES	40
3.3.1. ELECTROVÁLVULA	40
3.3.2. MOTOR ELÉCTRICO.....	41
3.4. HMI	41
3.4.1. PANTALLA.....	42
3.4.2. APLICACIÓN WEB.....	47
3.4.3. SISTEMA SMS	54
3.5. COMUNICACIONES	58
3.5.1. PANTALLA – CONTROLADOR.....	58
3.5.2. FRAMEWORK DE APLICACIÓN WEB	60
3.5.3. CONTROLADOR – SERVIDOR WEB	66
CAPÍTULO 4.....	72
4. PRUEBAS Y RESULTADOS	72
4.1. CALIBRACIÓN DE SENSORES	72
4.1.1. CALIBRACIÓN DE SENSOR DE NIVEL	72
4.1.2. CALIBRACIÓN DE SENSOR DE TEMPERATURA.....	73
4.1.3. SENSOR DE PH	74
4.1.4. SENSOR DE PRESIÓN	76
4.2. PRUEBAS DE FUNCIONAMIENTO	77
4.2.1. PRUEBA DE ACTUADORES	82
4.3. RESULTADOS	84
CONCLUSIONES	86

RECOMENDACIONES.....	90
BIBLIOGRAFÍA.....	92
ANEXOS	96
Anexo A: Diagrama de Bloque General del Proceso.	96
Anexo B: Escalador.....	96
Anexo C: Programa de Ejemplo de Prosoft, con los cambios realizados.....	97
Anexo D: Adaptación de la pantalla al controlador.	134
Anexo E: Programación Etapa de Carga.....	136
Anexo F: Programación Etapa Digestión.....	140
Anexo G: Mapeo de Periféricos y Configuraciones marcha y Paro...	142
Anexo H: Modelo de la aplicación WEB.	148
Anexo I: Vista Página Principal de la aplicación WEB.	148
Anexo J: Vista Etapa de Carga de la aplicación WEB.	151
Anexo K: Vista Sistema de Tanques de la aplicación WEB.	152
Anexo L: Vista Etapa de Digestión de la aplicación WEB.....	154
Anexo M: Vista Monitoreo de Variables de la aplicación WEB.	156
Anexo N: Vista Crear Usuario de la aplicación WEB.	160
Anexo O: Vista Ingreso al sistema de la aplicación WEB.	164
Anexo P: Controlador del Proceso de la aplicación WEB.	167
Anexo Q: Javascript de consulta cada segundo de la aplicación WEB.	224

ÍNDICE DE FIGURAS

Figura 1.1 Clasificación de tipos de Energía	4
Figura 2.1 Proceso de Biodigestión.	12
Figura 3.1 Diagrama de bloques del sistema planteado	20
Figura 3.2 PLC Allen Bradley con todos sus módulos integrados.	22
Figura 3.3 PLC Allen Bradley Micro850.	22
Figura 3.4 Sistema de tanques.	27
Figura 3.5 Ilustración del tanque de carga.	31
Figura 3.6 Diagrama de flujo de datos de la etapa de carga	32
Figura 3.7 Diagrama de flujo de datos de la etapa de digestión.....	35
Figura 3.8 Pantalla local de inicio del sistema.....	43
Figura 3.9 Pantalla local del sistema de tanques.	44
Figura 3.10 Pantalla local de la etapa de carga	45
Figura 3.11 Pantalla local de la etapa de digestión.	46
Figura 3.12 Pantalla local de monitoreo de variables 1.....	46
Figura 3.13 Pantalla local de monitoreo de variables 2.....	47
Figura 3.14 Pantalla de inicio de sesión de aplicación web.....	49
Figura 3.15 Pantalla principal del sistema de aplicación web.....	49
Figura 3.16 Pantalla de creación de usuarios de aplicación web.	50
Figura 3.17 Pantalla de usuario con privilegios restringidos de aplicación web.	50
Figura 3.18 Pantalla de sistema de tanques de aplicación web	51

Figura 3.19 Pantalla de etapa de carga de aplicación web.	52
Figura 3.20 Pantalla de etapa de digestión de aplicación web.	53
Figura 3.21 Pantalla de monitoreo de variables de aplicación web.	53
Figura 3.22 Diagrama de envío de alertas por mensajes de texto.	56
Figura 3.23 Diagrama de envío y recepción de estado de variables por mensajes de texto.....	57
Figura 3.24 Ubicación del protocolo CIP en el modelo OSI.....	60
Figura 3.25 Diagrama de bloques de secuencia de datos.	62
Figura 3.26 Diagrama UML de la base de datos.....	64
Figura 3.27 Interfaz de programación de Visual Studio Express.	65
Figura 3.28 Estructura de mensajes Modbus.....	68
Figura 3.29 Mapeo de variables Modbus TCP.....	70
Figura 3.30 Librería del protocolo Modbus para lenguaje C#.....	71
Figura 4.1 Circuito sensor de nivel.....	73
Figura 4.2 Circuito sensor de temperatura.....	74
Figura 4.3 Punta de prueba de pH American Marine PINPOINT.....	75
Figura 4.4 Circuito amplificador de señal de pH.....	76
Figura 4.5 Sensor de presión MPX5700GP.....	77
Figura 4.6 Maqueta.....	80
Figura 4.7 Conexiones del panel de control.....	82
Figura 4.8 Electroválvula genérica de 12v.	83
Figura 4.9 Motor de limpiaparabrisas de 12v.	84

ÍNDICE DE TABLAS

Tabla # 1. Tabla de direccionamiento Modbus RTU y TCP.....	69
--	----

INTRODUCCIÓN

En la actualidad, el Ecuador está experimentando cambios en la matriz productiva, una evidencia de ello, es que la utilización de gas para el consumo doméstico, está siendo reemplazada por el uso de cocinas de inducción. Por lo tanto, el estado dejará de subsidiar el gas, lo cual hará que el precio del mismo se eleve. Esta medida afectará principalmente a la zona rural, donde se emplea el gas como fuente de energía.

Ante esta problemática, se plantea utilizar una fuente de energía renovable muy poco empleada en el país, como lo es la producción de gas a partir de la descomposición de desechos orgánicos. Este tipo de energía está catalogada como energía de biomasa residual, que es parte de la clasificación de energías renovables o limpias. En zonas rurales, los desechos orgánicos animales y vegetales son abundantes y apropiados para la producción de gas.

Un biodigestor es un sistema que permite la utilización de desechos orgánicos para la producción de gas metano y fertilizante. La implementación del mismo no es compleja, sin embargo el producto obtenido no es óptimo, debido a que no se controlan las diversas variables que intervienen en el proceso, como lo son la temperatura a la que trabaja el biodigestor, condiciones climáticas, el nivel de pH que posee la mezcla desechos-agua, el control de las proporciones

introducidas en el biodigestor o si simplemente no se lleva un control de la cantidad de gas generado.

Para poder controlar y monitorear remotamente este proceso, se ha implementado la idea de manipularlo mediante mensajes de texto o a través de internet. De esta manera se facilita el acceso al sistema y se ahorra la movilización hasta el sitio donde está construido el biodigestor.

Además se ha implementado un sistema de alarmas, que permite mantener informado al usuario acerca de alguna anomalía que presente el sistema.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN

1.1. OBJETIVO GENERAL

Implementar un sistema de telecontrol de un biodigestor para manipular el proceso de la generación de biogás, ofreciendo a los usuarios herramientas tecnológicas para la administración del mismo; de forma local mediante una pantalla táctil, y remotamente a través una página web, sumando un sistema de alertas por medio de mensajes de texto.

1.2. OBJETIVOS ESPECÍFICOS

- Identificar las variables más importantes que intervienen en el proceso de biodigestión y utilizar los sensores adecuados para la medición de estas variables.

- Configurar un dispositivo lógico programable (PLC), capaz de encargarse de automatizar el proceso en base a los sensores de entrada.
- Utilizar una interfaz gráfica en sitio, donde el usuario pueda observar el estado del proceso.
- Diseñar una aplicación web, capaz de manejar diversos perfiles de usuarios con sus permisos respectivos.
- Configurar un sistema de alertas que permita enviar mensajes de texto a números de teléfonos previamente establecidos, durante el desarrollo del sistema.
- Asegurar la correcta interacción de los diversos componentes y que el diseño de las interfaces de administración resulten amigables para el usuario.

1.3. INFORMACIÓN GENERAL

Resulta necesario conocer los factores externos a la implementación de un biodigestor, para poder cuantificar cuáles son realmente sus beneficios, y para ello, se procede a redactar los problemas en diversos ámbitos para los cuales un biodigestor, se convierte en una solución alterna o innovadora. Entre las ventajas que posee un biodigestor, se considera que lo más importante es el impacto generado en el aspecto ambiental y socioeconómico. El valor agregado de esta implementación consiste en brindar una solución

automatizada a los dueños de haciendas y fincas, para que sean capaces de controlar la generación de biogás, los factores asociados al biodigestor y realizarlo de forma remota.

Cabe recalcar que todo el análisis que se presentará a continuación, se realizó basándose en la familias, haciendas o fincas en la zona rural, que es donde se puede explotar las bondades de un biodigestor, mas no en la zona urbana, donde un biodigestor solamente sería una opción para una empresa con amplio espacio, y con la necesidad de generar su propio gas, pero no aplicado a las familias de esta zona.

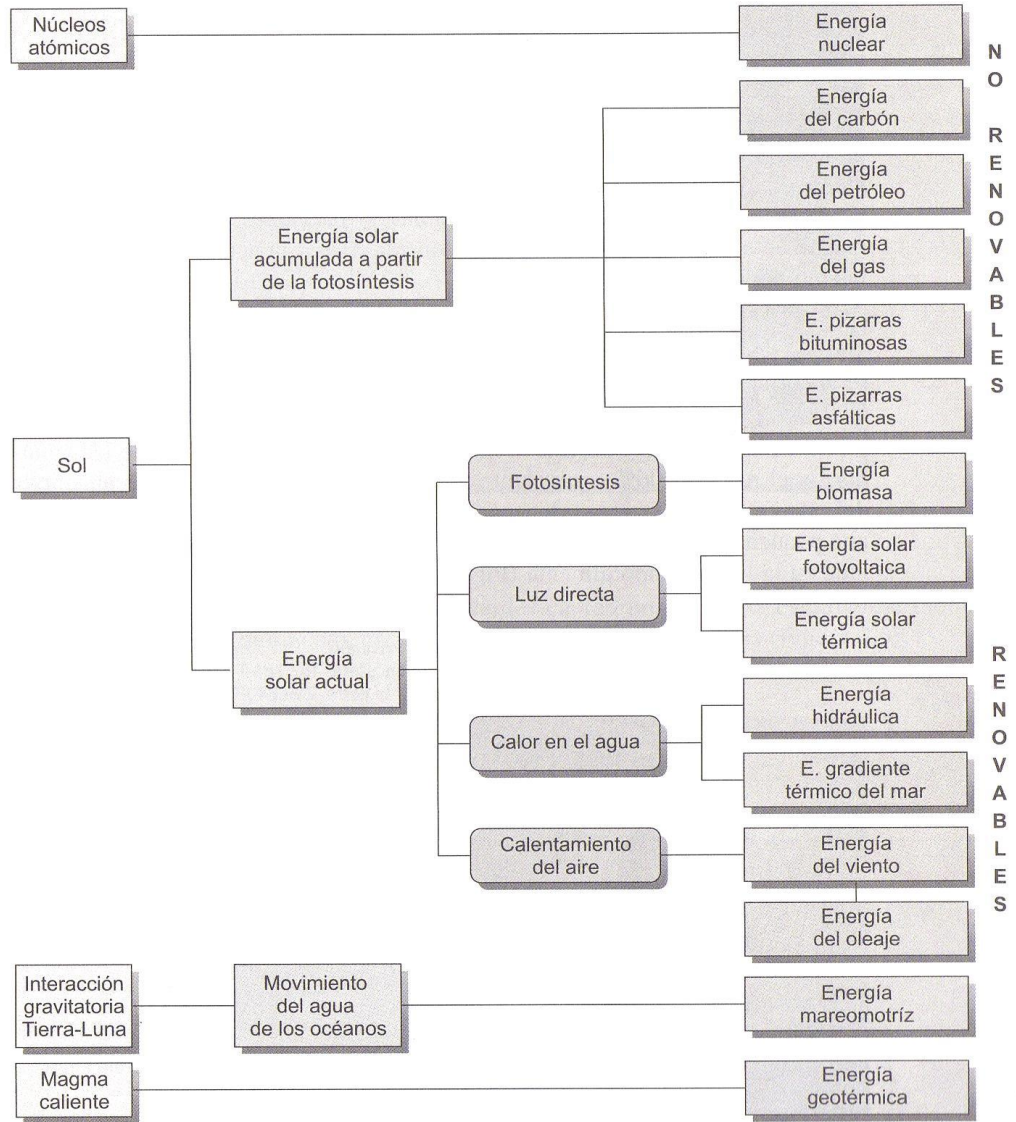


Figura 1.1 Clasificación de tipos de Energía

1.4. IMPACTO AMBIENTAL

La mala administración de los desperdicios genera contaminación ambiental, debido a que estos pueden acabar en el mar o acumularse, generando mal olor, atrayendo plagas o desembocando en emisiones de gases. Un biodigestor genera un gran impacto ambiental, dado que promueve la reutilización de los desechos orgánicos, evitando que estos generen contaminación.

En las zonas rurales ciertas familias han optado por la utilización de leña para la cocción de los alimentos, lo cual claramente fomenta la deforestación de los bosques, siendo una tala innecesaria y en muchos casos desmesurada, ya que estas son zonas aptas para emplear el estiércol de animales en el biodigestor, de esta manera se genera el gas y se impide la tala de árboles. “Se ha calculado que un 1m^3 de biogás utilizado para cocinar, evita la deforestación de 0,335 hectáreas de bosque con un promedio de 10 años de vida de los árboles” [1].

Los desechos, si no son bien tratados pueden desembocar en algún río cercano, provocando una serie de perjuicios para la población como: el taponamiento de los afluentes, las inundaciones, la pérdida de sembríos, entre otros; a esto hay que sumarle que para la limpieza de los establos, donde se encuentra el ganado, se debe usar agua en grandes cantidades. Sin embargo

con la implementación de un biodigestor en el lugar, los residuos se convierten en la base para el funcionamiento del sistema de biodigestión, generando una administración adecuada de los desechos y evitando el desperdicio del agua, evidenciando que con la utilización de un biodigestor, el más beneficiado es el medio ambiente.

1.5. IMPACTO SOCIOECONÓMICO

En Ecuador se plantea el cambio de la matriz productiva, con esto se propone la utilización de la electricidad para cocinar, dejando de lado el uso del gas licuado. Este gas hasta el 2016 será subsidiado, con lo cual el costo del tanque de gas aumentará para los ciudadanos en una cantidad considerable. Actualmente se establece que el costo del tanque de gas doméstico para la venta al público es de \$2.50 con el subsidio, mientras que sin él, lo más probable es que el valor se establezca cercano al que se tiene para el uso comercial, aproximadamente \$16.

Teniendo esto a consideración, hay que notar claramente las diferencias entre las zonas urbanas y rurales. La zona urbana en su mayoría cuenta con una buena alimentación eléctrica, favoreciendo el plan del gobierno actual, mismo que propone una estrategia para la adquisición de las cocinas de inducción y la instalación de líneas de 220V. Mientras en las zonas rurales, resulta muy complicado tener una instalación de 220V, además la pérdida del suministro

de energía eléctrica resulta muy común, lo cual genera un gran desafío para esta población. El adquirir una cocina de inducción mediante los planes que brinda el gobierno, puede resultar asequible, pero la gran mayoría seguramente tendrá problemas con el consumo eléctrico.

En las empresas agroindustriales, se piensa más allá que en el subsidio del gas, ya que este no se aplica en fines comerciales. En muchos artefactos se requiere la utilización del gas, es por eso que un biodigestor en este tipo de negocios, es una solución viable y que puede generar mayor productividad.

La gran ventaja de las zonas rurales es que, a diferencia de las zonas urbanas, poseen amplio espacio para la implementación de un biodigestor. Adicionalmente gozan de una gran cantidad de animales para la generación de excrementos. El biodigestor es una alternativa versátil, ya que en estas zonas es muy común encontrar ganado, y con sus desechos se puede generar el gas. Con esto se evitan la adquisición de lo que serán los costosos tanques de gas y tendrán una nueva alternativa de muy bajo costo, que se suma a la opción de utilizar las cocinas de inducción.

Otro factor importante es el ahorro en abonos químicos, ya que las familias o haciendas dedicadas a la producción agrícola, pueden utilizar los residuos de biosol, generados por el biodigestor como abono natural, puesto que estos al

igual que los abonos químicos, resultan con altos niveles de nitrógeno, fósforo y potasio.

1.6. IMPACTO TECNOLÓGICO

El uso de la tecnología en las zonas rurales del país, es un elemento que está en crecimiento, debido a que las necesidades son mayores. Con el tiempo las personas han comprendido que el uso de herramientas tecnológicas genera mayor productividad, y les permite obtener mejoras en las ganancias, sin importar la rama a la que se dediquen. Por ello las personas de las zonas rurales, cada vez salen más en búsqueda de nuevas soluciones tecnológicas.

Dentro del proceso de biodigestión se incluyen diversas variables que sin el uso de los equipos adecuados, resulta imposible medirlas. Esto resulta problemático dado a la falta de información de las personas que lo utilizan. El problema central es la falta de control de los factores que afectan el rendimiento del biodigestor. Si se domina las variables que intervienen en el proceso, se tomarán mejores decisiones sobre la generación del biogás.

Es aquí donde se implementa la opción de saber el estado de las variables y un sistema que mantenga estos valores óptimos, de esta manera se podrá sacar un real provecho de un biodigestor, a través de una interfaz y

herramienta de administración que sea muy visual y permita entender a las personas lo que está sucediendo con él.

Otro de los graves problemas de control del proceso, se da debido a que el correcto desempeño del biodigestor se encuentra directamente relacionado con el grado de atención, que se le brinda a las variables del mismo. El administrador del biodigestor suele ser el dueño de la hacienda, o en el caso de las familias, el responsable del hogar, pero este no puede estar siempre en el lugar donde se encuentra el biodigestor. Para él resulta clave tener una herramienta que le permita tener la movilidad necesaria, sin dejar de estar al tanto de las variables de este.

Si un administrador no tiene un estadístico de la cantidad de gas generado, no será eficiente el proceso y se desperdiciarán los recursos, por lo que se plantea la herramienta en la que el administrador se mantenga siempre informado de lo que ocurre, con su sistema de digestión de desechos, ya sea consultando el estado del proceso o por el aviso de alarmas de fugas o falta de materia prima, sin importar su ubicación física, mientras tenga acceso a la red.

CAPÍTULO 2

2. BIODIGESTOR

2.1. CONCEPTO DE BIODIGESTOR

Un biodigestor o digestor de desechos orgánicos es, en términos generales, un contenedor hermético e impermeable donde se colocan materiales orgánicos animales o vegetales a fermentar. Este contenedor también es llamado reactor.

El sistema puede ser constituido por varias etapas, en las cuales puede incluir una cámara de carga y nivelación de agua residual antes del digestor, una cámara para depositar el biogás, filtros para tratamiento posterior de biogás y biosol.

El proceso de biodigestión ocurre debido a la presencia de un conjunto de microorganismos bacterianos anaeróbicos presentes en los desechos fecales, que al mezclarse con los desechos orgánicos de origen vegetal y animal, se crea una combinación de gases con alto contenido de metano también llamado biogás, el cual es excesivamente eficiente al emplearse como combustible. El resultado de este fenómeno, es la obtención de desechos con un alto grado de concentración de nutrientes y materia orgánica, los cuales son excelentes fertilizantes que se pueden utilizar al instante, debido a que el proceso de digestión anaerobio elimina el mal olor y la proliferación de insectos.

Se debe tener en cuenta condiciones fundamentales en el proceso como: pH, presión y temperatura, con la finalidad de obtener óptimos resultados.

Implementar un biodigestor es sencillo y económico, dependiendo de los materiales a emplearse, este proceso se está efectuando mucho en comunidades rurales alejadas y subdesarrolladas donde se busca solventar el problema energético-ambiental y también un correcto tratamiento de los desechos animales y vegetales.

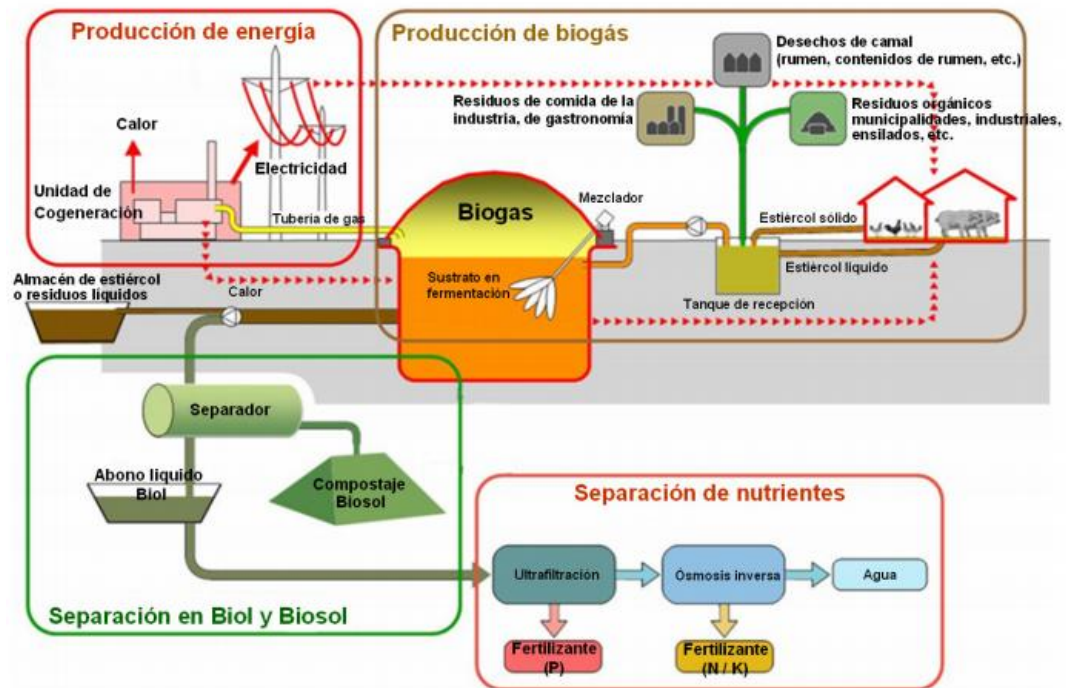


Figura 2.1 Proceso de Biodigestión.

2.2. CLASIFICACIÓN DE BIODIGESTOR

Los biodigestores se clasifican dependiendo de varios parámetros como:

- Continuidad de carga de materia orgánica.
- Estructura.
- Forma.
- Depósito de biogás.

2.2.1. CONTINUIDAD DE CARGA DE MATERIA

Se subclasifican en:

2.2.1.1. DISCONTINUOS

También conocido como sistema batch, consiste en cargar el digestor totalmente una sola vez y la extracción del residuo se realiza cuando la producción de biogás haya terminado. Comúnmente consta de un tanque cerrado herméticamente conectado a un gasómetro, que es un depósito donde se almacena el gas producido. Este biodigestor se usa cuando la obtención de materia orgánica no es constante, por lo que, si se quiere una producción de biogás asidua, habría que conectar varios digestores y cargarlos a diferentes tiempos.

Este sistema es idóneo para emplearse en laboratorios y así estudiar el comportamiento de la generación de biogás y biosol. La construcción de este digestor es fácil de implementar y se puede llegar a generar gas en una relación 1:1 con la carga.

2.2.1.2. SEMI CONTINUO

Este modelo de biodigestor es el más aplicado en las zonas rurales de manera doméstica. Generalmente son contenedores enterrados, que se cargan de materia diariamente por acción de la gravedad. La cantidad de desechos orgánicos dependen del tiempo de fermentación, y si se cuidan las condiciones

del proceso, la producción de biogás puede llegar a ser constante. El uso de un gasómetro es opcional, ya que el biogás se aloja en la parte superior del biodigestor, desplazando los desechos orgánicos debido a la presión generada.

Existen dos modelos principales de este tipo, el chino y el hindú, los cuales no han tenido mucha aceptación debido a la dificultad de su implementación y elevados costos. El modelo chino posee un domo fijo, mientras que el hindú tiene un domo flotante.

2.2.1.3. CONTINUOS

Este tipo de biodigestor es destinado principalmente para el procesamiento de aguas residuales. Usualmente son instalaciones considerables, donde es imperioso el uso de equipos especializados para los procesos que intervienen, así como para su control. Este modelo de digestor es de tipo industrial, genera abundante cantidad de biogás utilizado en la industria, además de ser empleado para el procesamiento de aguas negras.

2.2.2. ESTRUCTURA

De acuerdo a su estructura, los biodigestores se clasifican en:

- Digestores de ladrillo.

- De mampostería.
- De hormigón.
- De hormigón armado.
- De materiales plásticos.
- De metales como acero inoxidable.

2.2.3. FORMA

Dependiendo de su forma geométrica, los biodigestores han sido desarrollados para optimizar el proceso de fermentación y extracción del biogás producido, se clasifican de la siguiente manera:

- Digestores de cámara vertical cilíndrica.
- De cámara esférica.
- De cámara ovalada.
- De cámara rectangular.
- De cámara cuadrada.

2.2.4. ALMACENAMIENTO DE GAS

Estos biodigestores también se clasifican según el almacenamiento del biogás:

2.2.4.1. DOMO FIJO

El biogás producido es utilizado constantemente en quemadores, por lo que no se mantiene almacenado mucho tiempo. Se lo implementa frecuentemente con sistemas de llenado semi-continuo y presentan presiones altas si el gas no se utiliza constantemente.

2.2.4.2. DOMO MÓVIL

En este tipo de digestores, la presión se mantiene invariable debido a que la cúpula del mismo, se eleva a medida que la presión aumenta, por lo tanto no existe riesgo de explosión por sobrepresión.

2.2.4.3. DEPÓSITO FLOTANTE

El almacenamiento de biogás es externo al digestor. Consta de contenedor suspendido sobre el agua, brindando movilidad y presión constante al desplazar el agua de su interior.

2.2.4.4. GASÓMETRO DE CAUCHO O PLÁSTICO

De igual manera que el depósito flotante, el almacenamiento de gas es externo al digestor. Consta de un contenedor de caucho o plástico de alta densidad, que permite soportar la presión de gas generada.

2.3. CARACTERÍSTICAS Y PRINCIPIO DE FUNCIONAMIENTO

Digestor o contenedor: Depósito hermético e impermeable, fabricado de materiales no corrosivos que no reaccionen al interactuar con desechos orgánicos o humedad. En su interior se produce la fermentación anaerobia.

Entrada de carga o mezcla: Depósito donde se combinan desechos orgánicos, agua y otros. Aquí se homogeniza la materia para reducir desechos grandes, para enviarlas al digestor.

Gasómetro: Depósito de almacenamiento de gas producido.

Salida de efluentes: Conducto por donde se extraen los residuos luego del proceso de digestión.

La digestión de desechos orgánicos en un biodigestor es un proceso de descomposición anaeróbico, es decir sin la presencia de oxígeno. El proceso de generación de gas metano, también llamado metanogénesis, se debe a una serie de reacciones bioquímicas, donde intervienen gran variedad de microorganismos que oxidan una parte del carbono presente para formar CO_2 y reduce la otra parte para formar metano, ambos compuestos químicamente estables.

Para la generación de gas, principalmente se utilizan aguas residuales residenciales, de pequeña industria, desechos municipales y agrícolas.

CAPÍTULO 3

3. DISEÑO DEL SISTEMA DE TELECONTROL DEL BIODIGESTOR

El diseño ha sido realizado tomando en cuenta los componentes principales, presentes en la automatización de procesos, como son: sensores, actuadores, controlador y la interfaz hombre-máquina (HMI “Human Machine Interface”). Los sensores son los que brindan la entrada al sistema, indicando el valor actual de las variables de interés. Estos datos son procesados por el controlador y en base a su programación activará o desactivará los actuadores, que son los dispositivos encargados de generar algún efecto dentro del proceso. La interfaz hombre-máquina, se refiere a las formas en que el usuario final puede interactuar con el sistema.

La comunicación entre el controlador y los dispositivos HMI, se basa en diversos protocolos de comunicación. Por lo tanto estos protocolos forman parte de la solución presentada, teniendo así un diagrama como el mostrado en la figura 3.1.

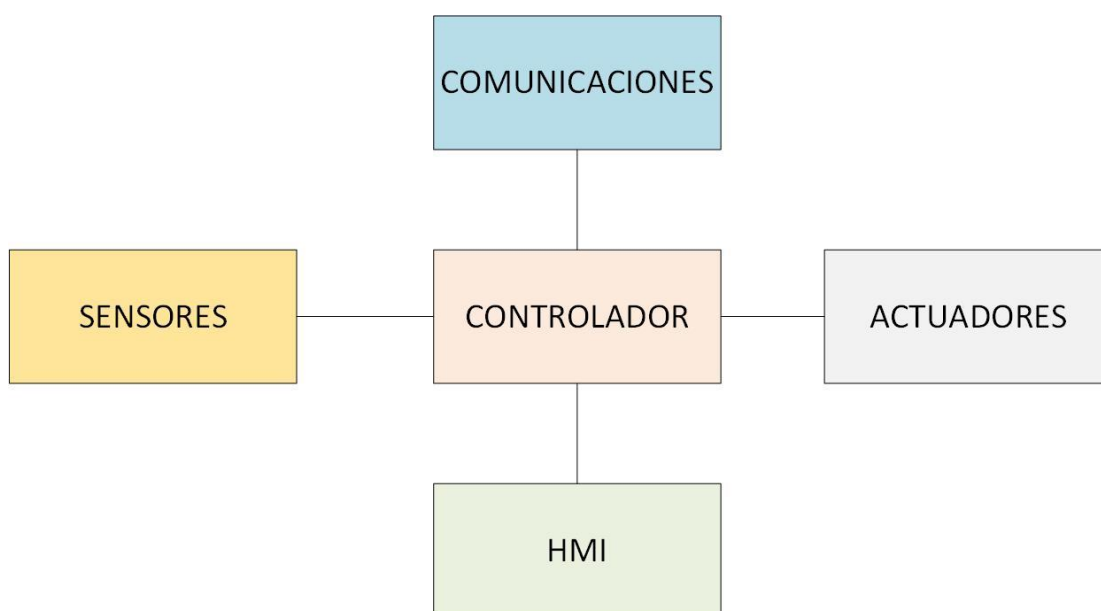


Figura 3.1 Diagrama de bloques del sistema planteado

3.1. CONTROLADOR

3.1.1. PLC

Se utiliza como controlador del proceso el equipo micro850 de “Rockwell Automation”, una empresa dedicada a brindar soluciones de automatización industrial, con sede principal en Milwaukee, Wisconsin en Estados Unidos. Dentro de su gama de productos resaltan 2 marcas preponderantes como son:

“Allen-Bradley” y “Rockwell Software”. Actualmente es considerada una de las empresas líderes en soluciones industriales.

Entre sus productos se incluyen protección de circuitos, variadores, motores, controladores programables y demás elementos de “hardware” necesarios para realizar un trabajo de automatización de un proceso industrial junto con soluciones de “software” como herramientas de programación de PLCs, HMI, SCADA, herramientas de simulación, entre otros.

La familia de PLCs Micro 800 es un conjunto de sistemas de control de bajo costo, que puede incrementar sus capacidades mediante la adquisición de módulos adicionales. La intención de esta gama de productos de “Allen-Bradley” es brindar una solución que se ajuste a las necesidades de sus clientes, sin tener que pagar por características que no son necesarias.

Los controladores de esta familia pueden tener mayores capacidades mediante módulos de expansión o enchufables. Los módulos de expansión aumentan el tamaño del equipo, y maximizan en un número mayor las entradas y salidas del PLC. Mientras los módulos enchufables, a pesar de crecer en menor cantidad las entradas y salidas que los módulos de expansión, poseen la ventaja de formar parte del equipo sin acrecentar su tamaño.

