

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**CREACIÓN DE PLATAFORMA DE SOFTWARE SUPERVISORIO
EN JAVA PARA EL LABORATORIO DE REDES INDUSTRIALES
DEL DACI - FIEE**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

ANDRADE CABRERA LUIS ANDRÉS

DIRECTOR: MSc. EGUEZ SARZOSA VICENTE ADRIÁN

CODIRECTOR: PhD. YACCHIREMA VARGAS DIANA CECILIA

Quito, julio 2022

AVAL

Certificamos que el presente trabajo fue desarrollado por Luis Andrés Andrade Cabrera, bajo nuestra supervisión.

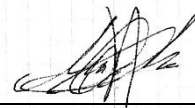
MSc. Vicente Adrián Eguez Sarzosa
DIRECTOR DEL TRABAJO DE TITULACIÓN

PhD. Diana Cecilia Yacchirema Vargas
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Luis Andrés Andrade Cabrera, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.



Luis Andrés Andrade Cabrera

DEDICATORIA

Este trabajo de titulación es el resultado del conjunto de esfuerzos de quienes más me apoyan por lo cual se los dedico a ellos.

A mi mamá, Florencia Cabrera por siempre estar pendiente de mí y apoyarme en todas las adversidades.

A mi papá, Luis Andrade por ser mi inspiración y motivarme cada vez que hablamos.

A mi hermana, Andrea Andrade por ayudarme a crecer y ser más responsable.

AGRADECIMIENTO

En este espacio tengo como propósito agradecer a las personas que me han ayudado a llegar hasta aquí, empezando con agradecer a mi mamá y a mi papá por toda la sabiduría, esfuerzo y recursos que han invertido en mi educación y formación como persona, que me han permitido soñar, pero aún más lograrlo.

Quiero agradecer también a mis tres hermanos, primero Andrea mi hermana que me ha ayudado innumerables veces, que ha sido un apoyo grande cuando la vida se me ha vuelto color gris, luego mis dos hermanos del alma, Alex y Nicole quienes se han vuelto los hermanos mayores que nunca tuve, quienes me han enseñado demasiadas cosas del diario vivir y en quienes sé que puedo confiar mi vida de ser necesario.

A continuación, quiero agradecer a mi universidad la Escuela Politécnica Nacional, a mi facultad de Sistemas, a mis compañeros de cada clase con quienes me he desarrollado con quienes he aprendido, disfrutado y también sufrido en este camino de educación superior. Además, agradezco a los grupos que me han acogido junto a ellos como son el grupo de Telegram de la facultad con amigos como Charles, Anthony, Carlos y muchos más que siempre interactuamos, el Club Gold con mis amigos como Héctor, Gutenberg, y Alejandro, y finalmente el grupo que lo considero mi segunda familia los Power Rangers con quienes hemos vivido un sinfín de aventuras y hemos evolucionado a punto de confiar mucho entre nosotros.

Finalmente, agradezco a una persona que es muy especial para mí, la primera persona en la que pienso al despertar y la última con quien hablo antes de dormir, Alisson, quien en estos 5 años y un poco más, me ha ayudado a crecer como persona, a encontrarme cuando más lo necesitaba, a ser mejor persona, más agradable, más amable con los demás. Saber que cuento con su soporte, con su confianza y con su orgullo hacia mí me permite ir más allá de cualquier límite que mi mente piense que tiene.

ÍNDICE DE CONTENIDO

RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN	1
1.1 OBJETIVOS	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	2
1.3.1 SISTEMAS SCADA.....	2
1.3.2 ESTRUCTURA INTERNA DE UNA APLICACIÓN SCADA.....	3
1.3.3 PROTOCOLOS DE COMUNICACIÓN USADOS EN SISTEMAS SCADA	5
1.3.4 TIPOS DE PÁGINAS HMI.....	8
1.3.5 ESTÁNDARES DE DISEÑO DE APLICACIONES HMI	11
1.3.6 FUNCIONALIDAD DE ALARMAS.....	26
1.4 HERRAMIENTAS, FRAMEWORKS Y LIBRERÍAS DE DESARROLLO .	27
1.4.1 IntelliJ IDEA.....	27
1.4.2 MySQL	28
1.4.3 MySQL Connector/J.....	28
1.4.4 JavaFX.....	28
1.4.5 Git	28
1.4.6 GitHub	28
1.4.7 Método de Priorización MoSCoW.....	29
1.4.8 SCRUM.....	29
2. METODOLOGÍA	34
2.1. Metodología SCRUM	34
2.2. Roles SCRUM.....	34
2.3. Requerimientos iniciales	35
2.4. DESCRIPCIÓN DEL SISTEMA.....	37
2.4.1 Modelo de la base de datos	37
2.4.2 Arquitectura del sistema.....	38
2.4.3 Patrón de diseño	39
2.5. Historias Épicas	40
2.6. Product Backlog	41

2.7.	Desarrollo.....	52
2.7.1	Sprint 0	52
2.7.2	Sprint 1	52
2.7.3	Sprint 2	56
2.7.4	Sprint 3	62
2.7.5	Sprint 4	70
2.7.6	Sprint 5	78
2.7.7	Sprint 6	95
2.7.8	Sprint 7	101
2.8.	Retrospectiva	109
3.	RESULTADOS Y DISCUSIÓN	110
3.1.	LibreHMI Plataforma de software supervisorio.....	110
3.2.	PRUEBAS DE USABILIDAD	113
3.2.1	Usabilidad y Facilidad de Uso	113
3.2.2	Escala de Usabilidad del Sistema.....	114
3.2.3	Resultados de encuesta SUS	116
4.	CONCLUSIONES Y RECOMENDACIONES.....	117
4.1.	CONCLUSIONES.....	117
4.2.	RECOMENDACIONES	119
5.	REFERENCIAS BIBLIOGRÁFICAS	121
	ANEXOS	124

RESUMEN

Las aplicaciones de software supervisorio, son una herramienta clave en el acontecer industrial, ya que permite monitorizar el estado de diversos dispositivos industriales que tienen conexión a la red a través de los PLC (Controlador Lógico Programable, por sus siglas en inglés) que se comunican mediante muchos protocolos entre ellos el protocolo Ethernet/IP™. Sin embargo, este tipo de software dedicado tiene costos muy altos y existen varios usuarios que requieren de opciones más limitadas pero esenciales y de manera gratuita, para ello, son necesarias alternativas que las implementen.

Este trabajo de titulación presenta una propuesta a modo de alternativa de software que permitirá monitorizar los dispositivos del área industrial a través de un controlador Ethernet/IP™, donde estos se mostrarán dentro de un lienzo a modo de representaciones gráficas y las distintas pantallas diseñadas con dichas representaciones podrán guardarse en archivos. Para realizar la implementación de esta propuesta se utilizó la metodología SCRUM para así poder asegurar la calidad del software, además de estimar los tiempos de obtención de entregables e integrar los nuevos incrementos sin afectar el sistema.

Finalmente, se obtuvo un software que cumple con las necesidades más críticas en cuanto al diseño y utilización de pantallas HMI (Interfaz Humano-Máquina, por sus siglas en inglés) para el monitoreo de dispositivos del área industrial, permitiendo al usuario cubrir con sus necesidades sin tener que pagar un alto precio.

PALABRAS CLAVE: HMI, Ethernet/IP, JavaFX, Java, Industria, PLC, Plataforma de Software, Supervisorio

ABSTRACT

Supervisory software applications are a key tool in industrial events, since they allow monitoring the status of various industrial devices that are connected to the network through PLCs that communicate through many protocols, including the Ethernet/IP™ protocol. However, this type of dedicated software has very high costs and there are several users who require more limited but essential options, and for this, alternatives are necessary to implement them.

This degree work presents a proposal as a software alternative that will allow monitoring the devices of the industrial area through an Ethernet/IP™ controller, where these will be displayed within a canvas as graphic representations and the different screens designed with said representations may be saved to files. To carry out the implementation of this proposal, the SCRUM methodology is used to ensure the quality of the software, in addition to estimating the times for obtaining deliverables and integrating the new increments without affecting the system.

Finally, a software will be obtained that will meet the most critical needs in terms of the design and use of HMI screens for monitoring devices in the industrial area, allowing the user to cover their needs without having to pay a high price.

KEYWORDS: HMI, Ethernet/IP, JavaFX, Java, Industrial, PLC.

1. INTRODUCCIÓN

El software específico para el desarrollo de prácticas de laboratorio es clave en el progreso y el aprendizaje de los estudiantes de la carrera de Electrónica y Control, así como de la carrera de Automatización que pertenecen a la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, ya que es un requerimiento solicitado por las empresas para poder implementar sistemas de control industrial [1].

El software de control industrial que es instruido en el Departamento de Automatización y Control Industrial(DACI) tiene un costo por licencia, del Software AVEVA Edge, con valores que van desde los \$3,240 hasta los \$20,740 dólares americanos [2]. El software AVEVA permite crear distintas pantallas de usuario para administrar distintos dispositivos, así como comunicarse a dichos dispositivos utilizando una gran variedad de controladores, que suman más de 250, además permite el registro de eventos para asegurar la trazabilidad de acciones que los usuarios inician, entre otras más opciones.

El DACI dentro de sus prácticas desarrolladas utiliza el software Wonderware® InTouch® HMI, que es una versión anterior del software mencionado AVEVA Edge [3], sin embargo, no son utilizadas todas las posibilidades que ofrece este software, es decir, es posible optar por alternativas con opciones más limitadas, pero que sean las más utilizadas, y que además, los estudiantes puedan adquirirlo para su uso personal de manera legal.

Las alternativas serían beneficiosas no solo para la universidad y el DACI, sino también para la comunidad en general, es decir, la pequeña y mediana empresa [4], que necesita incentivar su desarrollo así como su productividad gracias al software de control supervisorio que puede gestionar de manera eficiente los sistemas de administración y control industrial, con una alternativa de bajo costo no repercuta en el presupuesto de las Pequeñas Y Medianas Empresas(PYMES).

Tomando en consideración lo antes expuesto el presente proyecto integrador propone la creación de un software para la implementación de aplicaciones de control supervisorio bajo licencia libre y a bajo costo que permita el desarrollo de programas de control industriales, que puedan ser utilizados en las prácticas del DACI, que incluyan elementos e interacciones específicas que sean similares a los utilizados actualmente en dichas prácticas. Este software será desarrollado en el lenguaje de programación JAVA, de modo que permita su ejecución en diferentes sistemas operativos.

1.1 OBJETIVOS

El objetivo general de este proyecto integrador es implementar una plataforma de software para el desarrollo de aplicaciones de control supervisorio constituido por una aplicación de escritorio en JAVA que permita la comunicación con PLCs MicroLogix 1100 a través del protocolo EtherNet/IP™ para el Departamento de Automatización y Control Industrial (DACI) de la Escuela Politécnica Nacional

Los objetivos específicos de este Proyecto Integrador son:

- Determinar los requisitos del sistema a nivel funcional y el alcance del software supervisorio.
- Diseñar la arquitectura e interfaces de usuario buscando asemejarse a HMIs profesionales.
- Implementar una aplicación de escritorio en JAVA que permita la interoperabilidad entre sistemas operativos, y se conecte a sistemas SCADA implementados en los laboratorios.
- Aplicar pruebas de aceptación para comprobar que se cumplan con los requerimientos solicitados por el DACI.
- Comprobar la calidad del producto a través de encuestas, para medir la usabilidad y facilidad de uso del sistema.

1.2 ALCANCE

Este proyecto integrador busca implementar una aplicación que pueda ejecutarse en distintos sistemas operativos de escritorio como Windows y Linux. Este software deberá tener un comportamiento similar al de HMIs profesionales, es decir, la capacidad para guardar archivos con el estado actual de un proyecto, así como conectarse a través de la red para conectarse a los PLC, específicamente el modelo MicroLogix 1100, utilizando el protocolo Ethernet/IP™.

1.3 MARCO TEÓRICO

1.3.1 SISTEMAS SCADA

Un sistema de supervisión es aquél que permite asegurar el correcto funcionamiento de un proceso aún si suceden situaciones anómalas [5]. Todo sistema de supervisión busca automatizar tareas, para esto requiere de toda la información, así como conocimiento existente del proceso [5].

La implementación actual de un sistema de supervisión por parte de la industria se ha reflejado en los sistemas de control de supervisión y adquisición de datos (SCADA, por sus siglas en inglés) que tal y como se infiere desde su nombre adquieren los datos permitiendo su control y supervisión [5].

Los sistemas SCADA pueden ser solo una aplicación o varias, que se comunican digitalmente con los diferentes dispositivos necesarios en la industria como instrumentos y actuadores, además proveen una interfaz gráfica de alto nivel para sus operadores y usuarios [5].

Las siglas SCADA pueden ser utilizadas para referirse hacia las aplicaciones que se han diseñado para el proceso como para los entornos que permiten su desarrollo [5].

Los usuarios de un sistema SCADA requieren las siguientes funcionalidades básicas:

- Adquirir y almacenar los datos de los procesos [5].
- Representar gráficamente las variables que pertenezcan a un proceso, entonces serán necesarias animaciones para indicarle al usuario que los datos varían con el tiempo, además se debe permitir la monitorización de dichas variables [5].
- Control, que permita actuar sobre los distintos miembros del proceso, como autómatas y reguladores autónomos. También podría afectar directamente al proceso mediante entradas y salidas remotas [5].
- Arquitectura que sea abierta y permite flexibilidad para ampliación, así como adaptación a los cambios que un proceso pueda desarrollar a lo largo del tiempo [5].

1.3.2 ESTRUCTURA INTERNA DE UNA APLICACIÓN SCADA

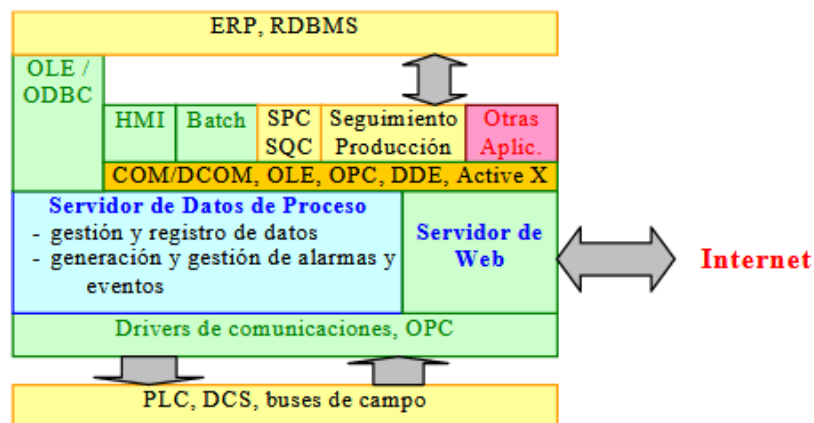


Figura 1.1. Estructura de un sistema SCADA [5]

Un sistema SCADA puede definirse en líneas generales a través de los siguientes elementos que se pueden observar en la Figura 1.1:

- **Driver de comunicación:** Es el agente encargado de administrar la comunicación entre los servidores de datos y los dispositivos que pertenezcan al proceso, además es quien implementa los protocolos que permiten dichas comunicaciones [5].
- **Servidor de datos del Proceso:** Es el agente que detectará y gestionará tanto alarmas como eventos que se desencadenen dentro del proceso, así como almacenar los datos para realizar análisis y posterior retroalimentación. Además, debe tener acceso directo hacia las bases de datos que se estén utilizando en el proceso [5].
- **Servidor Web:** Es necesario para un proceso cuyos datos se han lanzado para que sean comunicados a través de internet, administrando su disponibilidad y su acceso. Además habilita la comunicación desde equipos remotos para poder recibir información de la planta [5].
- **Interfaz Humano Máquina (HMI, por sus siglas en inglés):** Es la interfaz gráfica que permite la visualización de las variables del proceso a través de figuras, gráficos, objetos animados, textos, listas, múltiples ventanas, etc [5].
- **OLE/ODBC:** Es el estándar de Microsoft que permite comunicarse hacia distintos tipos de sistemas de bases de datos, se requiere cuando un sistema SCADA se conectará con bases de datos de sistemas que le son superiores y pertenecen a la gestión de la empresa como por ejemplo un sistema de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés) o un sistema de Planificación de los Recursos de Fabricación (MRP, por sus siglas en inglés) [5].
- **Batch:** Es una aplicación que permite la gestión de solicitudes y también de los procesos por lotes. Sin embargo, este paquete es solo opcional o se oferta a través de terceros [5].
- **Controles Estadísticos de Procesos / Control Estadístico de Calidad (SPC/SQC, por sus siglas en inglés):** Es una aplicación que permite realizar el seguimiento de variables del proceso, así como el control de la calidad de estas a través de distintos atributos, además de realizar cálculos estadísticos que permiten obtener gráficas que describen la calidad del proceso [5].

- **Seguimiento de Producción:** Son las aplicaciones que se han adaptado a cada proceso y también a cada empresa, por lo que permiten un amplio espectro de configuraciones, permitiendo generar consultas hacia la base de datos del proceso, así como la generación de listados específicos de elementos del proceso [5].

