

UNIVERSIDAD TÉCNICA DE AMBATO



**FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN,
TELECOMUNICACIONES E INDUSTRIAL**

**MAESTRÍA EN AUTOMATIZACIÓN Y SISTEMAS DE
CONTROL**

Tema:

**“DISEÑO DE UNA PLATAFORMA WEB DE SOFTWARE LIBRE PARA LA
CREACIÓN DE HMIS PARA LOS PLC S7-1200 Y SU INCIDENCIA EN LA
REDUCCIÓN DE COSTOS DE MONITOREO”**

Trabajo de Investigación, Previo a la obtención del Grado Académico de Magister en
Automatización y Sistemas de Control.

AUTOR: Ingeniero Alex Vladimir Nuñez Ramires
DIRECTOR: Ingeniero German Patricio Encalada Ruiz, Mg.

Ambato – Ecuador
Junio - 2019

A la Unidad Académica de Titulación de la Facultad de Tecnologías de la Información, Telecomunicaciones e Industrial.

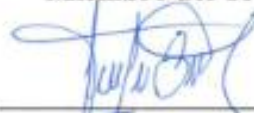
El Tribunal receptor del Trabajo de Investigación presidido por la Ingeniera Elsa Pilar Urrutia Urrutia Mg., e integrado por los señores: Ingeniero Marcelo Vladimir García Sánchez, PhD., Ingeniero Franklin Wilfrido Salazar Logroño Mg. e Ingeniero Carlos Diego Gordón Gallegos PhD., designados por la Dirección de Postgrado de la Universidad Técnica de Ambato, para receptor el Trabajo de Investigación con el tema: **“DISEÑO DE UNA PLATAFORMA WEB DE SOFTWARE LIBRE PARA LA CREACIÓN DE HMIS PARA LOS PLC S7-1200 Y SU INCIDENCIA EN LA REDUCCIÓN DE COSTOS DE MONITOREO”**, elaborado y presentado por el Ingeniero Alex Vladimir Nuñez Ramires, para optar por el Grado Académico de Magister en Automatización y Sistemas de Control; una vez escuchada la defensa oral del Trabajo de Investigación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la UTA.



Ing. Elsa Pilar Urrutia Urrutia Mg.
Presidente del Tribunal



Ing. Marcelo Vladimir García Sánchez, PhD.
Miembro del Tribunal



Ing. Franklin Wilfrido Salazar Logroño Mg.
Miembro del Tribunal



Ing. Carlos Diego Gordón Gallegos, PhD.
Miembro del Tribunal

AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

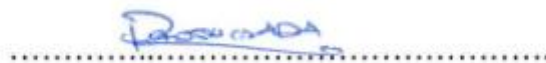
La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo de Investigación con el tema: “**DISEÑO DE UNA PLATAFORMA WEB DE SOFTWARE LIBRE PARA LA CREACIÓN DE HMIS PARA LOS PLC S7-1200 Y SU INCIDENCIA EN LA REDUCCIÓN DE COSTOS DE MONITOREO**”, le corresponde exclusivamente al Ingeniero Alex Vladimir Nuñez Ramires, Autor, bajo la dirección del Ingeniero Patricio German Encalada Ruiz Mg., Director del Trabajo de Investigación; y el patrimonio intelectual del mismo a la Universidad Técnica de Ambato.



Ingeniero Alex Vladimir Núñez Ramires

C.C.: 1803942000

AUTOR



Ingeniero German Patricio Encalada Ruiz Mg.

C.C.: 1803881935

DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnico de Ambato, para que el Trabajo de Investigación, sirva como documento disponible para su lectura, consulta y procesos de investigación, según las normas de la institución.

Cedo los Derechos de mi Trabajo, con fines de difusión pública, además apruebo la reproducción de esta, dentro de las regulaciones de la Universidad.



Ingeniero Alex Vladimir Núñez Ramires

C.C.: 1803942000

AUTOR

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
A la Unidad Académica de Titulación	ii
AUTORÍA DEL TRABAJO DE INVESTIGACIÓN	iii
DERECHOS DE AUTOR	iv
ÍNDICE GENERAL DE CONTENIDOS	v
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	x
DEDICATORIA	xii
AGRADECIMIENTO	xiii
RESUMEN EJECUTIVO	xiii
EXECUTIVE SUMMARY	xv
INTRODUCCIÓN	1
CAPITULO I	3
1.1 Tema de investigación	3
1.2 Planteamiento del Problema	3
1.2.1 Contextualización	3
1.2.2 Árbol del Problema	5
1.2.3 Análisis Crítico	5
1.2.4 Prognosis	6
1.2.5 Formulación del Problema	6
1.2.6 Interrogantes de la investigación	6
1.2.7 Delimitación del problema	7
1.3 Justificación	7
1.4 Objetivos	9
1.3.1 Objetivo General	9
1.4.2 Objetivos específicos	9
CAPÍTULO II	10
2.1 Antecedentes Investigativos	10
2.2 Fundamentación Filosófica	13
2.3 Categorías fundamentales	13

2.3.1	Red de inclusiones Conceptuales	13
2.3.2	Constelación de ideas	14
2.3.3	Sistema SCADA	15
2.3.4	Componentes de un SCADA	16
2.3.5	Interfaz hombre maquina (HMI)	19
2.3.6	Necesidades de un Sistema SCADA	24
2.3.7	Funciones de un Sistema SCADA	24
2.3.8	Equipos de control	25
2.3.9	Protocolos de comunicación	27
2.3.10	Plataformas Web	29
2.4	Hipótesis	31
2.5	Señalamiento de variables de la hipótesis	31
CAPÍTULO III		32
3.1	Enfoque	32
3.2	Modalidad básica de la información	32
3.3	Operacionalización de variables	33
3.4	Recolección de información	37
3.5	Procesamiento y análisis	37
CAPÍTULO IV		38
4.1	Análisis de Resultados	38
4.1.1	Lenguajes y herramientas de programación web	38
4.1.2	HTML para programación de páginas Web	38
4.1.3	CSS para programación de páginas Web	40
4.1.4	JavaScript para programación de páginas web	42
4.1.5	AJAX	45
4.1.6	JQuery	48
4.1.7	JSON	50
4.1.8	OPC UA	51
4.1.9	IDEs de programación web	53
4.1.10	Norma ISA 101 para la creación de interfaces hombre maquina (HMI)..	55
CAPÍTULO V		58
5.1	Conclusiones	58
5.2	Recomendaciones	60

CAPÍTULO VI	61
6.1 Antecedentes de la propuesta.....	61
6.2 Justificación	63
6.3 Objetivos de la propuesta.....	65
6.3.1 Objetivo General.....	65
6.3.2 Objetivos Específicos	65
6.4 Análisis de factibilidad	65
6.4.1 Factibilidad Operativa	65
6.4.2 Factibilidad Técnica	65
6.4.3 Factibilidad Económica	66
6.4.4 Proyección a futuro.....	66
6.5 Fundamentación Científico Técnica	66
6.5.1 IDE de programación seleccionado.....	66
6.5.2 PLC S7-1200 y TÍA PORTAL	67
6.5.3 Elección del protocolo de comunicación.....	67
6.5.4 Diagrama de funcionamiento de la plataforma web.....	68
6.5.5 Diseño de la estructura de la plataforma mediante HTML	70
6.5.6 Diseño de la apariencia de la plataforma mediante CSS	72
6.5.7 Diseño de los elementos virtuales	72
6.5.8 Diseño de la comunicación con el protocolo OPC UA.....	83
1.5 Comprobación de la Hipótesis de la propuesta.....	100
6.8 Conclusiones y Recomendaciones	111
BIBLIOGRAFÍA	113

ÍNDICE DE FIGURAS

Fig. 1. 1: Relación Causa-Efecto	5
Fig. 2. 1: Variable dependientes e independientes de las categorías fundamentales.....	13
Fig. 2. 2: Constelación de ideas de la Variable independiente	14
Fig. 2. 3: Constelación de la variable dependiente	14
Fig. 2. 4 Sistema SCADA.....	16
Fig. 2. 5: Sistema de Redundancia	17
Fig. 2. 6: Arquitectura General de un RTU	18
Fig. 2. 7: Partes que constituyen un PLC (Controlador Lógico Programable).....	25
Fig. 2. 8: Clasificación de los actuadores	27
Fig. 4. 1: Estructura de una Regla CSS.	41
Fig. 6. 1: Diagrama de funcionamiento de la plataforma Web.....	68
Fig. 6. 2: Estructura plataforma web empleando HTML	70
Fig. 6. 3. Descripción código HTML	71
Fig. 6. 4. Aplicación de estilo visual CSS.	72
Fig. 6. 5. Adición de las dependencias para la utilización de la librería jqxgauge.....	73
Fig. 6. 6: Código del control gauge	74
Fig. 6. 7: Control Gauge	74
Fig. 6. 8: Creación del contenedor del control y posicionamiento	75
Fig. 6. 9: Creación del cuadro de propiedades del control	77
Fig. 6. 10: Código de la sub rutina que establece la variable a monitorear	78
Fig. 6. 11: Código de la sub rutina que agrega un nuevo control al área de trabajo	80
Fig. 6. 12: Código de la subrutina para cambiar el tamaño del control.....	81

Fig. 6. 13: Subrutina para cerrar el cuadro de propiedades	82
Fig. 6. 14: Código de la sub rutina remover control.....	82
Fig. 6. 15: Código de la sub rutina para desplazar el control	83
Fig. 6. 16: Diagrama de flujo de funcionamiento del Cliente OPC UA.....	84
Fig. 6. 17: Establecimientos de parámetros del protocolo OPC UA	85
Fig. 6. 18: Código que permite establecer la conexión con el servidor OPC UA.....	86
Fig. 6. 19: Código que envía la información de las variables al servidor OPC UA.....	87
Fig. 6. 20: Código que nos permite obtener las variables contenidas en el servidor OPC UA	88
Fig. 6. 21: Código que permite monitorear una variable del servidor OPC UA.	89
Fig. 6. 22: Código que escribe la información en una variable del servidor OPC UA.	90
Fig. 6. 23: Diagrama de flujo de funcionamiento del Servidor OPC UA.	91
Fig. 6. 24: Establecimiento de parámetros del servidor OPC UA.....	92
Fig. 6. 25: Código para crear las variables a ser monitoreadas.	92
Fig. 6. 26: Código para escribir las variables del PLC.....	93
Fig. 6. 27: Pruebas de conexión.....	94
Fig. 6. 28: Monitorización de variables.....	95
Fig. 6. 29: Escritura de variables.	95
Fig. 6. 30: Esquema para la captura del tráfico de la red.	96
Fig. 6. 31: Captura de trafico al iniciar el cliente y servidor OPC UA	96
Fig. 6. 32: Detalle del protocolo de transmisión.....	97
Fig. 6. 33: Grafica de Ancho de banda (bits/seg)	98
Fig. 6. 34: Grafica de la transmisión de paquetes.....	99
Fig. 6. 35: Tiempo de procesamiento de paquetes	99
Fig. 6. 36: Gráfica del calculo del Chi-cuadrado.....	103

ÍNDICE DE TABLAS

Tabla 3- 1. Variable independiente: Plataforma Web de software libre para crear HMIs	33
Tabla 3- 2. Variable dependiente: Costos de monitorización.....	35
Tabla 4- 1: Etiquetas del lenguaje HTML.....	39
Tabla 4- 2: Eventos definidos por JavaScript.....	45
Tabla 6- 1: Detalle cuantitativo del análisis de tráfico de la Red.....	100
Tabla 6- 2: Frecuencias observadas del sistema.....	101
Tabla 6- 3: Frecuencias esperadas del sistema.....	102
Tabla 6- 4: Cuadro comparativo de características.....	104
Tabla 6- 5: Valoración de características de los software para creación de pantallas HMI	107
Tabla 6- 6: Análisis de costos de softwares para creación de pantallas HMI.....	108

DEDICATORIA

A toda mi familia.

En especial a mis padres, y mi tía

Que han sido mi apoyo y mi sustento

AGRADECIMIENTO

*A mis padres, a mis hermanas y a
toda mi familia por su paciencia,
apoyo y comprensión.*

A Dios por todas sus bendiciones

UNIVERSIDAD TÉCNICA DE

AMBATO

**FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN,
TELECOMUNICACIONES E INDUSTRIAL / DIRECCIÓN DE POSGRADO**

MAESTRÍA EN AUTOMATIZACIÓN DE SISTEMAS DE CONTROL

TEMA:

DISEÑO DE UNA PLATAFORMA WEB DE SOFTWARE LIBRE PARA LA
CREACIÓN DE HMIS PARA LOS PLC S7-1200 Y SU INCIDENCIA EN LA
REDUCCIÓN DE COSTOS DE MONITOREO.

AUTOR: Ingeniero Alex Vladimir Nuñez Ramires.

DIRECTOR: Ingeniero German Patricio Encalada Ruiz, Mg.

FECHA: 18 de Enero del 2019

RESUMEN EJECUTIVO

En el presente trabajo de investigación se evalúa una nueva propuesta para la creación de HMIs a través de una plataforma web de acceso libre. Se analizó la incidencia en el costo de monitorización que provee el trabajo de investigación propuesto frente a los sistemas convencionales de pago. El proyecto se basó en estudiar, analizar y detectar las principales necesidades de una plataforma web para la creación de HMIs enfocados a la monitorización de diferentes procesos de la industria.

Adicionalmente se realizó una investigación sobre la mejor tecnología para establecer la comunicación entre el controlador y la plataforma web, optando como mejor opción la utilización del protocolo OPC UA. Mismo que permite proveer mejoras en cuanto a seguridad de la información, así como mejoras en la gestión de las variables a monitorizarse y tiempos de respuestas de comunicación. Mediante la utilización de este protocolo, las técnicas y lenguajes de programación web estudiados tales como: HTML, CSS, JavaScript, AJAX y JQuery se hace posible la integración de los procesos industriales y el internet, permitiendo la interconexión de cada vez más dispositivos a la red. Lo cual plantea un gran avance en la industria 4.0.

Descriptores: HMIs, HTML, CSS, JavaScript, Sistemas de libre acceso, Sistemas propietarios, Protocolo OPC UA, Industria 4.0, AJAX, JQuery

UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL / DIRECCIÓN DE POSGRADO**

MAESTRÍA EN AUTOMATIZACIÓN Y SISTEMAS DE CONTROL

THEME:

DESIGN OF A FREE SOFTWARE WEB PLATFORM TO CREATE HMIS FOR THE
S7-1200 PLC AND ITS INCIDENCE IN THE REDUCTION OF MONITORING
COSTS.

AUTHOR: Engenieer Alex Vladimir Nuñez Ramires.

DIRECTED BY: Engenieer German Patricio Encalada, Mg.

DATE: January 18, 2019

EXECUTIVE SUMMARY

In the present research work, a new proposal for the creation of Human Machine Interfaces (HMIs) is evaluated through a free access web platform. The incidence in the cost of monitoring provided by the proposed research work compared to conventional payment systems was analyzed. The project was based on studying, analyzing and detecting the main needs of a web platform for the creation of HMIs focused on the monitoring of different processes in the industry.

Additionally, an investigation was carried out on the best technology to establish communication between the controller and the web platform, opting as the best option using the OPC UA protocol. Same that allows to provide improvements in terms of information security, as well as improvements in the management of the variables to be monitored and communication response times. By using this protocol, the techniques and web programming languages studied such as: HTML, CSS, JavaScript, AJAX and JQuery make possible the integration of industrial processes and the internet, allowing the interconnection of more and more devices to the net. Which poses a breakthrough in industry 4.0.

Descriptors: HMIs, HTML, CSS, JavaScript, Open Access Systems, Proprietary Systems, OPC UA Protocol, Industry 4.0, AJAX, JQuery

INTRODUCCIÓN

En el presente trabajo de investigación se estudian las técnicas de programación web actuales. Con la finalidad de construir una alternativa a los softwares existentes para la creación de HMIs. Este proyecto tiene como objetivo proveer al usuario final un sistema basado en servicios web, de libre acceso y de bajo costo. Se ha recopilado una gran cantidad de información acerca de estos sistemas y se ha planteado una solución basándose en trabajos similares. Partiendo de la creación de una interfaz amigable de fácil uso. Dotándole de un aspecto visual estético y atractivo y diseñando instrumentos virtuales que permitan interactuar con la información proveniente del controlador. Además de estudiar la forma de establecer una comunicación en tiempo real entre el controlador y la plataforma web.

El Capítulo I describe la problemática actual existente alrededor del tema de investigación. Además se justifica el propósito de investigación acerca del tema. Planteándose preguntas acerca de los motivos que desencadenan la problemática a estudiar y resolver. Se realiza además un análisis crítico del tema generando los objetivos a cumplirse a lo largo del desarrollo del presente proyecto de investigación.

El Capítulo II analiza la información recolectada sobre trabajos similares desarrollados en el ámbito local e internacional. Información que permitirá orientar al presente proyecto en la dirección correcta, de manera que aporte a la solución de problemas actuales y futuros en el campo de estudio. Formando parte de los trabajos a tomar en consideración para futuras investigaciones. Es por ello que se analiza las estructuras necesarias para construcción de este tipo de sistemas. Llegando a establecer una hipótesis y a identificar las variables que intervienen en el desarrollo del mismo.

El Capítulo III detalla la metodología a seguir para la realización del presente trabajo de investigación. Creando tablas en donde se identifican los factores más relevantes a estudiar y se generan preguntas acerca de que herramientas de programación web serán necesarias para llevar a cabo el proyecto propuesto. Además se intenta delimitar el proyecto hacia una solución adecuada y factible. Finalizando en el establecimiento de estrategias para la recolección de la información a ser estudiada.

El Capítulo IV resuelve las dudas y preguntas planteadas en el Capítulo III describiendo todas las técnicas de programación necesarias para la consecución del presente trabajo de investigación. En este capítulo se describen los métodos, lenguajes, librerías y técnicas de programación necesarias para crear la propuesta planteada.

En el Capítulo V se presentan las conclusiones y recomendaciones que se determinaron a través del estudio de la información presentada en el Capítulo IV. Estableciendo puntos de partida claros hacia la solución y desarrollo del proyecto.

El Capítulo VI es la culminación de todo el estudio y procesamiento de la información recolecta. En este capítulo se describe el desarrollo del proyecto. Empezando por los softwares e IDEs de programación utilizados, siguiendo con la descripción paso a paso del desarrollo de la plataforma web mediante HTML, CSS y JAVASCRIPT. Posteriormente se describe el proceso de intercambio de información entre la plataforma web y el controlador a través del protocolo OPC UA. Terminando con la presentación de las pruebas de funcionamiento del proyecto las conclusiones y recomendación.

CAPITULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Tema de investigación

Diseño de una plataforma web de software libre para la creación de HMIs para los PLC S7-1200 y su incidencia en la reducción de costos de monitoreo.

1.2 Planteamiento del Problema

1.2.1 Contextualización

El desarrollo de las interfaces hombre maquina (HMI) ha surgido en relación con los avances tecnológicos y el desarrollo humano. Considerando como HMI desde las primeras interfaces diseñadas para las computadoras de primera generación hasta las interfaces modernas de última tecnología que poseemos en la actualidad. Aunque el desarrollo en el campo de la informática ha sido breve en gran medida gracias al apareamiento del internet, la estructura básica de las interfaces hombre maquina sigue siendo la misma. Este desarrollo acelerado de la tecnología del cual la humanidad ha sido participe en las últimas décadas ha traído consigo avances importantes siendo cada vez más común el apareamiento de dispositivos autónomos y procesos automatizados complejos. Es por ello que el desarrollo de nuevas tecnologías e iniciativas para la creación de interfaces hombre maquina están siendo desarrolladas.(Parga Carlos Roman, 2013; Pasquale, Kim, & Pasquale, 2015)

Entre los desarrollos más importantes que han sufrido las interfaces hombre maquina en los últimos años se destacan los avances en las comunicaciones y en la capacidad de integración con otros sistemas. Estos avances han provocado que esta tecnología se masifique, puesto que ahora es mucho más accesible. Por lo cual hoy en día podemos encontrar una amplia gama de opciones de sistemas HMI en el mercado, orientados a satisfacer distintos niveles de necesidades. Poniendo a disposición desde sistemas HMI básicos, que pueden ser simplemente una interfaz de panel con una mínima cantidad de datos hasta llegar a las soluciones empaquetadas, a las que pueden agregárseles herramientas de espectro más amplio. Lo cual ha

hecho que la mayoría de las empresas, sin importar su tamaño, no conciben sus procesos sin herramientas de este tipo.(Revista Electroindustria, 2007)

El desarrollo de esta tecnología y la masificación de los productos han hecho posible que ya no sean un tema aparatado sino que tengan una participación mucho más grande en la industria. Siendo así que estos dispositivos ya no sean simplemente de uso específico en países de primer mundo como Estados Unidos, Alemania, Inglaterra entre muchos más países industrializados. Sino que en la actualidad se puede ver cómo han tenido una acogida grande por las Pymes en países de Latinoamérica como Chile o Brasil. Lo cual ha contribuido a la creación de mejores interfaces simplificando los procesos y haciéndolas más amigables para los usuarios finales.

En la actualidad, los HMI ofrecen incluso plataformas cliente-servidor multidisciplinarias, es decir, que son capaces de aplicarse a un SCADA eléctrico, a un sistema de procesos o a una pequeña planta. Los sistemas HMI ya no son sólo un panel que actúa como interfaz, sino que tienen mucha más potencialidad: pueden mandar correos y alarmas al gerente de planta y unir ese control de eventos con la gestión de manera que los pequeños empresarios puedan acceder a grandes funcionalidades a partir de un HMI. Una de las grandes ventajas es que se configuran de acuerdo a cómo se necesiten. Sin embargo, hay que ser cuidadosos en que los operarios sean capaces de incorporarlos adecuadamente.(David, 2011)

Si bien es cierto que este tipo de tecnología se ha dado a conocer de gran manera en la actualidad. En la industria nacional aún no existe la acogida esperada por parte de las empresas nacionales. Esto se debe en gran medida a que los costos de implementar este tipo sistemas representa una alta barrera a superar, así como la necesidad de personal capacitada para poder operar este tipo de herramientas. Es por ello que muy pocas empresas constan con este tipo de sistemas para la monitorización de sus procesos. Por lo cual es necesario la creación de sistemas enfocados a superar estas limitaciones presentando interfaces sencillas de utilizar haciendo uso de herramientas de código libre que permitan disminuir los costos que supone la utilización de este tipo de sistemas.(Gualotuña Barahona & Villagómez Palacios, 2014)

1.2.2 Árbol del Problema

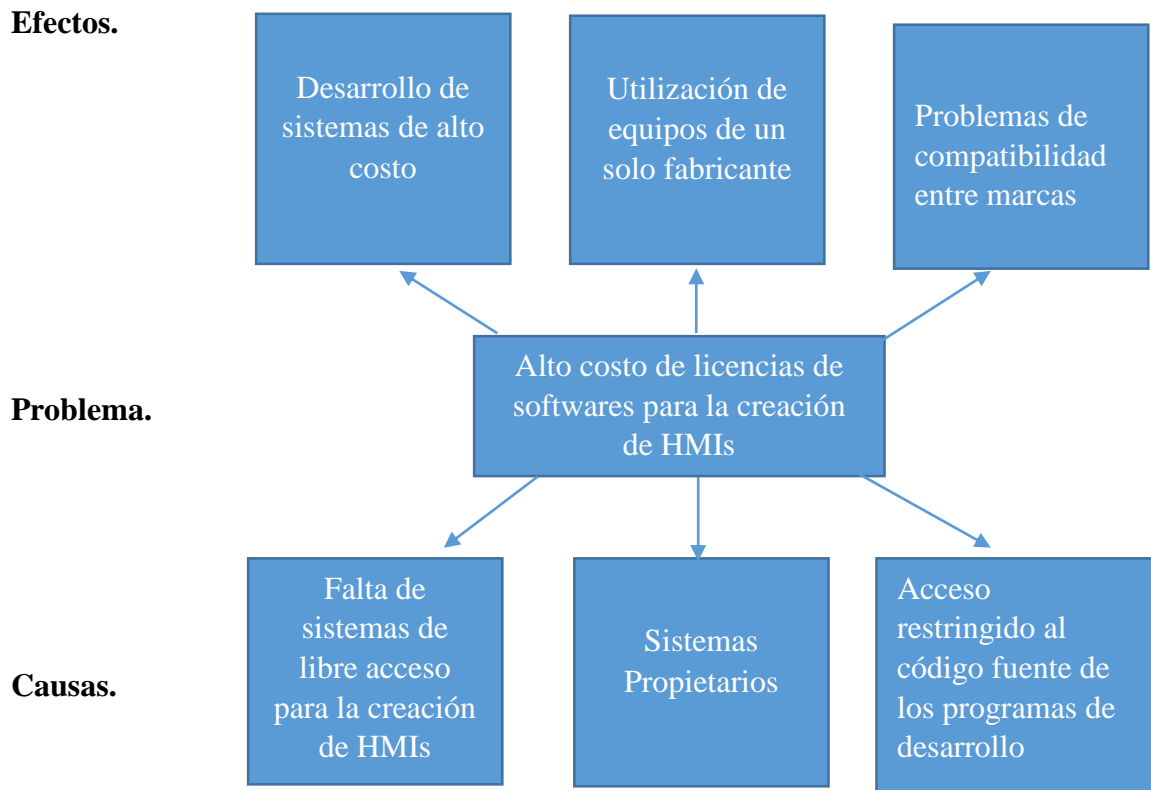


Fig. 1. 1: Relación Causa-Efecto

Elaborado por: El Investigador

1.2.3 Análisis Crítico

En el Ecuador desde ya hace algunos años se busca incrementar la matriz productiva del país lo cual implica un crecimiento de la industria. Esto a su vez implica el mejoramiento de maquinaria y tecnología. Con la tendencia a mejorar, automatizar y simplificar los procesos. Es por ello que se necesita tanto hardware como software a costos competitivos.

Muchas de las industrias ecuatorianas siguen realizando varios procesos de forma manual pudiendo estos ser automatizados sin ningún problema. Esto se debe a que muchas veces el costo beneficio de mejorar sus industrias no representa una ventaja. Esta falta de implementación tecnológica provoca que muchas de estas industrias no puedan alcanzar un buen desarrollo y de esa manera impulsar la matriz productiva del país.

En cuanto a lo que se refiere a software es mucho más complejo, debido a que en el país las industrias tienen que utilizar equipos de ciertas marcas y con sus respectivas licencias. Las cuales muchas de las veces son de un costo sumamente alto. Provocando que muchas empresas en particular las pequeñas no pueden costear estos gastos de infraestructura. Por lo cual el lado de monitorio y software muchas veces queda relegado a segundo plano. Es por ello que la propuesta de una plataforma libre para la creación de HMIs presentaría grandes ventajas a la industria ya que se mejoraría precios de producción y se evitaría el pago de licencias.

1.2.4 Prognosis

Si no se encuentran alternativas viables en lo que se refiere a softwares para creación de pantallas de monitorización HMIs los costos de monitorización de los procesos seguirán siendo elevados, y consecuentemente relegados.

Esta falta de implementación tecnológica no permitirá que las empresas tengan un punto de crecimiento esperado. Lo cual implica a su vez una falta de producción o seguir produciendo lo mismo que se producía. Con lo cual no se podrá obtener más ingresos económicos.

Al producirse reducción en los ingresos económicos no se podrá mejorar la calidad de los productos fabricados, así como el tiempo de fabricación y sus costos. Al no existir una implementación tecnológica adecuada especialmente en el área de monitorización tampoco se podrá reducir las fallas en producción.

1.2.5 Formulación del Problema

¿Cómo influye un sistema libre para la creación de HMIs en la reducción de costos de monitorización?

1.2.6 Interrogantes de la investigación

¿Qué problemas conlleva la falta de tecnología en la industria?

¿Existe alguna manera de mejorar el rendimiento de la industria?

¿De qué manera incide un proceso de monitorización de libre acceso en el desarrollo de las empresas?

1.2.7 Delimitación del problema

- **Límite de contenido**

Área Académica: Ingeniería Electrónica, Automatización, HMIs.

Línea de investigación: Automatización.

- **Delimitación espacial**

Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato.

- **Delimitación temporal**

El trabajo de investigación está estimado a ser finalizado dentro de 6 meses, iniciando luego de la aprobación del Honorable Consejo Académico de la Facultad de ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.3 Justificación

En la actualidad las interfaces de monitorización son de vital importancia en los procesos industriales modernos y de manufactura. Debido a que proveen un desempeño óptimo de los sistemas dinámicos mejorando la producción, disminuyendo la repetitividad, las operaciones rutinarias y disminuyendo considerablemente la probabilidad de error. Las necesidades actuales que se presentan en el mundo de la automatización tiene que ver con la forma en que los seres humanos interactúan y se relación con las maquinas. Siempre con el objetivo de hacer un óptimo aprovechamiento de las tecnologías informáticas. Es por este motivo que se propuso el presente tema de investigación con el propósito de emplear una nueva forma de crear HMIs que se adapten a las necesidades del cliente final sin perder calidad de visualización y confiabilidad.(Arenas L, Castilla A, 2015)

Anteriormente las páginas web solían ser de un diseño visual muy limitado y poco atractivo pero el gran avance que ha tenido el internet ha hecho que se considere la creación de pantallas

de monitorización mediante la utilización de programación web. Ya que hoy en día podemos encontrar páginas web con un estilo visual realmente bueno con gráficos de alta calidad y una presentación de la información excelente. Lo cual hace al internet la plataforma ideal para desarrollar unos nuevos modelos de interfaces hombre maquina basado estos servicios. Además de aprovechar el gran avance que está teniendo el desarrollo de nuevos métodos de intercambio de información basados en la nube. Esto a su vez supondría una integración con la industria 4.0 que está teniendo un gran avance en la actualidad. Es por ello que la mayoría de los procesos ahora suelen suceder a través de internet. Un proyecto como el planteado abriría la puerta a nuevos proyectos e investigaciones integrando estas nuevas herramientas y colaborando en el mejoramiento de los nuevos protocolos de comunicación orientados a la automatización y la monitorización de procesos como es el caso de OPC UA, Mqtt, Http, AMQP entre muchos otros más.(Gorecky, Schmitt, Loskyll, & Zühlke, 2014)

A nivel internacional se han desarrollado pocos proyectos similares al propuesto ya que aún es una tecnología nueva que está en desarrollo. Una de las propuestas planteadas es la iniciativa Node Red que permite establecer comunicación con sensores y actuadores a través de los diferentes protocolos antes mencionado. Aunque esta plataforma permite crear varios proyectos esta aun distante de la propuesta planteada que es crear pantallas de monitorización de manera sencilla sin la necesidad de introducir líneas de código. (JS Foundation, 2019; Rajalakshmi & Shahnasser, 2018)

A nivel de Latino América se encuentra más difundida la plataforma Open Automation Software. Esta plataforma permite crear pantallas de monitorización web a través de la utilización de OPC, Mqtt, API REST y muchos otros protocolos más. Aunque las herramientas que posee para la creación de los proyectos es bastante extensa la creación de los mismos es bastante complicada ya que aún se debe realizar la programación web la cual a su vez se muestra con las librerías de los drivers y protocolos utilizados para establecer comunicación con los controladores.(Open Automation Software, 2019)

A nivel de Ecuador y la región central no se encontraron trabajos similares que permitan crear HMIs a través de la web de manera sencilla y sobre todo que su costo sea accesible. Por lo cual la realización del tema de investigación planteado contribuiría a desarrollar sistemas que le

permitan a la industria nacional hacer uso de estas herramientas, a costos accesibles y asemejándose a la confiabilidad que proveen los sistemas propietarios. Además permitiría ahondar en el estudio y desarrollo tanto de nuevas técnicas herramientas para el desarrollo de pantallas de monitorización así como de los protocolos de la industria 4.0

1.4 Objetivos

1.3.1 Objetivo General

Diseñar una plataforma web de libre acceso para la creación de HMIs web para los PLC S7-1200.

1.4.2 Objetivos específicos

- Analizar las normas y estándares de presentación de datos para la presentación de HMIs.
- Diseñar una plataforma web libre para la creación de HMIs
- Hacer un análisis de reducción de costos para la creación de HMIs, con la herramienta diseñada.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes Investigativos

De la investigación realizada se concluyó que existen pocos trabajos desarrollados tanto por estudiantes como instituciones referentes al tema de los cuales citaremos los siguientes a continuación:

El trabajo realizado por Raivo Sell en colaboración con Tauno Otto denominado “Online Engineering for Future Factory” desarrollado en la Universidad de Missouri de Estados Unidos describe las posibilidades de la ingeniería online orientado a instituciones de educación superior. El cual busca crear un entorno integrado de herramientas multidisciplinarias combinando las áreas de Mecatrónica, Robótica, Manufactura y Automatización. En el cual se estableció una infraestructura a través de un plugin de java que permite a los usuarios acceder a un PLC y controlarlo mediante un sitio web (SELL & OTTO, 2009). La propuesta presentada en este trabajo es de gran interés y relevancia en relación con el trabajo de investigación a desarrollarse ya que permite conocer ciertas alternativas de comunicación entre las aplicaciones web y los dispositivos terminales. Este permitirá poseer una idea más clara al momento de optar por una determinada tecnología para establecer el intercambio de información en el trabajo de investigación planteado.

Otro trabajo relacionado es el realizado por Ayyaz Ismail Nadaf, M. K. Bhanarkar y Prashant Prabhakar Shah denominado “Review Paper on AJAX Comet and Websocket Uses for Web HMI/SCADA”. Este trabajo describe los principios básicos de la utilización de JavaScript y HTML5 en el intercambio de información con dispositivos industriales como PLCs a través de internet con lo cual se pueden diseñar plataformas para el control y monitorización de diferentes ambientes de trabajo (Ayyaz Ismail, M. K., & Prashant Prabhakar, 2015). El análisis de las técnicas de programación utilizadas en este trabajo se convertirá en el punto de partida para desarrollar la propuesta planteada, ya que provee información valiosa acerca del diseño de las interfaces web y la comunicación a través del protocolo socket.io indispensable en el intercambio de información entre páginas y aplicaciones web. Lo cual permitirá establecer un

esquema básico para la creación de la interfaz web de la aplicación y ayudara a entender los principios básicos de HTML5 y JavaScript indispensables para el desarrollo de los instrumentos virtuales.

De igual manera el trabajo de investigación desarrollado por Traian Turc denominado “Using Web Services in SCADA Applications”, describen el proceso de comunicación y acceso a la información a través de los servicios web, lidiando de esta manera con el problema de tener grandes bases de datos y creando así HMI para el monitorización de procesos (Turc, 2015). De igual manera que el trabajo presentado por Ayyaz Ismail Nadaf, M. K. Bhanarkar y Prashant Prabhakar Shah descrito anteriormente este trabajo permitirá conocer más alternativas de comunicación entre aplicaciones web para desarrollar HMIs. Lo cual permitirá tener unos conocimientos más amplios sobre estas herramientas de comunicación otorgando la oportunidad de realizar un mejor análisis al momento de escoger la herramienta de comunicación adecuada para desarrollar el trabajo de investigación planteado.

Finalmente tenemos el trabajo desarrollado por Hosny A. Abbas y Ahmed M. Mohamed denominado “Review on the Design of Web Based SCADA Systems Base on OPC DA Protocol”, en el cual se expone la idea de interactuar entre los servicios web y OPC a través de XML para interactuar con los controladores lógicos programables PLCs (Abbas & Mohamed, 2011). Este trabajo nos presenta una nueva manera de interactuar con los servicios web y el protocolo OPC, abriendo una nueva visión hacia la utilización de protocolos para establecer la comunicación en entre las aplicaciones web y los dispositivos terminales. Lo cual nos permite tener un mejor entendimiento de cómo funcionan las comunicaciones en este tipo de sistemas y además abre el campo hacia la idea de estudiar nuevos protocolos con el objetivo de proveer la mejor alternativa para el proyecto de investigación planteado.

Alternativamente a los trabajos mencionados anteriormente existen ciertas plataformas web desarrolladas por empresas las mismas que permiten realizar HMI Webs las cuales serán descritas a continuación.

Una de las plataformas desarrollada para la creación de HMI Webs se denomina Open Automation Software, es una plataforma que permite crear interfaces para desplegar, visualizar

e interactuar con servidores de datos en tiempo real, permitiendo un gran rango de aplicaciones en un dispositivo con un web browser. Esta plataforma usa varios lenguajes y herramientas de programación web tales como: HTML, CSS, Javascript, JQuery y Json. Permitiendo que los usuarios puedan utilizar sus conocimientos de programación para personalizar sus diseños. Además se puede interconectar con varias marcas y modelos de PLCs entre ellas Allen Bradley y Siemens (Open Automation Web Site, 2018). El estudio de un sistema propietario como el presente nos permitirá tener idea clara del producto que se pretende ofrecer al usuario final. Lo cual permitirá crear tanto interfaces como controles virtuales enfocados a prestar una experiencia agradable al usuario, permitiendo además conocer la manera en la cual lo hace un software más sofisticado y emulando varias las características y pasos desarrollados en este tipo de sistemas.

Otra plataforma para la creación de HMIs web SCADAs es Orion WebServer el cual permite construir pantallas interactivas accesibles desde cualquier navegador web. A través del cual nos podemos conectar con Orion y visualizar la información de diferentes dispositivos. Pudiendo visualizar pantallas tan simples con mensajes de textos hasta pantallas detalladas con gráficas y múltiples accesos de seguridad. Proveyendo de esta manera control de pantallas, alarmas, trends, y diagnósticos (NovaTech, 2015). Al igual que el trabajo anterior esta iniciativa permitirá conocer sobre todo cual es el aspecto visual que este tipo de sistemas pretenden ofrecer al usuario final. Lo cual se podrá tomar como referencia para el desarrollo de los elementos e interfaces del proyecto de investigación planteado.

Estas investigaciones han sentado el precedente para la creación de proyecto de investigación. En las primeras investigaciones se hace un estudio de las técnicas de programación así como de ciertas herramientas que permitirán desarrollar las estructuras de los controles web a desarrollarse. Además se presenta una idea de ciertos servicios web y protocolos utilizados como OPC DA. Herramientas que pueden ser útiles para establecer la comunicación con los diferentes controladores. Mientras que las investigaciones y proyectos desarrolladas por las empresas Software Automation Software y Orion WebServer permite tener una idea de cómo debe ser el aspecto visual de las aplicaciones a desarrollarse.

2.2 Fundamentación Filosófica

La electrónica actual y los sistemas de control nos permiten automatizar y monitorear procesos; lo cual anteriormente no se realizaba. Brindándonos una gran ventaja para el desarrollo de las diferentes industrias ya que la monitorización de las actividades realizadas por las mismas, implica un avance y el mejoramiento en la eficiencia de sus procesos.

Al ser más eficientes para recolectar y presentar información lograra que las industrias disminuyan errores en la realización de los procesos de manufactura. Se mejorara tiempos de trabajo, y se lograra disminuir la complejidad del trabajo realizado y por lo tanto se reducirá costos y se aumentara la eficiencia. Además se incrementara la calidad de los productos realizados. Esto se logra mediante la monitorización, la automatización y precisión de los dispositivos utilizados.

2.3 Categorías fundamentales

2.3.1 Red de inclusiones Conceptuales

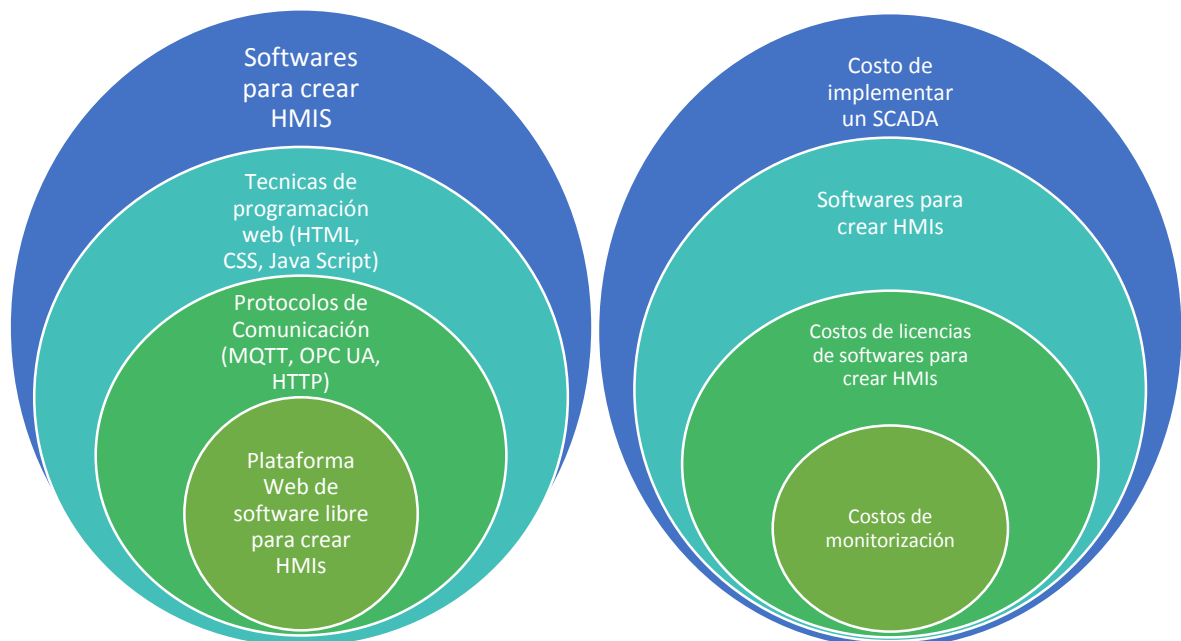


Fig. 2. 1: Variable dependientes e independientes de las categorías fundamentales

Elaborado por: El Investigador

2.3.2 Constelación de ideas

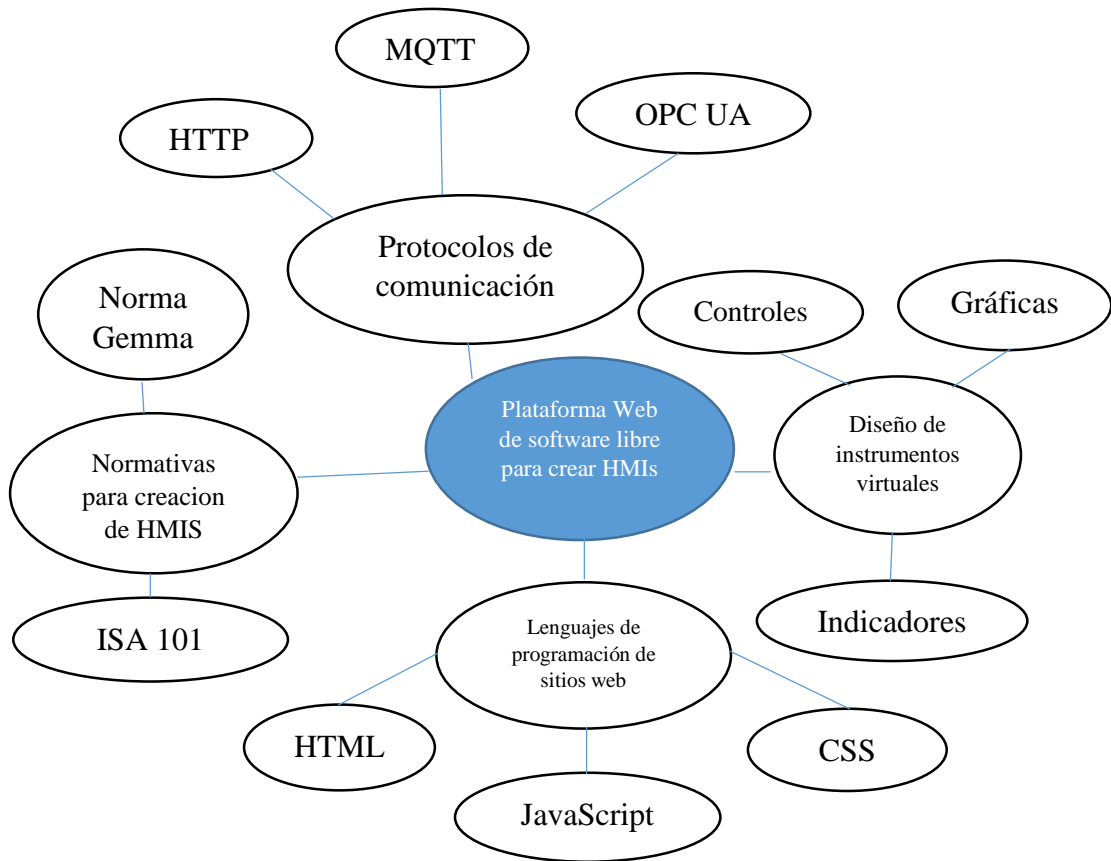


Fig. 2. 2: Constelación de ideas de la Variable independiente

Elaborado por: El Investigador

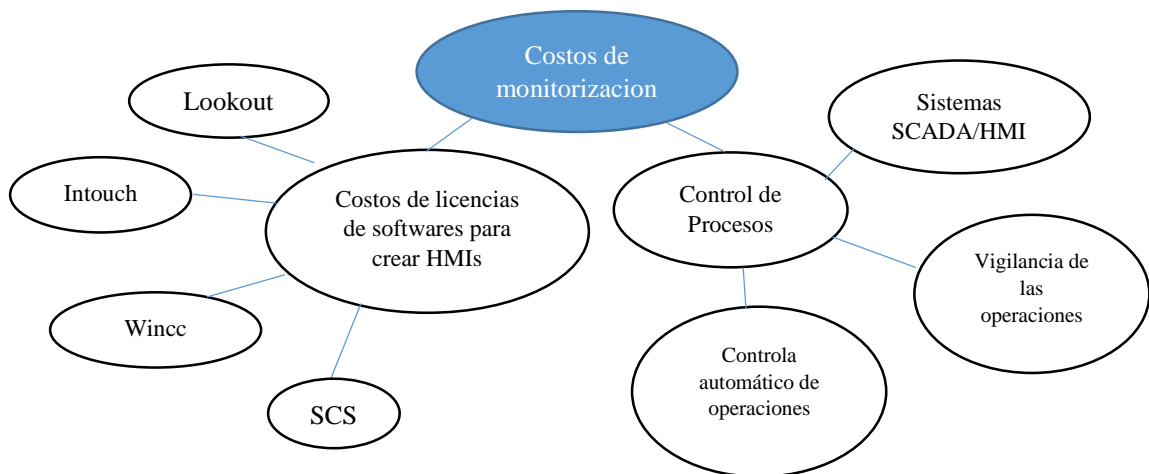


Fig. 2. 3: Constelación de la variable dependiente

Elaborado por: El Investigador

2.3.3 Sistema SCADA

Los sistemas SCADA (Supervisory Control And Data Adquisition) son aplicaciones de software diseñadas con la finalidad de controlar y supervisar procesos a distancia que permite el acceso a datos remotos. Utilizando las herramientas de comunicación con los dispositivos de campo (controladores autónomos, sensores, etc.) necesarias en cada caso y controlando el proceso de forma automática, desde una computadora u otro dispositivo. (Rodríguez Penin, 2007)

En un sistema SCADA se incluyen muchos subsistemas, así la adquisición de los datos puede estar a cargo de un PLC (Controlador Lógico Programable) el cual recibe las señales y las envía a las estaciones remotas usando protocolos determinados. Otra manera podría ser a través de una computadora que realice la toma de datos vía hardware especializado y luego transmita la información a un equipo de radio mediante su puerto serial, y así un sinnúmero de alternativas. (Mendiburu, 2013)

“Las tareas de Supervisión y Control generalmente están más relacionadas con el software SCADA, en él, el operador puede visualizar cada una de las estaciones remotas que conforman el sistema, sus estados, las situaciones de alarma y tomar acciones físicas sobre algún equipo lejano, la comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.”.(Mendiburu, 2013)

Generalmente se vincula el software al uso de una computadora o de un PLC, la acción de control es realizada por los controladores de campo, pero la comunicación del sistema con el operador es necesariamente vía computadora. Sin embargo el operador puede gobernar el proceso en un momento dado si es necesario. En la figura 2.4 se puede observar cómo está compuesto un sistema SCADA.(Mendiburu, 2013)

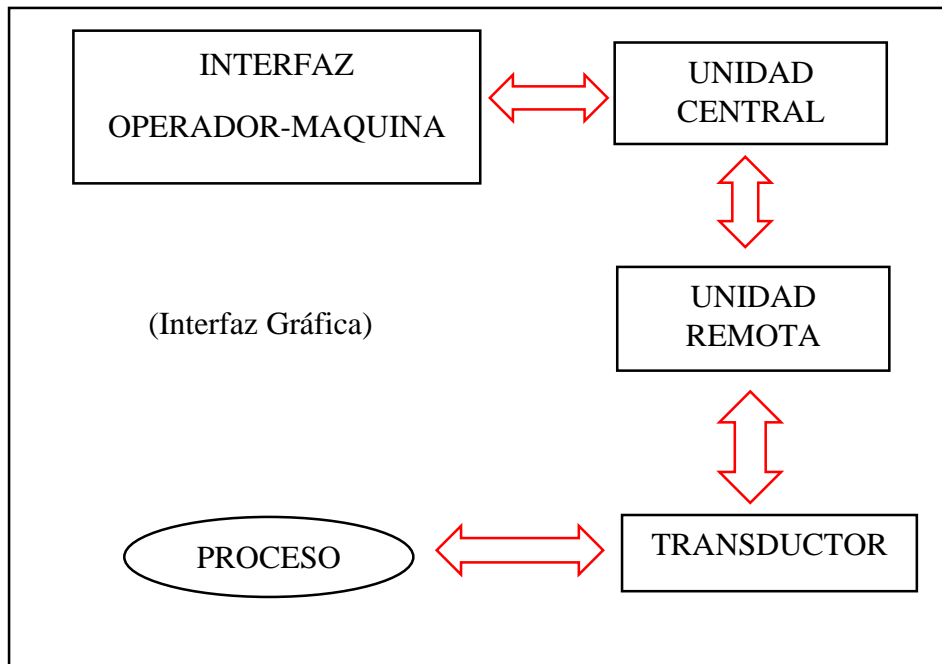


Fig. 2. 4 Sistema SCADA

Elaborado por: El Investigador

2.3.4 Componentes de un SCADA

Los componentes de un sistema SCADA se pueden ubicar en dos grupos principales: Hardware y Software.