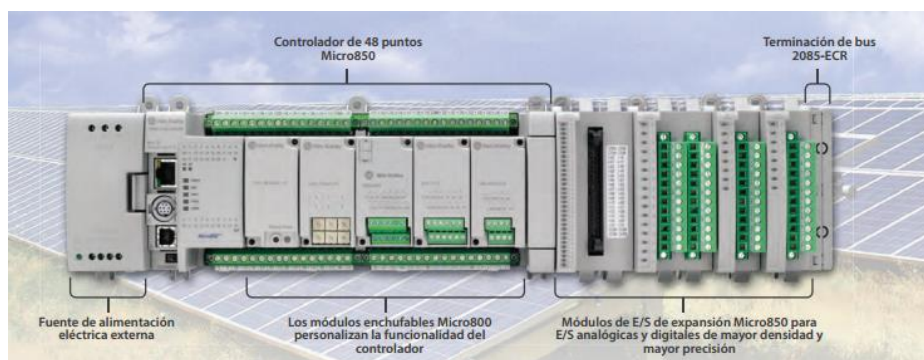


Figura 3.2 PLC Allen Bradley con todos sus módulos integrados.

Cada uno de los controladores programables tiene características que lo diferencian de otro, como suele ser: el número de entradas y salidas disponibles, el número de módulos de expansión que se pueden colocar, tarjeta de red integrada y otras diversas funciones. El equipo que se utilizó para el desarrollo de este trabajo es el micro850, elegido principalmente porque cuenta con una interfaz Ethernet para comunicación y posee 10 salidas digitales y 14 entradas digitales, que son suficientes para esta implementación.



Figura 3.3 PLC Allen Bradley Micro850.

“Connected Components Workbench (CCW)” se trata de un programa de la línea “Allen-Bradley” que permite la configuración de diversas soluciones, entre las que incluyen: la familia de controladores micro800, pantallas HMI “Panel View”, variadores, entre otros elementos de la marca “Allen-Bradley”, que los presentan como una solución muy robusta y completa al momento de una implementación, para tener un solo medio de configuración de todos los equipos integrantes. El programa es un desarrollo sobre Microsoft Visual Studio añadiendo los drivers de Rockwell Automation. La descarga del programa es gratuita, mediante un registro previo en la página de Rockwell.

Brinda al usuario la facilidad de programación con tres métodos: diagrama de bloques funcionales, lenguaje de escalera y texto estructurado. Su interfaz permite al usuario la facilidad de realizar simulación del proceso y efectuar pruebas con el controlador, previo a una implementación, dado que simula la activación tanto de las entradas como de las salidas.

Para obtener los resultados de la medición del PH, temperatura y presión en el controlador, es necesario contar con entradas analógicas. Tal como viene de fábrica el micro850 no posee esta característica, para lo cual se añade el módulo enchufable “2080 IF4” de “Allen-Bradley”, cuya función es brindar cuatro entradas analógicas. Este módulo es capaz de sensar voltaje de 0 a 10

voltios, y corriente entre 0 y 20 mA, cada entrada es independiente y capaz de realizar la medición por uno de los métodos, voltaje o corriente.

Para el envío de alertas por mensajes de textos, resultó necesario añadir un segundo módulo adaptable al controlador, pero a diferencia del módulo de entradas analógicas, este no pertenece a “Allen-Bradley”, es desarrollado y distribuido por la empresa “ProSoft Technology” que se especializa en soluciones de comunicación para controladores de automatización, y se enfocan en brindar soluciones compatibles con la marca “Rockwell Automation” y “Schneider Electric”.

Al no ser un producto de “Allen-Bradley”, no es reconocido este módulo dentro del programa “Connected Components Workbench”, pero con el fin de integrarse completamente a la solución, la empresa “ProSoft Technology” ha desarrollado sus propias funciones para el programa y tiene una buena explicación del uso del mismo. Posee un conector SMA para cable coaxial en su parte frontal, lo que permite colocar la antena fuera del panel para obtener una mejor recepción de la señal.

3.1.2. DISEÑO DE AUTOMATIZACIÓN

Con el fin de controlar el proceso que realiza un biodigestor, se diseñó un mecanismo adaptable a cualquier tipo de biodigestor, en el que se reduzca la necesidad de la interacción o participación humana durante el proceso.

Esta propuesta consta de tres etapas bien marcadas, para el fácil entendimiento de cada una de ellas y poder realizar una búsqueda de fallos por separado. Para demostrar el funcionamiento del diseño propuesto, se realizó una maqueta en la cual se evidencian cada una de las etapas que serán mencionadas. Las fases del diseño son: Sistema de Tanques, Etapa de carga y Etapa de digestión.

3.1.2.1. SISTEMA DE TANQUES

El sistema de tanques es el mecanismo donde inicia el proceso de digestión, en este se propone colocar cuatro tanques para almacenar los materiales necesarios en el proceso. Estos tanques son para almacenar: agua, desechos, líquido ácido y líquido alcalino. Los tanques se encuentran a una altura superior a la del biodigestor, ya que este normalmente está situado debajo de la tierra, esto facilita el paso de los materiales en cada tanque sin necesidad de una bomba, sino que se utiliza la gravedad y se controla el paso mediante electroválvulas.

Las electroválvulas son controladas desde el PLC y abiertas en el momento en que se necesita el material de cada uno de los tanques, proceso que se realiza en la etapa de carga.

Del proceso, esta etapa es la que requiere la participación de una persona a cargo del biodigestor, dado que el llenado de los tanques no puede ser automatizado. El único tanque que se puede optimizar del proceso es el tanque de agua, en el que se propone tener una bomba de agua para el llenado del mismo, trabajando en conjunto con una boya para mantener el tanque de agua siempre en un nivel alto.

Los desechos deberán ser recogidos de los establos donde se almacenan los animales para ser depositados en el tanque correspondiente, dentro del tanque de desechos se coloca un sensor de nivel bajo, que permite enviar una alerta por mensaje de texto al administrador de la necesidad de llenar el tanque.

Los tanques correspondientes a líquidos ácidos y alcalinos, poseen el mismo control que el tanque de desechos, con niveles bajos y enviando alerta mediante mensajes de texto en caso de requerir del líquido y que él mismo esté por debajo del nivel mínimo. Para aumentar el nivel de concentración de pH se utiliza un líquido alcalino, en este caso se trata de carbonato de sodio o

también conocido como ceniza de soda, que es lo que normalmente se utiliza en el tratamiento de piscinas. No se trata de un líquido como tal, por lo que antes de ser insertado en el tanque, debe ser disuelto en agua.

Para disminuir la concentración de pH, se debe almacenar en el tanque de ácido cualquier tipo de ácido, para esta implementación se recomienda emplear “Ácido trioxonítrico” o también conocido como “Trioxinato de hidrógeno”.

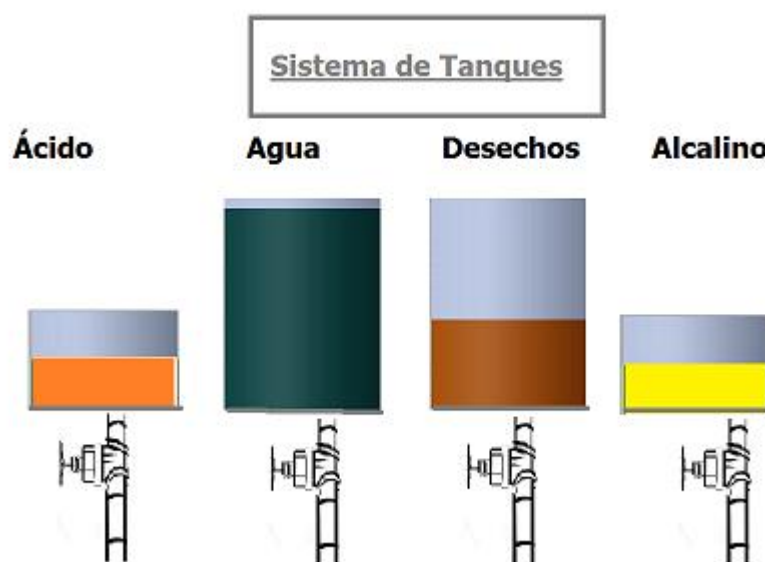


Figura 3.4 Sistema de tanques.

3.1.2.2. ETAPA DE CARGA

La segunda etapa en la solución es la automatización del proceso, para ello solo es necesario recibir una señal de marcha de parte de una persona a cargo

del biodigestor, que puede llegar mediante la pantalla del sistema ubicada en el sitio o de forma remota, mediante un mensaje de texto o botón de la página web.

Normalmente para comenzar el proceso de biodigestión, la persona a cargo debe realizar la mezcla de desechos orgánicos con agua en proporciones iguales, evitando o separando aquellos desechos que sean ácidos como por ejemplo: las cascaras de limón o naranja. Esta etapa elimina la necesidad de que este proceso sea manual, sino que se asegura de que la mezcla sea homogénea y la concentración de pH se mantenga en un nivel óptimo.

El elemento principal de la etapa de carga es el contenedor de la mezcla, que es un reservorio en el cual se almacenan tanto desechos, como agua para su posterior mezcla. Este reservorio es una etapa previa al ingreso de la mezcla al biodigestor, y su capacidad no tiene una relevancia importante, dado que la misma se realiza en este, las veces que sean necesarias para poder llenar el biodigestor.

El reservorio contiene tres sensores de nivel, el primero de ellos sirve para indicar cuando el contenedor esté vacío, significando esto que se ha realizado una nueva cargada de mezcla al biodigestor. El segundo nivel que se encuentra en el contenedor, indica la cantidad máxima de desechos que se

permitirá por cada cargada. Mientras que la última etapa, indica el nivel máximo de la mezcla, es decir, indica la cantidad máxima de agua que se añada a los desechos previamente agregados. Dado que la mezcla debe ser en iguales proporciones, la distancia entre el tercer nivel (nivel de agua) y el final del reservorio debe ser el doble, que la distancia entre el segundo nivel (nivel de desechos) respecto del final del reservorio.

Mediante el controlador micro850, se envía la orden de abrir la válvula del tanque de desechos, la cual se cierra automáticamente una vez que los desechos alcanzan el segundo nivel. Una vez cerrada la válvula de desechos, se abre la válvula del tanque de agua, hasta llegar al tercer nivel en donde esta se cierra.

En esta etapa, también se emplea un agitador, que se trata de un motor capaz de hacer girar unas aspas, para de esta forma mezclar el agua con los desechos. Este agitador es accionado desde el controlador una vez que se ha llenado el reservorio de tanque para cada cargada, y se mantiene prendido hasta homogenizar la mezcla, tiempo que sí dependerá del tamaño del reservorio; para la maqueta realizada el tiempo es de 1 minuto, con lo cual el material queda mezclado.

En esta etapa se realiza la validación del nivel de pH presente en la mezcla, para optimizar la biodigestión al asegurar que la mezcla se mantiene neutral (entre 6 y 8). Para ello, en el reservorio se coloca un sensor de pH, que sirve de entrada para el controlador, y el valor medido funciona para determinar el tanque que se debe abrir para mantener la mezcla neutral. En caso de que el valor de pH sea menor a seis, se procede a abrir la válvula del tanque de alcalino, y a la vez se activa el agitador para que el líquido alcalino pueda mezclarse, la válvula se detiene una vez que se tiene la mezcla neutral. El proceso es similar en caso de que el pH sea mayor a ocho, con la diferencia de que la válvula que se abre ahora, es la del tanque de ácido, activándose el agitador, de igual forma se detiene el proceso cuando la mezcla sea neutral.

Al tener una mezcla neutral, la misma es enviada al biodigestor, abriéndose la válvula colocada en el reservorio, y por efecto de la gravedad la mezcla ingresa al biodigestor. Se cierra la válvula cuando el nivel de mezcla está por debajo del primer nivel del reservorio, lo que indica que el reservorio se encuentra vacío, y se puede realizar una siguiente mezcla.

El proceso de carga se repite las veces que sean necesarias hasta tener el biodigestor lleno o en el nivel deseado. Esto dependerá del administrador del biodigestor, de la cantidad de material que desea utilizar en cada proceso de biodigestión, siendo el sistema capaz de adaptarse a estos requerimientos.

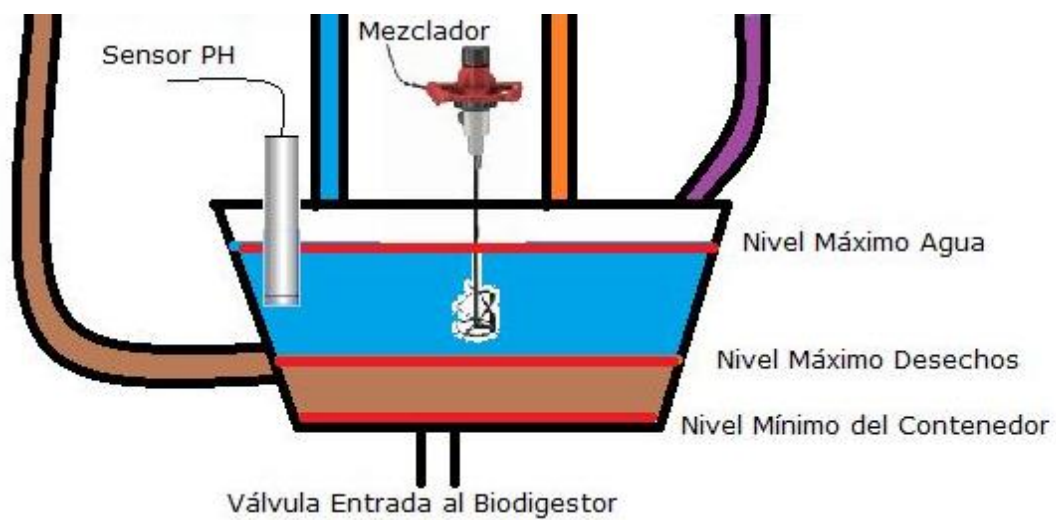


Figura 3.5 Ilustración del tanque de carga.

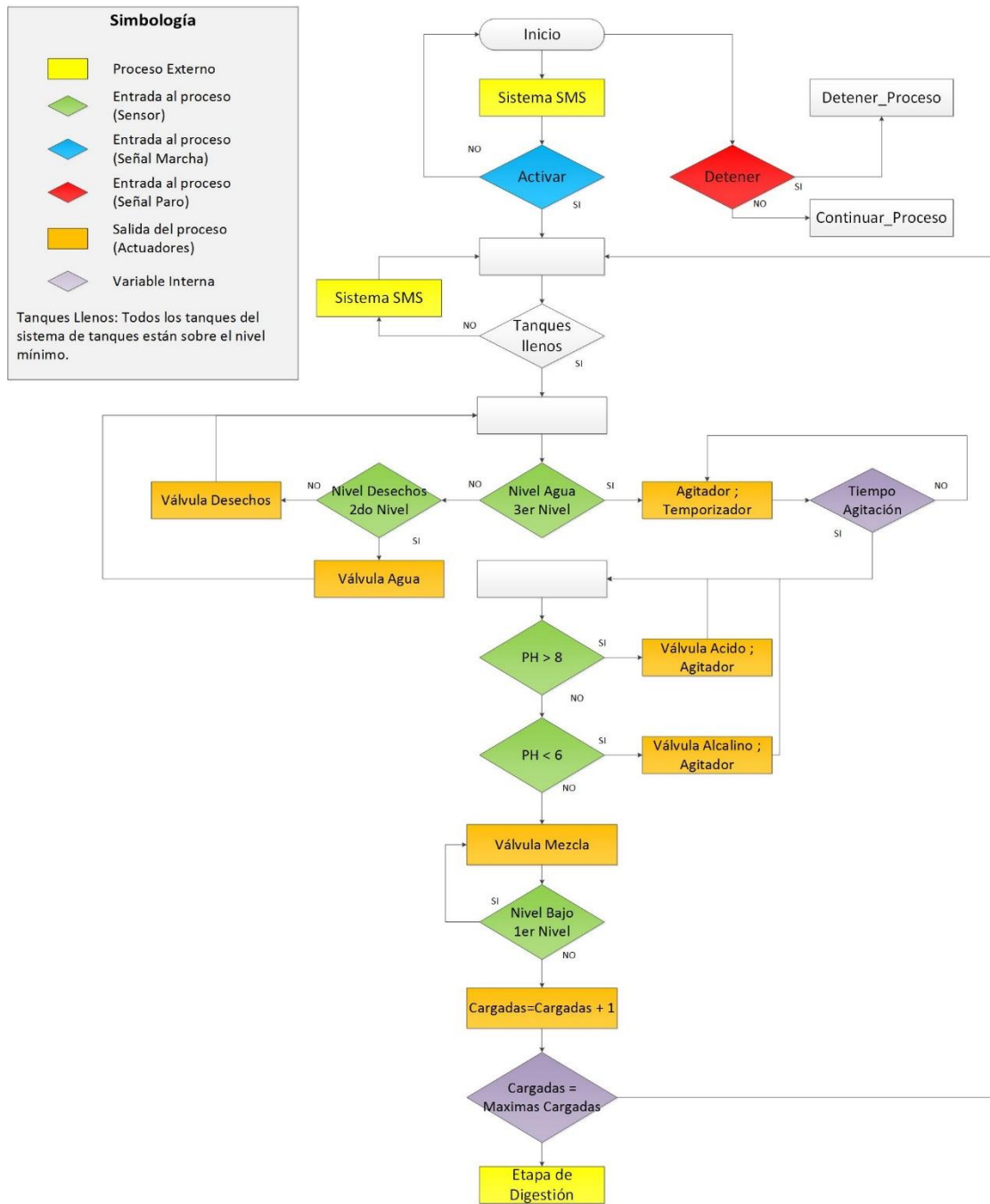


Figura 3.6 Diagrama de flujo de datos de la etapa de carga

3.1.2.3. ETAPA DE DIGESTIÓN

La etapa de digestión es donde se encuentra el biodigestor, ya con la mezcla homogénea dispuesta a ser descompuesta por las partículas anaeróbicas dentro del biodigestor. Dentro de esta etapa las variables sobre las que se van a tener control son: la temperatura y la presión.

La temperatura ambiente a la que es sometida el biodigestor se relaciona directamente con el tiempo de retención de la mezcla dentro del biodigestor. En temperaturas de 30°C se estima que se debe retener la mezcla por diez días, mientras que si se encuentra en lugares fríos a 10°C la mezcla deberá estar retenida durante al menos 55 días. Se toman en cuenta estas dos medidas de temperatura, debido a que son de las más representativas para el país, que en la región costa se tienen temperaturas alrededor de 30°C, mientras en ciertas zonas de la sierra se manejan temperaturas cercanas a los 10°C. El administrador es capaz de establecer valor, para lo cual el sistema le notifica la temperatura a la que está el biodigestor para tomar una correcta decisión.

Durante el tiempo de digestión, se alertará al administrador en caso de producirse cambios bruscos de temperatura, para que lo tenga como referencia sobre la cantidad de gas generado. Estas alertas son enviadas mediante mensajes a una lista de números de teléfonos.

La presión del gas generada dentro de la bolsa del biodigestor podría dañar la bolsa en caso de no ser controlada. Para ello se ubica un sensor de presión dentro del biodigestor, que permite conocer la presión que ejerce el gas. La presión que soporta la bolsa va a depender del material con el que se haya elaborado la misma.

En caso de que el sensor de presión se encuentre dañado, este marcará un valor de 0 psi o valores negativos. Este es un caso considerado de emergencia, dado que se habría perdido el control de la presión y se procede a abrir la válvula de gas y a la vez se prende el quemador para que este gas no llegue a su destino final, sino que sea desechado.

Estos controles de temperatura y presión se llevan a cabo durante el tiempo de digestión, una vez que este tiempo ha concluido el proceso habrá finalizado. Se abre la válvula de gas para que este salga del biodigestor hacia su destino, que dependerá del uso que se le dé al biodigestor o simplemente pueden almacenar el gas para su futuro uso. Del resultado final también se genera un fertilizante conocido como biosol, que es muy utilizado en las plantaciones, y para poderlo extraer se abre la válvula de biosol ubicada en la parte inferior del biodigestor. Se notifica al administrador que el proceso ha finalizado y se queda en espera a la orden de un nuevo inicio.

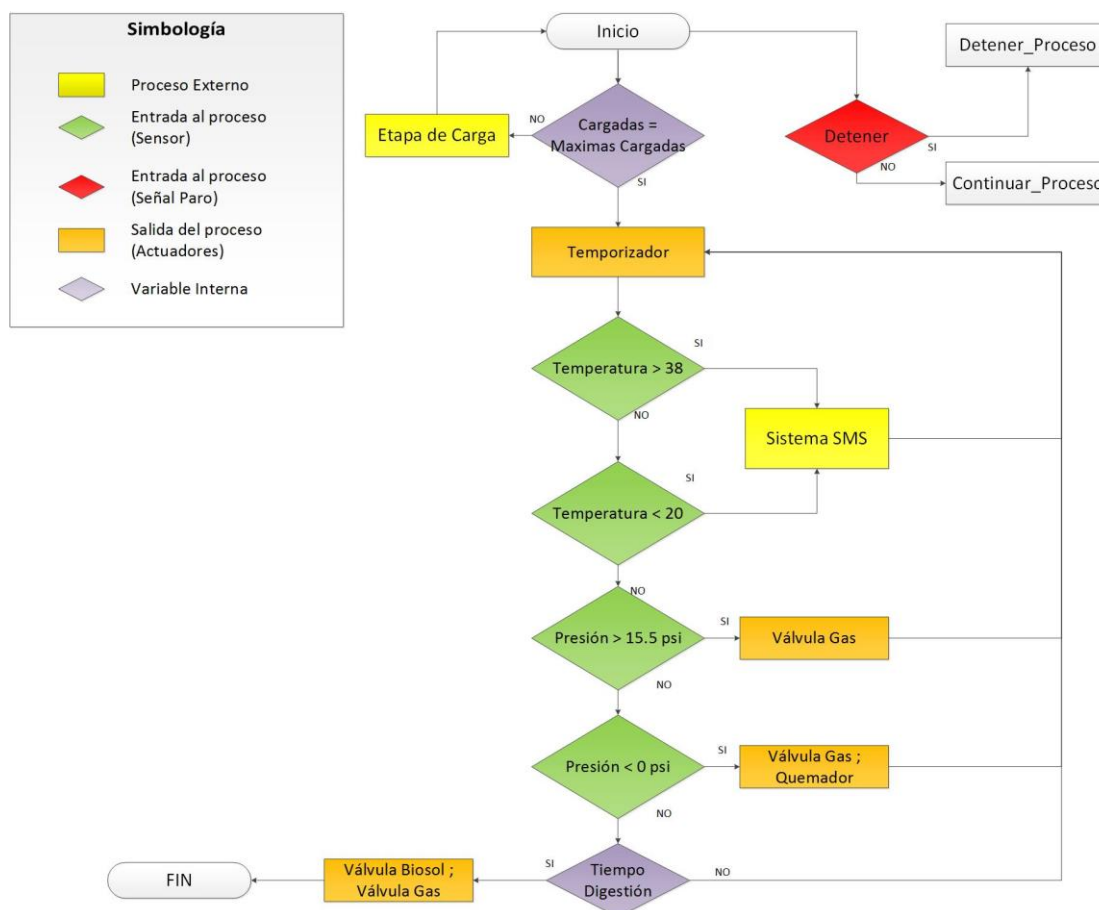


Figura 3.7 Diagrama de flujo de datos de la etapa de digestión.

3.2. SENSORES

Un sensor es un dispositivo capaz de transformar magnitudes físicas o químicas del entorno, también llamadas variables de instrumentación, en magnitudes eléctricas. Estas variables de instrumentación pueden ser por ejemplo: temperatura, presión, pH, fuerza, tensión, etc.

Un sensor siempre está en contacto con la variable de instrumentación y adapta esta señal que mide para enviarla a otro dispositivo que pueda interpretarla.

El diseño planteado utiliza diferentes sensores para medir variables como: presión, temperatura, pH y nivel de líquido.

3.2.1. SENSOR DE NIVEL

Para realizar la medición de nivel de relleno de líquidos, lodos o granulados, es necesario la utilización de un sensor de nivel. Existe una amplia gama de dispositivos que permiten realizar esta función dependiendo de la aplicación donde se emplee: agua, cemento, minería, química, etc.

Su funcionamiento se basa en la detección de la altura del líquido, para comprobar si ha alcanzado o no un punto establecido. Este detector se clasifica en:

- Sonda
- Flotador
- Presión diferencial
- Por burbujeo
- Radioactivo
- Capacitivo

- Ultrasónico
- Conductivo o resistivo

El sistema de biodigestión utiliza tanques con diversos tipos de líquidos como: agua, desechos orgánicos, ácidos y alcalinos. Es necesario utilizar un tipo de sensor que determine el nivel. Para esto se debe emplear sensores ultrasónicos, ya que al no estar en contacto con la solución, no influye la composición del mismo al realizar la medición.

3.2.2. SENSOR DE TEMPERATURA

Elemento que convierte la intensidad de calor en variación eléctrica, ya sea de voltaje, corriente o resistencia. Este sensor ofrece una salida continua en función de la temperatura del sistema, permitiendo monitorearla. Existen tres tipos de sensores de temperatura:

- Termistores.
- RTD.
- Termopares.

Un termistor funciona en base a la variación de resistencia de un semiconductor con respecto a la temperatura que es sometido. El problema de este tipo de sensores radica en la no linealidad de su comportamiento, por lo que es complicada su calibración.

Un RTD se basa en la variación de la resistencia de un conductor con respecto a la temperatura que es sometido. Normalmente se emplean metales como: platino, cobre, molibdeno y níquel. De estos metales, el más utilizado es el platino por tener mayor rango de temperatura, mejor linealidad y rapidez.

Un termopar o termocupla basa su funcionamiento en el efecto termoeléctrico, que permite transformar directamente el calor en electricidad o viceversa.

Para una óptima generación de gas, es necesario mantener el biodigestor a una temperatura entre 20° C y 40° C, por esta razón se debería utilizar una termocupla para monitorear la temperatura.

3.2.3. SENSOR DE PH

Para poder medir la alcalinidad o acidez de una disolución, es necesario utilizar un pH-metro. Este pH-metro tiene dos partes: una punta de prueba que está en contacto con la solución a medir y un acoplador que amplifica la señal emitida por la punta de prueba antes de enviarla al controlador.

La punta de prueba utiliza un método electroquímico para medir el pH de una disolución. Esto se basa en medir el potencial eléctrico que se desarrolla en una membrana de vidrio, la cual separa dos soluciones con diferente

concentración de protones. Este potencial es conducido hacia un amplificador mediante un alambre, normalmente Ag/AgCl.

El nivel de acidez o alcalinidad es muy importante en el proceso de digestión, ya que si la mezcla desechos-agua tiene un nivel de pH menor a 8, la disolución se vuelve alcalina y tiende a pudrirse. Por el contrario si el nivel de pH es mayor a 6, la disolución se vuelve ácida y el gas producido se vuelve muy corrosivo e inutilizable.

3.2.4. SENSOR DE PRESIÓN

Los sensores de presión, son muy comunes en todo proceso industrial o sistema. Su objetivo es transformar una fuerza por unidad de superficie, en un voltaje equivalente a esa presión ejercida. Esta fuerza puede ser aplicada a un punto en una superficie o distribuirse sobre esta.

Las mediciones son muy amplias, desde valores muy bajos hasta cientos de toneladas por unidad de área. Para realizar esta medición se utilizan sensores que tienen elementos sensibles a la presión y que produce una señal eléctrica que cambia a medida que varía la presión.

En el caso del biodigestor se requiere utilizar un sensor para medir la presión de gas generado y así poder tomar decisiones de acuerdo a los requerimientos.

3.3. ACTUADORES

Los actuadores son componentes de salida que se encargan de transformar las señales emitidas por el controlador en esfuerzos físicos, necesarios para generar el movimiento requerido por múltiples automatismos. En función del tipo de energía que se requiere para generar el movimiento se clasifican en:

- Hidráulicos.
- Eléctricos.
- Neumáticos.

Estos actuadores generalmente deben estar aislados eléctricamente del sistema de control, por lo que se utilizan elementos de conmutación como octoacopladores o relés. En este caso se ha utilizado relés de 24v para manejar actuadores de diferente voltaje y aislarlos del circuito de control.

3.3.1. ELECTROVÁLVULA

Es una válvula electromecánica que controla el paso de un fluido a través de una tubería. Esta electroválvula consta principalmente de dos partes: una bobina solenoide y la válvula. Al pasar una corriente a través del solenoide convierte energía eléctrica, mediante magnetismo, en energía mecánica para abrir o cerrar la válvula.

3.3.2. MOTOR ELÉCTRICO

Un motor eléctrico transforma la energía eléctrica en energía mecánica, empleado en diversas áreas como: electrodomésticos, equipos electrónicos, máquinas industriales, etc.

Un motor de corriente continua permite incorporar una reducción mecánica y tener un giro estable, pero su principal inconveniente es que debido a su inercia, su movimiento no es preciso.

En la etapa de carga del sistema del biodigestor se debe utilizar un motor de corriente continua para realizar la mezcla de agua y desechos orgánicos, y diluir las porciones de desechos grandes con la finalidad de tener una solución homogénea.

3.4. HMI

Durante la planificación del proyecto se ha decidido contar con tres formas de interacción entre el humano y el sistema. El primer método consiste en una pantalla táctil ubicada cerca al biodigestor, donde pueden observar el estado del proceso e interactuar con él mismo. El segundo método fue pensado para usuarios remotos, y así poder tener un verdadero telecontrol del proceso. Este método se basa en una aplicación web, a la que el usuario puede acceder

desde cualquier dispositivo que posea un navegador web, como por ejemplo: los dispositivos móviles “Smartphones” y computadores. La última forma para interactuar con el usuario es mediante mensajes de texto, y aunque es principalmente utilizado para el envío de alertas, también es factible que el usuario envíe un mensaje al controlador para consultar valores específicos o para activar señales del proceso.

3.4.1. PANTALLA

La pantalla, al igual que la mayoría de los equipos utilizados, es un producto de la línea de “Allen-Bradley”. Su nombre es “PanelView Component C600” y es una pantalla táctil de 5.7 pulgadas a color TFT. La finalidad de esta pantalla es tener una forma de interactuar desde el sitio con una herramienta que forma parte de la solución.

Las pantallas a mostrar son configuradas desde el mismo programa CCW. Para mantener la división de las etapas se ha creado una vista para cada una de las etapas mencionadas en el diseño del controlador. En cada vista se muestran los elementos que participan de la misma con su estado actual. En la pantalla principal, se encuentran los botones de marcha “Start” y paro del proceso “Stop”.

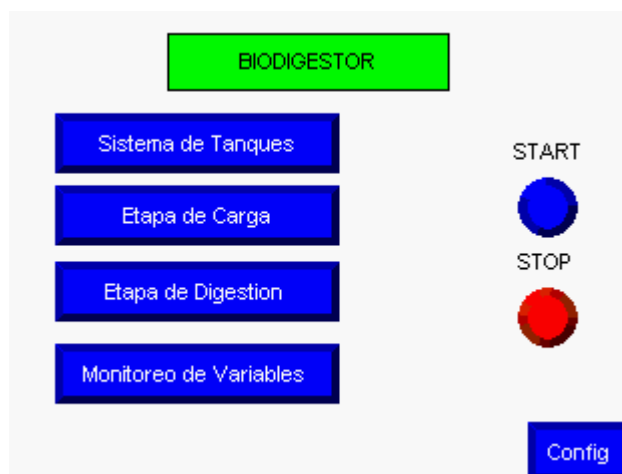


Figura 3.8 Pantalla local de inicio del sistema.

Dentro de la vista del sistema de tanques se muestran los cuatro tanques reservorios de la solución. Cada uno de los tanques posee dos estados, lleno o vacío, con su respectiva imagen que identifica el nivel de material en el tanque. El tanque cambia su imagen dependiendo de si se ha activado o no el sensor de nivel correspondiente, esto permite notar cuando falta material en algún tanque. Adicionalmente en esta vista existen tuberías con sus electroválvulas. En ellas se refleja el paso del material en caso de que la válvula se encuentre abierta, caso contrario la tubería se nota vacía.