Además de los elementos anteriormente expuestos existen otros que se encuentran en el cuadro naranja en la parte central de la Figura 1.1 que son estándares de Microsoft tales como DDE, OLE, COM/DCOM, ActiveX y OPC que permiten la integración de aplicaciones de terceros o también aplicaciones específicas diseñadas particularmente para un proceso [5]. En síntesis, en el área del cuadro naranja se muestra cómo se integran los sistemas SCADA con los datos de planta y a su vez con los sistemas de gestión de costos, producción, pedidos, materiales, etc [5].

1.3.3 PROTOCOLOS DE COMUNICACIÓN USADOS EN SISTEMAS SCADA

Como fue expuesto anteriormente los sistemas SCADA tienen varios tipos de comunicaciones, entonces para asegurar que estas sean continuas, confiables y eficientes, varios tipos de protocolos de comunicación han sido diseñados [6]. Estos protocolos buscan cumplir con los requerimientos comunicacionales de aplicaciones industriales y deben tomar en cuenta por ejemplo las capacidades de procesamiento de sus componentes como un PLC [6]. A continuación, se detalla una descripción general de los protocolos más utilizados en sistemas SCADA.

1.3.3.1 Protocolos Basadas en Fieldbus

Fieldbus es una familia de protocolos que se diseñaron para ser un sistema de red que permita el control distribuido en tiempo real de aplicaciones industriales [6]. Este conjunto de protocolos ofrece ventajas al compararse con el cableado paralelo ya que al utilizar solamente un cable en todos los dispositivos permite que las redes se diseñen y desplieguen con mayor eficiencia y rapidez [6].

Dentro de esta familia de protocolos se encuentran los siguientes:

- BITBUS [6]
- Foundation Fieldbus H1 [6]
- PROFIBUS [6]
- WorldFIP [6]

1.3.3.2 Protocolos Basados en Ethernet

A lo largo del tiempo se han observado las ventajas de la tecnología de red Ethernet, que han impulsado el desarrollo de muchos protocolos de comunicación industrial que lo incorporan en sus soluciones [6]. Los sistemas de aplicación industrial requieren de una latencia muy baja, por lo que se desarrolló el estándar de Ethernet Industrial que modifica la capa de control de acceso a medios (MAC, por sus siglas en inglés) [6]. Ethernet Industrial ofrece muchas ventajas sobre otros tipos de redes industriales, una de ellas es una mayor velocidad en comparación a otras comunicaciones de serie, ya que hace uso de las capacidades de los cables Cat5e/Cat6 y también fibra óptica [6]. Ethernet Industrial con su capa MAC modificada no anula la utilización de direcciones MAC normales, por lo que estas pueden seguir siendo utilizadas para la identificación de dispositivos [6].

Dentro de esta familia de protocolos se encuentran los siguientes:

- Protocolo de Red Distribuida (DNP3, por sus siglas en inglés) [6].
- EtherCAT [6].
- Fundación Ethernet de Alta Velocidad (HSE, por sus siglas en inglés) [6].
- Comisión Internacional Electrotécnica (IEC, por sus siglas en inglés) 61850 [6].
- PROcess Field NET(PROFINET) [6].
- SERCOS III [6].
- Ethernet Powerlink [6].
- Protocolos de automatización en tiempo real para Ethernet Industrial (RAPIEnet, por sus siglas en inglés) [6].

1.3.3.3 Protocolos Seriales

Dentro de esta familia de protocolos se encuentran los siguientes:

- El conjunto de estándares de la familia IEC 60780 [6].
- Modbus [6].
- Unitronics PCOM [6].

1.3.3.4 Protocolo Industrial Común (CIP, por sus siglas en inglés)

Es un conjunto de protocolos que provee una infraestructura para las comunicaciones industriales. A continuación, se muestra en la Figura 1.2. Pila del Protocolo Industrial Común [6] donde se resume la pila del protocolo [6].

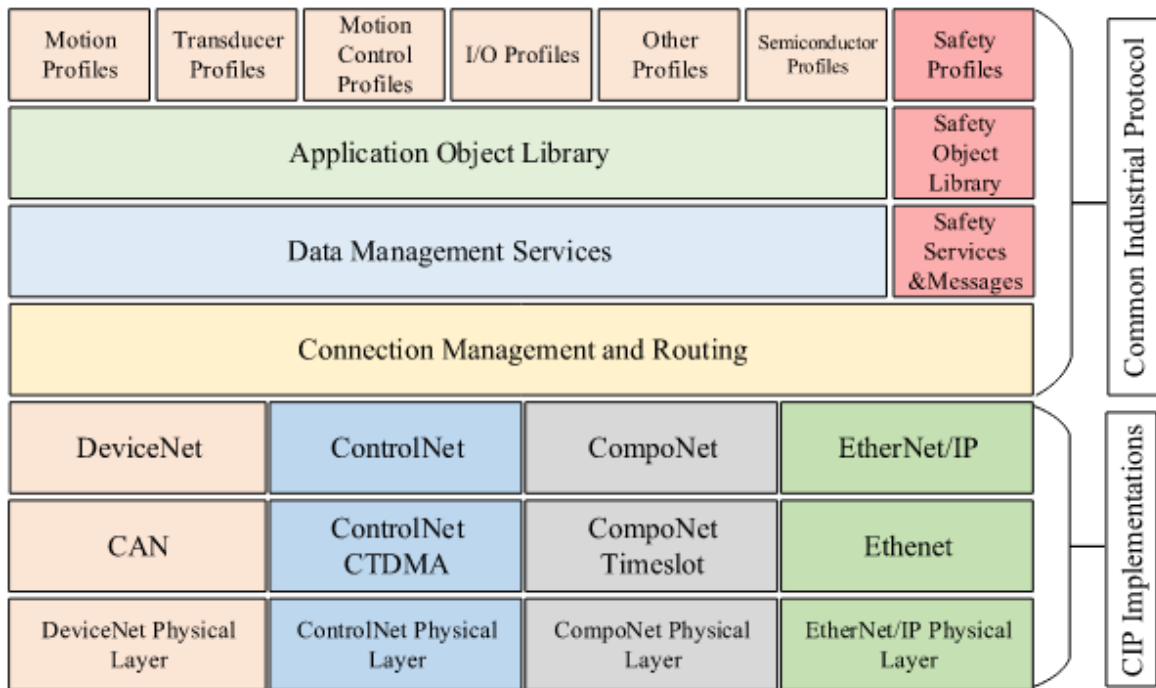


Figura 1.2. Pila del Protocolo Industrial Común [6]

En el nivel superior de la pila mostrada en la Figura 1.2 se tienen varios perfiles para distintos tipos de dispositivos permitiendo aumentar la interoperabilidad y también la coherencia de dichos dispositivos sin importar su proveedor [6]. Luego se presenta la biblioteca de objetos de aplicación que facilita una interfaz de aplicación, donde cada uno de los objetos tendrá sus atributos, servicios y comportamientos [6]. Después, los servicios de gestión de datos que se encargan de definir los modelos de direccionamiento para las entidades del CIP, incluyendo los tipos de datos que son admitidos [6]. A continuación, la capa de Gestión de Conexión y Enrutamiento que permiten la definición de los mecanismos para la transmisión de mensajes a través de múltiples redes, es decir, es una interfaz entre las capas de los protocolos superior e inferior [6]. Adicionalmente en la parte superior de la Figura 1.2 se tienen tres capas adicionales que permiten mejorar la seguridad de este protocolo, las cuales son Perfiles de seguridad, Biblioteca de Objetos de seguridad, y Servicios y Mensajes de seguridad [6]. Finalmente, en las capas inferiores se tienen cuatro protocolos de las capas de red y transporte y que son admitidos por CIP [6].

Los protocolos que pertenecen a esta familia son los siguientes:

- DeviceNet [6].
- ControlNet [6].
- CompoNet [6].
- EtherNet/IP [6], este protocolo será el utilizado en este trabajo de titulación y será detallado en la siguiente sección.

1.3.3.5 Protocolo EtherNet/IP™

EtherNet/IP™ (EIP, por sus siglas en inglés) pertenece a la familia de protocolos CIP™ gestionada por ODVA™ que busca fomentar las tecnologías de información y comunicación abiertas que permitan su interoperabilidad en ambientes industriales [7]. EIP utiliza CIP como su protocolo de capa superior, y extiende las propiedades de Ethernet TCP/IP, lo que le permite trabajar con otros protocolos que se ejecuten en la capa de transporte TCP/UDP estándar y también con otras redes CIP [7].

EIP fue ideado como una solución basada en estándares para la interoperabilidad entre las redes empresariales de fabricación, y permite la conectividad de internet y la empresa en cualquier momento y lugar utilizando switches de internet disponibles comúnmente. Es necesario acotar que el “IP” de EtherNet/IP significa “Industrial Protocol” (Protocolo Industrial) al pertenecer a la familia de protocolos CIP™, no debe ser confundido con el “IP” de TCP/IP que en cambio significa “Internet Protocol” (Protocolo De Internet) [7].

Para la implementación de EIP, solo se requiere de un mecanismo para codificar los mensajes CIP en las tramas Ethernet, gracias a estas tramas y a su estructura no existen limitaciones particulares para implementar EIP [7].

1.3.4 TIPOS DE PÁGINAS HMI

Los sistemas SCADA como se expuso previamente tienen un interfaz gráfica a través de los HMI, que muestran y actualizan los valores de los procesos de acuerdo a su variación a lo largo del tiempo, además manejará las entradas necesarias para los operadores [8]. Los HMI deben comunicarse eficientemente con los elementos necesarios de la planta, que estarán almacenados en una base de datos, añadiendo controladores, creando tags de PLCs, que se refieren a ubicaciones de memoria dentro de dichos PLC, luego de definir dichos elementos se podrán obtener las propiedades necesarias para determinar su estado, como la información obtenida desde el controlador y la ubicación de memoria [8].

A continuación, se presentan las páginas HMI más comunes:

1.3.4.1 Página principal HMI

En esta página se muestran los comandos de inicio, así como los accesos e inicios de sesión [8].

1.3.4.2 Página de Navegación

Esta página es esencial para la aplicación HMI, al permitir acceder rápidamente a las áreas específicas de la aplicación [8]. Desde esta página los operadores pueden moverse de una pantalla a otra a través de botones como: “Schematic”, Base de Datos, Eventos/Alarmas, Informes, sistema, etc [8].

1.3.4.3 Páginas Esquemáticas HMI

Dentro de estas páginas se encontrarán diagramas esquemáticos de la planta, es decir, dentro de estas páginas se simulará el entorno real, como por ejemplo, una imagen de los generadores deberá simular la del entorno real [8]. Para realizar las simulaciones se utilizan distintos objetos, cada uno de ellos representará un elemento de la planta [8]. Se pueden reconocer dos tipos de objetos en este tipo de páginas:

- Objetos digitales, que representan los dispositivos de la planta, y muestran el estado actual de dichos dispositivos a través de colores [8].
- Objetos analógicos, en cambio, son valores que representan las lecturas analógicas, y muestran el estado actual de los dispositivos a través de colores y valores [8].

Suele ser usual que se represente los errores que pueden ocurrir en la planta a través del color rojo, mientras que para representar que los objetos están bajo control se puede usar el color gris, esto significa que un administrador modificó el estado actual del elemento y no se puede asegurar que sea el mismo que el estado actual del elemento en la planta, entonces el objeto permanecerá en gris hasta que el dispositivo remoto acepte o rechace la acción de control y vuelva a enviar el estado en el que se encuentra [8]. A continuación, se muestra un resumen de los colores más utilizados y sus significados en los objetos de las páginas esquemáticas:

Tabla 1.1 Significados de los colores en las páginas esquemáticas [8].

Color	Estado Actual para Objetos Digitales	Estado Actual para Objetos Analógicos
Verde	Operando Normalmente	Operando Normalmente
Magenta	Apagado	-

Rojo	Error Alarma No disponible No se encuentra activo No definido	Bajo Alto
Gris	Bajo Control	Bajo Control Entrada Manual

1.3.4.4 Páginas de Alarmas y Eventos

En estas páginas se muestran tanto alarmas del sistema como eventos, en forma de texto con información acerca de dicho evento o alarma, como la fecha y la hora de la ocurrencia así como el elemento donde fue definida la alarma, además de una descripción de dicha alarma [8]. También los colores tienen sus significados en estas páginas como se muestra en la Tabla 1.2:

Tabla 1.2 Significado de los colores en las páginas de Alarmas y Eventos [8].

Color	Significado
Verde	Evento
Rojo	Alarma No Reconocida
Gris	Enviando Acción de Control Alarma Reconocida

1.3.4.5 Páginas de Tendencias HMI

Las páginas de tendencias mostrarán un cuadro de las lecturas analógicas a lo largo del tiempo, conteniendo las tendencias de las variables que el operador requiera seleccionar para elegir la curva de los valores calculados o los monitoreados [8]. La utilidad de estas páginas surge al poder elegir entre distintos gráficos, y poder seleccionar la escala de tiempo para que sea por distintos intervalos de tiempo como hora, día, semana, etc [8].

1.3.4.6 Páginas de Informe HMI

Las páginas de informe o reporte son muy similares a las páginas de tendencias, sin embargo las páginas de informe tomarán las variables en función del tiempo en un formato de texto y a través de una tabla, los tiempos que suelen ofrecerse en estas páginas son Informe Diario, Semanal, etc [8].

1.3.4.7 Páginas de Sistema HMI

Este tipo de páginas sirven para mostrar los estados de conexión de las computadoras en las estaciones, donde el color verde representará que una computadora está disponible y

conectada y el color rojo un estado de error, dado por que la computadora no está disponible al estar apagada o interrumpida [8].

1.3.4.8 Páginas de Operador HMI

En esta página se muestran en cambio opciones para los operadores como agregar notas en un cuadro de texto donde se registre la fecha y la hora de dicha nota, además de otra página que permite el registro de los nombres de los operadores, el nombre de la estación actual, así como la fecha y la hora [8]. En estas páginas también se administra el inicio y finalización de sesión de los diferentes operadores con sus cuentas, administrando también los turnos del operador en las tablas Existencia del Operador [8].

1.3.5 ESTÁNDARES DE DISEÑO DE APLICACIONES HMI

Los HMI son la interfaz a través de la cual los usuarios tienen acceso al sistema SCADA permitiendo su supervisión, entonces mostrar la información y el contexto correctos para cada uno de los contextos posibles de un proceso permitirá que el operario detecte y responda más eficientemente ante situaciones anormales [9].

Una HMI debe ser diseñada para reducir el tiempo de inactividad, el desperdicio de espacio o de tiempo, así como el mejoramiento de la calidad y la productividad [9]. Al mostrar a los operadores solo la información más necesaria de cada contexto se permitirá una mejor toma de decisiones [9].

1.3.5.1 Convenciones de Colores para HMI

En el entorno de trabajo de aplicaciones industriales donde se usan los HMI, estos deben ser diseñados para alinearse a un modelo mental acorde al flujo de trabajo de tareas asignadas a cada usuario, es decir, los usuarios deben poder hallar la información necesaria, así como los controles con una mínima necesidad en cuanto a reconocimiento de colores [9].

A continuación, se exponen una serie de pautas en cuanto al color para crear pantallas:

- Alarmas: Se recomienda utilizar colores brillantes e intensos, además no deben usarse esos colores para nada más [9].
- Datos en vivo: Se recomienda utilizar colores menos intensos, por ejemplo el azul oscuro o verde oscuro, que distraerán menos y al mismo tiempo permitirán diferenciar claramente los datos de otros estáticos [9].

- Fondo de pantalla: Se recomienda la utilización de colores no saturados, tales como el gris claro que reduzca la interferencia con otros colores utilizados en la interfaz [9].
- Deben evitarse la utilización de colores previamente utilizados en otros objetos, por ejemplo, los asignados para alarmas y datos en vivo. Si se necesita hacer énfasis en algún elemento se debería acentuar el color de línea [9].
- La representación del estado no dependerá únicamente del color, se recomienda la utilización de características adicionales como el relleno, la forma o inclusive un texto simple que determine el estado actual [9].
- Se debe estandarizar la utilización del color, que sea consistente y siempre aplicado de manera rigurosa, además de documentado [9].
- El contraste con el fondo de pantalla debe ser tomado a consideración para evitar la fatiga visual [9].

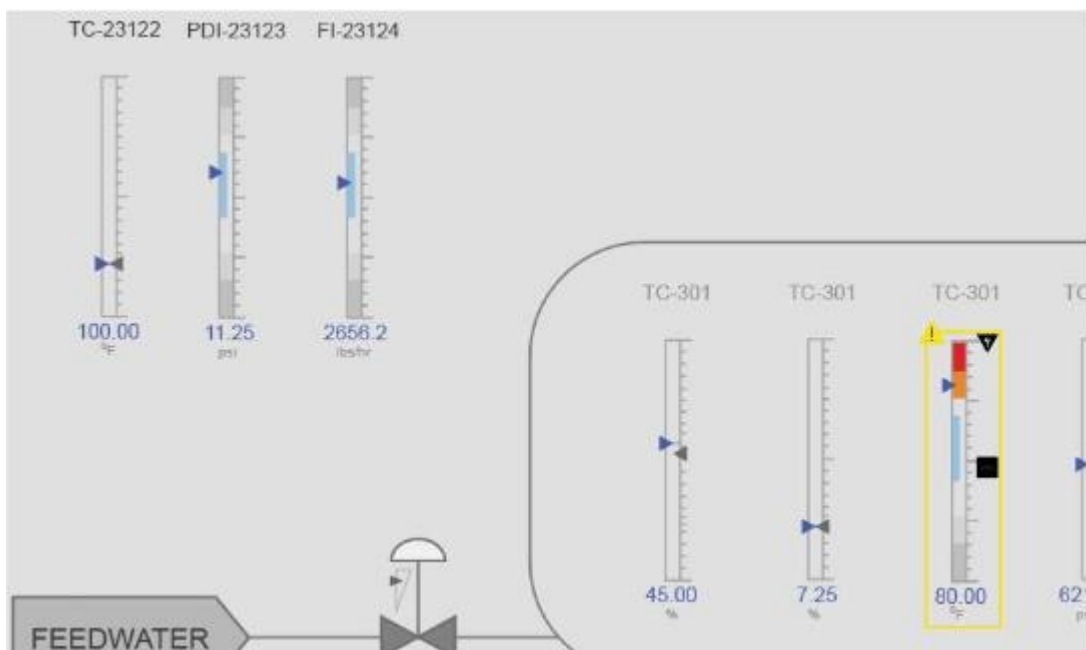


Figura 1.3 Ejemplo de utilización adecuada del color [9].