2.3.4.1 Hardware

El hardware es el elemento físico y su estrategia se fundamenta básicamente en el concepto de redundancia, entendida como la capacidad de un elemento, de asumir las funciones de otro de forma transparente al sistema.

El principio de redundancia se aplica a todos los niveles de un sistema, desde componentes individuales hasta sistemas enteros (fuentes de alimentación, backup de datos, sistemas de comunicación) tal como se indica en la Figuras 2.5 se presenta un ejemplo de sistema de

redundancia. De esta manera es posible continuar trabajando en caso de fallo de uno de los componentes.

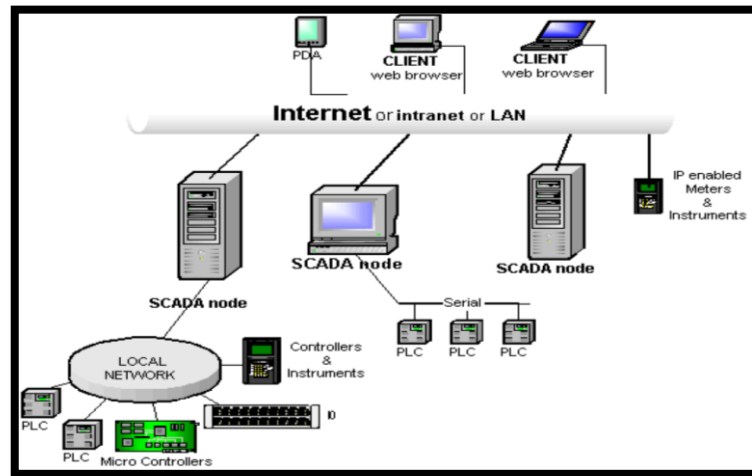


Fig. 2. 5: Sistema de Redundancia

Fuente: (Instituto Nacional de Estadística e Informática, 2015)

Los componentes básicos de hardware son los siguientes:

- a) **Unidad Terminal Maestra (MTU):** La Unidad Terminal Maestra son los servidores y el software responsable para comunicarse con el equipo de campo (RTU's, PLC's, etc.). En estos se encuentra el software HMI corriendo para las estaciones de trabajo en el cuarto de control o en cualquier otro espacio. En un sistema SCADA pequeño, la Unidad Terminal Maestra puede ser una sola, pero en un sistema SCADA a gran escala, la Unidad Terminal Maestra puede incluir muchos servidores, aplicaciones de software distribuido, y sitios de recuperación de desastre.(Chavarría Meza, 2007)

Esta terminal ejecuta las acciones de mando (programadas) en base a los valores actuales de las variables medidas. La programación se realiza por medio de bloques de programa en lenguaje de alto nivel (como C, Basic, etc.). También se encarga del almacenamiento y procesamiento ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.(Chavarría Meza, 2007)

- b) **Unidad Terminal Remota (RTU):** La Unidad Terminal Remota RTU es un dispositivo instalado en una posición remota que obtiene datos, los descifra en un formato y

transmite los datos de nuevo a una unidad terminal maestra (MTU). La RTU también recoge la información del dispositivo principal y pone los procesos en ejecución que son dirigidos por la MTU. En la figura 2.6 se puede visualizar la arquitectura general de un RTU. (Rodríguez Penin, 2007)

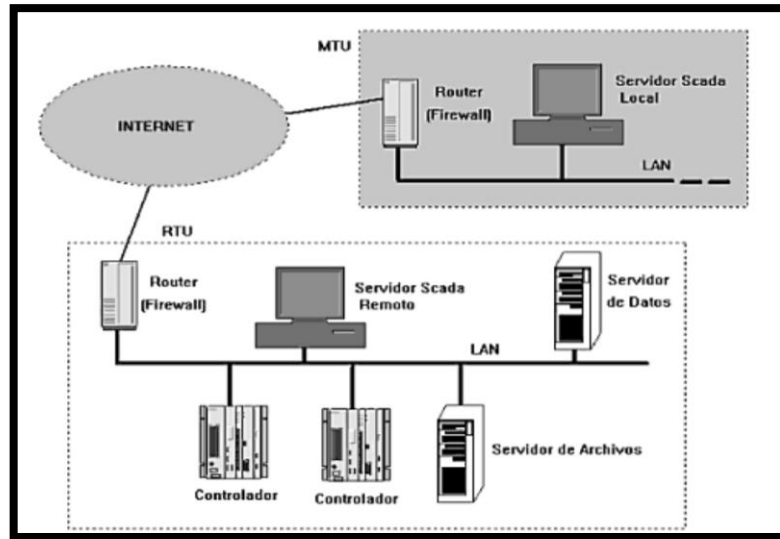


Fig. 2. 6: Arquitectura General de un RTU

Fuente: (Rodríguez Penin, 2007)

- c) **Sistema de Comunicaciones:** El Sistema de Comunicaciones se encarga de la transferencia de información del punto donde se realizan las operaciones, hasta el punto donde se supervisa y controla el proceso. Lo conforman los transmisores, receptores y medios de comunicación. (Chavarría Meza, 2007)
- d) **Transductores:** El transductor es un dispositivo capaz de transformar o convertir un determinado tipo de energía de entrada, en otra de diferente de salida. El nombre del transductor ya nos indica cual es la transformación que realiza, aunque no necesariamente la dirección de la misma. Es un dispositivo usado para obtener la información de entornos físicos y conseguir (a partir de esta información) señales o impulsos eléctricos o viceversa. En este caso permitirá la conversión de una señal física en una señal eléctrica (y viceversa). Su calibración es muy importante para que no haya problema con la confusión de valores de los datos. (Chavarría Meza, 2007)

2.3.4.2 Software

El software es el equipo lógico o soporte lógico de un sistema informático. “Un programa de tipo HMI se ejecuta en un ordenador o Terminal gráfico y unos programas específicos le permiten comunicarse con los dispositivos de control de planta (hacia abajo) y los elementos de gestión (hacia arriba). Estos programas son lo que denominamos controladores (o driver) de comunicaciones.” (Rodríguez Penin, 2007).

“El driver realiza la función de traducción entre el lenguaje del programa SCADA y el del Autómata. Generalmente la configuración del controlador de comunicaciones se realiza durante la instalación del software principal o como programa de acceso externo al ejecutar la aplicación principal”. (Rodríguez Penin, 2007)

En la actualidad la aparición de la informática permite que el control se realice de manera más sencilla. Reemplazando los grandes cuadros de control por monitores que permiten mostrar la misma información ocupando menos espacio. Logrando con esto simplificar cambios con tan solo algunas modificaciones en el código de la aplicación para que apareciera en la pantalla. Entre los paquetes de softwares más comunes para el desarrollo de estas pantallas de monitorización se destacan los siguientes: IFIX de Intellution, SCS de Omron, WinCC de Siemens, RS-View de Rockwell Automation, InTouch de Wonderware, Cimplicity de GE-Fanuc. (López & Zaragoza, 2014)

2.3.5 Interfaz hombre maquina (HMI)

La interfaz hombre maquina (HMI) se define como el sistema para que el operador pueda visualizar y monitorizar el proceso desarrollado por la planta. El mismo que le permite interactuar con los medios tecnológicos implementados. Esta interfaz generalmente incluye controles donde el operador se puede interconectar con el sistema SCADA. Las interfaces humano maquina varían ampliamente, desde paneles de control para centrales nucleares hasta botones de entrada en un celular.

Un HMI es una manera fácil de estandarizar la supervisión de las RTU's múltiples o de los PLC's (Controladores Lógicos Programables). Los HMI se vinculan a bases de datos, que

utilizan los datos recopilados de los PLC's o de las RTU's para proporcionar las tendencias, los datos de diagnóstico y manejo de información así como el cronograma de procedimientos de mantenimiento, logística y localización de averías.(Arenas L, Castilla A, 2015; Chavarría Meza, 2007)

Existen dos componentes imprescindibles en las interfaces humano máquina. El primero es la entrada, un usuario necesita de algún medio para decirle a la máquina que es lo que debe hacer. El segundo es la salida, esta salida permite a la maquina mantener al usuario actualizado acerca del progreso de los procesos, o la ejecución de comandos en un espacio físico.

Una adecuada interfaz hombre maquina busca, en primer lugar obtener el estado del proceso en una sola vista y persigue captar la situación en forma rápida, crear condiciones para la toma de decisiones correctas, que los equipos se utilicen en forma óptima y segura, garantizar la confiabilidad al máximo y cambiar con facilidad los niveles de actividades del operador.(Arenas L, Castilla A, 2015)

2.3.5.1 Características básicas de los HMI

Los aspectos esenciales que se realizan en la comunicación hombre maquina son:

a) **Indicación del estado del proceso**

Para ello se pueden utilizar equipos convencionales tales como: terminales de video, impresoras, registradores, diodos emisores de luz. Para la selección en pantallas se pueden emplear: mouse, teclados, lápiz ópticos y touch screen.

b) **Tratamiento e indicación de alarmas (buscan informar al operador de una situación anormal)**

Las alarmas se ejecutan en la propia pantalla, mediante símbolos que aparecen intermitentemente, cambios repetidos de color en el nombre de alguna variable o grupo, intermitencia de textos, mensajes etc. Esto se realiza a través del uso de videos específicos, indicaciones sonoras o impresoras para imprimir los mensajes de alarma.

c) **Ejecución de acciones de mando**

Este tipo de acciones se pueden realizar por técnicas convencionales (pulsadores, interruptores, potenciómetros, etc.), o mediante teclados, lápices ópticos, mouses, pantallas táctiles etc.

Otro punto importante a tener en cuenta son las características del puesto de mando, ya que se debe buscar comodidad para el operador proveyendo ciertas condiciones tales como: temperatura estable, presión atmosférica ligeramente superior al exterior, muebles cómodos y funcionales y buena iluminación. (Arenas L, Castilla A, 2015)

2.3.5.2 Tipos de HMI

Interface de manipulación directa es el nombre de una clase general de interfaces de usuario. Estas interfaces permiten a los usuarios manipular los objetos que se les presenten, con las acciones que correspondan con el mundo físico. A continuación se presenta los tipos de interfaz de usuario más comunes:

a) **Interfaces graficas de usuario (GUI).**

Este tipo de interfaces aceptan la entrada a través de un dispositivo como teclado de la computadora o el ratón y proporciona una salida grafica en la pantalla del ordenador. Hay por lo menos dos principios diferentes utilizados en el diseño de interfaces graficas de usuario, las interfaces de usuario orientada a objetos y las interfaces orientadas a aplicaciones.

b) **Interfaces basadas en Web de usuario o interfaces de usuario web (IUF)**

Estas interfaces son una subclase de las interfaces graficas de usuario que aceptan una entrada y proporcionan una salida mediante las páginas web que se transmiten a través de internet y vistos por el usuario mediante un navegador web.

c) **Las pantallas táctiles**

Las pantallas táctiles son dispositivos que aceptan una entrada a través del tacto de los dedos o un lápiz. Se utiliza en una amplia cantidad de dispositivos móviles y muchos tipos de procesos industriales y máquinas de autoservicio.

d) **Interfaces de línea de comandos**

En este tipo de interfaces el usuario proporciona la entrada al escribir una cadena de comandos con el teclado del ordenador y el sistema proporciona una salida de impresión de texto en la pantalla del ordenador. Este tipo de interfaz es utilizada por los programadores y administradores de sistemas, en los ambientes científicos y de ingeniería, y por los usuarios de computadoras personales de tecnología avanzada.

e) **Interfaces de voz del usuario**

Este tipo de interfaces acepta la entrada y proporcionan una salida mediante la generación de mensajes de voz. La entrada del usuario se realiza pulsando las teclas o botones, o responder verbalmente a la interfaz.(Arenas L, Castilla A, 2015)

2.3.5.3 Criterios generales para la creación de una interfaz hombre maquina (HMI).

Los HMI presentan el medio de interacción entre un usuario y un determinado hardware, para el caso de control de procesos el HMI, debe ser capaz de mostrar al usuario datos básicos de todo sistema de control de procesos, tales como: variable de proceso, variable de control y set point o variable de consigna, todo esto presentado en tiempo real, es decir en el momento mismo de la ejecución de las diferentes variaciones.

Se recomienda que una HMI contenga tanto componentes gráficos como componentes numéricos. Así mismo se debe utilizar terminología estandarizada y clara para el usuario final. También, se recomienda que las variables del proceso, set point y variable de control sean lo más claras posibles para el usuario; de igual manera, se debe mantener un registro histórico de las variaciones ocurridas, esto con el fin de estudiar su comportamiento y poder realizar las predicciones respectivas. En resumen un HMI debe proporcionar una explicación transparente y oportuna en tiempo real de lo que ocurre en el proceso.(Ruíz Lizama, Inche Mitma, & Chung Pinzás, 2016)

2.3.5.4 Normativa para la creación de interfaces hombre maquina

Para el desarrollo de interfaces hombre máquina, existen varias alternativas y fuentes de guía como también varias normas internacionales para ejecutar diferentes procesos industriales. Las normativas internacionales para el buen uso, desempeño y diseño del HMI son de carácter técnico exclusivamente, las implementaciones pueden ir cambiando debido a que cada norma o guía maneja diferentes ideas de desarrollo. Sin embargo las normativas y guías más utilizadas

son la norma ISA 101, la norma ISO 9241 y la guía GEDIS. Mismas que son usadas como referencia en el diseño de las interfaces humano-máquina.

ISA SP 101

La norma ISA SP 101 en su comité (ISA 101, Human Machine Interfaces, 2015) estableció un estándar, recomendaciones prácticas y reportes técnicos pertinentes para aplicaciones en el desarrollo de interfaces hombre máquina aplicadas al control y monitorización de procesos. El alcance de la norma ISA SP 101 aborda temas como: filosofía del HMI, guía de estilo del HMI, plataforma orientada para el HMI así como diseño del HMI. (Penin, 2007)

Norma ISO 9241

La norma UNE-EN ISO 9241, en los capítulos del 14 al 17 describe la manera de realizar el diseño ergonómico de programas para equipos con pantallas de visualización de datos. Enumerando una serie de ideas las cuales se busca que sirvan de modelo de referencia a la hora de realizar el planteamiento y desarrollo de las interfaces gráficas.

Esta normativa establece que las aplicaciones se deben adaptar a la tarea para la cual se han diseñado, además de ser amigables para el usuario de manera que el diálogo sea limpio. Presentando y exigiendo solamente la información estrictamente necesaria. Lo cual permite informar del progreso al interlocutor de forma comprensible.

Se establece también que la aplicación debe poder adaptarse al nivel de capacitación del usuario. Permitiendo ser controlable por el mismo, y no de manera contraria. Además deberá ser tolerante a fallos y con herramientas de corrección automáticas. (Eras, Quito, Bolivar, & Matamoros, 2010)

Aunque las dos normativas presentan muy buenas recomendaciones al momento de desarrollar pantallas de monitorización la normativa más difundida es la norma ISA 101. Por lo cual el usuario es el encargado de regirse a una determinada normativa y seguir los consejos de ubicación, presentación y visualización tanto de información como de elementos a utilizar. Permitiendo que las pantallas de monitorización se diseñen de una manera adecuada y que permita la rápida asimilación de sus funciones y características.

2.3.6 Necesidades de un Sistema SCADA

Para que se pueda implementar un sistema SCADA es necesario que el proceso a controlar cumpla con las siguientes características:

- El número de variables a monitorear es alto.
- El proceso está geográficamente distribuido. Esta condición no se limita, ya que puede instalarse un SCADA para la supervisión y control de un proceso concentrado en una localidad.
- La información del proceso se necesita en el momento en que se producen los cambios, o sea, la información se requiere en tiempo real.
- Optimizar y facilitar las operaciones de la planta, así como la toma de decisiones, tanto gerenciales como operativas. (Chavarría Meza, 2007),(Rodríguez Penin, 2007)

2.3.7 Funciones de un Sistema SCADA

La principal función que realiza un sistema scada es la supervisión remota de instalaciones y equipos. Lo cual permite al operador conocer el estado de desempeño de las instalaciones y los equipos alojados en la planta. Permitiendo de esta manera dirigir las tareas de mantenimiento y estadística de fallas. Además de la función de supervisión los sistemas scadas nos permiten realizar un control remoto de las instalaciones y equipos de la planta, dándonos la posibilidad de activar o desactivar equipos remotamente tales como: válvulas, interruptores, motores, etc.

Otra de las funciones primordiales de los sistemas scada es la recolección y procesamiento de los datos. El conjunto de datos adquiridos conforma la información que alimenta el sistema, esta información es procesada, analizada, y comparada con datos anteriores, y con datos de otros puntos de referencia, dando como resultado una información confiable y veraz. Esta funcionalidad de los sistemas scada trabaja en conjunto con la función de almacenamiento de información, permitiendo que los datos adquiridos y procesados sean guardados para su posterior análisis.

Finalmente la función de visualización brinda imágenes en movimiento que representan el comportamiento del proceso. Esta función otorga al operador la impresión de estar presente dentro de una planta real. Esta función de visualización permite también presentar graficas que corresponden a curvas de las señales analizadas. Esta funcionalidad trabaja de la mano con la funcionalidad de presentación de alarmas ya que a través de la visualización del proceso, se evalúa el estado de las variables, y en dependencia de ciertos valores programados se ejecuta la funcionalidad de generación de alarmas. Estas alarmas alertan al operador la presencia de una condición perjudicial o fuera de lo aceptable. (Rodríguez Penin, 2007)

2.3.8 Equipos de control

Los equipos de control utilizados en un sistema SCADA son los PLC que son controladores lógicos programables que están constituidos con un conjunto de tarjetas o circuitos impresos sobre los cuales están ubicados componentes electrónicos.

2.3.8.1 Controladores Lógicos Programables

Un controlador lógico programable (PLC) es una forma especial de microprocesador el cual usa una memoria programable para almacenar instrucciones e implementar funciones tales como: lógicas, secuencias, temporizadas, contadores, aritméticas entre otras. Con el objetivo de controlar máquinas y procesos y están diseñados para ser operados por ingenieros con conocimientos de lenguajes de programación. En la figura 2.7, se puede visualizar las partes que constituye un PLC.(Bolton, 2006)

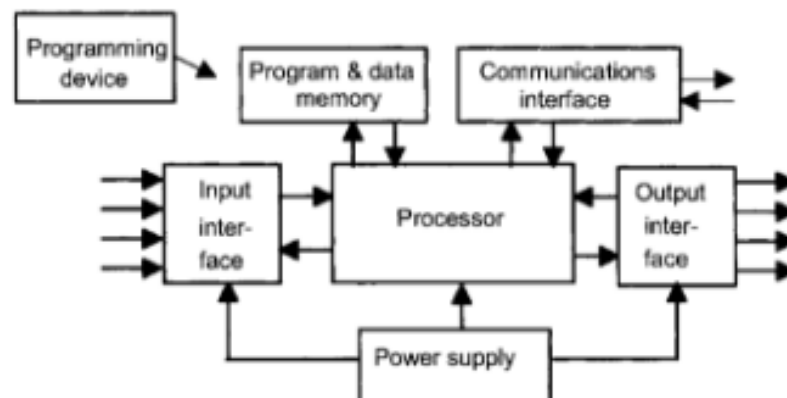


Fig. 2. 7: Partes que constituyen un PLC (Controlador Lógico Programable)

Fuente: (Bolton, 2006)

2.3.8.2 Sensores

Un sensor es un dispositivo eléctrico y/o mecánico que convierte magnitudes físicas o químicas llamadas variables de instrumentación (luz, magnetismo, presión, temperatura, distancia, aceleración, desplazamiento, presión, fuerza, humedad etc.) a una señal eléctrica.

Un fenómeno físico a ser medido es captado por un sensor, y muestra en su salida una señal eléctrica dependiente del valor de la variable física.

La señal eléctrica puede ser resistencia eléctrica (con una RTD), capacidad eléctrica (con un sensor de humedad), tensión eléctrica (un termopar), corriente eléctrica (fototransistor), etc. La señal eléctrica es modificada por un sistema de acondicionamiento de señal, cuya salida es un voltaje.

El sensor dispone de una circuitería que transforma y/o amplifica la tensión de salida, la cual pasa a un conversor A/D, conectado a un PC. El convertidor A/D transforma la señal de tensión continua en una señal discreta. (Germán Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2014)

2.3.8.3 Actuadores

Un actuador es un dispositivo que posee la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía.

Los actuadores se dividen en dos grandes grupos. El primer grupo se define por el tipo de energía que utilizan y dentro del cual se encuentran los actuadores neumáticos, actuadores hidráulicos y los actuadores eléctricos. El segundo grupo se define por el tipo de movimiento que estos realizan encontrándose dentro de este grupo los actuadores lineales y los actuadores rotativos. En la figura 2.8 se aprecia la clasificación de los actuadores antes mencionados identificándolos por el tipo de energía que utilizan y dentro de los cuales encontramos actuadores que realizan movimientos tanto lineales como rotativos. (Germán Corona Ramírez et al., 2014)

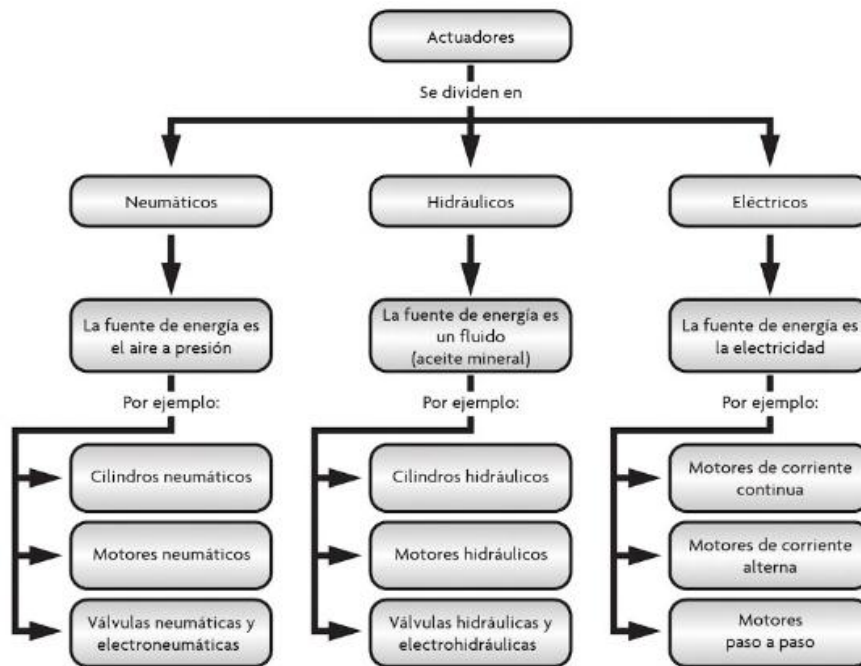


Fig. 2. 8: Clasificación de los actuadores

Fuente: (Germán Corona Ramírez et al., 2014)

2.3.9 Protocolos de comunicación

2.3.9.1 HTTP (REST/JSON) (Hypertext Transfer Protocol)

HTTP es la base del modelo cliente-servidor usado para la Web. El método más seguro de implementar el HTTP en un dispositivo IoT es incluir solo un cliente, no un servidor. En otras palabras, es más seguro si el dispositivo de IoT puede iniciar conexiones a un servidor Web, pero no es capaz de recibir solicitudes de conexión. Ya que es aconsejable no permitir que máquinas externas tengan acceso a la red local donde se encuentran instalados los dispositivos de IoT.

HTTP es un protocolo muy accesible por ser de código abierto, además de poseer numerosas librerías. Es efectivo para enviar grandes cantidades de información, como por ejemplo lecturas de sensores minuto a minuto o cada hora; aunque no es adecuado ni para enviar actualizaciones en periodos de tiempo del orden de milisegundos ni para enviar información de video. Es muy recomendable asegurar la información transmitida aplicando el protocolo criptográfico SSL/TLS sobre HTTP, lo que genera el protocolo de aplicación HTTPS. No

obstante, el método más seguro consiste en incluir en el dispositivo IoT solo un cliente HTTP, no un servidor HTTP, de manera que el dispositivo IoT pueda iniciar conexiones a un servidor web, pero no sea capaz de recibir solicitudes de conexión. (Castro Gil, Díaz Orueta, & Mur Pérez, 2000)

2.3.9.2 MQTT (Message Queuing Telemetry Transport)

El protocolo de transporte de telemetría de cola de mensajes (MQTT) es un protocolo de código abierto que se desarrolló y optimizó para dispositivos restringidos y redes de bajo ancho de banda, alta latencia o poco confiables. Es un transporte de mensajería de publicación/suscripción que es extremadamente ligero e ideal para conectar dispositivos pequeños a redes con ancho de banda mínimo.

MQTT es eficiente en términos de ancho de banda, independiente de los datos y tiene reconocimiento de sesión continua, porque usa TCP. Tiene la finalidad de minimizar los requerimientos de recursos del dispositivo y, a la vez, tratar de asegurar la confiabilidad y cierto grado de seguridad de entrega con calidad del servicio.

MQTT se orienta a grandes redes de dispositivos pequeños que necesitan la supervisión o el control de un servidor de back-end en Internet. No está diseñado para la transferencia de dispositivo a dispositivo. Tampoco está diseñado para realizar "multidifusión" de datos a muchos receptores. El MQTT es simple y ofrece pocas opciones de control. Las aplicaciones que usan MQTT, por lo general, son lentas en el sentido de que la definición de "tiempo real" en este caso se mide habitualmente en segundos. (Boyd et al., 2017)

2.3.9.3 OPC UA (Unified Architecture)

OPC UA es un estándar de nueva generación que surge a partir del OPC, el cual es conocido por proveer una interfaz estándar para comunicarse con los PLC. OPC UA es un protocolo cliente/servidor multiplataforma, donde los clientes se conectan, navegan, leen y escriben al equipamiento industrial. Este protocolo ofrece un modo binario y otro basado en SOAP (Web Service).

Las tecnologías y metodologías innovadoras, como los nuevos protocolos de transporte, los algoritmos de seguridad, los estándares de codificación o los servicios de aplicación pueden incorporarse en OPC UA mientras se mantiene la compatibilidad retroactiva para los productos existentes. Es una opción para conectar información de sensores y PLC en las aplicaciones industriales existentes como sistemas MES (Manufacturing Execution System) y SCADA (Supervisory Control And Data Acquisition).

Este estándar implementa medidas de seguridad para las comunicaciones, como son autenticación de la aplicación, autenticación y autorización del usuario, disponibilidad del servidor, auditabilidad del sistema y confidencialidad e integridad, lo que lo convierte en un protocolo muy seguro. OPC UA ofrece seguridad nativa que incluye autenticación y autorización, cifrado e integridad de datos vía firmas. Para la versión SOAP hace uso de la especificación de seguridad para Web Service desarrollada por IBM, WS-SecureConversation, mientras que en la versión binaria hace uso de una implementación específica de la anterior. Soporta certificados X.509 para la autenticación y se puede integrar con directorio activo y PKI. (Wilamowski & Irwin, 2011)

2.3.10 Plataformas Web

Una plataforma web es una nueva manera de ofrecer servicios en internet gracias a la suma y combinación de diversas tecnologías que permiten utilizar la red como plataforma de aplicaciones. (Nafría, 2008)

2.3.10.1 Lenguajes de programación Web

Al momento de realizar una página web se debe tomar en consideración algunos parámetros importantes para iniciar con este trabajo los cuales son: un boceto inicial de la página, estructurar la información, e identificación de la audiencia. Para poder integrar todos estos componentes y construir nuestro sitio web debemos tomar en consideración que necesitaremos de varios lenguajes de programación utilizados en esta área tales como:

- **HTML**

Este es un lenguaje artificial que los ordenadores son capaces de interpretar y diseñado para que los programadores redacten instrucciones que los navegadores ejecutaran para originar la página web. Es decir un lenguaje que el ordenador interpreta y genera una respuesta.

Las siglas HTML significa HyperText Markup Language (lenguaje de marcas de hipertexto). El hipertexto en una computadora es texto que posee referencias (hipervínculos, links o enlaces) a otro texto. Para simplificar podemos decir que el hipertexto es aquel texto que pulsamos con el mouse del ordenador y nos conduce a otro texto cuando utilizamos Internet. Pero además texto, el hipertexto puede estar formado por tablas, imágenes u otros elementos.

HTML sirve para estructurar documentos (títulos, párrafos, listas, etc.), pero no describe la apariencia o el diseño de un documento sino que ofrece las herramientas necesarias para dar formato, según la capacidad del servidor web en el que se almacene las páginas y la capacidad del navegador.

Este lenguaje posee dos grandes ventajas que lo hacen prácticamente indispensable a la hora de diseñar una presentación web: su compatibilidad y la facilidad que plantea su aprendizaje debido al reducido número de etiquetas en las que se apoya.(Equipo Vertice, 2009)

- **CSS**

Las hojas de estilo en cascada, conocida por todos como **CCS** por sus siglas, del inglés (Cascading Style Sheets), son las encargadas de estructurar y presentar los documentos creados con el lenguaje de marcado HTML, de manera gráfica y visual. Permittiéndonos crear sitios visualmente atractivos y con algunas animaciones que antes estaban sujetas a contar con tecnologías como Flash Player o JavaScript.(Fernando Luna, Claudio Peña Millahual, 2015)

- **JavaScript**

JavaScript posee un diseño de sintaxis similar al del lenguaje C con nombres y convenciones adoptados propiamente del lenguaje de Java. Este lenguaje nos ayuda a desarrollar gran parte de la sintaxis propia que complementa a un sitio web, la que se ejecutara mayormente del lado del cliente. JavaScript permite acceder al contenido de hardware propio de cada dispositivo de escritorio o móvil que cargue la web. Junto con HTML y CSS son los lenguajes principales al momento de desarrollar páginas web. (Fernando Luna, Claudio Peña Millahual, 2015)

2.4 Hipótesis

El diseño de una plataforma web de software libre influye en la reducción de costos en el monitorización de procesos y en la interfaz hombre – máquina.

2.5 Señalamiento de variables de la hipótesis

Variable Independiente

Plataforma Web de software libre para crear HMIs

Variable dependiente

Costos de monitorización

CAPÍTULO III

METODOLOGÍA

3.1 Enfoque

El proyecto propuesto tendrá un enfoque cualitativo pues se validará a través de calificar y evaluar las características de la plataforma web propuesta, las mismas que proveerán información de las capacidades y beneficios de la misma, gracias a indicaciones se determinara de manera concluyente en que forma influye el sistema planteado, obteniendo conclusiones de importancia para el proyecto.

3.2 Modalidad básica de la información

Bibliografía Documental

La investigación tendrá esta modalidad debido a que se utilizara fuentes de información como: libros, manuales técnicos, publicaciones, internet, revistas especializadas. También se recurrirá a fuentes primarias obtenidas de experimentación, y datos de trabajos realizados con anterioridad.

De Campo

Se trabajará con la modalidad de campo ya que es necesario realizar pruebas y experimentos en condiciones reales, con la finalidad de recabar información que permita al investigador acercarse a la mejor solución para el problema de investigación.

3.3 Operacionalización de variables

Tabla 3- 1. Variable independiente: Plataforma Web de software libre para crear HMIs

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS BÁSICOS	TÉCNICAS E INSTRUMENTOS
Se entiende por plataforma web al conjunto de elementos que conforman las páginas web como aplicaciones, formularios, convertidores instancias entre otros.	Variables de entrada	Información proveniente de los sensores de nuestros procesos.	¿Existe monitorización de los procesos a través del envío de información mediante sensores?	Revisión de instrumentos de medición. Exploración de instrumentos y métodos de comunicación y transmisión de información.
		Nivel de programación	¿Qué nivel de programación es necesario para el desarrollo de este proyecto?	Estudio de la complejidad de los lenguajes y técnicas para creación de plataformas web.
	Plataforma Web	Costo de implementación	¿Qué lenguajes de programación y técnicas son necesarias para el desarrollo de la	Estudio de plataformas y lenguajes de programación adecuadas según factibilidad y adaptabilidad.

Variables de salida	Presentación de información en pantallas con instrumentos virtuales.	plataforma web propuesta? ¿Qué costos conlleva esta implementación? ¿Es necesaria la visualización y presentación la información del proceso en tiempo real?	Indagación de costos actuales de dispositivos para realizar la implementación de estos sistemas. Recolección de información acerca de los instrumentos virtuales para presentación de información.
---------------------	--	--	---

Elaborado por: El Investigador

Tabla 3- 2. Variable dependiente: Costos de monitorización

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ÍTEMS BÁSICOS	TÉCNICAS E INSTRUMENTOS
Los costos de monitorización en este proyecto se evaluarán en función de costo beneficio de usar un software licenciado a usar la plataforma de acceso libre propuesta.	Softwares licenciados	Tipos de softwares utilizados.	¿Cuáles son los softwares más empleados para la creación de HMIs?	Observación
	Beneficios de uso de software de pago	Valoración de los beneficios de usar un software de pago.	¿Cuál es el costo de las licencias estos softwares? ¿Qué beneficios presentan estos softwares de pago en comparación con un sistema de libre acceso?	Investigación Realización de pruebas comparación de datos y resultados en función de la experimentación.

Beneficios al usar una plataforma de software libre	Valoración de los beneficios de usar un software de acceso libre.	¿Los sistemas de acceso libre presentan beneficios o problemas?
--	--	---

Elaborado por: El Investigador

3.4 Recolección de información

La recopilación de la información se la realizará principalmente de trabajos similares al tema de investigación. Los cuales nos permitan encontrar elementos comunes útiles para el desarrollo del proyecto. También se procederá a fuentes de información externas como libros, artículos, manuales e internet. Lo cual permitirá contrastar y completar la información ya adquirida.

Ya que la mayoría del proyecto se basara en programación web se recurrirá a páginas web de comunidades de desarrolladores. Las mismas que muchas de las veces aportan herramientas nuevas que aún no han sido documentadas. Toda la información recolectada para el desarrollo del proyecto nos permitirá que la información a presentar en el las pantallas de monitorización (HMIs) sea la correcta y no tenga componentes erróneos.

3.5 Procesamiento y análisis

El procesamiento y análisis de la información bibliográfica parte de la investigación y análisis de las diferentes formas de programación web enfocadas a la creación de elementos virtuales tales como plugins o librerías que permitan diseñar elementos que intercambien información a través de herramientas web. Instrumentos que nos permitirán llevar a cabo la visualización y control de procesos. Para crear dichos instrumentos es necesario la correcta selección de elementos y lenguajes de programación. Lenguajes que nos permitan intercambiar información en tiempo real con los dispositivos de control. De esta idea se plantea una hipótesis que se pretende responder al finalizar el presente trabajo de investigación. Posteriormente los resultados de la propuesta desarrollada permitirán decidir si tanto la hipótesis como la formulación del problema han sido contestadas. Además permitirá que este trabajo sirva como punto de referencia para futuras investigaciones.

CAPÍTULO IV

ANÁLISIS DE LA INFORMACIÓN RECOLECTADA

4.1 Análisis de Resultados

4.1.1 Lenguajes y herramientas de programación web

Con el propósito de cumplir con la propuesta establecida en el presente trabajo de investigación, se realizó una búsqueda de los principales lenguajes, técnicas y herramientas de programación de aplicaciones web. Llegando a la conclusión que los lenguajes que se describirán a continuación son los más importantes para el desarrollo del presente trabajo.

4.1.2 HTML para programación de páginas Web

HTML es el lenguaje básico para la construcción de páginas web. Este lenguaje se encuentra estrictamente organizado a través de etiquetas. Estas etiquetas son palabras claves y atributos rodeados por los signos mayor y menor (por ejemplo, `<html lang="es">`). En este caso, **html** es la palabra clave y **lang** es el atributo con el valor **es**. HTML ha pasado por varias actualizaciones tales como: XHTML1, DOM Nivel 2, HTML4, hasta llegar a su última versión HTML5.

HTML5 es la quinta revisión del lenguaje HTML y nace como respuesta a la incompatibilidad que los antiguos lenguajes de programación web presentaban. Dado que Java y Flash comenzaron a quedarse a tras frente a otros lenguajes como JavaScript que presentaba herramientas que los anteriores no preveían para desarrollar la nueva generación de aplicaciones que se demandaban. HTML5 surge como respuesta a estos problemas ya que permitía la óptima integración de JavaScript, HTML y CSS.

HTML5 es, de hecho, una mejora de esta combinación, el pegamento que une todo. HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. A partir de ahora, HTML provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y

Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales.(Garro, 2015; Gauchat, 2012)

A continuación se presentará las etiquetas más representativas del lenguaje HTML esenciales para la realización de sitios web. Ver tabla 4.1.

Tabla 4- 1: *Etiquetas del lenguaje HTML.*

Etiqueta	Función
<!--...-->	Define un comentario
<!DOCTYPE>	Define el tipo de documento
<a>	Define un hipervínculo
<body>	Define el cuerpo del documento
 	Define un salto de línea
<button>	Define un botón clickeable
<div>	Define una sección en un documento
<head>	Define información acerca del documento
<header>	Define la sección de encabezado del documento
<html>	Define la raíz del documento
	Define una imagen
<input>	Define un control de entrada de texto
<label>	Define el rótulo para un elemento <input>
	Define un ítem de una lista
<link>	Define la relación entre un documento y un recurso externo (generalmente con hojas de estilo)
<nav>	Define un link de navegación
<option>	Define una opción en una lista desplegable
<script>	Define un script del lado cliente
<select>	Define un drop-down list
	Define una pequeña sección de un documento
<style>	Define un estilo para la información de un documento
<title>	Define un título para el documento
	Define una lista desordenada

Fuente: (Garro, 2015; Gauchat, 2012)

4.1.3 CSS para programación de páginas Web

CSS es un lenguaje que trabaja en conjunto con HTML para proveer estilos visuales a los elementos del documento tales como: tamaño, color, fondo, forma, bordes entre otros. Comúnmente se emplea para dar formato visual a documentos html o xhtml que funcionan como espacios web, y su utilización es imprescindible para la creación de páginas web complejas.

CSS es un complemento que nace de la necesidad de diseñar la información de tal manera que podemos separar el contenido de la presentación y, viceversa, por una misma fuente de información, generalmente definida mediante un lenguaje de marcaje. Ofrece diferentes presentaciones en función de dispositivos, servicios, contextos o aplicativos. Por lo que un mismo documento html, mediante diferentes hojas de estilo, puede ser presentado por pantalla, por impresora, por lectores de voz o por tabletas braille. Separamos el contenido de la forma, composición, colores y fuentes.(Gauchat, 2012; Puig, 2014)

CSS al igual que html paso por varias especificaciones, mejorando en cada una de ellas varias de sus características tales como: tipo de letra, colores de texto y fondo, espacio entre caracteres, espacio entre líneas y palabras, alineación de tablas, márgenes etc. A través de las diferentes especificaciones que se han realizado se pretende unificar la sintaxis y el modo de definir una hoja de estilos. (Puig, 2014)

4.1.3.1 Estructura de una regla CSS.

CSS define una serie de términos que permiten describir cada una de las partes que componen una regla de estilos. En la figura 4.1 se muestra las partes que conforman una regla de estilos CSS.

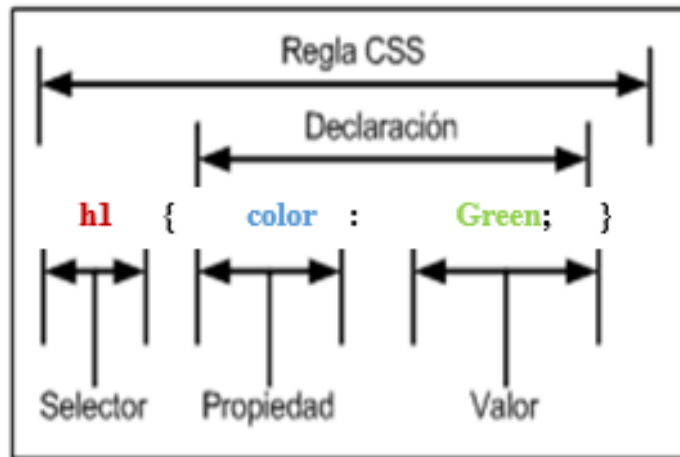


Fig. 4. 1: Estructura de una Regla CSS.

Elaborado por: El Investigador

Los diferentes términos se definen a continuación:

- **Regla:** Se define como regla a cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" (`{`), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" (`}`).
- **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
- **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
- **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
- **Valor:** establece el nuevo valor de la característica modificada en el elemento.

Un archivo CSS puede contener un número ilimitado de reglas CSS, cada regla se puede aplicar a varios selectores diferentes y cada declaración puede incluir tantos pares propiedad/valor como se desee.

El estándar CSS 2.1 define 115 propiedades, cada una con su propia lista de valores permitidos. Por su parte, los últimos borradores del estándar CSS 3 ya incluyen 239 propiedades.(Eguiluz, 2009)

4.1.4 JavaScript para programación de páginas web

HTML5 puede ser imaginado como un edificio soportado por tres grandes columnas: HTML, CSS y Javascript. En los apartados anteriores se ha descrito los elementos incorporados en HTML y las nuevas propiedades que hacen a CSS la herramienta ideal para diseñadores. Ahora es momento de develar lo que puede ser considerado como uno de los pilares más fuertes de esta especificación: Javascript.