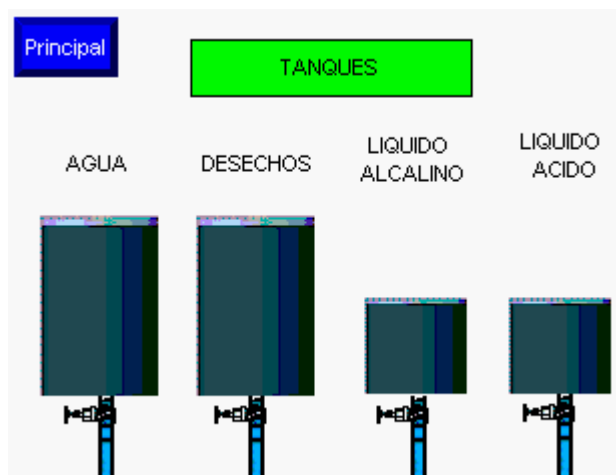


Figura 3.9 Pantalla local del sistema de tanques.

La vista de la etapa de carga cuenta con el contenedor de la mezcla, el cual cambia su estado acorde a los sensores de nivel que se encuentren activados teniendo tres niveles posibles: vacío, nivel máximo de desechos y nivel máximo de agua. Se tiene la tubería con la válvula que permitirá el ingreso de la mezcla al biodigestor. Para el agitador, se ha colocado un ventilador, que permanece estático mientras el agitador esté apagado y realiza giros en cuanto el agitador se activa.

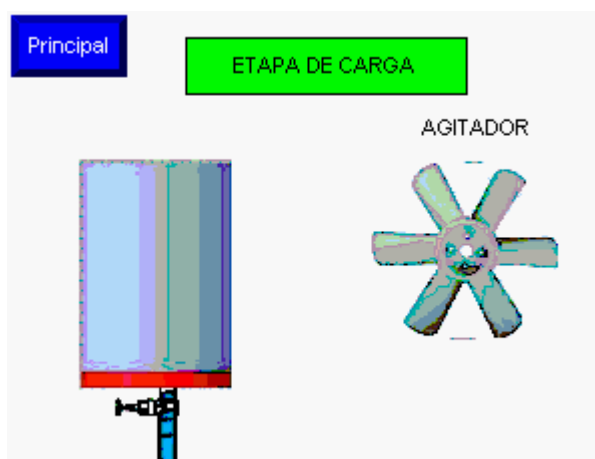


Figura 3.10 Pantalla local de la etapa de carga

En la etapa de digestión se tiene el biodigestor, cuyo nivel va variando acorde a las cargas de mezclas realizadas. También se prende la llama de un mechero cuando se activa el quemador, en caso de algún problema con el sensor de presión. Además presenta el tiempo de digestión, que permite al usuario saber el tiempo que lleva corriendo el proceso. Por último está la válvula de biosol, que indica cuándo el biodigestor está permitiendo la salida de este fertilizante.

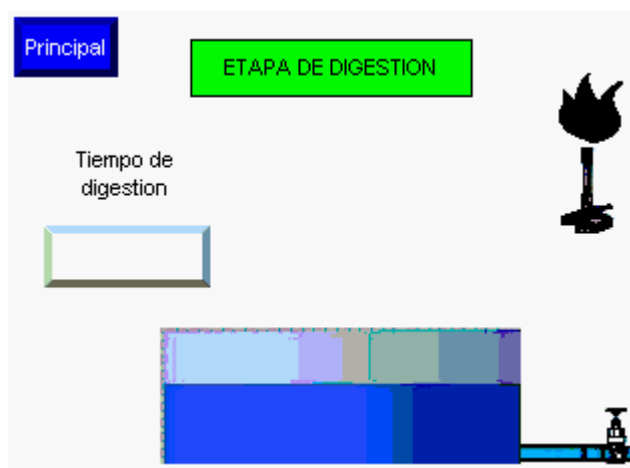


Figura 3.11 Pantalla local de la etapa de digestión.

La opción de monitoreo de las variables muestra una vista donde indica el valor de las tres variables del proceso. Se encuentra también un botón “Gráfico” que permite observar el valor de las variables mediante un gráfico.

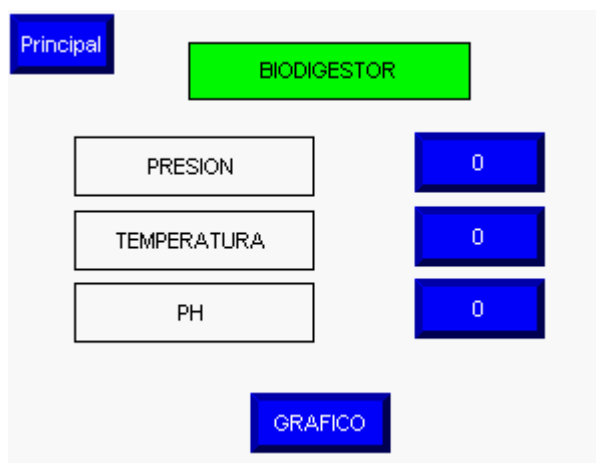


Figura 3.12 Pantalla local de monitoreo de variables 1.



Figura 3.13 Pantalla local de monitoreo de variables 2.

3.4.2. APLICACIÓN WEB

Para realizar el control del sistema de forma remota, se necesita de una aplicación para la plataforma del usuario. La herramienta es pensada para la amplia gama de plataformas que se conocen hoy en día como Linux, Windows y MacOS. Si a esto se suma la necesidad de equipos móviles como “Smartphones” o “Tablets”, plataformas en la que se podría tener el control remoto del proceso. Es por esto que se desarrolla un aplicativo web, que solo requiere de un navegador para poder visualizarlo.

Las vistas creadas en el aplicativo web fueron diseñadas tomando como base las vistas de la pantalla táctil ubicada en el sitio del biodigestor, para mantener una uniformidad entre las interfaces de interacción del usuario. Se manejan

usuarios y las vistas a las que cada uno de ellos puede acceder. El usuario que se designe como tipo “Administrador”, será capaz de: crear usuarios, editarlos, cambiar los permisos y eliminarlos.

La página web fue creada utilizando html principalmente y su diseño realizado con css. Fue diseñada usada “responsive design” lo que permite que las vistas se adapten al tamaño del dispositivo en el que se observa la aplicación.

El acceso a cada una de las vistas dependerá de los permisos que posea el usuario que ingresa al sistema. Se cuenta con las siguientes vistas para que el usuario interactúe:

- Ingreso al sistema
- Sistema de Tanques
- Etapa de Carga
- Etapa de Digestión
- Monitoreo de Variables
- Manejo de usuarios

La primera vista de ingreso al sistema sirve para poder autenticar al usuario. Solo pueden ingresar al sistema las personas que hayan sido creadas como usuarios previamente.

The screenshot shows a web application interface. At the top, a black navigation bar contains the text: SISTEMA DE TANQUES ETAPA DE CARGA ETAPA DE DIGESTION MONITOREO DE VARIABLES. Below this, a green header bar displays 'Iniciar Sesión'. The main content area is a light green box containing two input fields: 'Usuario (Login):' and 'Contraseña:'. Below these fields is a button labeled 'Iniciar sesión'.

Figura 3.14 Pantalla de inicio de sesión de aplicación web.

Cuando el usuario se ha autenticado se le muestran las vistas a las que tiene acceso. El administrador es el único que posee acceso al manejo de los usuarios y de la opción de poner en marcha el proceso o de detenerlo, en caso de considerarlo necesario.

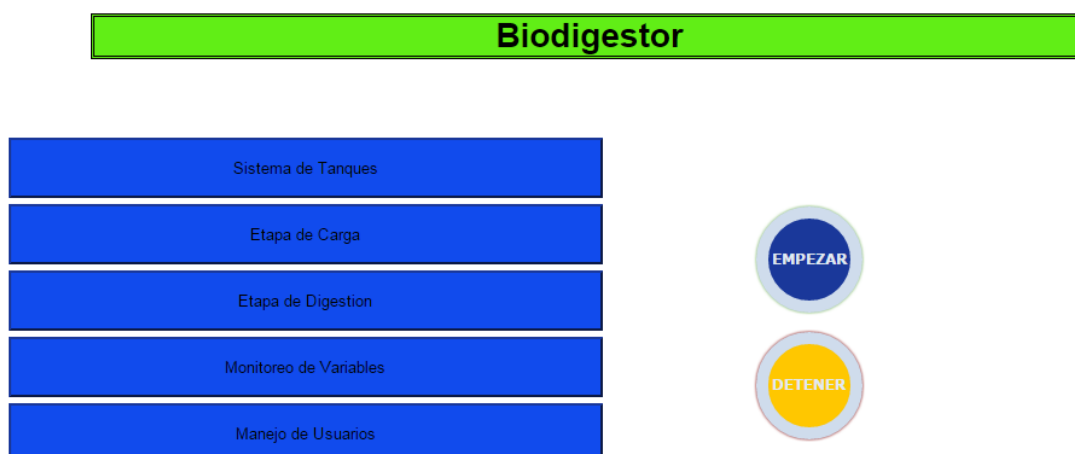


Figura 3.15 Pantalla principal del sistema de aplicación web.

Crear Usuario

Usuario (Login):

Nombre:

Apellido:

Contraseña:

Estado:

Activo Inactivo

Tipo de Usuario:

Administrador Estandar

Elija las vistas a las que tiene acceso:

- Sistema de Tanques
- Etapa de Carga
- Etapa de Digestion
- Monitoreo de Variables

Figura 3.16 Pantalla de creación de usuarios de aplicación web.

Los demás usuarios verán las vistas que el administrador le ha permitido, sin posibilidad de poner en marcha o detener el proceso.

Biodigestor

Sistema de Tanques

Etapa de Digestion

Figura 3.17 Pantalla de usuario con privilegios restringidos de aplicación web.

La vista del sistema de tanques incluye los 4 tanques con sus respectivas válvulas. Además consta con tres botones que solo son visibles para el administrador. Estos le permiten activar o detener el proceso y existe un botón llamado “manual”, que le faculta abrir una válvula independientemente del estado del proceso.

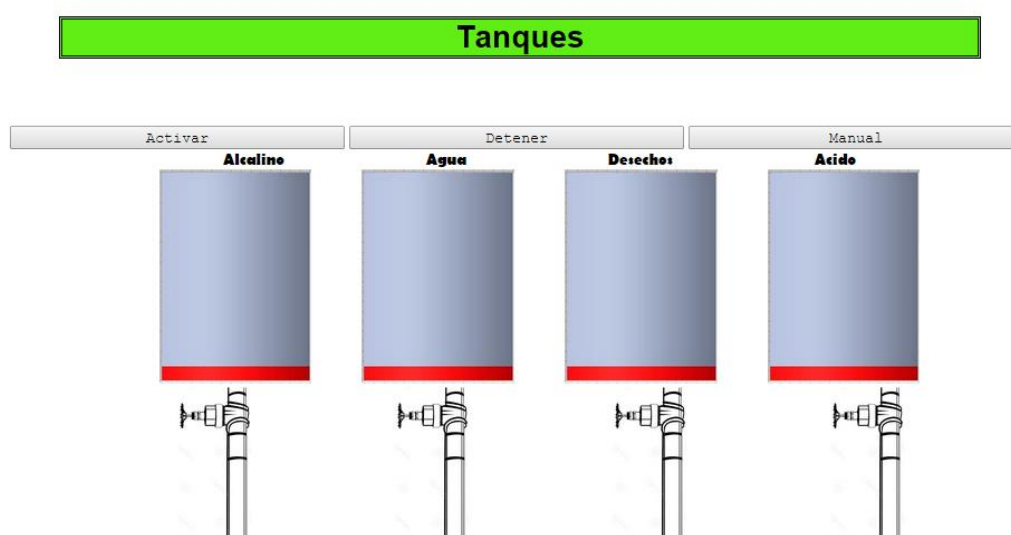


Figura 3.18 Pantalla de sistema de tanques de aplicación web

La vista de la etapa de carga cuenta de igual forma con los tres botones disponibles para el administrador. El tanque contenedor de la mezcla que funciona de igual forma que en la pantalla táctil ubicada en sitio. La electroválvula y el agitador se activa cuando se realiza la mezcla de los materiales. Al presionar el botón “manual” se puede abrir la válvula y activar el agitador con dar clic a las imágenes respectivas.

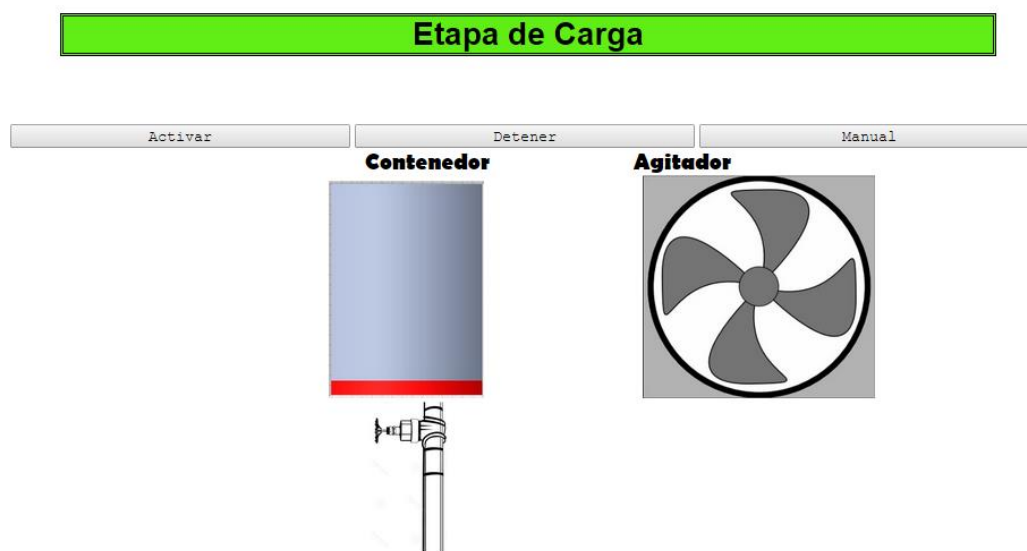


Figura 3.19 Pantalla de etapa de carga de aplicación web.

En la etapa de digestión podemos observar de igual forma los 3 botones para el usuario administrador. Tenemos el tiempo de digestión que ha transcurrido, el quemador cuya llama es vista cuando se activa la señal del quemador, el biodigestor con su válvula de gas en la parte superior, y la válvula de biosol a un costado.

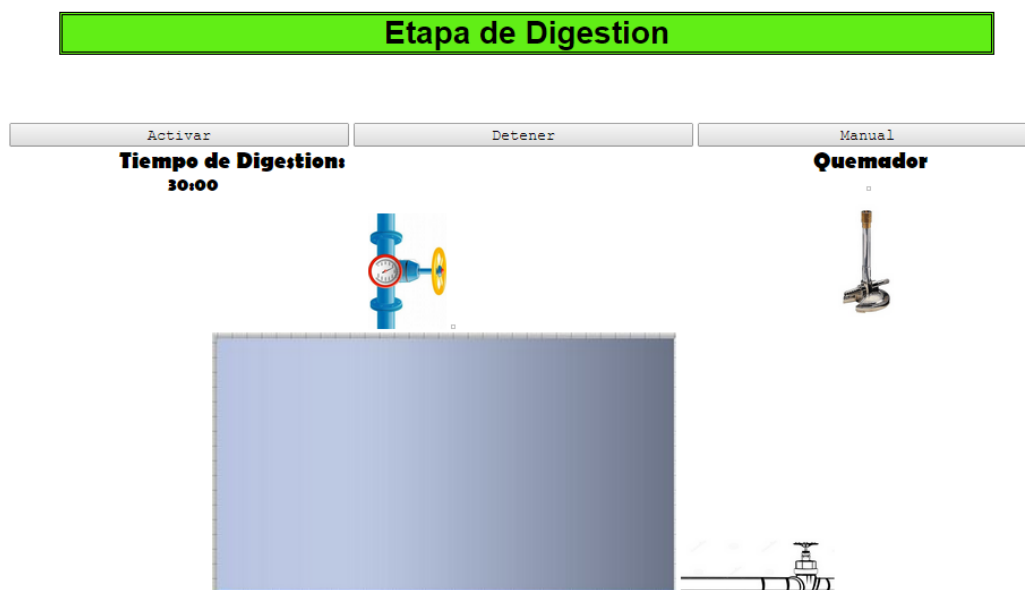


Figura 3.20 Pantalla de etapa de digestión de aplicación web.

La opción de monitoreo de variables muestra los valores de presión, temperatura y PH a los que está sometido el proceso, mostrando una barra que marca el valor mínimo y máximo.

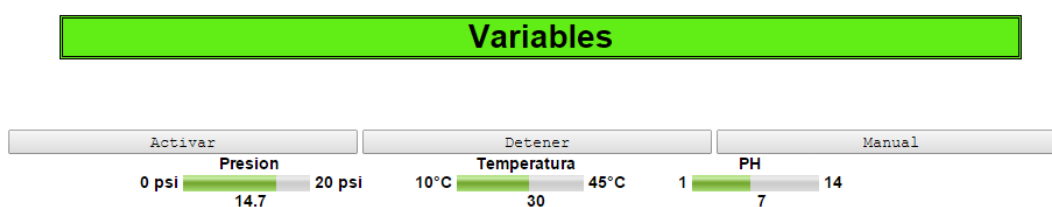


Figura 3.21 Pantalla de monitoreo de variables de aplicación web.

3.4.3. SISTEMA SMS

El sistema de mensajes de texto fue diseñado principalmente para alertas, pero también fue desarrollado de forma que el usuario pueda interactuar con la solución. Para poder realizar esto posible, fue necesaria la utilización de un módulo enchufable para el controlador “micro850”. En este módulo se coloca una tarjeta SIM de cualquier operador de telefonía local, para que pueda enviar y recibir mensajes de texto. El módulo utilizado es un producto de la empresa “Prosoft”, dedicada a la fabricación de dispositivos de comunicación para controladores. Se enfocan mucho en que sus productos sean adaptables a las 2 principales marcas de equipo de automatización como son: “Rockwell Automation” y “Schneider Electric”.

El módulo no es reconocido dentro de la interfaz del “Connected Component Workbench”, pero con la finalidad de que toda la solución sea configurable desde una sola interfaz, “Prosoft” ha diseñado un código que puede ser importado dentro del CCW para poder interactuar con este módulo. Con esto se logra programar el momento de envío del mensaje o qué acciones realizar, en caso de recibir un mensaje de texto.

Las alertas son enviadas a un grupo de destinatarios previamente especificados por el administrador. Entre las alertas que se envían están las siguientes:

- Nivel bajo en tanque de desechos: Indica a los destinatarios la falta de desechos en el tanque de desechos.
- Nivel bajo en tanque de alcalino: Señala que el líquido alcalino se encuentra en nivel inferior al mínimo. Esta alerta se envía solo si es necesario disminuir el pH.
- Nivel bajo en tanque de ácido: Muestra que la cantidad de líquido ácido que se encuentra en el tanque, es inferior al nivel mínimo. Esta alerta se envía solo si es necesario aumentar el pH.
- Mezcla lista: La mezcla de los materiales ha sido satisfactoria teniendo el nivel de pH correcto. Se envía al final de cada carga de la etapa de carga.
- Biodigestor lleno: Indica el momento en que se han realizado todas las cargadas desde el contenedor de la etapa de carga hacia el biodigestor.
- Presión alta: Es enviada en caso de que la presión del gas sea mayor a la presión máxima soportada por el biodigestor.
- Presión baja: En caso de que la presión del biodigestor marque valores negativos se envía esta alerta, dado que indica un comportamiento anormal del sensor.
- Temperatura alta: Es enviada esta alerta si la temperatura dentro del biodigestor es mayor a la de trabajo.
- Temperatura baja: Indica que la temperatura es inferior a la de trabajo del biodigestor.

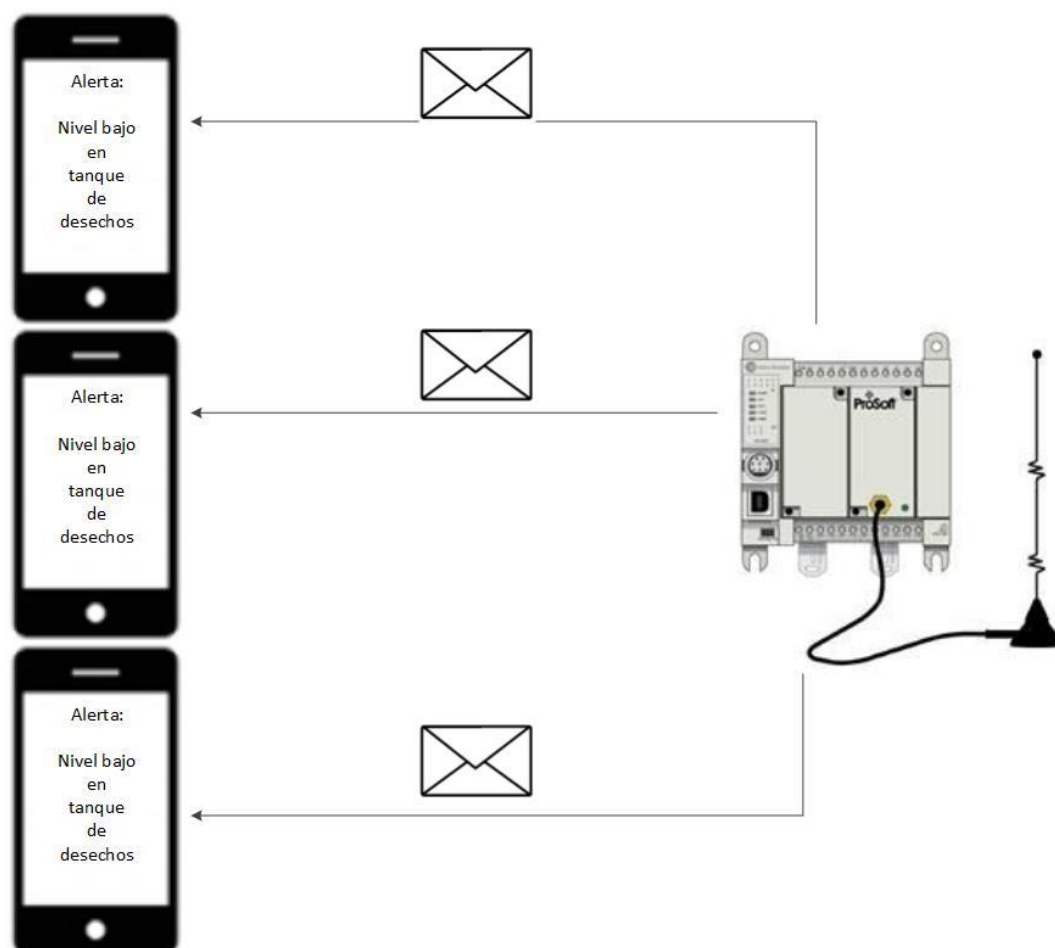


Figura 3.22 Diagrama de envío de alertas por mensajes de texto.

El dispositivo es capaz de responder ante las peticiones específicas del usuario. Para ello es necesario recibir un mensaje de texto con una palabra que esté colocada dentro del código. Para la solución se han definido las siguientes palabras claves:

- EMPEZAR: Al recibir esta palabra se valida el valor actual de la temperatura. Si la temperatura es correcta se inicia el proceso del

biodigestor y se responde confirmando el inicio del mismo. Si la temperatura es alta o baja, se envía la alerta y se queda atento a que el usuario envíe “CONTINUAR”.

- CONTINUAR: Es la señal para iniciar el proceso, a pesar de que los valores de temperatura no son adecuados.
- DETENER: Hace que el proceso se detenga, trabajando con la misma señal que el botón de paro “Stop”.
- REINICIAR: Señal que reactiva el proceso luego de que se ha detenido anteriormente.
- TEMPERATURA: Se le indica la temperatura dentro del biodigestor al usuario que consulta.

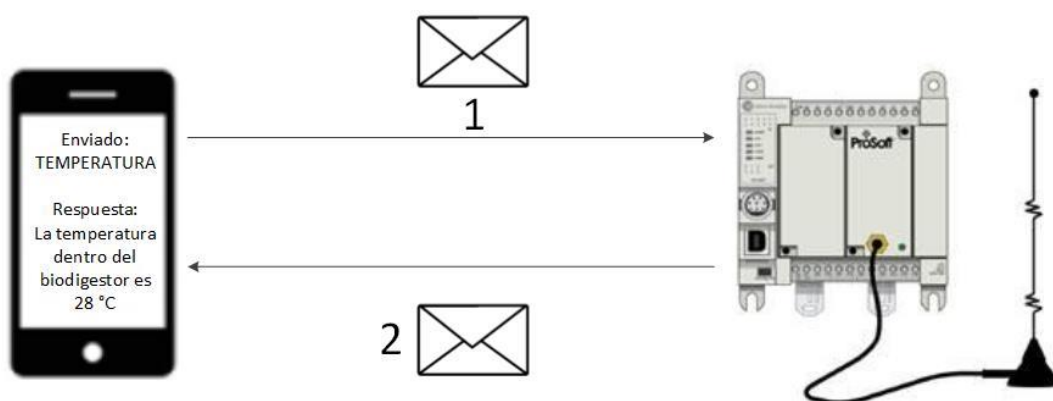


Figura 3.23 Diagrama de envío y recepción de estado de variables por mensajes de texto.

3.5. COMUNICACIONES

En este numeral se tratará sobre los diversos protocolos utilizados en el proyecto para lograr que los elementos interactúen entre sí. Cabe destacar que siempre el medio de transmisión es “Ethernet” entre todos los dispositivos, siendo estos el controlador, la pantalla táctil y el servidor web. Para la comunicación entre ellos se dispone de un conmutador, pero para que el usuario pueda acceder de forma remota se utiliza un enrutador, dentro del que se crea una VPN L2TP/IPsec.

3.5.1. PANTALLA – CONTROLADOR

La comunicación entre la pantalla táctil (PanelView C600) y el controlador (micro850) puede ser tanto serial como “Ethernet”. El protocolo recomendado a utilizar, dado que ambos dispositivos son productos de “Allen-Bradley”, es el “Allen-Bradley CIP”. Es un protocolo que puede utilizar ambos medios de comunicación, pero se ha decidido implementar “Ethernet”, dado que es la tendencia actual.

El protocolo CIP es muy utilizado para comunicación entre dispositivos industriales, de allí que sus siglas en inglés son “Common Industrial Protocol”, en español “Protocolo Industrial Común”. Fue creado como uso principal para

controlar el estado de las entradas y salidas de los dispositivos, así como la información interna para realizar un diagnóstico del sistema que se controla.

CIP es un protocolo de la capa de aplicación que puede ser aplicado sobre diversos tipos de transporte, creándose así las tecnologías “EtherNet/IP”, “ControlNet” y “DeviceNet”. Soluciones que son controladas por la organización ODVA (Open DeviceNet Vendors Association). Los productos de las marcas miembros de esta organización deben pasar por pruebas previo a su certificación, siendo el caso de la mayoría de los productos de “Rockwell Automation” lo que les permite usar estas tecnologías.

En base a esto “Rockwell automation” decidió unir las tres soluciones y crear la arquitectura “NetLinx” que permite la integración de las diversas soluciones que implementan CIP. La arquitectura “NetLinx” ha sido reconocida y recomendada por la ODVA como una solución para la integración de los dispositivos.

“Allen-Bradley CIP” al trabajar sobre comunicación “Ethernet”, emula a “EtherNet/IP”, añadiendo los campos necesarios para que la interacción entre los dispositivos de la marca “Allen-Bradley”, sea totalmente transparente para el usuario. La fácil integración entre los dispositivos es lo que le brinda una ventaja total sobre otros protocolos, pero los equipos también pueden trabajar

con otro protocolo como “Modbus TCP”, en caso de querer realizar una implementación con dispositivos de otra marca.

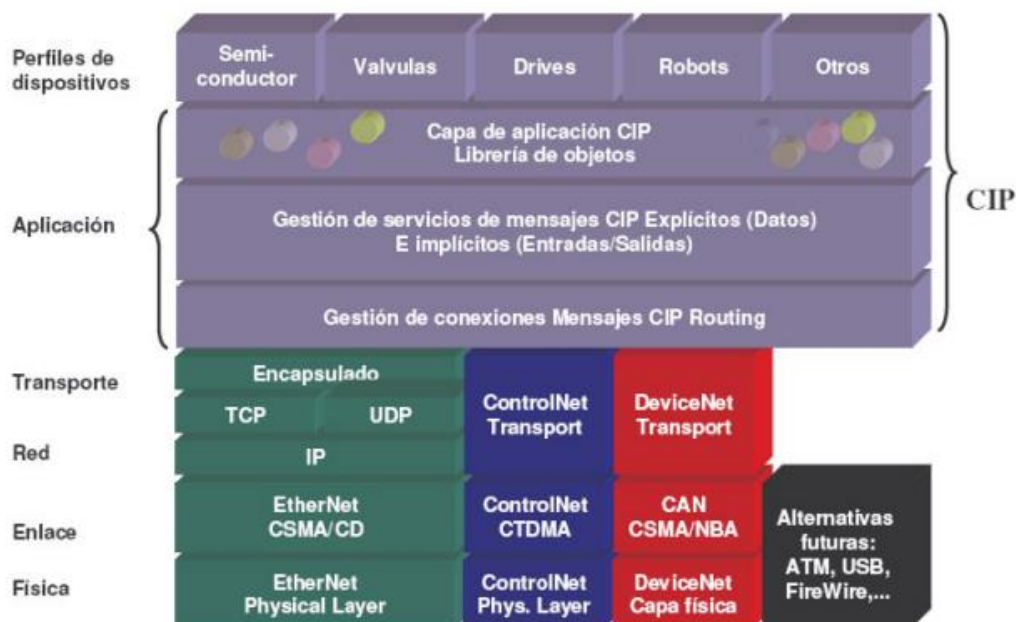


Figura 3.24 Ubicación del protocolo CIP en el modelo OSI.

3.5.2. FRAMEWORK DE APLICACIÓN WEB

La aplicación web ha sido desarrollada mediante el marco de trabajo (Framework) “ASP.NET”, y utilizando el modelo de programación “MVC” (Modelo-Vista-Controlador). El lenguaje de programación utilizado para el desarrollo fue “C#”.

ASP.NET es un marco de trabajo que permite la creación de páginas web dinámicas, ejecutando el código del lado del servidor, para luego enviar el resultado al cliente que son los navegadores web. Fue lanzado en el 2002

como parte del proyecto de Microsoft “.NET Framework”. En 2012 se anuncia que ASP.NET y ciertos componentes como “ASP.NET MVC”, “Razor” y “Web API” serán distribuidos mediante una licencia de código abierto como es “Apache License 2.0”.

ASP.NET soporta tres tipos de modelos de desarrollo: “Páginas web”, “MVC” y “Formularios Web”. De estos modelos, el más popular es el MVC superando a los tradicionales “Formularios Web” dado que es más ligero y totalmente integrable con todas las funciones de ASP.NET. La programación fue realizada con la ayuda de una herramienta de Microsoft llamada “Microsoft Visual Studio Express 2012 para Web” que es una versión libre de “Visual Studio 2012” con la finalidad de implementar aplicaciones web.

El patrón de diseño MVC es conocido así por las siglas de sus componentes, que son: Modelo, Vista y Controlador. Es un modelo de programación que consta de estas tres capas para separar el desarrollo de un proyecto, para tener un estándar de programación y realizar una búsqueda de errores de forma separada entre cada capa del modelo.

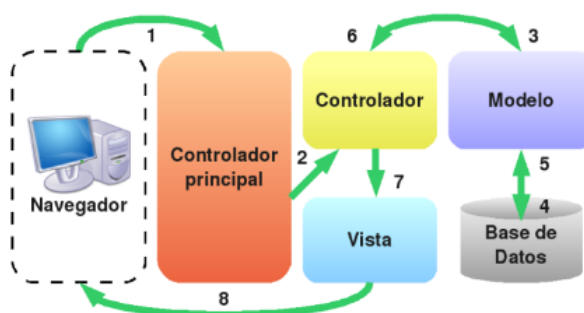


Figura 3.25 Diagrama de bloques de secuencia de datos.

El “Modelo” es la capa lógica encargada del manejo de los datos, por lo que en su mayoría esta capa obtiene y almacena registros de una base de datos. Para el desarrollo, la base de datos utilizada es “MySQL”. MySQL es un sistema de gestión de base de datos, actualmente su propietario es “Oracle Corporation” y se distribuye principalmente bajo licencia de código abierto “GNU GPL”, pero para aplicativos de uso comercial se debe adquirir una licencia.