Como se muestra en la Figura 1.3, el color se ha reservado para ser utilizado en alarmas y en los datos en vivo, se puede notar claramente que las alarmas tienen colores más visibles y llamativos, sin embargo todos los demás objetos tienen tonos de gris para no distraer de la información más importante [9].

1.3.5.2 Animaciones en Objetos HMI

Para mostrar que un objeto en la interfaz requiere de atención inmediata, se puede utilizar la animación de parpadeo o intermitente para un objeto, sin embargo, el texto de dicho objeto deberá acentuarse manteniendo su legibilidad, y se sugiere utilizar un borde u otro objeto parpadeante junto al texto [9]. Además, se debe proporcionar una opción al operador para detener el parpadeo cuando se percate del problema [9].

1.3.5.3 Tipos de Fuente y Tamaños de Fuente

Las pantallas HMI deben mostrar textos claros y fáciles de leer, permitiendo a los usuarios comprender la navegación, a continuación se mostrarán recomendaciones clave para tipos y tamaños de fuente [9]:

- Se recomienda utilizar la fuente Sans Serif para una mejor legibilidad [9].
- Se debe utilizar un tamaño de fuente uniforme tanto para los datos, títulos, cabeceras. Mientras que las etiquetas y anotaciones pueden ser más pequeñas [9].
- Datos prioritarios deben ser más grandes y en negrillas [9].
- Para otros datos que se busquen destacar se deben utilizar fuentes que sean más grandes que los de las etiquetas [9].

Es necesario comentar también que los tamaños de fuente dependerán en gran medida de donde se estén mostrando y como los usuarios están viendo el monitor, se necesitarán fuentes más grandes si dichos monitores estarán muy lejos de los usuarios [9].

1.3.5.4 Texto Estático

Este término se refiere a los textos que no cambiarán en la interfaz durante su funcionamiento, aquí se incluyen los textos que pueden ser alterados por los usuarios con los permisos suficientes, pero que no se modificarán con frecuencia [9].

A continuación, se muestran recomendaciones generales para los textos estáticos:

- Se debe reducir la utilización de abreviaturas ya que traducirlos a otros idiomas puede ser difícil [9].
- El tamaño debe ser proporcional a la jerarquía del texto, así como el tamaño del monitor [9].
- Debe diseñarse los espacios de los textos tomando en cuenta la traducción a otros idiomas [9].

1.3.5.4.1 Títulos

Los títulos son los textos que se muestran en la parte superior de la pantalla o de la página e identifica o describe el contenido de estos [9].

A continuación, se muestran recomendaciones generales para los títulos:

- Se debe utilizar una fuente en negrillas [9].
- Se debe usar un tamaño de fuente grande [9].
- Los títulos deben estar en la misma posición en cada pantalla para que los usuarios puedan identificarlo más rápidamente [9].

1.3.5.4.2 Cabeceras de Columnas

Estos textos se muestran en la parte superior de una columna de datos o controles, su función es describir el contenido que viene debajo de ellos [9].

Para estos textos se podría recomendar, por ejemplo:

- Utilizar una fuente en negrilla [9].
- Se debe utilizar subrayado, delimitando que es el encabezado de los datos que vienen después [9].
- La primera columna debería tener un encabezado con justificación hacia la izquierda [9].
- Columnas anchas deben también tener títulos justificados a la izquierda [9].
- Las comunas estrechas, en cambio, deben tener los encabezados justificados en el centro [9].
- Los títulos de las columnas siguientes deben utilizar la misma justificación [9].

1.3.5.4.3 Cabeceras de Grupos

Las cabeceras de grupos permiten delimitar y describir un conjunto de datos o controles en la pantalla [9]. Las cabeceras de grupos deben ser justificados hacia la izquierda con una fuente en negrillas [9].

1.3.5.4.4 Etiquetas

Las etiquetas son cualquier tipo de texto descriptivo y estático dentro de la interfaz para elementos como los datos, grupos, títulos de la pantalla, etc [9]. Las etiquetas para los datos deben justificarse a la izquierda, mientras que los datos deben estar al a derecha o debajo de dicha etiqueta [9].

A continuación, se presentarán recomendaciones para la utilización correcta de etiquetas:

- Se debe utilizar texto gris oscuro, más no negro [9].
- Se debe utilizar la fuente Sans Serif [9].
- Si se opta por otra fuente, esta deberá ser legible y tener un tamaño adecuado [9].
- La fuente elegida debe ser utilizada para todas las fuentes del diseño, añadiendo coherencia [9].
- Las palabras en mayúsculas deben reservarse para palabras aisladas, títulos, etiquetas breves, y designaciones de equipos [9].
- Para los demás casos se debe utilizar mayúsculas y minúsculas para aumentar la legibilidad de las etiquetas [9].
- Se recomienda para distinguir el texto utilizar diferentes tamaños o usando diferentes colores, es decir, diferentes tonos de gris [9].

Además de las recomendaciones previamente expuestas, los nombres de los elementos deben ser claros y entendibles, es decir, no se deben referir a un tanque como ABDFE-MBR-5342 incluso si esa es su etiqueta, deberían referirse a él como “Tanque A” o “Tanque de Filtrado C”, siendo más fácil de entender [9].

1.3.5.5 Datos Dinámicos

Como se dijo anteriormente, los datos deben presentarse de manera directa, y clara evitando distracciones, es decir, se debe proporcionar lo necesario y eliminar todo lo demás [9]. Un ejemplo es mostrar solamente el nivel de precisión necesario para que el usuario pueda trabajar de forma efectiva, es decir, si se requieren solo 2 decimales de precisión no debería mostrarse 3 o más [9]. La representación gráfica de los gráficos es más que necesaria y útil ya que permite que el usuario absorba la información con un vistazo rápido [9].

1.3.5.5.1 Datos de texto Dinámico

Los datos de texto dinámico deben usarse en el mínimo de los casos, ya que llegan a ser difíciles de entender con un vistazo rápido y requieren un nivel elevado de atención y enfoque [9].

Recomendaciones para datos de texto dinámicos:

- Las etiquetas deben estar a la izquierda o en la parte superior de los datos.

- Reducir el uso de abreviaciones ya que pueden dificultar la traducción a otros idiomas.

1.3.5.5.2 *Datos Numéricos Dinámicos*

Al igual que los datos de texto dinámicos este tipo de datos también son difíciles de entender de un vistazo rápido y requieren un nivel elevado de enfoque [9].

Recomendaciones para datos numéricos dinámicos:

- Las etiquetas deben estar a la izquierda o encima de los datos dinámicos [9].
- Este tipo de datos al ser numéricos deben tener sus correspondientes unidades de ingeniería, excepto si el contexto elimina esta necesidad o si el valor no tiene unidades [9].
- Es recomendable que las unidades de ingeniería aparezcan a la derecha o debajo de los datos numéricos dinámicos [9].
- Los datos numéricos del tipo entero deben evitar la utilización de un punto decimal [9].
- Si datos numéricos tienen como objetivo su comparación, estos deben justificarse, alineando sus puntos decimales [9].

1.3.5.5.3 *Gráficos de Barra*

Los gráficos de barra permiten mostrar una sola cantidad relacionándola con su límite, su rango o los umbrales de control y alarma, también se pueden utilizar varios gráficos de barras para representar una relación rápida y realizar comparaciones [9].

Las ventajas de este tipo de gráficos es que permite dar contexto en un espacio pequeño, al comparar el valor actual de los datos y su límites, con los gráficos de barra también se puede mostrar medidas que se asocian con elementos discretos [9]. Las desventajas en cambio son que no proporcionan datos históricos y su precisión numérica no es tan alta como el de una pantalla [9].

Elementos requeridos por un gráfico de barras:

- Contenedor de la barra y forma de fondo [9].
- Indicador del valor actual [9].

Componentes de categoría opcional para el gráfico de barras

- Nombre definido por el usuario [9].

- Nombre de etiqueta [9].
- Punto de alarma alta alta [9].
- Punto de alarma baja [9].
- Punto de alarma baja baja [9].
- Umbral alto, alto sin alarma [9].
- Umbral bajo sin alarma [9].
- Umbral bajo bajo sin alarma [9].
- Capacidad para configurar el número de marcas de graduación [9].
- Rango de funcionamiento normal [9].
- Set Point [9].
- Rango de funcionamiento histórico [9].
- Valor numérico del valor actual [9].
- Unidades de ingeniería [9].
- Indicación de alarma: que muestra la prioridad, así como el estado de la alarma. Las alarmas no reconocidas deben parpadear en gris, y las reconocidas deben mostrar continuamente el borde del color de la alarma [9].
- Notificaciones que permiten comunicación con los usuarios acerca de situaciones a normales que no son situaciones alarmantes [9].

A continuación, se presenta una imagen que muestra los elementos de un gráfico de barras.



Figura 1.4 Elementos requeridos y opcionales de un gráfico de barra [9]

Recomendaciones para gráficos de barra:

- El valor actual de la cantidad medida por el gráfico de barras debe mostrarse al lado o cerca de dicho gráfico, se debe procurar colocarlo debajo cuando sea posible [9].
- Los rangos de alarmas deben cambiar de colores para indicar la gravedad de las alarmas. Se debe ser consistente y coherente con los colores elegidos para otras alarmas anteriormente [9].
- Los gráficos de barras pueden tener dos orientaciones vertical y horizontal [9].
- Se recomienda siempre utilizar etiquetas que permitan la identificación del gráfico, y deberá estar directamente encima o debajo del gráfico [9]. La excepción es cuando el gráfico pertenece a un grupo que ya tiene una etiqueta [9].

Como se mencionó anteriormente se puede agrupar varios gráficos de barras para realizar comparaciones de datos relacionados.

Recomendaciones para grupos de gráficos de barra:

- El grupo de gráficos debe tener una etiqueta de grupo [9].
- Se deben agrupar solo gráficos con datos que se relacionen entre sí [9].
- Si todos los gráficos comparten unidades solo se mostrarán dichas unidades una sola vez [9].
- Los gráficos de barras que pertenezcan al grupo deben tener una identificación para diferenciarlos entre ellos [9].
- Los gráficos que pertenecen al grupo pueden mostrar escalas individuales si es que no comparten un rango similar [9].

1.3.5.5.4 *Tendencias*

Los gráficos de tendencias como se expuso anteriormente permiten mostrar una variable a lo largo del tiempo, y sirven para comparar valores similares o relacionados [9].

La ventaja principal de los gráficos de tendencias es que permiten mostrar información histórica lo que habilita a los usuarios comprender el estado actual y el futuro de ese elemento, en cambio su desventaja es que puede utilizar mucho espacio de la interfaz [9].

Los elementos requeridos dentro de un gráfico de tendencia son:

- Título de la tendencia

- Indicador del valor actual, es decir, las líneas de tendencia donde cada línea se diferenciará por un color diferente.

Existen elementos opcionales que son muy similares a los del gráfico de barras, como los puntos de alarma en sus distintos rangos, los rangos sin alarma, el “setpoint”, indicadores de alarmas, notificaciones y valores numéricos los demás elementos serían:

- Eje X y Eje Y [9].
- Límite mínimo y máximo para el eje Y [9].
- Escala para los dos ejes, como marcas de graduación o valores numéricos [9].
- Indicación de rango histórico [9].
- Navegación de tendencias [9].
- Reestablecer rango histórico [9].
- Funcionalidad de Panorama y Zoom [9].

Recomendaciones para los gráficos de Tendencias:

- Los gráficos de tendencias pueden tener datos históricos o en tiempo real [9].
- El eje Y permite indicar un límite mínimo y/o máximo [9].
- El rango de escala cambia según la selección del usuario [9].
- Los títulos o leyendas pueden representarse a través de texto, símbolos o ambos [9].
- La posición de las etiquetas dependerá del espacio disponible [9].

1.3.5.6 Iconos

1.3.5.6.1 Tamaño

El tamaño de los iconos dependerá de si estos son botones o informativos, si son botones se recomienda un tamaño de 32x32, en cambio para los de información se recomienda 16x16 [9].

1.3.5.6.2 Color

Como se expuso anteriormente acerca del color, los iconos también deben cumplir con la coherencia estándar del diseño, es decir, si existen iconos de alarma estos deberán utilizar los mismos colores que los previamente asignados para alarmas en otros objetos, y solo

estos podrán tener esos colores. Los iconos utilizados para la navegación no deberán tener ningún color, a menos que se refieran a una advertencia o problema [9].

1.3.5.7 Control de Entradas

Cada usuario tendrá su conjunto de tareas con objetivos que requerirán una serie de controles diferentes entre sí, se debe buscar dar solo las capacidades requeridas para lograr sus tareas [9]. Se debe buscar garantizar la coherencia entre los controles como [9], por ejemplo:

- Entre controles similares se deben tener métodos igualmente similares para ingresar información o comandos [9].
- La redacción, la ubicación del área de entrada y de información, así como la ubicación de los controles de gestión de interfaz deben ser coherentes entre sí, se busca así minimizar las demandas de los usuarios [9].
- No debería ser parte del comportamiento de la interfaz que el usuario requiera que recuerden códigos o secuencias específicas ni tampoco que realicen traducciones o conversiones [9].
- Se debe buscar automatizar al máximo en cuanto a lo que se requiere por parte del usuario [9].
- Se debe asegurar que el proceso de ingresar comandos sea lógico e intuitivo [9].
- Se debe distinguir claramente entre los campos de entrada y los controles del texto estático, el usuario no debe buscar o cuestionar si un control lo es o no [9].
- Se debe permitir la capacidad de editar la entrada antes de aplicarla al sistema [9].
- Ejecutar acciones con consecuencias significativas deben implementar mecanismos de confirmación para evitar una activación no deseada [9].

Los usuarios deben saber cuándo el sistema ha reconocido su comando y también el estado en el que está luego de dicho comando, cuando uno de estos comandos requiere un procesamiento prolongado, se debe mostrar comentarios para que los usuarios sepan que el sistema se encuentra en funcionamiento, evitando que sigan haciendo clics en el comando o la entrada en búsqueda de una retroalimentación [9]. Entonces cuando el proceso finaliza también debe mostrarse un comentario [9].

Al presionar un botón el sistema debe proporcionar información de que dicho botón fue presionado, esto puede ser a través del mismo botón o un control cerca de dicho botón indicando el estado del objeto [9].

Los controles que tienen la opción de deshabilitarse deben tener un claro estado habilitado donde se pueden ejecutar comandos o ingresar datos, y también un estado deshabilitado donde dichas opciones no están disponibles [9]. Se deben evitar también los botones con solo dos estados, es decir, un objeto motor puede tener un botón de “Inicio” y otro de “Parar” en vez de uno solo que cambie de etiqueta, con esto se busca evitar la operación no deseada por los usuarios al presionar el control varias veces [9].

Recomendaciones para el control de entradas:

- La acción de pasar el ratón sobre un objeto debe indicar que dicho objeto es un control procesable [9].
- Un control debe poder indicar que tiene la opción de desactivarse en función de la seguridad [9].
- Si un control puede deshabilitarse su apariencia también deberá cambiar en este estado [9].

1.3.5.7.1 Tamaño de Entrada

Los tipos de interacción que un HMI puede tener, sea teclado, mouse o un toque táctil afectarán el tamaño de las “hit zones” o zonas de impacto, así como la separación y espaciado entre los objetos [9]. Un toque táctil necesita de una mayor zona de impacto si se lo compara con el mouse y el teclado [9].

Se debe buscar que la biblioteca de objetos se dimensione para poder ser utilizado en entornos táctiles siguiendo las siguientes recomendaciones:

- Botones de comandos: La zona de impacto, para que un usuario interactúe debe tener mínimamente 40 píxeles de alto x 40 píxeles de ancho. El ancho puede variar si el texto supera el tamaño del botón [9].
- Botones de Navegación: Mínimo 35x35 píxeles, mientras que los botones de ayuda y los botones de los gráficos de tendencia son 32x32 píxeles [9].
- Los botones de paginación: 36 de alto x 26 de ancho en píxeles [9].
- Campos de entrada: 20 píxeles de alto y el ancho variará con la longitud de la entrada [9].