JavaScript es un lenguaje de programación de alto nivel que se utiliza para crear páginas web dinámicas. Es decir una página que incorpore efectos tales como textos interactivos, animaciones, acciones que se produzcan al activar cierto botón o elemento, ventanas con mensajes de aviso al usuario entre muchos más. La ventaja que nos provee JavaScript es que es un lenguaje interpretado. Es decir que no es necesario compilar los programas para ejecutarlos. En otras palabras los programas se pueden ejecutar directamente en cualquier navegador sin la necesidad de algún proceso intermediario.(Gauchat, 2012; Pérez, 2009)

4.1.4.1 DOM

La creación del Document Object Model o DOM es una de las innovaciones que más ha influido en el desarrollo de las páginas web dinámicas y de las aplicaciones web más complejas. DOM permite a los programadores web acceder y manipular las páginas XHTML como si fueran documentos XML. De hecho, DOM se diseñó originalmente para manipular de forma sencilla los documentos XML. A pesar de sus orígenes, DOM se ha convertido en una utilidad disponible para la mayoría de lenguajes de programación (Java, PHP, JavaScript) y cuyas únicas diferencias se encuentran en la forma de implementarlo.(Gauchat, 2012; González, 2017; Pérez, 2009; Puig, 2014)

- **Selectores**

El primer paso para poder manipular el DOM, es adquirir cierta destreza en el manejo de los selectores. Ya que estos nos permiten realizar las operaciones de lectura y modificación del DOM. A continuación se describirán los selectores más utilizados.

a) getElementById()

Permite la selección de un elemento por su **id**.

b) getElementsByName()

Permite la selección de varios elementos por su atributo **name**.

c) getElementsByTagName()

Permite la selección de varios elementos por su **etiqueta**.

d) getElementsByClassName()

Permite la selección de varios elementos por su **clase**.

e) querySelector()

Devuelve el primer elemento que coincida con el selector.

f) querySelectorAll()

Devuelve todos los elementos que coincidan con el selector en un pseudo-array.

4.1.4.2 Eventos

Crear páginas y aplicaciones web siempre ha sido mucho más complejo de lo que debería serlo debido a las incompatibilidades entre navegadores. A pesar de que existen decenas de estándares para las tecnologías empleadas, los navegadores no los soportan completamente o incluso los ignoran.

Las principales incompatibilidades se producen en el lenguaje XHTML, en el soporte de hojas de estilos CSS y sobre todo, en la implementación de JavaScript. De todas ellas, la incompatibilidad más importante se da precisamente en el modelo de eventos del navegador.

Así, existen hasta tres modelos diferentes para manejar los eventos dependiendo del navegador en el que se ejecute la aplicación.(Pérez, 2009)

- **Utilización de eventos**

Existen dos formas básicas de añadir eventos a nuestra aplicación. Una es por medio de html con atributos como *onclick*, y la otra desde el JavaScript, haciendo uso de métodos como *.addEventListener()*.Cuál es mejor o cuál es peor realmente varía en función de las circunstancias del código y su programador. A continuación se presenta ejemplos de utilización de estos dos métodos.(González, 2017; Pérez, 2009; Puig, 2014)

De la misma manera en la que se pueden crear eventos también existe la posibilidad de eliminarlos. Esto se hace debido a que estos ocupan memoria, por lo cual es un factor a tomar en cuenta para mejorar el rendimiento de las aplicaciones en especial cuando no se disponer de muchos recursos como es en el caso de los smartphones. Para eliminar un evento se debe utilizar a través de la instrucción *removeEventListener()*.

- **Tipos de eventos**

En este modelo, cada elemento o etiqueta XHTML define su propia lista de posibles eventos que se le pueden asignar. Un mismo tipo de evento (por ejemplo, pinchar el botón izquierdo del ratón) puede estar definido para varios elementos XHTML diferentes y un mismo elemento XHTML puede tener asociados varios eventos diferentes.

El nombre de cada evento se construye mediante el prefijo **on**, seguido del nombre en inglés de la acción asociada al evento. Así, el evento de pinchar un elemento con el ratón se denomina **onclick** y el evento asociado a la acción de mover el ratón se denomina **onmousemove**.(Gauchat, 2012; González, 2017; Pérez, 2009)

La siguiente tabla resume los eventos más importantes definidos por JavaScript:

Tabla 4- 2: *Eventos definidos por JavaScript*

Evento	Descripción	Elementos para los que está definido
Onchange	Deseleccionar un elemento que se ha modificado	<input>, <select>, <textarea>
OnClick	Pinchar y soltar el ratón	Todos los elementos
Ondblclick	Pinchar dos veces seguidas con el ratón	Todos los elementos
Onkeypress	Pulsar una tecla	Elementos de formulario y <body>
Onload	La página se ha cargado completamente	<body>
Onmousemove	Mover el ratón	Todos los elementos
Onmouseover	El ratón "entra" en el elemento (pasa por encima del elemento)	Todos los elementos
Onresize	Se ha modificado el tamaño de la ventana del navegador	<body>
Onselect	Seleccionar un texto	<input>, <textarea>

Fuente: (Pérez, 2009)

4.1.5 AJAX

El termino AJAX significa *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML). AJAX es una nueva técnica para crear aplicaciones web mejores, más rápidas y más interactivas con la ayuda de XML, HTML, CSS y Java Script. Ajax no es una tecnología en sí mismo, sino que en realidad se trata de varias tecnologías independientes que se unen de formas nuevas para alcanzar mejores resultados.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Las aplicaciones web convencionales transmiten información hacia y desde el servidor usando solicitudes síncronas. Significa que debe completar un formulario, presionar enviar y ser dirigido a una nueva página con nueva información del servidor.

Con AJAX, cuando presiona enviar, JavaScript hará una solicitud al servidor, interpretará los resultados y actualizará la pantalla actual. En el sentido más puro, el usuario nunca sabría que algo se transmitió al servidor. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

XML se usa comúnmente como el formato para recibir datos del servidor, aunque se puede usar cualquier formato, incluido el texto sin formato.

AJAX es una tecnología de navegador web independiente del software de servidor web. Un usuario puede continuar usando la aplicación mientras el programa cliente solicita información del servidor en segundo plano. Lo cual nos permite tener una interacción intuitiva y natural del usuario. (Gauchat, 2012; González, 2017; Pérez, 2008)

4.1.5.1 Métodos GET y POST en AJAX

Una de las funcionalidades más grandes de usar AJAX es solicitar datos al servidor mediante el proceso de petición respuesta entre un cliente y un servidor. Lo cual se lo hace a través de una petición HTTP ya sea esta de tipo GET o tipo POST.

- **GET** - Pide datos de un recurso especificado
- **POST** – Envía información a ser procesada por un recurso específico.

GET se utiliza básicamente para recuperar algunos datos del servidor. Cabe destacar que este método puede también retornar información almacenada en cache.

POST también se puede utilizar para obtener información del servidor. Sin embargo, este método nunca almacena en caché los datos, y a menudo se utiliza para enviar datos junto con la solicitud.(Gauchat, 2012; González, 2017; Pérez, 2008)

- **Estructura y Sintaxis de los métodos GET y POST**

Sintaxis de GET

```
$.get(URL, callback);
```

Donde:

- El parámetro **URL** define la dirección del archivo de donde se solicita la información.
- El parámetro opcional **callback** define el nombre de una función que se ejecutará si la solicitud tiene éxito.

El primer parámetro de \$.get () es la URL que deseamos solicitar ("demo_test.asp"). Mientras que el segundo parámetro es una función de devolución de llamada. El primer parámetro de devolución de llamada contiene el contenido de la página solicitada, y el segundo parámetro de devolución de llamada retiene el estado de la solicitud.

Sintaxis de POST

```
$.post(URL, data, callback);
```

Donde:

- El parámetro **URL** especifica la URL que desea solicitar.
- El parámetro **data** es opcional y especifica algunos datos los cuales se pueden enviar junto con la solicitud.
- El parámetro **callback** es opcional y es el nombre de una función que se ejecutará si la solicitud tiene éxito.

El primer parámetro de \$.post () es la URL que deseamos solicitar ("demo_test_post.asp"). Posteriormente en el segundo parámetro se envía los datos de nombre y ciudad junto con la solicitud. El script de ASP “demo_test_post.asp” lee los parámetros, los procesa y retorna un resultado. El tercer parámetro es una función callback, la misma que a la vez posee dos parámetros. El primer parámetro devuelve el contenido de la página solicitada, mientras el segundo parámetro retiene el estado de la solicitud.(Gauchat, 2012; González, 2017; Pérez, 2008)

4.1.6 JQuery

JQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, y fue diseñada para simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC y actualmente jQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

La librería JQuery se encuentra en un archivo pequeño y que se puede descargar desde www.jquery.com y luego incluirlo en los documentos usando la etiqueta `<script>`. Provee una API sencilla que cualquiera puede aprender y rápidamente aplicar a sus proyectos. Una vez que el archivo provisto por jQuery es incluido en nuestro documento, ya estamos listos para aprovechar los métodos simples incorporados por la librería y convertir nuestra web estática en una moderna y práctica aplicación. jQuery tiene la ventaja de proveer soporte para viejos navegadores y vuelve simple tareas cotidianas. Siendo posible su utilización junto con HTML5 o como una forma simple de reemplazar funciones de HTML5 en navegadores que no están preparados para esta tecnología.(Gauchat, 2012; González, 2017; Pérez, 2009)

4.1.6.1 Métodos de JQuery

A continuación se describirá los métodos más importantes de JQuery.

- **append()**

El método `append()` añade contenido html a un determinado contenedor o etiqueta.

- **appendTo()**

El método appendTo() copia o mueve el contenedor de un determinado elemento a otro.

- **attr()**

La función attr() retorna el valor del atributo que se pase como parámetro de un elemento o contenedor.

- **click()**

Permite ejecutar un determinado proceso al presionar un click en el elemento seleccionado.

- **dblclick()**

Al hacer doble click en el evento o contenedor con id="button" ejecutaríamos el código de dentro de la función.

- **height()**

Con la función height() modificamos el alto del contenedor.

- **hide()**

Con la función hide() lo que hacemos es ocultar el contenido de la etiqueta seleccionada.

- **position()**

La función position() permite obtener la posición de un elemento en relación a su "padre".

- **ready()**

Ready es un método propio de jQuery, que revisa si el DOM está listo para usarse. Es más útil que el window.onload, pues este debe esperar a que todos los elementos de la página estén cargados (como scripts e imagenes) para ejecutar. El "ready", en cambio, espera solo a la estructura.

- **remove()**

La función `remove()` permite remover la etiqueta seleccionada.

- **val()**

La función `val()` permite obtener el valor de la del control o etiqueta html seleccionada.

- **width()**

Con la función `width()` modificamos el ancho de la etiqueta seleccionada.

4.1.7 JSON

JavaScript Object Notation o JSON es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o de sus rendimientos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de `eval()` y el auge del procesamiento nativo de XML incorporado en los navegadores modernos. Por esa razón, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc, que atienden a millones de usuarios) cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente.

Si bien es frecuente ver JSON posicionado contra XML, también es frecuente el uso de JSON y XML en la misma aplicación. Por ejemplo, una aplicación de cliente que integra datos

de Google Maps con datos meteorológicos en SOAP hacen necesario soportar ambos formatos.(González, 2017; Murphey et al., 2011)

4.1.8 OPC UA

Open Productivity Collaboration Unified Architecture (OPC UA) fue desarrollado con el objetivo de superar las limitaciones que presentaba su antecesor OPC Clásico. El mismo fue diseñado para funcionar únicamente con sistemas Operativos Windows. Actualmente OPC UA es una plataforma independiente de los fabricantes y por tanto abierta ya que asegura un flujo de información entre componentes de distintos proveedores, siendo la OPC Foundation la asociación encargada del desarrollo de este estándar. OPC UA es una tecnología diseñada para comunicar datos de forma segura y operativa entre aplicaciones tanto en la industria de la automatización como en otros sectores,

La nueva especificación OPC UA se ha creado debido a que cada vez son más los retos en seguridad y modelado de datos. Por otro lado era necesario un protocolo más abierto a otros sistemas operativos y por tanto aplicable a tecnologías futuras.(Mahnke, Leitner, & Damm, 2009)

OPC UA tiene dos innovaciones o características que hay que destacar respecto al protocolo OPC clásico:

1. Reemplaza el protocolo COM y DCOM, específico de Windows, por protocolos abiertos e independientes para que puedan funcionar bajo otros sistemas operativos con importante implantación no solo a nivel de equipo informático como Windows, Linux, Mac, etc sino también a nivel de dispositivos de telefonía móvil como por ejemplo Android y diferentes tipos de controles y dispositivos de monitorización, sensores, controladores etc. Que interactúan con el mundo real incorporando además mecanismos de seguridad adicionales.(Mahnke et al., 2009)
2. Incorpora el modelo de información orientado a objetos que aglutina las funcionalidades tradicionales de OPC (como acceso a datos, históricos, alarmas, eventos, condiciones...) y otras nuevas e innovadoras orientadas a los tipos de datos y métodos.(Mahnke et al., 2009)

La característica de que la estructura del direccionamiento en los servidores OPC UA esté orientada a objetos y de que la interfaz para el acceso a dicha estructura sea totalmente genérica da lugar a que OPC UA se considere no solo una pasarela de comunicación entre dos equipos sino también un lenguaje de programación con capacidades de comunicación a través de redes.

OPC UA utiliza plataformas escalables, múltiples modelos de seguridad, múltiples capas de transporte y un sofisticado modelo de información para permitir que el controlador dedicado más pequeño interactúe libremente con aplicaciones complejas de servidor de alto nivel. OPC UA puede comunicar cualquier cosa, desde un simple estado de inactividad hasta cantidades masivas de información altamente compleja en toda la planta. (Mahnke et al., 2009)

OPC UA es un mecanismo sofisticado, escalable y flexible para establecer conexiones seguras entre clientes y servidores. Las características de esta tecnología se describen a continuación:

- a) **Escalabilidad:** OPC UA es escalable e independiente de la plataforma. Se puede admitir en servidores de gama alta y en sensores de gama baja. OPC UA utiliza perfiles detectables para incluir pequeñas plataformas integradas como servidores en un sistema OPC UA.
- b) **Un espacio de direcciones flexible:** el espacio de direcciones OPC UA se organiza en torno al concepto de un objeto. Los objetos son entidades que constan de variables y métodos y proporcionan una forma estándar para que los servidores transfieran información a los clientes.
- c) **Codificaciones y transportes comunes:** OPC UA utiliza transportes y codificaciones estándar para garantizar que la conectividad se pueda lograr fácilmente tanto en el entorno integrado como en el empresarial.
- d) **Seguridad:** OPC UA implementa un sofisticado modelo de seguridad que garantiza la autenticación de clientes y servidores, la autenticación de usuarios y la integridad de su comunicación.
- e) **Capacidad de Internet:** OPC UA es totalmente capaz de mover datos a través de Internet.
- f) **Un conjunto robusto de servicios:** OPC UA proporciona un conjunto completo de servicios para eventos, alarmas, lectura, escritura, descubrimiento y más.

- g) **Interoperabilidad certificada:** OPC UA certifica los perfiles de modo que se pueda garantizar la conectividad entre un cliente y un servidor utilizando un perfil definido.
- h) **Un modelo de información sofisticado:** OPC UA perfila más que solo un modelo de objeto. OPC UA está diseñado para conectar objetos de tal manera que la verdadera información pueda ser compartida entre clientes y servidores.
- i) **Sofisticada gestión de alarmas y eventos:** OPC UA proporciona un mecanismo altamente configurable para proporcionar alarmas y notificaciones de eventos a los clientes interesados. Los mecanismos de eventos y alarmas van mucho más allá de las alarmas de tipo de cambio de valor estándar que se encuentran en la mayoría de los protocolos.
- j) **Integración con modelos estándar de datos específicos de la industria:** la Fundación OPC está trabajando con varios grupos comerciales de la industria para definir modelos de información específicos para sus industrias y para respaldar esos modelos de información dentro de OPC UA.

4.1.9 IDEs de programación web

Existe una gran variedad de softwares empleados para desarrollar aplicaciones web y a continuación describiremos los más utilizados y difundidos.

4.1.9.1 Sublime Text

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

Se puede descargar y evaluar de forma gratuita. Sin embargo no es software libre o de código abierto y se debe obtener una licencia para su uso continuado, aunque la versión de evaluación es plenamente funcional y no tiene fecha de caducidad. Actualmente se encuentra en la versión número 3.

Entre las características más sobresalientes de este editor de código tenemos: la capacidad de pre visualizar la estructura del código, soporte de hasta 43 lenguajes de programación en texto plano, realización de búsquedas dinámicas, autocompletado y marcado de llaves, soporte para

diferentes plugins, coloreado y envoltura de sintaxis entre muchas otras más. Lo cual permite que la programación de las diferentes aplicaciones sea torne más sencilla. (Sublime Text, 2018)

4.1.9.2 Adobe Dreamweaver CC

Dreamweaver es una de las herramientas de diseño web más potente y conocida en el mercado y ha existido en diversas formas desde el cambio de siglo. La última versión de Dreamweaver está disponible como parte de la suite Creative Cloud de Adobe. Provee un atractivo tanto para usuarios principiantes como avanzados. Permitiendo la opción de elegir entre niveles de habilidad "principiante", "intermedio" y "avanzado". Esto ayuda a determinar la cantidad de funciones que se le muestran y si se le darán opciones para realizar tareas más complejas.

Además de una interfaz potente y de diseño refrescante, Dreamweaver admite prácticamente cualquier tipo de código web, desde HTML básico hasta CSS, PHP y JavaScript, por nombrar solo algunos. Si aún no está listo para editar el código sin procesar, Dreamweaver admite plantillas de WordPress y Drupal.

Otras características avanzadas incluyen una "visualización en vivo" para la vista previa de sitios web, así como soporte para certificados y un espacio de trabajo dedicado para desarrolladores.(Macromedia, 2018)

4.1.9.3 Bluefish

Bluefish es una de las herramientas de diseño web más pequeñas disponibles en la actualidad. El pequeño instalador pesa poco menos de 53 MB y la configuración solo demora unos minutos. Si bien la interfaz es solo de texto, está claramente diseñada teniendo en cuenta a los novatos ya que emplea barras de herramientas definidas, menús personalizables por el usuario y resaltado de sintaxis.

Si bien el foco principal está en HTML, Bluefish admite una amplia gama de otros lenguajes, incluidos PHP, Java, JavaScript, SQL, XML y CSS. A diferencia de las herramientas visuales de diseño web WYSIWYG, la interfaz de texto proporciona un código mucho más limpio.

Bluefish tiene una función de búsqueda excelente, que le permite encontrar texto en múltiples proyectos. La herramienta tampoco tiene problemas para hacer malabarismos con cientos de documentos al mismo tiempo. Aunque Bluefish admite el trabajo con archivos remotos, los diálogos y asistentes variados y útiles actualmente no admiten la carga directa de páginas web a través de FTP.(Bluefish, 2018)

4.1.10 Norma ISA 101 para la creación de interfaces hombre maquina (HMI)

Aunque la mayor parte de la investigación se centra en el estudio y análisis de las técnicas, lenguajes y herramientas para la construcción de sitios web es muy necesario estudiar la normativa que se debe seguir para la creación de pantallas HMI. Esto se lo debe hacer con la finalidad de proveer al usuario de la plataforma web todas las herramientas para el diseño adecuado de una pantalla HMI, siguiendo todas las reglas de visualización sugeridas por las normativas específicas que dictaminan la manera de diseñar pantallas HMI.

Aunque en sus principios los sistemas que se utilizaban para controlar procesos eran tecnológicamente simples, debido a que los procesos que se controlaban eran pequeños y no estaban dotados de tanta complejidad, con el tiempo han aumentado tanto en tamaño como en complejidad. Por lo cual ya no basta con uno o dos indicadores sino que se ha hecho necesario la utilización de muchos símbolos, indicadores, paneles sinópticos entre otros. Ante esta problemática han surgido a nivel internacional varias iniciativas a seguir a cerca del diseño de las interfaces HMI.

A pesar de todas estas normativas que se han desarrollado no se ha planteado aun un camino claro al cual seguir. Ante la falta de estándares claros en torno al mundo de la interface hombre maquina se crea, en Julio de 2005, el comité ISA-SP 101 con el objetivo de establecer estándares, practicas recomendadas y apoyo técnico, que permitan normalizar el campo de las interfaces hombre-máquina en los procesos productivos.

Esta normativa se creó con los siguientes objetivos: disminuir la tasa de errores, reducir los tiempos de aprendizaje y reducir costes de rediseño. La ISA administra los Grupos de Soporte Técnico de los Estados Unidos (USTAGs, United States Technical Advisory Groups) y

proporciona apoyo administrativo para el Comité Electrotécnico Internacional (IEC, International Electrotechnical Commission) y la Organización Internacional para la Estandarización (ISO, International Organization for Standardization) encargados de la estandarización de los procesos de control y medida. (Eras et al., 2010)

A continuación se detallará las particularidades de la presentación y los métodos de interacción con los objetos de las pantallas, así como una visión global de la consola de operaciones propuesta en la norma ISA 101, para la correcta creación de interfaces hombre máquina.

Consola de Operaciones

Mediante el uso adecuado de los elementos de entrada y salida se debe determinar una consola de operaciones la cual contenga la definición de HMI, esta debe constar de los elementos claves tales como: consola, estación, teclado, pantalla, mouse, ventanas Pop-up, elementos y símbolos gráficos.

Jerarquía de pantallas

Otro aspecto importante a tomar en cuenta es la jerarquía de las pantallas, ya que será de gran ayuda al momento de definir la navegación por la Interfaz. Esto permitirá una mejor navegación entre las pantallas de las estaciones a monitorizar.

Distribución de la pantalla

Las diferentes pantallas del HMI deben contar con una correcta distribución de su espacio de trabajo cumpliendo las normas establecidas para el diseño de las mismas, el buen uso de la pantalla exige ocupar todo el espacio de trabajo con el fin de generar la mayor información y la más adecuada para el usuario

Elementos Estáticos

Se debe tener en cuenta que existirán elementos del HMI que no proporcionarán una dinámica en el mismo pero que servirán de información dentro del proceso como es el caso de la fecha y hora así como graficas o logos de productos, marcas, o corporaciones. Dentro de los procesos industriales una característica muy importante al momento de visualizar una pantalla es proporcionar toda la información necesaria para que el operador del sistema no cree conflicto con el entorno de trabajo, por ende, es de suma importancia que los elementos estáticos

produzcan la información correcta de lo que se está realizando ya sea el caso de un mímico que no tenga movimiento pero que sea una representación original y de gran valor visual para el operador.

Texto Estático

Los títulos, despliegue de ayudas, etiquetas de valores, etc., son ejemplos de texto estático dentro de un interfaz humano máquina, estas no cambiarán en el tiempo, pero son de esencial importancia ya que proporciona al usuario la información del entorno de manera clara y detallada.

El título de cada pantalla es un claro ejemplo de texto estático no cambiará su información, pero sin él no se sabría qué proceso, subproceso, mensaje de ayuda o entorno de operación se encuentra el operador del sistema.

Valores dinámicos del proceso

Los valores dinámicos del proceso son valores cambiantes que a medida que se ejecuta una operación en el sistema éstos se actualizan, deben ser claros y por lo general están ligados directamente a los valores que entrega la estación de trabajo.

Navegación

Una vez creada la jerarquía de las pantallas se puede determinar la forma en la que se realizará la navegación teniendo en cuenta el nivel, forma de navegación y las herramientas para realizarlo. (Penin, 2007; Suarez, 2017)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- De la información analizada podemos decir que ante la gran variedad de lenguajes y técnicas existentes para la construcción de aplicaciones web existen tres que son de vital importancia para la creación de cualquier aplicación siendo estas: HTML, CSS y JavaScript respectivamente. Por lo que su dominio es de suma importancia para el desarrollo del presente proyecto de investigación.
- Al estudiar los tres lenguajes para programar aplicaciones web descritos anteriormente podemos concluir que a diferencia de otros lenguajes de programación los cuales poseen todas las herramientas para generar la definición, apariencia y lógica de cada aplicación, en las aplicaciones web no sucede lo mismo. Debido que en este ámbito tenemos tres lenguajes bien diferenciados pero complementarios entre sí. HTML que se encarga de la definición de elementos llamados etiquetas, CSS encargado de la apariencia de las aplicaciones y JavaScript encargado de proporcionarle lógica y dinamismo a las aplicaciones.
- Aunque los tres lenguajes de programación web descritos anteriormente son los más importantes para la programación es importante definir que método se utilizara para establecer la comunicación del PLC. Mediante el estudio de las diferentes técnicas para programar aplicaciones web se ha concluido que la utilización la AJAX para el intercambio de información será el corazón entre nuestra aplicación y el PLC para poder acceder a la información que se monitoreara.
- Cabe recalcar que aunque el aspecto técnico juega un papel de gran importancia en el desarrollo del presente proyecto siempre debe ir de la mano con la normativa establecida. Del análisis realizado sobre las diferentes normativas vigentes se ha podido determinar que la norma ISA 101, así como las recomendaciones de diseño para pantalla de monitorización de la norma ISO son las más adecuadas a tomar en cuenta para el diseño de la propuesta establecida.

- La elección de la correcta opción del protocolo de comunicación es vital para el exitoso funcionamiento del proyecto de investigación propuesto. Es por ello que después de analizar los diferentes protocolos de comunicación se ha concluido que la mejor opción para el desarrollo del proyecto es OPC UA ya que presenta mejoras significativas sobre los otros protocolos tales como: la escalabilidad, la seguridad de la información, la flexibilidad del espacio de direccionamiento y la capacidad de interactuar con el internet.

5.2 Recomendaciones

- Aplicar las metodologías establecidas por las distintas normativas existentes al momento de definir la apariencia y forma de los controles a desarrollarse para la aplican web.
- Tener a consideración el trabajo mediante la separación de diferentes técnicas de programación. Es decir escribir el código de cada lenguaje como CSS o HTML en archivos separados y luego enlazarlos al archivo principal. Ya que nos permitirá tener aplicaciones más escalable así como disminuir el grado de complejidad de las mismas.
- Se recomienda hacer uso del atributo class cuando se trabaja con características similares, especialmente a la hora de modificar su apariencia mediante CSS. Lo cual nos permitirá modificarlo una sola vez para que se realice el cambio correspondiente en toda la aplicación. Logrando de esta manera reducir sustancialmente las líneas de código.
- Evitar la introducción de código CSS o JavaScript dentro del código HTML o hacerlo únicamente cuando sea absolutamente necesario o cuando el código ingresado sea utilizado para una acción específica.
- Tener en cuenta que muchos de los procesos que se necesitan usar para realizar una determina acción no es necesario programarlos ya que JQuery posee varias herramientas que permiten realizar la misma acción con tan solo pocas líneas de código.

CAPÍTULO VI

PROPUESTA DE INVESTIGACIÓN

6.1 Antecedentes de la propuesta

El gran desarrollo que ha tenido el internet ha permitido que nuevas tecnologías y mejores dispositivos se desarrollen continuamente. Lo cual ha hecho que el campo de la monitorización de procesos debe adaptarse al cambiante entorno tecnológico en el que nos encontramos. Es por ello que el desarrollo de sistemas que permitan adaptarse a las nuevas tecnologías es indispensable. De ahí surge la idea de desarrollar una nueva propuesta para crear interfaces hombre maquina HMIs que permitan interactuar con el usuario de forma sencilla e intuitiva y que aprovechen los nuevos avances en materia de desarrollo web y comunicaciones industriales. Los mismos que hacen posible el desarrollo de nuevos y mejores sistemas compatibles con el cambiante mundo del internet y los datos actuales.

En la investigación efectuada en artículos científicos y proyectos realizados se ha encontrado diferentes propuestas que comparten varias características en común con el trabajo de investigación desarrollado por lo cual es de gran relevancia analizarlas. Por este motivo se presentan a continuación los trabajos de investigación más relevantes:

En la investigación desarrollada por Miriam Schleipen, Syed-Shirz Gilani, Tino Bischoff y Juluis Pfrommer denominada “OPC UA & Industrie 4.0 – enabling technology with high diversity and variability” se analiza las presentes demandas de la industria 4.0 especialmente en el campo de las comunicaciones. Ya que se espera encontrar un nuevo medio de intercambio de comunicación que provea flexibilidad, adaptabilidad, transparencia y muchos más requerimientos que demanda la industria 4.0. Por lo cual se considera como una alternativa adecuada la utilización del protocolo OPC UA. Debido a la variabilidad y flexibilidad que este protocolo provee, pudiendo ser utilizado desde simples procesos como adquisición de datos hasta procesos extensos y complejos como la monitorización, control y análisis de lo diferentes procesos de la industria. Además brinda de una alternativa de comunicación que permite la integración y combinación con varios dispositivos y estándares.(Schleipen, Gilani, Bischoff, & Pfrommer, 2016)

El estudio y análisis del trabajo de investigación denominado “Integration of OPC UA into a Web-based Platform to enhance interoperability” es de gran importancia para el desarrollo de la presente investigación ya que se asemeja a la propuesta planteada. En la investigación se presenta el diseño de una plataforma web capaz de acceder a uno o más servidores OPC UA, con el propósito de intercambiar información a través de la web. Además se plantea la idea de crear una plataforma en la que el usuario no tenga la necesidad de poseer conocimientos avanzados en cuanto a protocolos de comunicación, necesarios para crear aplicaciones basadas en el protocolo OPC UA. Se propone crear una interfaz en la que el usuario no tenga que lidiar con líneas de código o complicadas instrucciones de operación. Se hace también énfasis en la utilización del OPC UA debido a la seguridad que una plataforma que intercambia información de procesos industriales a través del internet debe poseer. Siendo ideal la utilización del protocolo OPC UA para el desarrollo de esta iniciativa debido a las grandes opciones de seguridad que proporciona.(Cavaliere, Di Stefano, Salafia, & Scropo, 2017)

Otro trabajo de investigación de gran interés es el desarrollado por Sten Gruner, Julius Pfrommer y Florian Palm denominado “Una extensión RESTful de OPC UA”. Este trabajo de investigación propone crear un conjunto de extensiones simples que permitan la comunicación RESTful para el protocolo OPC UA; aprovechando que las interfaces REST poseen un estilo de arquitectura muy extendido para implementaciones de servicios web y se basan en el enfoque orientado a los recursos para arquitecturas no centralizadas. Con lo cual se pretende mejorar el uso de recursos de comunicación para solicitudes de servicios esporádicos, permitiendo además que las aplicaciones se beneficien de las ventajas del estilo de arquitectura orientada a los recursos, como el almacenamiento y el acoplamiento de aplicaciones sueltas.(Gruner, Pfrommer, & Palm, 2015)

Como protocolo de comunicación estandarizado, OPC UA es el principal punto focal con respecto al intercambio de información en la iniciativa Industria 4.0. Pero también hay consideraciones para usarlo dentro de la Internet de las cosas (IoT). En la investigación denominada “Open Source as Enabler for OPC UA in Industrial Automation” se discute el hecho de que actualmente no se puede utilizar una implementación de referencia abierta en la investigación de forma gratuita. Lo cual representa un problema importante en este contexto. Se discute además como la utilización de código abierto podría estabilizar el trabajo teórico. Se analiza además los esfuerzos recientes para desarrollar una implementación abierta para OPC

UA que permitan cumplir con los requisitos de la tecnología de automatización práctica e industrial. Por lo cual se estudia la iniciativa open62541 proveyendo una descripción general de sus campos de aplicación y los principales problemas de investigación (Palm, Grüner, Pfrommer, Graube, & Urbas, 2015). Este trabajo de investigación es crucial para el desarrollo de la presente investigación puesto que se analizan alternativas del protocolo OPC UA de código libre lo cual nos permite tener una idea más clara al momento de elegir la correcta plataforma de utilización del protocolo.

En la investigación titulada “RESTful Industrial Communication With OPC UA” se explora las ventajas de las arquitecturas RESTful, es decir, las arquitecturas de software orientadas a servicios RESTful en entornos industriales. Las cuales proponen el uso de un conjunto fijo de interfaces de servicios para transferir representaciones de recursos heterogéneos en lugar de definir interfaces personalizadas para aplicaciones individuales. Entre estos servicios se incluyen las ventanas de comunicación como reducción a la sobrecarga de la comunicación y la posibilidad de introducir capas de almacenamiento en caché y ventajas de diseño del sistema; que incluyan interfaces de servicio estables en todas las aplicaciones y el uso de modelos de información orientados a los recursos en sistemas cibernéticos. Adicionalmente, Se propone crear una extensión RESTful para el protocolo binario de arquitectura unificada de comunicaciones de plataforma abierta (OPC UA) para aprovechar estas ventajas. Para llevar a cabo este objetivo se propone realizar modificaciones mínimas en las pilas OPC UA existentes sin causar cambios significativos en protocolo y dejándolo completamente funcional.(Gruner et al., 2015)

6.2 Justificación

En un entorno tecnológico que avanza a pasos agigantados las propuestas existentes para la monitorización y control de procesos deben avanzar a medida que la tecnología también lo hace siendo necesaria la creación de nuevas tecnologías para satisfacer las presentes necesidades del mercado empresarial y tecnológico (Revista Electroindustria, 2007). Es por ello que el desarrollo de un proyecto de investigación como el planteado es de gran ayuda para dotar de nuevas alternativas al momento de optar una herramienta para la monitorización de procesos industriales.

Hoy en día tanto los fabricantes como los distribuidores están buscando mejores formas de administrar de manera más eficiente los activos de sus operadores. Los avances en las tecnologías y estándares de comunicación actuales hacen que sea más fácil y más justificable entregar información desde la instrumentación de medición y permitir la administrar de los activos de instrumentación de manera más eficiente. Por lo que un enfoque flexible de integración entre los nuevos estándares de comunicación como OPC UA y las tecnologías para la administración de dispositivos de automatización de procesos, basados en servicios web es muy necesario. Por estos motivos la propuesta planteada deberá estar enfocada a proveer una solución a la problemática expuesta. (Van Tan, Yoo, & Yi, 2009)

En el futuro, los sistemas de monitorización se encontraran en componentes tan diversos que podrán estar situados en los dispositivos más diminutos y sencillos, con los cuales se podrá realizar la monitorización de procesos completos, independientemente de las plataformas y proveedores, ya que la mayoría de la información tendera a distribuirse a través del internet. Estos constantes cambios son causados por las demandas de los clientes, los cuales hoy en día demandas cada vez productos altamente personalizados. Para lograr obtener sistemas que se adapten a las cambiantes necesidades tanto del entorno tecnológico como del mercado es necesario el desarrollo de nuevas propuestas que permitan satisfacer estas necesidades. Por lo cual la profundización en el estudio y utilización de tecnología como las Arquitectura Unificada OPC (OPC-UA) y el desarrollo de nuevas tecnologías basados en servicios web será primordial.(Durkop, Imtiaz, Trsek, Wisniewski, & Jasperneite, 2013)

El desarrollo de un trabajo de investigación como el planteado permitirá incrementar el nivel de acceso a aplicaciones enfocadas a crear pantallas de monitorización. Incentivando de esta manera a incrementar los procesos de automatización dentro de la industria. Puesto que al utilizar alternativas de código libre tanto para el desarrollo de interfaces como en el intercambio de comunicación los costos de acceso a estos productos ya no supondrán una barrera tan alta a superar. Permitiendo generar productos de mejor calidad así como ganar en eficiencia y reducir los costos en errores de manufactura y producción.

6.3 Objetivos de la propuesta

6.3.1 Objetivo General

Crear un prototipo de una aplicación web de libre acceso para la creación de pantallas de monitorización HMI para los PLC-S7 1200.

6.3.2 Objetivos Específicos

- Utilizar la información bibliográfica como base para el desarrollo de la propuesta, considerando los distintos lenguajes de programación web utilizados para crear aplicaciones.
- Determinar la mejor forma de interactuar con los extractos de código para el desarrollo de la aplicación, así como la selección del mejor IDE para programar la aplicación.
- Seleccionar los mejores plugins para crear los instrumentos virtuales que formaran parte de la aplicación web.
- Determinar el mejor modo de establecer la comunicación entre la aplicación para crear las pantallas de monitorización y el PLC.

6.4 Análisis de factibilidad

6.4.1 Factibilidad Operativa

El dominio tanto de las características de los distintos lenguajes de programación así como de las herramientas proveídas por el IDE de programación utilizado, nos facilitara el desarrollo de la propuesta. Donde el entendimiento de las técnicas necesarias para la construcción de los elementos virtuales permitirá llevar a cabo la construcción de una aplicación exitosa que satisfaga todas las necesidades planteadas.

6.4.2 Factibilidad Técnica

Una vez analizada toda la información recolectada se ha determinado que el proyecto propuesto es factible. Siendo dos grandes barreras las importantes a superar la integración de los elementos virtuales en una interfaz estable y amigable para el usuario. Y la correcta construcción de la comunicación entre los instrumentos virtuales y nuestro controlador.

6.4.3 Factibilidad Económica

El estudio de las técnicas de programación de código libre hace del factor económico un aspecto sencillo de superar en lo referente a software. Puesto que los lenguajes que se utilizaran para el desarrollo del presente proyecto son gratuitos, pudiendo ser creados y modificados desde cualquier editor de texto sencillo como el bloc de notas. Siendo el factor más importante a considerar la adquisición del controlador. El mismo que se ha investigado su costo pudiendo bordear desde los 500 a 800 dólares en dependencia de su estado. Considerando de esta manera al factor económico como un aspecto posible de ser superado.

6.4.4 Proyección a futuro

Se espera que esta propuesta a largo plazo pueda ser mejorada y adoptada en la monitorización de procesos. Logrando con el tiempo su compatibilidad con más controladores de distintos proveedores. Proveyendo una herramientas de bajo costo para las diferentes industrias. Se espera además que esta alternativa permita impulsar la automatización y monitorización de procesos así como ser una muy buena alternativa en el ámbito estudiantil para la creación y el estudio de aplicaciones de monitorización web.

6.5 Fundamentación Científico Técnica

6.5.1 IDE de programación seleccionado

En el campo de la programación web la ventaja que tenemos es la facilidad que nos provee al momento de generar nuestro código. Ya que es tan simple como escribirlo en un editor de texto cualquiera, sin la necesidad de utilizar complicados y pesados softwares y compiladores.

Aunque existe una gran variedad de IDEs para generar nuestro código. Se ha elegido el software Sublime Text debido a su sencillez y simplicidad. Ya que este software permite añadir y enlazar cualquier parte de código desde un archivo externo de manera sencilla simplemente enlazándolo a través de la etiqueta link, para el caso de los estilos CSS y Script para el caso del código Javascript. Otro aspecto por lo cual se ha elegido este software, es la distinción que nos permite hacer de los diferentes tipos de atributos y lenguajes utilizados en nuestro proyecto mediante colores. Siendo esta la consideración más importante que se ha tomado en cuenta ya que la

extensa cantidad de código utilizado en la programación web a menudo hace que sea muy complicado la diferenciación de las distintas partes de nuestro proyecto.

6.5.2 PLC S7-1200 y TÍA PORTAL

Para el desarrollo de código de nuestro controlador se ha elegido el software para programación de PLCs TIA PORTAL ya que presenta un ambiente amigable para la programación haciéndolo simple e intuitivo. Presentado la ventaja de poder programar en varios bloques de programación y crear subrutinas que nos permiten hacer proyectos más escalables. Además de soportar la programación del controlador que hemos elegido el PLC S7-1200.(Siemens, 2014)

6.5.3 Elección del protocolo de comunicación.

Existen un sin número de protocolos de comunicación desarrollados para poder intercambiar información entre las aplicaciones web y los diferentes controladores existentes. Entre estos protocolos los más representativos son: HTTP, MQTT, API REST, WAMP y OPC UA. Del estudio realizado se ha optado como mejor opción la utilización del protocolo OPC UA debido a su alto grado de confiabilidad al momento de intercambiar información. Esto se debe al alto grado de seguridad que este protocolo proporciona. Lo cual permite que se pueda intercambiar información entre las aplicaciones desarrolladas de forma confiable. Además que permite que esta información ahora se pueda enviar a través de internet con más fiabilidad y se puedan desarrollar aplicaciones y proyectos de más alcance y más elaborados. Otro punto importante por el cual se ha elegido este protocolo es la velocidad con la cual se comparte la información. Al haber realizado pruebas con otros protocolos como HTTP y MQTT se encontró que a medida que se incrementa el número de variables a monitorizar el intercambio de información se ralentiza, cosa que no sucede con el protocolo OPC UA ya que se ha observado que el proceso de intercambio de información no se ve afectado tan drásticamente al incrementar las variables de monitorización. Esto se debe en gran medida al proceso de suscripción con el cual funciona este protocolo. Lo cual permite que la comunicación se realice de una manera más eficiente. Finalmente se escogió este protocolo ya que permite hacer una diferenciación de las variables que se monitorizara ya que permite identificar a cada variable mediante un node Id único para cada variable. Esta característica nos permite diferenciar de manera única a cada instancia de monitorización algo que no sucede en protocolos como MQTT en el cual se puede tener

instancias de monitorización con los mismos identificadores pero de diferentes procesos, pudiendo con esto causar problemas o tener lecturas erróneas al momento de monitorizar un determinado proceso.

Debido a estas ventajas que presenta este protocolo en relación con los demás descritos anteriormente se ha optado por utilizar el protocolo como medio de intercambio de comunicación entre la plataforma web desarrollada en el controlador utilizado.

6.5.4 Diagrama de funcionamiento de la plataforma web.

Antes de describir el proceso de desarrollo de la plataforma realizada es necesario entender la arquitectura de nuestra aplicación. Misma que permitirá la comprensión del funcionamiento básico del sistema planteado y la interacción con el controlador. Esta interacción permitirá adquirir la información necesaria para la monitorización de los diferentes procesos a realizarse. Por este motivo se ha diseñado un esquema que visualice la arquitectura de la plataforma que se ha desarrollado. Ver figura 6.1

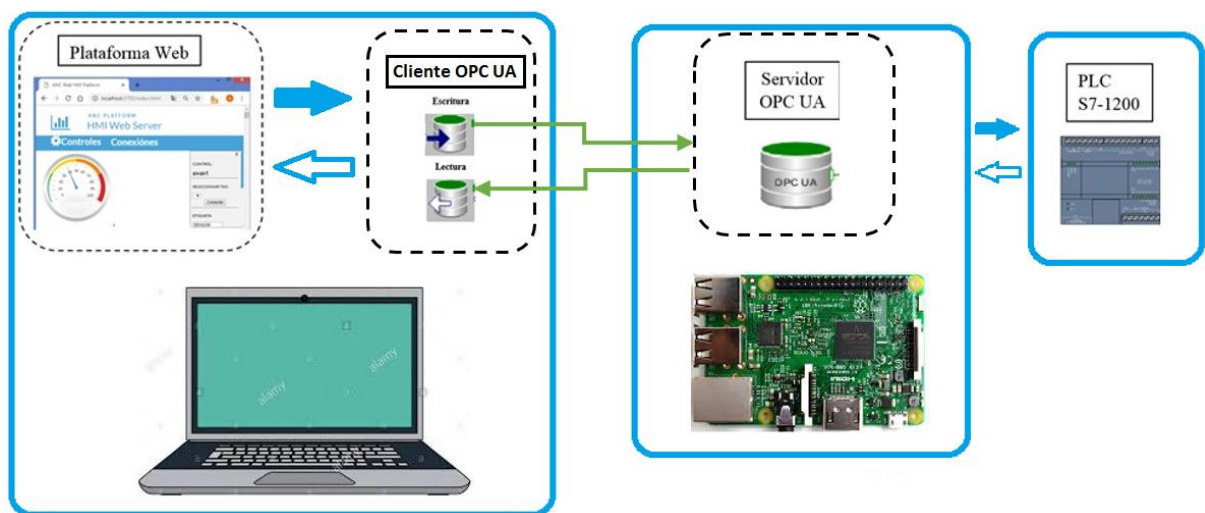


Fig. 6. 1: Diagrama de funcionamiento de la plataforma Web

Elaborado por: El investigador

La arquitectura del sistema desarrollado consta de un cliente OPC UA el cual permite interconectar la aplicación web con el servidor OPC UA. A su vez el servidor OPC UA puede estar contenido en cualquier dispositivo que pueda soportar y ejecutar dicho protocolo. Por último se tiene al controlador como dispositivo difusor de información a procesar y monitorizar.

El proceso se inicia cuando se ejecuta la plataforma web en el navegador, con el objetivo de crear una pantalla de monitorización. Al iniciar este proceso se debe ingresar ciertos parámetros. Estos parámetros son la dirección IP del controlador con el cual se desea establecer una conexión y las variables del controlador que se desea monitorizar. Al realizar este proceso la plataforma web envía esta información a través del protocolo socket.io hacia una instancia que se ha creado en el cliente OPC UA, esta instancia permite gestionar las variables que se van a monitorizar, recibe la información de la variable y la envía al servidor OPC UA para poder crear una nueva instancia de monitorización, la cual posteriormente puede ser accedida desde cada elemento de la plataforma web para suscribirse y recuperar información.

El servidor OPC UA es el encargado de gestionar tanto la información que proviene de la plataforma web a través del cliente OPC UA, así como la información que llega al servidor desde el controlador a través del protocolo NODE S7. Primeramente se recibe la información del controlador con el cual se desea establecer una conexión y se envía la dirección IP del controlador hacia el protocolo NODE S7. Posteriormente recibe los parámetros de las variables del controlador que se desea monitorizar como son su nombre y el node Id que se procederá a asociar a dicho nombre. Con estos dos parámetros el servidor OPC UA procede crear un arreglo de variables para monitorizar, y creando además una tabla de valores que son recuperados a través del protocolo NODES7. Es decir que cada variable del servidor OPC UA tendrá una dirección de memoria del PLC asignada a la misma, desde la cual se extraerá la información para su posterior monitorización.