La ventaja principal de esta base de datos es que es totalmente compatible con diversos tipos de sistemas operativos como Linux, Windows y MAC OS, manteniendo su rapidez en la lectura de datos. Es así que ha ido desarrollando múltiples herramientas para conexión con diversas plataformas, y para este trabajo se utiliza la versión desarrollada específicamente para conectarse con “Visual Studio”. Esta versión, a más de incluir el instalador de la base de datos incluye las librerías necesarias para que “Visual Studio” brinde las opciones al usuario para una conexión con la base de datos de forma gráfica y sencilla.

La administración de la base de datos puede realizarse mediante línea de comandos, pero a la vez poseen un administrador de su base de datos llamado "MySQL Workbench". Esta herramienta permite: crear las tablas, procedimientos, insertar registros, entre otras aplicaciones, de forma gráfica. Esta herramienta fue utilizada para la creación de las tablas del proyecto.

La base de datos ha sido netamente utilizada para el manejo de los usuarios que tienen acceso al aplicativo web. Es aquí donde se almacena el usuario y contraseña de los usuarios, así como los permisos que este obtiene una vez que haya ingresado al sistema. Se manejan permisos para cada uno de los botones principales, teniendo el administrador acceso a todas las opciones, sumado a la posibilidad de crear, borrar y editar los usuarios creados. Es el administrador el encargado de asignar a los usuarios creados las vistas a las que tienen acceso.

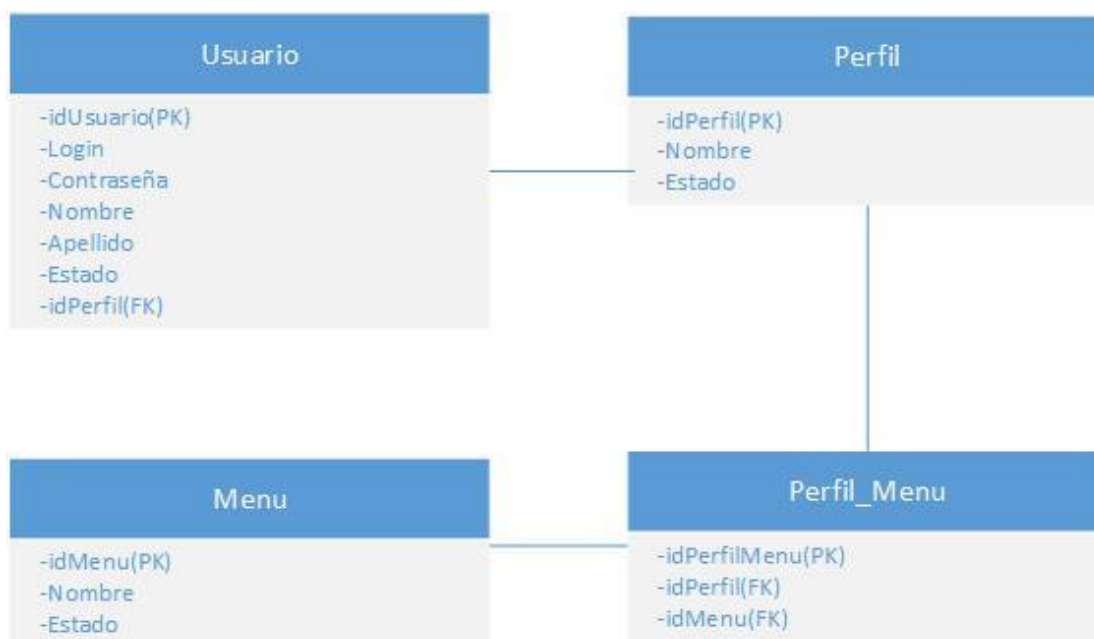
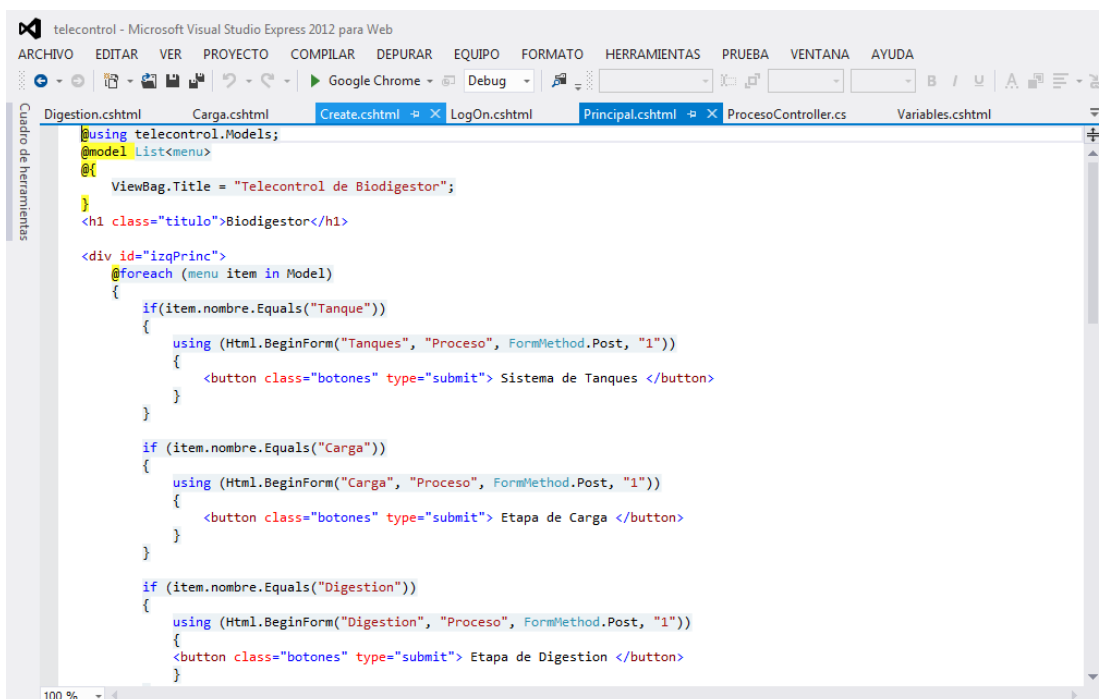


Figura 3.26 Diagrama UML de la base de datos.

La capa de “Vista” es la parte que muestra los datos al usuario y permite al mismo manipularlos mediante el navegador web. “Razor” es la sintaxis de programación de la capa de “Vista” que ha sido empleada para este trabajo. Razor es un motor de vistas y se encarga de proveer una sintaxis para la programación de las mismas. La sintaxis utilizada contiene como base a HTML, agregándole contenido dinámico mediante “C#” o “Visual Basic”. La extensión de las vistas depende del lenguaje de programación utilizado, si se usa “C#” el archivo de la vista es creado con extensión “.cshtml”, mientras si se usa “Visual Basic” la extensión es “.vbhtml”. Las extensiones son creadas de esta forma, ya que al final lo que el usuario observa en su navegador es una página en HTML, ya que todo el control agregado mediante el lenguaje de

programación, se realiza del lado del servidor y luego renderiza el valor que se va a mostrar al usuario como texto plano dentro de etiquetas HTML.

Al utilizar este motor de vistas se logra desligar de ciertas responsabilidades al controlador, realizando ciertas validaciones desde la misma vista y siendo capaz de contactarse directamente con el modelo, sin la necesidad de pasar por un controlador previamente. El lenguaje escogido para la implementación es “C#”.



```
using telecontrol.Models;
@model List<menu>
@{
    ViewBag.Title = "Telecontrol de Biodigestor";
}
<h1 class="titulo">Biodigestor</h1>
<div id="izqPrinc">
    @foreach (menu item in Model)
    {
        if(item.nombre.Equals("Tanque"))
        {
            using (Html.BeginForm("Tanques", "Proceso", FormMethod.Post, "1"))
            {
                <button class="botones" type="submit"> Sistema de Tanques </button>
            }
        }

        if (item.nombre.Equals("Carga"))
        {
            using (Html.BeginForm("Carga", "Proceso", FormMethod.Post, "1"))
            {
                <button class="botones" type="submit"> Etapa de Carga </button>
            }
        }

        if (item.nombre.Equals("Digestion"))
        {
            using (Html.BeginForm("Digestion", "Proceso", FormMethod.Post, "1"))
            {
                <button class="botones" type="submit"> Etapa de Digestion </button>
            }
        }
    }
}
```

Figura 3.27 Interfaz de programación de Visual Studio Express.

El “Controlador” es el encargado de manejar la interacción realizada por el usuario desde la “Vista” y de almacenar o extraer los datos del “Modelo”, además de actuar de intermediario entre las 2 capas mencionadas anteriormente. ASP.NET es capaz de manejar el “Controlador” mediante 2 lenguajes de programación como “C#” y “Visual Basic”, se ha escogido el desarrollo del mismo usando “C#”, debido a que su sintaxis permite al programador escribir menos código obteniendo la misma efectividad, adicionalmente “C#” ha llegado a tener mucha acogida en otras plataformas como el proyecto “Mono” para GNU/Linux, lo cual lo hace un lenguaje de programación en mayor crecimiento que “Visual Basic”.

3.5.3. CONTROLADOR – SERVIDOR WEB

La comunicación entre el controlador y el servidor que aloja el aplicativo web fue realizada a través de un cable “Ethernet” y utilizando el protocolo “Modbus” para el paso de la información entre ambos equipos.

El protocolo Modbus fue creado en un inicio para comunicación serial por la empresa “Modicon” que ahora es “Schneider Electric”. A partir de su creación empezó a ser empleado en los controladores lógicos programables, y fue creciendo hasta llegar a ser una de las principales opciones para comunicación entre dispositivos industriales electrónicos, volviéndose un protocolo estándar. Es común observar que sistemas de supervisión, control y adquisición de

datos (SCADA), a pesar de contar con conectores normalmente diseñados para cada fabricante, brindan la opción de conectarse con dispositivos mediante Modbus.

Existen dos variantes del protocolo modbus para el manejo de los datos, la más eficiente y comúnmente usada es “Modbus RTU”, pero ciertas implementaciones utilizan “Modbus ASCII” que es más fácil de entender pero resulta menos eficaz, ya que emplea una mayor cantidad de bytes para representar lo mismo que “Modbus RTU”.

Modbus TCP, es un desarrollo sobre el protocolo Modbus RTU que permite manejarlo sobre una red TCP/IP, siendo un protocolo de la capa de aplicación. Esto hace que la comunicación ya no sea serial, sino “ethernet” que es la tendencia a la que se apunta desde hace algunos años atrás. Mientras en Modbus RTU los dispositivos que intervienen son conocidos como “maestro” y “esclavo”, siendo el maestro el dispositivo que solicita la información al esclavo, en Modbus TCP se utiliza el esquema “cliente-servidor”, siendo el servidor un análogo al “esclavo” en RTU y de igual forma el cliente realiza la función del “maestro”.

Modbus TCP envía el mismo mensaje que Modbus RTU pero removiendo dos componentes, el control de redundancia cíclica (CRC) y el identificador del

esclavo. A su vez añade 7 bytes como encabezado MBAP (Modbus Application).

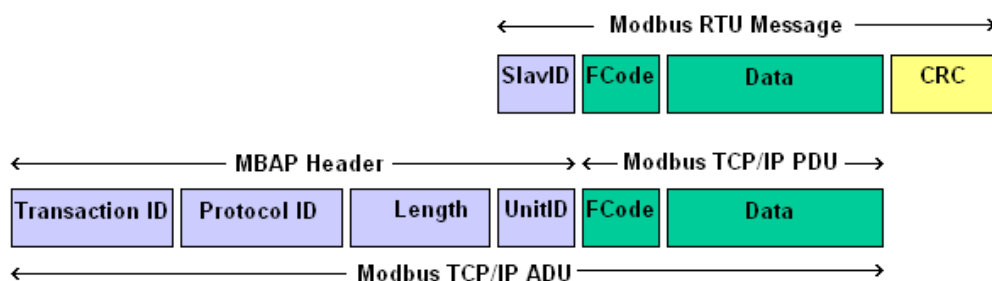


Figura 3.28 Estructura de mensajes Modbus.

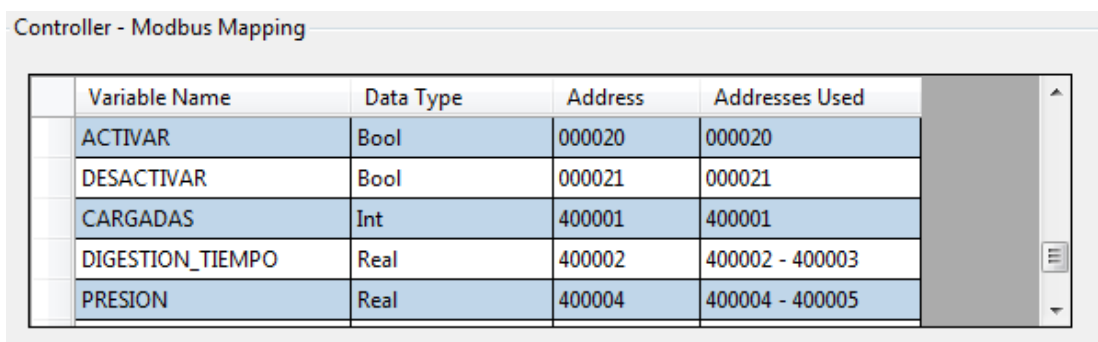
- Slave ID: Cada dispositivo esclavo tiene un número que lo identifica. Este lo utiliza el maestro para saber a quién contacta. No es usado en Modbus TCP.
- Function Code: Código que indica al esclavo el tipo de información y si el valor enviado es de lectura o de escritura.
- Data: El registro que se desea obtener o almacenar de una tabla.
- CRC: Control de Redundancia Cíclica, sirve para el manejo de los errores. No es usado en Modbus TCP.
- Transaction ID: El identificador de la transacción, son 2 bytes que sirven para mantener el orden de los paquetes en caso de retrasos en el envío.
- Protocol ID: Identificador del protocolo, el cliente de Modbus TCP siempre los ubica con valor: 00 00.

- Length: El tamaño son 2 bytes que indican el número de bytes que se recibirán a continuación como parte del mensaje.
- Unit Identifier: Es un análogo al identificador del esclavo, identifica al dispositivo que se realiza la consulta.

El controlador micro850 de “Allen-Bradley” es capaz que manejar el protocolo Modbus TCP. Para ello se utilizan cuatro tablas, cada una capaz de almacenar 9999 registros. Dentro de cada registro se almacena el valor de una variable, y desde el aplicativo web se leen estos registros cada 0,5 segundos.

Tablas de direccionamiento Modbus RTU y TCP			
<u>Uso</u>	<u>Tipo de Datos</u>	<u>Permisos</u>	<u>Direcciones</u>
Salidas - Variables Digitales	Bool	Lectura, Escritura	0001 – 9999
Entradas Digitales	Bool	Lectura	10001 – 19999
Entradas Analógicas	Int, Real	Lectura	30001 – 39999
Salidas – Variables Analógicas	Int, Real	Lectura, Escritura	40001 - 4999

Tabla # 1. Tabla de direccionamiento Modbus RTU y TCP



Variable Name	Data Type	Address	Addresses Used
ACTIVAR	Bool	000020	000020
DESACTIVAR	Bool	000021	000021
CARGADAS	Int	400001	400001
DIGESTION_TIEMPO	Real	400002	400002 - 400003
PRESION	Real	400004	400004 - 400005

Figura 3.29 Mapeo de variables Modbus TCP

El servidor web realiza la función de cliente Modbus TCP. Para ello se ha utilizado una librería en C# que permite a la aplicación ser capaz de realizar las peticiones de datos y cambiar los valores de las variables de forma remota. La librería en cuestión fue distribuida por Stephan Stricker mediante licencia CPOL (Code Project Open License). Esto es un sitio donde desarrolladores de diversas partes del mundo comparten sus trabajos para el uso de los demás usuarios sin fines comerciales. La librería se llama “Modbus TCP Class” y fue añadida en el proyecto brindando la capacidad de comunicar el PLC con el servidor web, mediante Modbus TCP.

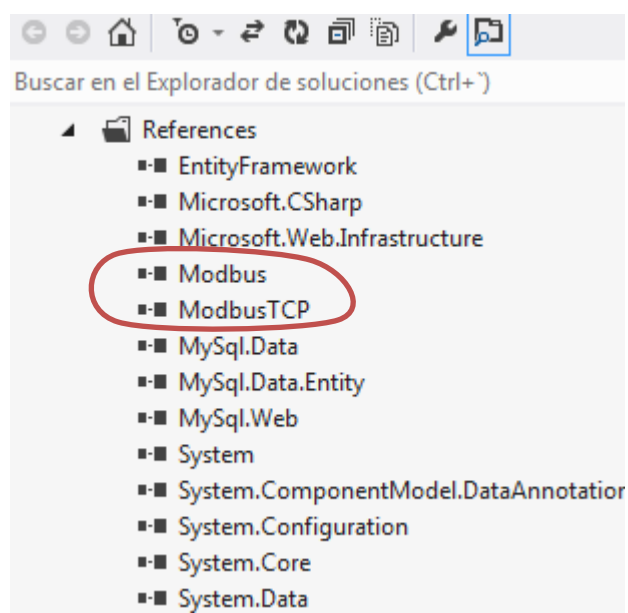


Figura 3.30 Librería del protocolo Modbus para lenguaje C#

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS

4.1. CALIBRACIÓN DE SENSORES

Para poder obtener una correcta medición de los datos, es necesario calibrar cada uno de los sensores analógicos y digitales, comparándolos con instrumentación dedicada para este tipo de variables.

4.1.1. CALIBRACIÓN DE SENSOR DE NIVEL

Para este proyecto se utilizó electrodos metálicos que al entrar en contacto con el agua, cierran un circuito y permiten la conmutación de un relé entre un nivel alto y bajo de voltaje.

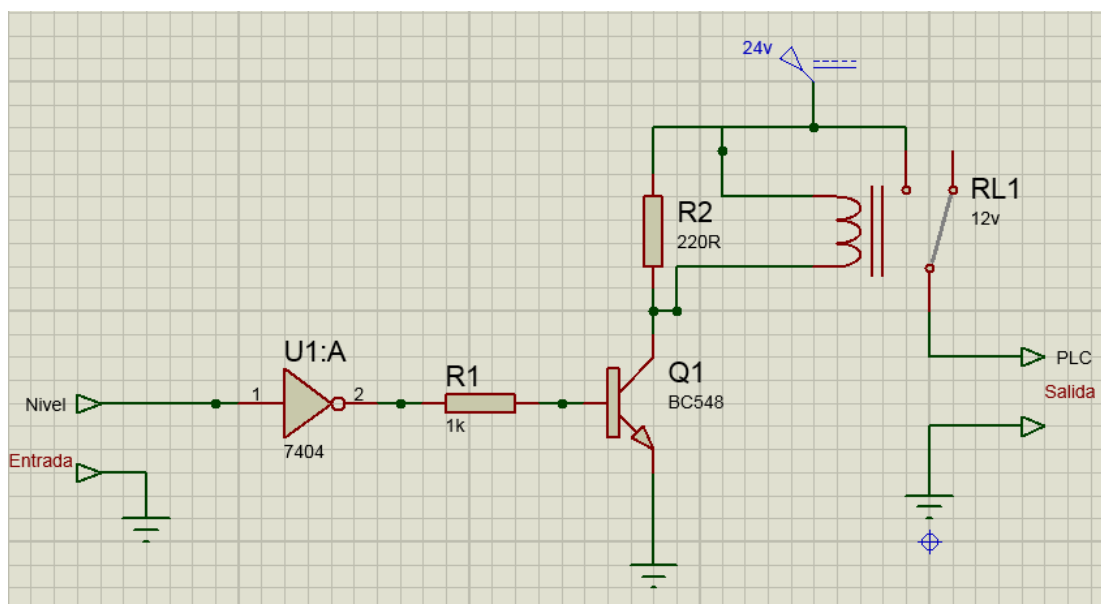


Figura 4.1 Circuito sensor de nivel.

Para realizar la calibración de este sensor, simplemente se colocaron los electrodos en un recipiente, luego se rellena con agua dicho recipiente hasta que los electrodos entren en contacto con el líquido. Se pudo verificar que la señal se activa cuando los electrodos hacen contacto mediante el agua.

4.1.2. CALIBRACIÓN DE SENSOR DE TEMPERATURA

Para este proyecto se ha utilizado el sensor de temperatura LM35 con una precisión calibrada de 1°C . Puede medir temperaturas en un rango que abarca desde -55°C hasta $+155^{\circ}\text{C}$. Este sensor tiene un comportamiento lineal, donde la variación de voltaje por cada grado centígrado equivale a 10mV en la salida.



Figura 4.3 Punta de prueba de pH American Marine PINPOINT.

El voltaje que entrega la punta de prueba del pH ingresa a un acoplador, el cual amplifica y regula la señal, para luego ingresarla al controlador. La señal amplificada y acoplada oscila entre 0v y 1.1v.

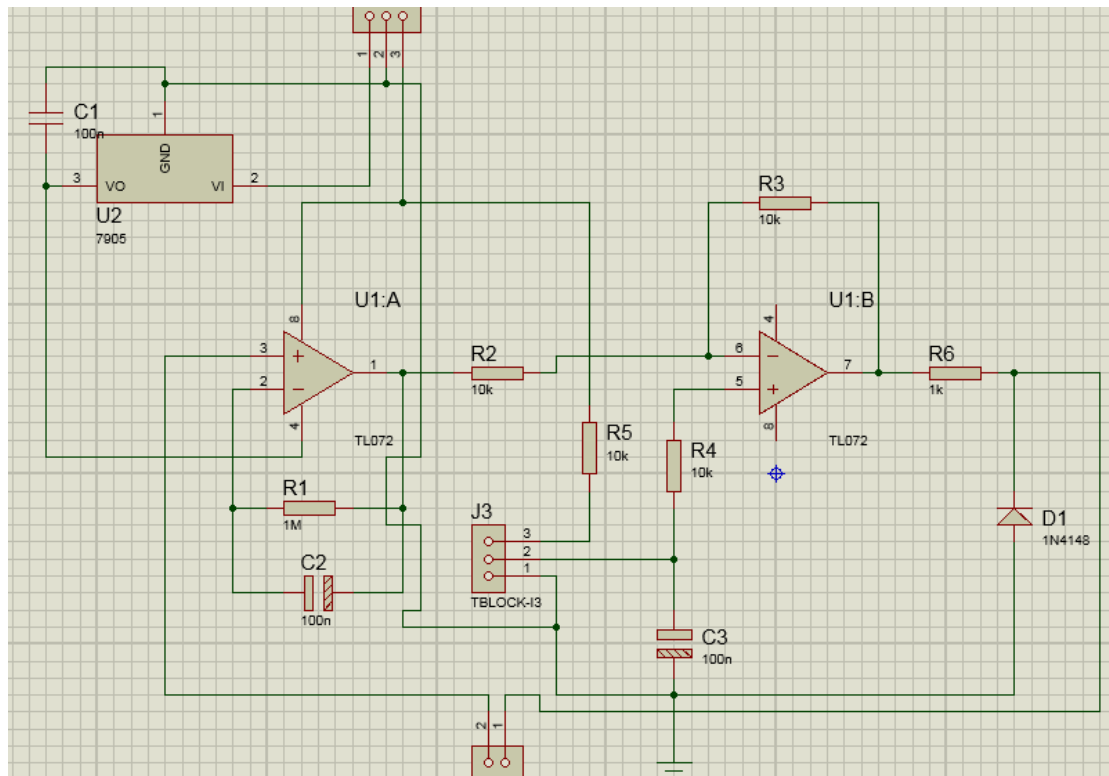


Figura 4.4 Circuito amplificador de señal de pH.

La calibración del circuito acoplador se la realiza variando la resistencia del potenciómetro hasta que el voltaje medido en el capacitor C3 sea de 0.275V. Para verificar que los valores medidos sean los correctos, se comparó con un colorímetro, obteniendo un error aproximado de 0.2 unidades de pH.

4.1.4. SENSOR DE PRESIÓN

Para esto se ha utilizado un sensor Motorola MPX5700GP que tiene una medición de 0 a 700KPA con una salida de voltaje de 0 a 4.7v. Para este

sensor no se utilizó circuito acoplador, ya que el voltaje de salida está dentro del rango admitido por el controlador.



Figura 4.5 Sensor de presión MPX5700GP

La calibración de este sensor se la realizó introduciendo aire en un envase cerrado y verificando que el valor medido por el sensor, aumente en función a la presión suministrada.

4.2. PRUEBAS DE FUNCIONAMIENTO

Para comprobar el correcto funcionamiento del proceso de automatización propuesto para el biodigestor, se ha diseñado una maqueta. La maqueta es una muestra a pequeña escala del sistema propuesto, utilizando los elementos mencionados anteriormente y añadiendo ciertos detalles para ajustar la solución a la maqueta.

La maqueta ha sido realizada usando madera mdf de 15mm con una estructura similar a un armario de tres repisas, donde cada repisa está enfocada a mostrar una de las etapas del proceso. Y del lado derecho se ubican el controlador, los relés, fuentes de alimentación y demás equipos utilizados para el proceso, ubicados en rieles y con canaletas y cerrado bajo llave para único acceso del administrador.

En la primera repisa se encuentra el sistema de tanques. Cada tanque fue realizado con cubos de acrílico. Las medidas de los tanques de agua y de desechos es la misma y es de 15x15x25 cm con un volumen de 5625 cm³, mientras que para los tanques balanceadores de pH se han utilizado tanques de 10x10x15 cm con un volumen de 1500 cm³. Dentro de cada uno de los tanques se han colocado los sensores de nivel bajo, incluso para el tanque de agua, debido a que utilizar una bomba de agua en estos pequeños tanques, pueden ocasionarles un daño y la boya sería muy difícil de colocar. Por esto se ha decidido que para la maqueta el tanque de agua también tendrá un sensor y alertará en caso de que el nivel sea inferior al debido.

Entre la primera y segunda repisa, debajo de cada tanque se encuentra una electroválvula que permite el paso del material al reservorio de la etapa de carga, ubicada en la segunda repisa. El reservorio de la segunda repisa, al igual que los tanques, ha sido realizado con un cubo acrílico de 30x20x25 cm

y con un volumen 150000 cm³, siendo así capaz de recibir el material proveniente de los tanques mencionados anteriormente.

Dentro del tanque se han colocado los tres sensores de nivel para determinar el flujo de los desechos, del agua y para conocer en qué momento el reservorio está vacío. En la segunda repisa se encuentra la punta de prueba del sensor de pH, ubicado dentro del reservorio, para que envíe los datos al controlador. Además se cuenta con un motor de limpiaparabrisas de vehículos, que realiza el giro de unas aspas soldadas en él, para realizar la mezcla de los desechos y el agua.

Debajo de la segunda repisa y pegado al reservorio, se ubica la electroválvula para el paso de la mezcla homogenizada al biodigestor. Para el biodigestor se ha utilizado una bolsa de plástico, donde se ha colocado la válvula de ingreso de la mezcla y la válvula de salida de gas en la parte superior, mientras en la parte inferior se encuentra la válvula de salida del biosol.

Dentro del biodigestor se ha colocado el sensor de nivel para saber cuándo se encuentra vacío el biodigestor, el sensor de temperatura que permitirá conocer si el tiempo de digestión utilizado ha sido o no el óptimo, y el sensor de presión que indicará el momento en que se debe abrir la válvula de salida de gas.



Figura 4.6 Maqueta.

Dado que para la prueba de funcionamiento no se usarán desechos, será imposible la generación del gas, para ello se utilizó un pequeño compresor que ingresará aire en la bolsa del biodigestor hasta llegar a generar la presión en la superficie de la bolsa. El quemador es simulado mediante una sirena que

alertará el encendido de esta variable, en caso de algún inconveniente en el proceso.

Para la prueba del sistema SMS se colocó una tarjeta SIM en el módulo integrable, con saldo suficiente para el envío de todas las alertas generadas. Además la antena del módulo está ubicada en la parte superior de la maqueta para mejorar la recepción de la señal.

El servidor web es colocado en una PC con Windows utilizando IIS (Internet Information Services) como servidor web. El controlador, la pantalla y la PC se encuentran conectados en una red LAN, para la comunicación entre los tres dispositivos. Se utiliza un enrutador inalámbrico para la conexión del usuario final con el servidor web, tanto desde una laptop como de dos dispositivos móviles.



Figura 4.7 Conexiones del panel de control.

4.2.1. PRUEBA DE ACTUADORES

Se utilizó electroválvulas metálicas de 12v normalmente cerrada de acción directa, con una válvula de 0.25 pulgadas de diámetro.



Figura 4.8 Electroválvula genérica de 12v.

Se verificó que al enviar la señal desde el controlador, la válvula se abre y da paso al líquido. Así mismo, al desactivar la señal, la electroválvula vuelve a su estado inicial.

Se utilizó un motor de corriente continua de 12 V, para simular el giro del agitador. Este es un motor de limpiaparabrisas de un bus de transporte que al enviar la señal desde el controlador, gira en ambos sentidos.



Figura 4.9 Motor de limpiaparabrisas de 12v.

4.3. RESULTADOS

El resultado de la realización de la maqueta ha sido satisfactorio, notándose que el funcionamiento del proceso es el adecuado para controlar las variables propuestas en el diseño de automatización. Es claro que la solución debe ser adecuada para una implementación real, en cuanto a los elementos físicos empleados para la maqueta, como por ejemplo: el motor utilizado debe ser de una mayor potencia, la válvula de desechos de mayor diámetro, entre otros detalles físicos, pero la lógica de programación utilizada sirve correctamente para la implementación real.

El valor máximo de la presión es ajustable, debido a que el mismo dependerá del material utilizado para el biodigestor. En el caso de la bolsa plástica, se realizaron pruebas previas, notándose que la presión máxima de la bolsa utilizada es de 15,5 psi.

Los mensajes de texto generados de las alertas enviadas, fueron entregados con éxito a los números celulares establecidos. De igual manera, cuando el usuario consultaba por uno de los códigos predeterminados, el sistema le entregaba la respuesta mediante el mensaje de texto únicamente a quien realizó la petición.

En el sistema web fue posible el manejo de los usuarios, brindando a cada uno de ellos el permiso respectivo, además de la paralización del proceso y realizar el método manual de activación de los actuadores por parte de un usuario administrador, teniendo así éxito en el diseño del aplicativo web.

Referente a las conexiones de los equipos, el medio fue "Ethernet", manteniendo siempre la prueba dentro de una red LAN. Esquema que puede ser utilizado en redes WAN, sin necesidad de realizar cambios en la configuración de los dispositivos intervinientes, sino que creando los accesos remotos respectivos, mediante VPNs o la publicación del servidor web a una IP pública mediante NAT.

CONCLUSIONES

Los factores que se consideran principales para poder determinar el éxito del trabajo presentado son: la adaptabilidad del sistema a diversos tipos de biodigestores, la facilidad de uso para los usuarios finales y la interacción entre todas las interfaces HMI.

1. El sistema de automatización diseñado para el telecontrol del proceso de biodigestión es adaptable a todo tipo de biodigestor, debido a que la mayor parte del control se lo realiza en una etapa previa al ingreso del material al biodigestor, para asegurar que el material a fermentar es óptimo para la generación de gas. Esto añade una gran ventaja a los usuarios que ya poseen un biodigestor y presenta una buena alternativa para aquellos que deseen implementar uno, pero se preocupan de los detalles que implica el cuidado del mismo.

2. El proceso de biodigestión incluye de forma predeterminada la generación de biosol, un fertilizante natural de alta calidad. Pese a que el trabajo presentado se ha enfocado en la generación de biogás, es adaptable para poder generar biosol en mayores cantidades y de mejor calidad, variando el nivel de pH de la mezcla a fermentar. Para ello el usuario puede cambiar los niveles máximo y mínimo de pH, en caso de querer generar biosol sin tomar en cuenta la cantidad de biogás generado.
3. El sistema brinda una alternativa tecnológica, diseñada de tal modo que el usuario no deba tener amplios conocimientos técnicos para manipular el sistema de biodigestión. Para ello se han empleado diversas interfaces intuitivas que permiten entender cada uno de los elementos del sistema. Dichas interfaces gráficas utilizadas son similares, lo que facilita el entendimiento del usuario y le brinda la misma experiencia, tanto en sitio como remotamente.
4. Se plantearon diversas formas de administración para que en caso de fallar alguna, el usuario no pierda el control de su sistema. Cada una de las interfaces de administración son capaces de reflejar los cambios realizados desde otra interfaz, validando que no interfiera una con otra. Por esto se puede concluir que las interfaces son capaces de interactuar entre sí.