- “Checkboxes” y radio botones: Requieren una zona de impacto de 20 píxeles de altura para el caso de uso de mouse y 30 píxeles de altura para el caso de uso de toque táctil. En cambio, el ancho mínimo es de 20 píxeles para uso del ratón sin etiqueta, 30 píxeles de ancho para toques táctiles sin etiquetas [9].
- Controles con etiquetas: Deberán incluir el control y la etiqueta en la zona de impacto [9].
- Botones Toggle(Alternadores): Requieren una altura de 28 píxeles y un ancho que incluya la etiqueta de alternancia [9].
- Área de toque mínima para un objeto en la pantalla: 30 x 30 píxeles es el mínimo absoluto, por defecto se sugiere 40 x 40 píxeles, incluyendo el objeto entero que puede ser más grande [9].
- Botones comandos deben tener 10 píxeles de espaciado entre ellos [9].
- Los botones de navegación deben tener 2 píxeles de espaciado [9].

1.3.5.7.2 Botones de Comando

Estos botones han sido diseñados para ser utilizados con la entrada de operadores booleanos que devuelven un cambio de estado o modo de funcionamiento, cada estado deberá tener su propio botón de comando independiente [9].

Recomendaciones para la implementación de botones de comando:

- El tamaño mínimo del área de contacto será tan grande como la punta de un dedo para que un operador no falle en la presión de un botón [9].
- La zona táctil puede ser más grande que lo anterior expuesto pero los límites, es decir, las separaciones deben mantenerse [9].
- Se deben colocar separaciones entre los botones de comando para evitar así que se ejecuten operaciones no intencionadas [9].
- La apariencia de estos botones debe diferenciarse claramente de los de navegación evitando que se emita un comando cuando un usuario tenía intención de navegar a otra pantalla [9].
- La apariencia de los botones debe diferenciarse claramente cuando están seleccionados o presionados [9].
- Texto o ícono en el botón:

- Debe mostrar el comando que se va a iniciar, no el estado actual del botón o de su variable asociada [9].
- Debe estar justificado en el centro [9].

Los comandos bien conocidos como Guardar, Restaurar, Descargar, Crear, Eliminar o Renombrar deben ser tratados como botones de comando [9].

1.3.5.7.3 Botones de Navegación

Se llaman de navegación por que estos botones permiten moverse a través de las distintas páginas, pantallas, o ventanas dentro del diseño del HMI [9].

Las recomendaciones para la implementación de botones de navegación son muy similares a las de los botones de Comando, añadiendo otras como:

- Los botones de navegación entre pestañas y páginas deben variar su apariencia mostrando que pestaña o página está en uso [9].
- El texto de la pestaña, página o el botón debe referirse al destino de la navegación [9].

1.3.5.7.4 Entrada Directa de Datos

La entrada de directa de datos se realiza a través de un campo editable dentro de la pantalla o a través de una ventana emergente que se ejecuta al dar clic sobre el valor que se desea modificar [9].

Recomendaciones generales para la entrada de datos directa:

- Se debe evitar que los usuarios ingresen datos que no estén dentro de los rangos válidos para cada variable [9].
- La entrada de datos analógicos debe mostrar a sus usuarios la unidad de ingeniería si es requerida [9].
- Se debe mostrar el valor máximo y el valor mínimo admitidos del valor que se está modificando [9].
- Las etiquetas y las unidades de ingeniería se mostrarán, pero con una fuente más pequeña que los datos, es decir, los datos deben destacar siempre [9].
- Se deben administrar las excepciones, los desbordamientos y los subdesbordamientos cuando se refiere a datos numéricos de punto flotante [9].

1.3.5.7.5 Controles deslizantes (Sliders)

Los controles deslizantes permiten que el usuario cambie los datos analógicos, a través de una representación en forma de objeto gráfico, existen dos tipos de controles deslizantes:

- De impacto inmediato [9].
- De impacto luego de alzar el ratón y/o realizar la verificación [9].

Recomendaciones generales para la implementación de controles deslizantes:

- Los controles deslizantes se utilizarán para la modificación de datos analógicos, pero también pueden usarse para evitar la utilización de un campo de entrada de datos [9].
- El indicado mostrará la posición actual del control, y también mostrará sus límites [9].
- El control deslizante debe poder diferenciarse claramente de un gráfico de barras [9].

1.3.5.7.6 Entradas de texto

Este tipo de entradas deben incluir las descripciones apropiadas, validaciones y etiquetas.

Recomendaciones para la implementación de entrada de texto:

- Los campos de entrada basarán su tamaño en su jerarquía, así como el tamaño de la pantalla [9].
- Estos objetos deben tener contexto, para ello se puede mover la entrada colocándolo cerca de un objeto o de una etiqueta estática [9].
- Las etiquetas de texto deben mostrarse a la izquierda o también pueden estar encima del campo de entrada de texto [9].

1.3.5.8 Configuraciones de Seguridad

Recomendaciones para implementar configuraciones de seguridad:

- Los controles deben mostrar visualmente cuando los usuarios tienen acceso restringido a un objeto debido a la seguridad [9].
- La seguridad puede restringir a los usuarios por su rol, también puede restringirlo dependiendo de si está calificado o no para una función, finalmente se puede restringir por la ubicación del dispositivo HMI [9].

El acceso a los distintos controles, así como las funciones del HMI debe administrarse mediante la implementación de los siguientes escenarios de seguridad:

- Reconocimiento y Restablecimiento de Alarmas [9].
- Configuración de alarma [9].
- Deshabilitar alarmas [9].
- Archivar alarmas [9].
- Anular permisos e interbloqueos [9].
- Poner el dispositivo en servicio o fuera de el [9].
- Cambiar configuraciones de seguridad del dispositivo [9].
- Cambiar configuraciones avanzadas del dispositivo [9].
- Modificar tiempos de retardo de alarma [9].
- Cambiar configuraciones de la interfaz HMI del dispositivo [9].
- Configurar los límites de los dispositivos [9].
- Configurar temporizadores de dispositivos [9].
- Cambiar los parámetros de configuración [9].
- Poner el dispositivo en modo simulación [9].
- Ingresar puntos de ajuste y variables de control [9].
- Anular entradas [9].
- Anular salidas [9].
- Procesamiento de excepciones [9].
- Anulación de parámetros de fase descargados [9].
- Anulación de “setpoints” previamente descargados [9].
- Seleccionar, ejecutar, mantener y reiniciar procedimientos, secuencias y ejecuciones por lotes [9].
- Control manual del control de supervisión [9].

- Forzar estados [9].
- Restablecer acumuladores de tiempo de ejecución [9].
- Responder las indicaciones [9].

Existen tres tipos de seguridad a ser considerados los cuales se expondrán en los siguientes puntos.

1.3.5.8.1 Seguridad basada en el rol

Este tipo de seguridad brinda a los usuarios acceso a distintos controles, así como diferentes variables y datos basándose en el rol que tienen dentro de la planta, ejemplos de roles pueden ser operador, técnico de mantenimiento, ingeniero, etc [9]. Cada planta tendrá diferentes roles, por lo que cada rol verá y tendrá disponible diferentes opciones, aquellos controles deshabilitados se mostrarán entonces en escalas de grises para diferenciarse de los controles disponibles [9].

1.3.5.8.2 Calificación del usuario

Cuando existen plantas más grandes, los usuarios están limitados a tener acceso únicamente a sus propias áreas de trabajo, es decir, un operador solo podrá operar equipos en un área de la planta pero no fuera ni de otra planta diferente [9].

1.3.5.8.3 Ubicación de la estación de trabajo

En sistemas más grandes, donde existen múltiples estaciones de trabajo, las operaciones del equipo se restringirán solo a ciertas áreas de trabajo o estaciones de trabajo, un ejemplo claro de esto es el “Control de línea de visión” que tendrá su estación de trabajo justo dentro del sitio que se encuentra controlando [9].

1.3.6 FUNCIONALIDAD DE ALARMAS

La funcionalidad de alarmas en los HMI es imprescindible ya que permite atraer la atención de los usuarios a través de una notificación acerca de condiciones anormales que deben ser atendidas en breve [9].

La aplicación debe tener un apartado para observar las alarmas más recientes, así como las de mayor prioridad, los usuarios deben poder administrar las alarmas, interactuando con ellas y filtrándolas a través del resumen de alarmas [9].

Para proporcionar contexto las alarmas deben mostrarse junto al elemento que las ejecutará, además las alarmas no reconocidas utilizarán una animación parpadeante hasta que sean reconocidas por el usuario [9].

1.3.6.1 Configuración adecuada del resumen de alarmas

El resumen de alarmas es un área que puede expresarse en forma de una sección de la interfaz HMI o una ventana emergente, permite a los usuarios ver e interactuar con un resumen de cada una de las alarmas y eventos ejecutándose en el área de proceso [9]. Dentro del resumen de alarmas los usuarios pueden reconocerlas, suprimirlas, deshabilitarlas, archivarlas o quitarlas, además se ofrece una opción de filtrado para el proceso actual, es decir, si el usuario cambia el área de proceso el resumen de las alarmas se actualizará automáticamente [9].

Además del resumen de alarmas pueden existir otras secciones como las siguientes:

- Historial de alarmas: Como se puede deducir de su nombre, aquí se mostrarán los datos históricos de las alarmas [9].
- Alarmas Archivadas: Aquí se mostrará la lista de alarmas definidas como archivadas proporcionando opciones para quitarlas [9].
- Explorador de Alarmas: En esta sección se mostrarán todas las alarmas existentes, así como el estado actual en el que se encuentran y la capacidad para habilitarlas o deshabilitarlas, eliminarlas, archivarlas y también poder ver los comentarios del operador [9].

1.3.6.2 Representación adecuada de alarmas

Las alarmas deben ser claramente diferenciables de otros objetos a través del color, la forma, y el ícono que indique la existencia de una alarma, además de dichos componentes visuales, también se puede mostrar un borde alrededor del componente [9]. El color del borde coincidirá necesariamente con el color de la prioridad y del ícono de alarma, después con alarmas no reconocidas el borde parpadeará entre un borde gris y un borde de color [9]. Cuando se reconozca la alarma, el borde permanecerá con el mismo color hasta que dicha alarma sea atendida y resulta [9].

1.4 HERRAMIENTAS, FRAMEWORKS Y LIBRERÍAS DE DESARROLLO

1.4.1 IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) especializado para lenguajes de la Máquina Virtual de Java (JVM, por sus siglas en inglés), su diferenciación es automatizar las tareas repetitivas en el desarrollo, para que el programador se enfoque en la escritura de código, y también ofrece tareas para la

finalización inteligente de código, con opciones para el análisis de código estático y refactorizaciones [10].

1.4.2 MySQL

MySQL es de las bases de datos más populares, es de código abierto, reconocida como una de las bases de datos más confiables, permite organizar los datos en una o más tablas y sus tipos se relacionan entre sí permitiendo entonces estructurar los datos, por ende permite el lenguaje SQL para la creación, modificación, y extracción de datos desde la base de datos [11].

1.4.3 MySQL Connector/J

MySQL Connector/J es una librería para la conectividad de MySQL con aplicaciones que sirven de cliente y están desarrolladas en el lenguaje de programación Java, implementando la API de Conectividad de Base de Datos en Java (JDBC, por sus siglas en inglés) que proporciona funciones de registro automático a través del Driver Manager, comprobaciones para realizar validaciones estandarizadas, excepciones SQL categorizadas, soporte para grandes actualizaciones, soporte para variantes de fecha y hora locales, etc [12].

1.4.4 JavaFX

JavaFX es un conjunto de paquetes gráficos y medios para el diseño, creación, pruebas e implementaciones de aplicaciones de cliente que funcionarán de manera consistente en diversas plataformas, al tener compatibilidad multiplataforma ofrece una experiencia en ejecución coherente para los desarrolladores y usuarios de aplicaciones de estas aplicaciones [13].

1.4.5 Git

Git es un sistema de control de versiones de código abierto creado por Linus Torvalds, permite crear ramas, fusionarlas y reescribir el historial del repositorio, permitiendo varios flujos de trabajo coexistiendo a la vez, también permite a los equipos de desarrollo revisar de manera eficiente el código de los demás, se considera a este software como el estándar moderno para el desarrollo de software [14].

1.4.6 GitHub

GitHub es una empresa que ofrece un servicio del mismo nombre para almacenar repositorios Git en la nube, su principal característica es la facilidad que ofrece a las personas y equipos para usar Git tanto para el control de versiones y para la colaboración,

si no existiese GitHub, la utilización de Git requeriría de más conocimientos técnicos así como la utilización de la línea de comandos [15].

1.4.7 Método de Priorización MoSCoW

El método de priorización MoSCoW es una técnica para definir la priorización de tareas y así permitir que los Stakeholders comprendan la importancia a cada entrega de un requisito [16]. El término MoSCoW es un acrónimo que deriva desde la primera letra de cada categoría de priorización definida dentro de esta técnica a continuación se muestra el significado de cada categoría [16]:

- Mo: Significa “Must Have” se refiere a los requerimientos que son críticos y deben ser implementados en un producto de manera prioritaria, incluso si uno de los requerimientos con esta prioridad no se ha tomado en cuenta el despliegue se considera incumplido [16].
- S: Significa “Should” se refiere a requerimientos que son importantes, pero no críticos para el despliegue [16].
- Co: Significa “Could Have” se refiere a requerimientos deseables pero que no son obligatorios para el despliegue, en general son mejoras de bajo costo [16].
- W: Significa “Won’t Have” se refiere a los requerimientos menos críticos o agregan poco valor que no corresponde a la estrategia del producto, pueden ser ignorados y revisarse en futuros despliegues [16].

1.4.8 SCRUM

Scrum es un marco de trabajo que permite organizar personas para enfrentar problemas complejos de adaptación, para entregar productos de manera creativa y productiva generando el mayor valor posible, para este proceso se requiere de un rol llamado Scrum Master que pueda fomentar un entorno donde [17]:

- Un Product Owner que será quien defina el trabajo para el problema complejo a través de un Product Backlog [17].
- El Equipo Scrum que convertirá las tareas definidas del Product Backlog en un incremento de valor durante un Sprint [17].
- El Equipo Scrum y los Stakeholders validarán los resultados y a través de la inspección permitirá al equipo ajustarse al próximo Sprint [17].
- El proceso se repetirá hasta terminar con el problema complejo [17].

Scrum utiliza una serie de elementos para su funcionamiento como:

1.4.8.1 Sprint

Los Sprints son eventos de duración fija de hasta un mes, un sprint comienza inmediatamente después de que un anterior sprint haya terminado [18]. Entonces durante el Sprint:

- No se realizan cambios que alteren y pongan en peligro el objetivo del Sprint [18].
- La calidad del incremento no debe disminuir [18].
- El Product Backlog se adecua y redefine según se requiera [18].
- El alcance también se puede redefinir y renegociar con el Product Owner a medida que se aprende del proyecto [18].

1.4.8.2 Sprint Planning

El Sprint Planning establece el trabajo que se ejecutará dentro del Sprint, esta planificación se define colaborativamente con todo el Equipo Scrum, el Product Owner debe asegurarse de que los integrantes tengan el conocimiento suficiente para discutir los elementos del Product Backlog y como estos definirán el objetivo del Sprint [19].

El Sprint Planning busca responder las siguientes preguntas:

1. ¿Por qué es valioso este Sprint? [19].
2. ¿Qué se puede hacer en este Sprint? [19].
3. ¿Cómo se realizará el trabajo elegido? [19].

1.4.8.3 Scrum Diario

El Scrum Diario es una pequeña reunión que se realiza con los desarrolladores del Equipo Scrum, debe durar no más de 15 minutos, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del Sprint, y se busca mejorar las comunicaciones, identificar los problemas, además de promover la toma de decisiones rápidas, y por ende eliminando la necesidad de otro tipo de reuniones [20].

1.4.8.4 Sprint Review

El Sprint Review tiene como propósito inspeccionar el resultado del Sprint, para realizar adaptaciones en futuros Sprint, el equipo presenta los resultados de su trabajo a las partes interesadas y comenta acerca del progreso hacia el objetivo del producto, con base a la información obtenida los participantes discuten sobre que se debe hacer después, incluso

el Product Backlog también se puede ajustar para encontrar nuevas oportunidades [21]. El Sprint Review es el penúltimo evento del Sprint con un límite de 4 horas como máximo para un Sprint de un mes, en Sprint más cortos el Review también lo será [21]. El Sprint Review incluirá los siguientes elementos:

- Los participantes incluirán el Equipo Scrum y las partes interesadas [21].
- Los miembros del Equipo Scrum explicarán que elementos del Product Backlog se han “Hecho” y cuáles no [21].
- Los Desarrolladores discuten lo que salió bien durante la realización del Sprint, que problemas surgieron y como se los resolvieron [21].
- Los Desarrolladores demuestran el trabajo realizado y responden las dudas que se tengan sobre el incremento [21].
- El Product Owner realiza un análisis de los trabajos pendientes del producto tal como está, entonces se realizan proyecciones de las fechas de entrega y los objetivos probables según el progreso hasta la fecha [21].
- Todo el grupo de Scrum colaborará sobre qué hacer a continuación, de modo que el Sprint Review proporcionará información sumamente necesaria para la próxima planificación del Sprint [21].
- Se realiza una revisión de como el mercado o el uso potencial del producto podrían cambiar o influir lo que es más valioso y que se puede hacer a continuación con el desarrollo [21].
- Se realiza una revisión de la línea de tiempo, el presupuesto, las capacidades potenciales, así como el mercado para las siguientes versiones a ser desarrolladas buscando mejorar su funcionalidad y capacidad del producto [21].