El proceso que realiza tanto el servidor y cliente OPC UA es transparente para el usuario final. Logrando con esto simplificar la utilización de la plataforma web y proveyendo al usuario de una interfaz intuitiva y fácil de utilizar. El proceso que realiza la plataforma web es recuperar la información que se encuentra en el controlador a través del proceso de suscripción a las variables que se hayan designado para su monitorización. Como se ha descrito anteriormente

el servidor es el que crea estas instancias de monitorización y el cliente es el que accede a las mismas cuando se seleccione una variable a través de la plataforma web. Esta información se recupera por el cliente y se envía hacia cada instrumento virtual de la plataforma web a través del protocolo socket.io. Al recibir la información la programación de los controles permite presentar esta información de la manera más estética posible haciendo uso de efectos visuales o indicadores de texto desarrollados en cada control virtual. Dando como resultado final una pantalla de monitorización que permitan identificar de manera eficiente el estado del proceso que se está monitorizando.

6.5.5 Diseño de la estructura de la plataforma mediante HTML

Para el diseño del presente proyecto se ha escogido una estructura sencilla de manera que sea intuitiva para el usuario. La cual consta de un encabezado en donde se despliega el nombre de la aplicación. Posteriormente se ha creado una barra donde se ha colocado el botón de herramientas y el botones conexiones y finalmente se ha dispuesto de un submenú en el cual han sido ubicados los controles elaborados para la presente aplicación. En la figura 6.2 se muestra la estructura principal del proyecto.



Fig. 6. 2: Estructura plataforma web empleando HTML

Elaborado por: El investigador

En la figura se puede observar la estructura principal de la página web establecida mediante código HTML, que como se observa es una estructura muy básica pero es lo único que se puede

obtener a través del lenguaje HTML. Mismo que como hemos descrito en el capítulo IV sub apartado 4.1.2 permite establecer la estructura básica de una página web.

Se ha diseñado una estructura simple utilizando únicamente ciertas etiquetas del lenguaje HTML para crearla. Para establecer el nombre que aparecerá en la pestaña del navegador que identificará a nuestra página web se ha utilizado la etiqueta `<title>`, posteriormente se utilizó la etiqueta `` y `<h1>` para definir el nombre que se desplegara en la página web. Mientras que para la barra de navegación se utilizaron las etiquetas `<nav>` y `` que permite crear listas de elementos. En la figura 6.3 se puede apreciar de mejor manera esta descripción.

```
<!DOCTYPE html>
<html lang="en" class="no-js">
  <head>
    <meta charset="utf-8">
    <title>ANC Web HMI Platform</title>
  </head>
  <body>
    <div class="container1" >
      <!-- ECABEZADO BARRA AZUL DE MENU PRINCIPAL -->
      <header class="clearfix" id="idclearfix">
        <span>ANC Platform</span>
        <h1>HMI Web Server</h1>
      </header>
    </div>
    <!-- MENU PRINCIPAL -->
    <div id="mySidenav" class="main">
      <nav id="menu" class="menuprincipal" >
        <ul>
          <!-- SUB MENU INDICADORES-->
          <li id="horizontal"><a href="#">CONTROLES</a>
            <div id="scroll" class="submenu">
              <ul class="controles" >
                <!-- GAUGE -->
                <li><a href="#">GUAGE</a>
                  <!-- CONTENEDOR INDICADOR 1-->
                </li>
                <!-- LED -->
                <li><a href="#">LED</a>
                </li>
                <!-- LABEL -->
                <li><a href="#">LABEL</a>
                </li>
                <!-- BOTON 1 -->
                <li><a href="#">BOTTON1</a>
                  <!-- CONTENEDOR CONTROL 1-->
                </li>
                <!-- GAUGE 1-->
                <li><a href="#">GAUGE 1</a>
              </ul>
            </div>
          </li>
        </ul>
      </nav>
    </div>
  </body>
</html>
```

Fig. 6. 3. Descripción código HTML

Elaborado por: El investigador

6.5.6 Diseño de la apariencia de la plataforma mediante CSS

Una vez establecida la estructura principal es necesario otorgarle un aspecto atractivo para el usuario. Lo cual se lo ha hecho a través del lenguaje CSS. En la figura 6.4 se puede observar un antes y un después de programar la apariencia de la página web.

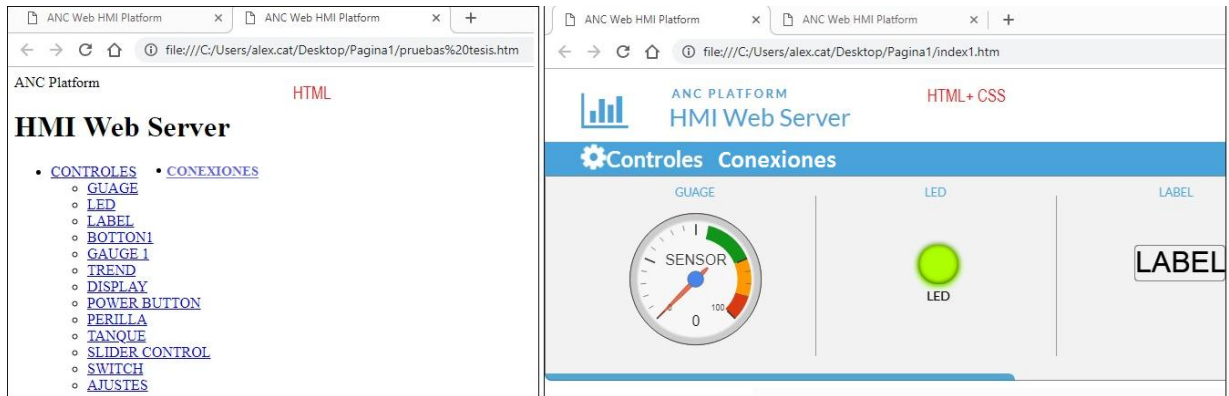


Fig. 6. 4. Aplicación de estilo visual CSS.

Elaborado por: El investigador

En el **Anexo 2** podemos encontrar el código de los estilos principales de la apariencia de nuestra página web con sus respectivas descripciones.

6.5.7 Diseño de los elementos virtuales

El proceso de creación de los instrumentos virtuales contenidos en la plataforma web consiste en la integración de todas las técnicas y lenguajes de programación web estudiados previamente en el capítulo 4. Además de la utilización de librerías y plugins que facilitan el proceso de visualización e intercambio de información. Siendo un proceso similar para la mayor parte controles y diferenciándose solamente en la función que realiza cada uno de ellos, así como de la librería o plugin que es necesario para su creación. A continuación se describirá de manera detallada la creación del primer control denominado **gauge** y enfatizaremos las partes primordiales de los demás ya que su código lo podemos encontrar de manera detallada en los **Anexos 1, 2 y 3** respectivamente.

6.5.7.1 Creación del control Gauge.

Para la creación de este instrumento virtual se ha utilizado la librería **jqWidgets** y el paquete **jqxGauge** perteneciente a la familia de gauges de esta librería. Esta librería nos permitirá hacer uso de este control modificarlo e intercambiar información a través del mismo.

Para describir el proceso de creación del control es necesario seguir una serie de pasos.

- 1) Enlazar las dependencias que permiten hacer uso de la librería para la utilización del paquete gauge. Este proceso se lo deberá hacer en el archivo principal, es decir en archivo de código HTML. Tal como se puede apreciar en la figura 6.5

```
<!-- LIBRERIA GAUGE 1 -->
<script src="js/jqxc core.js"></script>
<script src="js/jqxdata.js"></script>
<script src="js/jqxdraw.js"></script>
<script src="js/jqxchart.core.js"></script>
<script src="js/demos.js"></script>
<script src="js/jqxgauge.js"></script>
```

Fig. 6. 5. Adición de las dependencias para la utilización de la librería jqxgauge

Elaborado por: El investigador

- 2) Crear un nuevo archivo de extensión **.js** al cual se lo ha denominado **gauge.js**. Dentro de este archivo procederemos a escribir el código para crear el presente control.
- 3) Para poder generar este control es necesario establecer ciertos parámetros bases que son los que le darán la estructura y forma al control tales como los rangos, las leyendas, los colores del estilo visual, el tamaño del control etc. Lo cual se lo hace siguiendo la sintaxis propia del plugin a utilizarse la cual se describe a continuación:

`$(“Id de contener del control”).jqxGauge({Parámetro del control a establecer});`

- 4) Posteriormente se procede a crear la función que escucha el evento “valueChanging” es cual nos permite desplegar el valor del gauge en su cuadro de información numérica. Lo cual se realiza a través de las siguientes líneas de código.

```
$(“Id del contenedor del control”).on('valueChanging', function (e) {
    $(“Id de contenedor numerico del control”).text(e.args.value);
});
```

5) Finalmente se procede a llamar a la función que se encarga de generar el control enviándole a la misma como parametro el número de control que se está ejecutando en ese momento el cual por defecto se ha designado con el valor de cero al iniciar la plataforma. A continuación se puede apreciar el código de este control en la imagen 6.6.

```
//CODIGO DEL CONTROL
function gauge(iic){
    $('#gauge'+iic).jqxGauge({ min: iimin });
    $('#gauge'+iic).jqxGauge({ max: iimax });
    $('#gauge'+iic).jqxGauge({
        ranges: [
            { startvalue: iimin, endvalue: 0.25*iimax, style: { fill: '#4bb648', stroke: '#4bb648', endwidth: 5, startwidth: 1 },
            { startvalue: 0.25*iimax, endvalue: 0.5*iimax, style: { fill: '#fbd109', stroke: '#fbd109', endwidth: 10, startwidth: 5 },
            { startvalue: 0.5*iimax, endvalue: 0.75*iimax, style: { fill: '#ff8000', stroke: '#ff8000', endwidth: 13, startwidth: 10 },
            { startvalue: 0.75*iimax, endvalue: iimax, style: { fill: '#e02629', stroke: '#e02629', endwidth: 16, startwidth: 13 }
        ]},
        labels: [
            {
                distance: '1px',
                position: 'inside',
                interval: 0.1*iimax,
                offset: [0, -10],
                visible: true,
                formatvalue: function (value) {
                    return value;
                }
            }
        ],
        ticksMinor: { interval: 0.5*0.1*iimax, size: '5%' },
        ticksMajor: { interval: 0.1*iimax, size: '10%' },
        colorsScheme: 'scheme01',
        animationDuration: 0,
    });
    $('#gauge'+iic).jqxGauge({ height: iisize });
    $('#gauge'+iic).jqxGauge({ width: iisize });
    $('#gauge'+iic).on('valueChanging', function (e) {
        $('#gaugeValue').text(Math.round(e.args.value));
    });
}
```

Fig. 6. 6: Código del control gauge

Elaborado por: El investigador

En la figura 6.7 a continuación se muestra el resultado final del código descrito anteriormente.

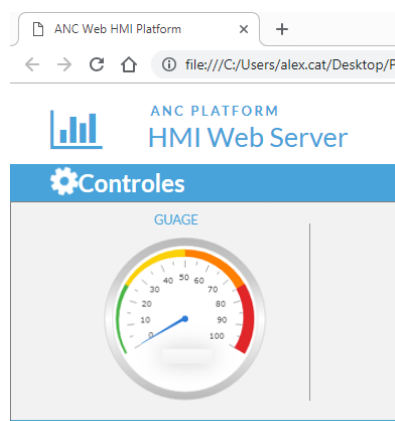


Fig. 6. 7: Control Gauge

Elaborado por: El investigador

6.5.7.2 Creación de contenedores y cuadro de herramientas para los controles.

Es necesario crear una estructura que nos permita establecer tanto el contenedor del control así como el cuadro de herramientas del mismo. Ya que es necesario disponer de controles que otorguen la posibilidad de cambiar las propiedades del control tales como: posición, tamaño, variables a leer, color, escalas etc. Para lo cual se ha llevado a cabo los siguientes pasos:

- 1) Crear un contenedor dentro del submenú **GAUGE**, mismo que se encuentra en el archivo de estructura **HTML** denominado **controlboxi1**. Este contendor permitira establecer la posición en la que se encontrará nuestro control mediante la asignación de cordoneras en la pantalla a través del lenguaje CSS. La figura 6.8 se muestra este proceso.

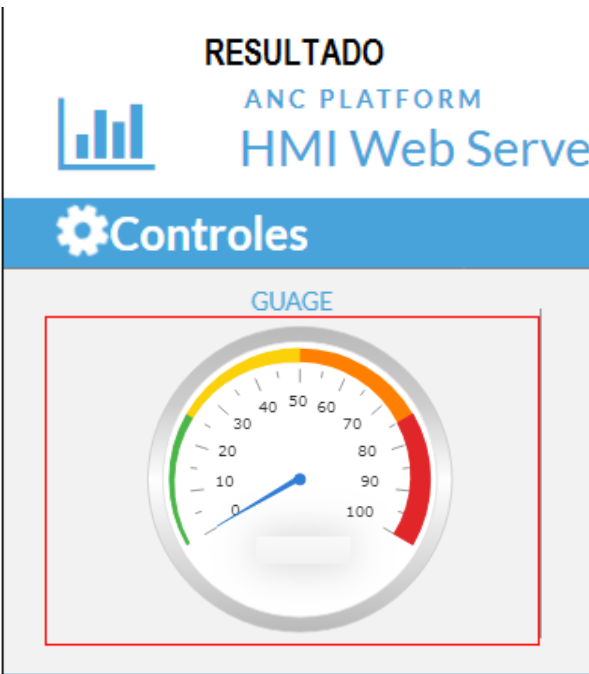
CÓDIGO CSS	RESULTADO
<pre>#controlboxi1{ border: 1px solid red; position: relative; width: 330px; height: 220px; } .indicador0{ position: absolute; width: 250px; height: 250px; left: 20%; top: 5%; z-index: -10; }</pre>	
CÓDIGO HTML	
<pre><!-- GAUGE --> GAUGE <!--CONTENEDOR INDICADOR 1--> <div id="controlboxi1"> <!--COPIA 0 INDICADOR 1--> <div id="indicador0" class="indicador"> <!--COPIA 1 INDICADOR 1--> <div id="indicador1" onclick="addi()" class="indicador"> <div id="propiedadesi1" class="propiedades" > </pre>	

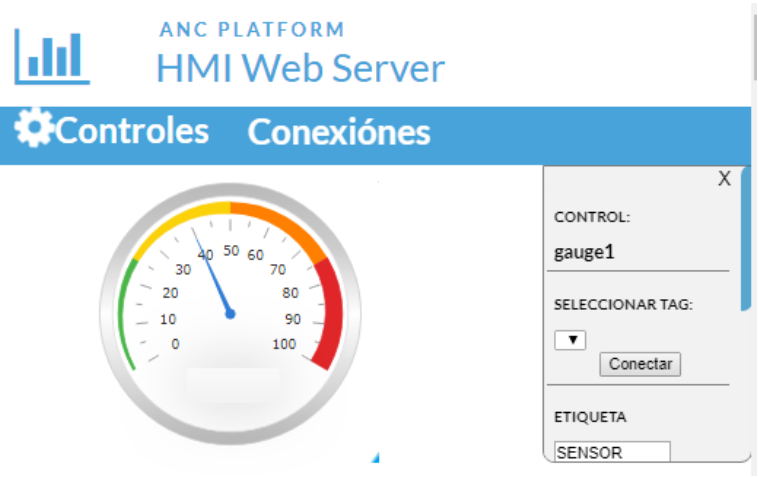
Fig. 6. 8: Creación del contenedor del control y posicionamiento

Elaborado por: El investigador

Como se puede apreciar en la figura anterior se procede a crear las estructuras del contenedor y también del control mediante etiquetas **div**. A las cuales es necesario establecerles los atributos de **id** y **class**. Debido a que son varios los controles que se han creado y a los cuales fue necesario dotarles de una apariencia atractiva al usuario, a través de código CSS. La solución más lógica fue identificar las características comunes a través de un nombre, mediante el atributo **class** con la finalidad de que se escriba un solo archivo de código css para todas ellas en lugar de crear archivos individuales. Por otra parte solamente es necesario disponer de un solo contenedor para nuestro control. El cual poseerá propiedades que serán únicas del control como la posición. Para ello es necesario la utilización del atributo **Id**. El cual identifica de manera única a un elemento de código HTML. Es decir que no puede existir otro elemento con el mismo **id**. Como se puede apreciar en la figura 6.9 el identificador del control se lo ha nombrado **controlbox1**.

A través del nombre los atributos antes mencionados es posible hacer referencia a los objetos a los cuales se desea cambiar sus propiedades mediante código CSS. Como se puede visualizar en la figura 6.9, se utiliza varias palabras clave para definir el ancho, alto o posición de la imagen tales como: **width, height, left** o **top** respectivamente. La integración de todas estas piezas de código nos da como resultado nuestro control al cual lo podemos visualizar en la parte superior derecha de la figura 6.8.

- 2) Posteriormente es necesario crear el contenedor del cuadro de propiedades para del control desarrollado. Proceso que se realiza de misma manera que se creó el contenedor del control a través de las etiquetas **div**. En la figura 6.9 se muestra el código para crear el cuadro de propiedades del control y su resultado.



CÓDIGO HTML

```

.v id="controlbox11">
<!--COPIA 0 INDICADOR 1-->
<div id="indicador0" class="indicador"></div>
<!--COPIA 1 INDICADOR 1-->
<div id="indicador1" onclick="addi1()" class="indicador"></div>
<div id="propiedades11" class="propiedades" >
<ul>
<!--ENLAZAR TAG-->
<li>
<ul>
<a href="#">ENLAZAR TAG</a>
<!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->
<span class="xicon" onclick="cerrari1()">X</span>
<input type="file" name="file-1[]" id="file-1" class="inputfile inputfile-1" onchange="etiquetai1(this)">
<label for="file-1"><span>Enlazar Tag</span></label>
<br>
<label id="labeltag1" class="labeltag">nombre</label>
</li>
<!--NOMBRE DEL CONTROL-->
<li> <a href="#">ETIQUETA</a>
<input class="etiqueta" id="etiquetai1" type="text" value="SENSOR" >
</li>
<!--INTRODUCIR RANGO DEL CONTROL-->
<li> <a href="#">RANGO</a>
<label>Min :</label>
<input class="rango" id="mini1" type="text" value="0" >
<br>
<label>Max:</label>
<input class="rango" id="maxi1" type="text" value="100" >
</li>
<!--MODIFICAR TAMAÑO DEL CONTROL-->
<li> <a href="#">TAMAÑO</a>
<input type="range" min="100" max="500" value="200" class="slider" id="slideri1" oninput="changesizei1()">
<label id="sizevali1" class="sizeval">:200</label>
</li>
<!--REMOVER CONTROL-->
<li><a href="#">REMOVER CONTROL</a>
<button type="button" id="removeri1" class="remover" onclick="removei1()">Remover</button>
</li>
</ul>
</div>

```

Creación de un símbolo para ocultar el cuadro de propiedades (x)
 Creación de boton para seleccionar la variable a monitorear
 Creación de un label para visualizar el valor de la variable monitoread
 Creación de la etiqueta input de tipo text que permite cambiar el nombre del control
 Creación de los rangos del control a modificarse a través de las etiquetas input de tipo text
 Creación de un slider mediante la etiqueta input de tipo range. Para modificar el tamaño del control
 Creación de un boton mediante la etiqueta button para remover el control del área de trabajo

Fig. 6. 9: Creación del cuadro de propiedades del control

Elaborado por: El investigador

El código del archivo de estilos para el cuadro de propiedades se lo puede apreciar de mejor manera en el **Anexo 2**.

6.5.7.3 Programación de las subrutinas del control

Como se ha descrito anteriormente, el objetivo del presente proyecto de investigación es crear una plataforma web que permita interactuar en tiempo real con los datos provenientes del controlador. Por lo cual es necesario que se pueda otorgarle dinamismo al control que se ha creado. Para lo cual se ha programado varias subrutinas que nos permitirán tener acceso a las propiedades del control y poder interactuar con las mismas. En la mayor parte de controles las subrutinas que describiremos a continuación son muy similares variando simplemente en la función que realiza el control. Todas estas subrutinas las podemos encontrar de una manera detallada en el **Anexo 3**.

- **Descripción de la subrutina para la enlazar la variable que se monitoreara**

Esta subrutina es la más importante de todas y representa el corazón de la plataforma web desarrollada, ya que en ella se encuentra el proceso de lectura de la información proveniente del controlador. En la figura 6.10 a continuación se muestra el código de esta subrutina la cual se describirá posteriormente.

```
var socket = io.connect('http://localhost:3700'); ①
function tagvalue1(){ ②
    var indice=actualgauge.substring(5,actualgauge.length);
    indicevar=indice;
    //OBTENER VALOR DE LA VARIABLE A CONECTAR ③
    document.getElementById("labeltag1").innerHTML=actualgauge;

    conexiones[0]=tagseleccionada.substring(0,tagseleccionada.search(",")); ④
    conexiones[1]=indicevar;

    //ENVIAR LA VARIABLE QUE VAMOS A MONITOREAR AL CLIENTE ⑤
    socket.emit('send', conexiones);

    //RECIBIR LA VARIABLE DEL CLIENTE OPCUA ⑥
    socket.on('valvariable', function (data) {
        console.log(data.value.length);

        console.log(data.value[parseInt(indicevar)-1]);
        console.log(indicevar);
        datavalue[parseInt(indicevar)]=data.value[parseInt(indicevar)-1];

        var tt=0;
        for(tt=0;tt<data.value.length;tt++){
            var valgauge=data.value[tt];
            document.getElementById('gaugeval'+(tt+1)).innerHTML=data.value[tt];
            $('#gauge'+(tt+1)).jqxGauge('value', Number(data.value[parseInt(tt)])); ⑦
        }
    });
}
```

Fig. 6. 10: Código de la sub rutina que establece la variable a monitorear

Elaborado por: El investigador

En el primer punto del código se puede observar que la declaración de una variable a la cual se la ha denominado `socket`. Esta variable permitirá hacer uso de la librería **socket.io**, misma que permitirá establecer la comunicación entre la página web y el cliente del protocolo que utilizaremos en este caso el OPC UA.

En el segundo punto se ha definido una función que se encargará de escuchar el evento `click` el cual se ejecuta cuando el botón `conectar` del control es presionado. Cuando el evento es detectado se obtiene el índice del botón y se lo almacena en la variable **indicevar** como se puede observar en el paso 3. Posteriormente se almacena el `id` y el índice de la variable a la cual deseamos conectarnos dentro del vector `conexiones`.

Cuando el vector `conexiones` posea toda la información lista se ejecuta el evento **socket.emit** el cual se encarga de enviar la información del nodo a monitorear hacia el cliente OPC UA, lo cual se lo hace en el paso número 5. Posteriormente se recibe la información de la variable a la cual nos hemos suscrito para monitorizar a través de la función **socket.on** como podemos observar en el paso número 6. Finalmente asignamos los valores obtenidos a nuestro control haciendo uso de la propiedad **value** del plugin tal como podemos observar en el paso número 7.

- **Descripción de la subrutina que añade el control al área de trabajo**

Esta subrutina permite interactuar con los controles y añadirlos al área de trabajo para poder diseñar nuestras pantallas de monitorización. En la figura 6.11 se mostrará el código creado para realizar este proceso.

```

//AÑADIR UN NUEVO CONTROL AL AREA DE TRABAJO
function addi1(){
    i1c++;
    congauge[i1c] = document.createElement("div");
    congauge[i1c].id="gauge"+i1c;

    gaugeval[i1c] = document.createElement("div");
    gaugeval[i1c].id="gaugeval"+i1c;
    gaugeval[i1c].className="c1gauge1";

    $("#mySidenav").append(congauge[i1c]);
    $("#mySidenav").append(gaugeval[i1c]);
    $("#propiedadesi1").appendTo("#mySidenav");

    //DEFINIR LAS COORDENADAS DE LA PANTALLA DONDE SE UBICARA EL CONTROL
    //DEFINIR EL TAMAÑO DEL CONTROL A AÑADIRSE
    document.getElementById(congauge[i1c].id).style.width=300+"px";
    document.getElementById(congauge[i1c].id).style.height=300+"px";
    document.getElementById(congauge[i1c].id).style.position="absolute";
    document.getElementById(congauge[i1c].id).style.left=200+"px";
    document.getElementById(congauge[i1c].id).style.top=300+"px";

    document.getElementById(congauge[i1c].id).style.border="10px solid transparent";
    document.getElementById(congauge[i1c].id).style.borderImage="url(ima1.png) round";
    document.getElementById(congauge[i1c].id).style.borderImageSlice="55%";

    document.getElementById(gaugeval[i1c].id).style.width=80+"px";
    document.getElementById(gaugeval[i1c].id).style.height=30+"px";
    document.getElementById(gaugeval[i1c].id).style.textAlign="center";

    var lp=document.getElementById(congauge[i1c].id).offsetLeft;
    var tp=document.getElementById(congauge[i1c].id).offsetTop;

    document.getElementById(gaugeval[i1c].id).style.left=(lp+0.28*300)+"px";
    document.getElementById(gaugeval[i1c].id).style.top=(tp+0.55*300)+"px";

    //UBICACIÓN DEL CUADRO DE PROPIEDADES
    document.getElementById("propiedadesi1").style.top=(0)+"px";
    document.getElementById("propiedadesi1").style.left=(mywidth-200)+"px";
    document.getElementById("propiedadesi1").style.display="block";

    gauge(i1c);
    $('#gauge'+i1c).jqxGauge('value', 40);
}

```

Fig. 6. 11: Código de la sub rutina que agrega un nuevo control al área de trabajo

Elaborado por: El investigador

El código para añadir un nuevo control se ejecuta cuando el evento doble click sobre el elemento es ejecutado. Una vez que se ejecuta este evento se procede a crear nuevas etiquetas html a través del comando **createElement**. Estas etiquetas permitirán tener tantos elementos como deseamos y que se almacene un id diferente para cada etiqueta creada como se puede apreciar en el paso número 1.

Posteriormente se agrega el control al área de trabajo a través del comando **append** paso número 2. Una vez que el control se haya agregado debemos de establecer sus

parámetros como la posición y tamaño del control. Lo cual se lo realiza a través de la manipulación se sus respectivas características de estilo como **width**, **height** o **top**, descritas en el paso número 3. Adicionalmente se deben definir las características de tamaño y posición del cuadro de propiedades del control como se puede apreciar en el paso número 4.

Finalmente se llama a la función encargada de crear la estructura principal del control a la cual se le envía como parámetro el índice del nuevo control. Esto permitirá que se pueda crear un control nuevo con un único Id que lo identifique como podemos observarlo en el paso número 5.

- **Descripción de la subrutina que permite cambiar el tamaño del control.**

Otra característica muy importante a tomar en cuenta es el tamaño del control. Es necesario tener la posibilidad de poder acceder a esta propiedad y poder modificarla con el objetivo de crear pantallas de un buen diseño y estéticamente aceptables. A continuación se muestra el código que hace posible cambiar el tamaño del control en la figura 6.12.

```
//CAMBIAR TAMAÑO DEL CONTROL
function changesizei1() {
    var indice=actualgauge.substring(5,actualgauge.length);
    document.getElementById("labeltagi1").innerHTML=indice+"ind";

    //ALMACENAR EL VALOR DEL SLIDER EN LA VARIABLE S1
    var s1=document.getElementById("slideri1").value;
    //VISUALIZAR EL VALOR DEL SLIDER EN LA ETIQUETA DEL CUADRO DE PROPIEDADES
    document.getElementById("sizevali1").innerHTML=":"+s1;
    //SELECCIONAR EL CONTROL A TRAVES DE SU ID Y CAMBIAR SU TAMAÑO
    $('#gauge'+indice).jqxGauge({ height: s1 });
    $('#gauge'+indice).jqxGauge({ width: s1 });
    var cw = $('#gauge'+indice).jqxGauge('width');
    var ch = $('#gauge'+indice).jqxGauge('height');

    var lp=document.getElementById("gauge"+indice).offsetLeft;
    var tp=document.getElementById("gauge"+indice).offsetTop;

    var tcv=parseInt((45/170)*s1);

    document.getElementById("gaugeval"+indice).style.width=tcv+"px";
    document.getElementById("gaugeval"+indice).style.top=(ch*0.74+tp)+"px";
    document.getElementById("gaugeval"+indice).style.left=(0.4*(cw)+lp)+"px";
}
```

Fig. 6. 12: Código de la subrutina para cambiar el tamaño del control

Elaborado por: El investigador

Como primer paso para poder modificar cualquier característica de un control es necesario determinar cuál es el control al cual se dirigirá dicha acción. Para ello se debe

obtener el índice del control como se puede observar en el paso número uno. Esto permitirá identificar a cual control se cambiara su tamaño. Posteriormente se procede a obtener el valor número del slider que nos permitirá tener un valor para asignarle al tamaño del control como podemos observar en el paso número 2. Finalmente hacemos uso de las propiedades **height** y **width** del control para establecer este valor paso número 3.

- **Descripción de la subrutina que permite cerrar el cuadro de propiedades del control.**

Aunque el cuadro de propiedades nos da la posibilidad de interactuar con el control y cambiar sus propiedades no es estético que siempre este visible por lo cual se ha diseñado una subrutina que permita ocultarlo cuando no sea necesario.

Para ocultar el cuadro de propiedades es un proceso muy simple, ya que únicamente es necesario hacer uso de la propiedad **display** y setearla en **none**. En la figura 6.13 se muestra el código de esta subrutina y su respectivo resultado.

```
//OCULTAR CUADRO DE CONTROLES
function cerrari1() {
    document.getElementById("propiedades1").style.display="none";
}
```

Fig. 6. 13: Subrutina para cerrar el cuadro de propiedades

Elaborado por: El investigador

- **Descripción de la subrutina que permite remover el control del área de trabajo**

Al igual que es necesario añadir un control al área de trabajo también es necesario removerlo. Para ello simplemente se debe hacer uso de la instrucción `remove` tal como se puede observar en la figura 6.14 a continuación.

```
//REMOVER CONTROL
function removei1() {
    var indice=actualgauge.substring(5,actualgauge.length);
    $("#gauge"+indice).remove();
    $("#gaugeval"+indice).remove();
}
```

Fig. 6. 14: Código de la sub rutina remover control

Elaborado por: El investigador

- **Descripción de la subrutina para desplazar el control por el área de trabajo.**

Para poder desplazar el control por el área de trabajo es necesario hacer uso de la instrucción `draggable` perteneciente a jQuery. A través de esta instrucción se puede realizar este proceso y establecer ciertos parámetros que permitirán que el control se mueva solamente en el área designada tal y como se observa en el código de la figura 6.15.

```
$("#"+event.target.id).draggable({
    containment: '#mySidenav',
    mouse: 'move',
    cursor: 'move',
    drag: function (event, ui) {
        var cw = $("#"+event.target.id).jqxGauge('width');
        var ch = $("#"+event.target.id).jqxGauge('height');

        var lp=document.getElementById(event.target.id).offsetLeft;
        var tp=document.getElementById(event.target.id).offsetTop;

        document.getElementById("gaugeval"+event.target.id.substring(5,event.target.id.length)).style.left=((0.37*cw)+lp)+"px";
        document.getElementById("gaugeval"+event.target.id.substring(5,event.target.id.length)).style.top=(ch*0.74+tp)+"px";
    },
});
```

Fig. 6. 15: Código de la sub rutina para desplazar el control

Elaborado por: El investigador

6.5.8 Diseño de la comunicación con el protocolo OPC UA.

Para establecer la comunicación entre la página web y el servidor se ha utilizado el protocolo OPC UA mismo que permite realizar la monitorización de las variables del PLC. Para ello fue necesaria la creación de dos archivos de código principales uno para el servidor OPC UA y el otro para el cliente OPC UA mismos que se describirán a continuación.

6.5.8.1 Cliente OPC UA

Para el cliente OPC AU se ha escrito un archivo de código el cual nos permitirá monitorizar de las variables que se han designado a través de un nodo Id. A continuación se presenta un diagrama de flujo de su funcionamiento en la figura 6.16 y posteriormente se describirá en detalle el código del mismo.

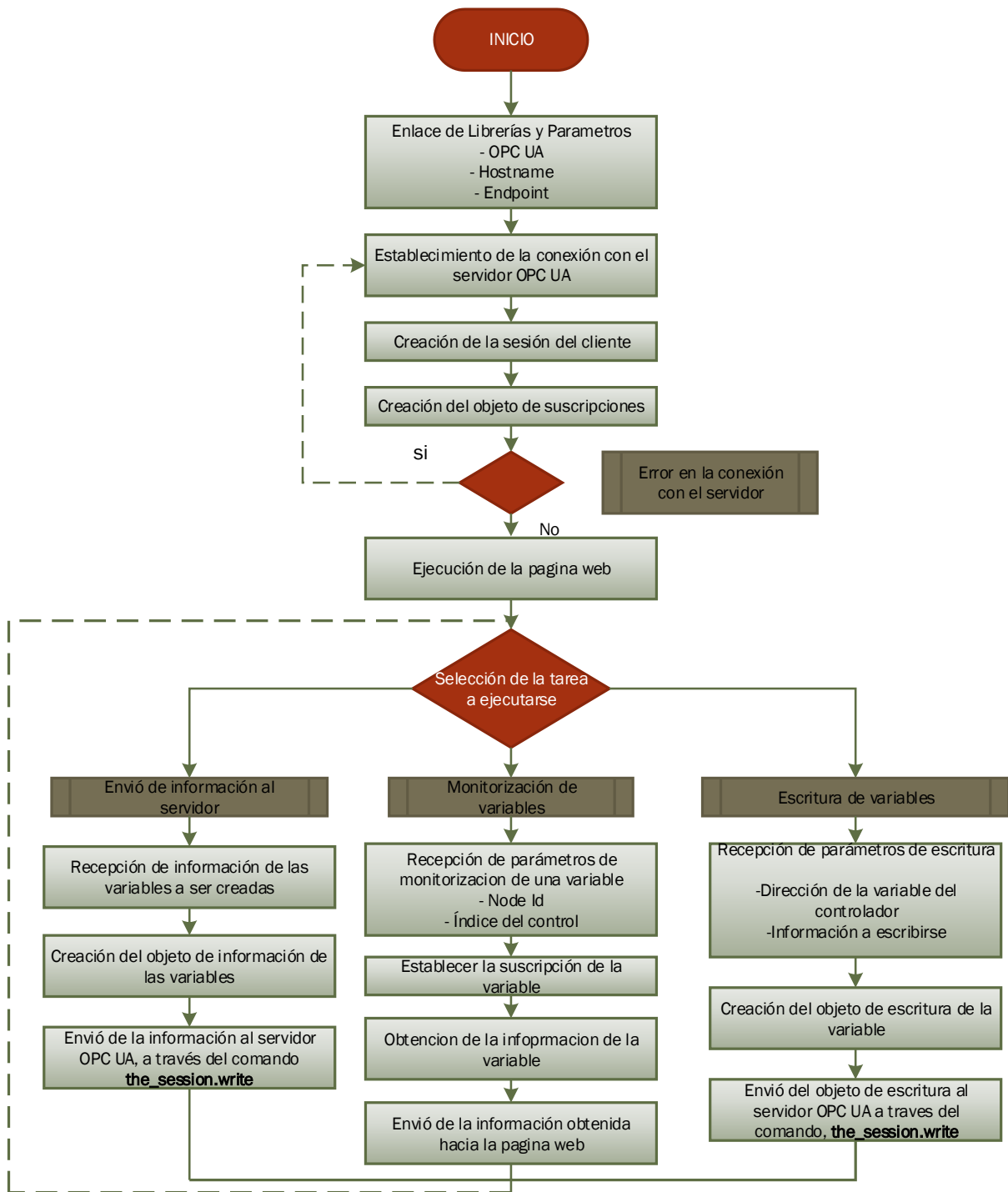


Fig. 6. 16: Diagrama de flujo de funcionamiento del Cliente OPC UA

Elaborado por: El investigador

Primeramente es necesario declarar una variable que nos permita enlazar nuestro protocolo. Este procedimiento se realiza a través de comando **require**. Posteriormente es necesario crear un nuevo objeto del cliente OPC UA, el cual se ha almacenado en la variable **client**. Variable mediante la cual podremos hacer uso de todos los métodos y clases pertenecientes al cliente OPC UA. Por último es necesario establecer los parámetros necesarios para establecer la conexión con el servidor por lo cual debemos crear un endpoint, mismo que define la dirección donde se encuentra alojado el servidor OPC UA. En la figura 6.17 se puede observar estos procedimientos.

```
//ENLAZAR EL PROTOCOLO OPC UA EN LA VARIABLE opcua
var opcua = require("node-opcua");
var async = require("async");
//CREAR UN NUEVO OBJETO PARA EL CLIENTE OPC UA
var client = new opcua.OPCUAClient({

});
//DEFINIR LOS PARAMETROS PARA LA CONEXION CON EL SERVIDOR
var hostname = require("os").hostname();
var endpointUrl = "opc.tcp://" + "ALEX" + ":26543";
```

Fig. 6. 17: Establecimientos de parámetros del protocolo OPC UA

Elaborado por: El investigador

Luego de establecer los parámetros principales procedemos a establecer la comunicación con el servidor OPC UA. Para llevar a cabo este proceso se han creado cuatro pasos importantes. En el primer paso se procede a establecer la conexión con el servidor a través del comando **client.connect**, a este comando es necesario enviarle como parámetro el endpoint que se le ha asignado al servidor. Posteriormente se debe crear una nueva sesión para el cliente que hemos designado. Lo cual se lo hace a través del comando **createSession**. Como tercer paso se debe crear un nuevo objeto para establecer una suscripción con el servidor OPC UA a través de la cual seremos capaces de monitorizar las variables que se encuentran en el servidor. Si todos estos procesos se han completado satisfactoriamente se establecerá una conexión exitosa con el servidor y se procederá a ejecutar la función que desplegara la página web.

El código perteneciente a estos procesos descritos lo podemos apreciar en la imagen 6.18 que se muestra a continuación.

```

async.series([
  // 1) INICIAR CONEXION CON EL SERVIDOR OPC UA
  function(callback) {
    console.log(" connecting to ", endpointUrl.cyan.bold);
    client.connect(endpointUrl,callback);
  },

  // 2) CREAR LA SESION DEL CLIENTE
  function(callback) {
    client.createSession(userIdentity,function (err,session){
      if (!err) {
        the_session = session;
        console.log(" session created".yellow);
      }
      callback(err);
    });
  },

  // 3)CREAR LA SUSCRIPCION PARA MONITOREAR LAS VARIABLES DEL SERVIDOR
  function(callback) {
    the_subscription=new opcua.ClientSubscription(the_session,{
      requestedPublishingInterval: 1000,
      requestedMaxKeepAliveCount: 10,
      requestedLifetimeCount: 2,
      maxNotificationsPerPublish: 10,
      publishingEnabled: true,
      priority: 10
    });

    the_subscription.on("started",function(){
      console.log("subscription started");
      callback();
    }).on("keepalive",function(){
      console.log("keepalive");
    });
  },function(err) {
    if (!err) {
      // 4)INICIAR LA PAGINA WEB
      startHTTPServer();
    } else {
      // cannot connect to client
      console.log(err);
    }
  }
]);

```

Fig. 6. 18: Código que permite establecer la conexión con el servidor OPC UA

Elaborado por: El investigador

Luego de establecer la conexión se han creado cuatro partes importantes de código, mismas que permitirán enviar los datos de la variable que se desea monitorizar al servidor OPC UA, leer las variables que se encuentran en el servidor y enviarlas a la página web, obtener la información de la variable monitorizada, y por ultimo escribir los valores de los controles de la página en las variables del servidor.

Para poder enviar los datos de las variables que deseamos leer se utiliza el evento **socket.on**, el cual nos permitirá escuchar la información que se envía desde la página web. Cuando esta información se adquiere se almacena en la variable **globalwrite**. Posteriormente es necesario crear un objeto que se enviara al servidor OPC UA. Este objeto poseerá el Id del nodo al cual se desea monitorizar. Esto se lo hace con el objetivo de que el servidor pueda crear una nueva variable que permita recupera la información del controlador. Finalmente se envía el objeto creado al servidor empleando el comando **the_session.write**. Esta parte del código se la puede observar en la imagen 6.19.

```
//EVENTO QUE ENVIA EL NOMBRE DE LAS VARIABLES PARA QUE SEAN ANADIDAS AL SERVIDOR
socket.on('crearvariable', function(data){
    console.log(data);
    var globalwrite = data;
    var nodesToWrite = [{
        nodeId: "ns=1;s=Variables",
        attributeId: 13,
        indexRange: null,
        value: {
            value: {
                dataType: opcua.DataType.String,
                value: globalwrite
            }
        }
    }];
    the_session.write(nodesToWrite, function(err,statusCode,diagnosticInfo) {
        if (!err) {
            console.log(" write ok" );
        }
    });
});
```

Fig. 6. 19: Código que envía la información de las variables al servidor OPC UA

Elaborado por: El investigador

Luego de que se han enviado la información de las variables a las cuales deseamos monitorizar el servidor devuelve la información de las mismas con su correspondiente Node Id. Para que puedan ser enviadas hacia los diferentes controles y puedan ser monitorizadas. Este proceso se lo hace a través del comando **the_subscription.monitor**. Dentro de este objeto se establecen parámetros tales como el nodeId encargado de enviar los nombres de las variables desde servidor así como el tiempo de muestreo de las mismas. En la figura 6.20 podemos observar el código de este proceso.

```

//+++++LEER LOS NOMBRES DE LAS VARIABLES DEL SERVIDOR+++++
//ESTE ENVENTO ENVIARA LOS IDS DE LAS VARIABLES CONTENIDOS EN EL SERVIDOR OPC UA
var monitoredItem = the_subscription.monitor(
{
  nodeId: nodeIdToMonitor,
  attributeId: 13
},
{
  samplingInterval: 2000,
  discardOldest: true,
  queueSize: 5
},opcua.read_service.TimestampsToReturn.Both,function(err) {
if (err) {
  console.log("Monitor "+ nodeIdToMonitor.toString() + " failed");
  console.log("ERR = ",err.message);
}
});

var valsend;
monitoredItem.on("changed", function(dataValue){
  valsend=dataValue.value.value;
  console.log("VARIABLES DEL SERVIDOR:= "+valsend);

  io.sockets.emit('message', {
    value: dataValue.value.value,
  });
});

```

Fig. 6. 20: Código que nos permite obtener las variables contenidas en el servidor OPC UA

Elaborado por: El investigador

Ahora se procederá a describir el código que nos permitirá monitorear una cierta variable y enviar su información hacia un determinado control. Esto se lo hace de igual manera a través del comando **the_subscription.monitor**. Además almacenaremos este objeto dentro del vector **tagmonitor** con el objetivo de poder monitorear diferentes variables a la vez y tener sus valores a disposición para que puedan ser desplegados en los diferentes controles. Una vez que se reciba la información proveniente del control que solicita determinamos el **nodeId** de la variable que se desea monitorizar y el índice del control en el cual se escribiera la información. Posteriormente se reciben los valores de dicha variable a través de evento **changed**. Lo cual permitira refrescar el valor de la variable cada vez que este cambie. Finalmente se almacena este valor y se lo envia hacia el control a través del evento **io.sockets.emit**. En la figura 6.21 se puede apreciar el código de este proceso.


```

//***** LECTURA DE VARIABLES *****
//*****MONITOREO DE VARIABLES PARA EL GAUGE INDICADOR*****
socket.on('send', function (data) {
    console.log("INDICE GAUGE1="+data[1]);
    console.log("VARIABLE GAUGE2="+data[0]);
    indice=parseInt(data[1]);

    tagmonitor[indice]= the_subscription.monitor(
    {
        nodeId: data[0],
        attributeId: 13
    },
    {
        samplingInterval: 1000,
        discardOldest: true,
        queueSize: 5
    },opcua.read_service.TimestampsToReturn.Both,function(err) {
    if (err) {
        console.log("Monitor "+ tagmonitor[indice]+ " failed");
        console.log("ERR = ",err.message);
    }
    });

    console.log(tagmonitor.length);

    var k1=1;

    for(k1=1;k1<tagmonitor.length;k1++){

        if((tagmonitor[k1]!==undefined))
        {
            tagmonitor[k1].on("changed", function(varValue){
                datadeopc[k1-2]=varValue.value.value;

                console.log("VALORES DE LOS GAUGE1="+datadeopc)
                //ENVIAR VALORES DE LAS VARIABLES AL GAUGE1
                io.sockets.emit('valvariable', {
                    value: datadeopc
                });
            });
        }else{
            datadeopc[k1-2]=0;
        }
    }

});

```

Fig. 6. 21: Código que permite monitorear una variable del servidor OPC UA.

Elaborado por: El investigador

De la misma manera que se adquiere el valor de las variables del servidor OPC UA, también se debe enviar los valores que se establezcan a través de un determinado control. Para lo cual se ha creado un objeto denominado **vartowrite**. Este objeto recibe como parámetros el node Id de la variable en la cual se escribiera la información así como la dirección de la variable del PLC donde se necesita escribir el valor. Esta información es consolidada en un array y posteriormente enviada al servidor a través de la instrucción **the_session.write** en la figura 6.22 podemos apreciar el código de este proceso.

```

//*****ESCRITURA DE VARIABLES*****
//*****LECTURA DE VARIABLES DEL LED*****
var enviodata=[];
socket.on('sendg2', function(data){

    console.log(data);
    console.log(data[0]);
    console.log(data[1]);
    enviodata[0]=data[0].toString();
    enviodata[1]=data[1].toString();

    vartowrite = [{
        nodeId: "ns=1;s=Escritura",
        attributeId: 13,
        indexRange: null,
        value: {
            value: {
                dataType: opcua.DataType.String,
                value: enviodata
            }
        }
    }];

    the_session.write(vartowrite, function(err,statusCode,diagnosticInfo) {
        if (!err) {
            console.log(" write ok" );
        }
    });
});
});

```

Fig. 6. 22: Código que escribe la información en una variable del servidor OPC UA.

Elaborado por: El investigador

6.5.8.2 Servidor OPC UA

A continuación se procederá a describir el proceso de funcionamiento del Servidor OPC UA, funcionamiento que se puede apreciar en el diagrama de flujo presentado en la figura 6.23 y posteriormente se describirá en detalle el código del mismo.