5. La aplicación web es capaz de procesar tantas peticiones como las características físicas del servidor lo permita, y es capaz de manejar diversos usuarios presentes al mismo tiempo, facilitando a cada usuario la observación en tiempo real los cambios efectuados por otro usuario.

6. La utilización de equipos dedicados a la automatización de procesos industriales, asegura el rendimiento del elemento principal que es el controlador, diseñado para soportar inclemencias climatológicas. A pesar de ser equipos normalmente obligados a la utilización de accesorios de su propia marca, se ha demostrado en este trabajo que también son capaces de acoplarse a una gama de sensores y actuadores de diversos fabricantes.

7. La marca “Rockwell Automation” posee un programa llamado “Factory Talk” que permite la implementación de telecontrol de sus controladores. La desventaja del mismo es que por cada usuario que accede al sistema se paga una licencia. Ante esto se presenta la alternativa del servidor web, que no requiere ningún tipo de licencias para el funcionamiento y trabaja con protocolos libres y comúnmente usados en la industria de automatización.

Por los factores mencionados anteriormente, se concluye que se trata de una propuesta innovadora y que puede llegar a tener auge en el país en las zonas rurales y en las industrias.

RECOMENDACIONES

1. El biodigestor es muy poco usado en el país, por lo que la principal recomendación consiste en realizar una campaña de difusión de este sistema en las zonas rurales, ya que debido al cambio de la matriz productiva, él mismo llegará a tener gran acogida, ya que el gas como fuente de energía es ampliamente útil.
2. Para la implementación real del sistema, se deberá tener en cuenta sensores y actuadores acorde al tipo de biodigestor, manteniendo el mismo controlador con su lógica de programación.
3. Las características físicas requeridas para el servidor web, dadas las constantes transacciones de consultas que realiza al controlador, deben tener al menos las siguientes características:
 - Memoria RAM: 2GB Mínimo ; 4GB Recomendado
 - Procesador: Intel Core 2 Duo 3GHz o Superior
 - Disco Duro: 50GB o Superior
4. La virtualización de servidores es una gran alternativa hoy en día, debido a que existen programas libres que permiten realizarlo. Dado que el servidor no requiere de muchas características, se puede considerar

adquirir un servidor e instalar en él un virtualizador, como por ejemplo “Xen Server”. En este mismo servidor se puede crear dos máquinas virtuales, una a cargo del servidor web y otra como base de datos para los usuarios.

5. Un aporte adicional al sistema, sería el almacenamiento de las variables que intervienen en el proceso dentro de la base de datos. Esto permitirá al administrar no solo tener telecontrol del sistema, sino que también tener un registro del estado de las variables en el proceso. Además es factible colocar el gas en un contenedor y ubicar un fluxómetro para poder llevar control de la cantidad de gas generada en cada proceso.

BIBLIOGRAFÍA

- [1] Lopez P., Solà A., “Sistematización y cuantificación de biodigestores”,
http://www.pequenasdonacionescr.org/component/docman/doc_download/71-biodigestores-impacto-social-economico-y-ambiental
- [2] Galán P., Proaño J., Sarmiento J., Toro J., “Diseño y Automatización de un Biodigestor”, Trabajo de Graduación, Universidad del Azuay
- [3] La Bioguía, “Biodigestores”, Fecha de Publicación: 03 de Enero del 2013.
<http://www.labioguia.com/biodigestores/>
- [4] Universo Porcino, “Instalaciones Porcinas: Biodigestores”, Fecha de Publicación: 09 de Abril del 2008.
http://www.universoporcino.com/articulos/que_es_un_biodigestor.html
- [5] Rockwell Automation, “Micro800 and Connected Components Workbench”,
http://literature.rockwellautomation.com/idc/groups/literature/documents/qr/2080-qr001_-en-p.pdf
- [6] ProSoft Technology, “ILX800-SMSG Micro800 Platform User Manual”,
<http://www.prosoft->

technology.com/content/download/6995/66053/version/4/file/ILX800_SMSG_User_Manual.pdf

[7] Muñoz J., “ESTUDIO DE APLICACIÓN DE LOS ESTÁNDARES DEVICENET Y CONTROLNET DE COMUNICACIONES INDUSTRIALES COMO SOLUCIÓN DE RED DE CAMPO Y PROCESO EN UNA PLANTA INDUSTRIAL”, Trabajo de Titulación, Universidad Austral de Chile.
<http://cybertesis.uach.cl/tesis/uach/2007/bmfcim971e/doc/parte/ii.pdf>

[8] Modbus-IDA, “MODBUS APPLICATION PROTOCOL SPECIFICATION”,
Fecha de Publicación: 28 de Diciembre de 2006.
http://www.modbus.org/docs/modbus_application_protocol_v1_1b.pdf

[9] Simply Modbus, “About Modbus”,
<http://www.simplymodbus.ca/FAQ.htm#FC>

[10] Burger T., “ASP.NET Razor view engine”, Última Edición: 02 de Febrero del 2015.
http://en.wikipedia.org/wiki/ASP.NET_Razor_view_engine

[11] W3School, “ASP.NET MVC TUTORIAL”,
http://www.w3schools.com/aspnet/mvc_intro.asp

[12] Stricker S., "Modbus TCP Class", Fecha de Publicación: 23 de Mayo del 2014,

<http://www.codeproject.com/Tips/16260/Modbus-TCP-class>

[13] Anónimo, "The HTML5 meter Element", Fecha de Publicación: 26 de Noviembre del 2013,

<https://css-tricks.com/html5-meter-element/>

[14] Montalvo A., "Sensores de Nivel", Fecha de Publicación: 22 de Enero del 2012,

<http://es.slideshare.net/AlanMontalvo/sensores-de-nivel-11210794>

[15] Camilosw, "Sensor", Última Edición: 01 de Mayo del 2015,

<http://es.wikipedia.org/wiki/Sensor>

[16] Anónimo, "Tipos de Actuadores Eléctricos",

http://www.ite.educacion.es/formacion/materiales/47/cd/mod2/2d_2.htm

[17] Sanhuesa A., "Sensores de Presión",

<http://sensoresdepresion.blogspot.com/>

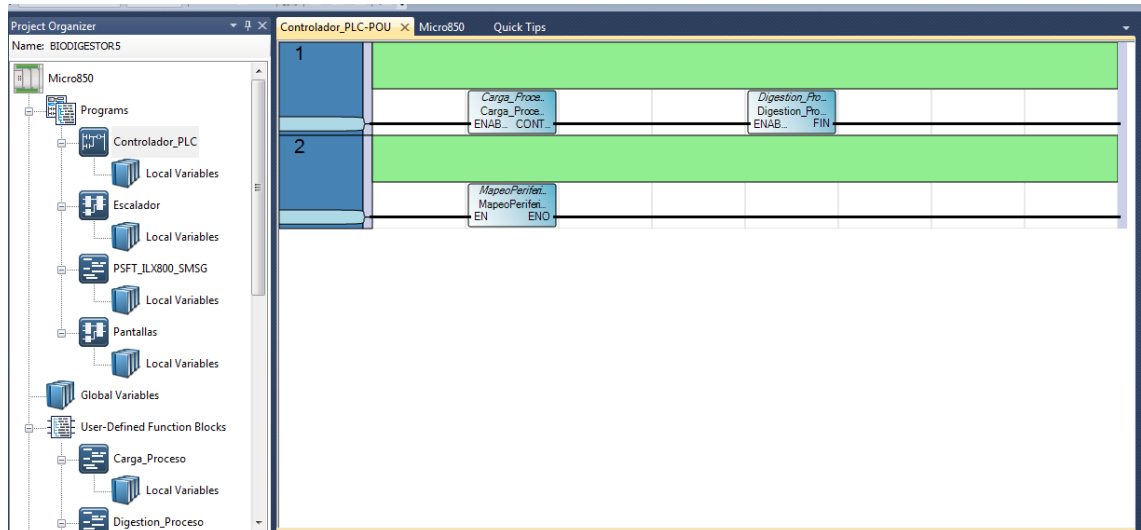
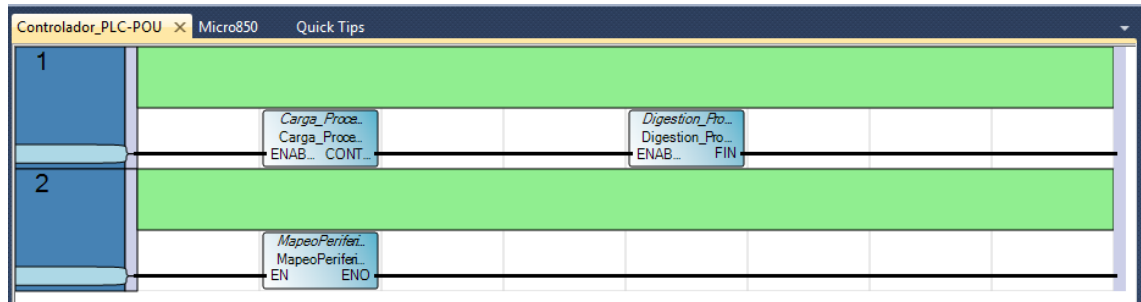
[18] Ortisa, “Electroválvula”, Última Edición: 16 de Abril del 2015,
<http://es.wikipedia.org/wiki/Electrov%C3%A1lvula>

[19] Technopat, “PH-metro”, Última Edición: 13 de Abril del 2015,
<http://es.wikipedia.org/wiki/PH-metro>

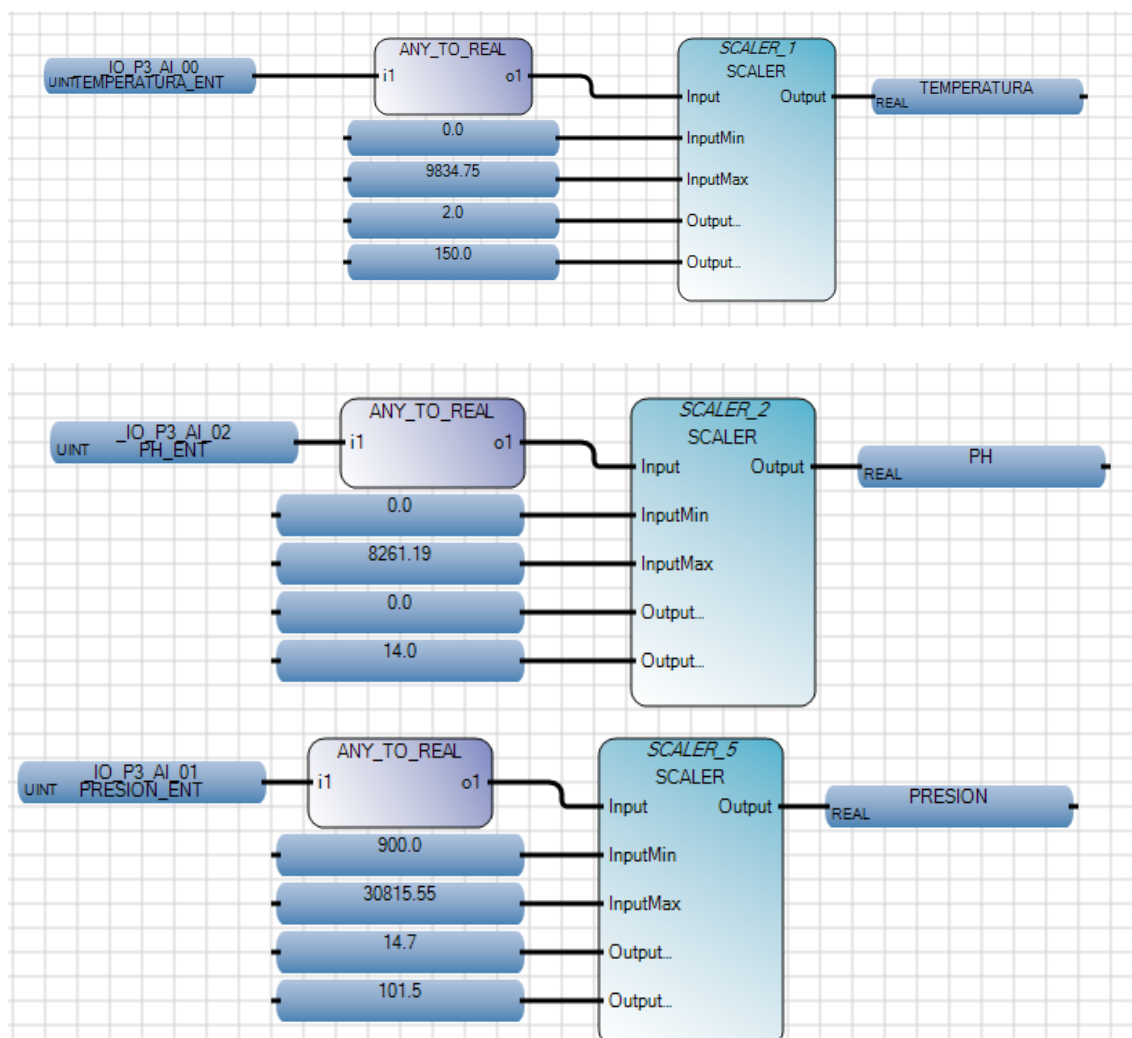
[20] Aparcana S., Jansen A., “Estudio sobre el Valor Fertilizante de los Productos del Proceso Fermentación Anaeróbica para producción de Biogás”,
http://www.german-profec.com/cms/upload/Reports/Estudio%20sobre%20el%20Valor%20Fertilizante%20de%20los%20Productos%20del%20Proceso%20Fermentacion%20Anaerobica%20para%20Produccion%20de%20Biogas_ntz.pdf

ANEXOS

Anexo A: Diagrama de Bloque General del Proceso.



ANEXO B: ESCALADOR.



ANEXO C: PROGRAMA DE EJEMPLO DE PROSOFT, CON LOS CAMBIOS REALIZADOS.

(* -----)

[POU NAME]: PSFT_ILX800_SMSG

[Category]: Plug-in

[Description]: Sample program for the ProSoft ILX800-SMSG plug-in
module

[Version]: 1.00

[Release Date]: June 28, 2012

-----*)

(*-----***** DO NOT MODIFY *****-----*)

IF __SYSVA_CYCLEDATE > Time#15s THEN

 UTIL_SMSInitTime := Time#10ms;

END_IF;

UTIL_progToRun(TRUE, UTIL_SMSInitTime);

UTIL_executeCmd(NOT UTIL_executeCmd.Q, time#1000ms);

UTIL_counter(UTIL_executeCmd.Q, UTIL_counter.Q, 2);

UTIL_R_TRIG_2(UTIL_progToRun.Q);

UTIL_R_TRIG_1(CONFIG_whiteListEnabled);

UTIL_F_TRIG_1(CONFIG_whiteListEnabled);

IF UTIL_R_TRIG_2.Q THEN

 UTIL_initialized := false;

 UTIL_SMSConfiged := false;

 UTIL_configRetryCount := 0;

```
    UTIL_sendMsgRetryCount := 0;

    CONFIG_ackEnabled := true;

END_IF;

IF UTIL_R_TRIG_1.Q OR UTIL_F_TRIG_1.Q THEN

    UTIL_SMSConfiged := false;

    UTIL_sendMsgRetryCount := 0;

END_IF;

IF not UTIL_SMSConfiged AND UTIL_counter.CV = 1 AND
UTIL_executeCmd.Q THEN

    UTIL_SMSPLUGIN_CONFIG_1( TRUE, CONFIG_slotID,
UTIL_PINNumber, CONFIG_whiteList, CONFIG_whiteListEnabled);

    IF UTIL_SMSPLUGIN_CONFIG_1.Q THEN

        IF UTIL_SMSPLUGIN_CONFIG_1.ErrCode = 0 THEN

            UTIL_SMSConfiged := true;

            UTIL_retryCount := 0;

            ELSE

                UTIL_configRetryCount := UTIL_configRetryCount + 1;

                IF UTIL_configRetryCount > 50 THEN

                    UTIL_SMSConfiged := true;

                    STATUS_errCount := STATUS_errCount + 1;

                END_IF;

            END_IF;
```

```
        END_IF;

    END_IF;

END_IF;

(*-----***** DO NOT MODIFY *****-----*)

(*-----*)
(* This section sends out a greeting SMS message through the module *)
(* when the module boots up. This message is sent to all receivers. *)
(* If you do not want the module to send out a greeting SMS message, *)
(* delete this entire section. *)
(* IMPORTANT: If you delete this section, you must uncomment the next *)
(* line and allow it to remain in the code, else the module will not *)
(* work. *)

(*UTIL_initialized := TRUE;*)

IF UTIL_initialized = FALSE AND UTIL_counter.CV = 2 AND
UTIL_executeCmd.Q = TRUE THEN
```

```
    UTIL_SEND_SMS_1(NOT UTIL_initialized, CONFIG_slotID,  
CONFIG_receiverList, DATA_greetingSMSText);  
    IF UTIL_SEND_SMS_1.Q = TRUE AND UTIL_SEND_SMS_1.ErrCode =  
0 THEN  
        UTIL_initialized := TRUE;  
        UTIL_retryCount := 0;  
    ELSE  
        IF UTIL_SEND_SMS_1.Q = TRUE THEN  
            IF UTIL_SEND_SMS_1.ErrCode <> 1 THEN  
                UTIL_sendMsgRetryCount := UTIL_sendMsgRetryCount + 1;  
            END_IF;  
            IF UTIL_sendMsgRetryCount > 50 THEN  
                UTIL_initialized := TRUE;  
                STATUS_errCount := STATUS_errCount + 1;  
            END_IF;  
        END_IF;  
    END_IF;  
END_IF;  
  
(*-----*)
```

```

(*-----***** DO NOT MODIFY *****-----*)
UTIL_systemInitialized := UTIL_initialized AND UTIL_SMSConfiged;
UTIL_TON_1( NOT UTIL_TON_1.Q, time#1s );
UTIL_TON_2( NOT UTIL_TON_2.Q, time#1503ms);
IF UTIL_TON_2.Q THEN
    UTIL_getDiagData := TRUE;
END_IF;
UTIL_READ_SMS_1( UTIL_TON_1.Q, CONFIG_slotID );
IF UTIL_TON_1.Q THEN
    IF UTIL_READ_SMS_1.Q AND UTIL_READ_SMS_1.ErrCode = 0
    THEN
        IF MLEN(UTIL_READ_SMS_1.SMSText)>0 AND
        MLEN(UTIL_READ_SMS_1.SMSText)<25 THEN
            DATA_recvedSMSText := UTIL_READ_SMS_1.SMSText;
            UTIL_newSMSRecved := TRUE;
            UTIL_sender := UTIL_READ_SMS_1.Sender;
        ELSE
            UTIL_newSMSRecved := FALSE;
        END_IF;
    END_IF;
END_IF;

```



```

END_IF;

UTIL_GET_DIAG(UTIL_systemInitialized AND UTIL_getDiagData,
CONFIG_slotID);

IF UTIL_getDiagData AND UTIL_GET_DIAG.Q THEN

    STATUS_signalStrength :=
((ANY_TO_INT(UTIL_GET_DIAG.SignalStrength)*2)-113);

    STATUS_networkStatus := UTIL_GET_DIAG.NetworkStatus;

    STATUS_SMSSendCount := UTIL_GET_DIAG.SMSSentCounter;

    STATUS_SMSRecvCount := UTIL_GET_DIAG.SMSRecvCounter;

    STATUS_SMSSendFailCount :=
UTIL_GET_DIAG.FailedSMSSentCounter;

    STATUS_netConnFailedRetries :=
UTIL_GET_DIAG.NetConnFailedRetries;

    STATUS_WLFiltCount := UTIL_GET_DIAG.FilteredSMSReceived;

    STATUS_SIMCardStatus := UTIL_GET_DIAG.SIMCardStatus;

    STATUS_majorRev := UTIL_GET_DIAG.MajorRev;

    STATUS_minorRev := UTIL_GET_DIAG.MinorRev;

    UTIL_getDiagData := false;

END_IF;

(*-----***** DO NOT MODIFY *****-----*)

```

```
(*-----*)
(* Generic Event-Triggered Text Message *)
(* Use this section to send out a user-defined SMS message to all *)
(* numbers on the Receiver List. Enter your desired text message string *)
(* in the Initial Value column for the DATA_customSMSText variable. *)
(* The message is sent when the program detects a HIGH or TRUE on the *)
(* DATA_customSMSTrigger bit. *)
```

```
IF DATA_customSMSTrigger = TRUE THEN
```

```
    UTIL_SEND_SMS_5( TRUE, CONFIG_slotID, CONFIG_receiverList,
DATA_customSMSText);
```

```
    DATA_customSMSTrigger := FALSE;
```

```
END_IF;
```

```
(*-----*)
```

```
IF ALERTA_DESECHOS = TRUE THEN
```

```
    SMS_COUNTER1:=SMS_COUNTER1+1;
```

```
    IF SMS_COUNTER1=1 THEN
```

```
SMS_ALERTA:= ALERTA;
```

```
    UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
```

```
ELSIF SMS_COUNTER1=3 THEN
    SMS_COUNTER1:=2;
END_IF;
END_IF;

IF ALERTA_DESECHOS = FALSE THEN
    SMS_COUNTER1:=0;
END_IF;

IF ALERTA_AGUA = TRUE THEN
    SMS_COUNTER8:=SMS_COUNTER8+1;
    IF SMS_COUNTER8=1 THEN
SMS_ALERTA:= ALERTA;
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
        ELSIF SMS_COUNTER8=3 THEN
            SMS_COUNTER8:=2;
        END_IF;
    END_IF;

IF ALERTA_AGUA = FALSE THEN
    SMS_COUNTER8:=0;
END_IF;

IF ALERTA_ACIDO = TRUE THEN
    SMS_COUNTER9:=SMS_COUNTER9+1;
```

```
        IF SMS_COUNTER9=1 THEN
SMS_ALERTA:= ALERTA;
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
        ELSIF SMS_COUNTER9=3 THEN
            SMS_COUNTER9:=2;
        END_IF;
    END_IF;

    IF ALERTA_ACIDO = FALSE THEN
        SMS_COUNTER9:=0;
    END_IF;

    IF ALERTA_ALCALINO = TRUE THEN
        SMS_COUNTER10:=SMS_COUNTER10+1;
        IF SMS_COUNTER10=1 THEN
SMS_ALERTA:= ALERTA;
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
        ELSIF SMS_COUNTER10=3 THEN
            SMS_COUNTER10:=2;
        END_IF;
    END_IF;

    IF ALERTA_ALCALINO = FALSE THEN
```

```
SMS_COUNTER10:=0;

END_IF;

(* Alerta al abrir valvula de desechos*)

IF _IO_EM_DO_01 = TRUE THEN

    SMS_COUNTER2:=SMS_COUNTER2+1;

    IF SMS_COUNTER2=1 THEN

SMS_ALERTA:='Se ha abierto la valvula del tanque de desechos';

        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,

SMS_ALERTA);

    ELSIF SMS_COUNTER2=3 THEN

        SMS_COUNTER2:=2;

    END_IF;

END_IF;

IF _IO_EM_DO_01 = FALSE THEN

    SMS_COUNTER2:=0;

END_IF;

(* Alerta al abrir valvula de agua*)

IF _IO_EM_DO_00 = TRUE THEN

    SMS_COUNTER3:=SMS_COUNTER3+1;

    IF SMS_COUNTER3=1 THEN

SMS_ALERTA:='Se ha abierto la valvula del tanque de agua';
```

```
    UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);

    ELSIF SMS_COUNTER3=3 THEN

        SMS_COUNTER3:=2;

    END_IF;

END_IF;

IF _IO_EM_DO_00 = FALSE THEN

    SMS_COUNTER3:=0;

END_IF;

IF SMS_FINAL_CARGADA = TRUE THEN

    SMS_ALERTA:='La mezcla ha sido homogenizada y enviada al
biodigestor';

    UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);

END_IF;

IF CARGADAS = MAX_CARGADAS THEN

    SMS_COUNTER4:=SMS_COUNTER4+1;

    IF SMS_COUNTER4=1 THEN

        SMS_ALERTA:='El biodigestor está lleno y va a comenzar el proceso de
digestion';
```

```
    UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,  
SMS_ALERTA);  
  
    ELSIF SMS_COUNTER4=3 THEN  
        SMS_COUNTER4:=2;  
    END_IF;  
END_IF;  
  
IF CARGADAS <> MAX_CARGADAS THEN  
    SMS_COUNTER4:=0;  
END_IF;  
  
IF TEMP_ALTA = TRUE THEN  
    SMS_COUNTER5:=SMS_COUNTER5+1;  
    IF SMS_COUNTER5=1 THEN  
SMS_ALERTA:='La temperatura es mas alta del valor normal';  
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,  
SMS_ALERTA);  
        ELSIF SMS_COUNTER5=3 THEN  
            SMS_COUNTER5:=2;  
        END_IF;  
    END_IF;  
  
IF TEMP_ALTA = FALSE THEN  
    SMS_COUNTER5:=0;  
END_IF;
```

```
IF TEMP_BAJA = TRUE THEN
    SMS_COUNTER6:=SMS_COUNTER6+1;
    IF SMS_COUNTER6=1 THEN
SMS_ALERTA:='La temperatura es mas baja del valor normal';
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
    ELSIF SMS_COUNTER6=3 THEN
        SMS_COUNTER6:=2;
    END_IF;
END_IF;

IF TEMP_BAJA = FALSE THEN
    SMS_COUNTER6:=0;
END_IF;

IF DESACTIVAR = TRUE THEN
    SMS_COUNTER7:=SMS_COUNTER7+1;
    IF SMS_COUNTER7=1 THEN
SMS_ALERTA:='El proceso se ha detenido';
        UTIL_SEND_SMS_6( TRUE, CONFIG_slotID, CONFIG_receiverList,
SMS_ALERTA);
    ELSIF SMS_COUNTER7=3 THEN
```



```

        SMS_COUNTER7:=2;
    END_IF;
END_IF;

IF DESACTIVAR = FALSE THEN
    SMS_COUNTER7:=0;
END_IF;

(*-----***** DO NOT MODIFY *****-----*)

IF UTIL_newSMSRecved AND MLEN(DATA_recvedSMSText)>=2 THEN

    UTIL_targetTagName := “;
    UTIL_targetTagValue := “;
    UTIL_tmpStr1:= “;
    UTIL_findTagName := FALSE;
    DATA_recvedSMSTxtLen := MLEN(DATA_recvedSMSText);
    FOR UTIL_pos:=1 TO DATA_recvedSMSTxtLen BY 1 DO
        UTIL_charValue := ASCII( DATA_recvedSMSText, UTIL_pos);
        IF UTIL_charValue <> 32 AND UTIL_charValue<>ASCII(‘=’, 1) AND
UTIL_pos<DATA_recvedSMSTxtLen THEN
            UTIL_tmpStr1 := INSERT(UTIL_tmpStr1, CHAR(UTIL_charValue),
MLEN(UTIL_tmpStr1)+1 );

```

```

ELSE
    IF UTIL_pos = DATA_recvedSMSTxtLen AND UTIL_charValue <> 32
AND UTIL_charValue<>ASCII( '=', 1) THEN
        UTIL_tmpStr1 := INSERT( UTIL_tmpStr1, CHAR(UTIL_charValue),
MLEN(UTIL_tmpStr1)+1);
            END_IF;
        IF NOT UTIL_findTagName AND MLEN(UTIL_tmpStr1)>0 THEN
            UTIL_findTagName := TRUE;
            UTIL_targetTagName := UTIL_tmpStr1;
            UTIL_tmpStr1 := "";
        ELSIF UTIL_findTagName AND MLEN(UTIL_tmpStr1)>0 THEN
            UTIL_targetTagValue := UTIL_tmpStr1;
            END_IF;
        END_IF;
    END_FOR;
    IF MLEN(UTIL_targetTagName)>0 AND MLEN(UTIL_targetTagValue)>0
THEN
        UTIL_setTagValueCmd := TRUE;
        UTIL_getTagValueCmd := FALSE;
        ELSIF MLEN(UTIL_targetTagName)>0 AND
MLEN(UTIL_targetTagValue)=0 THEN
            UTIL_setTagValueCmd := FALSE;

```

```

UTIL_getTagValueCmd := TRUE;
ELSE
UTIL_setTagValueCmd := FALSE;
UTIL_getTagValueCmd := FALSE;
UTIL_setCmdAcked := FALSE;
DATA_ackSMSText := “;
DATA_ackSMSText := ‘Invalid command received:’;
DATA_ackSMSText := INSERT(DATA_ackSMSText,
DATA_recvedSMSText, MLEN(DATA_ackSMSText)+1);
END_IF;
UTIL_newSMSRecved := FALSE;
END_IF;
IF UTIL_getTagValueCmd THEN
UTIL_tagName := “;
(*-----***** DO NOT MODIFY *****-----*)

(*-----*)
(* TIEMPO ENCENDIDO *)

IF FIND( UTIL_targetTagName, ‘TIEMPO ENCENDIDO’)>0 THEN

```

```

UTIL_tagName := '**NOTA: EL reloj del PLC se resetea luego de 20
dias** El tiempo que lleva encendido es: ';

UTIL_dateTimeInDINT := ANY_TO_DINT(__SYSVA_CYCLEDATE);

UTIL_curValue := UTIL_dateTimeInDINT -
(UTIL_dateTimeInDINT/1000)*1000;

    UTIL_tagValue := ANY_TO_STRING(UTIL_curValue);

    UTIL_dateTimeInDINT := UTIL_dateTimeInDINT/1000;

    UTIL_curValue := UTIL_dateTimeInDINT - (UTIL_dateTimeInDINT / 60 ) *
60;

    UTIL_tagValue := INSERT(UTIL_tagValue, 's', 1);

    UTIL_tagValue := INSERT(UTIL_tagValue,
ANY_TO_STRING(UTIL_curValue), 1);

    UTIL_dateTimeInDINT := UTIL_dateTimeInDINT / 60;

    UTIL_curValue := UTIL_dateTimeInDINT - UTIL_dateTimeInDINT / 60 *
60;

    UTIL_tagValue := INSERT(UTIL_tagValue, 'm', 1);

    UTIL_tagValue := INSERT(UTIL_tagValue,
ANY_TO_STRING(UTIL_curValue), 1);

    UTIL_dateTimeInDINT := UTIL_dateTimeInDINT / 60;

    UTIL_curValue := UTIL_dateTimeInDINT - UTIL_dateTimeInDINT / 24 *
24;

    UTIL_tagValue := INSERT(UTIL_tagValue, 'h', 1);

```

```

    UTIL_tagValue := INSERT(UTIL_tagValue,
ANY_TO_STRING(UTIL_curValue), 1);

    UTIL_curValue := UTIL_dateTimeInDINT / 24;
    UTIL_tagValue := INSERT(UTIL_tagValue, 'd', 1);
    UTIL_tagValue := INSERT(UTIL_tagValue,
ANY_TO_STRING(UTIL_curValue), 1);
(*-----*)

(*-----*)
(* EMPEZAR *)
(* Mensaje para comenzar el proceso *)

```

```

    ELSIF UTIL_targetTagName = 'EMPEZAR' THEN
        IF TEMPERATURA > TEMP_MAX THEN
            UTIL_tagName := 'La temperatura es muy alta, si desea continuar envie
CONTINUAR';

            ELSIF TEMPERATURA < TEMP_MIN THEN
                UTIL_tagName := 'La temperatura es muy baja, si desea continuar envie
CONTINUAR';

            ELSE
                ACTWEB := TRUE;
                UTIL_tagName := 'Se ha comenzado el proceso';

```

```

        END_IF;

        UTIL_tagValue := “;

(*-----*)

(* CONTINUAR *)

        ELSIF UTIL_targetTagName = ‘CONTINUAR’ THEN

ACTWEB := TRUE;

UTIL_tagName := ‘Se ha comenzado el proceso, acorde a su decision’;

UTIL_tagValue := “;

(*-----*)

(*TEMPERATURA *)

(*Consulta la temperatura del biodigestor para que el administrador lo sepa*)

(*Antes de empezar el proceso*)

        ELSIF UTIL_targetTagName = ‘TEMPERATURA’ THEN

UTIL_tagName := ‘La temperatura dentro del biodigestor es: ‘;

        UTIL_tagName :=

INSERT(UTIL_tagName,ANY_TO_STRING(TEMPERATURA),

MLEN(UTIL_tagName)+1);

        UTIL_tagValue := “;

```

(*-----*)

(* DETENER *)