Como resultado del Sprint Review es un Product Backlog revisado que define los elementos para el próximo Sprint [21].

1.4.8.5 Sprint Retrospective

El Sprint Retrospective permite la planificación de formas que permitan aumentar la calidad y la eficacia, esto lo hace el Equipo Scrum inspeccionando el último Sprint tomando en consideración las personas, sus interacciones, los procesos, así como las herramientas y su definición de realizado [22]. El equipo Scrum se encargará de identificar los cambios que resulten de mayor utilidad para perfeccionar su efectividad, y dichos cambios que sean

más influyentes e impactantes deben ser abordados en cuanto antes, inclusive siendo agregados al Sprint Backlog del próximo Sprint [22]. El Sprint Retrospective es la conclusión del Sprint y se encarga de responder y analizar los siguiente:

- ¿Qué salió bien del Sprint [22]?
- ¿Qué se puede mejorar [22]?
- ¿Qué nos comprometemos a mejorar en el próximo Sprint [22]?

1.4.8.6 Product Backlog

El Product Backlog es una lista emergente y priorizada de todo lo necesario para mejorar el producto, es entonces la fuente la fuente de trabajo principal para el Equipo Scrum, es así como sus elementos una vez que pueden ser realizados por el Equipo Scrum dentro de un Sprint son considerados entonces listos para la selección en un evento de Sprint Planning [23]. Para llegar a un estado de listo para la selección los elementos deben pasar por un proceso conocido como refinamiento, en el cual se divide en partes y se define aún más los elementos del Product Backlog generando elementos más pequeños y precisos [23]. El Product Owner tiene una responsabilidad muy alta dentro del desarrollo del Product Backlog ya que influyen en los desarrolladores ayudándolos a comprender y seleccionar las soluciones adecuadas [23]. En síntesis, el Product Backlog permite definir “que” cumplirá con el Product Goal, es decir, con el objetivo del producto que es un objetivo a largo plazo del Scrum Team, este objetivo debe ser cumplido o abandonado antes de asumir el siguiente [23].

1.4.8.6.1 Priorización del Product Backlog

La priorización es uno de los aspectos más importantes para elegir lo correcto que permite maximizar el valor entregado en un Sprint, es decir, los elementos del Product Backlog deben ordenarse para que los requerimientos que añaden mayor valor se completen primero, permitiendo al Scrum Team trabajar conjuntamente en dirección a un objetivo común [16]. A continuación, se presenta una figura que representa el flujo de trabajo de Scrum con un Product Backlog priorizado [16]:

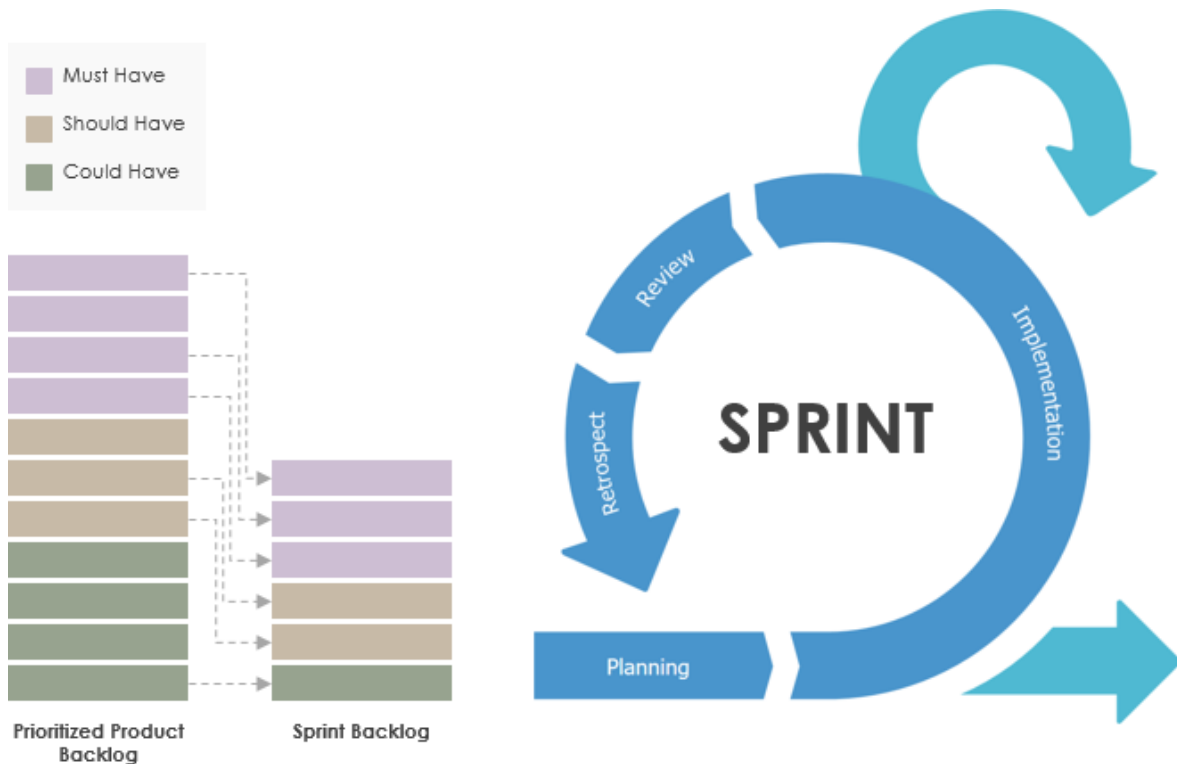


Figura 1.5 Ejemplo de Priorización del Product Backlog [16].

1.4.8.7 Sprint Backlog

El Sprint Backlog busca responder el Sprint Goal, a través de un subconjunto de elementos que pertenecen al Product Backlog para ser realizados dentro del Sprint en ejecución, así como un plan que permita la realización y entrega del incremento [24].

1.4.8.7.1 Sprint Goal

Es el único objetivo que guía la realización del Sprint, se entiende como un compromiso de los desarrolladores, debe proporcionar una gran flexibilidad en términos del trabajo necesario para su realización, además siguiendo las actividades para cumplir el Goal se mantiene una coherencia y un enfoque [24]. El Sprint Goal es creado durante el Sprint Planning y para luego filtrar los elementos añadidos al Sprint Backlog, a lo largo del tiempo que los desarrolladores trabajan durante el Sprint siempre deben tener en mente el Sprint Goal, en el caso de que el trabajo resulte diferente a lo esperado, se debe colaborar con el Product Owner para redefinir el alcance del Sprint Backlog dentro del mismo Sprint sin afectar así al Sprint Goal [24].

1.4.8.7.2 Incremento

Se determina a un incremento como un paso concreto que acerca al Scrum Team hacia el Product Goal, cada incremento nuevo se adiciona al anterior mientras se verifica a fondo que funcionen juntos de manera adecuada, un incremento proporciona valor al ser utilizable

[24]. Dentro de un Sprint puede crearse múltiples incrementos, y su suma total es presentada en el Sprint Review, sin embargo, un incremento puede presentarse a las partes interesadas antes de la finalización del Sprint. Finalmente, un trabajo no puede ser considerado como parte de un incremento a menos de que este cumpla con la definición de realizado (DoD, por sus siglas en inglés) [24].

2. METODOLOGÍA

Según [1]: la metodología “representa el cómo del Proyecto Integrador. En esta sección defina la forma en la que se desarrolló el proyecto. Inicialmente se define el tipo de investigación en la cual se inscribe este proyecto (básico, aplicado, exploratorio, descriptivo, explicativo). Posteriormente se especifican las técnicas e instrumentos a emplear para recolectar información (observación, experimentación, consulta, encuesta, entrevista, foro, entre otros), con los que se recogerá, organizará, sistematizará, interpretará y analizará la información. Finalmente se definen las diferentes etapas o fases metodológicas que permitirán estructurar adecuadamente el desarrollo del Proyecto Integrador”.

2.1. Metodología SCRUM

Para el desarrollo del trabajo se utilizó la metodología SCRUM que como se ha descrito en el marco teórico permite asegurar la calidad del producto, en primer lugar, permite plantear la realización de los requerimientos tomando en consideración cuanto esfuerzo tomará al equipo implementarlos, por lo que se puede estimar un tiempo para obtener entregables, así como el producto final, luego la lógica de incrementos que plantea esta metodología permite que las nuevas funcionalidades que se van añadiendo a lo largo de los sprint se integren a las anteriores obligatoriamente antes de avanzar a un nuevo sprint. Finalmente, la flexibilidad de SCRUM permite añadir nuevos requerimientos al backlog mientras se realiza el desarrollo de la aplicación, permitiendo que las funcionalidades más críticas del sistema se implementen aún si su planteamiento no es desde el principio. Estas características permiten entregar al usuario un software que cumpla con sus expectativas en cuanto a calidad y tiempo por lo cual implica una ventaja para la implementación de la aplicación en este trabajo.

2.2. Roles SCRUM

Este proyecto está integrado por tres miembros y cada uno tiene un rol diferente como se muestra en la Tabla 2.1:

Tabla 2.1 Roles del Equipo Scrum

Rol	Responsable
Product Owner	PhD. Silvana Del Pilar Gamboa Benítez
Scrum Master	MSc. Vicente Adrián Eguez Sarzosa
Development Team	Luis Andres Andrade Cabrera

2.3. Requerimientos iniciales

El proceso de elicitación de requerimientos fue realizado a través de entrevistas a la dueña del producto PhD. Silvana Gamboa. Para resumir los requerimientos iniciales, se ha generado la siguiente tabla que generaliza los módulos que la aplicación tendrá.

Tabla 2.2 Tabla de resumen de requerimientos iniciales

MÓDULO	REQUERIMIENTOS
Lienzo	Área donde se pueda agregar las representaciones gráficas.
	Dentro del lienzo se debe poder arrastrar, copiar, cortar y pegar las representaciones gráficas.
Tags	Debe permitir leer los valores de los tags desde la base de datos.
	Se permitirá también que el usuario defina sus propios tags locales que no dependerán de la base de datos.
Usuarios	La aplicación soportará que el usuario inicie sesión en los roles de administrador, diseñador, u operador.
	La aplicación podrá cambiar de modo de edición a modo de ejecución y para hacerlo de modo de ejecución a modo de edición será solicitando que el usuario inicie sesión como diseñador o administrador.
Exportación	La aplicación permitirá guardar los proyectos en un archivo de texto.

	También podrá guardar los proyectos en un archivo de texto cifrado.
Alarmas	La aplicación podrá definir condiciones a partir de los tags, y cuando estas se cumplan se notificarán al usuario como activadas, y cuando dejen de cumplir volverán a un estado de desactivadas.
Representaciones gráficas	<p>La aplicación contará con representaciones gráficas como:</p> <ul style="list-style-type: none"> • Figuras geométricas de línea, elipse y rectángulo. • Textos como Fecha y Hora del sistema, texto dinámico en base a un tag. • Imágenes que el usuario puede agregar desde el sistema de archivos y también símbolos HMI por defecto que el usuario puede elegir. • Un resumen de alarmas donde se pueda observar y administrar las distintas alarmas del proyecto. <ul style="list-style-type: none"> • Gráfico de tendencias representando distintos tags asociados.
	Ciertas representaciones gráficas permitirán que el usuario interactúe con ellas incluso en el modo de ejecución luego de dar doble clic sobre ellas y solicitar que el usuario inicie sesión.
Páginas o Ventanas	La aplicación podrá organizar las representaciones gráficas en varias páginas tantas como el usuario requiera.

	Estas páginas podrán exportarse en un archivo de texto y también en un archivo cifrado.
--	---

La aplicación se divide en varios módulos que se integrarán entre sí para tener su funcionalidad como las de software HMI profesionales.

2.4. DESCRIPCIÓN DEL SISTEMA

2.4.1 Modelo de la base de datos

La aplicación depende del esquema de base de datos del controlador Ethernet/IP™ desarrollada en [25], donde se definen las siguientes tablas:

- PLCS: Almacena la información requerida para conectarse a un PLC [25].
- INTERMEDIA: Es una tabla intermedia generada para relacionar la tabla PLCS con la tabla TAGS [25].
- TAGS: En esta tabla se registrará la información requerida para la conexión a un Tag de un PLC [25].
- ENTERO: Almacena el valor de un tag cuyo tipo de dato es entero [25].
- BOOL: Almacena el valor de un tag cuyo tipo de dato es booleano [25].
- FLOTANTE: Almacena el valor de un tag cuyo tipo de dato es flotante [25].

Se presenta a continuación en la Figura 2.1 el diagrama entidad relación de la base de datos del controlador.

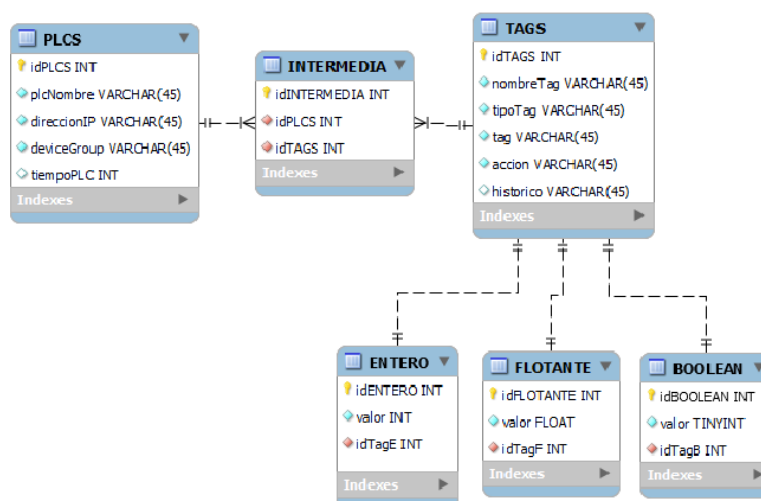


Figura 2.1 Diagrama entidad relación de la base de datos del controlador Ethernet/IP™ [25]

Además de la base de datos del controlador, se ha implementado un esquema de base de datos que almacena los datos del usuario y consta de una única tabla que se muestra en la Figura 2.2.

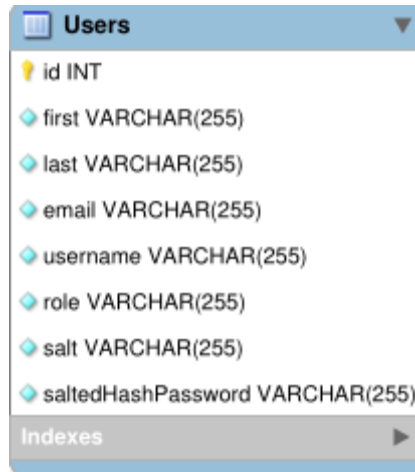


Figura 2.2 Diagrama entidad relación de la base de datos que almacena los datos del usuario

2.4.2 Arquitectura del sistema

Tomando en consideración el marco teórico en los puntos 1.3.4, 1.3.5, 1.3.6 se ha definido el estándar al cual se registrará la lógica de los gráficos y componentes HMI, es decir, la disponibilidad de opciones en cuanto a colores, o la forma de presentación de las alarmas busca asemejarse a dichos estándares. Para presentar la arquitectura del sistema se optó por separar al sistema en una arquitectura de tres niveles [26] donde cada nivel se describe como:

1. Nivel de presentación: Es la interfaz de usuario, es la forma en la cual el sistema permite la interacción y presenta la información al usuario final [26]. En el presente proyecto se optó por utilizar la librería JavaFX para brindar al usuario una interfaz moderna.
2. Nivel de aplicación: Este es el núcleo de la aplicación, aquí se procesa la información y se utiliza la lógica del negocio, es decir, el conjunto de reglas que el usuario espera que el sistema cumpla [26]. En este proyecto la lógica corresponde a los distintos objetos que tienen un comportamiento específico, como por ejemplo un resumen de alarmas o un rectángulo dinámico que muestra a manera de una barra de progreso el estado y valor actual de un Tag de la base de datos.
3. Nivel de datos: En este nivel se gestiona y almacena la información que la aplicación procesa, generalmente se utiliza un sistema de gestión de base de datos relacional

[26]. Se implementó esta capa utilizando MySQL, en este sistema de base de datos se consulta e interactúa con dos esquemas uno por parte del controlador EIP y otro esquema que gestiona los usuarios y sus contraseñas.