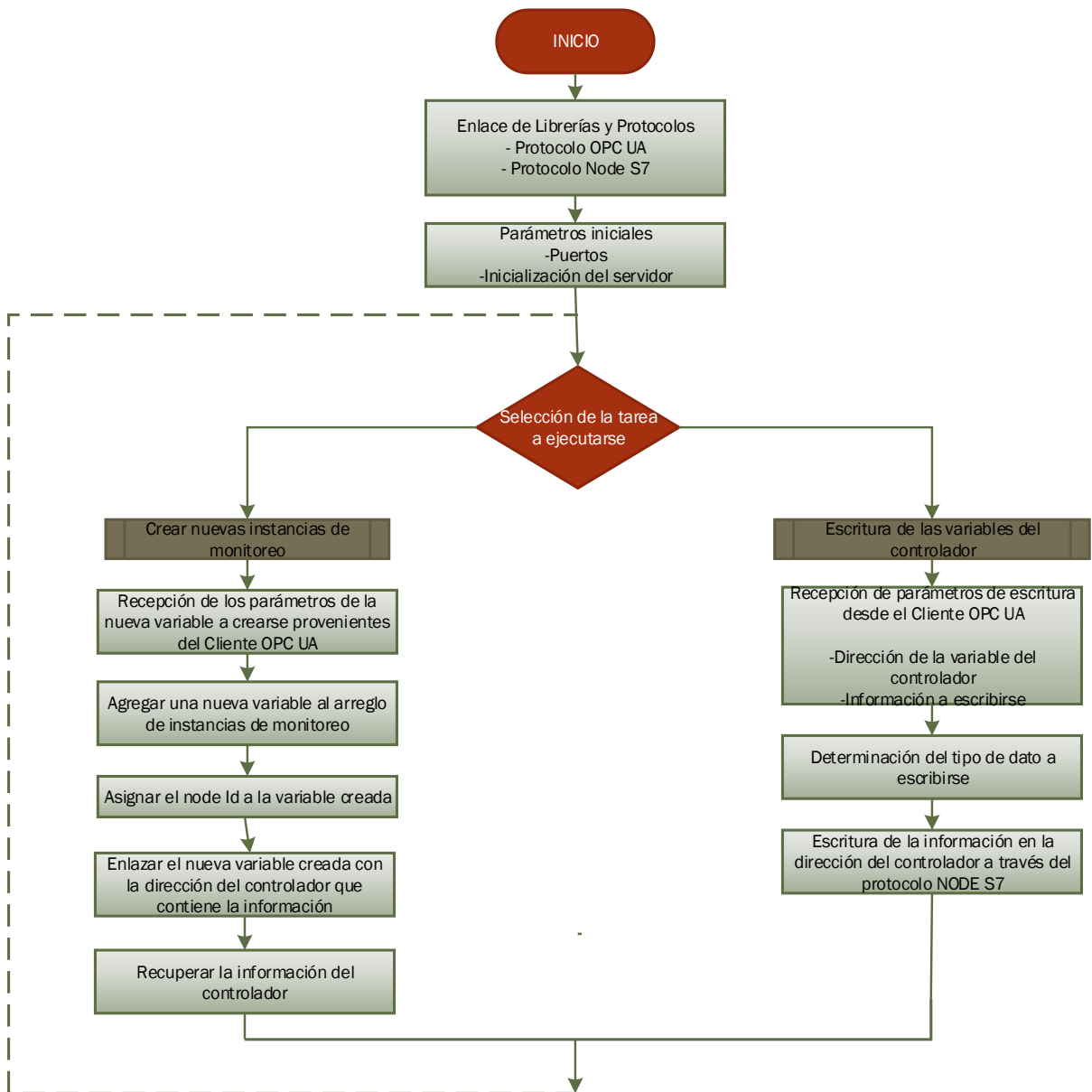


Fig. 6. 23: Diagrama de flujo de funcionamiento del Servidor OPC UA.

Elaborado por: El investigador

El servidor OPC UA será el encargado de leer la información de las variables del PLC. Para poder utilizar la instancia del servidor OPC UA al igual que en el cliente es necesario enlazar esta librerías a través del comando **require**. Posteriormente se debe crear una nueva instancia del servidor y establecer el puerto en el cual se escuchara las peticiones de los clientes. En la figura 6.24 podemos apreciar este código.

```

//ENLAZAR EL PROTOCOLO OPC UA Y LA LIBERIA NODES7
const opcua = require("node-opcua");
var nodes7 = require("nodes7");
// CREAR UNA NUEVA INSTANCIA DEL SERVIDOR OPCUA
const server = new opcua.OPCUAServer({
  port: 26543, // PUERTO DE ESCUCHA DEL SERVIDOR
  nodeset_filename: [
    opcua.standard_nodeset_file
  ]
});

```

Fig. 6. 24: Establecimiento de parámetros del servidor OPC UA.

Elaborado por: El investigador

Luego de haber establecido los parámetros de conexión se procede a crear dos piezas principales de código. La primera se encarga de recibir la información proveniente del cliente en el nodo “s=Variables”. Este nodo se encargara de recibir la información de la variables que se desear monitorizar y crear una instancia de esta con su respectivo id enviado por la página web. En la figura 6.25 se puede apreciar este código.

```

variables = namespace.addVariable({
  componentOf: node[0], //Referencia el a que nodo va a pertenecer la variable
  nodeId: "s=Variables",
  browseName: "Tags PLC0",
  dataType: "String",
  value: {
    get: function () {
      return new opcua.Variant({dataType: opcua.DataType.String, value: tagsvalue });
    },
    set: (variant)=> {
      tags = "ns=1"; "+s="+variant.value[0]; //CREACION DE LAS VARIABLES PARA MONITOREAR
      tagsvalue[0] = "ns=1"; "+s="+variant.value[0];
      tagsvalue[1] = variant.value[1];
      dataType[k]=variant.value[1]+"";
      console.log("*****");
      console.log(tags);
      console.log("VARIABLES TENTATIBAS"+tagsvalue);
      console.log(dataType[k]);

      //CREACION DE INSTANCIAS PARA MONITOREAR LAS VARIABLES DEL PLC
      tagcreated[k]= namespace.addVariable({
        componentOf: node[0], //Referencia al que pertenece la variable
        nodeId: "s="+variant.value[0], //Establece el id de la variable a monitorear
        browseName: variant.value[0], //Nombre de la variable que se desplegara en pantalla
        dataType: "Double", //Tipo de dato
        value: new opcua.Variant({dataType: opcua.DataType.Double, value: 1000.0})
      });
    }
  }
});

```

Fig. 6. 25: Código para crear las variables a ser monitoreadas.

Elaborado por: El investigador

En la segunda parte de código se recibe la información de los controles para poder escribir el valor de los mismos en las variables del PLC. Esta información llegara al nodo que se ha

designado como “s=Escritura”. Al obtener esta información se determina si es de tipo booleano o número y en base a esta decisión se escribe en el la dirección del PLC designada. A través del comando **conn.writeItems**. En la figura 6.26 se aprecia el código de este proceso.

```

//*****VARIABLE DE ESCRITURA*****
var valorescrito="0";
var valforplc;
var plcaddress;

if(k<1){
  var write = namespace.addVariable({
    componentof: node[0], //Referencia a que nodo va a pertenecer la variable
    nodeId: "s=Escritura",
    browseName: "WRITE",
    dataType: "String",
    value: {
      get: function () {
        return new opcua.Variant({dataType: opcua.DataType.String, value: valorescrito });
      },
      set: (variant)=> {
        valorescrito=variant.value;
        var valwrite;
        if(variant.value[1]=="false"){
          valwrite=false;
        }else{
          if(variant.value[1]=="true"){
            valwrite=true;
          }else{
            valwrite=Number(variant.value[1]);
          }
        }
      }
    }
  });
  console.log("Adress= "+variant.value[0]);
  console.log("Valor= "+variant.value[1]);
  conn.writeItems([variant.value[0].toString()], [valwrite]);
}
});
}
}

```

Fig. 6. 26: Código para escribir las variables del PLC.

Elaborado por: El investigador

6.6 Pruebas de conexión y funcionamiento

6.6.1 Pruebas de conexión

Con el objetivo de corroborar el correcto funcionamiento de la plataforma web se realizaron pruebas de comunicación entre la plataforma web y el servidor, así como del servidor y el controlador. Posteriormente se verificara como se está comportando el tráfico de la red a través de la herramienta wiresharck.

A continuación procederemos a mostrar en la imagen 6.27 como se está enviando la información desde la página web al cliente OPC UA y posteriormente la información que esta siendo adquirida por el servidor.

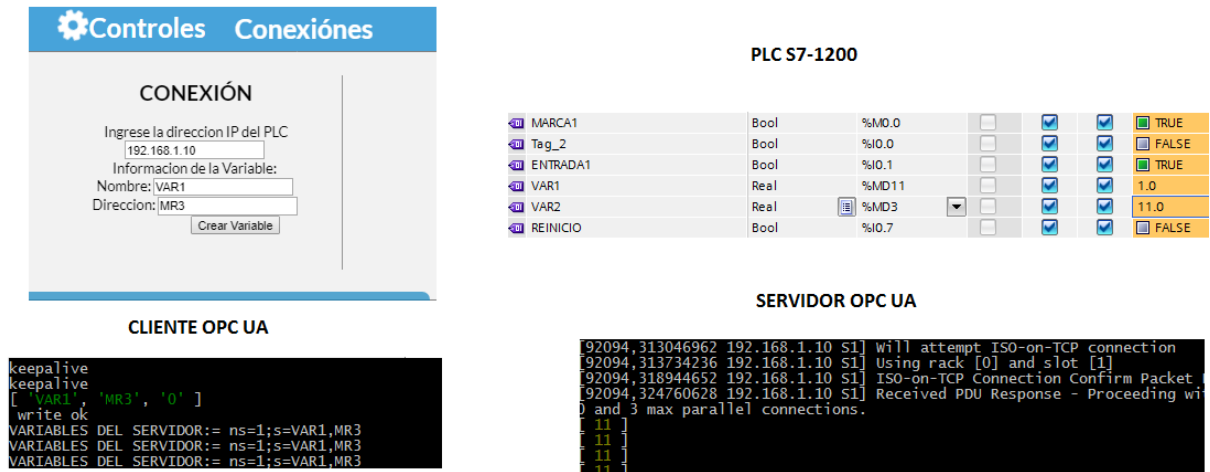


Fig. 6. 27: Pruebas de conexión.

Elaborado por: El investigador

Como se puede observar en la imagen anterior la página web envía la información al cliente OPC UA a través de web sockets. En este caso la información que se ha recibido es una variable denominada VAR1 con su respectiva dirección del PLC MR3. El cliente OPC selecciona esta información y la envía al servidor OPC UA. Una vez que el servidor OPC UA recibe esta información se crea una nueva variable para monitorizarla con un node id “ns=1;s=VAR1”, y además se establece la conexión con el PLC seleccionado. A continuación se añade la dirección al protocolo NODES7 y se recupera la información que se encuentra en la dirección MR3. Esta dirección es interpretada por el protocolo NODES7 y se recupera la información que se encuentra en la dirección MD3 del PLC como se puede apreciar en este caso el valor de esta variable es de 11.

Una vez que se ha establecido una variable para ser monitorizada la podemos enlazar a uno de los controles desarrollados como se puede observar en la figura 6.28.

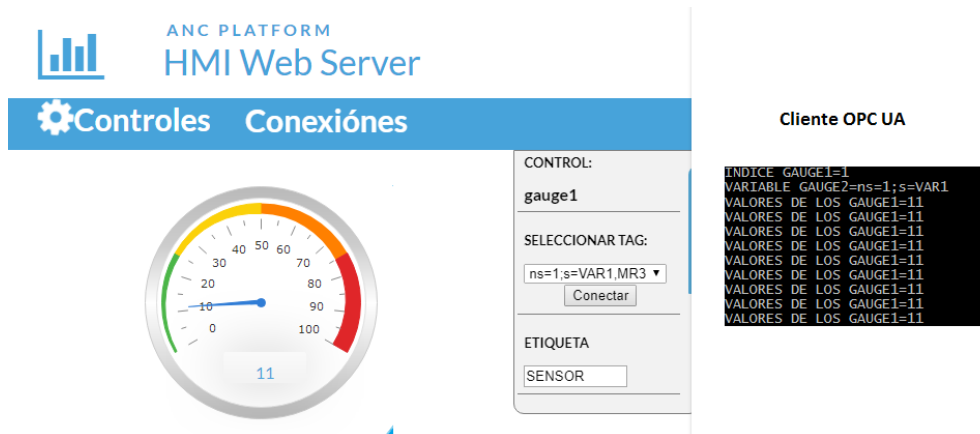


Fig. 6. 28: Monitorización de variables.

Elaborado por: El investigador

Como se puede apreciar en la figura, una vez se ha enviado al servidor OPC UA la información de las variables para su monitorización se tiene a disposición la lista de variables contenidas en el recuadro Seleccionar Tag. Al seleccionar la variable y presionar el botón conectar esta información se envía al cliente OPC UA y este a su vez establece la monitorización de la misma solicitando la información de la variable al servidor. Para el proceso de escritura de una variable en el PLC su proceso es muy similar. En este caso el control envía el node Id de la variable que recibirá la información a escribir. Posteriormente esta información se escribe en la dirección del PLC que se ha designado a través del protocolo NODES7. En la figura 6.29 se puede observar el proceso de escritura de la variable MR25.

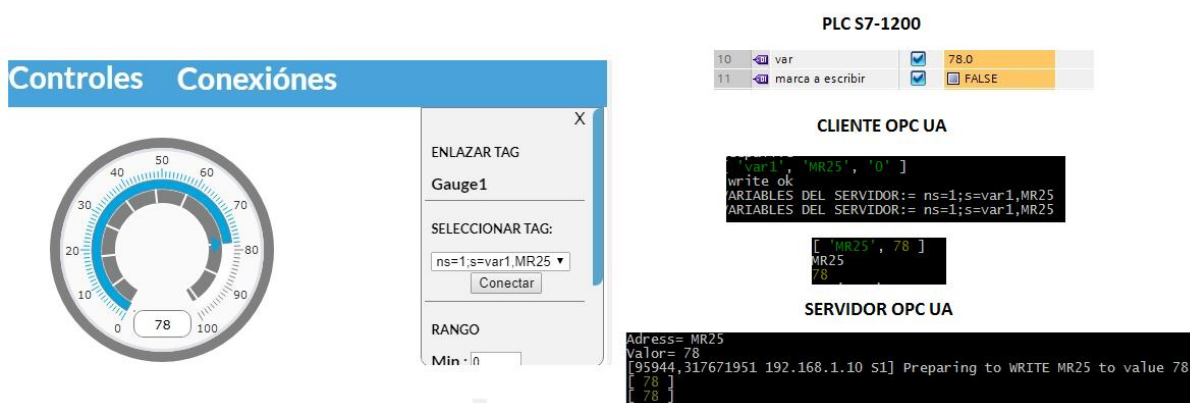


Fig. 6. 29: Escritura de variables.

Elaborado por: El investigador

6.6.2 Análisis de Tráfico

Paralelo a las pruebas de comunicación que se establecen entre la página web diseñada, el cliente, el servidor y el controlador es necesario determinar cuál es el tráfico que se genera en la red. Para lo cual se hará uso de la herramienta de software denominada Wireshark. Con esta herramienta es posible capturar el tráfico que atraviesa por las diferentes interfaces de la red. En la figura 6.30 se establece un esquema de cómo se monitorizara el tráfico de la red.

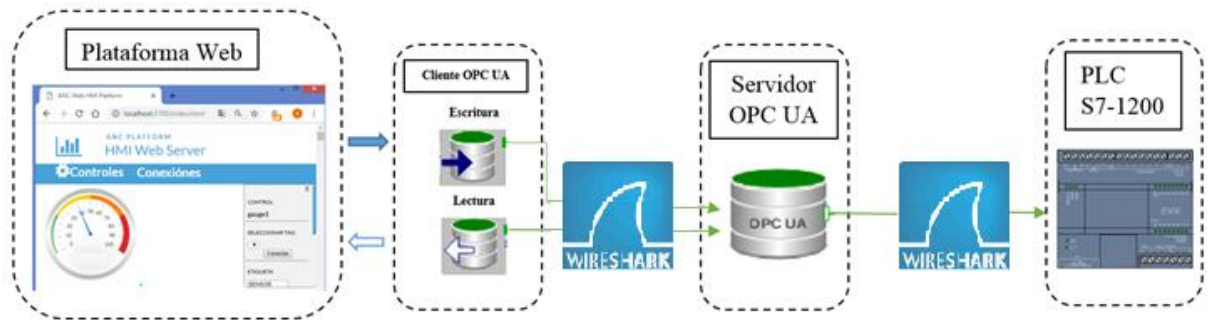


Fig. 6. 30: Esquema para la captura del tráfico de la red.

Elaborado por: El investigador

Se procede a iniciar el servidor y cliente OPC UA y capturar el tráfico que se genera al realizar este proceso. En la figura 6.30 podemos observar la captura de paquetes que se generan.

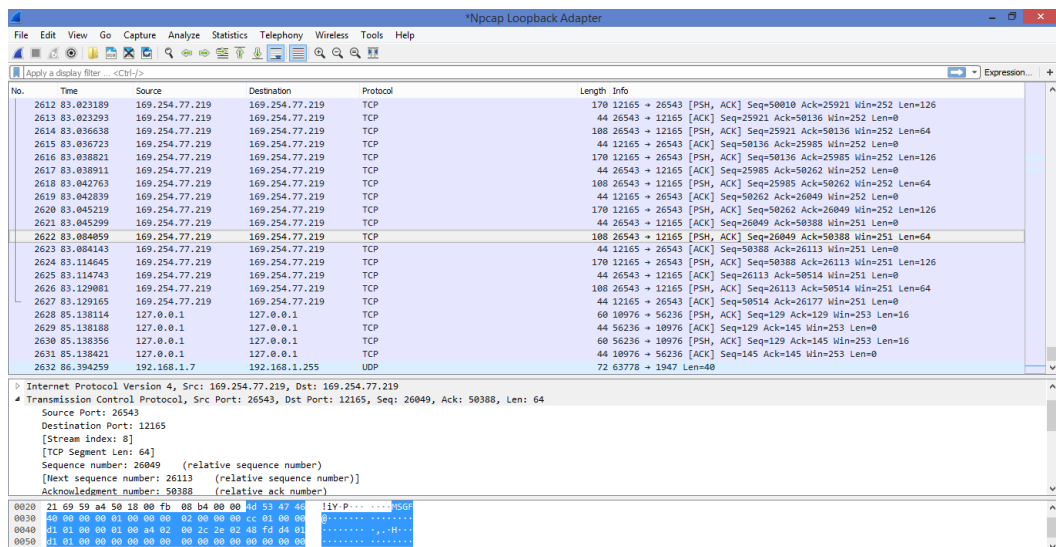


Fig. 6. 31: Captura de tráfico al iniciar el cliente y servidor OPC UA

Elaborado por: El investigador

En la figura podemos apreciar el tráfico que se genera al iniciar el proceso y proceder a enviar la información hacia el servidor OPC UA. Como se puede observar en la figura se genera tráfico saliendo del puerto 26543. En los parámetros descritos en el código del servidor OPC UA hemos designado a este puerto para establecer la comunicación. Como podemos observar se genera una comunicación bidireccional a través del envío de peticiones PSH y ACK respectivamente. Como podemos apreciar estos paquetes se envían a través del protocolo TCP ya que OPC UA desarrollado para Node js trabaja bajo este protocolo. A continuación se mostrara el detalle del potoco en la figura 6.32.

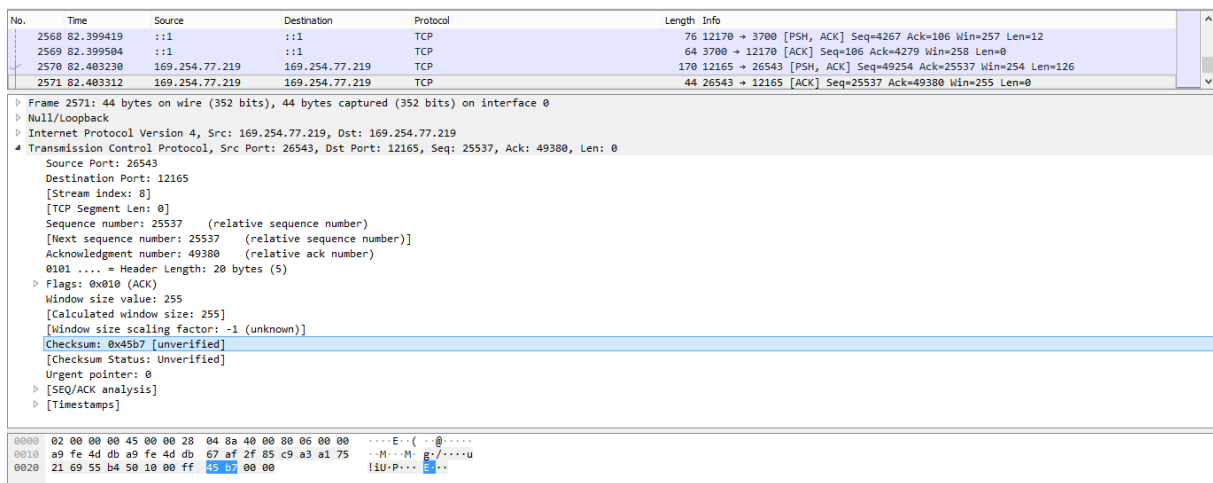


Fig. 6. 32: Detalle del protocolo de transmisión

Elaborado por: El investigador

A continuación se muestra la gráfica de Uso de Ancho de Banda en la figura 6.33. Grafica que se ha obtenido a través de la herramienta Graph IO en Wireshark, y mediante la cual se realizara un análisis del tráfico de paquetes que se genera al utilizar la plataforma web. En la gráfica podemos observar que existen picos de trafico los cuales se generan cuando un constante envío de información hacia el servidor se establece.

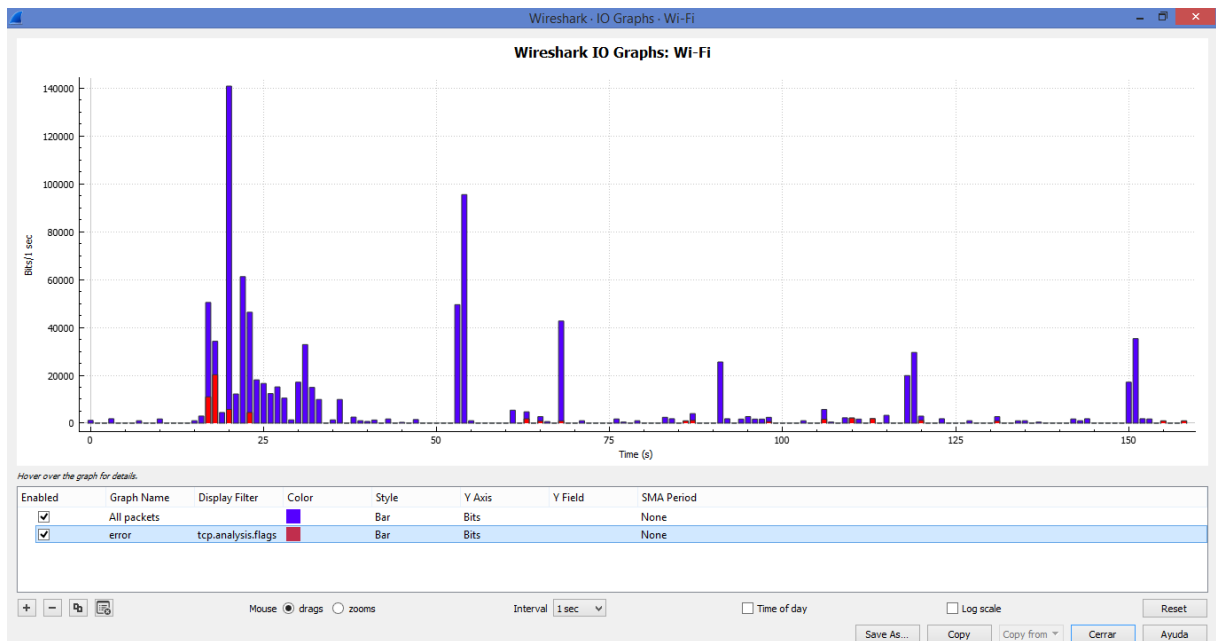


Fig. 6. 33: Grafica de Ancho de banda (bits/seg)

Elaborado por: El investigador

Como podemos apreciar en la figura 6.31 se observa que el pico más prominente llega a alcanzar los 140000 bits por segundo. De lo cual se concluye que cuando más se consume ancho de banda llega a alcanzar los 0.0175 (Megabytes por segundo). Este resultado nos permite determinar que el ancho de banda consumido por el protocolo es realmente bajo, por lo cual es una opción ideal para implementarlo en sistemas de bajos recursos.

En cuanto al análisis de paquetes del sistema y los errores que se generan en la comunicación se puede apreciar en la gráfica 6.34. Teniendo como número máximo de paquetes un pico de alrededor de 48 paquetes por segundo. De la gráfica también podemos apreciar que se presentan errores en la comunicación, siendo estos datos de paquetes perdidos o no útiles. Hecho que sucede debido a la naturaleza del protocolo TCP sobre el cual se ejecuta el protocolo OPC UA desarrollado en Node.js.

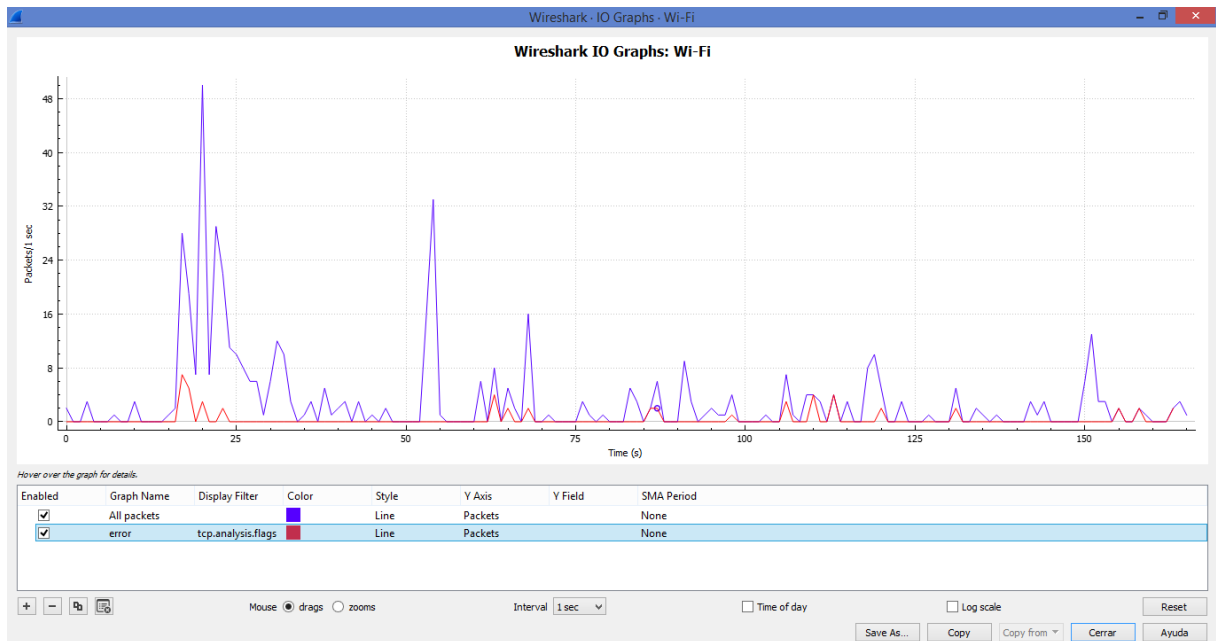


Fig. 6. 34: Grafica de la transmisión de paquetes

Elaborado por: El investigador

A continuación se procederá a analizar el tiempo de procesamiento de los paquetes generado en la gráfica 6.35. En la cual se expone el tiempo de captura de un paquete TCP. El cual cómo podemos apreciar en la figura alcanza un valor promedio de 50 ms.

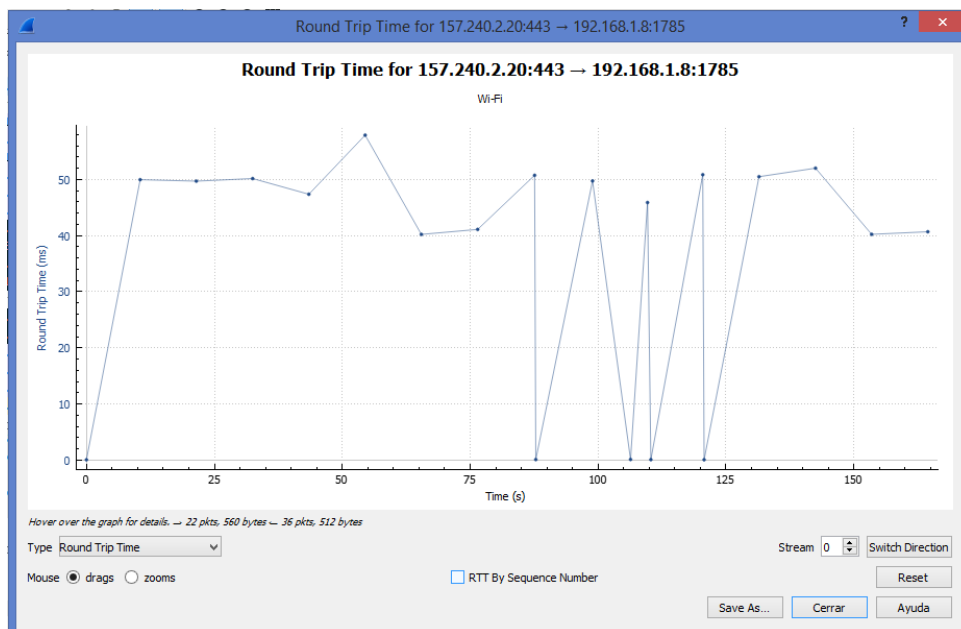


Fig. 6. 35: Tiempo de procesamiento de paquetes

Elaborado por: El investigador

A través del análisis de tráfico obtenido utilizando la herramienta Wireshark e interpretando las gráficas de ancho de banda, procesamiento de paquetes y tiempo de procesamiento de los mismos se procederá a realizar una comparación de paquetes entregados y recibidos por el protocolo OPC UA. La misma que se presenta en la tabla 6.1.

Tabla 6- 1: *Detalle cuantitativo del análisis de tráfico de la Red*

No.	Origen	Destino	PE(se)	PR(ce)	PP(nu)
1	192.168.1.8	192.168.1.10	42	37	5
2	192.168.1.10	192.168.1.8	36	33	3
3	104.27.142.156	192.168.1.8	40	35	5
4	192.168.1.8	104.27.142.156	41	34	7
5	192.168.1.8	34.237.55.225	38	33	5
6	34.237.55.225	192.168.1.8	39	35	4

Elaborado por: El investigador

PE(se): Paquetes entregados, sin errores

PR(ce): Paquetes recibidos, o con errores

PP(nu): Paquetes Perdidos, no útiles

1.5 Comprobación de la Hipótesis de la propuesta

Del análisis realizado y habiendo obtenido los valores estadísticos desplegados en la tabla 6-2 se procede a efectuar la siguiente comprobación de la Hipótesis planteada para el presente trabajo de investigación: El diseño de una plataforma web de software libre influye en la reducción de costos de monitorización de procesos y en la interfaz hombre-máquina.

Para llevar a cabo esta comprobación de manera efectiva se utilizara la información obtenida acerca del comportamiento de tráfico en la plataforma web utilizando el protocolo OPC UA. Todo esto se lo realizara mediante la prueba de chi-cuadrado y la aplicación de sus fórmulas detalladas a continuación.

Cálculo de Chi-cuadrado, Ecuación 1.

Ecuación 1: Chi-cuadrado

$$\lambda^2 = \sum \frac{(fo - fe)^2}{fe}$$

Cálculo de frecuencias esperadas, Ecuación 2.

Ecuación 2: Frecuencias esperadas

$$fe = \frac{\sum fila * \sum columna}{\sum Total}$$

Cálculo de grados de libertad, Ecuación 3.

Ecuación 3: Grados de libertad

$$v = (cant. filas - 1) * (cant. columnas - 1)$$

Cálculo del parámetro p , Ecuación 4.

Ecuación 4: Parámetro p

$$p = 1 - (nivel de significancia)$$

En la tabla 6-2 se muestra el detalle de frecuencias observadas mediante el análisis de tráfico de la red.

Tabla 6- 2: Frecuencias observadas del sistema

No. Transferencias	Paquetes entregados, sin errores	Paquetes Perdidos, no útiles	Suma filas
1	42	5	47
2	36	9	45
3	40	2	42
4	41	1	42
5	38	7	45
6	39	10	49
Suma columnas	236	34	270

Elaborado por: El investigador

Al aplicar la ecuación 2 podemos obtener los valores de frecuencias esperadas las cuales las podemos observar en la tabla 6-3. De igual manera se procede a realizar el cálculo correspondiente para obtener el valor del chi-cuadrado y los grados de libertad los cuales vienen dados por las ecuaciones 1 y 3 correspondientemente.

$$\begin{array}{lcl} \text{Chi-cuadrado} & \lambda^2 = & 11.80 \\ \text{Grados de libertad} & v = & 5 \end{array}$$

Tabla 6- 3: Frecuencias esperadas del sistema

No. Transferencias	fe PE(se)	fe PP(nu)	Suma filas
1	41.08	5.9	47
2	39.3	5.7	45
3	36.7	5.3	42
4	36.7	5.3	42
5	39.3	5.7	45
6	42.8	6.2	49
Suma columnas	236	34	270

Elaborado por: El investigador

Se ha determinado el valor del parámetro p a través de la ecuación 4 y considerando el error al rechazar la hipótesis nula siendo verdadera. Por lo cual se ha establecido una significancia del 95%. Valor que establece que hay una probabilidad del 0.5% de que la hipótesis nula sea verdadera.

$$p = 0,95$$

Al realizar la comprobación correspondiente de los valores del parámetro p y v en la Tabla de valores Críticos del chi-cuadrado, obtenemos un resultado de 11.070. Lo cual se puede apreciar de mejor manera en la figura 6.36.

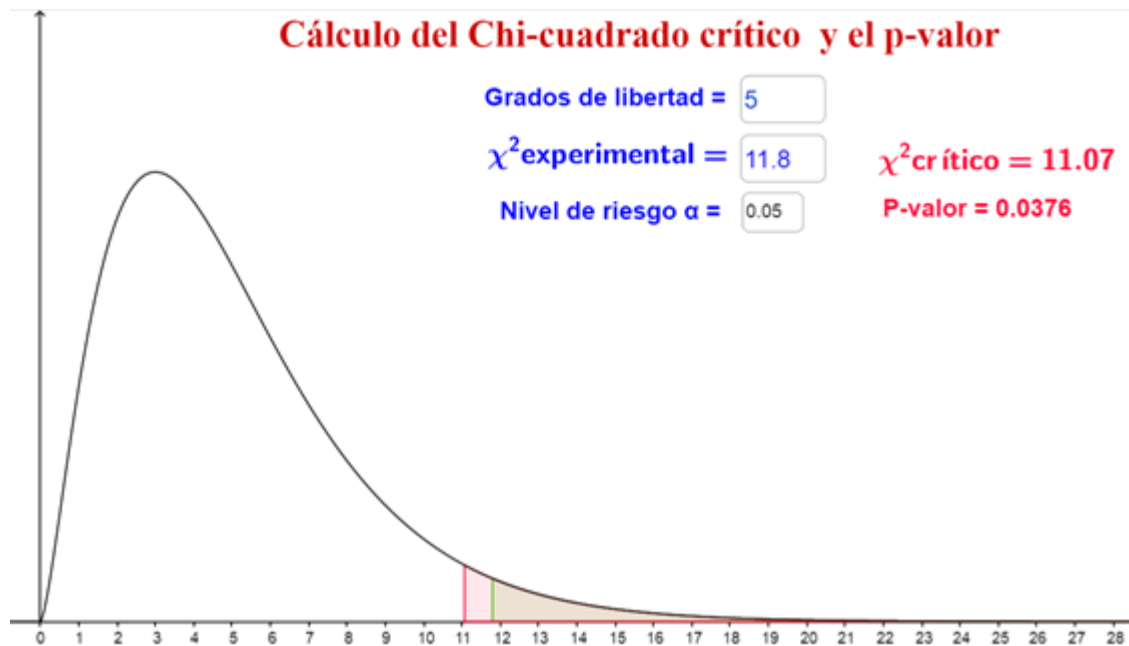


Fig. 6. 36: Gráfica del cálculo del Chi-cuadrado

Elaborado por: El investigador

$$\lambda^2 \text{ calculado (11,80)} \leq \lambda^2 \text{ crítico (11,070)}$$

No cumple

Del resultado obtenido se aprecia que el valor del chi-cuadrado es mayor que su valor crítico, por lo cual no cumple con la comparación expuesta. Debido a ello se concluye que el sistema desarrollado funciona de una manera adecuada pudiendo ser utilizado en lugar cualquier otro software comercial. Con lo cual se puede decir que el diseño de un sistema de software libre para la creación de pantallas de monitorización si influye en la reducción de los costos de monitorización.

6.6 Cuadro comparativo de características de la plataforma desarrollada.

A continuación se han desarrollado dos cuadros comparativos de las principales características entre la plataforma desarrollada y los diferentes softwares utilizados para programación de pantallas HMI. Mismo que se muestra en las tablas 6-4 y 6-5.

Tabla 6- 4: Cuadro comparativo de características

Software	Descripción	Descripción		Características
		Libre	Propietario	
WINCC	Es un sistema de visualización de procesos escalable y dotado de potentes funciones para la supervisión de procesos automatizados.		Siemens	<p>Disponible con dos tipos de licencia paquete completo y licencia para runtime de 128 hasta 256k Power tags</p> <p>Costo en dependencia del número de tags a utilizarse.</p> <p>Configuración fácil y escalable</p> <p>Microsoft SQL Server integrado para registro histórico de datos.</p>
INTOUCH	Es un software de visualización HMI con gráficos de alta calidad.			Gráficos de alta calidad.

	<p>Lo cual permite a los desarrolladores crear aplicaciones orientadas a crear contenido significativo que impulse la productividad y el ahorro.</p>	<p>Wonderware</p>	<p>Costo en dependencia del número de tag a utilizarse.</p> <p>Proporciona información contextualizada intuitiva para el usuario.</p>
<p>LABVIEW</p>	<p>Software de potente entorno de desarrollo grafico para el diseño de aplicaciones y presentación de datos, basado en un lenguaje de programación de baja complejidad.</p>	<p>National Instruments</p>	<p>Intuitivo</p> <p>Herramientas de desarrollo y librerías de alto nivel específicas para aplicaciones</p> <p>Cientos de funciones para E/S, control y análisis de datos.</p> <p>Depuración grafica integrada.</p> <p>Costo de licencias elevado.</p>

<p>OPEN AUTOMACIÓN SOFTWARE</p>	<p>Software basado en programación web para la monitorización, análisis y entrega de datos.</p>	<p>Open Automation Software</p>	<p>De programación compleja</p>
			<p>Permite la interconexión con PLCs de varias marcas, bases de datos y plataformas IoT.</p>
			<p>Posee un diseño modular en red</p>
			<p>Necesario el pago de suscripción para su utilización</p>
			<p>Sistema basado en programación web.</p>
			<p>Interfaz sencilla que permite la creación de HMIs sin la necesidad de programación.</p>
<p>PROYECTO DESARROLLADO</p>	<p>Plataforma web basada en JavaScript, HTML, CSS y OPC UA para la</p>	<p>Código Libre</p>	

creación de pantallas de monitorización.

Sistema de código abierto y libre de costo

Permite la creación de pantallas HMI con graficas de muy buena calidad.

Permite una comunicación segura y eficiente a través de protocolo OPC UA.

Elaborado por: El investigador

Tabla 6- 5: Valoración de características de los software para creacion de pantallas HMI

Software	Interfaz sencilla	Código abierto	No se necesita programar	Configuración rápida	Comunicación eficiente	Costo bajo	Valoración
WINCC	✓		✓	✓	✓		4
INTOUCH	✓			✓	✓		3
LABVIEW					✓		1
OPEN AUTOMATION SOFTWARE					✓	✓	2
PROYECTO DESARROLLADO	✓	✓	✓	✓	✓	✓	5

Elaborado por: El investigador

6.7 Análisis de precios de licencias de software

El objetivo del presente trabajo de investigación es la creación de una plataforma web de libre acceso para la creación de pantallas HMI, que permita a las empresas de la industria disponer de una opción de bajo costo y de características competitivas al momento de optar por un software para crear sus pantallas de monitorización. Es por este motivo que es de suma importancia hacer un análisis de costos de utilización de los principales softwares para de monitorización en relación a la propuesta planteada. Por lo cual se ha realizado un estudio a nivel nacional como internacional de los software más utilizados para llevar a cabo esta tarea, arrojando como resultado que uno de los software en utilizados para la creación de HMIs en la región central es WinCC especialmente en la Región 3, en donde la utilización de este software pre denomina. Es por este motivo que a continuación se presentara los costos referenciales de la utilización de estos softwares.

Tabla 6- 6: Análisis de costos de softwares para creación de pantallas HMI

SOFTWARE	COSTO
WINCC	El costo de la utilización de WinCC establecido por la empresa SIEMENS varía en relación de las características que se desee emplear, más específicamente por el número de Tags a utilizar, variando en su costo desde los 500 dólares americanos hasta los 7000 dólares. Ver anexo 4. (Siemens, 2019a, 2019b)
INTOUCH	De igual manera otro software muy utilizado en la industria y de muy buenas características para crear pantallas de monitorización es In-Touch. El cual su precio de utilización varia al igual que el anterior en dependencia de parámetros como Tags a utilizarse, Data Logging, Histogramas en muchos otros parámetros más. Variando en el costo de su licencia desde los 1100 dólares la versión más básica hasta los 10500 la versión más equipada. (WonderWare, 2019)
LABVIEW	Al igual que los anteriores Labview es una de las opciones más difundidas al momento de hablar de automatización y HMIs, variando en sus costos de licencias desde los 500 dólares americanos por año

	<p>hasta llegar a los 5500 por año por equipo. Lo cual lo convierte en uno de los softwares más costosos y por ende una opción poco viable de implementarse.</p>
	<p>(National Instruments, 2019)</p>
<p>OPEN AUTOMACIÓN SOFTWARE</p>	<p>Open Automation Software es una plataforma para la monitorización basada en la web. Aunque su costo es relativamente menor a los demás sigue siendo una versión de pago. Sus costos de utilización varían al igual que los softwares anteriores en dependencia de las características de utilización como Tags, Histogramas, Data Logging entre muchos otros más variando en su costo desde los 700 dólares aproximadamente hasta los 1600. Ver anexo 4.</p>
	<p>(Open Automation Web Site, 2018)</p>
<p>PROPUESTA PLANTEADA</p>	<p>A diferencia de los softwares de versión de pago descritos anteriormente la plataforma diseñada en el presente trabajo de investigación estaría alojada en la red libre para su descarga y utilización. El presente trabajo de investigación estaría disponible bajo la licencia GPL en la cual el desarrollador conserva los derechos de autor, pero permite su distribución, modificación y uso siempre y cuando, en el caso de que el software se modifique el nuevo software que se desarrolle como resultado quede obligatoriamente bajo la misma licencia. (Salas Pasuy, Acuña, Manrique, Glogower Abadi, & Peña Valenzuela, 2010) Esto permitiría que más personas y desarrolladores podrían mejorar sus características y modificarlo en dependencia de sus necesidades haciéndolo cada vez una herramienta más robusta y completa para crear pantallas de monitorización HMIs.</p>

Elaborado por: El investigador

De la información presentada en la tabla 6.2 se puede concluir que el costo del software de para la creación de pantallas de monitorización es un factor realmente importante a tomar en cuenta al momento de pensar en automatización. Es por ello que una plataforma como la propuesta incidiría de manera directa en el desarrollo de más procesos automatizados y

monitorizados ya que es una plataforma libre que la industria podría aprovechar para saltar esa gran barrera que supone la utilización de software propietario. Con lo cual se podría conseguir una reducción en fallas de producción y se podría mejorar la calidad de los productos realizados al contar con procesos monitorizados constantemente.

6.8 Conclusiones y Recomendaciones

Conclusiones

- El análisis y entendimiento de las técnicas de programación web permitió concluir que la forma de crear una aplicación web es muy diferente a la de crear software habitual. Ya que para crear una aplicación web funcional se necesita de tres lenguajes de programación que son HTML, CSS y JavaScript mientras que para crear un software normal se lo puede hacer con un solo lenguaje y en un solo software de programación.
- Es de suma importancia definir el correcto orden de las dependencias que se utilizara para la creación de la plataforma web, ya que ubicarlas en un lugar no adecuado causa conflictos entre el código de los plugins.
- Es importante entender el uso de los atributos clase e id en el código, ya que su correcta interpretación evita repetir innecesariamente extractos de código que se puede usar para más de un objeto.
- La correcta utilización del protocolo Socket.io permite que se pueda actualizar los parámetros de información tanto de los controles virtuales de la plataforma web, así como de la información proveniente del controlador y del servidor OPC UA respectivamente de una manera adecuada y eficiente.
- Se concluyó también que la utilización del protocolo OPC UA es la mejor opción para establecer la comunicación con la página web y el controlador ya que posee un sistema de discriminación de variables a través de node Ids, lo cual evita que se puedan crear copias innecesarias de estas variables. Así como determinar exactamente cuál es la variable que se desea monitorear.
- Se concluyo además que a través de utilización del protocolo OPC UA se obtuvo una mejora en el rendimiento de la comunicación en comparación con otros protocolos como MQTT o los métodos get y post para enviar información sobre http. Ya que el proceso de suscripción hacia las diferentes variables a monitorizar permite que la información se pueda recuperar de una manera mas rápida y eficiente.