ELSIF UTIL_targetTagName = 'DETENER' THEN

DESWEB:=TRUE;

(*-----*)

(* **PLCSTATUS request** *)

(* This section responds to a PLCSTATUS request by retrieving *)

(* controller status variable values, and building a response SMS *)

(* message to send out through the module. Delete this entire section *)

(* if it is not required by your application. *)

(* You can customize parts of this section as described in comments *)

(* within the section. *)

(* IMPORTANT: If you have deleted the previous request code section, *)

(* and this is the first request code section, then you need to change *)

(* the ELSIF in the next line to IF. *)

ELSIF UTIL_targetTagName = 'ESTADO' THEN

(*RUN_PROG_REM_FAULT*)

```

IF __SYSVA_MAJ_ERR_HALT =TRUE THEN
UTIL_tagName := 'ALARMADO, Falla existente, ';
END_IF;
(*DO: xxxx*)
(* This sample code retrieves the ON/OFF status of digital outputs *)
(* 0 and 1, and inserts them into the PLCSTATUS response SMS
message. *)
(* Delete or add digital outputs as required. *)
UTIL_tagName := INSERT(UTIL_tagName, 'SALIDAS, VALVULA DE
AGUA: ', MLEN(UTIL_tagName)+1);
(*Digital Output 0*)
IF _IO_EM_DO_00 = TRUE THEN
UTIL_tagName := INSERT(UTIL_tagName, 'Abierto',
MLEN(UTIL_tagName)+1);
ELSE
UTIL_tagName := INSERT(UTIL_tagName, 'Cerrado',
MLEN(UTIL_tagName)+1);
END_IF;
(*Digital Output 1*)
UTIL_tagName := INSERT(UTIL_tagName, ', VALVULA DE DESECHOS:
', MLEN(UTIL_tagName)+1);
IF _IO_EM_DO_01 = TRUE THEN

```



```
    UTIL_tagName := INSERT(UTIL_tagName, 'Abierto',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'Cerrado',
MLEN(UTIL_tagName)+1);

    END_IF;

    UTIL_tagName := INSERT(UTIL_tagName, ', ',
MLEN(UTIL_tagName)+1);

    (*Digital Output 2*)

    UTIL_tagName := INSERT(UTIL_tagName, ', VALVULA DE ACIDO: ',
MLEN(UTIL_tagName)+1);

    IF _IO_EM_DO_02 = TRUE THEN

    UTIL_tagName := INSERT(UTIL_tagName, 'Abierto',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'Cerrado',
MLEN(UTIL_tagName)+1);

    END_IF;

    UTIL_tagName := INSERT(UTIL_tagName, ', ',
MLEN(UTIL_tagName)+1);

    (*Digital Output 3*)
```

```
    UTIL_tagName := INSERT(UTIL_tagName, ' VALVULA DE ALCALINO: ',
MLEN(UTIL_tagName)+1);

    IF _IO_EM_DO_03 = TRUE THEN

    UTIL_tagName := INSERT(UTIL_tagName, 'Abierto',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'Cerrado',
MLEN(UTIL_tagName)+1);

    END_IF;

    UTIL_tagName := INSERT(UTIL_tagName, ' ',
MLEN(UTIL_tagName)+1);

(*DI: xxxx*)

(* This sample code retrieves the ON/OFF status of digital inputs *)

(* 0 and 1, and inserts them into the PLCSTATUS response SMS
message. *)

(* Delete or add digital inputs as required. *)

    UTIL_tagName := INSERT(UTIL_tagName, 'NIVEL DE LA MEZCLA EN
CARGA: ', MLEN(UTIL_tagName)+1);

(*Digital Input 1*)

    IF _IO_EM_DI_01 = TRUE THEN
```

```
    UTIL_tagName := INSERT(UTIL_tagName, 'LLENO',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'INFERIOR AL MAXIMO',
MLEN(UTIL_tagName)+1);

    END_IF;

    (*Digital Input 2*)

    UTIL_tagName := INSERT(UTIL_tagName, 'NIVEL TANQUE DE
DESECHOS: ', MLEN(UTIL_tagName)+1);

    IF _IO_EM_DI_02 = TRUE THEN

    UTIL_tagName := INSERT(UTIL_tagName, 'POR ARRIBA DEL MINIMO',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'DEBAJO DEL MINIMO',
MLEN(UTIL_tagName)+1);

    END_IF;

    UTIL_tagValue := "";

    (*-----*)
```

```
(*-----*)
(* MSGSTATUS request *)
(* This section responds to a MSGSTATUS request by retrieving SMS *)
(* message status variable values, and building a response SMS *)
(* message to send out through the module. Delete this entire section *)
(* if it is not required by your application. *)
(* IMPORTANT: If this is the first request code section, then you need *)
(* to change the ELSIF in the next line to IF. *)
```

```
ELSIF UTIL_targetTagName = 'ESTADO SMS' THEN
(*Msgs in*)
UTIL_tagName := 'Mensajes Recibidos: ';
UTIL_tagName := INSERT(UTIL_tagName,
ANY_TO_STRING(STATUS_SMSRecvCount), MLEN(UTIL_tagName)+1);
(*Msgs out*)
UTIL_tagName := INSERT(UTIL_tagName, ', Mensajes Enviados: ',
MLEN(UTIL_tagName)+1);
UTIL_tagName := INSERT(UTIL_tagName,
ANY_TO_STRING(STATUS_SMSSendCount), MLEN(UTIL_tagName)+1);
(*Failed msgs out*)
```

```

    UTIL_tagName := INSERT(UTIL_tagName, ' , Intentos fallidos de envio: ',
MLEN(UTIL_tagName)+1);

```

```

    UTIL_tagName := INSERT(UTIL_tagName,
ANY_TO_STRING(STATUS_SMSSendFailCount),
MLEN(UTIL_tagName)+1);

```

```

    UTIL_tagValue := "";

```

```

(*-----*)

```

```

(*-----*)

```

```

(* NETSTATUS request *)

```

```

(* This section responds to a NETSTATUS request by retrieving *)

```

```

(* network status variable values, and building a response SMS *)

```

```

(* message to send out through the module. Delete this entire section *)

```

```

(* if it is not required by your application. *)

```

```

(* IMPORTANT: If this is the first request code section, then you need *)

```

```

(* to change the ELSIF in the next line to IF. *)

```

```

    ELSIF UTIL_targetTagName = 'ESTADO CELULAR' THEN

```

```

        (*Signal strength*)

```

```
    UTIL_tagName := 'Potencia de la señal (dBm): ';
```

```
    UTIL_tagName := INSERT(UTIL_tagName,  
ANY_TO_STRING(STATUS_signalStrength), MLEN(UTIL_tagName)+1);
```

```
    UTIL_tagName := INSERT(UTIL_tagName, ' ',  
MLEN(UTIL_tagName)+1);
```

```
    (*Network status*)
```

```
    UTIL_tagName := INSERT(UTIL_tagName, 'Estado de Red: ',  
MLEN(UTIL_tagName)+1);
```

```
    IF STATUS_networkStatus = 48 THEN
```

```
        UTIL_tagName := INSERT(UTIL_tagName, 'No Registrado,Buscando, ',  
MLEN(UTIL_tagName)+1);
```

```
    ELSIF STATUS_networkStatus = 49 THEN
```

```
        UTIL_tagName := INSERT(UTIL_tagName, 'Registrado, ',  
MLEN(UTIL_tagName)+1);
```

```
    ELSIF STATUS_networkStatus = 50 THEN
```

```
        UTIL_tagName := INSERT(UTIL_tagName, 'No Registrado, ',  
MLEN(UTIL_tagName)+1);
```

```
    ELSIF STATUS_networkStatus = 51 THEN
```

```
        UTIL_tagName := INSERT(UTIL_tagName, 'Registro Denegado, ',  
MLEN(UTIL_tagName)+1);
```

```
    ELSIF STATUS_networkStatus = 52 THEN
```

```

    UTIL_tagName := INSERT(UTIL_tagName, 'Desconocido, ',
MLEN(UTIL_tagName)+1);

    ELSIF STATUS_networkStatus = 53 THEN

    UTIL_tagName := INSERT(UTIL_tagName, 'Registrado: Roaming, ',
MLEN(UTIL_tagName)+1);

    ELSE

    UTIL_tagName := INSERT(UTIL_tagName, 'Error, ',
MLEN(UTIL_tagName)+1);

    END_IF;

    (*Connection failures*)

    UTIL_tagName := INSERT(UTIL_tagName, 'Fallas de Conexion: ',
MLEN(UTIL_tagName)+1);

    UTIL_tagName := INSERT(UTIL_tagName,
ANY_TO_STRING(STATUS_netConnFailedRetries),
MLEN(UTIL_tagName)+1);

    UTIL_tagValue := "";

    (*-----*)

```

```

(*-----***** DO NOT MODIFY *****-----*)

    ELSE

DATA_ackSMSText := 'FALLO: comando invalido';

    END_IF;

    IF MLEN(UTIL_tagName)>0 THEN

DATA_sentSMSText := UTIL_tagName;

    DATA_sentSMSText := INSERT( DATA_sentSMSText, ' ',
MLEN(DATA_sentSMSText)+1);

    DATA_sentSMSText := INSERT( DATA_sentSMSText, UTIL_tagValue,
MLEN(DATA_sentSMSText)+1);

    ELSE

    DATA_sentSMSText := DATA_ackSMSText;

END_IF;

UTIL_succeed := FALSE;

UTIL_getTagValueCmd := FALSE;

END_IF;

IF NOT UTIL_succeed THEN

    UTIL_ackReceivers[1] := UTIL_sender;

    UTIL_ackReceivers[2] := “;

    UTIL_ackReceivers[3] := “;

    UTIL_ackReceivers[4] := “;

    UTIL_ackReceivers[5] := “;

```



```

    UTIL_SEND_SMS_3( TRUE, CONFIG_slotID, UTIL_ackReceivers,
DATA_sentSMSText);

    IF UTIL_SEND_SMS_3.Q AND UTIL_SEND_SMS_3.ErrCode = 0 THEN
        UTIL_succeed := TRUE;
    ELSE
        UTIL_succeed := FALSE;
        UTIL_retryCount := UTIL_retryCount + 1;
        IF UTIL_retryCount > UTIL_maxRetryTime THEN
            UTIL_retryCount := 0;
            UTIL_succeed := TRUE;
            STATUS_errCount := STATUS_errCount + 1;
        END_IF;
    END_IF;
END_IF;

IF UTIL_setTagValueCmd THEN
    UTIL_setTagValueCmd := FALSE;
    DATA_ackSMSText := 'OK: Set ';
    DATA_ackSMSText := INSERT(DATA_ackSMSText,
UTIL_targetTagName, MLEN(DATA_ackSMSText)+1 );
    DATA_ackSMSText := INSERT(DATA_ackSMSText, '=',
MLEN(DATA_ackSMSText)+1 );

```

```

DATA_ackSMSText := INSERT(DATA_ackSMSText,
UTIL_targetTagValue, MLEN(DATA_ackSMSText)+1 );
(*-----***** DO NOT MODIFY *****-----*)

(*-----*)
(* ADD/DELETE phone number commands *)
(* This section responds to SMS message commands to ADD or DELETE a
*)
(* phone number. A new phone number is always added to the 5th element
*)
(* of the Receiver List array. *)
(* Delete this entire section if it is not required by your application. *)

(*ADD*)
IF UTIL_targetTagName = 'ADD' THEN
    CONFIG_receiverList[5] := UTIL_targetTagValue;
    DATA_ackSMSText := 'OK: ';
    DATA_ackSMSText := INSERT( DATA_ackSMSText,
UTIL_targetTagValue, MLEN(DATA_ackSMSText)+1);

```

```

DATA_ackSMSText := INSERT( DATA_ackSMSText, ' added',
MLEN(DATA_ackSMSText)+1);

(*DELETE*)

ELSIF UTIL_targetTagName = 'DELETE' THEN

    DATA_ackSMSText := 'FAILED: ';

    DATA_ackSMSText := INSERT( DATA_ackSMSText,
UTIL_targetTagValue, MLEN(DATA_ackSMSText)+1);

    DATA_ackSMSText := INSERT( DATA_ackSMSText, ' not found',
MLEN(DATA_ackSMSText)+1);

    FOR UTIL_pos := 1 TO 5 BY 1 DO

        IF FIND(CONFIG_receiverList[UTIL_pos], UTIL_targetTagValue) = 1
THEN
            CONFIG_receiverList[UTIL_pos] := "";

            DATA_ackSMSText := 'OK: ';

            DATA_ackSMSText := INSERT( DATA_ackSMSText,
UTIL_targetTagValue, MLEN(DATA_ackSMSText)+1);

            DATA_ackSMSText := INSERT( DATA_ackSMSText, ' deleted',
MLEN(DATA_ackSMSText)+1);

            END_IF;

        END_FOR;

(*-----*)

```

(*-----*)

(* **WHITELIST ON/OFF command** *)

(* This section responds to SMS message commands to enable or disable
*)

(* Whitelist filtering of incoming SMS messages. *)

(* Delete this entire section if it is not required by your application. *)

(* IMPORTANT: If this is the first command code section, then you need *)

(* to change the ELSIF in the next line to IF. *)

ELSIF UTIL_targetTagName = 'WHITELIST' THEN

IF UTIL_targetTagValue = 'OFF' OR UTIL_targetTagValue = '0' THEN

 CONFIG_whiteListEnabled := FALSE;

ELSIF UTIL_targetTagValue = 'ON' or UTIL_targetTagValue = '1' THEN

 CONFIG_whiteListEnabled := TRUE;

ELSE

 DATA_ackSMSText := UTIL_invalidTagValue;

END_IF;

(*-----*)

```
(*-----*)
(* OUTPUTx ON/OFF command *)
(* This section responds to SMS message commands to turn digital *)
(* outputs ON or OFF. *)
(* This sample code handles digital outputs 0 and 1. *)
(* Delete or add digital outputs as required. *)
(* IMPORTANT: If this is the first command code section, then you need *)
(* to change the ELSIF in the next line to IF. *)
```

```
(*Digital Output 0*)
```

```
ELSIF UTIL_targetTagName = 'OUTPUT0' OR UTIL_targetTagName =
```

```
'00' THEN
```

```
IF UTIL_targetTagValue = 'OFF' OR UTIL_targetTagValue = '0' THEN
```

```
_IO_EM_DO_00 := FALSE;
```

```
ELSIF UTIL_targetTagValue = 'ON' or UTIL_targetTagValue = '1' THEN
```

```
_IO_EM_DO_00 := TRUE;
```

```
ELSE
```

```
DATA_ackSMSText := UTIL_invalidTagValue;
```

```
END_IF;
```

```
(*Digital Output 1*)
```

```
ELSIF UTIL_targetTagName = 'OUTPUT1' OR UTIL_targetTagName =  
  'O1' THEN
```

```
IF UTIL_targetTagValue = 'OFF' OR UTIL_targetTagValue = '0' THEN
```

```
  _IO_EM_DO_01 := FALSE;
```

```
ELSIF UTIL_targetTagValue = 'ON' or UTIL_targetTagValue = '1' THEN
```

```
  _IO_EM_DO_01 := TRUE;
```

```
ELSE
```

```
DATA_ackSMSText := UTIL_invalidTagValue;
```

```
END_IF;
```

```
(*-----*)
```

```
(*-----***** DO NOT MODIFY *****-----*)
```

```
ELSE
```

```
DATA_ackSMSText := 'FAILED: Invalid command';
```

```
END_IF;
```

```
        UTIL_setCmdAcked := FALSE;
END_IF;
IF (NOT CONFIG_ackEnabled) THEN
    UTIL_setCmdAcked := True;
    UTIL_retryCount := 0;
    return;
ELSE
    IF NOT UTIL_setCmdAcked AND MLEN(DATA_ackSMSText)>0 AND
MLEN(UTIL_sender)>0 THEN
        UTIL_ackReceivers[1] := UTIL_sender;
        UTIL_ackReceivers[2] := “;
        UTIL_ackReceivers[3] := “;
        UTIL_ackReceivers[4] := “;
        UTIL_ackReceivers[5] := “;
        UTIL_SEND_SMS_4( NOT UTIL_setCmdAcked, CONFIG_slotID,
UTIL_ackReceivers, DATA_ackSMSText);
        IF UTIL_SEND_SMS_4.Q AND UTIL_SEND_SMS_4.ErrCode=0 THEN
            UTIL_setCmdAcked := TRUE;
            UTIL_retryCount := 0;
        ELSE
            UTIL_retryCount := UTIL_retryCount + 1;
            IF UTIL_retryCount > UTIL_maxRetryTime THEN
```

```

STATUS_errCount := STATUS_errCount+1;

UTIL_setCmdAcked := TRUE;

DATA_ackSMSText := “;

UTIL_retryCount := 0;

    END_IF;

    END_IF;

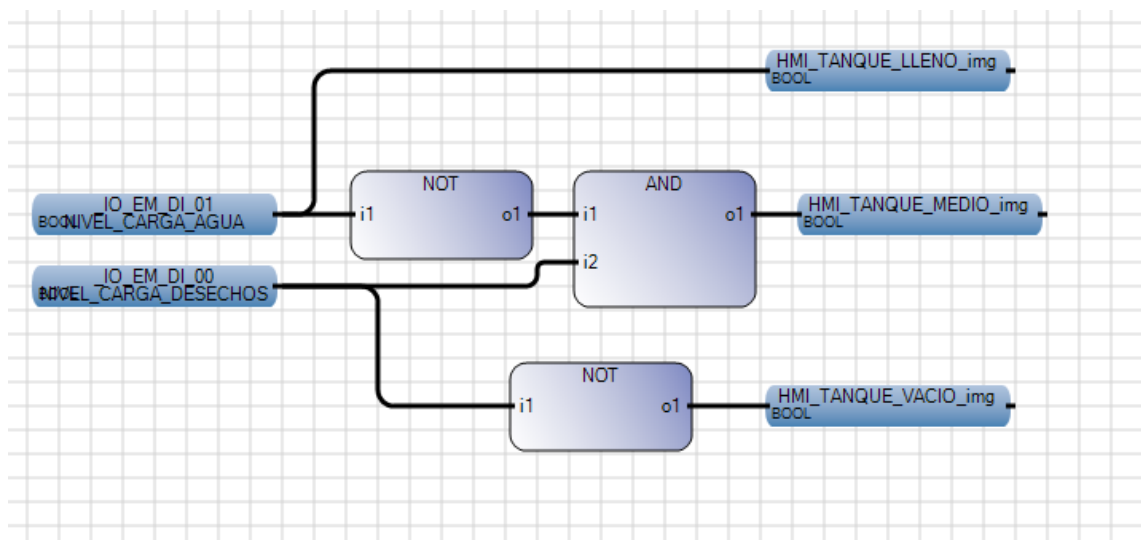
END_IF;

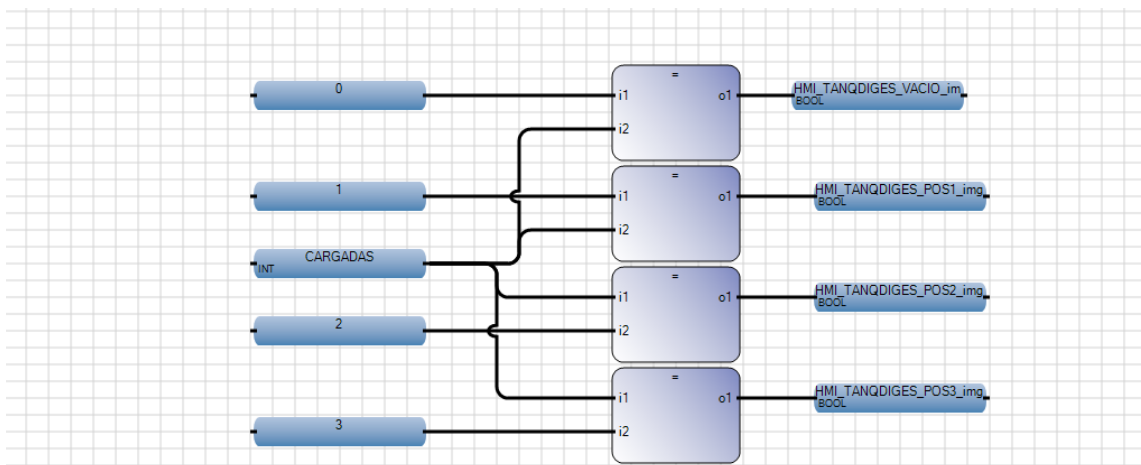
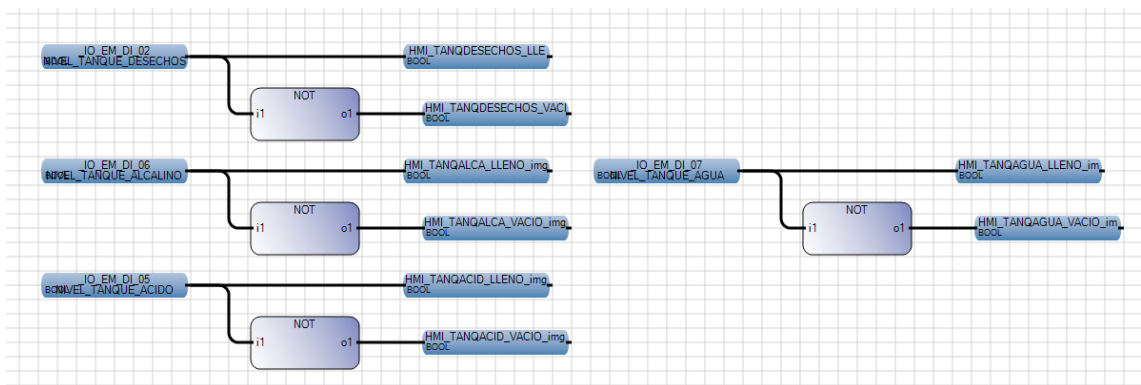
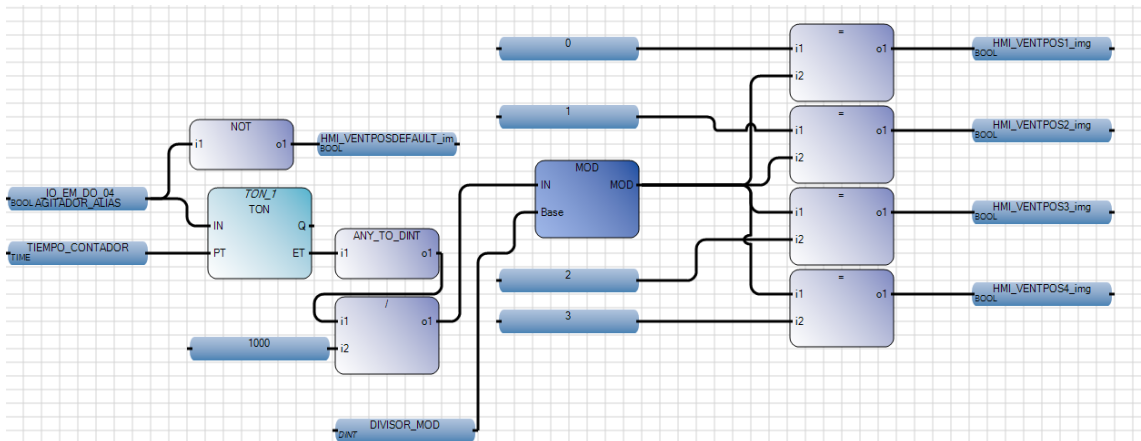
END_IF;

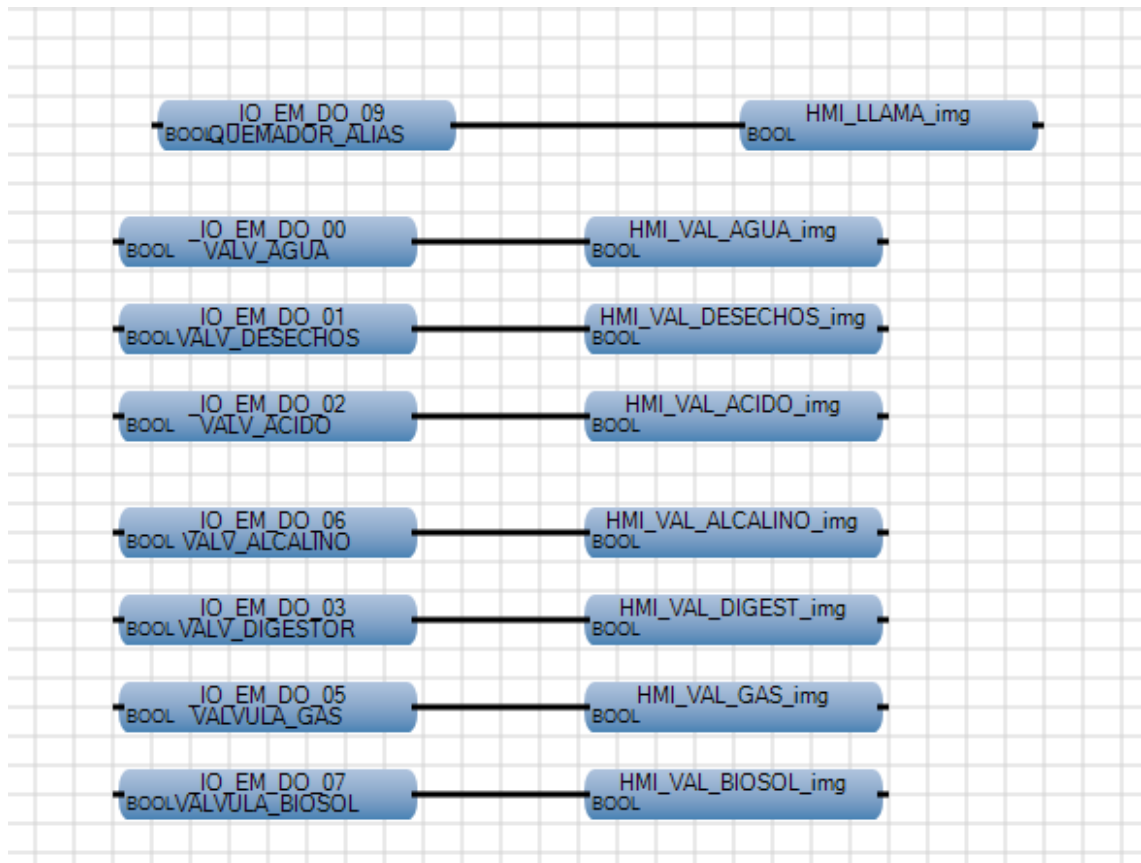
(*-----***** DO NOT MODIFY *****-----*)

```

ANEXO D: ADAPTACIÓN DE LA PANTALLA AL CONTROLADOR.







ANEXO E: PROGRAMACIÓN ETAPA DE CARGA.

```
IF MANUAL=FALSE AND EMERGENCIA=FALSE AND PARAR=FALSE
```

```
THEN
```

```
VALVULA_AGUA :=FALSE;
```

```
VALVULA_DESECHOS :=FALSE;
```

```
VALVULA_DIGESTOR :=FALSE;
```

```
VALVULA_ACIDO := FALSE;
```

```
VALVULA_ALCALINO := FALSE;
```

SMS_FINAL_CARGADA:=FALSE;

(*Cerrando todas las valvulas de los tanques en un inicio*)

IF CARGADAS<>MAX_CARGADAS THEN

 IF MARCHAR = TRUE THEN

 (*Proceso de llenado del tanque de carga SOLO si hay nivel de desechos adecuado*)

 (*Recordar que minimo de desechos es verdadero mientras este en contacto con el agua *)

 IF MINIMO_DESECHOS=TRUE THEN

 ALERTA_DESECHOS:=FALSE;

 IF MINIMO_AGUA=TRUE THEN

 ALERTA_AGUA:=FALSE;

 IF EXPULSAR=FALSE THEN

 IF NIVEL_AGUA=FALSE THEN

 IF NIVEL_DESECHOS=FALSE THEN

 VALVULA_DESECHOS:=TRUE;

 ELSIF NIVEL_DESECHOS=TRUE THEN

 VALVULA_AGUA:=TRUE;

 END_IF;

 ELSIF NIVEL_AGUA=TRUE THEN

 IF CONTADOR.Q=FALSE THEN

 CONTAR:=TRUE;

```
CONTADOR(CONTAR,T#60s);
AGITADOR:=TRUE;
TIEMPO_AGITA:=ANY_TO_STRING(CONTADOR.ET);
ELSIF CONTADOR.Q=TRUE THEN
AGITADOR:=FALSE;
IF PH>=PHMAX THEN
(*Se envia alerta y se para el proceso hasta que lo reinicie*)
IF NIVEL_ACIDO=FALSE THEN
ALERTA:='El nivel de pH es bajo en el tanque de acido';
ALERTA_ACIDO:=TRUE;
ELSE
VALVULA_ACIDO:=TRUE;
AGITADOR:=TRUE;
ALERTA_ACIDO:=FALSE;
END_IF;
ELSIF PH<=PHMIN THEN
(*Se envia alerta y se para el proceso hasta que lo reinicie*)
IF NIVEL_ALCALINO=FALSE THEN
ALERTA:='El nivel de pH es bajo en el tanque alcalino';
ALERTA_ALCALINO:=TRUE;
ELSE
VALVULA_ALCALINO:=TRUE;
```

```
AGITADOR:=TRUE;
ALERTA_ALCALINO:=FALSE;
    END_IF;
ELSE
VALVULA_ACIDO:=FALSE;
VALVULA_ALCALINO:=FALSE;
    AGITADOR:=FALSE;
    EXPULSAR:=TRUE;
END_IF;
END_IF;
END_IF;
END_IF;

IF CARGA_VACIA=TRUE AND EXPULSAR=TRUE THEN
    VALVULA_DIGESTOR:=TRUE;
END_IF;

IF CARGA_VACIA=FALSE AND EXPULSAR=TRUE THEN
    VALVULA_DIGESTOR:=FALSE;
    VALVULA_ACIDO:=FALSE;
    VALVULA_ALCALINO:=FALSE;
    VALVULA_AGUA:=FALSE;
    VALVULA_DESECHOS:=FALSE;
```

```
EXPULSAR:=FALSE;

CARGADAS:=CARGADAS+1;

CONTAR:=FALSE;

CONTADOR(CONTAR,T#60s);

SMS_FINAL_CARGADA:=TRUE;

END_IF;

ELSIF MINIMO_AGUA=FALSE THEN

ALERTA:='El nivel del tanque de agua es bajo, se ha detenido el
proceso';

ALERTA_AGUA:=TRUE;

END_IF;

(*Si no hay mas del minimo de desechos activar la alerta por SMS y se
detiene el proceso*)

ELSIF MINIMO_DESECHOS=FALSE THEN

ALERTA:='El nivel del tanque de desechos es bajo, se ha detenido el
proceso';

ALERTA_DESECHOS:=TRUE;

END_IF;

END_IF;

END_IF;
```

ANEXO F: PROGRAMACIÓN ETAPA DIGESTIÓN.

```

IF MANUAL=FALSE AND EMERGENCIA=FALSE AND PARAR=FALSE
THEN

VALV_BIOSOL:=FALSE;

VALV_GAS:=FALSE;

QUEMADOR:=FALSE;

COMPRESOR:=FALSE;

ALERTA_PRESION_MAX:=FALSE;

ALERTA_PRESION_MIN:=FALSE;

    IF CARGADAS=MAX_CARGADAS THEN

IF CONTADOR.Q=FALSE THEN

                                CONTAR:=TRUE;
                                CONTADOR(CONTAR,T#60s);
                                COMPRESOR:=TRUE;
                                IF PRESION<=0.0 THEN

QUEMADOR:=TRUE;

VALV_GAS:=TRUE;

ALERTA:='El sensor de presion marca negativo';

ALERTA_PRESION_MIN:=TRUE;

                                ELSIF PRESION>=PRESION_MAX
                                THEN

VALV_GAS:=TRUE;

    ALERTA:='La presion en la bolsa es muy alta, se expulsa el gas';

        ALERTA_PRESION_MAX:=TRUE;

    ELSE

        ALERTA_PRESION_MAX:=FALSE;

```

```
ALERTA_PRESION_MIN:=FALSE;
END_IF;
IF TEMPERATURA>=TEMP_MAX THEN
TEMP_ALTA:=TRUE;
ELSIF TEMPERATURA<=TEMP_MIN THEN
TEMP_BAJA:=TRUE;
ELSE
TEMP_ALTA:=FALSE;
TEMP_BAJA:=FALSE;
END_IF;
ELSIF CONTADOR.Q=TRUE THEN
VALV_GAS:=TRUE;
COMPRESOR:=TRUE;
END_IF;
END_IF;

END_IF;
```

ANEXO G: MAPEO DE PERIFÉRICOS Y CONFIGURACIONES MARCHA Y PARO.