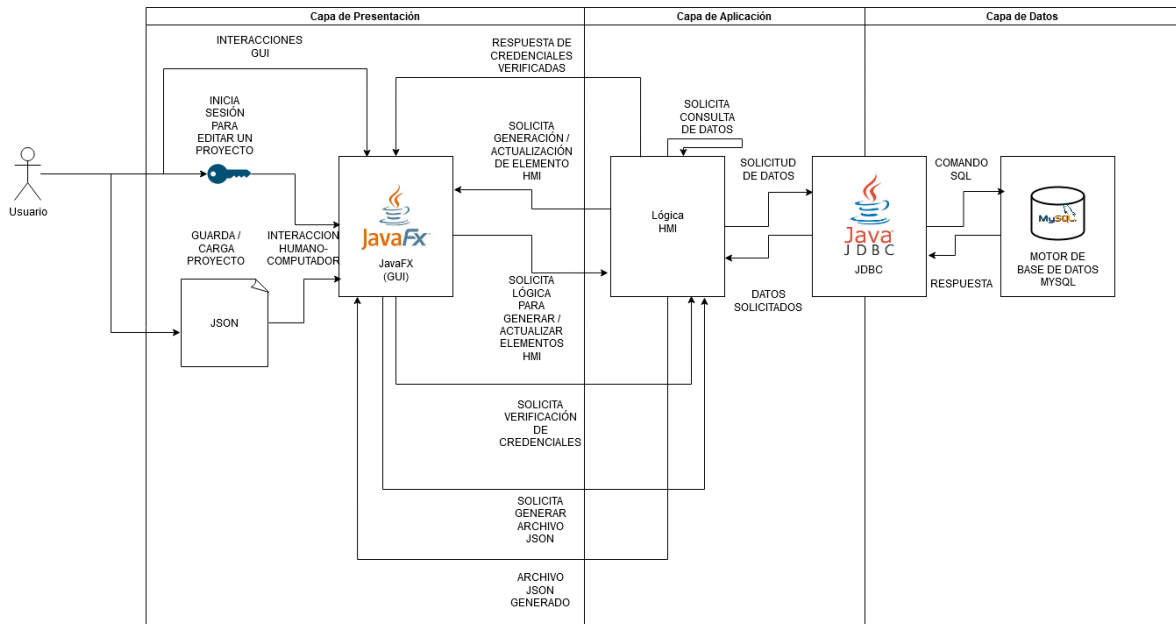


Figura 2.3 Arquitectura del sistema

La arquitectura del sistema se encuentra graficada en la Figura 2.3, donde se detallan los tres niveles, así como las interacciones más importantes entre los componentes.

2.4.3 Patrón de diseño

Un patrón de diseño es un conjunto de terminologías estándares que permiten al desarrollador enfrentar de manera más eficiente un escenario particular, porque ha sido evaluado por muchos otros desarrolladores a lo largo del tiempo, y se vuelven un procedimiento que ha evolucionado proponiendo una solución mejor para ciertos problemas [27].

El diagrama de la Figura 2.3, aunque permite ejemplificar cómo se comporta el sistema no es el patrón de diseño, el patrón seleccionado es el patrón de diseño Modelo-Vista-Controlador (MVC). Esta forma de diseño divide la aplicación en tres partes, similares a la arquitectura de tres niveles que se detallan a continuación [28]:

1. **Modelo:** Es el núcleo del funcionamiento de la aplicación, ya que contiene únicamente su información, es decir, no conoce como mostrar los datos al usuario. Además, se encarga de controlar la lógica y las reglas de la aplicación [28].

2. Vista: Es la parte que se muestra al usuario, ya que permite visualizar los datos del modelo, es decir, accede a los datos del modelo y los representa en distintos elementos gráficos como tablas, diagramas, así como gráficos de barras [28].
3. Controlador: Es la parte que más trabaja en este patrón, se encarga de administrar las entradas desde el usuario y procesar dichas entradas a modo de instrucciones de la aplicación. Se ejecuta como intermediario entre el modelo y la vista, actualizando los datos del modelo y este al estar interconectado con la vista refleja su cambio en ella [28].

En la Figura 2.4, se ilustra el patrón MVC del actual proyecto, donde las vistas que irán desde las ventanas hasta las representaciones gráficas interactivas se graficarán en JavaFX, luego existe el modelo que serán un conjunto de clases con los datos para recrear un gráfico cuya información viene de dos fuentes en primer lugar un archivo JSON que indicará que figura se graficará y luego la base de datos que tiene los valores actuales que se representarán. Obviamente para la interacción entre la vista y el modelo se contará con diversos controladores que permitirán el funcionamiento adecuado de la aplicación.

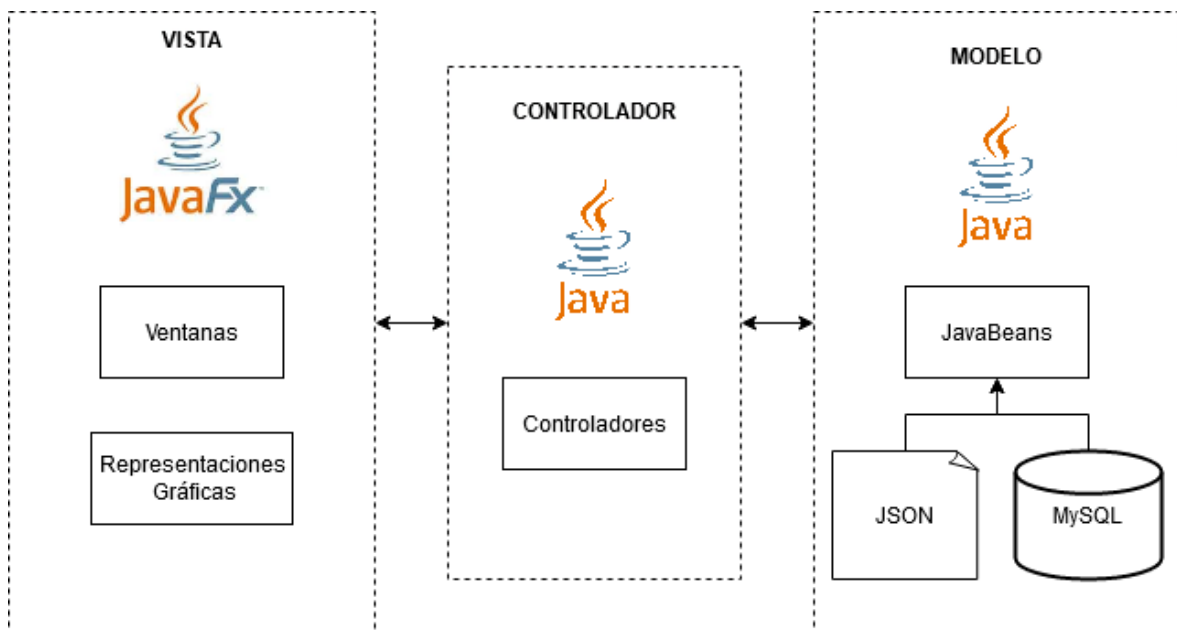


Figura 2.4 Patrón del diseño del sistema

2.5. Historias Épicas

Entonces para comenzar el proyecto se ha definido en una historia épica, esta describirá el propósito principal del proyecto como se puede ver en la Tabla 2.3:

Tabla 2.3 Historias Épicas

ID	Título	Prioridad
HE1	Desarrollo de una aplicación que muestre en forma gráfica el estado de las variables que fueron adquiridas desde un proceso industrial.	Alta

2.6. Product Backlog

A continuación, se presentan las historias de usuario que derivaron de la entrevista a la dueña del producto, asimismo se ha definido una estimación del esfuerzo requerido para llevarla a cabo. También se describirá la prioridad, tomando en cuenta el aporte en valor de cada historia.

Nomenclaturas de la Tabla:

- **US:** Historia de usuario
- **##:** Numero de historia, por ejemplo 01.

A continuación, se muestra las historias de usuario:

Tabla 2.4 Product Backlog

Código	Descripción	Prioridad (MoSCoW)	Estimación (Fibonacci)	Criterios de aceptación
US-01	Como diseñador, quiero agregar representaciones gráficas de objetos del área industrial, para poder visualizar un área de trabajo simulada similar a la que se tiene en el entorno real.	Mo	8	<ul style="list-style-type: none"> • Poder agregar al lienzo distintas representaciones gráficas. • Contar con las representaciones gráficas de tanques, válvulas, etc.

US-02	Como diseñador, quiero agregar nuevas representaciones gráficas de objetos del área industrial sin afectar la existencia de otros objetos, para poder tener nuevos objetos sin la necesidad de esperar una nueva versión del software.	S	13	<ul style="list-style-type: none"> Las representaciones gráficas, deben poder "importarse", de modo que un usuario agregue una imagen de origen externo y se pueda acceder en las próximas ejecuciones.
US-03	Como diseñador, quiero asociar variables del tipo entero, flotante o bool a las representaciones gráficas para poder monitorizar sus estados en tiempo real.	Mo	5	<ul style="list-style-type: none"> Cada representación gráfica aceptará diferentes tipos de datos, por ejemplo, un tanque puede aceptar tipo entero o flotante, mientras que un motor o una luz solo serán del tipo booleano.
US-04	Como Diseñador, Quiero Definir los tiempos de actualización	Co	5	<ul style="list-style-type: none"> Las representaciones gráficas actualizan constantemente su estado leyéndolo desde la base de datos, sin embargo, estos

	del estado de las variables, Para poder diferenciar entre las representaciones que requieran más prioridad en su monitoreo			tiempos de actualización deben poder ser definidos por el diseñador.
US-05	Como Diseñador, Quiero Agregar gráficos de tendencias asociadas a una de las variables que pueden ser representadas con una ventana de X número de segundos hacia atrás, Para poder monitorizar como se ha desarrollado el comportamiento de dicha variable	Co	13	<ul style="list-style-type: none"> Mostrar los gráficos de tendencia para una variable representada, pudiendo definir con cuanto tiempo hacia el pasado se mostrará dicha gráfica. Esta gráfica también será mostrada dentro del lienzo como otra representación gráfica.
US-06	Como diseñador, quiero agregar textos que	S	5	<ul style="list-style-type: none"> Un valor que venga desde la base de datos del controlador además de ser asociado a una

	reflejen el valor actual de una variable asociable a una representación gráfica, para monitorizar el valor actual textual de sus estados en tiempo real.			representación gráfica debe también poder ser asociado a un texto que varíe acorde a su valor desde la base de datos.
US-07	Como diseñador, quiero agregar una animación a una representación gráfica indicando el cambio de su estado, para poder monitorizar su estado sin necesidad de leer su valor textual.	Co	8	<ul style="list-style-type: none"> • Una representación gráfica en forma de rectángulo que podrá ser asociada a variables del tipo entero, flotante y Bolean de modo que se pueda mostrar en forma de una barra de progreso que vaya llenándose, además de poder definirse el relleno de forma horizontal o vertical. • En el caso de los booleanos Falso será 0 y Verdadero será 100.
US-08	Como administrador, quiero crear cuentas de usuario identificadas por usuario y contraseña que pertenezcan a	Co	5	<ul style="list-style-type: none"> • Poder crear distintas cuentas de usuarios, que pertenezcan a distintos roles. Los roles se diferencian en: • Un Administrador, puede crear los distintos roles y

	<p>los Roles "Diseñador", "Operador", "Administrador", para poder brindar distintos tipos de privilegios según el tipo de usuario.</p>			<p>todo lo que puede hacer un Diseñador y Operador.</p> <ul style="list-style-type: none"> • Un Diseñador, puede editar el HMI creado, y todo lo que puede hacer un Operador, es decir, puede acceder a los modos de Ejecución y Edición. • Un Operador, puede visualizar los objetos, así como administrar alarmas, es decir, solo puede acceder al modo de ejecución.
US-09	<p>Como diseñador, quiero definir alarmas, es decir, condiciones en las variables para que al momento de cumplir o no dichas condiciones se notifique al operador cuando se está ejecutando el HMI.</p>	Mo	8	<ul style="list-style-type: none"> • Los tipos de condiciones admitidos serán, "mayor que", "mayor o igual que", "menor que", "menor o igual que", "Igual a". • Cuando una condición es definida y esta no se cumple, se debe registrar este evento como una alarma. La notificación de una alarma debe realizarse de modo que se refleje en el resumen de alarmas.
US-10	<p>Como diseñador, quiero organizar mis</p>	S	5	<ul style="list-style-type: none"> • El software debe permitir diseñar en varios lienzos y a su vez relacionarlos entre sí, es decir, debe

	<p>representacion es gráficas en distintas ventanas y poder moverme entre ellas desde la interfaz diseñada, para poder reducir la cantidad de objetos por pantalla.</p>			<p>existir un objeto dentro del lienzo para que cuando el HMI este en ejecución al dar clic en dicho objeto se pase al siguiente lienzo y a su vez poder regresar al mismo.</p>
US-11	<p>Como diseñador, quiero agregar la variable del sistema de fecha y hora como una representación gráfica en forma textual, para poder observar su valor en el lienzo.</p>	Co	3	<ul style="list-style-type: none"> • Poder agregar las variables del sistema de fecha y hora como una representación dinámica en forma de texto dentro del lienzo de diseño y que esta funcione también dentro de la ejecución.
US-12	<p>Como operador, quiero poder acceder a un "Resumen de Alarmas", para poder interactuar con un resumen de</p>	S	8	<ul style="list-style-type: none"> • El software debe proveer un área de Resumen de alarmas, que estará relacionado con las condiciones definidas dentro de las alarmas creadas.

	las alarmas actuales y pasadas, donde pueda reconocer, suprimir, deshabilitar, archivar o quitar las alarmas.			<ul style="list-style-type: none"> Desde el Resumen de alarmas se puede "Reconocer" Una Alarma, esto es ya que cuando se ejecuta una alarma se creará en un estado de "no reconocida", este estado también deberá mostrarse en forma gráfica en el objeto, cuando el operador reconoce la alarma indica.
US-13	Como diseñador, quiero tener un lienzo, para poder agregar las distintas representaciones gráficas, así como copiarlas, cortarlas y pegarlas dentro del mismo.	Mo	8	<ul style="list-style-type: none"> Área donde se puedan agregar representaciones gráficas. Dentro del lienzo en el modo de edición se podrá mover las representaciones gráficas a través de dar clic y arrastrarla hacia una nueva posición. Dentro del lienzo en el modo de edición se podrá copiar, cortar y pegar las representaciones gráficas dentro del mismo.
US-14	Como diseñador, quiero guardar el proyecto en un archivo, para poder continuar con el diseño o que un	Mo	8	<ul style="list-style-type: none"> Poder obtener un archivo del diseño actual del HMI, con todas las representaciones gráficas, sus nombres de variable, sus tipos de datos, sus variables asociadas, sus condiciones definidas, sus posiciones, etc.

	operador lo pueda ejecutar.			<ul style="list-style-type: none"> El software debe poder reconocer el archivo y recuperar el estado de los objetos en su interior, y representarlo en el lienzo y recrear las distintas ventanas que pueden existir.
US-15	Como diseñador, quiero tener una ventana de inicio donde pueda acceder a los proyectos previamente creados y acceder a ellos, para facilitar el acceso a los proyectos previamente iniciados, así como iniciar proyectos nuevos.	S	3	<ul style="list-style-type: none"> Al iniciar el Software de debe mostrar esta pantalla mostrando los anteriores proyectos realizados y dando la opción de crear uno nuevo.
US-16	Como Diseñador, Quiero Definir contraseñas maestras en los HMI desarrollados, Para Poder Controlar quien puede ejecutar	S	5	<ul style="list-style-type: none"> Poder definir una contraseña maestra para la edición y ejecución del archivo de proyecto HMI, más allá que la correspondiente de los usuarios y sus roles.

	y editar los diseños.			
US-17	Como diseñador, quiero definir la precisión de los datos del tipo flotante, para poder reducir o aumentar la visibilidad de los decimales.	Co	3	<ul style="list-style-type: none"> Un dato del tipo flotante que viene desde la base de datos puede tener muchos decimales que pueden superar el admitido dentro del lienzo, el software debe permitir definir el tamaño máximo de cuantos decimales serán mostrados.
US-18	Como usuario, Quiero agregar un tag que se ejecuta localmente sin necesidad de la base de datos	Co	5	<ul style="list-style-type: none"> Poder crear un tag que no dependa de un valor de la base de datos, y pueda ser seleccionado para ser mostrado/asociado/modificado como si se tratase de un tag normal, Debe guardarse dentro de los archivos de proyecto y cargarse junto con él.
US-19	Como usuario, quiero un menú con las distintas representaciones gráficas en forma de íconos para poder agregarlas al lienzo.	Mo	1	<ul style="list-style-type: none"> Tener un área de distintos íconos en forma de cuadrícula con los distintos objetos que se pueden agregar al lienzo.
US-20	Como usuario quiero un menú horizontal superior con las	Mo	1	<ul style="list-style-type: none"> Quiero una barra de menús que contenga los distintos menús

	opciones de archivo, editar, alarmas, configurar y ventanas para permitir al usuario acceder a ellas.			<p>disponibles y se mostrará en la parte superior.</p> <ul style="list-style-type: none"> • Tendrá las opciones de Archivo, Editar, Alarmas, Configurar, Ventanas, Ayuda
US-21	Como usuario quiero un menú para poder acceder a las distintas pántinas creadas para poder a través de las distintas páginas creadas.	Mo	1	<ul style="list-style-type: none"> • Un menú que permita intercambiar entre las distintas ventanas en cualquier modo.
US-22	Como usuario quiero un modo de edición donde se puede modificar el archivo de proyecto y un modo de ejecución donde solo se podrá visualizar los objetos del proyecto, para poder utilizar la aplicación en estos distintos modos.	Mo	3	<ul style="list-style-type: none"> • La aplicación podrá cambiar de los modos: <ul style="list-style-type: none"> ○ Ejecución: Donde no se puede alterar las propiedades de los objetos estáticos ○ Edición: Donde se puede modificar todo lo que existe dentro del proyecto si se tienen los privilegios correspondientes. ○ Al cambiar del modo de ejecución al modo de Edición

				se solicitarán las credenciales correspondientes.
US-23	Como usuario quiero poder exportar e importar solamente las ventanas hacia otro proyecto	Mo	5	<ul style="list-style-type: none"> • Generar un archivo que contenga los datos de una sola ventana, este archivo podrá ser en texto plano o encriptado, Además este mismo archivo se podrá cargar en otros proyectos.
US-24	Como usuario quiero un área donde se muestren los distintos menús del lienzo que se posicionará en el lado izquierdo de la aplicación para se podrá mostrarla u ocultarla y así no perder espacio en la pantalla para agregar representaciones gráficas.	Mo	1	<ul style="list-style-type: none"> • Un área que contenga los menús de Agregar objetos al lienzo y la lista de ventanas, este menú solo se mostrará en el modo de edición.
US-25	Como usuario quiero poder agregar una figura geométrica como una línea	S	3	<ul style="list-style-type: none"> • Poder agregar las figuras geométricas de Línea y Círculo con las mismas propiedades de cualquier otra representación gráfica.

	y un círculo para poder utilizarlas en el proyecto.			
--	---	--	--	--

2.7. Desarrollo

2.7.1 Sprint 0

Release Planning

Tomando en consideración las distintas historias de usuario definidas en el Product Backlog y con una velocidad estimada de desarrollo por Sprint de 20 puntos, se determina que se requieren al menos 6 Sprints para terminar el producto final, definiéndose el Release Planning correspondiente al proyecto. A continuación, se muestran las historias a desarrollarse en cada uno de ellos:

Tabla 2.5 Release Planning

Release Planning						
Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7
US-06	US-03	US-01	US-02	US-12	US-05	US-04
US-11	US-07	US-08	US-09	US-15		US-16
US-13		US-10	US-14	US-19		US-18
			US-17	US-20		US-23
				US-21		
				US-22		
				US-24		
				US-25		
16	13	18	19	21	13	20

2.7.2 Sprint 1

2.7.2.1 Sprint Planning

Como objetivo principal del primer Sprint se buscó tener las características básicas que permitan el desarrollo de las próximas historias de usuario, por lo que se implementará el lienzo donde todas las representaciones gráficas se agregarían en un futuro, además de implementar las primeras que serían solamente textos pero que deben asociarse a un tag de la base de datos y deben variar con el tiempo.