Recomendaciones

- Se recomienda tener bastante cuidado con el manejo de los id de las etiquetas ya que si un id es establecido dos veces en diferente lugar del programa se ejecutara normalmente pero los procesos asociados a dicho id se ejecutaran de manera errónea.
- Se recomienda tratar de separar en su mayoría los lenguajes de programación web ya que tener una combinación de los mismos puede causar problemas al momento de modificarlos.
- Es recomendable también crear interfaces simples con las que el usuario pueda interactuar de manera intuitiva.
- Se recomienda al usuario de la plataforma tomar en consideración las normas para creación de pantallas de monitorización como la norma Gemma o la ISA 101. Con lo cual se puede evitar la mala utilización de la plataforma web.
- Se recomienda poner bastante cuidado al establecer los nombres de los equipos a interactuar con el protocolo OPC UA, ya que si estos se establecen de manera errónea se presentaran problemas conexión. Esto se debe a que los certificados de seguridad creados por el protocolo intentaran establecer los parámetros de conexión con información que no es la adecuada.
- Es recomendable tener en cuenta que en lenguaje Java Scripr utilizado para la programación tanto del cliente, del servidor OPC UA, asi como de la interfaz de la plataforma web no realiza una diferenciación entre tipo de datos de las variables. Lo cual nos permite tener tener variables de distitntios tipos e incluso arrays con varios tipos de datos lo cual permite simplificar la programacion especialmente en los procesos de envio de información entre el cliente y el servidor OPC UA.

BIBLIOGRAFÍA

- Abbas, H. A., & Mohamed, A. M. (2011). Review on the Design of Web Based SCADA Systems Based on OPC DA Protocol. *International Journal of Computer Networks (IJCN)*, 2(6), 266–277. Retrieved from <http://www.cscjournals.org/journals/IJCN/issue-manuscripts.php?v=2&i=6>
- Arenas L, Castilla A, R. D. (2015). Interfaz Hombre Máquina, 1–4. Retrieved from <http://www.uco.es/grupos/eatco/automatica/ihm/>
- Ayyaz Ismail, N., M. K., B., & Prashant Prabhakar, S. (2015). Review Paper on AJAX Comet and WebSocket Uses for Web HMI/SCADA. *International Journal of Engineering Research and General Science*, (October). Retrieved from <http://pnrsolution.org/Datacenter/Vol3/Issue5/133.pdf>
- Bluefish. (2018). Bluefish Editor : Home. Retrieved from <http://bluefish.openoffice.nl/index.html>
- Bolton, W. (William). (2006). *Programmable logic controllers*. Elsevier/Newnes. Retrieved from <https://books.google.com/books?id=zsyTTGxCIdMC&printsec=frontcover&dq=PLCs&hl=es&sa=X&ved=0ahUKEwiggpDV8LrZAhVtkeAKHe9GBzgQ6AEIODAC#v=onepage&q=PLCs&f=false>
- Boyd, B., Gauci, J., Robertson, M. P., Duy, N. Van, Gupta, R., Gucer, V., ... International Business Machines Corporation. International Technical Support Organization. (2017). *Building Real-time mobile solutions with MQTT and IBM MessageSight*.
- Castro Gil, M.-A., Díaz Orueta, G., & Mur Pérez, F. (2000). *Comunicaciones industriales : principios básicos*. UNED - Universidad Nacional de Educación a Distancia.
- Cavalieri, S., Di Stefano, D., Salafia, M. G., & Scroppo, M. S. (2017). Integration of OPC UA into a web-based platform to enhance interoperability. *IEEE International Symposium on Industrial Electronics*, 1206–1211. <https://doi.org/10.1109/ISIE.2017.8001417>
- Chavarría Meza, L. E. (2007). *Scada System ´S & Telemetry*. Universidad De Mexico. <https://doi.org/MM2086SOI9069>

- David, F. (2011). *Diseño de un interfaz hombre hombre- máquina con un observador dinámico en tiempo real*. Universidad Autonoma de Madrid.
- Durkop, L., Imtiaz, J., Trsek, H., Wisniewski, L., & Jasperneite, J. (2013). Using OPC-UA for the autoconfiguration of real-time Ethernet systems. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)* (pp. 248–253). IEEE.
<https://doi.org/10.1109/INDIN.2013.6622890>
- Eguiluz, J. (2009). *Introducción a CSS*.
- Equipo Vertice. (2009). *Diseño básico de páginas web en HTML*. Publicaciones Vértice.
 Retrieved from
<https://books.google.com/books?id=Q4VL8Tiy7gcC&printsec=frontcover&dq=html&hl=es&sa=X&ved=0ahUKEwi2z5nBzLzZAhUIXIMKHTi-CBgQ6AEIMDAB#v=onepage&q=html&f=false>
- Eras, C., Quito, S., Bolivar, Z., & Matamoros, H. (2010). Sistema SCADA, 1–6.
- Fernando Luna, Claudio Peña Millahual, M. I. (2015). *PROGRAMACION WEB Full Stack 1 - Ecosistema Web: Desarrollo*. Retrieved from
<https://books.google.com/books?id=ucQ9DwAAQBAJ&pg=PA16&dq=lenguajes+de+programacion+web&hl=es&sa=X&ved=0ahUKEwj5ffvyrzZAhUEq1kKHGLBOsQ6AEILTAB#v=onepage&q=lenguajes de programacion web&f=false>
- Garro, A. (2015). *HTML5*.
- Gauchat, J. D. (2012). *El grab libro de HTML5, CSS3 y Javascript* (Primera ed). Barcelona: Marcombo.
- Germán Corona Ramírez, L., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores*. Larousse - Grupo Editorial Patria. Retrieved from
<https://books.google.com/books?id=wMm3BgAAQBAJ&printsec=frontcover&dq=sensores&hl=es&sa=X&ved=0ahUKEwiqk43s-brZAhUJON8KHQehBZcQ6AEILDAB#v=onepage&q=sensores&f=false>
- González, U. G. (2017). *JavaScript , ¡ Inspírate !*
- Gorecky, D., Schmitt, M., Loskyll, M., & Zühlke, D. (2014). Human-machine-interaction in

- the industry 4.0 era. *Proceedings - 2014 12th IEEE International Conference on Industrial Informatics, INDIN 2014*, 289–294.
<https://doi.org/10.1109/INDIN.2014.6945523>
- Gruner, S., Pfrommer, J., & Palm, F. (2015). A RESTful extension of OPC UA. In *2015 IEEE World Conference on Factory Communication Systems (WFCS)* (pp. 1–4). IEEE.
<https://doi.org/10.1109/WFCS.2015.7160557>
- Gualotuña Barahona, C. G., & Villagómez Palacios, C. H. (2014). Diseño e Implementación del sistema automatizado con interfaz HMI (Interfaz Hombre Maquina) para para la máquina termo-selladora de envases de pintura controlada por el PLC SIMATIC S7-1212C y supervisada por el panel táctil KTP400PN, de la Industria Envat. Retrieved from <https://www.dspace.ups.edu.ec/handle/123456789/6862>
- Instituto Nacional de Estadística e Informática. (2015). *Instituto Nacional de Estadística e Informática*.
- JS Foundation. (2019). Node-RED. Retrieved April 24, 2019, from <https://nodered.org/>
- López, M., & Zaragoza, J. (2014). Aplicación docente de sistema SCADA y sensor inteligente sobre maqueta de pruebas. Retrieved from <http://repositorio.upct.es/handle/10317/4242>
- Macromedia. (2018). Adobe Dreamweaver CC | Software de diseño para sitios web.
 Retrieved from www.adobe.com/la/products/dreamweaver.html
- Mahnke, W., Leitner, S.-H., & Damm, M. (2009). *OPC unified architecture*. Springer-Verlag.
- Mendiburu, H. (2013). *Instrumentación Virtual Industrial*.
- Murphey, R., Aragón, G. R., Maronna, P., Ciccale, D., Padolsey, D. J., & Irish, P. (2011). *Fundamentos de jQuery Índice general*.
- Nafría, I. (2008). *Web 2.0 : el usuario, el nuevo rey de internet*. Gestión 2000. Retrieved from https://books.google.com/books?id=1fZi_ndyc-wC&pg=PA16&dq=plataforma+web&hl=es&sa=X&ved=0ahUKEwjowajzwLzZAhVIGt8KHWJIBkgQ6AEIMjAC#v=onepage&q=plataforma web&f=false
- National Instruments. (2019). LabVIEW - National Instruments. Retrieved January 8, 2019, from <http://www.ni.com/es-mx/shop/labview/select-edition.html>

- NovaTech. (2015). WEbserver HMI | NovaTech Substation Automation & Cyber Security. Retrieved February 19, 2018, from <https://www.novatechweb.com/substation-automation/webserver-hmi/>
- Open Automation Software. (2019). Pricing | Industrial Internet of Things Data Platform |. Retrieved January 8, 2019, from <https://openautomationsoftware.com/products/pricing/>
- Open Automation Web Site. (2018). IIOT, Industrial Internet of Things Platform | Open Automation Software. Retrieved February 19, 2018, from <https://openautomationsoftware.com/>
- Palm, F., Grüner, S., Pfrommer, J., Graube, M., & Urbas, L. (2015). Open source as enabler for OPC UA in industrial automation. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2015–Octob.* <https://doi.org/10.1109/ETFA.2015.7301562>
- Parga Carlos Roman. (2013). *Desarrollo de una interfaz hombre máquina usando smartphone con aplicación a brazo manipulador*. INSTITUTO POLITECNICO NACIONAL DE ZACATENCO. Retrieved from <https://www.cs.cinvestav.mx/TesisGraduados/2013/TesisCarlosParga.pdf>
- Pasquale, G. De, Kim, S., & Pasquale, D. De. (2015). GoldFinger : Wireless human-machine interface with dedicated software and biomechanical energy harvesting system, 4435(c). <https://doi.org/10.1109/TMECH.2015.2431727>
- Penin, A. R. (2007). *Sistemas SCADA*. Retrieved from <http://books.google.com/books?id=32kgCNG34TwC&pgis=1>
- Pérez, J. E. (2008). Ajax, 251.
- Pérez, J. E. (2009). *JavaScript*.
- Puig, J. C. (2014). *CSS3 y Javascript avanzado*. *CSS3 y Javascript avanzado* (Vol. 1).
- Rajalakshmi, A., & Shahnasser, H. (2018). Internet of things using node-red and alexa. *2017 17th International Symposium on Communications and Information Technologies, ISCIT 2017, 2018–Janua*, 1–4. <https://doi.org/10.1109/ISCIT.2017.8261194>
- Revista Electroindustria. (2007). *Revista Electroindustria - Sistemas de Interfaz Hombre-*

- Máquina, HMI: Una ventana abierta a los procesos. Retrieved June 19, 2019, from <http://www.emb.cl/electroindustria/articulo.mvc?xid=837>
- Rodríguez Penin, A. (2007). *Sistemas SCADA*. Retrieved from <https://books.google.com/books?id=cNQfjbBcUq8C&printsec=frontcover&dq=sistemas+scada+mexico+libros&hl=es&sa=X&ved=0ahUKEwjAi9mKmlrZAhVqx1kKH6mBh8Q6AEIKDAA#v=onepage&q&f=false>
- Ruíz Lizama, E., Inche Mitma, J. L., & Chung Pinzás, A. R. (2016). Desarrollo de una interfaz hombre máquina orientada al control de procesos. *Industrial Data*, 11(1), 70. <https://doi.org/10.15381/idata.v11i1.6039>
- Salas Pasuy, B., Acuña, D. C., Manrique, P. V., Glogower Abadi, D., & Peña Valenzuela, D. (2010). *Software libre y software propietario : impacto jurídico, económico y cultural en Colombia*.
- Schleipen, M., Gilani, S. S., Bischoff, T., & Pfrommer, J. (2016). OPC UA & Industrie 4.0 - Enabling Technology with High Diversity and Variability. *Procedia CIRP*, 57, 315–320. <https://doi.org/10.1016/j.procir.2016.11.055>
- SELL, R., & OTTO, T. (2009). Online Engineering for Future Factory. *Annals of DAAAM & Proceedings*, (November 2009), 1695–1696. Retrieved from <http://libezproxy.open.ac.uk/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=47081341&site=ehost-live&scope=site>
- Siemens. (2014). Manual Controlador programable S7-1200.
- Siemens. (2019a). Distribuidor oficial de productos electricos Siemens en Argentina. Retrieved January 8, 2019, from <http://www.control-technics.com/lista-de-precios-siemens.asp?l=1&offset=320>
- Siemens. (2019b). Siemens Software TIA Portal V14 WinCC Runtime - PLC-City - PLC-City. Retrieved January 8, 2019, from <https://www.plc-city.com/shop/en/siemens-software-tia-portal-v14-wincc-runtime.html>
- Suarez, A. (2017). Guía de Estilo final. Retrieved from https://www.academia.edu/35795978/Guía_de_Estilo_final

- Sublime Text. (2018). Sublime Text - A sophisticated text editor for code, markup and prose. Retrieved from <http://www.sublimetext.com/>
- Turc, T. (2015). Using WEB Services in SCADA Applications. *Procedia Technology*, 19(April), 584–590. <https://doi.org/10.1016/j.protcy.2015.02.083>
- Van Tan, V., Yoo, D.-S., & Yi, M.-J. (2009). Device Integration Approach to OPC UA-Based Process Automation Systems with FDT/DTM and EDDL (pp. 1001–1012). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04020-7_108
- Wilamowski, B. M., & Irwin, J. D. (2011). *The industrial electronics handbook Industrial communication systems*. CRC Press.
- WonderWare. (2019). HMI Software - Wonderware InTouch - WonderWare. Retrieved January 8, 2019, from <http://www.wonderware.es/HMI-SCADA/InTouch/>

ANEXO 1 CÓDIGO HTML

ESTRUCTURA DE LA PLATAFORMA WEB

```
<!DOCTYPE html>
<html lang="en" class="no-js">
  <head>
    <meta charset="utf-8">
    <title>ANC Web HMI Platform</title>

    <!-- ESTILOS INTERFACE PRINCIPAL -->
    <link rel="stylesheet" type="text/css" href="css/default.css"/>
    <link rel="stylesheet" type="text/css" href="css/component.css"/>

    <!-- AWSOME FAMILY -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

    <!-- ESTILO BOTONI -->
    <link rel="stylesheet" href="css/boton1.css" type="text/css" media="screen" />

    <!-- ESTILO PROPIEDADES -->
    <link rel="stylesheet" type="text/css" href="css/propiedades.css" />

    <!-- INDICADOR 1 -->
    <link rel="stylesheet" href="js/jqx.base.css" type="text/css" />

    <!-- ESTILO LED -->
    <link rel="stylesheet" type="text/css" href="css/led.css" />

    <!-- ESTILO C2GAUGE1 -->
    <link rel="stylesheet" type="text/css" href="css/c2gauge.css" />

    <!-- ESTILO LABEL -->
    <link rel="stylesheet" type="text/css" href="css/label.css" />

    <!-- ESTILO DE FUENTES-->
    <link rel="stylesheet" type="text/css" href="css/fontselect-default.css" />

    <!-- FUENTES -->
    <link rel="stylesheet" type="text/css" href="css/fontselect-alternate.css" />

    <!--TREND -->
    <link rel="stylesheet" type="text/css" href="css/dataviz.chart.css" />
    <link rel="stylesheet" type="text/css" href="themes/le-frog/styles.css" />
```

```

<link rel="stylesheet" type="text/css" href="css/trend.css" />

<!--SLIDER-->
<link rel="stylesheet" type="text/css" href="css/dataviz.gauges.css" />

<!-- DISPLAY-->
<link rel="stylesheet" type="text/css" href="css/display.css" />

<!--POWER BOTTON-->
<link rel="stylesheet" type="text/css" href="css/style2.css">

<!--PERILLA-->
<link href="http://www.jqueryscript.net/css/jqueryscipttop.css" rel="stylesheet"
type="text/css">
<link href="jquery.rotaryswitch.css" rel="stylesheet" type="text/css">

<!-- PERILLA -->
<link rel="stylesheet" type="text/css" href="css/perilla.css"/>

<!-- SWITCH -->
<link rel="stylesheet" type="text/css" href="css/switch.css"/>

<!-- SLIDER -->
<link rel="stylesheet" type="text/css" href="css/icslider.css"/>

<!-- TANQUE2 -->
<link rel="stylesheet" type="text/css" href="css/tanque2.css"/>

<!-- AJUSTES -->
<link rel="stylesheet" type="text/css" href="css/ajustes.css"/>

<!--sockets-->
<script src='/socket.io/socket.io.js'></script>

<script>(function(e,t,n){var
r=e.querySelectorAll("html")[0];r.className=r.className.replace(/(^\s)no-
js(\s|$)/,"$1js$2"))(document,window,0);
</script>

<!--TANQUE-->
<script src="https://cdn.anychart.com/releases/v8/js/anychart-
base.min.js?hcode=be5162d915534272a57d0bb781d27f2b"></script>
<script src="https://cdn.anychart.com/releases/v8/js/anychart-
ui.min.js?hcode=be5162d915534272a57d0bb781d27f2b"></script>

```



```

<script src="https://cdn.anychart.com/releases/v8/js/anychart-
exports.min.js?hcode=be5162d915534272a57d0bb781d27f2b"></script>
<script src="https://cdn.anychart.com/releases/v8/js/anychart-linear-
gauge.min.js?hcode=be5162d915534272a57d0bb781d27f2b"></script>
<script src="http://code.jquery.com/jquery-1.12.0.min.js"></script>
<link href="https://cdn.anychart.com/releases/v8/css/anychart-
ui.min.css?hcode=be5162d915534272a57d0bb781d27f2b" type="text/css" rel="stylesheet">
<link href="https://cdn.anychart.com/releases/v8/fonts/css/anychart-
font.min.css?hcode=be5162d915534272a57d0bb781d27f2b" type="text/css"
rel="stylesheet">

<!-- CLIENTE JAVASCRIPT MQTT -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js"
type="text/javascript"></script>

<!-- SCRIPTS Y DEPENDENCIAS -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

<!-- DEPENDENCIAS DE EL C2GAUGE1 -->
<!-- Ignite UI Required Combined CSS Files -->
<link href="http://cdn-
na.infragistics.com/igniteui/2018.1/latest/css/themes/infragistics/infragistics.theme.css"
rel="stylesheet" />
<link href="http://cdn-
na.infragistics.com/igniteui/2018.1/latest/css/structure/infragistics.css" rel="stylesheet" />
<script src="http://ajax.aspnetcdn.com/ajax/modernizr/modernizr-2.8.3.js"></script>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script src="http://code.jquery.com/ui/1.11.1/jquery-ui.min.js"></script>

<!-- JQUERY -->
<script src="http://code.jquery.com/jquery-1.12.0.min.js"></script>
<script src="js/jquery-1.12.4.min.js"></script>
<script src="jquery-2.0.2.min.js"></script>
<script src="https://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<script src="https://rawgit.com/aterrien/jQuery-
Knob/master/dist/jquery.knob.min.js"></script>

<!-- LIBRERIA GAUGE 1 -->
<script src="js/jqxc.js"></script>
<script src="js/jqxdata.js"></script>
<script src="js/jqxdraw.js"></script>
<script src="js/jqxchart.core.js"></script>
<script src="js/demos.js"></script>
<script src="js/jqxcgauge.js"></script>

```

```

<!-- LIBRERIA GAUGE 2-->
<script src="js/jqxknob.js"></script>
<script src="js/jqxnumberinput.js"></script>

<!-- TREND -->

<!-- SLIDER -->
<script src="js/dataviz.gauges.min.js" type="text/javascript"></script>

<!-- Ignite UI Required Combined JavaScript Files -->
<script src="http://cdn-
na.infragistics.com/igniteui/2018.1/latest/js/infragistics.core.js"></script>
<script src="http://cdn-
na.infragistics.com/igniteui/2018.1/latest/js/infragistics.dv.js"></script>

<!-- SLIDER -->
<script src="js/dataviz.gauges.min.js" type="text/javascript"></script>

<!--PERILLA-->
<script src="js/jquery.rotaryswitch.js"></script>

<!--*****LLAMADO DE LOS CONTROLES*****-->
<!--*****FUENTES*****-->
<script src="js/jquery.fontselect.js"></script>
<!--*****GAUGE1*****-->
<script src = "indicadores/indicador1/indicador1.0.js"></script>
<!--*****BOTONI*****-->
<script src = "controles/boton1/boton1.1.js"></script>
<!--*****LED*****-->
<script src = "indicadores/led/led1.js"></script>
<!--*****C2GAUGE1*****-->
<script src = "controles/gauge1/gauge0.js"></script>
<!--*****LABEL*****-->
<script src = "indicadores/label/label1.js"></script>
<!--*****TREND*****-->
<script src = "indicadores/trend/trend0.js"></script>
<!--*****DISPLAY*****-->
<script src = "indicadores/display/display1.js"></script>
<!--*****POWER BUTTON*****-->
<script src = "controles/pboton/pboton1.js"></script>
<!--*****PERILLA*****-->
<script src = "controles/perilla/perilla1.js"></script>
<!--*****TANQUE1*****-->

```

```

<script src = "controles/tanque1/tanq0.js"></script>
<!--*****SLIDER1*****-->
<script src = "controles/slider/slider.js"></script>
<!--*****SWITCH*****-->
<script src = "controles/switch/switch.js"></script>
<!--*****AJUSTES*****-->
<script src = "controles/ajustes/ajustes.js"></script>
<!--*****CONEXIONES*****-->
>
<script src = "controles/conexiones/conexiones.js"></script>
</head>

<body>
<div class="container1" >

<!-- ICONO DE MENU -->
<i class="fa fa-bar-chart" style="font-size:56px; position: absolute; top:30px; left:50px;
"></i>
<!-- ECABEZADO BARRA AZUL DE MENU PRINCIPAL -->
<header class="clearfix" id="idclearfix">
<span>ANC Platform</span>
<h1>HMI Web Server</h1>
</header>
</div>
<!-- MENU PRINCIPAL -->
<div id="mySidenav" class="main">
<!--COPIA 1 TREND-->

<nav id="menu" class="menuprincipal" >
<ul>

<!--SUB MENU INDICADORES-->
<li id="horizontal"><a href="#" style="font-size:30px; Position: Absolute; Left:50px;
Top:-15px;"><i class="fa fa-cog fa-spin" style="font-size:40px;"></i>Controles</a>

<div id="scroll" class="submenus">

<ul class="controles" >
<!-- GAUGE 1-->
<li><a href="#">GUAGE</a>
<!--CONTENEDOR INDICADOR 1-->
<div id="controlbox1">
<!--INDICADOR 1-->
<div id="gauge0" ondblclick="addi()" class="indicador" ></div>
<div id="gaugeValue" class="c1gauge1" style="font-family: Sans-Serif;
text-align: center; font-size: 17px; width: 70px;"></div>

```

```

<div id="propiedades1" class="propiedades" >
  <ul>
    <!--ENLAZAR TAG-->
    <li>
      <a href="#">CONTROL: </a>
      <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
->
      <span class="xicon" onclick="cerrar1()">X</span>
      <!--NOMBRE DEL CONTROL-->
      <label id="labeltag1" class="labeltag">nombre</label>
    </li>
    <li>
      <!--ESCRIBIR EL TAG A ENLAZAR-->
      <a href="#">SELECCIONAR TAG: </a>
      <select id="variables1" >
      </select>
      <br>
      <button type="button" id="settag1" class="remove"
onclick="tagvalue1()">Conectar</button>
    </li>
    <!--NOMBRE DEL CONTROL-->
    <li> <a href="#">ETIQUETA</a>
      <input class="etiqueta" id="etiqueta1" type="text"
value="SENSOR" >
    </li>
    <!--INTRODUCIR RANGO DEL CONTROL-->
    <li> <a href="#">RANGO</a>
      <label>Min :</label>
      <input class="rango" id="mini1" type="text" value="0"
onkeypress="emini1(event)" >
      <br>
      <label>Max:</label>
      <input class="rango" id="maxi1" type="text" value="100"
onkeypress="emaxi1(event)" >
    </li>
    <!--MODIFICAR TAMAÑO DEL CONTROL-->
    <li> <a href="#">TAMAÑO</a>
      <input type="range" min="170" max="500" value="200"
class="slider" id="slider1" oninput="changesize1()">
      <label id="sizeval1" class="sizeval">:200</label>
    </li>
    <!--REMOVED CONTROL-->
    <li><a href="#">REMOVED CONTROL</a>

```

```

        <button type="button" id="removeri1" class="remove"
onclick="removei1()">Remover</button>
    </li>

</ul>
</div>
</div>
</li>

<!-- LED -->
<li><a href="#">LED</a>
    <div id="controlboxi3">

        <!--COPIA 1 LED-->
        <div id="i3led1" ondblclick="i3add1()" class="led">
        <div id="i3propiedades1" class="propiedades" >
            <ul>
                <!--ENLAZAR TAG-->
                <li>
                    <a href="#">CONTROL</a>
                    <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
->
                    <span class="xicon" onclick="i3cerrar1()">X</span>
                    <label id="i3labeltag1" class"labeltag">nombre:</label>
                    </li>
                    <li>
                        <a href="#">SELECCIONAR TAG: </a>
                        <select id="variablesled" >
                        </select>
                        <br>
                        <button type="button" id="settagled" class="remove"
onclick="comunicacionled()">Conectar</button>
                    </li>

                    <!--COLOR FUENTE-->
                    <li><a href="#">COLOR</a>
                        <input type="color" id="i3color1" class="color"
value="#000000" oninput="i3changeicolor1()">
                    </li>

                    <!--REMOVED CONTROL-->
                    <li><a href="#">REMOVED CONTROL</a>
                        <button type="button" id="i3remove1" class="remove"
onclick="i3remove1()">Remover</button>
                    </li>

                </ul>

```

```

    </div>
  </div>
</div>
</li>

<!-- LABEL -->
<li><a href="#">LABEL</a>
  <div id="controlbox1" >
    <!--COPIA 1 LABEL-->
    <div id="i4move1" class="i4move" ondblclick="i4add1()">
      <label id="i4label1" class="i4label" >LABEL</label>

      <div id="i4propiedades1" class="propiedades" >
        <ul>
          <!--ENLAZAR TAG-->
          <li>
            <a href="#">CONTROL: </a>
            <!--ICONO PARA CERRAR EL CUADRO DE
HERRAMIENTAS-->
            <span class="xicon" onclick="i4cerrar1()">X</span>

            <!--NOMBRE DEL CONTROL-->
            <label id="i4nombrecontrol" class="labeltag">nombre</label>
          </li>
          <li><a href="#">ESTILO</a>

            <select id="i4estilo1" onchange="i4changeestilo1()">
              <option value="Solid">Solid</option>
              <option value="None">Ninguno</option>
              <option value="Dashed">Dashed</option>
              <option value="Dotted">Dotted</option>
              <option value="Double">Double</option>
              <option value="Groove">Groove</option>
              <option value="Ridge">Ridge</option>
              <option value="Inset">Inset</option>
              <option value="Outset">Outset</option>

            </select>

          </li>

          <!--INTRODUCIR CONTENIDO DEL LABEL-->
          <li> <a href="#">CONTENIDO</a>
            <input class="label" id="i4etiqueta1" type="text"
value="LABEL" oninput="i4etiqueta1()">
          </li>

```

```

<!--FUENTE DEL LABEL-->
<li> <a href="#">FUENTE</a>
  <input class="font" id="i4font1" type="text" >
</li>

<!--MODIFICAR TAMAÑO DEL CONTROL-->
<li> <a href="#">TAMAÑO DE LA FUENTE</a>
  <input type="range" min="5" max="100" value="40"
class="slider" id="i4slider1" oninput="i4changesize1()">
  <label id="i4sizeval1" class="sizeval">:40</label>
</li>

<!--COLOR FUENTE-->
<li><a href="#">COLOR DE FUENTE</a>
  <input type="color" id="i4color1" class="color"
value="#000000" oninput="i4changeicolor1()">
</li>

<!--COLOR DE FONDO-->
<li><a href="#">COLOR DE FONDO</a>
  <input type="color" id="i4colorf1" class="color"
value="#ffffff" oninput="i4changecolorf1()">
</li>

<!--REMOVER CONTROL-->
<li><a href="#">REMOVER CONTROL</a>
  <button type="button" id="i4remove1" class="remover"
onclick="i4remove1()">Remover</button>
</li>

</ul>
</div>
</div>
</div>
</li>

<!--BOTON 1 -->
<li><a href="#">BOTTON1</a>
  <!--CONTENEDOR CONTROL 1-->
  <div id="controlboxb1" >

  <!--DIV BOTON1-->
  <div id="c1boton1" class="botonc1" ondblclick="addb1()">
    <div class="switch" id="switch0" >
      <span class="thumb" > </span>
      <input type="checkbox" />

```

```

</div>
</div>

<div id="c1propiedadesb1" class="propiedades" >
  <ul>
    <!--MOSTRAR NOMBRE DEL CONTROL-->
    <li>
      <a href="#">ENLAZAR TAG</a>
      <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
->
      <span class="xicon" onclick="cerrarb1()">X</span>
      <label id="labeltagb1" class="labeltag">nombre</label>
    </li>

    <!--ENLAZAR TAG-->
    <li>
      <a href="#">SELECCIONAR TAG: </a>
      <select id="variablesboton1" >
      </select>
      <br>
      <button type="button" id="settagboton1" class="remove"
onclick="comunicacionb1()">Conectar</button>
    </li>

    <!--COLOR-->
    <li><a href="#">CAMBIAR COLOR</a>
      <input type="color" id="c1colorb1" class="color"
value="#0f8ad7" oninput="b1changeColor1()">
    </li>

    <!--REMOVER CONTROL-->
    <li><a href="#">REMOVER CONTROL</a>
      <button type="button" id="removeb1" class="remove"
onclick="removeb1()">Remover</button>
    </li>
  </ul>
</div>
</div>
</li>

<!-- GAUGE 2-->
<li><a href="#">GAUGE 1</a>
  <div id="c2gauge0" class="controlboxg" >
  <div id="Gauge0" class="c2move" ondblclick="c2add1()">
  </div>

```



```

<div id="c2propiedades1" class="propiedades" >
  <ul>
    <!--ENLAZAR TAG-->
    <li><a href="#">ENLAZAR TAG</a>
      <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
- >
      <span class="xicon" onclick="c2cerrar1()">X</span>
      <label id="c2labeltag1" class"labeltag">nombre</label>
    </li>

    <!--SELECCIONAT TAG-->
    <li>
      <a href="#">SELECCIONAR TAG: </a>
      <select id="variablesg2" >
      </select>
      <br>
      <button type="button" id="settag2" class="remover"
onclick="tagvalueg2()">Conectar</button>
    </li>

    <!--INTRODUCIR RANGO DEL CONTROL-->
    <li> <a href="#">RANGO</a>
      <label>Min :</label>
      <input class="rango" id="c2min1" type="text" value="0">
      <br>
      <label>Max:</label>
      <input class="rango" id="c2max1" type="text" value="100"
onkeypress="c2maximo(event)">
    </li>

    <!--REMOVER CONTROL-->
    <li><a href="#">REMOVER CONTROL</a>
      <button type="button" id="c2remove1" class="remover"
onclick="c2remove1()">Remover</button>
    </li>

  </ul>
</div>
</div>
</li>

<!--TRENDS-->
<li><a href="#">TREND</a>
  <div id="controlboxt" >

  <div id="trend0" class="i6trend" ondblclick="addtrend()"></div>

```

```

<div id="i6propiedades1" class="propiedades" >
  <ul>
    <!--ENLAZAR TAG-->
    <li>
      <a href="#">CONTROL: </a>
      <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
- >
      <span class="xicon" onclick="cerrart1()">X</span>
      <!--NOMBRE DEL CONTROL-->
      <label id="labeltag1" class"labeltag">nombre</label>
    </li>
    <li>
      <!--ESCRIBIR EL TAG A ENLAZAR-->
      <a href="#">SELECCIONAR TAG: </a>
      <select id="variablestrend" >
      </select>
      <br>
      <button type="button" id="settagtrend" class="remove"
onclick="tagvaluetrend()">Conectar</button>
    </li>
    <!--NOMBRE DEL CONTROL-->
    <li> <a href="#">TITULO</a>
      <input class="etiqueta" id="titulotrend" type="text"
value="GRAFICA" onkeypress="titulot(event)">
      <a href="#">DESCRIPCION</a>
      <input class="etiqueta" id="descripciontrend" type="text"
value="" onkeypress="descripciont(event)">
      <a href="#">EJE Y</a>
      <input class="etiqueta" id="ytrend" type="text"
value="" onkeypress="ejeyt(event)">
      <a href="#">EJE X</a>
      <input class="etiqueta" id="xtrend" type="text" value=""
onkeypress="ejext(event)">
    </li>
    <!--INTRODUCIR RANGO DEL CONTROL-->
    <li> <a href="#">RANGO</a>
      <label>Min :</label>
      <input class="rango" id="mintrend" type="text" value="0"
onkeypress="mint(event)" >
      <br>
      <label>Max:</label>
      <input class="rango" id="maxtrend" type="text" value="100"
onkeypress="maxt(event)" >
    </li>
    <!--MODIFICAR TAMAÑO DEL CONTROL-->

```

```

<li> <a href="#">TAMAÑO</a>
  <input type="range" min="200" max="800" value="200"
class="slider" id="slidertrend" oninput="changesize1()">
  <label id="sizeval1" class="sizeval">:200</label>
</li>

<!--REMOVER CONTROL-->
<li><a href="#">REMOVER CONTROL</a>
  <button type="button" id="removet1" class="remove"
onclick="removet1()">Remover</button>
</li>

</ul>

</div>
</div>
</li>

<!--DISPLAY-->
<li><a href="#">DISPLAY</a>
  <div id="controlboxd" >

    <!--COPIA 1 DISPLAY-->
    <div id="display0" class="i7display" style="width=125px;height=150px"
ondblclick="i7add1()"></div>
    <div id="i7propiedades1" class="propiedades" >
      <ul>
        <!--ENLAZAR TAG-->
        <li>
          <a href="#">CONTROL</a>
          <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->
          <span class="xicon" onclick="i7cerrar1()">X</span>
          <label id="i7labeltag1" class="labeltag">nombre:</label>
        </li>

        <li>
          <a href="#">SELECCIONAR TAG: </a>
          <select id="variablesdis" >
          </select>
          <br>
          <button type="button" id="settagdisp" class="remove"
onclick="comunicaciondisplay()">Conectar</button>
        </li>

        <!--Tipo de display-->
        <li><a href="#">MODO</a>

```

```

        <select id="i7type1" onchange="i7changetype1()">
            <option value="fourteenSegment">14Segmentos</option>
            <option value="sevenSegment">7Segmentos</option>
        </select>
    </li>
    <!--DIGITOS-->
    <li><a href="#">NUMERO DE DIGITOS</a>
        <select id="i7digitos1" onchange="i7changedigitos1()">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
            <option value="9">9</option>
        </select>
    </li>
    <!--COLOR FUENTE-->

    <li><a href="#">COLOR DE FONDO</a>
        <input type="color" id="i7colorf1" class="color"
value="#000000" oninput="i7changeicolorf1()">
    </li>
    <!--COLOR DE FONDO-->
    <li><a href="#">COLOR DE NUMERO</a>
        <input type="color" id="i7color1" class="color" value="#ffffff"
oninput="i7changeicolor1()">
    </li>

    <!--REMOVER CONTROL-->
    <li><a href="#">REMOVER CONTROL</a>
        <button type="button" id="i7remove1" class="remover"
onclick="i7remove1()">Remover</button>
    </li>

</ul>
</div>

</div>
</li>

<!--POWER BOTTON-->
<li><a href="#">POWER BUTTON</a>
    <div id="controlboxpb" >

```

```

<!--COPIA 1 POWER BUTTON-->
<div id="powerboton0" class="c3pboton" ondblclick="c3add1()">
  <section class="c3section">
    <a href="#" id="c3pboton0" class="powerb">&#xF011;</a>
    <span class="c3span"></span>
  </section>
</div>

  <div id="c3propiedades1" class="propiedades" >
    <ul>
      <!--NOMBRE CONTROL-->
      <li>
        <a href="#">CONTROL</a>
        <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->
        <span class="xicon" onclick="c3cerrar1()">X</span>
        <label id="c3labeltag1" class="labeltag">nombre:</label>
      </li>
      <!--ENLAZAR TAG-->
      <li>
        <a href="#">SELECCIONAR TAG: </a>
        <select id="variablespb" >
        </select>
        <br>
        <button type="button" id="settagpb" class="remove"
onclick="comunicacionpb()">Conectar</button>
      </li>

      <!--COLOR FUENTE-->
      <li><a href="#">COLOR DE FUENTE</a>
        <input type="color" id="c3color1" class="color"
value="#000000" oninput="c3changeocolor1()">
      </li>

      <!--REMOVER CONTROL-->
      <li><a href="#">REMOVER CONTROL</a>
        <button type="button" id="c3remove1" class="remove"
onlick="c3remove1()">Remover</button>
      </li>

    </ul>
  </div>
</div>
</li>

<!--PERILLA-->
<li><a href="#">PERILLA</a>

```

```

<div id="controlboxper">

  <div id="moveper" class="conper">
    <div id="perilla0" class="c4move" ondblclick="c4add1()">
      <input type="text" class="rotarySwitch2" value=" 0" id="c4perilla1">
    </div>
  </div>

  <div id="c4propiedades1" class="propiedades" >
    <ul>
      <!--ENLAZAR TAG-->
      <li>
        <a href="#">CONTROL</a>
        <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
- >
        <span class="xicon" onclick="c4cerrar1()">X</span>
        <label id="c4labeltag1" class"labeltag">nombre:</label>
        </li>
        <li>
          <a href="#">SELECCIONAR TAG: </a>
          <select id="variablesper" >
          </select>
          <br>
          <button type="button" id="settagper" class="remover"
onclick="comunicacionper()">Conectar</button>
          </li>

          <!--REMOVER CONTROL-->
          <li><a href="#">REMOVER CONTROL</a>
            <button type="button" id="c4remove1" class="remover"
onclick="removeper()">Remover</button>
          </li>

        </ul>
      </div>
    </div>
  </li>

  <!--TANQUE-->
  <li><a href="#">TANQUE</a>
    <div id="controlboxtan">

    <!--TANQUE 0-->
    <div id="c5move1" class="c5move">
      <div id="chart-container" ondblclick="c5add1()"></div>
      <div id="c5propiedades1" class="propiedades" >
        <ul>

```

```

<!--ENLAZAR TAG-->
<li><a href="#">CONTROL:</a>
  <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->
  <span class="xicon" onclick="c5cerrar1()">X</span>
  <label id="c5labeltag1" class="labeltag">nombre</label>
</li>

<li>
  <!--ESCRIBIR EL TAG A ENLAZAR-->
  <a href="#">SELECCIONAR TAG: </a>
  <select id="variablestan" >
  </select>
  <br>
  <button type="button" id="settagtan" class="remover"
onlick="tagvaluetan()">Conectar</button>
</li>

  <!--NOMBRE DEL CONTROL-->
  <li> <a href="#">ETIQUETA</a>
  <input class="etiqueta" id="c5etiquetai1" type="text"
value="tanque" onkeypress="c5nombre1(event)">
</li>

  <!--INTRODUCIR RANGO DEL CONTROL-->
  <li> <a href="#">RANGO</a>

  <label>Max:</label>
  <input class="rango" id="c5max1" type="text" value="100"
oninput="c5range1()">
</li>

  <!--COLOR DE LIQUIDO-->
  <li><a href="#">COLOR</a>
  <input type="color" id="c5color1" class="color"
value="#000000" oninput="c5change1()">
</li>

  <!--REMOVER CONTROL-->
  <li><a href="#">REMOVER CONTROL</a>
  <button type="button" id="c5remove1" class="remover"
onlick="c5remove1()">Remover</button>
</li>

</ul>
</div>
</div>

```

```

</div>
</li>

<!--SLIDER CONTROL-->
<li><a href="#">SLIDER</a>
  <div id="controlboxics1">

    <!--SLIDER COPIA 1-->
    <div id="islider0" class="icslider" ondblclick="icadd1()"></div>
    <div id="icpropiedades1" class="propiedades" >
      <ul>

        <!--ENLAZAR TAG-->
        <li><a href="#">CONTROL:</a>
          <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->

          <span class="xicon" onclick="iccerrar1()">X</span>
          <label id="iclabeltag1" class="labeltag">nombre</label>
        </li>

        <li>
          <!--ESCRIBIR EL TAG A ENLAZAR-->
          <a href="#">SELECCIONAR TAG: </a>
          <select id="variablesld" >
            </select>
          <br>
          <button type="button" id="settagsld" class="remove"
oninput="icrange1()" >
          <button type="button" id="settagsld" class="remove"
oninput="icrange1()">Conectar</button>
        </li>

        <!--INTRODUCIR RANGO DEL CONTROL-->
        <li> <a href="#">RANGO</a>
          <label>Min :</label>
          <input class="rango" id="icmin1" type="text" value="-60"
oninput="icrange1()" >
          <br>
          <label>Max:</label>
          <input class="rango" id="icmax1" type="text" value="100"
oninput="icrange2()">
        </li>

        <!--MODIFICAR TAMAÑO DEL CONTROL-->
        <li> <a href="#">TAMAÑO</a>
          <input type="range" min="160" max="800" value="220"
class="slider" id="icslidert1" oninput="icchangesize1()">
          <label id="icsizeval1" class="sizeval">:220</label>
        </li>

```



```

        <!--REMOVER CONTROL-->
        <li><a href="#">REMOVER CONTROL</a>
          <button type="button" id="icremove1" class="remover"
onclick="icremove1()">Remover</button>
        </li>

      </ul>
    </div>

  </div>
</li>

<!--SWITCH-->
<li><a href="#">SWITCH</a>
  <div id="controlboxsw">
    <!--COPIA 0 SWITCH-->
    <div id="c6move0" class="c6move" ondblclick="c6add1()">

      <span class="switch12">
        <span class="switch-border1">
          <span class="switch-border2">
            <input id="switch0" type="checkbox" checked />
            <label for="switch0"></label>
            <!--SPAN DE LA FORMA-->
            <span class="switch-top"></span>
            <span class="switch-shadow"></span>
            <span class="switch-handle"></span>
            <span class="switch-handle-left"></span>
            <span class="switch-handle-right"></span>
            <span class="switch-handle-top"></span>
            <span class="switch-handle-bottom"></span>
            <span class="switch-handle-base"></span>
            <!--FORMA DE LA LUZ VERDE-->
            <span class="switch-led switch-led-green">
              <span class="switch-led-border">
                <span class="switch-led-light">
                  <span class="switch-led-glow"></span>
                </span>
              </span>
            </span>
            <!--FORMA DE LA LUZ ROJA-->
            <span class="switch-led switch-led-red">
              <span class="switch-led-border">
                <span class="switch-led-light">
                  <span class="switch-led-glow"></span>
                </span>
              </span>
            </span>
          </div>
        </div>
      </li>

```

```

        </span>
    </span>

    </span>
</span>
</span>
</div>

<div id="c6propiedades1" class="propiedades" >
    <ul>
        <!--ENLAZAR TAG-->
        <li><a href="#">CONTROL:</a>
            <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-->
            <span class="xicon" onclick="c6cerrar1()">X</span>
            <label id="c6labeltag1" class="labeltag">nombre</label>
        </li>

        <li>
            <!--ESCRIBIR EL TAG A ENLAZAR-->
            <a href="#">SELECCIONAR TAG: </a>
            <select id="variablesw" >
            </select>
            <br>
            <button type="button" id="settagsw" class="remove"
onlick="comunicacionsw()">Conectar</button>
        </li>

        <!--REMOVER CONTROL-->
        <li><a href="#">REMOVER CONTROL</a>
            <button type="button" id="c6remove1" class="remove"
onlick="c6remove1()">Remove</button>
        </li>
    </ul>
</div>
</div>
</li>

<!--AJUSTES-->
<li><a href="#">AJUSTES</a>
    <div id="apropiedades1" class="propiedades1" >
        <ul>
            <!--TEXTURAS-->
            <li><a href="#">TEXTURAS</a>

            <!--MODO-->

```

```

<select id="atexturas1" onchange="texturas()">
  <option value="textura1">TEXTURA1</option>
  <option value="textura2">TEXTURA2</option>
  <option value="textura3">TEXTURA3</option>
  <option value="textura4">TEXTURA4</option>
  <option value="textura5">TEXTURA5</option>
  <option value="textura6">TEXTURA6</option>
  <option value="textura7">TEXTURA7</option>
  <option value="textura8">TEXTURA8</option>
</select>
</li>
<!--CAMBIAR COLOR DE FONDO-->
<li><a href="#">COLOR DE FONDO</a>
  <input type="color" id="acolor" class="color" value="#000000"
oninput="colorajustes()">
</li>
<!--INSERTAR IMAGEN-->
<li><a href="#">INSERTAR IMAGEN</a>
<div id="imgcontainer">

  <input type="file" name="afile-1[]" id="afile-1" class="inputfile
inputfile-1" onchange="atag1(this)">
  <label for="afile-1"><span>Imagen</span></label>

  <div id="c7propiedades1" class="propiedades" >
  <ul>
    <!--ENLAZAR TAG-->
    <li><a href="#">ENLAZAR TAG</a>
      <!--ICONO PARA CERRAR EL CUADRO DE HERRAMIENTAS-
->

      <span class="xicon" onclick="c7cerrar1()">X</span>
      <input type="file" name="c7file-1[]" id="c7file-1"
class="inputfile inputfile-1" onchange="c7tag1(this)">
      <label for="c7file-1"><span>Enlazar Tag</span></label>
      <br>
      <label id="valor"></label>
      <br>
      <label id="c7labeltag1" class"labeltag">nombre</label>
    </li>

    <!--REMOVER CONTROL-->
    <li><a href="#">REMOVER CONTROL</a>
      <button type="button" id="c7remove1" class="remover"
onclick="c7remove1()">Remover</button>
    </li>

  </ul>