(*Mapeo Salidas*)

_IO_EM_DO_00 := VALVULA_AGUA;
_IO_EM_DO_01 := VALVULA_DESECHOS;
_IO_EM_DO_02 := VALVULA_ACIDO;
_IO_EM_DO_03 := VALVULA_DIGESTOR;
_IO_EM_DO_04 := AGITADOR;
_IO_EM_DO_05 := VALV_GAS;
_IO_EM_DO_06 := VALVULA_ALCALINO;
_IO_EM_DO_07 := VALV_BIOSOL;
_IO_EM_DO_08 := COMPRESOR;
_IO_EM_DO_09 := QUEMADOR;

(* Mapeo Entradas *)

NIVEL_DESECHOS := _IO_EM_DI_00;
NIVEL_AGUA := _IO_EM_DI_01;
MINIMO_DESECHOS := _IO_EM_DI_02;
CARGA_VACIA := _IO_EM_DI_03;
BIOD_VACIO := _IO_EM_DI_04;
NIVEL_ACIDO:= _IO_EM_DI_05;
NIVEL_ALCALINO := _IO_EM_DI_06;
MINIMO_AGUA := _IO_EM_DI_07;
ACTIVAR := _IO_EM_DI_08 OR ACTWEB;
DESACTIVAR := _IO_EM_DI_09;

(* Asignacion desde Web*)

TIEMPO_DIG := ANY_TO_TIME(DIGESTION_TIEMPO);

IF ACTIVAR=TRUE THEN

 MARCHAR := TRUE;

 EMERGENCIA:=FALSE;

 PARAR:=FALSE;

 CONT_DET:=0;

 CONTWEB:=0;

END_IF;

IF DESACTIVAR = FALSE THEN

 CAMBDET:=TRUE;

END_IF;

IF DESACTIVAR = TRUE AND CAMBDET =TRUE THEN

 CONT_DET:=CONT_DET+1;

 CAMBDET:=FALSE;

END_IF;

IF DESWEB = TRUE THEN

 CONTWEB:=CONTWEB+1;

 DESWEB:=FALSE;

END_IF;

IF ACTWEB = TRUE THEN

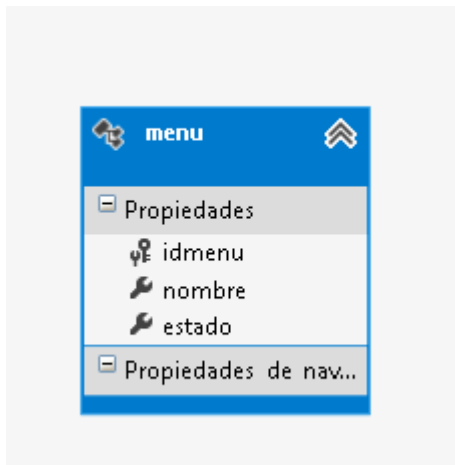
 ACTWEB:=FALSE;

```
END_IF;
IF CONT_DET = 0 THEN
    IF CONTWEB = 0 THEN
        EMERGENCIA:=FALSE;
        PARAR:=FALSE;
        ELSIF CONTWEB = 1 THEN
            EMERGENCIA:=FALSE;
            PARAR:=TRUE;
            MARCHAR:=FALSE;
            ELSIF CONTWEB = 2 THEN
                EMERGENCIA:=TRUE;
                PARAR:=FALSE;
                MARCHAR:=FALSE;
            END_IF;
        ELSIF CONT_DET = 1 THEN
            IF CONTWEB = 0 THEN
                EMERGENCIA:=FALSE;
                PARAR:=TRUE;
                MARCHAR:=FALSE;
                ELSIF CONTWEB = 1 THEN
                    EMERGENCIA:=FALSE;
                    PARAR:=TRUE;
```

```
MARCHAR:=FALSE;
    ELSIF CONTWEB = 2 THEN
EMERGENCIA:=TRUE;
PARAR:=FALSE;
MARCHAR:=FALSE;
    END_IF;
ELSIF CONT_DET = 2 THEN
    EMERGENCIA:=TRUE;
    PARAR:=FALSE;
    MARCHAR:=FALSE;
END_IF;
IF EMERGENCIA=TRUE THEN
    VALVULA_AGUA:=FALSE;
    VALVULA_DESECHOS:=FALSE;
    VALVULA_ACIDO:=FALSE;
    VALVULA_DIGESTOR:=TRUE;
    AGITADOR:=FALSE;
    VALV_GAS:=TRUE;
    VALVULA_ALCALINO:=FALSE;
    VALV_BIOSOL:=TRUE;
    COMPRESOR:=FALSE;
    QUEMADOR:=TRUE;
```

```
        MANUAL:=FALSE;
        MARCHAR:=FALSE;
END_IF;
IF PARAR=TRUE THEN
    VALVULA_AGUA:=FALSE;
    VALVULA_DESECHOS:=FALSE;
    VALVULA_ACIDO:=FALSE;
    VALVULA_DIGESTOR:=FALSE;
    AGITADOR:=FALSE;
    VALV_GAS:=FALSE;
    VALVULA_ALCALINO:=FALSE;
    VALV_BIOSOL:=FALSE;
    COMPRESOR:=FALSE;
    QUEMADOR:=FALSE;
    MANUAL:=FALSE;
    MARCHAR:=FALSE;
    EMERGENCIA:=FALSE;
END_IF;
IF MANUAL=TRUE THEN
    MARCHAR:=FALSE;
END_IF;
```

ANEXO H: MODELO DE LA APLICACIÓN WEB.



ANEXO I: VISTA PÁGINA PRINCIPAL DE LA APLICACIÓN WEB.

```
@using telecontrol.Models;
```

```
@model List<menu>
```

```
@{  
    ViewBag.Title = "Telecontrol de Biodigestor";
```

```

}
<h1 class="titulo">Biodigestor</h1>

<div id="izqPrinc">

    @foreach (menu item in Model)

    {
        if(item.nombre.Equals("Tanque"))

        {
            using (Html.BeginForm("Tanques", "Proceso", FormMethod.Post, "1"))

            {
                <button class="botones" type="submit"> Sistema de Tanques </button>
            }
        }

        if (item.nombre.Equals("Carga"))

        {
            using (Html.BeginForm("Carga", "Proceso", FormMethod.Post, "1"))

            {
                <button class="botones" type="submit"> Etapa de Carga </button>
            }
        }

        if (item.nombre.Equals("Digestion"))

        {
            using (Html.BeginForm("Digestion", "Proceso", FormMethod.Post, "1"))

            {
                <button class="botones" type="submit"> Etapa de Digestion </button>
            }
        }

        if (item.nombre.Equals("Variables"))

        {
            using (Html.BeginForm("Variables", "Proceso", FormMethod.Post, "1"))

            {

```

```

<button class="botones" type="submit"> Monitoreo de Variables </button>
    }
}
    if (item.nombre.Equals("Usuarios"))
    {
        using (Html.BeginForm("Index", "Usuario", FormMethod.Get, null))
        {
            <button class="botones" type="submit"> Manejo de Usuarios </button>
        }
    }
}
</div>

```

@foreach (menu item in Model)

```

{
    if(item.nombre.Equals("Controlar"))
    {
<div id="derPrinc">

        <div class="boton"> <div class="botonCircular" id="botonAzul"><a
class="boton" onclick="enviarActivar()">MARCHA</a></div>

        </div>

        <br />

        <div class="boton">

            <div class="botonCircular" id="botonRojo"><a href="#" class="boton"
onclick="enviarDetener()">PARO</a></div>

        </div>

</div>

```



```
}
}
```

ANEXO J: VISTA ETAPA DE CARGA DE LA APLICACIÓN WEB.

```
@using telecontrol.Models;
```

```
@model List<menu>
```

```
@{
    ViewBag.Title = "Sistema de Tanques";
```

```
}
<h1 class="titulo">Etapa de Carga</h1>
```

```
<div id="etiquetasCarga">
```

```
@foreach (menu item in Model)
```

```
{
    if(item.nombre.Equals("Controlar"))
```

```
{
    <div class="botonesProc">
```

```
<button class="activar" onclick="enviarActivar()">Activar</button>
```

```
<button class="detener" onclick="enviarDetener()">Detener</button>
```

```
<button class="manual" onclick="enviarManual()">Manual</button>
```

```
</div>
```

```
}
}
<label>Contenedor</label>
```

```
<label>Agitador</label>
```

```

</div>

<div id="graficosCarga">
    
    
</div>



```

ANEXO K: VISTA SISTEMA DE TANQUES DE LA APLICACIÓN WEB.

```

@using telecontrol.Models;

@model List<menu>

@{
    ViewBag.Title = "Sistema de Tanques";
}

<h1 class="titulo">Tanques</h1>

@foreach (menu item in Model)
{
    if(item.nombre.Equals("Controlar"))
    {
        <div class="botonesProc">

<button class="activar" onclick="enviarActivar()">Activar</button>

<button class="detener" onclick="enviarDetener()">Detener</button>

```

```

        <button class="manual" onclick="enviarManual()">Manual</button>
    </div>

}
}
<div id="etiquetasTanques">

    <label>Alcalino</label>

    <label>Agua</label>

    <label>Desechos</label>

    <label>Acido</label>

</div>

<div id="lostanques">

</div>

<div id="valvulas">

```

```

    
    
    
</div>

```

ANEXO L: VISTA ETAPA DE DIGESTIÓN DE LA APLICACIÓN WEB.

```

@using telecontrol.Models;
@model List<menu>
@{
    ViewBag.Title = "Etapa de Digestion";
}
<h1 class="titulo">Etapa de Digestion</h1>
@foreach (menu item in Model)
{
    if(item.nombre.Equals("Controlar"))
    {
        <div class="botonesProc">
            <button class="activar" onclick="enviarActivar()">Activar</button>
        </div>
    }
}

```

```

<button class="detener" onclick="enviarDetener()">Detener</button>
    <button class="manual" onclick="enviarManual()">Manual</button>
</div>

}
}
<div id="etiquetasDig">

    <label id="tem">Tiempo de Digestion:</label>

    <label id="quem">Quemador</label>

</div>

<div id="graf0" >

    <p id="tiempoDig">30:00</p>

</div>

<div id="graf1">

    <img id="llama" src="" />

</div>

<div>

    <img id="gas" src="" />

</div>

<div id="graf2">

```

```

<img id="aire" src="" />







</div>

```

ANEXO M: VISTA MONITOREO DE VARIABLES DE LA APLICACIÓN WEB.

```

@using telecontrol.Models;

@model List<menu>

@{
    ViewBag.Title = "Monitoreo de Variables";
}
<h1 class="titulo">Variables</h1>

@foreach (menu item in Model)
{
    if(item.nombre.Equals("Controlar"))
    {
        <div class="botonesProc">

<button class="activar" onclick="enviarActivar()">Activar</button>

<button class="detener" onclick="enviarDetener()">Detener</button>

        <button class="manual" onclick="enviarManual()">Manual</button>

        </div>

```

```

    }
}
<div id="etiquetasVar">

    <label id="presLab">Presion</label>

    <label id="temLab">Temperatura</label>

    <label id="phLab">PH</label>

</div>

<div id="grafVar">

    <label id="minPres">0 psi</label>

    <meter id="metPres" value="14.7" min="0" max="20"></meter>

    <label id="mpm">20 psi</label>

    <label id="minTemp">10°C</label>

    <meter id="metTemp" value="30" min="10" max="45"></meter>

    <label id="mtm">45°C</label>

    <label id="minPH">1</label>

    <meter id="metPH" value="7" min="1" max="14" ></meter>

    <label>14</label>

</div>

<div id="valVar">

    <p id="presion">14.7</p>

    <p id="temperatura">30</p>

    <p id="ph">7</p>

```

```
</div>
```

```
@foreach (menu item in Model)
```

```
{
    if(item.nombre.Equals("Controlar"))
```

```
{
```

```
<div id="varini">
```

```
    <label class="centrado2">Variables Iniciales </label><br />
```

```
<br />
```

```
    <div class="centrado2">
```

```
        <label class="jts">Temperatura Máxima: </label><p class="jts"
```

```
id="tmaxima"></p><br/>
```

```
</div>
```

```
<div class="cambiar">
```

```
    <input type="number" name="tempmaxima" id="tempmax" min="20"
```

```
max="50" step="0.5" />
```

```
    <button onclick="enviarTempmax()">Cambiar</button><p
```

```
id="a1"></p><br />
```

```
</div>
```

```
<div class="centrado2">
```

```
    <label class="jts">Temperatura Mínima: </label><p class="jts"
```

```
id="tminima"></p><br/>
```

```
</div>
```

```
    <div class="cambiar">
```



```

<input type="number" name="tempminima" id="tempmin" min="0"
max="25" step="0.5" />

<button onclick="enviarTempmin()">Cambiar</button><p
id="a2"></p><br />

</div>

<div class="centrado2">

<label class="jts">pH Máximo: </label><p class="jts" id="phmx"></p>

</div>

<div class="cambiar">

<input type="number" name="phmax" id="phmax" min="7" max="14"
step="1" />

<button onclick="enviarpHmax()">Cambiar</button><p
id="a3"></p><br />

</div>

<div class="centrado2">

<label class="jts">pH Mínimo: </label><p class="jts" id="phmn"></p>

</div>

<div class="cambiar">

<input type="number" name="phmin" id="phmin" min="1" max="7"
step="1" />

<button onclick="enviarpHmin()">Cambiar</button><p id="a4"></p><br />

</div>

```

```

<div class="centrado2">

    <label class="jts">Presión Máxima: </label><p class="jts"
id="presimax"></p><br/>

    </div>

    <div class="cambiar">

        <input type="number" name="presionmax" id="pmax" min="14.7"
step="0.1" />

        <button onclick="enviarpresmax()">Cambiar</button><p
id="a5"></p><br />

    </div>

</div>

}
}

```

ANEXO N: VISTA CREAR USUARIO DE LA APLICACIÓN WEB.

```

@model telecontrol.Models.usuario

@{
    ViewBag.Title = "Crear Usuario";
}
<h1 class="titulo">Crear Usuario</h1>

```

```

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
type="text/javascript"></script>

<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>

@using (Html.BeginForm()) {

    @Html.ValidationSummary(true)

    <div class="creacion">

        <div class="volver">

            @Html.ActionLink("Lista de Usuarios", "Index")

            </div>

            <div class="editor-field">

                <label>Usuario (Login): </label>

                @Html.EditorFor(model => model.login)

                @Html.ValidationMessageFor(model => model.login)

            </div>

            <div class="editor-field">

                <label>Nombre: </label>

                @Html.EditorFor(model => model.nombre)

                @Html.ValidationMessageFor(model => model.nombre)

            </div>

            <div class="editor-field">

                <label>Apellido: </label>

```

```

    @Html.EditorFor(model => model.apellido)

    @Html.ValidationMessageFor(model => model.apellido)

</div>

<div class="editor-field">

<label>Contraseña: </label>

    @Html.EditorFor(model => model.contrasena)

    @Html.ValidationMessageFor(model => model.contrasena)

</div>

    @Html.HiddenFor(model => model.estado, "A")

    @* <div class="editor-field">

        <label class="centrado">Estado: </label><br /><br />

        @Html.RadioButtonFor(model => model.estado, "A") <span>
Activo</span>

        @Html.RadioButtonFor(model => model.estado, "I") <span>
Inactivo</span>

        @Html.ValidationMessageFor(model => model.estado)

</div>*@

<div class="editor-field">

<label class="centrado"> Tipo de Usuario: </label><br /><br />

<input type="radio" value="0" id="admin" /><span>Administrador</span>

    <input type="radio" value="1" id="est"/><span>Estandar</span>

</div>

```

```

<div class="editor-field">
    <label>Elija las vistas a las que tiene acceso: </label><br />
    <div id="checks">
        <br />
        <input type="checkbox" value="1" id="st" /><label>Sistema de
Tanques</label><br />
        <input type="checkbox" value="2" id="ec" /><label>Etapa de
Carga</label><br />
        <input type="checkbox" value="4" id="ed" /><label>Etapa de
Digestion</label><br />
        <input type="checkbox" value="8" id="mv" /><label>Monitoreo de
Variables</label>
    </div>
    @Html.HiddenFor(model => model.perfil, new { id = "tipoPerfil"})
    @Html.ValidationMessageFor(model => model.perfil)
</div>
<p>
    <input type="submit" value="Crear Usuario" class="inses"/>
</p>
</div>
}

```

ANEXO O: VISTA INGRESO AL SISTEMA DE LA APLICACIÓN WEB.

```
@model telecontrol.Models.usuario
```

```
@{
```

```
Layout = "";
```

```
ViewBag.Title = "Ingreso al Sistema";
```

```
}
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" name="viewport" content="width=device-width;
```

```
initial-scale=1.0;" />
```

```
<title>@ViewBag.Title</title>
```

```
<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet"
```

```
type="text/css" />
```

```
<script src="@Url.Content("~/Scripts/jquery-1.7.1.min.js")"
```

```
type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
```

```
type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
```

```
type="text/javascript"></script>
```

```

<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.js")"
type="text/javascript"></script>

<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")"
type="text/javascript"></script>

<script src="@Url.Content("~/Scripts/modernizr-2.5.3.js")"
type="text/javascript"></script>

<script src="@Url.Content("~/Scripts/ajaxplc.js")"
type="text/javascript"></script>

</head>

<body id="bodyLog">

<h1 class="titulo">Telecontrol de Biodigestor para generación de Gas</h1>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")"
type="text/javascript"></script>

<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>

@Html.ValidationSummary(true, "No se ha iniciado la sesión. Corrija los
errores e inténtelo de nuevo.")

@using (Html.BeginForm()) {

<div class="creacion">

<div class="foto">

```

```

<div class="editor-field">

<label>Usuario (Login): </label>

@Html.TextBoxFor(m => m.login)

@Html.ValidationMessageFor(m => m.login)

</div>

<div class="editor-field">

<label>Contraseña: </label>

@Html.PasswordFor(m => m.contrasena)

@Html.ValidationMessageFor(m => m.contrasena)

</div>

<p>

<input type="submit" value="Iniciar sesión" class="inses"/>

</p>

</div>

<div class="foto" id="imag">



</div>

</div>

}
</body>

</html>

```


ANEXO P: CONTROLADOR DEL PROCESO DE LA APLICACIÓN WEB.

```
using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using telecontrol.Models;

using ModbusTCP;

using Modbus;

using System.Net.Sockets;

namespace telecontrol.Controllers

{

    public class ProcesoController : Controller

    {

        private biodigestorEntities db = new biodigestorEntities();

        private ModbusTCP.Master plc;

        private byte[] data1;

        private byte[] data;

        private byte[] data2;
```

```

//
// GET: /Proceso/

public ActionResult Index()
{
    return View(db.proceso.ToList());
}

public ActionResult Principal()
{
    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}

//-----Inicio Tanques -----

public string obtenerNivelDesechos() {
string nivel;

try
{
    plc = new Master("192.168.1.7", 502);

    plc.ReadCoils(1, Convert.ToByte(0), 0, 1, ref data);
}
}

```

```
    catch (Exception ex)
    {
        throw ex;
    }
    finally {
        if (data[0] == 0)
        {
            nivel = "tanquevacio.JPG";
        }
        else if (data[0] == 1)
        {
            nivel = "tanquelleno.JPG";
        }
        else {
            nivel = "";
        }
        plc.disconnect();
    }
    return nivel;
}

public string obtenerNivelAgua() {
    string nivel;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 1, 1, ref data);
    }
}
```

```
}
    catch (Exception ex)

{
    throw ex;

}
finally

{

    if (data[0] == 0)

        {
            nivel = "tanquevacio.JPG";

        }
        else if (data[0] == 1)

            {
                nivel = "tanquelleno.JPG";

            }
            else

                {
                    nivel = "";

                }

            plc.disconnect();

        }

    return nivel;

}

public string obtenerNivelAcido(){

    string nivel;

    try

        {

            plc = new Master("192.168.1.7", 502);
```

```
plc.ReadCoils(1, Convert.ToByte(0), 2, 1, ref data);
```

```
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        if (data[0] == 0)  
        {  
            nivel = "tanquevacio.JPG";  
        }  
        else if (data[0] == 1)  
        {  
            nivel = "tanquelleno.JPG";  
        }  
        else  
        {  
            nivel = "";  
        }  
        plc.disconnect();  
    }  
    return nivel;  
}  
public string obtenerNivelAlcalino()  
{  
    string nivel;  
    try
```

```
{
plc = new Master("192.168.1.7", 502);

plc.ReadCoils(1, Convert.ToByte(0), 3, 1, ref data);

}
catch (Exception ex)

{
throw ex;

}
finally
{
if (data[0] == 0)

{
nivel = "tanquevacio.JPG";
}
else if (data[0] == 1)

{
nivel = "tanquelleno.JPG";
}
else

{
nivel = "";
}

plc.disconnect();
}

return nivel;
}

public string obtenerValvulaDesechos() {

string estado;
```

```
        try
        {
            plc = new Master("192.168.1.7", 502);

            plc.ReadCoils(1, Convert.ToByte(0), 4, 1, ref data);

        }
        catch (Exception ex)
        {
            throw ex;
        }
        finally
        {
            if (data[0] == 1)
            {
                estado = "tuberia_vert_ab.jpg";
            }
            else
            {
                estado = "tuberia_vert_cerr.jpg";
            }

            plc.disconnect();
        }

        return estado;
    }

    public void enviarValvulaDesechos()
    {
        bool estadovalv;
    }
}
try
```

```
{
plc = new Master("192.168.1.7", 502);

plc.ReadCoils(1, Convert.ToByte(0), 4, 1, ref data);

if (data[0] == 1)

    {
    estadovalv = false;
    }
else

    {
    estadovalv = true;
    }

plc.WriteSingleCoils(5, Convert.ToByte(0), 4, estadovalv, ref data);

}
catch (Exception)

{
throw;
}
plc.disconnect();
}

public string obtenerValvulaAgua()
{
    string estado;

    try

    {
plc = new Master("192.168.1.7", 502);

plc.ReadCoils(1, Convert.ToByte(0), 5, 1, ref data);
```



```
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        if (data[0] == 1)  
        {  
            estado = "tuberia_vert_ab.jpg";  
        }  
        else  
        {  
            estado = "tuberia_vert_cerr.jpg";  
        }  
        plc.disconnect();  
    }  
    return estado;  
}  
  
public void enviarValvulaAgua()  
{  
    bool estadovalv;  
  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.ReadCoils(1, Convert.ToByte(0), 5, 1, ref data);  
  
        if (data[0] == 1)
```

```
    {
        estadovalv = false;
    }
    else
    {
        estadovalv = true;
    }
    plc.WriteSingleCoils(5, Convert.ToByte(0), 5, estadovalv, ref data);

}
catch (Exception)
{
    throw;
}
plc.disconnect();
}

public string obtenerValvulaAcido()
{
    string estado;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 6, 1, ref data);

    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
```

```
{
    if (data[0] == 1)
    {
        estado = "tuberia_vert_ab.jpg";
    }
    else
    {
        estado = "tuberia_vert_cerr.jpg";
    }
    plc.disconnect();
}
return estado;
}

public void enviarValvulaAcido()
{
    bool estadovalv;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 6, 1, ref data);

        if (data[0] == 1)
        {
            estadovalv = false;
        }
        else
        {
            estadovalv = true;
        }
    }
}
```

```
    }  
    plc.WriteSingleCoils(5, Convert.ToByte(0), 6, estadovalv, ref data);  
  
    }  
    catch (Exception)  
  
    {  
        throw;  
  
    }  
    plc.disconnect();  
  
    }  
    public string obtenerValvulaAlcalino()  
  
    {  
        string estado;  
  
        try  
  
        {  
            plc = new Master("192.168.1.7", 502);  
  
            plc.ReadCoils(1, Convert.ToByte(0), 7, 1, ref data);  
  
        }  
        catch (Exception ex)  
  
        {  
            throw ex;  
  
        }  
        finally  
  
        {  
            if (data[0] == 1)  
  
            {  
                estado = "tuberia_vert_ab.jpg";  
  
            }  
            else
```

```
        {
            estado = "tuberia_vert_cerr.jpg";
        }
        plc.disconnect();
    }
    return estado;
}

public void enviarValvulaAlcalino()
{
    bool estadovalv;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 7, 1, ref data);

        if (data[0] == 1)
        {
            estadovalv = false;
        }
        else
        {
            estadovalv = true;
        }

        plc.WriteSingleCoils(5, Convert.ToByte(0), 7, estadovalv, ref data);
    }
    catch (Exception)
    {
        throw;
    }
}
```

```

    }
    plc.disconnect();
}

//-----Fin Tanques-----

//-----Inicio Carga-----

    public string obtenerNivelContenedor() {
string nivel;

    try

        {
            plc = new Master("192.168.1.7", 502);

            plc.ReadCoils(1, Convert.ToByte(0), 8, 1, ref data);//Nivel vacio
contenedor

            plc.ReadCoils(1, Convert.ToByte(0), 9, 1, ref data1);//Nivel desechos

            plc.ReadCoils(1, Convert.ToByte(0), 10, 1, ref data2);//Nivel lleno, ya con
agua

        }
        catch (Exception ex)

        {
            throw ex;

        }
        finally

        {
            if (data2[0] == 1)

```

```

    {
        nivel = "tanquelleno.JPG";
    }
    else if (data1[0] == 1)
    {
        nivel = "tanquemedio.JPG";
    }
    else
    {
        nivel = "tanquevacio.JPG";
    }
    plc.disconnect();
}
return nivel;
}
public string obtenerAgitador() {
string posicion;
try
{
    plc = new Master("192.168.1.7", 502);
    plc.ReadCoils(1, Convert.ToByte(0), 11, 1, ref data);//agitador
}
catch (Exception ex)
{
    throw ex;
}
finally
{

```

```
        if (data[0] == 1)
        {
            posicion = "agitador.gif";
        }
        else {
            posicion="Captura.JPG";
        }

        plc.disconnect();

    }
    return posicion;
}

public void enviarAgitador() {
    bool estado;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 11, 1, ref data);

        if (data[0] == 1)
        {
            estado = false;
        }
        else
        {
            estado = true;
        }
        plc.WriteSingleCoils(5, Convert.ToByte(0), 11, estado, ref data);
    }
}
```



```
    }  
    catch (Exception)  
    {  
        throw;  
    }  
    plc.disconnect();  
}  
  
public string obtenerValvulaDigestor() {  
    string estado;  
  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.ReadCoils(1, Convert.ToByte(0), 12, 1, ref data);  
  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        if (data[0] == 1)  
        {  
            estado = "tuberia_vert_ab.jpg";  
        }  
        else  
        {  
            estado = "tuberia_vert_cerr.jpg";  
        }  
    }  
}
```

```
        plc.disconnect();
    }
    return estado;
}

    public void enviarValvulaDigesor()
    {
    bool estadovalv;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 12, 1, ref data);

        if (data[0] == 1)
        {
            estadovalv = false;
        }
        else
        {
            estadovalv = true;
        }
        plc.WriteSingleCoils(5, Convert.ToByte(0), 12, estadovalv, ref data);
    }
    catch (Exception)
    {
        throw;
    }
    plc.disconnect();
}
```

```

    }

//-----FIN CARGA-----

//-----INICIO DIGESTION -----

public string obtenerNivelDigestor() {

string nivel;

try

{
    plc = new Master("192.168.1.7", 502);

    plc.ReadCoils(1, Convert.ToByte(0), 13, 1, ref data); //Nivel vacio digestor

    plc.ReadHoldingRegister(2, Convert.ToByte(0), 0, 1, ref data1);

}
catch (Exception ex)

{
    throw ex;

}
finally

{

    if (data[0] == 0)

        {
            nivel = "DIGESTOR_VACIO.JPG";

        }
        else

            {

                if (data1[1] == 1){

                    nivel = "DIGESTOR_POS1.JPG";

                }

            }

}

}

```

```
    }  
    else if (data1[1] == 2)  
    {  
        nivel = "DIGESTOR_POS2.JPG";  
    }  
    else {  
        nivel = "DIGESTOR_POS3.JPG";  
    }  
    }  
    }  
    plc.disconnect();  
}  
return nivel;  
}  
public string obtenerValvulaGas()  
{  
    string estado;  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
        plc.ReadCoils(1, Convert.ToByte(0), 14, 1, ref data);  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally
```

```
{  
    if (data[0] == 1)  
    {  
        estado = "../Content/images/gas.JPG";  
    }  
    else  
    {  
        estado = "";  
    }  
    plc.disconnect();  
}  
return estado;  
  
}  
public void enviarValvulaGas()  
{  
    bool estadovalv;  
  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.ReadCoils(1, Convert.ToByte(0), 14, 1, ref data);  
  
        if (data[0] == 1)  
        {  
            estadovalv = false;  
        }  
        else  
        {  
            estadovalv = true;  
        }  
    }  
}
```

```
    }
    plc.WriteSingleCoils(5, Convert.ToByte(0), 14, estadovalv, ref data);

    }
    catch (Exception)

    {
        throw;
    }
    plc.disconnect();
}

public string obtenerValvulaBiosol()
{
    string estado;

    try

    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 15, 1, ref data);

    }
    catch (Exception ex)

    {
        throw ex;
    }
    finally

    {
        if (data[0] == 1)

        {
            estado = "tuberia_horiz_ab.jpg";

        }
        else
```

```
        {
            estado = "tuberia_horiz_cerr.jpg";
        }
        plc.disconnect();
    }
    return estado;
}

public void enviarValvulaBiosol()
{
    bool estadovalv;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 15, 1, ref data);

        if (data[0] == 1)
        {
            estadovalv = false;
        }
        else
        {
            estadovalv = true;
        }

        plc.WriteSingleCoils(5, Convert.ToByte(0), 15, estadovalv, ref data);
    }
    catch (Exception)
    {
        throw;
    }
}
```

```
    }
    plc.disconnect();
}

public string obtenerQuemador()
{
    string estado;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 16, 1, ref data);

    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        if (data[0] == 1)
        {
            estado = "../Content/images/llama.png";
        }
        else
        {
            estado = "";
        }

        plc.disconnect();
    }
}
```



```
        return estado;
    }

    public void enviarQuemador()
    {
        bool estadovalv;

        try
        {
            plc = new Master("192.168.1.7", 502);

            plc.ReadCoils(1, Convert.ToByte(0), 16, 1, ref data);

            if (data[0] == 1)
            {
                estadovalv = false;
            }
            else
            {
                estadovalv = true;
            }

            plc.WriteSingleCoils(5, Convert.ToByte(0), 16, estadovalv, ref data);

        }
        catch (Exception)
        {
            throw;
        }
        plc.disconnect();
    }

    public string obtenerCompresor()
```

```
{
    string estado;

    try
    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadCoils(1, Convert.ToByte(0), 17, 1, ref data);

    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        if (data[0] == 1)
        {
            estado = "../Content/images/aire.JPG";
        }
        else
        {
            estado = "";
        }

        plc.disconnect();
    }

    return estado;
}

public void enviarCompresor()
```

```
{
bool estadovalv;

try

{
plc = new Master("192.168.1.7", 502);

plc.ReadCoils(1, Convert.ToByte(0), 17, 1, ref data);

if (data[0] == 1)

    {
    estadovalv = false;
    }
else

    {
    estadovalv = true;
    }

plc.WriteSingleCoils(5, Convert.ToByte(0), 17, estadovalv, ref data);

}
catch (Exception)

{
throw;
}

plc.disconnect();
}

public string obtenerTiempoDig() {

string tiempo;

double tiempoDig;

double minutos;
```

```
double segundos;

try
{
    plc = new Master("192.168.1.7", 502);

    plc.ReadHoldingRegister(2, Convert.ToByte(0), 1, 2, ref data);

}
catch (Exception)
{
    throw;
}
finally
{
    tiempoDig = conversionModbus(data[2], data[3], data[0], data[1]);

    tiempoDig = tiempoDig / 60000;

    minutos = Math.Truncate(tiempoDig);

    segundos = tiempoDig - minutos;

    segundos = segundos * 60;

    segundos = Math.Truncate(segundos);

    tiempo = minutos + " Minutos, " + segundos + " Segundos";

}
plc.disconnect();

return tiempo;
}

//----- FIN DIGESTION -----
```

```
//----- INICIO VARIABLES-----
```

```
public string obtenerPresion()
```

```
{  
double result;
```

```
try
```

```
{  
plc = new Master("192.168.1.7", 502);
```

```
plc.ReadHoldingRegister(2, Convert.ToByte(0), 3, 2, ref data);
```

```
}  
catch (Exception ex)
```

```
{  
throw ex;
```

```
}  
finally
```

```
{  
result = conversionModbus(data[2], data[3], data[0], data[1]);
```

```
result = Math.Round(result, 2);
```

```
}
```

```
plc.disconnect();
```

```
return Convert.ToString(result);
```

```
}
```

```
public string obtenerTemperatura()
```

```
{  
double result;
```

```
try
```

```

    {
    plc = new Master("192.168.1.7", 502);

    plc.ReadHoldingRegister(2, Convert.ToByte(0), 5, 2, ref data);

    }
    catch (Exception ex)

    {
    throw ex;
    }
    finally

    {
    result = conversionModbus(data[2], data[3], data[0], data[1]);

    result = Math.Round(result, 2);

    }
    plc.disconnect();

    return Convert.ToString(result);
    }

    public string obtenerPH()

    {
    double result;

    try

    {
    plc = new Master("192.168.1.7", 502);

    plc.ReadHoldingRegister(2, Convert.ToByte(0), 7, 2, ref data);

    }
    catch (Exception ex)

    {