2.7.2.2 Sprint Backlog

Para el primer Sprint se seleccionaron las historias de usuario que se muestran a continuación en la Tabla 2.6 de historias de usuario:

Tabla 2.6 Backlog Sprint 1

ID	Descripción	Tareas
US-06	Como diseñador, quiero agregar textos que reflejen el valor actual de una variable asociable a una representación gráfica, para monitorizar el valor actual textual de sus estados en tiempo real.	<ul style="list-style-type: none">• Asociar los valores de un tag leídos desde la base de datos se reflejen en el "label".• Implementar un Hilo que lea cada X segundos el valor de la base de datos y lo actualice en caso de cambio.
US-11	Como diseñador, quiero agregar la variable del sistema de fecha y hora como una representación gráfica en forma textual, para poder observar su valor en el lienzo.	<ul style="list-style-type: none">• Asociar los valores de la fecha y hora actual del sistema y se reflejen en el "label".• Implementar forma de que el Formato de fecha y hora se pueda leer desde una variable.• Implementar un Hilo que sea cada 1 segundos el valor de fecha y hora.
US-13	Como diseñador, quiero tener un lienzo, para poder agregar las distintas representaciones gráficas, así como copiarlas, cortarlas y pegarlas dentro del mismo.	<ul style="list-style-type: none">• Implementar el lienzo donde se puedan agregar las "Representaciones Gráficas".• Implementar el clic y arrastrar para poder cambiar la posición de un objeto dentro del lienzo.• Implementar la acción de copiar una representación gráfica hacia el portapapeles.• Implementar la acción de cortar una representación gráfica hacia el portapapeles.• Implementar la acción de pegar una representación gráfica desde el portapapeles.

2.7.2.3 Ejecución del Sprint

Creación del lienzo y primeros objetos que se pueden agregar

La aplicación requiere un área donde se puedan agregar las distintas representaciones gráficas por lo que pare diferenciar un objeto seleccionado de uno que no se agregó un borde al objeto, como se ve a continuación en la Figura 2.5.



Figura 2.5 Pareja de "labels" agregados al lienzo, con el primero no seleccionado y el segundo seleccionado.

Implementación de las opciones de copiar y cortar hacia el portapapeles

Se implementó la opción para poder copiar o cortar hacia el portapapeles una representación gráfica, para hacerlo se debe seleccionar la figura y luego dar clic derecho para que se muestre el menú Contextual donde se podrán elegir las opciones, tal y como se muestra en la Figura 2.6, además se habilitó los atajos de teclado usuales para copiar como es "Ctrl + C" y para cortar "Ctrl + X".

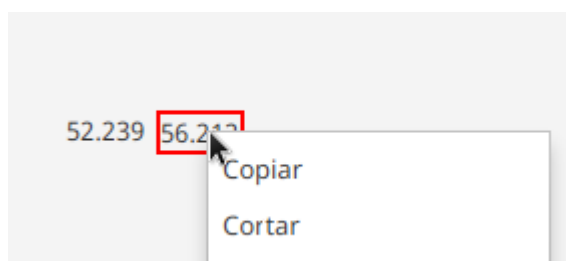


Figura 2.6 menú contextual sobre una representación gráfica seleccionada, con las opciones de copiar o cortar.

Implementación de la opción de pegar desde el portapapeles

Se implementó la opción para poder pegar una representación gráfica desde el portapapeles al dar clic en alguna parte libre dentro del lienzo, donde se mostrará un menú contextual con la opción de pegar y se agregará una representación gráfica con las mismas propiedades que la original, esto se puede observar en la Figura 2.7.



Figura 2.7 Menú contextual para pegar una "representación gráfica"

Implementación de texto asociado a la fecha y hora del sistema

Se implementó un "label" que se encuentra asociado fecha y hora actual del sistema que además se actualizará cada segundo, como se puede ver en la Figura 2.8, además será

posible cambiar el formato de fecha y hora, ya que se encuentra asociado a un atributo de dicha representación gráfica.



Figura 2.8 Representación gráfica en forma de texto que se encuentra asociado a la fecha y hora actuales del sistema

2.7.2.4 Sprint Review

Tabla 2.7 Criterios de aceptación del Sprint 1

ID	Descripción	Criterio de Aceptación	Cumple
US-06	Como diseñador, quiero agregar textos que reflejen el valor actual de una variable asociable a una representación gráfica, para monitorizar el valor actual textual de sus estados en tiempo real.	La aplicación puede leer los valores desde la base de datos y representarlos en tiempo real.	SI
US-11	Como diseñador, quiero agregar la variable del sistema de fecha y hora como una representación gráfica en forma textual, para poder observar su valor en el lienzo.	La aplicación puede leer la fecha y hora actuales del sistema y cambiar en tiempo real.	SI
US-13	Como diseñador, quiero tener un lienzo, para poder agregar las distintas representaciones gráficas, así como copiarlas, cortarlas y pegarlas dentro del mismo.	La aplicación cuenta con un lienzo donde se pueden agregar las distintas representaciones gráficas, moverlas, y realizar las acciones de copiar, cortar y pegar las figuras.	SI

2.7.2.5 Sprint Retrospective

¿Qué salió bien en la iteración?

Durante el desarrollo se utilizó la extensión de IDE SonarLint que permite ayudar en la mejora del código siguiendo una serie de métricas ayudando a evitar errores de código que afectan la calidad final del proyecto.

¿Qué se puede mejorar?

La aplicación actualmente solo puede leer los datos desde la base de datos una vez por segundo, sin embargo, es necesario que se pueda personalizar a que intervalo de tiempo es requerida la lectura de los datos. Adicionalmente, surge la necesidad de implementar tags locales que no requieran de la base de datos.

2.7.3 Sprint 2

2.7.3.1 Sprint Planning

El objetivo de este Sprint busca definir los distintos tipos de datos de tag representables desde la base de datos del controlador Ethernet/IP, es decir, entero, flotante o bool. Además de crear el objeto dinámico a modo de barra de progreso que refleje los valores de los tags.

2.7.3.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.8:

Tabla 2.8 Backlog Sprint 2

ID	Descripción	Tareas
US-03	Como diseñador, quiero asociar variables del tipo entero, flotante o bool a las representaciones gráficas para poder monitorizar sus estados en tiempo real.	<ul style="list-style-type: none">• Crear los objetos que permitan la lectura y operaciones con los valores desde la base de datos.• Enlazar los tags para su lectura y escritura a través de un hilo que se ejecute cada X segundos.• Validar los objetos por su tipo de datos para así diferenciar entre tags Enteros, Flotantes o Booleanos.
US-07	Como diseñador, quiero agregar una animación a una representación gráfica indicando el cambio de su estado, para poder monitorizar su estado sin necesidad de leer su valor textual.	<ul style="list-style-type: none">• Implementar la animación de relleno dentro de un rectángulo en los modos:<ul style="list-style-type: none">○ Horizontal de Izquierda a Derecha.○ Horizontal de Derecha a Izquierda.○ Vertical de Arriba hacia Abajo.○ Vertical de Abajo hacia Arriba.• Implementar la selección de colores para representar el relleno y sin relleno.• Implementar los valores máximos y mínimos para representar el 100% de relleno y el 0% de relleno.• Agregar una opción que muestre el valor actual en porcentaje del tag asociado.

2.7.3.3 Ejecución del Sprint

Creación de ventanas que permitan asociar los diferentes tipos de tags de tipo entero, flotante y bool

La aplicación asociará los tags de distintos tipos, desde la base de datos a distintas opciones y elementos del lienzo, por lo que es necesario una ventana que permita seleccionarlos, dicha ventana puede observarse en la Figura 2.9.

Nombre del PLC	IP	Grupo	Nombre del Tag	Tipo	Dirección	Acción
plc1	192.168.1.11	tostado	sp_temp_tostado	Flotante	F8:3	Escritura
plc1	192.168.1.11	tostado	temp_tostado	Flotante	F8:0	Lectura
plc1	192.168.1.11	tostado	vel_horno	Flotante	F8:1	Lectura
plc1	192.168.1.11	tostado	status_horno	Bool	B3:0/0	Escritura
plc1	192.168.1.11	tostado	status_compuerta1	Bool	B3:0/1	Lectura
plc1	192.168.1.11	tostado	status_molino1	Bool	B3:0/2	Escritura
plc1	192.168.1.11	tostado	status_blower	Bool	B3:0/3	Lectura
plc1	192.168.1.11	tostado	status_molino2	Bool	B3:0/4	Lectura
plc1	192.168.1.11	tostado	vel_molinos	Flotante	F8:2	Lectura
plc1	192.168.1.11	tostado	nivel	Entero	N7:0	Lectura
plc2	192.168.1.12	mezclado	status_mezclador	Bool	B3:0/0	Escritura
plc2	192.168.1.12	mezclado	sp_vel_mezclador	Entero	N7:0	Escritura
plc2	192.168.1.12	mezclado	vel_mezclador	Flotante	F8:0	Lectura
plc2	192.168.1.12	mezclado	nivel_leche	Entero	N7:1	Lectura

Figura 2.9 Ventana de selección de tag a ser asociado a una operación

Implementación de una ventana de selección de propiedades para la figura dinámica

Para poder agregar una figura dinámica en el lienzo será necesario en primer lugar definir una serie de propiedades y atributos para su funcionamiento por lo que estos serán definidos en la ventana que puede observarse en la Figura 2.10, en primera instancia se tiene un cuadro de texto, donde se podrá asociar los tags de la base de datos al dar clic en el botón inferior con el texto “Añadir tag” que mostrará la ventana de la Figura 2.9, luego se tienen dos seleccionadores de color para el color de relleno o color primario y para el color sin relleno o color secundario, después se tiene los distintos tipos de orientaciones disponibles para el usuario que se detallarán más adelante, a continuación se tiene la definición de los límites de valor mínimo y valor máximo para realizar los cálculos en cuanto al porcentaje de relleno, la siguiente sección tiene la definición de la precisión de los decimales que se implementará en futuros Sprint. Por último, se tiene el área de botones

con el botón “Limpiar tags” que elimina las variables previamente añadidas, luego se tiene el botón de “Añadir tag” que ya fue descrito previamente, y finalmente el botón de “OK” para terminar la definición de los parámetros.

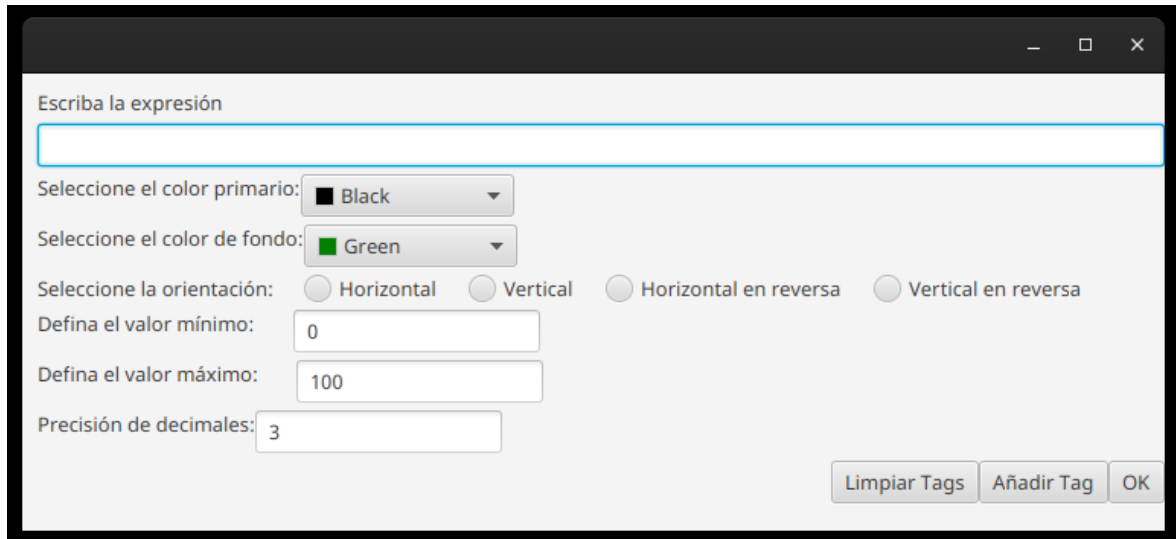


Figura 2.10 Ventana de definición de propiedades para el rectángulo dinámico

Implementación del rectángulo dinámico con la animación de relleno horizontal de izquierda a derecha

Para representar de forma gráfica y en tiempo real como cambian los valores de un tag de la base de datos, se implementó un rectángulo simulando el relleno de figuras a partir del porcentaje equivalente dentro de límites de valor mínimo y valor máximo definidos por el usuario, se puede observar el resultado en la Figura 2.11, donde se ha definido un tag flotante con valor 56.212, que se ve reflejado por los dos colores el negro como el relleno y el verde como sin relleno, donde el color de relleno es claramente mayor al 50%, ya que se han definido los límites de valor mínimo y valor máximo en 0 y 100 respectivamente.

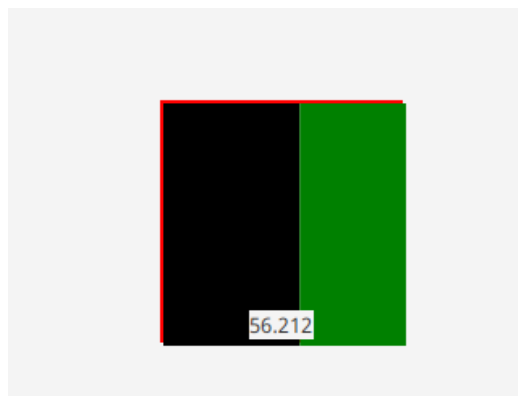


Figura 2.11 Figura dinámica representando un tag con valor 56.212

Además, podemos ver cómo cambia la figura dinámica si se cambia el límite inferior de 0 a 50, como se muestra en la Figura 2.12.

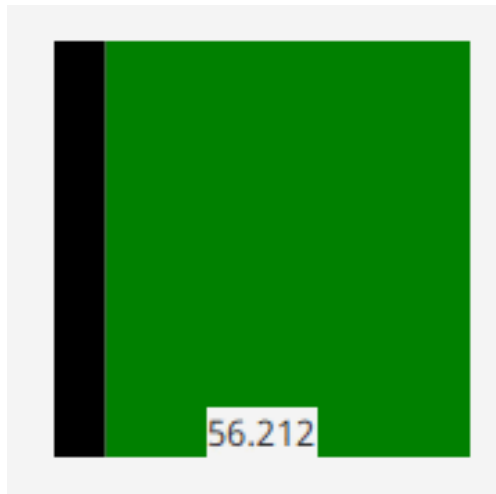


Figura 2.12 Figura dinámica con valor real de 56.212 y con el límite de valor mínimo en 50

Implementación del rectángulo dinámico con la animación de relleno horizontal de derecha a izquierda

Se implementó la opción para representar el relleno en forma horizontal en el sentido de derecha a izquierda, siguiendo los mismos parámetros anteriores solo que cambiando el modo de relleno por lo que el valor porcentual y el recorrido de la figura son diferentes, como se muestra en la Figura 2.13.