```

```

        </div>
    </div>
</li>
</ul>
</div>
</li>

</ul>
</div>
</li>

<!--EDITAR CONEXIONES-->
<li id="horizontal"><a href="#" style="font-size:30px; Position: Absolute;
Left:250px; Top:-10px;">Conexiones</a>
<div id="scroll" class="submenus">
    <ul class="controles" >
        <li>
            <h1>CONEXIÓN</h1>
            <label>Ingrese la direccion IP del PLC</label>
            <input id="ipaddress" type="text" value="192.168.1.1">
            <br>
            <label>Informacion de la Variable:</label>
            <br>
            <label>Nombre:</label>
            <input id="nombrevariable" type="text">
            <br>
            <label>Direccion:</label>
            <input id="datatype" type="text">
            <br>
            <button type="button" id="ipval" class="remover" onclick="newtag()">Crear
Variable</button>
        </li>

        <li>
        </li>
    </ul>
</div>
</li>

</ul>
</nav>
</div>

<!--CODIGO JAVASCRIPT BOTON ENLAZAR TAG-->
<script src="js/custom-file-input.js"></script>
<!--FUNCION PARA DESPLEGAR EL MENU De CONTROLES-->
<script src="js/cbpHorizontalMenu.min.js"></script>

```

```
<script>
  $(function() {
    cbpHorizontalMenu.init();
  });
</script>
```

```
</body>
</html>
```

ANEXO 2

ESTILOS CSS

- Estilos del encabezado de la interface Web.

```
/* General Demo Style */
@import url(http://fonts.googleapis.com/css?family=Lato:300,400,700);

@font-face {
font-family: 'fontawesome';
src:url('../fonts/fontawesome.eot');
src:url('../fonts/fontawesome.eot?#iefix') format('embedded-opentype'),
url('../fonts/fontawesome.svg#fontawesome') format('svg'),
url('../fonts/fontawesome.woff') format('woff'),
url('../fonts/fontawesome.ttf') format('truetype');
font-weight: normal;
font-style: normal;
}

body, html { font-size: 100%; padding: 0; margin: 0;}

/* Reset */
*,
*:after,
*:before {
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}

.clearfix:before,
.clearfix:after {
content: " ";
display: table;
}

.clearfix:after {
clear: both;
}

body {
font-family: 'Lato', Calibri, Arial, sans-serif;
color: #47a3da;
}

a {
```

```
color: #47a3da;
text-decoration: none;
}
```

```
a:hover {
color: #000;
}
```

```
.container1 > header {
width: 100%;
max-width: 100%;
margin: 30 auto;
padding: 30 auto;
}
```

```
.container1 > header {
padding: 1.5em;
}
```

```
.container1 > header h1 {
font-size: 2.125em;
line-height: 1.3;
margin: 0;
color: #47a3da;
float: left;
font-weight: 400;
padding: 0em 0em 0em 4.3em;
}
```

```
.container1 > header span {
display: block;
font-weight: 700;
text-transform: uppercase;
letter-spacing: 0.2em;
padding: 0em 0em 0em 150px;
}
```

```
.container1 > header nav {
float: right;
}
```

```
.container1 > header nav a {
display: block;
float: left;
position: relative;
width: 2.5em;
```

```
height: 2.5em;
background: #fff;
border-radius: 50%;
color: red;
margin: 0 0.3em;
border: 4px solid white;
text-indent: -8000px;
}
```

```
.container1 > header nav a:after {
content: attr(data-info);
color: #47a3da;
position: absolute;
width: 600%;
top: 120%;
text-align: right;
right: 0;
opacity: 0;
pointer-events: none;
}
```

```
.container1 > header nav a:hover:after {
opacity: 1;
}
```

```
.container1 > header nav a:hover {
background: #47a3da;
}
```

```
.icon-drop:before,
.icon-arrow-left:before {
font-family: 'fontawesome';
position: absolute;
top: 0;
width: 100%;
height: 100%;
speak: none;
font-style: normal;
font-weight: normal;
line-height: 2;
text-align: center;
color: #47a3da;
-webkit-font-smoothing: antialiased;
text-indent: 8000px;
padding-left: 8px;
}
```



```

.container1 > header nav a:hover:before {
color: #fff;
}

.icon-drop:before {
content: "\e000";
}

.icon-arrow-left:before {
content: "\f060";
}

/*-----ICONO DE MENU-----*/
.menuicono {
display: inline-block;
cursor: pointer;
border: 1px white;
position: absolute;
left:40px;
top:20px;
}

.bar1, .bar2, .bar3 {
width: 45px;
height: 5px;
background-color: #47a3da;;
margin: 6px 0;
transition: 0.4s;
}

```

- **Estilo de principal de la interfaz Web**

```

/*PROPIEDADES NAVEGADOR DE LA ETIQUETA NAVEGADOR*/

.main {
position: absolute;
width: 100%;
height: 10000px;
display: inline-block;
}

```

```

    top:150px;
}
/*-----MENU PRINCIPAL-----*/
.menuprincipal {
    position: absolute;
    width: 100%;
    height: 50px;
    top: -50px;
    display: inline-block;
    z-index: 20;
    background: #47a3da;
}

/* general ul style */
/* TAMANO Y COLOR DEL MENU PRINCIPAL*/
.menuprincipal >ul {
    position: absolute;
    list-style-type: none;
    width: 100%;/*TAMANO DEL NAVEGADOR*/
    height: 30px;
}

/*MENU PRINCIPAL*/
/*TIPO DE LETTRA Y SANGRIAS DE UL LI Y A*/
.menuprincipal > ul > li > a {
    font-weight: 700;
    color: #fff;
    display: inline-block;
}

/*-----SUB MENUS-----*/

/*MENU DESPLEGABLE DE CONTROLES*/
.submenus{
    position: absolute;
    overflow-y:auto;
    overflow-x:auto;
    color: black;
    width: 100%;/*TAMANO MENU DE CONTROLES*/
    height: 282px;
    left: 0px;
    top:30px;
    background: #f2f2f2; /*PLOMO CLARO*/
    border-radius: 0px 0px 10px 10px;
}

```

```

    border: 1px solid grey ;
    z-index: 20;
    display: block;
}

/*PROPIEDADES, TAMANO, BACKGROUND DEL CONTENEDOR DE SUBMENUS*/
/* first level ul style */

.submenus >ul {
    padding-top: 0.7em;
    list-style-type: none;
    border-radius: 10px 8px 0px 10px;
    background: #f2f2f2;
    width: 4350px; /*TAMANO DEL NAVEGADOR*/
    display: inline-block;
}

.submenus >ul >li >a: hover
{
    width: 100px;
    color: #fff;
    border-radius: 10px;
    background: #47a3da;
}

/*oculta menu de controles*/
.submenus{
    display: none;
    position: absolute;
    margin-left: 0px;
}

/*despliega menu de controles*/
.cbp-hropen .submenus {
    display: block;
    padding-bottom: 0em;
}

/*-----MENU DE CONTROLES----- */

/*MANTIENE LOS CONTROLES DE FORMA HORIZONTAL Y CREA LAS LINEAS DE
DIVISION*/
.controles > li {

```

```

    display: inline-block;
    width: 330px;
    height: 250px;
    text-align: center;
    padding: 15px 10px 3px -9px;
}

.controles:after {
    clear: both;
}

.controles:before,
.controles:after {
    content: " ";
    display: table;
}

/* PONER LA LINEA DE DIVISION EN LOS CONTROLES */

.controles > li:nth-child(n) {
    float: left;
    box-shadow: 16px 0 0 -15px grey;
}

/*-----SCROLL----- */

#scroll::-webkit-scrollbar {
    width: 10px;
    height: 10px;
}

#scroll::-webkit-scrollbar-thumb {
    border-radius: 12px 100px ;
    background: linear-gradient(left, rgb(73, 125, 189), rgb(73, 125, 189));
    box-shadow: inset 1px 0 10px 10px rgba(1, 125, 189, 0.65);
}

#scroll::-webkit-scrollbar-track {
    border-radius: 12px 100px;
}

#scroll::-webkit-scrollbar-thumb:hover {
    background: linear-gradient(left, #333, #5333);
}

```

```
/* -----Pantalla media----- */
```

```
@media screen and (max-width: 52.75em) {  
  .menuprincipal {  
    font-size: 80%;  
  }  
}
```

```
@media screen and (max-width: 43em) {
```

```
  .menuprincipal {  
    font-size: 120%;  
    border: none;  
  }
```

```
  .menuprincipal > ul > li {  
    display: block;  
    border-bottom: 4px solid #47a3da;  
  }
```

```
  .menuprincipal > ul > li > a {  
    display: block;  
    padding: 1em 3em;  
  }
```

```
  .menuprincipal .cbp-hrsub {  
    position: relative;  
  }
```

```
  .cbp-hrsub h4 {  
    padding-top: 0.6em;  
  }
```

```
}
```

```
@media screen and (max-width: 36em) {  
  .menuprincipal .cbp-hrsub-inner > div {  
    width: 100%;  
    float: none;  
    padding: 0 2em;  
  }  
}
```

- **Estilo del cuadro de propiedades de los controles**

```

/*-----CONTENEDOR DE LOS CONTROLES-----*/
#controlbox1 {
  position: relative;
  width: 330px;
  height: 220px;
}

.indicador0 {
  position: absolute;
  width: 250px;
  height: 250px;
  left: 20%;
  top: 5%;
  z-index: -10;
}

/*-----*/
.indicador {
  position: absolute;
  width: 250px;
  height: 210px;
  left: 20%;
  top: 5%;
}
.indicador:hover {
  cursor: pointer;
}

/*-----Contenedor de las propiedades-----*/
.propiedades {
  display: none;
  position: absolute;
  overflow-y: auto;
  overflow-x: none;
  border-radius: 0px 0px 10px 10px;
  background: #f2f2f2; /*PLOMO CLARO*/
  width: 180px; /*TAMANO DEL NAVEGADOR*/
  height: 255px;
  margin-left: 0px;
  margin-bottom: 2px;
  border: 1px solid grey ;
}

.propiedades>ul {
  list-style-type: none;
  border-radius: 10px 8px 0px 10px;
  background: #f2f2f2;
}

```

```
width: 130px; /*TAMANO DEL NAVEGADOR*/  
height: 245px;  
display: inline-block;  
}
```

```
.propiedades > ul > li > a {  
  font-weight: 700;  
  font-size: 13px;  
  padding: 1em 0em 1em;  
  color: black;  
  display: block;  
}
```

```
.propiedades > ul > li {  
  font-weight: 700;  
  padding: 1em 0em 1em;  
  color: black;  
  display: block;  
  margin-left: -37px;  
  margin-right: -30px;  
  padding: 0.4em;  
}
```

```
.propiedades > ul>li:nth-child(-n+4) {  
  border-bottom: 1px solid grey;  
}
```

```
/*-----SCROLL PROPIEDADES----- */
```

```
.propiedades::-webkit-scrollbar{  
  width:10px;  
  height:-2px;  
}
```

```
.propiedades::-webkit-scrollbar-thumb {  
  border-radius: 100px 12px ;  
  background: linear-gradient(left, rgb(73, 125, 189), rgb(73, 125, 189));  
  box-shadow: inset 1px 0px 10px 10px rgba(1, 125, 189,0.65);  
}
```

```
.propiedades::-webkit-scrollbar-track {  
  border-radius: 12px 100px;  
}
```

```
.propiedades::-webkit-scrollbar-thumb:hover {  
  background: linear-gradient(left, #333, #5333);  
}
```

```
/*-----ICONO DE X-----*/
```

```
.xicon {  
  left: 150px;  
  top: 1px;  
  position: absolute;  
  color: black;  
  font: message-box;  
  text-decoration: none;  
  padding-top: 1px;  
}
```

```
.xicon: hover {  
  background-color: #47a3da;  
  color: #FFF;  
  width: 12px;  
  height: 20px;  
  text-align: left;  
  cursor: pointer;  
  border-radius: 3px;  
}
```

```
/*-----REMOVER CONTROL-----*/
```

```
.remover {  
  margin-left: 27%;  
}
```

```
/*-----Rango-----*/
```

```
.rango {  
  width: 50px;  
}
```

```
/*-----ETIQUETA DEL CONTROL-----*/
```

```
.etiqueta{  
  width: 100px;  
}
```

```
/*-----Slider-----*/
```

```
.slider {  
  -webkit-appearance: none;  
  width: 100px;  
  height: 15px;  
  border-radius: 5px;  
  background: #d3d3d3;  
  outline: none;  
  opacity: 0.7;  
  -webkit-transition: .2s;  
  transition: opacity .2s;
```



```

}

.slider::-webkit-slider-thumb {
  -webkit-appearance: none;
  appearance: none;
  width: 15px;
  height: 20px;
  border-radius: 20%;
  background: #4CAF50;
  cursor: pointer;
}

.slider:hover {
  opacity: 1;
}

.slider::-moz-range-thumb {
  width: 25px;
  height: 25px;
  border-radius: 50%;
  background: #4CAF50;
  cursor: pointer;
}

/*----- inputfile-----*/
.js .inputfile {
  width: 1px;
  height: 1px;
  opacity: 0;
  overflow: hidden;
  position: absolute;
  z-index: -1;
}

.inputfile + label {
  max-width: 90%;
  font-size: 1rem;
  /* 20px */
  font-weight: 700;
  text-overflow: ellipsis;
  white-space: nowrap;
  cursor: pointer;
  display: inline-block;
  overflow: hidden;
  padding: 0.625rem 1.25rem;
  /* 10px 20px */

```

```

}

.no-js .inputfile + label {
    display: none;
}

.inputfile:focus + label,
.inputfile.has-focus + label {
    outline: 1px dotted #000;
    outline: -webkit-focus-ring-color auto 5px;
}

/* style 1 */

.inputfile-1 + label {
    color: #f1e5e6;
    background-color: #d3394c;
}

.inputfile-1:focus + label,
.inputfile-1.has-focus + label,
.inputfile-1 + label:hover {
    background-color: #722040;
}

/* LABEL*/
.label{
    width: 100px;
}

/* FONTS */
.font{
    width: 100px;
}

/* PROPIEDADES LABEL *
#i4propiedades1 > ul>li:nth-child(-n+6) {
    border-bottom: 1px solid grey;}

```

ANEXO 3
CÓDIGO DE LOS INSTRUMENTOS
VIRTUALES

- **GAUGE INDICADOR**

```
var ilmax=100;
var ilmin=0;
var ilsize=230
var congauge=[];
var ilc=0;
var wil;
var hi1;
var gaugeval=[];
var actualgauge=0;
var topic=[];
var socket = io.connect('http://localhost:3700');
var i=1;
//VARIABLES DE LA COMUNICACION
var tagseleccionada;
var indicevar;
var conexiones=[];
var datavalue=[];
var aux;

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("propiedadesi1").style.left=(myWidth-200)+"px";
})

$(document).ready(function () {
    gauge(0);

    aux=null;
    //EVENTO QUE RECIBE LAS VARIABLES
    socket.on('message', function (data) {

        if(data.value!=undefined){
            if(data.value[0]!=aux){
                aux=data.value[0];

                var x=document.getElementById('variables1');
                var z = document.createElement("option");

                z.setAttribute("value", data.value);
```

```

        var t = document.createTextNode(data.value);
        z.appendChild(t);
        x.appendChild(z);
    }
}

});

//DETERMINA EL OBJETO SELECCIONADO
$("#variables1").click(function(event) {
    tagseleccionada=event.target.value;

});

//RESTABLE EL CUADRO DE PROPIEDADES
$("#div").dblclick(function(event) {
    if(event.target.id.substring(0,5)=="gauge"){
        document.getElementById("propiedadesi1").style.display="block";
    }
});

$("#div").click(function(event) {
    if(event.target.id.substring(0,5)=="gauge"){
        document.getElementById("labeltagi1").innerHTML=event.target.id;

        actualgauge=event.target.id;
        //poner en primer plano herramientas del control seleccionado
        document.getElementById("c1propiedadesb1").style.zIndex=1; //boton1
        document.getElementById("propiedadesi1").style.zIndex=20; //gauge1
        document.getElementById("i3propiedades1").style.zIndex=1; //led
        document.getElementById("i4propiedades1").style.zIndex=1; //label
        document.getElementById("c2propiedades1").style.zIndex=1; //gauge2
        document.getElementById("i6propiedades1").style.zIndex=1; //trend
        document.getElementById("i7propiedades1").style.zIndex=1; //display
        document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
        document.getElementById("c4propiedades1").style.zIndex=1; //perilla
        document.getElementById("c5propiedades1").style.zIndex=1; //tanque
        document.getElementById("icpropiedades1").style.zIndex=1; //eslider
        document.getElementById("c6propiedades1").style.zIndex=1; //switch

        $("#"+event.target.id).draggable({
            containment: '#mySidenav',
            mouse: 'move',
            cursor: 'move',
            drag: function (event, ui) {
                var cw = $("#"+event.target.id).jqxGauge('width');

```

```

var ch = $("#"+event.target.id).jqxGauge('height');

var lp=document.getElementById(event.target.id).offsetLeft;
var tp=document.getElementById(event.target.id).offsetTop;

document.getElementById("gaugeval"+event.target.id.substring(5,event.target.id.length)).
style.left=((0.37*cw)+lp)+"px";

document.getElementById("gaugeval"+event.target.id.substring(5,event.target.id.length)).
style.top=(ch*0.74+tp)+"px";
    },
    });
}
});
});

```

//AÑADIR UN NUEVO CONTROL AL AREA DE TRABAJO

```

function add1(){
    ilc++;
    congauge[ilc] = document.createElement("div");
    congauge[ilc].id="gauge"+ilc;

    gaugeval[ilc] = document.createElement("div");
    gaugeval[ilc].id="gaugeval"+ilc;
    gaugeval[ilc].className="c1gauge1";

    $("#mySidenav").append(congauge[ilc]);
    $("#mySidenav").append(gaugeval[ilc]);
    $("#propiedades1").appendTo("#mySidenav");

    //TAMAÑO DEL CONTROL A SER AGREGADO
    //DEFINIR EL TAMAÑO DEL CONTROL A AÑADIRSE
    document.getElementById(congauge[ilc].id).style.width=300+"px";
    document.getElementById(congauge[ilc].id).style.height=300+"px";
    document.getElementById(congauge[ilc].id).style.position="absolute";
    document.getElementById(congauge[ilc].id).style.left=200+"px";
    document.getElementById(congauge[ilc].id).style.top=300+"px";

    document.getElementById(congauge[ilc].id).style.border="10px solid
transparent";
    document.getElementById(congauge[ilc].id).style.borderImage="url(ima1.png)
round";
    document.getElementById(congauge[ilc].id).style.borderImageSlice="55%";

    document.getElementById(gaugeval[ilc].id).style.width=80+"px";
    document.getElementById(gaugeval[ilc].id).style.height=30+"px";

```

```

document.getElementById(gaugeval[i1c].id).style.textAlign="center";

//DEFINIR LAS COORDENADAS DE LA PANTALLA DONDE SE UBICARA EL
CONTROL
var lp=document.getElementById(congauge[i1c].id).offsetLeft;
var tp=document.getElementById(congauge[i1c].id).offsetTop;

document.getElementById(gaugeval[i1c].id).style.left=(lp+0.28*300)+"px";
document.getElementById(gaugeval[i1c].id).style.top=(tp+0.55*300)+"px";

//UBICACIÓN DEL CUADRO DE PROPIEDADES
document.getElementById("propiedadesi1").style.top=(0)+"px";
document.getElementById("propiedadesi1").style.left=(myWidth-200)+"px";
document.getElementById("propiedadesi1").style.display="block";

gauge(i1c);
$("#gauge"+i1c).jqxGauge('value', 40);
}

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-----COMUNICACION-----
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

function tagvaluei1(){

var indice=actualgauge.substring(5,actualgauge.length);
indicevar=indice;
//OBTENER VALOR DE LA VARIABLE A CONECTAR
conexiones[0]=tagseleccionada.substring(0,tagseleccionada.search(", "));
conexiones[1]=indicevar;

//ENVIAR LA VARIABLE QUE VAMOS A MONITOREAR AL CLIENTE
socket.emit('send', conexiones);

//RECIBIR LA VARIABLE DEL CLIENTE OPCUA
socket.on('valvariable', function (data) {
console.log(data.value.length);

console.log(data.value[parseInt(indicevar)-1]);
console.log(indicevar);
datavalue[parseInt(indicevar)]=data.value[parseInt(indicevar)-1];

var tt=0;
for(tt=0;tt<data.value.length;tt++){

var valgauge=data.value[tt];
document.getElementById('gaugeval'+(tt+1)).innerHTML=data.value[tt];
$("#gauge'+(tt+1)).jqxGauge('value', Number(data.value[parseInt(tt)]));

```

```

    }
  });
}

//OCULTAR CUADRO DE CONTROLES
function cerrari1() {
  document.getElementById("propiedades1").style.display="none";
}

//REMOVER CONTROL
function removei1() {
  var indice=actualgauge.substring(5,actualgauge.length);
  $("#gauge"+indice).remove();
  $("#gaugeval"+indice).remove();
}

//Ingresar rango minimo
function emini1(e) {
  var indice=actualgauge.substring(5,actualgauge.length);
  var min=parseInt(document.getElementById('mini1').value);
  $('#gauge'+indice).jqxGauge({ min: min});
}

//Ingresar rango maximo
function emaxi1(e) {
  var indice=actualgauge.substring(5,actualgauge.length);
  if (e.keyCode === 13) {
    var min=parseInt(document.getElementById('mini1').value);
    var max=parseInt(document.getElementById('maxi1').value);
    $('#gauge'+indice).jqxGauge({ max: max });
    $('#gauge'+indice).jqxGauge({

      ranges: [
        { startValue: min, endValue: min+0.25*(max-min), style: { fill: '#4bb648',
stroke: '#4bb648' }, endWidth: 5, startWidth: 1 },
        { startValue: min+0.25*(max-min), endValue: min+0.5*(max-min), style: {
fill: '#fbd109', stroke: '#fbd109' }, endWidth: 10, startWidth: 5 },
        { startValue: min+0.5*(max-min), endValue: min+0.75*(max-min), style: {
fill: '#ff8000', stroke: '#ff8000' }, endWidth: 13, startWidth: 10 },
        { startValue: min+0.75*(max-min), endValue: min+(max-min), style: { fill:
'#e02629', stroke: '#e02629' }, endWidth: 16, startWidth: 13
        }],

      labels:
      {

```

```

        distance: '1px',
        position: 'inside',
        interval: 0.1*(max-min),
        offset: [0, -10],
        visible: true,
        formatValue: function (value) {
            return value;
        }
    },

    ticksMinor: { interval: 0.5*0.1*(max-min), size: '5%' },
    ticksMajor: { interval: 0.1*(max-min), size: '10%' },
    colorScheme: 'scheme01',
    animationDuration: 0,

    });
}

}

//CAMBIAR TAMAÑO DEL CONTROL
function changesizei1() {
    var indice=actualgauge.substring(5,actualgauge.length);
    document.getElementById("labeltagi1").innerHTML=indice+"ind";

    //ALMACENAR EL VALOR DEL SLIDER EN LA VARIABLE S1
    var s1=document.getElementById("slideri1").value;
    //VISUALIZAR EL VALOR DEL SLIDER EN LA ETIQUETA DEL CUADRO DE
PROPIEDADES
    document.getElementById("sizevali1").innerHTML=":"+s1;
    //SELECCIONAR EL CONTROL A TRAVES DE SU ID Y CAMBIAR SU
TAMAÑO
    $('#gauge'+indice).jqxGauge({ height: s1 });
    $('#gauge'+indice).jqxGauge({ width: s1 });
    var cw = $("#gauge"+indice).jqxGauge('width');
    var ch = $("#gauge"+indice).jqxGauge('height');

    var lp=document.getElementById("gauge"+indice).offsetLeft;
    var tp=document.getElementById("gauge"+indice).offsetTop;

    var tcv=parseInt((45/170)*s1);

    document.getElementById("gaugeval"+indice).style.width=tcv+"px";
    document.getElementById("gaugeval"+indice).style.top=(ch*0.74+tp)+"px";
    document.getElementById("gaugeval"+indice).style.left=(0.4*(cw)+lp)+"px";
}

//CODIGO DEL CONTROL

```



```

function gauge(i1c){

    $('#gauge'+i1c).jqxGauge({ min: i1min });
    $('#gauge'+i1c).jqxGauge({ max: i1max });
    $('#gauge'+i1c).jqxGauge({

        ranges: [{
            startValue: i1min,
            endValue: 0.25*i1max,
            style: { fill: '#4bb648', stroke: '#4bb648' }, endWidth: 5, startWidth: 1 },
            { startValue: 0.25*i1max, endValue: 0.5*i1max, style: { fill: '#fbd109',
stroke: '#fbd109' }, endWidth: 10, startWidth: 5 },
            { startValue: 0.5*i1max, endValue: 0.75*i1max, style: { fill: '#ff8000', troke:
'#ff8000' }, endWidth: 13, startWidth: 10 },
            { startValue: 0.75*i1max, endValue: i1max, style: { fill: '#e02629', stroke:
'#e02629' }, endWidth: 16, startWidth: 13
            }],

        labels:
        {
            distance: '1px',
            position: 'inside',
            interval: 0.1*i1max,
            offset: [0, -10],
            visible: true,
            formatValue: function (value) {
                return value;
            }
        },

        ticksMinor: { interval: 0.5*0.1*i1max, size: '5%' },
        ticksMajor: { interval: 0.1*i1max, size: '10%' },
        colorScheme: 'scheme01',
        animationDuration: 0,

    });

    $('#gauge'+i1c).jqxGauge({ height: i1size });
    $('#gauge'+i1c).jqxGauge({ width: i1size });

    $('#gauge'+i1c).on('valueChanging', function (e) {
        $('#gaugeValue').text(Math.round(e.args.value));
    });
}

```

- **LED**

```

var tag="LED";
var c1=0;
var lp,tp,hs,nt;
var i3value=0;
var color="#8d003b";
var nt1;
var vl;
var l1color1="#ABFF00";
var l1color2="#304701";
var l1color3="#89FF00";
var conled;
var valled;
var myWidth;
var actualled;
//VARIABLES DE LA COMUNICACION
var auxled;
var tagged;
var conexionesled=[];
var dataled=[];
var colorled=[];

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("i3propiedades1").style.left=(myWidth-200)+"px";
})

//CAMBIAR COLOR
function i3changecolor1(){
    var inputled = document.querySelector('#i3color1')
    colorled[actualled]=inputled.value;

    document.querySelector('#led'+actualled).style.setProperty('--i3led1-
color',colorled[actualled]);
    document.querySelector('#led'+actualled).style.setProperty('--i3led2-color',l1color2);
    document.querySelector('#led'+actualled).style.setProperty('--i3led3-
color',colorled[actualled]);
}

//DESPARECER CUANDRO DE CONTROLES
function i3cerrar1() {
    document.getElementById("i3propiedades1").style.display="none";
}

//REMOVED CONTROL
function i3remove1() {
    $("#led"+actualled).remove();

```

```

}

//ANADIR CONTROL AL AREA DE TRABAJO
function i3add1() {
  c11++;

  conled = document.createElement("div");
  conled.id="led"+c11;
  conled.className="led";

  //CREAR EL LABEL
  valled = document.createElement("p");
  valled.id="ledval"+c11;
  valled.className="lednombre";

  //ANADIR EL LABEL
  conled.appendChild(valled);
  $("#mySidenav").append(conled);
  $("#i3propiedades1").appendTo("#mySidenav");

  //DEFINIR PROPIEDADES DEL LABEL
  document.getElementById(conled.id).style.position="absolute";
  document.getElementById(conled.id).style.left=200+"px";
  document.getElementById(conled.id).style.top=300+"px";

  /*-----OBTENER POSICION DEL OBJETO-----*/
  document.getElementById("i3propiedades1").style.top=(0)+"px";
  document.getElementById("i3propiedades1").style.left=(myWidth-200)+"px";
  document.getElementById("i3propiedades1").style.display="block";
  /*-----*/
}

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-----COMUNICACION-----
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
function comunicacionled() {

  var indice=actualled;
  //OBTENER VALOR DE LA VARIABLE A CONECTAR
  document.getElementById("i3labeltag1").innerHTML="Led"+actualled;
  tagled=document.getElementById("variablesled").value;

  conexionesled[0]=tagled.substring(0,tagled.search(","));
  conexionesled[1]=indice;

  //ENVIAR LA VARIABLE QUE VAMOS A MONITOREAR AL CLIENTE
  socket.emit('sendled', conexionesled);
}

```

```

//RECIBIR LA VARIABLE DEL CLIENTE OPCUA
socket.on('valuesled', function (data) {
    console.log(data.value.length);

    console.log(data.value[parseInt(indice)-1]);
    console.log(indice);

    dataled[parseInt(indice)]=data.value[parseInt(indice)-1];

    var tt=0;
    for(tt=0;tt<data.value.length;tt++){

        var val=data.value[tt];
        console.log(val);
        if (val=="0"){
            document.querySelector('#led'+(tt+1)).style.setProperty('--i3led1-
color', "#b3b3b3");
            document.querySelector('#led'+(tt+1)).style.setProperty('--i3led2-
color', "#8c8c8c");
            document.querySelector('#led'+(tt+1)).style.setProperty('--i3led3-
color', "#8c8c8c");
        }else{
            //console.log(colorled[actualled]);
            if(colorled[tt+1]!==undefined){
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led1-
color',colorled[tt+1]);
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led2-
color',l1color2);
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led3-
color',colorled[tt+1]);
            }else{
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led1-
color', "#ABFF00");
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led2-
color', "#304701");
                document.querySelector('#led'+(tt+1)).style.setProperty('--i3led3-
color', "#89FF00");
            }
        }
    }
});

```

```

}
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

//EVENTO QUE SE DISPARA AL INICIO DE TODO EL PROCESO
$(document).ready(function() {
    $("div").dblclick(function(event) {
        if(event.target.id.substring(0,3)=="led"){
            document.getElementById("i3propiedades1").style.display="block";
        }
    });
//ESCUCHA LAS VARIABLES DEL SERVIDOR
socket.on('message', function (data) {

    if(data.value!=undefined){
        if(data.value[0]!=auxled){
            auxled=data.value[0];

            var x=document.getElementById('variablesled');
            var z = document.createElement("option");

            z.setAttribute("value", data.value);

            var t = document.createTextNode(data.value);
            z.appendChild(t);
            x.appendChild(z);
        }
    }
});

//DETERMINA INDICE DEL LED ESCOGIDO
$("div").click(function(event) {
    if(event.target.id.substring(0,3)=="led"){

        actualled=event.target.id.substring(3,event.target.id.length);
        document.getElementById("i3labeltag1").innerHTML="Led"+actualled;
        //poner en primer plano herramientas del control seleccionado
        document.getElementById("c1propiedadesb1").style.zIndex=1; //boton1
        document.getElementById("propiedadesi1").style.zIndex=1; //gauge1
        document.getElementById("i3propiedades1").style.zIndex=20; //led
        document.getElementById("i4propiedades1").style.zIndex=1; //label
        document.getElementById("c2propiedades1").style.zIndex=1; //gauge2
        document.getElementById("i6propiedades1").style.zIndex=1; //trend
        document.getElementById("i7propiedades1").style.zIndex=1; //display
        document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
        document.getElementById("c4propiedades1").style.zIndex=1; //perilla
        document.getElementById("c5propiedades1").style.zIndex=1; //tanque
    }
});

```

```
document.getElementById("icpropiedades1").style.zIndex=1; //eslider
document.getElementById("c6propiedades1").style.zIndex=1; //switch
```

```
//PERMITE MOVER LA IMAGEN
$("#led"+actualled).resizable();
$("#led"+actualled).draggable({
    containment:'#mySidenav',
    mouse:'move',
    cursor:'move',
});
}
});
});
```

- **BOTON**

```
var cb1=0;
var checkbox;
var conboton;
var switchval;
var checkboxval;
var myWidth;
var actualboton;
var index;
//VARIABLES DE LA COMUNICACION
var auxb1;
var indicevarb1;
var tagseleccionadab1;
var conexionesb1=[];
var VDIRb1=[];

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("c1propiedadesb1").style.left=(myWidth-200)+"px";
})
```

```
auxb1=null;
//se reciben variables del servidor
socket.on('message', function (data) {
    if(data.value!=undefined){
        if(data.value[0]!=auxb1){
            auxb1=data.value[0];
            var x=document.getElementById('variablesboton1');
            var z = document.createElement("option");
            //z.id = "var"+ii;
            z.setAttribute("value", data.value);
```

```

        var t = document.createTextNode(data.value);
        z.appendChild(t);
        x.appendChild(z);
    }
}
});

//Hacer visible cuadro de propiedades
$(document).ready(function() {
    $("div").dblclick(function(event) {
        if (event.target.id.substring(0,5)=="boton"){
            document.getElementById("c1propiedadesb1").style.display="block";
        }
    });

//se determina el evento seleccionado
    $("div").click(function(event) {
        if (event.target.id.substring(0,9)=="botonspan"){

            actualboton=event.target.id.substring(9,event.target.id.length);
            //escribir el nombre del boton seleccionado
            document.getElementById("labeltagb1").innerHTML="boton"+actualboton;

//poner en primer plano herramientas del control seleccionado
            document.getElementById("c1propiedadesb1").style.zIndex=20; //boton1
            document.getElementById("propiedadesi1").style.zIndex=1; //gauge1
            document.getElementById("i3propiedades1").style.zIndex=1; //led
            document.getElementById("i4propiedades1").style.zIndex=1; //label
            document.getElementById("c2propiedades1").style.zIndex=1; //gauge2
            document.getElementById("i6propiedades1").style.zIndex=1; //trend
            document.getElementById("i7propiedades1").style.zIndex=1; //display
            document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
            document.getElementById("c4propiedades1").style.zIndex=1; //perilla
            document.getElementById("c5propiedades1").style.zIndex=1; //tanque
            document.getElementById("icpropiedades1").style.zIndex=1; //eslider
            document.getElementById("c6propiedades1").style.zIndex=1; //switch

//PERMITE MOVER LA IMAGEN
            $("#boton"+actualboton).draggable({
                containment:'#mySidenav',
                mouse:'move',
                cursor:'move',
            });
        }
    });
});

```

```

(function() {
    var $$ = function(selector) {
        return
Array.prototype.slice.call(document.querySelectorAll(selector));
    }

    document.addEventListener("click", function(event) {
        if( event.target.id.substring(0,9)=="botonspan"){

        $$("#boton"+actualboton).forEach(function(switchControl) {

            if (switchControl.className === ("switch on")) {
                switchControl.lastElementChild.checked
= true;
            }
            else
            {
                switchControl.lastElementChild.checked = false;
            }

            if (switchControl.className === "switch on") {
                switchControl.className = 'switch off';
            } else {
                switchControl.className = ("switch on");
            }

            checkbox = switchControl.lastElementChild;

document.getElementById("labeltagb1").innerHTML="boton"+actualboton+"="+(checkbox
checked);

//OBTENER VALOR DE LA VARIABLE A CONECTAR
conexionesb1[1]=!(checkbox.checked);
conexionesb1[0]=VDIRb1[actualboton];

if(conexionesb1[0]!==undefined){
    socket.emit('sendg2', conexionesb1);
}

});
}
    }, false);
})();

//FUNCION QUE LEER LOS DATOS PROVENIENTES DEL PLC

```



```

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-----
COMUNICACION----
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
function comunicacionb1(){
    tagseleccionadab1=document.getElementById("variablesboton1").value;

    VDIRb1[actualboton]=tagseleccionadab1.substring(tagseleccionadab1.search(",")+1,tagseleccionadab1.length);
}

//DESPARECER CUANDRO DE CONTROLES
function cerrarb1() {
    document.getElementById("c1propiedadesb1").style.display="none";
}

//REMOVER CONTROL
function removeb1() {
    $("#boton"+actualboton).remove();
}

function b1changeColor1() {
    //CAMBIAR EL COLOR DEL CONTROL
    var input = document.querySelector("#c1colorb1");
    document.querySelector("#boton"+actualboton).style.setProperty('--led-color',input.value);
}

//ANADIR CONTROL AL AREA DE TRABAJO
function addb1() {
    cb1++;

    //CREAR EL BOTON
    conboton = document.createElement("div");
    conboton.id="boton"+cb1;
    conboton.className="switch";

    switchval = document.createElement("span");
    switchval.id="botonspan"+cb1;
    switchval.className="thumb";

    checkboxval = document.createElement("input");
    checkboxval.setAttribute("type","checkbox");
    checkboxval.id="checkbox"+cb1;
    checkboxval.className="";

    //ANADIR EL LABEL
    conboton.appendChild(switchval);
}

```

```

conboton.appendChild(checkboxval);

$("#mySidenav").append(conboton);
$("#c1propiedadesb1").appendTo("#mySidenav");

//DEFINIR PROPIEDADES DEL LABEL
document.getElementById(conboton.id).style.left=200+"px";
document.getElementById(conboton.id).style.top=300+"px";

/*-----OBTENER POSISION DEL OBJETO-----*/
document.getElementById("c1propiedadesb1").style.top=(0)+"px";
document.getElementById("c1propiedadesb1").style.left=(myWidth-200)+"px";
document.getElementById("c1propiedadesb1").style.display="block";

}

```

- **GAUGE CONTROL**

```

var c2max1=100;
var c2min1=0;
var c2value1=c2min1;
var c2width=180;
var c2height=180;
var cg2=0;
var congauge2;
var gauge2;
var gauge2val;
var input=[];
var actualgaugec;
//VARIABLES DE LA COMUNICACIONES
var auxg2;
var indicevarg2;
var tagseleccionadag2;
var conexionesg2=[];
var VDIR=[];

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("c2propiedades1").style.left=(myWidth-200)+"px";
})

$(document).ready(function () {
    drawgauge(0);
    auxg2=null;
    //se reciben variables del servidor

```

```

socket.on('message', function (data) {

    if(data.value!=undefined){
        if(data.value[0]!=auxg2){
            auxg2=data.value[0];
            var x=document.getElementById('variablesg2');
            var z = document.createElement("option");
            //z.id = "var"+ii;
            z.setAttribute("value", data.value);

            var t = document.createTextNode(data.value);
            z.appendChild(t);
            x.appendChild(z);
        }
    }
});

//DETERMINA EL OBJETO SELECCIONADO
$("#variablesg2").click(function(event) {
    tagseleccionadag2=event.target.value;
});

//Restablecer el cuadro de propiedades
$("#div").dblclick(function(event) {
    console.log(event.target.id);
    if(event.target.id.substring(0,7)=="c2gauge"){
        document.getElementById("c2propiedades1").style.display="block";
    }
});

//Funcion que determina que el objeto sleccionado
$("#div").click(function(event) {
    if(event.target.id.substring(0,7)=="c2gauge"){
        actualgaugec=event.target.id.substring(7,event.target.id.length);
        document.getElementById("c2labeltag1").innerHTML="Gauge"+actualgaugec;

        //poner en primer plano herramientas del control seleccionado
        document.getElementById("c1propiedadesb1").style.zIndex=1; //boton1
        document.getElementById("propiedadesi1").style.zIndex=1; //gauge1
        document.getElementById("i3propiedades1").style.zIndex=1; //led
        document.getElementById("i4propiedades1").style.zIndex=1; //label
        document.getElementById("c2propiedades1").style.zIndex=20; //gauge2
        document.getElementById("i6propiedades1").style.zIndex=1; //trend
        document.getElementById("i7propiedades1").style.zIndex=1; //display
        document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
        document.getElementById("c4propiedades1").style.zIndex=1; //perilla
        document.getElementById("c5propiedades1").style.zIndex=1; //tanque
    }
});

```

```

document.getElementById("icpropiedades1").style.zIndex=1; //eslider
document.getElementById("c6propiedades1").style.zIndex=1; //switch

conexionesg2[0]=VDIR[actualgaugec];

//PERMITE MOVER LA IMAGEN
$("#c2gauge"+actualgaugec).draggable({
    containment:'#mySidenav',
    mouse:'move',
    cursor:'move',
});
}
});
});

function c2remove1(){
    $("#c2gauge"+actualgaugec).remove();
}

function c2cerrar1(){
    document.getElementById("c2propiedades1").style.display="none";
}

//Ingresar rango maximo
function c2maximo(e) {
    if (e.keyCode === 13) {
        var min=parseInt(document.getElementById('c2min1').value);
        var max=parseInt(document.getElementById('c2max1').value);

        $('#Gauge'+actualgaugec).jqxKnob({ max: max });
        $('#Gauge'+cg2).jqxKnob({
            value: c2value1,
            min: min,
            max: max,
            width:300,
            height:300,
            startAngle: 120,
            endAngle: 420,
            snapToStep: true,
            rotation: 'clockwise',
            style: { stroke: '#dfe3e9', strokeWidth: 3, fill: { color: '#fafafa', } },
            marks: {
                colorRemaining: { color: 'grey', border: 'grey' },
                colorProgress: { color: '#00a4e1', border: '#00a4e1' },
                type: 'line',
                offset: '71%',
                thickness: 1,
            }
        });
    }
}

```

```

        size: '6%',
        majorSize: '9%',
        majorInterval: 0.1*(max-min),
        minorInterval: 0.01*(max-min)
    },
    spinner: {
        style: { fill: { color: 'grey'}, stroke: 'white' },
        innerRadius: '45%', // specifies the inner Radius of the dial
        outerRadius: '60%', // specifies the outer Radius of the dial
        marks: {
            colorRemaining: '#fff',
            colorProgress: '#fff',
            type: 'line',
            offset: '46%',
            thickness: 2,
            size: '14%',
            majorSize: '14%',
            majorInterval: 0.1*(max-min),
            minorInterval: 0.1*(max-min)
        },
    },
    labels: {
        offset: '88%',
        step: 0.1*(max-min),
        visible: true
    },
    progressBar: {
        style: { fill: '#00a4e1', stroke: 'grey' },
        size: '9%',
        offset: '60%',
        background: { fill: 'white', stroke: 'grey' }
    },
    pointer: { type: 'arrow', style: { fill: '#00a4e1', stroke: 'grey' }, size: '59%', offset:
'49%', thickness: 20 }
    });
}
}

```

```

function c2add1(){
    cg2++;
    //CREAR EL CONTENEDOR DE PERILLA
    congaugue2 = document.createElement("div");
    congaugue2.id="c2gauge"+cg2;
    congaugue2.className="controlboxg";

    gauge2 = document.createElement("div");

```

```

gauge2.id="Gauge"+cg2;
gauge2.className="c2move";

gauge2val = document.createElement("div");
gauge2val.id="gauge2val"+cg2;
gauge2val.className="inputField";

//ANADIR EL LA PERILLA
congauge2.appendChild(gauge2);
congauge2.appendChild(gauge2val);

$("#mySidenav").append(congauge2);
$("#c2propiedades1").appendTo("#mySidenav");

//TAMANO DEL CONTROL A SER AGREGADO
document.getElementById(congauge2.id).style.position="absolute";
document.getElementById(congauge2.id).style.width=312+"px";
document.getElementById(congauge2.id).style.height=250+"px";
document.getElementById(congauge2.id).style.left=250+"px";
document.getElementById(congauge2.id).style.top=300+"px";

//DAR FORMATO AL BORDE
document.getElementById(congauge2.id).style.border="10px solid transparent";
document.getElementById(congauge2.id).style.borderImage="url(ima1.png)
round";
document.getElementById(congauge2.id).style.borderImageSlice="55%";

//UBICACION DE PROPIEDADES
document.getElementById("c2propiedades1").style.top=(0)+"px";
document.getElementById("c2propiedades1").style.left=(myWidth-200)+"px";
document.getElementById("c2propiedades1").style.display="block";

drawgauge(cg2);
}

//-----COMUNICACION-----
function tagvalueg2(){

//OBTENER VALOR DE LA VARIABLE A CONECTAR
document.getElementById("c2labeltag1").innerHTML="Gauge"+actualgaugec;

VDIR[actualgaugec]=tagseleccionadag2.substring(tagseleccionadag2.search(",")+1,tagseleccionadag2.length);
conexionesg2[0]=VDIR[actualgaugec];

$('#Gauge'+actualgaugec).on('change', function (event) {

```

```

    if (event.args.changeSource == 'propertyChange' || event.args.changeSource ==
'val') { return; }
    conexionesg2[1]=event.args.value;
    socket.emit('sendg2', conexionesg2);
    //console.log("VEC= "+conexionesg2);
  });
}

```

```

function drawgauge(cg2){
  $('#Gauge'+cg2).jqxKnob({
    value: c2value1,
    min: c2min1,
    max: c2max1,
    width:c2width,
    height:c2height,
    startAngle: 120,
    endAngle: 420,
    snapToStep: true,
    rotation: 'clockwise',
    style: { stroke: '#dfe3e9', strokeWidth: 3, fill: { color: '#fafafa', } },
    marks: {
      colorRemaining: { color: 'grey', border: 'grey' },
      colorProgress: { color: '#00a4e1', border: '#00a4e1' },
      type: 'line',
      offset: '71%',
      thickness: 1,
      size: '6%',
      majorSize: '9%',
      majorInterval: c2max1/10,
      minorInterval: c2max1/100
    },
    spinner: {
      style: { fill: { color: 'grey'}, stroke: 'white' },
      innerRadius: '45%', // specifies the inner Radius of the dial
      outerRadius: '60%', // specifies the outer Radius of the dial
      marks: {
        colorRemaining: '#fff',
        colorProgress: '#fff',
        type: 'line',
        offset: '46%',
        thickness: 2,
        size: '14%',
        majorSize: '14%',
        majorInterval: c2max1/10,
        minorInterval: c2max1/10
      },
    },
  });
}

```

```

    },
    labels: {
        offset: '88%',
        step: c2max1/10,
        visible: true
    },
    progressBar: {
        style: { fill: '#00a4e1', stroke: 'grey' },
        size: '9%',
        offset: '60%',
        background: { fill: 'white', stroke: 'grey' }
    },
    pointer: { type: 'arrow', style: { fill: '#00a4e1', stroke: 'grey' }, size: '59%', offset:
'49%', thickness: 20 }
});