```

```

        throw ex;
    }
    finally

    {
result = conversionModbus(data[2], data[3], data[0], data[1]);

result = Math.Round(result, 2);

    }
    plc.disconnect();

    return Convert.ToString(result);
    }
//----- FIN VARIABLES -----

public string obtenerManual()

{
string resultado;

try

    {
plc = new Master("192.168.1.7", 502);

plc.ReadCoils(1, Convert.ToByte(0), 18, 1, ref data);

    }
    catch (Exception ex)

    {
        throw ex;
    }
    finally

    {
        if (data[0] == 1)

```

```
        {
            resultado = "#61ee16";
        }
        else

            {
                resultado = "#1a389a";
            }

        plc.disconnect();

    }
    return resultado;

}

public void enviarManual()
{
    bool estadovalv;

    try

        {
            plc = new Master("192.168.1.7", 502);

            plc.ReadCoils(1, Convert.ToByte(0), 18, 1, ref data);

            if (data[0] == 1)

                {
                    estadovalv = false;
                }
            else

                {
                    estadovalv = true;
                }

            plc.WriteSingleCoils(5, Convert.ToByte(0), 18, estadovalv, ref data);
```



```
    }  
    catch (Exception)  
    {  
        throw;  
    }  
    plc.disconnect();  
}  
  
public string obtenerActivar()  
{  
    string resultado;  
  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.ReadCoils(1, Convert.ToByte(0), 19, 1, ref data);  
  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        if (data[0] == 1)  
        {  
            resultado = "#61ee16";  
        }  
        else  
        {  
            resultado = "#1a389a";  
        }  
    }  
}
```

```
    }  
    plc.disconnect();  
  
    }  
    return resultado;  
  
    }  
    public void enviarActivar()  
    {  
        try  
        {  
            plc = new Master("192.168.1.7", 502);  
  
            plc.WriteSingleCoils(5, Convert.ToByte(0), 19, true, ref data);  
  
        }  
        catch (Exception)  
        {  
            throw;  
        }  
        plc.disconnect();  
    }  
    public string obtenerDetener()  
    {  
        string resultado;  
  
        try  
        {  
            plc = new Master("192.168.1.7", 502);  
  
            plc.ReadCoils(1, Convert.ToByte(0), 20, 1, ref data);
```

```
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        if (data[0] == 1)  
        {  
            resultado = "#f61111";  
        }  
        else  
        {  
            resultado = "#ffc700";  
        }  
        plc.disconnect();  
    }  
    return resultado;  
}  
  
public void enviarDetener()  
{  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.WriteSingleCoils(5, Convert.ToByte(0), 20, true, ref data);  
    }  
    catch (Exception)
```

```
        {
            throw;
        }
        plc.disconnect();
    }

    public void enviarTempmax() {
        var te = Request.Form[0];
        double tempera = Convert.ToDouble(te);
        byte[] temperaturamax = escribirFloat(tempera);

        try
        {
            plc = new Master("192.168.1.7", 502);
            plc.WriteMultipleRegister(8, Convert.ToByte(0), 9, temperaturamax);
        }
        catch (Exception)
        {
            throw;
        }
        plc.disconnect();
    }

    public void enviarTempmin() {
        var te = Request.Form[0];
        double tempera = Convert.ToDouble(te);
        byte[] temperaturamin = escribirFloat(tempera);

        try
```

```
{
    plc = new Master("192.168.1.7", 502);
plc.WriteMultipleRegister(8, Convert.ToByte(0), 11, temperaturamin);

}
catch (Exception)
{
    throw;
}
plc.disconnect();
}

public void enviarpHmax() {
var ph = Request.Form[0];
double phdob = Convert.ToDouble(ph);
byte[] phmax = escribirFloat(phdob);

    try
    {
plc = new Master("192.168.1.7", 502);

plc.WriteMultipleRegister(8, Convert.ToByte(0), 13, phmax);

    }
    catch (Exception)
    {
        throw;
    }
    plc.disconnect();
}

public void enviarpHmin()
```

```
{
var ph = Request.Form[0];

double phdob = Convert.ToDouble(ph);

byte[] phmin = escribirFloat(phdob);

    try

    {
plc = new Master("192.168.1.7", 502);

plc.WriteMultipleRegister(8, Convert.ToByte(0), 15, phmin);

    }
    catch (Exception)

    {
        throw;
    }
    plc.disconnect();

}

public void enviarpresmax() {
var pres = Request.Form[0];

double presdob = Convert.ToDouble(pres);

byte[] presmax = escribirFloat(presdob);

    try

    {
plc = new Master("192.168.1.7", 502);

plc.WriteMultipleRegister(8, Convert.ToByte(0), 17, presmax);
```

```
    }  
    catch (Exception)  
    {  
        throw;  
    }  
    plc.disconnect();  
}  
  
public string obtenerTempMax()  
  
    {  
    double result;  
  
    try  
    {  
        plc = new Master("192.168.1.7", 502);  
  
        plc.ReadHoldingRegister(2, Convert.ToByte(0), 9, 2, ref data);  
  
    }  
    catch (Exception ex)  
    {  
        throw ex;  
    }  
    finally  
    {  
        result = conversionModbus(data[2], data[3], data[0], data[1]);  
  
        result = Math.Round(result, 2);  
    }  
  
    plc.disconnect();  
  
    return Convert.ToString(result);  
}
```

```
    }  
    public string obtenerTempMin()  
    {  
        double result;  
  
        try  
        {  
            plc = new Master("192.168.1.7", 502);  
  
            plc.ReadHoldingRegister(2, Convert.ToByte(0), 11, 2, ref data);  
  
        }  
        catch (Exception ex)  
        {  
            throw ex;  
        }  
        finally  
        {  
            result = conversionModbus(data[2], data[3], data[0], data[1]);  
  
            result = Math.Round(result, 2);  
        }  
  
        plc.disconnect();  
  
        return Convert.ToString(result);  
    }  
    public string obtenerpHMax()  
    {  
        double result;  
  
        try  
        {
```



```
plc = new Master("192.168.1.7", 502);

plc.ReadHoldingRegister(2, Convert.ToByte(0), 13, 2, ref data);

    }
    catch (Exception ex)

    {
        throw ex;
    }
    finally

    {
        result = conversionModbus(data[2], data[3], data[0], data[1]);

        result = Math.Round(result, 2);

    }

plc.disconnect();

return Convert.ToString(result);

}

public string obtenerpHMin()

{
    double result;

    try

    {
        plc = new Master("192.168.1.7", 502);

        plc.ReadHoldingRegister(2, Convert.ToByte(0), 15, 2, ref data);

    }
    catch (Exception ex)

    {
```

```
        throw ex;
    }
    finally

    {
result = conversionModbus(data[2], data[3], data[0], data[1]);

result = Math.Round(result, 2);

    }
plc.disconnect();
return Convert.ToString(result);
}

public string obtenerPresMax()
{
double result;

try

    {
plc = new Master("192.168.1.7", 502);

plc.ReadHoldingRegister(2, Convert.ToByte(0), 17, 2, ref data);

    }
    catch (Exception ex)

    {
        throw ex;
    }
    finally

    {
result = conversionModbus(data[2], data[3], data[0], data[1]);

result = Math.Round(result, 2);
```

```
    }  
    plc.disconnect();  
    return Convert.ToString(result);  
    }  
    static double conversionModbus(byte b1, byte b2, byte b3, byte b4)  
    {  
    int valor1, valor2, valor3, valor4;  
    valor1 = Convert.ToInt32(b1);  
    valor2 = Convert.ToInt32(b2);  
    valor3 = Convert.ToInt32(b3);  
    valor4 = Convert.ToInt32(b4);  
    string binario = "";  
    string s1 = Convert.ToString(valor1, 2);  
    while (s1.Length < 8)  
    {  
        s1 = "0" + s1;  
    }  
    string s2 = Convert.ToString(valor2, 2);  
    while (s2.Length < 8)  
    {  
        s2 = "0" + s2;  
    }  
    string s3 = Convert.ToString(valor3, 2);  
    while (s3.Length < 8)  
    {  
        s3 = "0" + s3;  
    }  
    }
```

```
    }  
    string s4 = Convert.ToString(valor4, 2);  
    while (s4.Length < 8)  
    {  
        s4 = "0" + s4;  
    }  
    binario = s1 + s2 + s3 + s4;  
    char signo = binario[0]; string exponente = binario.Substring(1, 8);  
  
    string mantisa = binario.Substring(9);  
  
    int m1;  
    int m2;  
    double m3 = 1.0;  
    int contador = 0;  
    if (signo == '1')  
    {  
        m1 = -1;  
    }  
    else {  
        m1 = 1;  
    }  
    m2 = Convert.ToInt32(exponente, 2) - 127;  
    foreach (char bit in mantisa) {  
        contador = contador - 1;  
        if (bit == '1') {  
            m3 = m3 + Math.Pow(2, contador);  
        }  
    }  
}
```

```
    }  
  }  
  double dato;  
  dato = m1 * m3 * Math.Pow(2,m2);  
  return dato;  
}  
static byte[] escribirFloat(double numero)  
{  
  //byte[] mod;  
  int signo;  
  int expo;  
  double div;  
  double mantisa;  
  string bin;  
  if (numero < 0)  
  {  
    signo = -1;  
    bin = "1";  
  }  
  else  
  {  
    signo = 1;  
    bin = "0";  
  }  
  expo = Convert.ToInt32(Math.Truncate((Math.Log10(Math.Abs(numero))  
/ (Math.Log10(2.00))));
```

```
div = Math.Pow(2, expo);  
expo = expo + 127;  
bin = bin + Convert.ToString(expo, 2);  
mantisa = (numero / div) * signo;  
mantisa=mantisa-1;  
for (int cont=-1 ; cont > -24 ; cont--)  
{  
    if (mantisa - Math.Pow(2, cont) >= 0)  
    {  
        mantisa = mantisa - Math.Pow(2, cont);  
        bin = bin + "1";  
    }  
    else  
    {  
        bin = bin + "0";  
    }  
}  
int tam = bin.Length / 8;  
byte[] bytes = new byte[tam];  
for (int i = 0; i < tam; ++i)  
{  
    bytes[i] = Convert.ToByte(bin.Substring(8 * i, 8), 2);  
}  
var interm = bytes[2];  
bytes[2] = bytes[0];  
bytes[0] = interm;
```

```
interm = bytes[3];

bytes[3] = bytes[1];

bytes[1] = interm;

    return bytes;
}

//
// GET: /Proceso/Details/5

public ViewResult Details(int id)
{
    proceso proceso = db.proceso.Find(id);

    return View(proceso);
}

//
// GET: /Proceso/Create

public ActionResult Create()
{
    return View();
}

//
// POST: /Proceso/Create

[HttpPost]

public ActionResult Create(proceso proceso)
{
    if (ModelState.IsValid)
    {
        db.proceso.Add(proceso);

        db.SaveChanges();
    }
}
```

```
return RedirectToAction("Index");
    }
    return View(proceso);
}
public ActionResult Tanques()
{
    //usuario Usuario = new usuario();
    //Usuario = (usuario)Session["DatosUsuario"];

    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}
[HttpPost]
public ActionResult Tanques(string iden)
{
    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}
```



```
    }  
  }  
  
  public ActionResult Carga()  
  
  {  
    if (Session["DatosUsuario"] != null)  
  
    {  
      return View((List<menu>)Session["Opciones"]);  
  
    }  
    else  
  
    {  
      return RedirectToAction("LogOn", "Usuario");  
  
    }  
  }  
  [HttpPost]  
  
  public ActionResult Carga(string id)  
  
  {  
    if (Session["DatosUsuario"] != null)  
  
    {  
      return View((List<menu>)Session["Opciones"]);  
  
    }  
    else  
  
    {  
      return RedirectToAction("LogOn", "Usuario");  
  
    }  
  }  
  
  public ActionResult Digestion()  
  
  {  
    if (Session["DatosUsuario"] != null)
```

```
{
    return View((List<menu>)Session["Opciones"]);
}
else
{
    return RedirectToAction("LogOn", "Usuario");
}
}
[HttpPost]

public ActionResult Digestion(string id)
{
    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}

public ActionResult Variables()
{
    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}
```

[HttpPost]

```
public ActionResult Variables(string iden)
{
    if (Session["DatosUsuario"] != null)
    {
        return View((List<menu>)Session["Opciones"]);
    }
    else
    {
        return RedirectToAction("LogOn", "Usuario");
    }
}
//
// GET: /Proceso/Edit/5
```

```
public ActionResult Edit(int id)
{
    proceso proceso = db.proceso.Find(id);
    return View(proceso);
}
//
// POST: /Proceso/Edit/5
```

[HttpPost]

```
public ActionResult Edit(proceso proceso)
{
    if (ModelState.IsValid)
    {
        db.Entry(proceso).State = EntityState.Modified;
        db.SaveChanges();
    }
}
```

```

return RedirectToAction("Index");
    }
    return View(proceso);
}
//
// GET: /Proceso/Delete/5

public ActionResult Delete(int id)
{
    proceso proceso = db.proceso.Find(id);
    return View(proceso);
}
//
// POST: /Proceso/Delete/5

[HttpPost, ActionName("Delete")]
public ActionResult DeleteConfirmed(int id)
{
    proceso proceso = db.proceso.Find(id);
    db.proceso.Remove(proceso);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    db.Dispose();
    base.Dispose(disposing);
}
}
}

```

Controlador de Usuarios:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using telecontrol.Models;

namespace telecontrol.Controllers
{
    public class UsuarioController : Controller
    {
        private biodigestorEntities db = new biodigestorEntities();

        // GET: /Usuario/
        public ActionResult Index()
        {
            return View(db.usuario.ToList());
        }

        // GET: /Usuario/Details/5
        public ActionResult Details(int id)
        {
```

```
usuario usuario = db.usuario.Find(id);

return View(usuario);

}

//
// GET: /Usuario/Create

public ActionResult Create()

{
    return View();
}

//
// POST: /Usuario/Create

[HttpPost]

public ActionResult Create(usuario usuario)

{
    if (ModelState.IsValid)

    {
        usuario.estado = "A";

        db.usuario.Add(usuario);

        db.SaveChanges();

        return RedirectToAction("Index");

    }

    return View(usuario);
}

//
// GET: /Usuario/Edit/5

public ActionResult Edit(int id)

{
```

```
usuario usuario = db.usuario.Find(id);

return View(usuario);

}

//
// POST: /Usuario/Edit/5

[HttpPost]

public ActionResult Edit(usuario usuario)

{
    if (ModelState.IsValid)

    {
        db.Entry(usuario).State = EntityState.Modified;

        db.SaveChanges();

        return RedirectToAction("Index");

    }

    return View(usuario);

}

//
// GET: /Usuario/Delete/5

public ActionResult Delete(int id)

{
    usuario usuario = db.usuario.Find(id);

    return View(usuario);

}

//
// POST: /Usuario/Delete/5

[HttpPost, ActionName("Delete")]

public ActionResult DeleteConfirmed(int id)
```

```

    {
    usuario usuario = db.usuario.Find(id);

    db.usuario.Remove(usuario);

    db.SaveChanges();

    return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
    db.Dispose();

    base.Dispose(disposing);
    }

    public ActionResult LogOn()
    {
    return View();
    }

    public ActionResult LogOff()
    {
    Session.RemoveAll();

    return View();
    }

    [HttpPost]

    public ActionResult LogOn(usuario model)
    {
    List <menu> Botones = new List<menu>();

    usuario Usuario = new usuario();

    if (ModelState.IsValid)

```



```

    {
        if (ValidaUser( model))

            {
                Usuario = (usuario)Session["DatosUsuario"];

                Botones = OpcionesDeUsuario(Usuario.perfil);

                Session.Add("Opciones", Botones);

                return RedirectToAction("Principal", "Proceso");

            }
        else

            {
                ModelState.AddModelError("", "El nombre de usuario o la contraseña
                especificados son incorrectos.");

            }

        // Si llegamos a este punto, es que se ha producido un error y volvemos a
        mostrar el formulario

        return View(model);

    }

    public List<menu> OpcionesDeUsuario(int idPerfil)

    {
        try

        {
            List<perfil_menu> opciones = db.perfil_menu.Where(x=>x.idperfil ==
            idPerfil).ToList();

            List<menu> menu = new List<menu>();

            foreach (perfil_menu item in opciones)

                {

```

```

        menu.Add(db.menu.Where(j => j.idmenu ==
item.idmenu).FirstOrDefault());
    }
    return menu;
}
catch (Exception)
{
    throw;
}
}
bool ValidaUser (usuario model)
{
    bool valida = true;

    usuario usuario = db.usuario.Where(x=> x.login == model.login &&
x.contrasena == model.contrasena).FirstOrDefault();

    if (usuario == null)
    {
        valida = false;
    }
    Session.Add("DatosUsuario", usuario);

    return valida;
}
}
}

```

ANEXO Q: JAVASCRIPT DE CONSULTA CADA SEGUNDO DE LA APLICACIÓN WEB.

```
//Suma de Permisos

window.onload = function asignaEventos() {

    document.getElementById("st")
        .addEventListener("click", suma, false);

    document.getElementById("ec")
        .addEventListener("click", suma, false);

    document.getElementById("ed")
        .addEventListener("click", suma, false);

    document.getElementById("mv")
        .addEventListener("click", suma, false);

    document.getElementById("admin")
        .addEventListener("click", admin, false);

    document.getElementById("est")
        .addEventListener("click", est, false);

};

function admin() {

    document.getElementById("st").checked = true;
    document.getElementById("ec").checked = true;
    document.getElementById("ed").checked = true;
    document.getElementById("mv").checked = true;
```

```
document.getElementById("st").disabled = true;
document.getElementById("ec").disabled = true;
document.getElementById("ed").disabled = true;
document.getElementById("mv").disabled = true;
document.getElementById("est").checked = false;
suma();
}
function est() {
    document.getElementById("st").checked = false;
    document.getElementById("ec").checked = false;
    document.getElementById("ed").checked = false;
    document.getElementById("mv").checked = false;
    document.getElementById("st").disabled = false;
    document.getElementById("ec").disabled = false;
    document.getElementById("ed").disabled = false;
    document.getElementById("mv").disabled = false;
    document.getElementById("admin").checked = false;
}
function suma() {
    var check1 = document.getElementById("st");
    var check2 = document.getElementById("ec");
    var check3 = document.getElementById("ed");
    var check4 = document.getElementById("mv");
```

```

var adm = document.getElementById("admin");

var resultado = document.getElementById("tipoPerfil");

var s = 0;

if (adm.checked) {
    s = 0;
}
else {

if (check1.checked && check2.checked && check3.checked &&
check4.checked)

    s = Number(check1.value) + Number(check2.value) +
Number(check3.value) + Number(check4.value);

    else if (check1.checked && check2.checked && check3.checked &&
!check4.checked)

        s = Number(check1.value) + Number(check2.value) +
Number(check3.value);

    else if (check1.checked && check2.checked && check4.checked &&
!check3.checked)

        s = Number(check1.value) + Number(check2.value) +
Number(check4.value);

    else if (check1.checked && check3.checked && check4.checked &&
!check2.checked)

```

```
s = Number(check1.value) + Number(check4.value) +  
Number(check3.value);  
  
else if (check4.checked && check2.checked && check3.checked &&  
!check1.checked)  
  
s = Number(check4.value) + Number(check2.value) +  
Number(check3.value);  
  
else if (check1.checked && check2.checked && !check3.checked &&  
!check4.checked)  
  
s = Number(check1.value) + Number(check2.value);  
  
else if (check1.checked && check3.checked && !check2.checked &&  
!check4.checked)  
  
s = Number(check1.value) + Number(check3.value);  
  
else if (check3.checked && check2.checked && !check1.checked &&  
!check4.checked)  
  
s = Number(check3.value) + Number(check2.value);  
  
else if (check1.checked && check4.checked && !check2.checked &&  
!check3.checked)  
  
s = Number(check1.value) + Number(check4.value);  
  
else if (check4.checked && check2.checked && !check1.checked &&  
!check3.checked)  
  
s = Number(check4.value) + Number(check2.value);
```

```

        else if (check3.checked && check4.checked && !check1.checked &&
!check2.checked)
            s = Number(check3.value) + Number(check4.value);
        else if (check1.checked && !check2.checked && !check3.checked &&
!check4.checked)
            s = Number(check1.value);
        else if (check2.checked && !check1.checked && !check3.checked &&
!check4.checked)
            s = Number(check2.value);
        else if (check3.checked && !check1.checked && !check2.checked &&
!check4.checked)
            s = Number(check3.value);
        else if (check4.checked && !check2.checked && !check3.checked &&
!check1.checked)
            s = Number(check4.value);
    }
    /* Operadores ternarios

    var valor1 = check1.checked ? Number(check1.value) : 0;

    var valor2 = check2.checked ? Number(check2.value) : 0;

    s = valor1 + valor2;

    */
    resultado.value = s;

```

```

}
//----- Web Responsive -----
$(function () {

    var btn_movil = $('#nav-mobile'),
        menu = $('#menu').find('ul');

    // Al dar click agregar/quitar clases que permiten el despliegue del menú
    btn_movil.on('click', function (e) {
        e.preventDefault();

        var el = $(this);

        el.toggleClass('nav-active');
        menu.toggleClass('open-menu');
    });
});
//-----Envio de señales-----

function enviarValvulaDesechos() {

    $.get("/Proceso/enviarValvulaDesechos", null, function (data) {

    })
}

function enviarValvulaAgua() {

    $.get("/Proceso/enviarValvulaAgua", null, function (data) {

    })
}

function enviarValvulaAcido() {

```



```
$.get("/Proceso/enviarValvulaAcido", null, function (data) {  
  
  })  
}  
  
function enviarValvulaAlcalino() {  
  
  $.get("/Proceso/enviarValvulaAlcalino", null, function (data) {  
  
  })  
}  
  
function enviarAgitador() {  
  
  $.get("/Proceso/enviarAgitador", null, function (data) {  
  
  })  
}  
  
function enviarValvulaDigestor() {  
  
  $.get("/Proceso/enviarValvulaDigestor", null, function (data) {  
  
  })  
}  
  
function enviarValvulaGas() {  
  
  $.get("/Proceso/enviarValvulaGas", null, function (data) {  
  
  })  
}  
  
function enviarValvulaBiosol() {  
  
  $.get("/Proceso/enviarValvulaBiosol", null, function (data) {  
  
  })  
}  
  
function enviarQuemador() {  
  
  $.get("/Proceso/enviarQuemador", null, function (data) {
```

```
    })  
  }  
  
  function enviarCompresor() {  
  
    $.get("/Proceso/enviarCompresor", null, function (data) {  
  
    })  
  }  
  
  function enviarManual() {  
  
    $.get("/Proceso/enviarManual", null, function (data) {  
  
    })  
  }  
  
  function enviarActivar() {  
  
    $.get("/Proceso/enviarActivar", null, function (data) {  
  
    })  
  }  
  
  function enviarDetener() {  
  
    $.get("/Proceso/enviarDetener", null, function (data) {  
  
    })  
  }  
  
  function enviarTempmax() {  
  
    var temp = $("#tempmax").val();  
  
    var valid = document.getElementById("tempmax")  
  
    if (valid.checkValidity() == false)  
  
    {  
      document.getElementById("a1").innerHTML = valid.validationMessage;  
    }  
  }  
}
```

```

else if (temp == null || temp == "") {
    document.getElementById("a1").innerHTML = "Ingrese un valor";
}
else {
    document.getElementById("a1").innerHTML = "Enviando...";
    $.post("/Proceso/enviarTempmax", { tempmax: temp }, function (data) {
        document.getElementById("a1").innerHTML = "Valor Almacenado";
    })
}
}
function enviarTempmin() {
    var temp = $("#tempmin").val();
    var valid = document.getElementById("tempmin")

    if (valid.checkValidity() == false) {
        document.getElementById("a2").innerHTML = valid.validationMessage;
    }
    else if (temp == null || temp == "") {
        document.getElementById("a2").innerHTML = "Ingrese un valor";
    }
    else {
        document.getElementById("a2").innerHTML = "Enviando...";
        $.post("/Proceso/enviarTempmin", { tempmin: temp }, function (data) {
            document.getElementById("a2").innerHTML = "Valor Almacenado";
        })
    }
}
}

```

```
}  
  
function enviarpHmax() {  
  
    var ph = $("#phmax").val();  
  
    var valid = document.getElementById("phmax")  
  
    if (valid.checkValidity() == false) {  
        document.getElementById("a3").innerHTML = valid.validationMessage;  
    }  
    else if (ph == null || ph == "") {  
        document.getElementById("a3").innerHTML = "Ingrese un valor";  
    }  
    else {  
        document.getElementById("a3").innerHTML = "Enviando...";  
        $.post("/Proceso/enviarpHmax", { phmax: ph }, function (data) {  
            document.getElementById("a3").innerHTML = "Valor Almacenado";  
        })  
    }  
}  
  
function enviarpHmin() {  
  
    var ph = $("#phmin").val();  
  
    var valid = document.getElementById("phmin")  
  
    if (valid.checkValidity() == false) {  
        document.getElementById("a4").innerHTML = valid.validationMessage;
```

```

}
else if (ph == null || ph == "") {
    document.getElementById("a4").innerHTML = "Ingrese un valor";
}
else {
    document.getElementById("a4").innerHTML = "Enviando...";
    $.post("/Proceso/enviarpHmin", { phmax: ph }, function (data) {
        document.getElementById("a4").innerHTML = "Valor Almacenado";
    })
}
}
}

```

```

function enviarpresmax() {

    var pres = $("#pmax").val();
    var valid = document.getElementById("pmax")

    if (valid.checkValidity() == false) {
        document.getElementById("a5").innerHTML = valid.validationMessage;
    }
    else if (ph == null || ph == "") {
        document.getElementById("a5").innerHTML = "Ingrese un valor";
    }
    else {
        document.getElementById("a5").innerHTML = "Enviando...";
        $.post("/Proceso/enviarpresmax", { presmax: pres }, function (data) {

```

```

        document.getElementById("a5").innerHTML = "Valor Almacenado";
    })
}
}
//-----Lectura desde el PLC cada 0.5 segundos -----

window.setInterval(function () {

    cache: false;

    //-----Tanques-----

    var url2 = "/Proceso/obtenerNivelDesechos";
    $.get(url2, null, function (data) {
        $("#desechos").attr('src', "../Content/images/" + data);
    });
    var url3 = "/Proceso/obtenerNivelAgua";
    $.get(url3, null, function (data) {
        $("#agua").attr('src', "../Content/images/" + data);
    });
    var url4 = "/Proceso/obtenerNivelAcido";
    $.get(url4, null, function (data) {
        $("#acido").attr('src', "../Content/images/" + data);
    });
    var url5 = "/Proceso/obtenerNivelAlcalino";
    $.get(url5, null, function (data) {
        $("#alcalino").attr('src', "../Content/images/" + data);
    });

```

```
});  
var url6 = "/Proceso/obtenerValvulaDesechos";  
$.get(url6, null, function (data) {  
    $("#valv_des").attr('src', "../Content/images/" + data);  
});  
var url7 = "/Proceso/obtenerValvulaAgua";  
$.get(url7, null, function (data) {  
    $("#valv_agua").attr('src', "../Content/images/" + data);  
});  
var url8 = "/Proceso/obtenerValvulaAcido";  
$.get(url8, null, function (data) {  
    $("#valv_acido").attr('src', "../Content/images/" + data);  
});  
var url9 = "/Proceso/obtenerValvulaAlcalino";  
$.get(url9, null, function (data) {  
    $("#valv_alcalino").attr('src', "../Content/images/" + data);  
});  
//----FIN TANQUES---  
//-----Carga-----  
var url10 = "/Proceso/obtenerNivelContenedor";  
$.get(url10, null, function (data) {  
    $("#cont").attr('src', "../Content/images/" + data);  
});  
var url11 = "/Proceso/obtenerAgitador";
```

```

$.get(url11, null, function (data) {
    $("#agita").attr('src', "../Content/images/" + data );
});
var url12 = "/Proceso/obtenerValvulaDigestor";
$.get(url12, null, function (data) {
    $("#valvula_carga").attr('src', "../Content/images/" + data);
});

//-----FIN CARGA-----

//----- DIGESTION -----

var url13 = "/Proceso/obtenerNivelDigestor";
$.get(url13, null, function (data) {
    $("#digestor").attr('src', "../Content/images/" + data);
});
var url14 = "/Proceso/obtenerValvulaGas";
$.get(url14, null, function (data) {
    $("#gas").attr('src', data);
});
var url15 = "/Proceso/obtenerValvulaBiosol";
$.get(url15, null, function (data) {
    $("#valvDig").attr('src', "../Content/images/" + data);
});
var url16 = "/Proceso/obtenerQuemador";
$.get(url16, null, function (data) {

```



```

        $("#llama").attr('src', data);
    });
    var url17 = "/Proceso/obtenerCompresor";
    $.get(url17, null, function (data) {
        $("#aire").attr('src', data);
    });

    var url18 = "/Proceso/obtenerTiempoDig";
    $.get(url18, null, function (data) {
        $("#tiempoDig").html(data);
    });
    //-----FIN DIGESTION -----

    var url19 = "/Proceso/obtenerActivar";
    $.get(url19, null, function (data) {
        $(".activar").css("background", data);
    });

    var url20 = "/Proceso/obtenerDetener";
    $.get(url20, null, function (data) {
        $(".detener").css("background", data);
    });

    var url21 = "/Proceso/obtenerManual";
    $.get(url21, null, function (data) {
        $(".manual").css("background", data);
    });

```

```
var url22 = "/Proceso/obtenerPresion";

$.get(url22, null, function (data) {
    $("#metPres").attr('value', data);
    $("#presion").html(data);
});

var url23 = "/Proceso/obtenerTemperatura";

$.get(url23, null, function (data) {
    $("#metTemp").attr('value', data);
    $("#temperatura").html(data);
});

var url24 = "/Proceso/obtenerPH";

$.get(url24, null, function (data) {
    $("#metPH").attr('value', data);
    $("#ph").html(data);
});

var url25 = "/Proceso/obtenerTempMax";

$.get(url25, null, function (data) {
    $("#tmaxima").html(data + "°C");
    $("#mtm").html(data + "°C");
    $("#metTemp").attr('max', data);
});

var url26 = "/Proceso/obtenerTempMin";
```

```
$.get(url26, null, function (data) {  
    $("#tminima").html(data + "°C");  
    $("#minTemp").html(data + "°C");  
    $("#metTemp").attr('min', data);  
});  
var url27 = "/Proceso/obtenerpHMax";  
$.get(url27, null, function (data) {  
    $("#phmx").html(data);  
});  
var url28 = "/Proceso/obtenerpHMin";  
$.get(url28, null, function (data) {  
    $("#phmn").html(data);  
});  
var url29 = "/Proceso/obtenerPresMax";  
$.get(url29, null, function (data) {  
    $("#presimax").html(data + "psi");  
    $("#mpm").html(data + "psi");  
    $("#metPres").attr('max', data);  
});  
}, 1000);
```