```

```

input[cg2] = $('<div class="inputField">');
$('#c2gauge'+cg2).append(input[cg2]);

```

```

var inputOptions = {
    width: 55,
    height: 25,
    decimal: 0, // starting value same as widget
    textAlign: 'center',
    decimalDigits: 0,
    inputMode: 'simple'
};

```

```

$(input[cg2]).jqxNumberInput(inputOptions);

```

```

$(input[cg2]).on('mousedown', function(event){
    event.stopPropagation();
});

```

```

$(input[cg2]).on('keyup', function () {
    var val = $(this).val();
    $('#Gauge'+cg2).val(val);
});

```

```

$(input).on('change', function () {
    var val = $(this).val();
    $('#Gauge'+cg2).val(val);
});

```

```

$('#Gauge'+cg2).on('change', function (event) {

```



```

        if (event.args.changeSource == 'propertyChange' || event.args.changeSource ==
'val') { return; }
        $(input[cg2]).val(event.args.value);
    })
}

```

- **TREND**

```

var tc1=0;
var contrend=[];
var actualtrend;
var settings=[];
var sampleData=[]
//VARIABLES DE LA COMUNICACION
var auxtrend;
var tagtrend;
var conexionestrend=[];
var datatrend=[];
var itrend=0;

```

```

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("i6propiedades1").style.left=(myWidth-200)+"px";
});

```

```

$(document).ready(function () {
    grafica(0);

```

```

    auxtrend=null;

```

```

//EVENTO QUE RECIBE LAS VARIABLES

```

```

    socket.on('message', function (data) {

```

```

        if(data.value!=undefined){
            if(data.value[0]!=auxtrend){
                auxtrend=data.value[0];

```

```

                var x=document.getElementById('variablestrend');
                var z = document.createElement("option");

```

```

                z.setAttribute("value", data.value);

```

```

                var t = document.createTextNode(data.value);
                z.appendChild(t);
                x.appendChild(z);

```

```

    }
  }
});

//RESTABLE EL CUADRO DE PROPIEDADES
$("div").dblclick(function(event) {
  if(event.target.id.substring(0,5)=="trend"){
    document.getElementById("i6propiedades1").style.display="block";
  }
});

//DETERMINA EL ID DEL OBJETO
$("div").click(function(event) {
  if(event.target.id.substring(0,5)=="trend"){
    document.getElementById("labeltag1").innerHTML=event.target.id;

    actualtrend=event.target.id;
    //poner en primer plano herramientas del control seleccionado
    document.getElementById("i6propiedades1").style.zIndex=1; //boton1
    document.getElementById("propiedadesi1").style.zIndex=1; //gauge1
    document.getElementById("i3propiedades1").style.zIndex=1; //led
    document.getElementById("i4propiedades1").style.zIndex=1; //label
    document.getElementById("c2propiedades1").style.zIndex=1; //gauge2
    document.getElementById("i6propiedades1").style.zIndex=20; //trend
    document.getElementById("i7propiedades1").style.zIndex=1; //display
    document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
    document.getElementById("c4propiedades1").style.zIndex=1; //perilla
    document.getElementById("c5propiedades1").style.zIndex=1; //tanque
    document.getElementById("icpropiedades1").style.zIndex=1; //eslider
    document.getElementById("c6propiedades1").style.zIndex=1; //switch

    $("#"+event.target.id).draggable({
      containment:'#mySidenav',
      mouse:'move',
      cursor:'move',
    });
  }
});
});

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-----
COMUNICACION-----
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

```

function tagvaluetrend(){
    var indice=actualtrend.substring(5,actualtrend.length);
    //OBTENER VALOR DE LA VARIABLE A CONECTAR
    tagtrend=document.getElementById("variablestrend").value;

    conexionestrend[0]=tagtrend.substring(0,tagtrend.search(","));
    conexionestrend[1]=indice;

    //ENVIAR LA VARIABLE QUE VAMOS A MONITOREAR AL CLIENTE
    socket.emit('sendtrend', conexionestrend);

    //RECIBIR LA VARIABLE DEL CLIENTE OPCUA
    socket.on('valuestrend', function (data) {
        //console.log(data.value.length);

        //console.log(data.value[parseInt(indicetan)-1]);
        //console.log(indicetan);
        //datatan[parseInt(indicetan)]=data.value[parseInt(indicetan)-1];
        var valtrend=[];
        var k3=0;
        for(k3=0;k3<10;k3++){
            valtrend[k3]=50;
            console.log(valtrend);
        }

        sampleData[itrend]=data.value[0];
        itrend++;

        $('#trend'+indice).jqxChart('refresh');
    });
}

```

```

function grafica(contrend){
    var it=0
    for(it=0;it<5;it++)
    {
        sampleData[it]=100*Math.random();
    }
    sampleData [contrend]= sampleData;
    console.log(sampleData);
    settings[contrend] = {
        title: "Titulo",
        Description:"Descripcion de la grafica",
        showLegend: true,
        padding: { left: 5, top: 5, right: 15, bottom: 5 },
        titlePadding: { left: 0, top: 0, right: 0, bottom: 10 },
        source: sampleData[contrend],
    }
}

```

```

xAxis:
{
  text: 'x',
  valuesOnTicks: true
},
colorScheme: 'scheme01',
seriesGroups:
[
  {
    type: 'line',
    source: sampleData[contrend],
    toolTipFormatFunction: function (value, itemIndex, serie, group,
categoryValue, categoryAxis) {
      var dataItem = sampleData[itemIndex];
      return '<DIV style="text-align:left"><b>Index:</b> ' +
        itemIndex + '<br /><b>Value:</b> ' +
        value + '<br /></DIV>';
    },
    valueAxis:
    {
      minValue: 0,
      maxValue: 100,
      title: { text: 'Yvalue<br>' }
    },
    series:
    [
      { emptyPointsDisplay: 'skip', displayText: 'XValue', lineWidth: 2,
symbolSize: 8}
    ]
  }
]
}
// create the chart
$("#trend"+contrend).jqxChart(settings[contrend]);
}

```

```

function addtrend(){
  tc1++;
  contrend[tc1] = document.createElement("div");
  contrend[tc1].id="trend"+tc1;

  $("#mySidenav").append(contrend[tc1]);
  $("#i6propiedades1").appendTo("#mySidenav");

  document.getElementById("labeltag1").innerHTML="trend"+tc1;
  //TAMANO DEL CONTROL A SER AGREGADO
}

```

```

document.getElementById(contrend[tc1].id).style.width=400+"px";
document.getElementById(contrend[tc1].id).style.height=300+"px";
document.getElementById(contrend[tc1].id).style.position="absolute";
document.getElementById(contrend[tc1].id).style.left=200+"px";
document.getElementById(contrend[tc1].id).style.top=100+"px";

document.getElementById(contrend[tc1].id).style.border="10px solid blue";
document.getElementById(contrend[tc1].id).style.borderImage="url(ima1.png)
round";
document.getElementById(contrend[tc1].id).style.borderImageSlice="55%";

//ubicacion recuadro del valor
var lp=document.getElementById(contrend[tc1].id).offsetLeft;
var tp=document.getElementById(contrend[tc1].id).offsetTop;

document.getElementById(contrend[tc1].id).style.left=(lp+0.28*300)+"px";
document.getElementById(contrend[tc1].id).style.top=(tp+0.55*300)+"px";

//UBICACION DE PROPIEDADES
document.getElementById("i6propiedades1").style.top=(0)+"px";
document.getElementById("i6propiedades1").style.left=(myWidth-200)+"px";
document.getElementById("i6propiedades1").style.display="block";

grafica(tc1);
}

//OCULTAR CUADRO DE CONTROLES
function cerrart1() {
    document.getElementById("i6propiedades1").style.display="none";
}

//REMOVER CONTROL
function removet1() {
    var indice=actualtrend.substring(5,actualtrend.length);
    $("#"+indice).remove();
}

//Ingresar rango minimo
function emini1(e) {
    var indice=actualtrend.substring(5,actualtrend.length);
    var min=parseInt(document.getElementById('mintrend').value);
}

//CAMBIA TAMANO

```

```

function changesize1() {
    var indice=actualtrend.substring(5,actualtrend.length);
    //ALMACENAR EL VALOR DEL SLIDER EN LA VARIABLE S1
    var s1=document.getElementById("slidertrend").value;
    //VISUALIZAR EL VALOR DEL SLIDER EN LA ETIQUETA DEL CUADRO DE
PROPIEDADES
    document.getElementById("sizevalt1").innerHTML=":"+s1;

    //SELECCIONAR EL CONTROL A TRAVES DE SU ID Y CAMBIAR SU
TAMAÑO
    document.getElementById("trend"+indice).style.width=s1*1.25+"px";
    document.getElementById("trend"+indice).style.height=(s1)+"px";

    $("#trend"+indice).jqxChart('refresh');
    $("#trend"+indice).jqxChart('update');

}
//Cambiar rango minimo
function mint(e){
    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var min=parseInt(document.getElementById('mintrend').value);
        var max=parseInt(document.getElementById('maxtrend').value);
        settings[indice].seriesGroups[0].valueAxis.minValue=min;
        settings[indice].seriesGroups[0].valueAxis.maxValue=max;
        $("#trend"+indice).jqxChart('refresh');
        $("#trend"+indice).jqxChart('update');
    }
}
//Cambiar rango maximo
function maxt(e){
    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var min=parseInt(document.getElementById('mintrend').value);
        var max=parseInt(document.getElementById('maxtrend').value);
        settings[indice].seriesGroups[0].valueAxis.minValue=min;
        settings[indice].seriesGroups[0].valueAxis.maxValue=max;
        console.log(settings[indice]);

        $("#trend"+indice).jqxChart('refresh');
        $("#trend"+indice).jqxChart('update');
    }
}
//Cambiar titulo
function titulot(e){

```

```

    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var titulo=document.getElementById('titulotrend').value;
        $('#trend'+indice).jqxChart({title: titulo});
        $('#trend'+indice).jqxChart('refresh');
        console.log(settings[indice]);
    }
}

//Cambiar descripcion
function descripciont(e){
    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var descripcion=document.getElementById('descripciontrend').value;
        settings[indice].Description=descripcion;
        $('#trend'+indice).jqxChart({description: descripcion});
        // refresh the chart element
        $('#trend'+indice).jqxChart('refresh');
    }
}

//Cambiar leyenda del eje y
function ejeY(e){
    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var ejeY=document.getElementById('ytrend').value;

        settings[indice].seriesGroups[0].series[0].displayText=ejeY;
        //settings[indice].seriesGroups[0].valueAxis.maxValue=max;
        console.log(settings[indice]);
        $('#trend'+indice).jqxChart('refresh');
        $('#trend'+indice).jqxChart('update');
    }
}

//Cambiar leyenda eje X
function ejeX(e){
    var indice=actualtrend.substring(5,actualtrend.length);
    if (e.keyCode === 13) {
        var ejeX=document.getElementById('xtrend').value;

        settings[indice].seriesGroups[0].valueAxis.title.text=ejeX;
        //settings[indice].seriesGroups[0].valueAxis.maxValue=max;
        console.log(settings[indice]);
    }
}

```

```

        $('#trend'+indice).jqxChart('refresh');
        $('#trend'+indice).jqxChart('update');
    }
}

//eliminar control
function removet1(){
    var indice=actualtrend.substring(5,actualtrend.length);
    $("#trend"+indice).remove();
}

```

- **TANQUE**

```

var tanque=[];
var data=[];
var scale=[];
var axis=[];
var tani=0;
var contanque=[];
var propiedades=[];
var actualtanque;
var indicetan;
var myWidth;
var maxtan=100;
//VARIABLES DE LA COMUNICACION
var auxtan;
var indicevartan;
var tagtan;
var conexionestan=[];
var VDIRtan=[];
var datatan=[]

$(window).resize(function(){
    myWidth = window.innerWidth;
    document.getElementById("c5propiedades1").style.left=(myWidth-200)+"px";
})

$(document).ready(function(){
    tank(tani);
    auxtan=null;

//EVENTO QUE RECIBE LAS VARIABLES
    socket.on('message', function (data) {

        if(data.value!=undefined){
            if(data.value[0]!=auxtan){

```



```

auxtan=data.value[0];

var x=document.getElementById('variablestan');
var z = document.createElement("option");

z.setAttribute("value", data.value);

var t = document.createTextNode(data.value);
z.appendChild(t);
x.appendChild(z);
}
}

));

//RESTABLECE EL CUADRO DE PROPIEDADES
$("div").dblclick(function(event) {
    if(event.target.id.substring(0,3)=="tan"){
        document.getElementById("c5propiedades1").style.display="block";
    }
});

//Funcion que determina que el objeto sleccionado es un tanque
$("div").click(function(event) {
    if(event.target.id.substring(0,3)=="tan"){
        document.getElementById("labeltag1").innerHTML=event.target.id;

        actualtanque=event.target.id;
        indicetan=event.target.id.substring(3,event.target.id.length);
        document.getElementById("c5labeltag1").innerHTML="Tanque"+indicetan;

        //poner en primer plano herramientas del control seleccionado
        document.getElementById("c1propiedadesb1").style.zIndex=1; //boton1
        document.getElementById("propiedadesi1").style.zIndex=1; //gauge1
        document.getElementById("i3propiedades1").style.zIndex=1; //led
        document.getElementById("i4propiedades1").style.zIndex=1; //label
        document.getElementById("c2propiedades1").style.zIndex=1; //gauge2
        document.getElementById("i6propiedades1").style.zIndex=1; //trend
        document.getElementById("i7propiedades1").style.zIndex=1; //display
        document.getElementById("c3propiedades1").style.zIndex=1; //powerboton
        document.getElementById("c4propiedades1").style.zIndex=1; //perilla
        document.getElementById("c5propiedades1").style.zIndex=20; //tanque
        document.getElementById("icpropiedades1").style.zIndex=1; //eslider
        document.getElementById("c6propiedades1").style.zIndex=1; //switch

        $("#"+actualtanque).resizable();
        $("#"+actualtanque).draggable({

```

```

        containment: '#mySidenav',
        mouse: 'move',
        cursor: 'move',
    });
}
});
});

```

```
function c5add1(){
```

```

    tani++;
    contanque[tani] = document.createElement("div");
    contanque[tani].id="tan"+(tani);
    $("#mySidenav").append(contanque[tani]);
    $("#c5propiedades1").appendTo("#mySidenav");

```

```
//TAMANO DEL CONTROL A SER AGREGADO
```

```

document.getElementById(contanque[tani].id).style.width=300+"px";
document.getElementById(contanque[tani].id).style.height=300+"px";
document.getElementById(contanque[tani].id).style.position="absolute";
document.getElementById(contanque[tani].id).style.left=200+"px";
document.getElementById(contanque[tani].id).style.top=300+"px";

```

```
//DAR FORMATO AL BORDE
```

```

document.getElementById(contanque[tani].id).style.border="10px solid transparent";
document.getElementById(contanque[tani].id).style.borderImage="url(ima1.png)
round";
document.getElementById(contanque[tani].id).style.borderImageSlice="55%";

```

```
document.getElementById("c5labeltag1").innerHTML="Tanque"+(tani);
```

```
//UBICACION DE PROPIEDADES
```

```

document.getElementById("c5propiedades1").style.top=(0)+"px";
document.getElementById("c5propiedades1").style.left=(myWidth-200)+"px";
document.getElementById("c5propiedades1").style.display="block";

```

```

    tank(tani);
    tanque[tani].container(contanque[tani].id);
    tanque[tani].draw();

```

```
}
```

```

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@-----
COMUNICACION-----
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

```
function tagvaluetan(){
```

```

//OBTENER VALOR DE LA VARIABLE A CONECTAR
tagtan=document.getElementById("variablestan").value;

conexionestan[0]=tagtan.substring(0,tagtan.search(","));
conexionestan[1]=indicetan;

//ENVIAR LA VARIABLE QUE VAMOS A MONITOREAR AL CLIENTE
socket.emit('sendtanque', conexionestan);

//RECIBIR LA VARIABLE DEL CLIENTE OPCUA
socket.on('valuestanque', function (data) {
    //console.log(data.value.length);

    //console.log(data.value[parseInt(indicetan)-1]);
    //console.log(indicetan);
    //datatan[parseInt(indicetan)]=data.value[parseInt(indicetan)-1];

    var tani=0;
    for(tani=0;tani<data.value.length;tani++){
        var valgaugetan=data.value[tani];
        if(tanque[tani+1]!==undefined){
            tanque[tani+1].data([maxtan,valgaugetan]);
        }
    }
});
}

function c5changeColor1(){
    propiedades[indicetan].color(document.querySelector("#c5color1").value);
}

function c5nombre1(e) {

    if (e.keyCode === 13) {
        tanque[indicetan].title().text(document.getElementById("c5etiqueta1").value);
    }
}

//OCULTAR CUADRO DE CONTROLES
function c5cerrar1() {
    document.getElementById("c5propiedades1").style.display="none";
}

function c5range1(){
    maxtan=document.getElementById("c5max1").value;
    var val=maxtan*0.5;
    //Modificar datos de escala y valor

```

```

    tanque[indicetan].data([maxtan, val]);
    //Cambiar escalas
    tanque[indicetan].scale().maximum(maxtan);
}

function c5remove1(){
    $("#"+actualtanque).remove();
}

function tank(tani){
    //-----VALORES ESTATICOS-----
    //-----
    // Crear los datos Iniciales
    var data= [100, 50];

    //Seleccionar el tipo de gauge a linear
    tanque[tani]= anychart.gauges.tank();

    //Zindex
    tanque[tani].zIndex(-1);

    //Ingresar datos en el gauge
    tanque[tani].data(data);

    //Fondo del tanque transparente
    tanque[tani].background().fill('#FFF ');

    // Establecer Titulo del tanque
    tanque[tani].title()
        .enabled(true)
        .useHtml(true)
        .padding([5,15])
        .text("");

    //-----ESCALAS-----
    // Valores maximos de escalas
    tanque[tani].scale()
        .minimum(0)
        .maximum(100)
        // Set axis tick intervals
        .ticks({'interval': 0.1*tanque[tani].scale().maximum()})
        .minorTicks({'interval': 10});

    // Distancia de las escalas
    tanque[tani].axis().offset();
    tanque[tani].axis().minorTicks(true)

```

```

        .width('-2')
        .offset('10%');

// Unidades
tanque[tani].axis().labels()
    .useHtml(true)
    .format('{% Value} L');

// Leyenda del recuadro
tanque[tani].tooltip().format('{% Value} L');

//-----PROPIEDADES DEL TANQUE-----
propiedades[tani] =tanque[tani].tank(1);
propiedades[tani].name("");
propiedades[tani].color('blue');
propiedades[tani].offset('10%');
propiedades[tani].width('70%');

//-----TANQUE INICIAL-----

tanque[tani].container('chart-container');
tanque[tani].draw();
document.getElementById("chart-container").style.width=300+"px";
document.getElementById("chart-container").style.height=240+"px";
}

```

ANEXO 4

**TABLAS DE COSTOS REFERENCIALES DE LICENCIAS
DE SOFTWARE PARA CREACIÓN DE HMIs**

• **TABLA DE COSTOS WINCC**

Versión	Descripción	Costo
6AV2104-0BA04-0AA0 Siemens WinCC RT Advanced 128 PowerTags V14	Simatic WinCC Runtime Advanced 128 PowerTags V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	\$522.05
6AV2104-0DA04-0AA0 Siemens WinCC RT Advanced 512 PowerTags V14	Simatic WinCC Runtime Advanced 512 PowerTags V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	\$1,687.38
6AV2104-0FA04-0AA0 Siemens WinCC RT Advanced 2048 PowerTags V14	Simatic WinCC Runtime Advanced 2048 PowerTags V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	\$2,597.08
6AV2104-0HA04-0AA0	Simatic WinCC Runtime Advanced 4096 PowerTags	\$3,896.12

Siemens WinCC RT Advanced 4096 PowerTags V14	V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	
6AV2104-0KA04-0AA0 Siemens WinCC RT Advanced 8192 PowerTags V14	Simatic WinCC Runtime Advanced 8192 PowerTags V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	\$5,193.15
6AV2104-0LA04-0AA0 Siemens WinCC RT Advanced 16384 PowerTags V14	Simatic WinCC Runtime Advanced 16384 PowerTags V14, runtime software in TIA Portal; single license; software and documentation on DVD; license key on USB	\$7,137.20

- COSTOS DE LICENCIAS DE LABVIEW

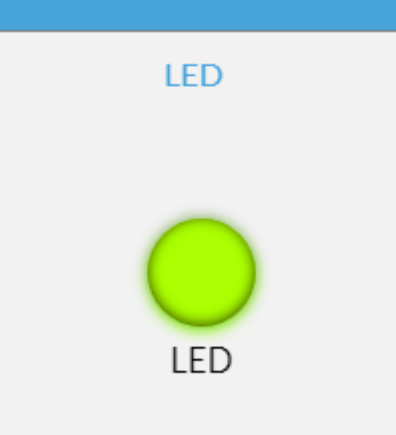
Software	Descripción	Costo
LabVIEW Base	<ul style="list-style-type: none"> • Recomendado para aplicaciones de medidas de escritorio • Incluye controladores de dispositivos para hardware de NI e instrumentos de terceros • Incluye matemáticas básicas y procesamiento de señales 	\$ 440/año
LabVIEW Completo	<ul style="list-style-type: none"> • Recomendado para matemáticas avanzadas en línea y procesamiento de señales • Se requiere para complementos de procesamiento de señales • Se requiere para hardware FPGA y en tiempo real 	\$ 3,300.00
LabVIEW Profesional	<ul style="list-style-type: none"> • Recomendado para aplicaciones que requieren validación de código • Incluye código y capacidades de implementación de aplicaciones • Incluye múltiples complementos de ingeniería de software 	\$ 5,500.00

- COSTOS DE LICENCIAS DE OPEN AUTOMATION SOFTWARE

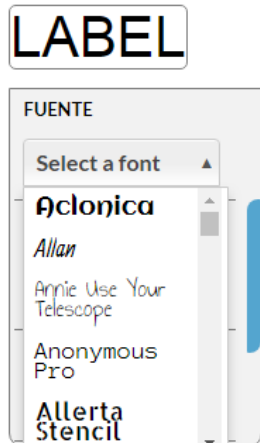
Componentes de la plataforma	Costo
Data Historian	\$695
Alarm Logging and Notification	\$795
Device and Data Connectors	\$495
IoT Connectors	\$695
Visualization Tools	\$1595
Programmatic Interface	\$995

ANEXO 5
DESARROLLO CONTROLES

Anteriormente se ha descrito de manera detallada el proceso de creación de los instrumentos virtuales, por lo cual a continuación se describirá de manera breve los demás instrumentos creados haciendo énfasis en sus partes más importantes.

DESCRIPCIÓN DE LOS ELEMENTOS VIRTUALES DISEÑADOS		
INSTRUMENTO	CÓDIGO RELEVANTE	DESCRIPCIÓN
	<pre> :root { /* :root is a clever way of saying "the outer-most element" */ --l3led1-color: #ABFF00; /* the default */ --l3led2-color: #304701; --l3led3-color: #89FF00; } #controlbox13{ position: relative; width: 330px; height: 210px; } #controlbox13 p { font-size: 17px; text-align: center; margin: auto; margin-top: 60px; } .led { margin: 0 auto; position: absolute; width: 54px; height: 54px; top: 30%; left: 43%; background-color: var(--l3led1-color); border-radius: 50%; box-shadow: rgba(0, 0, 0, 0.2) 0 -1px 7px 1px, inset var(--l3led2-color) 0 -1px 9px, var(--l3led3-color) 0 2px 12px; } #l3led01{ z-index: 1; } #l3led1{ z-index: 2; } </pre>	<p>La particularidad de este control es que no fue necesario la utilización de un plugin para su creación. Le concedió una forma circular mediante el lenguaje CSS a través del atributo border-radius, además se establecieron tres tipos de variables para modificar sus colores a través de la estructura que se muestra a continuación.</p> <p>:root{ --variable:valor de la variable;}</p> <p>Mediante JavaScript se creó una subrutina que permite cambiar el valor de estas variables para cambiar su color. El código detallado de este control se lo puede encontrar en el Anexo 3.</p>

LABEL



```
$(function(){  
  $('#i4font1').fontselect().change(function(){  
    // replace + signs with spaces for css  
    var font = $(this).val().replace(/\+/g, ' ');  
    // split font into family and weight  
    font = font.split(':');  
    // set family on paragraphs  
    $('#i4label1').css('font-family', font[0]);  
  });  
});
```

Al igual que el control anterior no se ocupó ningún plugin para crear el control pero se utilizó una librería que permite cambiar el tipo de fuente del label. En la figura del código se muestra las líneas de código pertenecientes a la librería que se utilizaron.

BOTÓN



CSS

```
.switch {
  font: bold 17px Helvetica, Arial, Sans-serif;
  margin: auto;
  height: 35px;
  width: 97px;
  border: 1px solid #979797;
  border-radius: 20px;
  margin-top: 0px;
  box-shadow: inset 0 1px 3px #BABABA, inset 0 12px 3px 2px rgba(232, 232, 232, 0.5);
  cursor: pointer;
  overflow: hidden;
  padding: 2px;
  position: absolute;
  left: 40;
  top: 35;
}
.switch input[type=checkbox] {
  display: none;
}
.switch::before {
  content: "";
  display: block;
  height: 30px;
  width: 0px;
  position: absolute;
  border-radius: 20px;
  -webkit-box-shadow: inset 0 1px 2px #0063B7, inset 0 12px 3px 2px rgba(0, 127, 234, 0.1);
  box-shadow: inset 0 1px 2px #0063B7, inset 0 6px 3px 2px rgba(200, 200, 150, 0.5);
  background-color: var(--led-color);
}
.switch.on::before {
  width: 90px;
}
```

JAVA SCRIPT

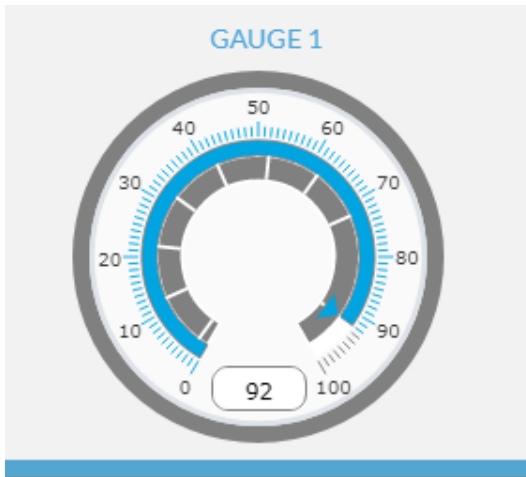
```
$$("#switch2").forEach(function(switchControl) {
  if (switchControl.className === ("switch on")) {
    switchControl.lastElementChild.checked = true;
  }
}
switchControl.addEventListener("click", function toggleSwitch() {
  if (switchControl.className === "switch on") {
    switchControl.className = 'switch off';
  } else {
    switchControl.className = ("switch on");
  }
}
checkbox = switchControl.lastElementChild;
checkbox.checked = !checkbox.checked;
url = "Variables/"+variable;
name =aux;
ival = +checkbox.checked;
sdata=escape(name)+'='+ival;
//alert(sdata);
$.post(url,sdata,function(result){});
```

Al igual que el control anterior solamente se ocupó código CSS para crear el aspecto visual del control. Mediante la propiedad **border-radius** y **box-shadow**. Las cuales permiten modificar la forma, colores y sombras de los distintos objetos de HTML.

Posteriormente se programó un evento click del botón, el cual se creó a partir de una etiqueta checkbox. Este evento permite evaluar cuando el botón cambie de on a off y generar un valor de 1 para encendido y 0 para apagado.

Este valor posteriormente será añadido como parámetro del comando post que escribirá un valor lógico de encendido o apagado en la variable del PLC.

GAUGE CONTROL



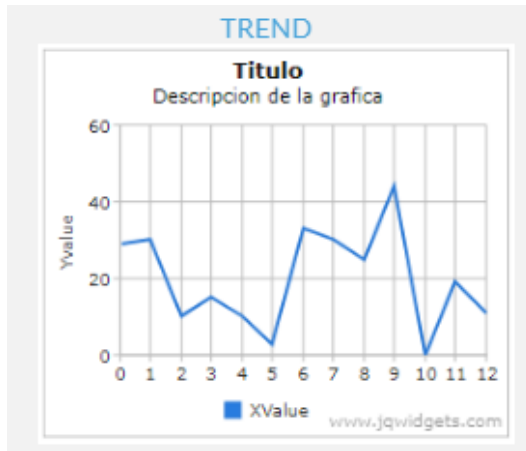
```
function drawgauge(cg2){
    $('#Gauge'+cg2).jqxKnob({
        value: c2value1,
        min: c2mini,
        max: c2max1,
        width:c2width,
        height:c2height,
        startAngle: 120,
        endAngle: 420,
        snapToStep: true,
        rotation: 'clockwise',
        style: { stroke: '#dfe3e9', strokeWidth: 3, fill: { color: '#fafafa', } },
        marks: {
            colorRemaining: { color: 'grey', border: 'grey' },
            colorProgress: { color: '#00a4e1', border: '#00a4e1' },
            type: 'line',
            offset: '71%',
            thickness: 1,
            size: '6%',
            majorSize: '9%',
            majorInterval: c2max1/10,
            minorInterval: c2max1/100
        },
        spinner: {
            style: { fill: { color: 'grey', stroke: 'white' },
            innerRadius: '45%', // specifies the inner Radius of the dial
            outerRadius: '60%', // specifies the outer Radius of the dial
            marks: {
                colorRemaining: '#fff',
                colorProgress: '#fff',
                type: 'line',
                offset: '46%',
                thickness: 2,
                size: '14%',
                majorSize: '14%',
                majorInterval: c2max1/10,
                minorInterval: c2max1/10
            },
        },
        labels: {
            offset: '88%',
            step: c2max1/10,
            visible: true
        },
        progressBar: {
            style: { fill: '#00a4e1', stroke: 'grey' },
            size: '9%',
            offset: '60%',
        }
    });
}
```

La característica más importante de este control es la utilización del plugin jqxKnob. El cual nos permite crear este gauge y modificar varias propiedades del mismo a través de estructuras muy simples. Por ejemplo para adquirir o setear el valor del gauge se lo hace mediante la siguiente estructura.

```
$('#Gauge'+cg2).on('change', function (event) {
    if (event.args.changeSource == 'propertyChange' || event.args.changeSource == 'val') { return; }
    $(input[cg2]).val(event.args.value);
})
```

La misma que utiliza el evento onchange para actualizar el valor que tomar el control al variarlo

TREND



```
function grafica(contrend){
  var it=0
  for(it=0;it<it++;)
  {
    sampleData[it]=100*Math.random();
  }
  sampleData [contrend]= sampleData;
  console.log(sampleData);
  settings[contrend] = {
    title: "Titulo",
    Description:"Descripcion de la grafica",
    showLegend: true,
    padding: { left: 5, top: 5, right: 15, bottom: 5 },
    titlePadding: { left: 0, top: 0, right: 0, bottom: 10 },
    source: sampleData[contrend],

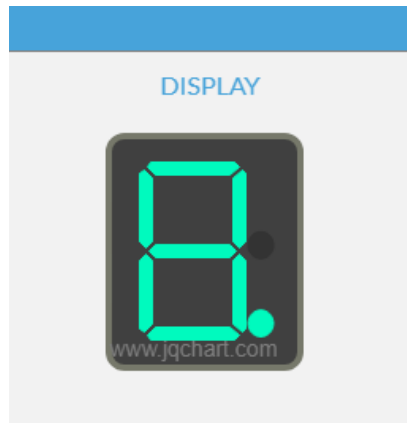
    xAxis:
    {
      text: 'x',
      valuesOnTicks: true
    },
    colorScheme: 'scheme01',
    seriesGroups:
    [
      {
        type: 'line',
        source: sampleData[contrend],
        tooltipFormatFunction: function (value, itemIndex, serie, group, categoryValue, categoryAxis) {
          var dataItem = sampleData[itemIndex];
          return '<div style="text-align:left"><b>Index:</b> ' +
            itemIndex + '<br /><b>Value:</b> ' +
            value + '<br /></div>';
        },
        valueAxis:
        {
          minValue: 0,
          maxValue: 100,
          title: { text: 'Yvalue<br>' }
        },
        series:
        [
          { emptyPointsDisplay: 'skip', displayText: 'Xvalue', lineWidth: 2, symbolSize: 0 }
        ]
      }
    ]
  }
  // create the chart
  $("#trend"+contrend).jqxChart(settings[contrend]);
}
```

Para la creación de este control se utilizó el plugin **jqWidgets**. Este plugin es muy sencillo de utilizar primeramente hay que crear una matriz de los datos que se visualizaran en el trend

Posteriormente hay que definir la estructura del control donde se designan cada una de las propiedades del mismo a través de una estructura Json. Para esto caso hemos definido esta estructura con el nombre settings[contrend].

Finalmente cuando se tengan todas las propiedades del trend establecidas se llama al contenedor del control el cual se lo ha denominado **#trend**

DISPLAY

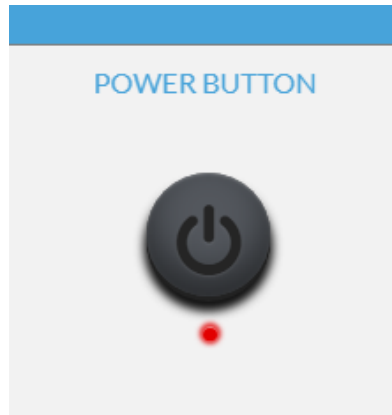


```
i7state1 = {
  background: '#404040',
  border: {
    padding: 10,
    lineWidth: 4,
    strokeStyle: '#76786A'
  },
  digits: 1,
  segmentMode: 'sevenSegment',
  text: '8',
  textForeground: 'rgba(0, 250, 190, 1)',
  textForegroundUnlit: '#333333'
};
var segmentedDisplay = new DataViz.SegmentedDisplay(i7state1);
segmentedDisplay.write('i7display1');
updateDisplay();

function updateDisplay() {
  segmentedDisplay.setState(i7state1);
  setTimeout('updateDisplay()', 500);
}
```

Este control utiliza la librería denominada jqchart haciendo referencia al objeto **DataViz.SegmentedDisplay**. La estructura que se emplea para crear este control es similar a la del control anterior siendo únicamente ciertas características diferentes las que se cambian como el número de dígitos y el color del display.

POWER BUTTON



HTML

```
<!--POWER BOTTON-->
<li><a href="#">POWER BUTTON</a>
<div id="controlboxpb" >
  <!--COPIA 0 POWER BUTTON-->
  <div id="c3container0" class="c3pboton" >
    <section class="c3section">
      <a href="#" id="c3pboton0" class="powerb">&#xF011;</a>
      <span class="c3span"></span>
    </section>
  </div>
```

CSS

```
.powerb {
  font-family: "FontAwesome";
  text-shadow: 0px 1px 1px rgba(250,250,250,0.1);
  font-size: 32pt;
  display: block;
  position: absolute;
  text-decoration: none;
  box-shadow: 0px 3px 0px 0px rgb(34,34,34),
             0px 7px 10px 0px rgb(17,17,17),
             inset 0px 1px 1px 0px rgba(250, 250, 250, .2),
             inset 0px -12px 35px 0px rgba(0, 0, 0, .5);
  width: 70px;
  height: 70px;
  border: 0;
  color: rgb(37,37,37);
  border-radius: 35px;
  text-align: center;
  line-height: 79px;
  background-color: var(--colorbt);

  transition: color 350ms ease, text-shadow 350ms;
  -o-transition: color 350ms ease, text-shadow 350ms;
  -moz-transition: color 350ms ease, text-shadow 350ms;
  -webkit-transition: color 350ms ease, text-shadow 350ms;
}
```

Este control se desarrolló netamente con código HTML y CSS mediante la modificación de las propiedades **box-shadow** y **background-color** y la aplicación de variables para los colores del control, de la misma manera que se realizó anteriormente.

PERILLA



```
$(function() {
  perilla=$('#c4perilla1').rotaryswitch({
    beginDeg: 0,
    lengthDeg: 360,
    minimum: 0,
    maximum: 101,
    step: 1,
    showInput: true,
    showMarks: true,
    themeClass: 'big'
  });

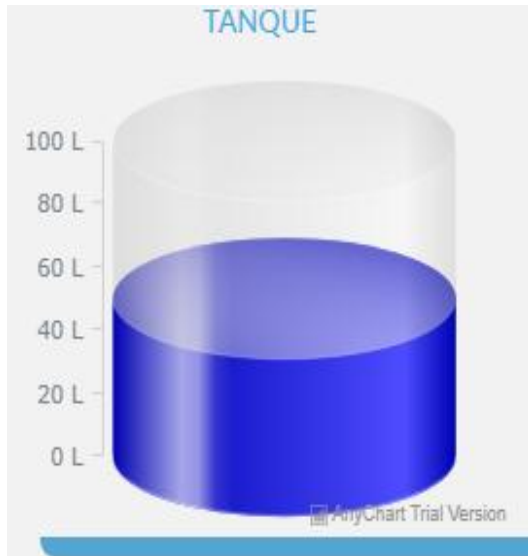
  perilla.on('change', function() {
    c4value1=perilla.val();
    //document.getElementById("c4labeltag1").innerHTML=c4value1;
    url = "Variables/"+c4variable1;
    name =c4aux1;
    ival = c4value1;
    sdata=escape(name)+'='+ival;
    //alert(sdata);
    $.post(url,sdata,function(result){});
    //DESPLEGAR EL VALOR DE LA VARIABLE DEL PLC SELECCIONADA
    document.getElementById("c4labeltag1").innerHTML=c4value1;
  });
});
```

Para crear este control utilizamos el plugin **rotarySwitch**. Este plugin es muy sencillo de usar basta definir las características o parámetros del control mediante JSON.

Para recuperar el valor del control es necesario utilizar el evento onchange el cual se ejecuta cada vez que exista un cambio en la variable. Y posteriormente debemos acceder al valor mediante el siguiente comando.

```
$('#rotarySwitch').rotaryswitch().val();
```

TANQUE



```
function tank(tani){
  //-----VALORES ESTATICOS-----
  //-----
  // Crear los datos Iniciales
  var data= [100, 50];
  //Seleccionar el tipo de gauge a linear
  tanque[tani]= anychart.gauges.tank();
  //Zindex
  tanque[tani].zIndex(-1);
  //Ingresar datos en el gauge
  tanque[tani].data(data);
  //Fondo del tanque transparente
  tanque[tani].background().fill('#FFF ');
  // Establecer Titulo del tanque
  tanque[tani].title()
    .enabled(true)
    .useHtml(true)
    .padding([5,15])
    .text("");

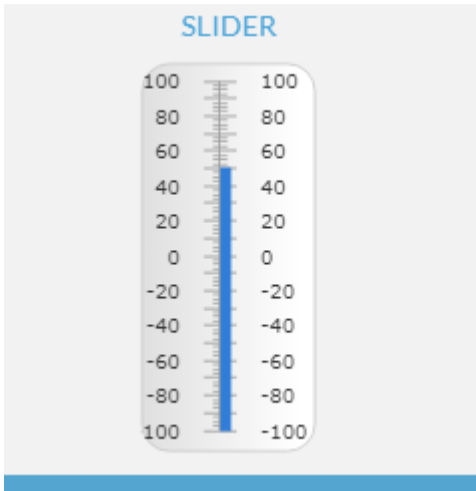
  //-----ESCALAS-----
  // Valores maximos de escalas
  tanque[tani].scale()
    .minimum(0)
    .maximum(100)
    // Set axis tick intervals
    .ticks({'interval': 0.1*tanque[tani].scale().maximum()})
    .minorTicks({'interval': 10});
  // Distancia de las escalas
  tanque[tani].axis().offset();
  tanque[tani].axis().minorTicks(true)
    .width('-2')
    .offset('10%');
  // Unidades
  tanque[tani].axis().labels()
    .useHtml(true)
    .format('{%value} L');

  // Leyenda del recuadro
  tanque[tani].tooltip().format('{%value} L');

  //-----PROPIEDADES DEL TANQUE-----
  propiedades[tani] =tanque[tani].tank(1);
  propiedades[tani].name("");
  propiedades[tani].color('blue');
  propiedades[tani].offset('10%');
  propiedades[tani].width('70%');
```

Para la creación de este control este control se utilizó el plugin **AnyChart** el cual nos permite crear el control tanque. Este plugin presenta una estructura en forma de array para ingresar la información correspondiente al control y la cual dara forma al mismo como podemos observar en el código presentado.

SLIDER



```
function drawslider(cslid){
    $('#islider'+cslid).jqxLinearGauge({
        orientation: 'vertical',
        width: '100px',
        height: '220px',
        labels: { interval: 20 },
        ticksMajor: { size: '10%', interval: 10 },
        ticksMinor: { size: '5%', interval: 2.5, style:
        max: 100,
        min: -100,
        value: -60,
        pointer: { size: '6%' },
        colorScheme: 'scheme01',
    });
    $('#islider'+cslid).jqxLinearGauge('value', 50);
}
```

Este control utiliza el plugin jqWidgets para su construcción. Este plugin es similar al del Control Gauge y su estructura define todas las características del mismo a través de código JSON.

Para acceder y modificar sus propiedades se lo hace de la siguiente manera

```
$('#islider'+cslid).jqxLinearGauge('value', 50);
```

SWITCH



```
<!--SWITCH-->
<li><a href="#">SWITCH</a>
<div id="controlboxsw">
  <!--COPIA 0 SWITCH-->
  <div id="c6move0" class="c6move">
    <span class="switchl2">
      <span class="switch-border1">
        <span class="switch-border2">
          <input id="switch012" type="checkbox" checked />
          <label for="switch112"></label>
          <span class="switch-top"></span>
          <span class="switch-shadow"></span>
          <span class="switch-handle"></span>
          <span class="switch-handle-left"></span>
          <span class="switch-handle-right"></span>
          <span class="switch-handle-top"></span>
          <span class="switch-handle-bottom"></span>
          <span class="switch-handle-base"></span>
          <span class="switch-led switch-led-green">
            <span class="switch-led-border">
              <span class="switch-led-light">
                <span class="switch-led-glow"></span>
              </span>
            </span>
          </span>
          <span class="switch-led switch-led-red">
            <span class="switch-led-border">
              <span class="switch-led-light">
                <span class="switch-led-glow"></span>
              </span>
            </span>
          </span>
        </span>
      </span>
    </span>
  </div>
</span>
</span>
</div>
</div>
```

El código de este control es en su mayoría HTML, en especial etiquetas span las cuales se utilizan para crear la forma del control y modificarlo a través de código CSS.

ANEXO 6
MANUAL DE USUARIO

Como se ha descrito en el desarrollo de la propuesta la plataforma desarrollada se ha dispuesto de una manera que su utilización se fácil e intuitiva para el usuario final por lo cual se resumirá su utilización en 5 sencillos pasos.

- 1) Establecer los parámetros de la variable a monitorizar y la dirección IP del controlador con el cual se desea establecer la conexión como se puede observar en la presente figura.



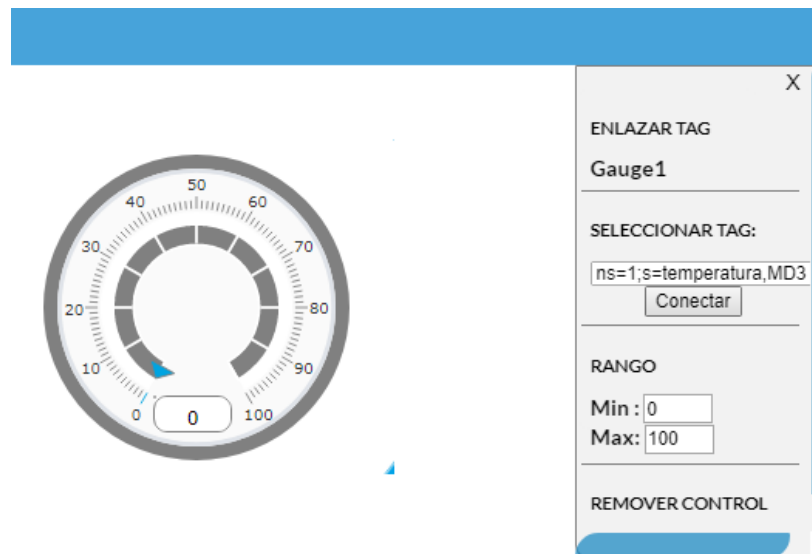
The screenshot displays the 'ANC PLATFORM HMI Web Server' interface. At the top, there is a blue header with a gear icon and the text 'Controles Conexiones'. Below this, the main content area is titled 'CONEXIÓN'. It contains a form with the following fields and values:

- 'Ingrese la direccion IP del PLC': 192.168.1.10
- 'Informacion de la Variable:':
 - 'Nombre:': temperatura
 - 'Direccion:': MD3
- A 'Crear Variable' button is located below the 'Direccion:' field.

- 2) Para ubicar un control en el área de trabajo para su posterior utilización se debe hacer doble click sobre el mismo y este se desplegara automaticamente en el área de trabajo como se puede observar en la figura a continuación.



- 3) Para poder establecer la monitorización se una determinada variable es necesario ubicarse en el control al cual se le enlazara la variable a monitorizar. Seleccionar la variable en la lista desplegable denominada como Seleccionar Tag y proceder a presionar el botón conectar como se observa a continuación.



- 4) Es necesario también ocultar los cuadros de propiedades de los diferentes controles por lo cual se ha dispuesto de una funcion para ocultarlos como se muestra a continuación.



- 5) Finalmente es necesario poder remover los controles que ya no serán necesarios o simplemente se requiera eliminar para lo cual se ha dispuesto del botón remover en el cuadro de propiedades del control como se muestra a continuación.