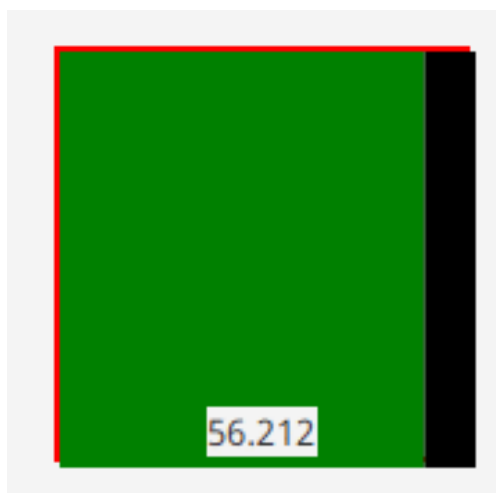


Figura 2.13 Figura dinámica con valor real de 56.212 y con el límite de valor mínimo en 50 pero con el sentido de derecha a izquierda

Implementación del rectángulo dinámico con la animación de relleno vertical de arriba hacia abajo

En esta implementación se puede observar en la Figura 2.14, que el sentido del relleno ha cambiado y el porcentaje de relleno por parte del color negro es mayor y viene de origen desde arriba.

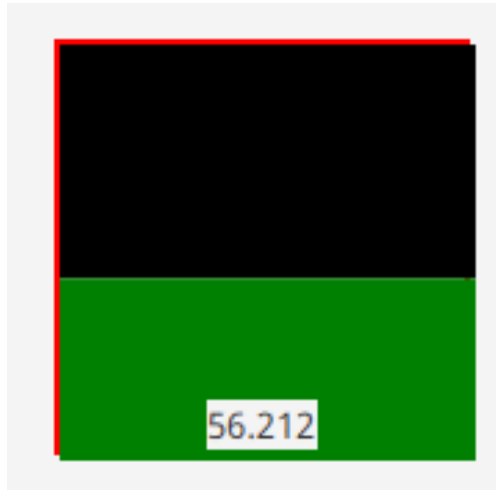


Figura 2.14 Figura dinámica con valor real de 56.212 representado en la figura de relleno vertical de arriba hacia abajo

Implementación del rectángulo dinámico con la animación de relleno vertical de abajo hacia arriba

A continuación, se muestra en la Figura 2.15 un comportamiento similar al anterior, pero con el relleno viene en el sentido desde abajo.

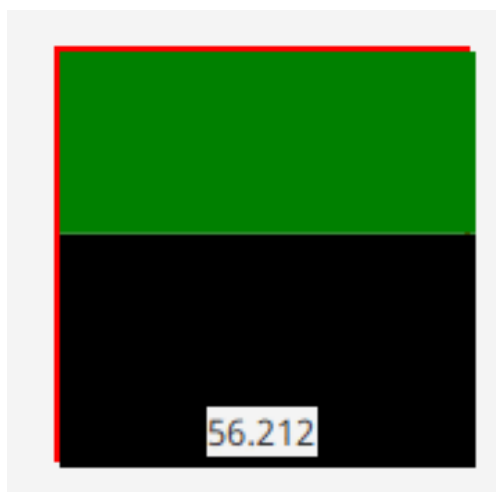


Figura 2.15 Figura dinámica con valor real de 56.212 representado en la figura de relleno vertical de abajo hacia arriba

Implementación del cambio de colores del color de relleno y del color sin relleno

En esta sección se muestra el cambio de colores de relleno y sin relleno de la Figura 2.15 donde se cambió el color de relleno desde Negro hacia un color Rojo oscuro y el de sin relleno de verde a azul, este comportamiento se ilustra en la Figura 2.16.

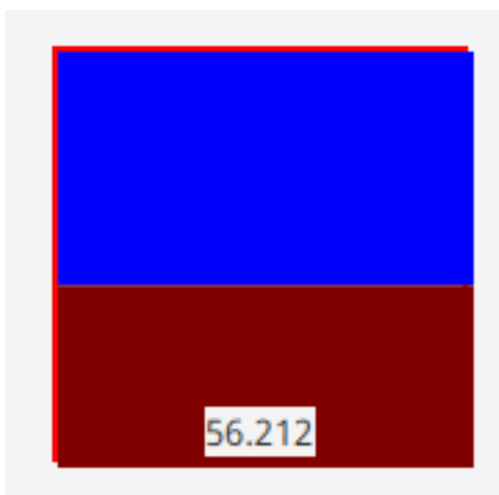


Figura 2.16 Figura dinámica con los colores por defecto modificados.

2.7.3.4 Sprint Review

Tabla 2.9 Criterios de aceptación del Sprint 2

ID	Descripción	Criterio de Aceptación	Cumple
US-03	Como diseñador, quiero asociar variables del tipo entero, flotante o bool a las representaciones gráficas para poder monitorizar sus estados en tiempo real.	La aplicación permite la selección de los distintos tipos de tags y asociarlos a un objeto.	SI
US-07	Como diseñador, quiero agregar una animación a una representación gráfica indicando el cambio de su estado, para poder monitorizar su estado sin necesidad de leer su valor textual.	La aplicación hace una lectura y representación dinámica de los tags desde la base de datos y los muestra a modo de relleno en distintas orientaciones disponibles al usuario.	SI

2.7.3.5 Sprint Retrospective

¿Qué salió bien en la iteración?

Se implementó correctamente una figura dinámica, que depende del valor actual de un tag para simular una animación de relleno dentro de un rectángulo, además de poder definir sus propiedades como sus dimensiones y sus colores.

¿Qué se puede mejorar?

Actualmente solo existe un tipo de figura dinámica con forma de un rectángulo, sin embargo, se podrían implementar para más tipos de figuras definidas por el usuario.

2.7.4 Sprint 3

2.7.4.1 Sprint Planning

El objetivo de este Sprint busca implementar una galería de representaciones gráficas de objetos del área industrial que permitan al usuario agregarlos y tener un fácil acceso a figuras que le serán útiles para simular el entorno real. También se busca implementar la creación de roles que diferenciaran entre los distintos tipos de usuarios y sus necesidades. Además, se implementará la creación de múltiples páginas dentro de un mismo proyecto permitiendo una mejor organización de las figuras dentro de un proyecto.

2.7.4.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.10.

Tabla 2.10 Backlog Sprint 3

ID	Descripción	Tareas
US-01	Como diseñador, quiero agregar representaciones gráficas de objetos del área industrial, para poder visualizar un área de trabajo simulada similar a la que se tiene en el entorno real.	<ul style="list-style-type: none">• Agregar una biblioteca de figuras de objetos del área industrial más utilizados disponibles para la selección del usuario.• Implementar una ventana de que permitan la selección de las distintas figuras, así como editar las propiedades de estas para agregarlas al lienzo.• Agregar opción para girar las figuras.• Agregar opción para reflejar las figuras.• Agregar opción para editar los colores de la figura.
US-08	Como administrador, quiero crear cuentas de usuario identificadas por usuario y contraseña que pertenezcan a los Roles "Diseñador", "Operador", "Administrador", para poder brindar distintos tipos de privilegios según el tipo de usuario.	<ul style="list-style-type: none">• Diseñar e implementar la base de datos de usuarios.• Agregar opción para generar base de datos desde la aplicación.• Agregar opción para guardar las credenciales de acceso a MySQL.• Implementar ventana para Registro y Edición de Usuarios.• Implementar ventana de Inicio de Sesión.
US-10	Como diseñador, quiero organizar mis representaciones gráficas en distintas ventanas y poder moverme entre ellas	<ul style="list-style-type: none">• Implementar la opción para crear nuevas páginas o ventanas dentro de un proyecto.• Implementar ventana para definir las propiedades de una página.

	desde la interfaz diseñada, para poder reducir la cantidad de objetos por pantalla.	<ul style="list-style-type: none"> ● Implementar la opción para moverse entre páginas. ● Implementar la opción para editar las propiedades de una página.
--	---	---

2.7.4.3 Ejecución del Sprint

Implementación de ventana de selección de figuras

La aplicación permitirá la selección de distintas figuras predeterminadas para ello es necesario implementar una ventana que permita su selección, dicha ventana puede observarse en la Figura 2.17, donde se presentan las distintas categorías disponibles al usuario, al dar clic en una de ellas se desplegarán las diferentes imágenes predeterminadas, este comportamiento se ilustra con la categoría seleccionada “1. Tuberías”, además una figura también ha sido seleccionada, luego en la parte inferior se muestran dos botones uno que llevará a las propiedades de imagen y otro que terminará el proceso de agregar la figura predeterminada al lienzo.



Figura 2.17 Ventana de selección de figuras predeterminadas

Implementar opción para editar las propiedades de la figura

La aplicación ofrece la opción para definir una serie propiedades de imagen, al dar clic en el botón “Opciones de Imagen” de la ventana detallada en la Figura 2.18, donde se podrán definir propiedades tales como el valor del ancho y del alto, así como habilitar la opción de preservar el radio al momento de cambiar el tamaño por defecto de la imagen, es decir, que si una imagen es cuadrada esta se mantenga así aunque los valores de ancho y alto

sean propios de un rectángulo. Otro par de opciones disponibles son las de reflejar horizontal o verticalmente, es decir, como se vería la imagen reflejada en un espejo posicionado horizontal o verticalmente. Además, existe la opción de definir un valor para rotar la imagen en un número X de grados. Finalmente se ofrecen las opciones de modificar una serie de valores de color de la imagen, para ello es necesario seleccionar el botón de radio en “Modificar”, y se habilitará un seleccionador de color para simular un coloreo a la imagen, también se puede definir los valores de contraste, brillo, saturación y tono, que permiten que la imagen tenga cambio de color sin necesidad de utilizar herramientas más avanzadas.



Figura 2.18 Ventana de propiedades de Imagen

A continuación, se mostrarán una serie de imágenes comparando la imagen original con los valores predeterminados con otra con sus propiedades alteradas. En primer lugar, se tiene la comparación de la imagen predeterminada original junto a la imagen con sus valores de ancho y alto duplicados, esto puede ser observado en la Figura 2.19.

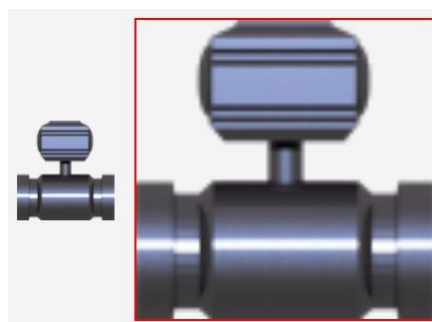


Figura 2.19 Par de imágenes predeterminadas con una imagen con el doble en valores de ancho y alto

Después tenemos la demostración de la utilización de la opción de “Preservar el Radio al Redimensionar” en la Figura 2.20, poniéndose a prueba con una imagen que es rectangular

y que al definirse un ancho y alto que corresponderían al de una imagen cuadrada rellenará toda el área cuadrada si la opción no está habilitada como se ve en la imagen seleccionada, es decir, la imagen con un borde de color rojo, luego en la imagen de la parte inferior derecha se puede ver el resultado si la opción está habilitada, por lo que se tiene un ancho igual al de la figura superior derecha pero con un alto mucho menor.

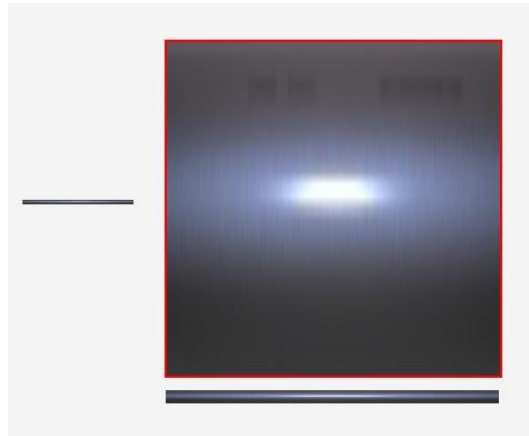


Figura 2.20 Comparación de imágenes con los valores de ancho y alto modificados, pero con la opción de "Preservar el Radio al Redimensionar" activado y desactivado

Llegados a este punto se demostrará el funcionamiento de la opción de rotación para rotar la imagen en la cantidad de grados definida por el usuario, entonces en la Figura 2.21, se expone la misma imagen, pero rotada en distintos valores en grados, la segunda imagen superior se encuentra a 45 grados, la primera imagen inferior se encuentra a 90 grados y la última imagen inferior se encuentra a 180 grados.



Figura 2.21 Conjunto de Imágenes rotadas en 45, 90 y 180 grados

Finalmente, se expondrá la opción de modificación de color de imagen, en este ejemplo de uso se seleccionó un color de luz rojo, así como un contraste con valor de 25 entonces el resultado se puede ver en la Figura 2.22.

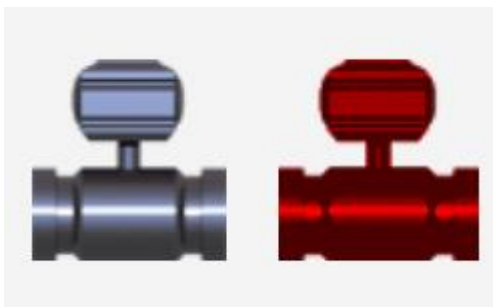


Figura 2.22 Imagen con su par modificado colores y modificado el contraste

En contraste, si cambiamos los valores de brillo a 25 obtenemos un resultado como el de la Figura 2.23.

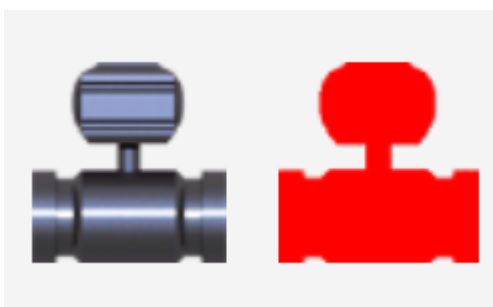


Figura 2.23 Imagen con su par modificado colores incluyendo un valor de brillo 25

En el mismo sentido, si modificamos los valores de contraste a 25 obtenemos un resultado como en la Figura 2.24.

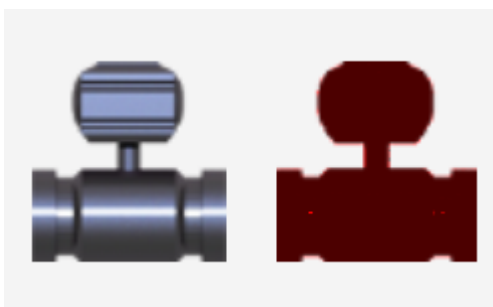


Figura 2.24 Imagen con su par modificado colores incluyendo un valor de contraste 25

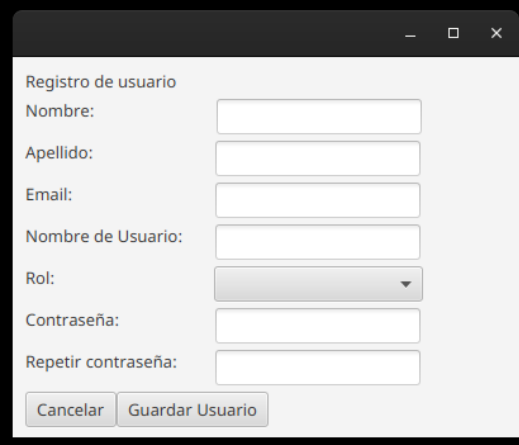
Implementación de ventana de creación de cuentas de usuario

La aplicación está destinada a funcionar para distintos tipos de usuarios con diferentes privilegios, los tipos de usuarios serán:

- Administrador, este tipo de usuario podrá crear usuarios de cualquier tipo, así como todos los privilegios de los demás tipos de usuarios.

- Diseñador, este tipo de usuario podrá crear y editar los proyectos de diseño, así como todos los privilegios del usuario operador.
- Operador, este tipo de usuario se encargará de la ejecución de los proyectos, así como la administración de alarmas en cuando a reconocimiento.

Para cumplir con este requerimiento se realizó una ventana de creación de los distintos tipos de usuarios, donde se han definido distintos campos acerca de un usuario como son: nombre, apellido, email, nombre de usuario, tipo de rol o tipo de usuario, así como la contraseña y confirmación de contraseña, esta ventana se ilustra en la Figura 2.25.



The image shows a software window titled "Registro de usuario". It contains the following fields from top to bottom: "Nombre:" with a text input box; "Apellido:" with a text input box; "Email:" with a text input box; "Nombre de Usuario:" with a text input box; "Rol:" with a dropdown menu; "Contraseña:" with a text input box; and "Repetir contraseña:" with a text input box. At the bottom of the window, there are two buttons: "Cancelar" and "Guardar Usuario".

Figura 2.25 Ventana de registro de usuario

Implementación de ventana de administración de usuarios

El usuario administrador requiere de una opción para poder administrar los distintos tipos de usuarios registrados, así que para ello se implementó una ventana que muestra los distintos usuarios registrados, además cuenta con las opciones para crear un nuevo usuario, editar un usuario y eliminar un usuario, seleccionando una fila de la tabla de usuarios y dando clic derecho desplegando dichas opciones, esto puede observarse en la Figura 2.26.

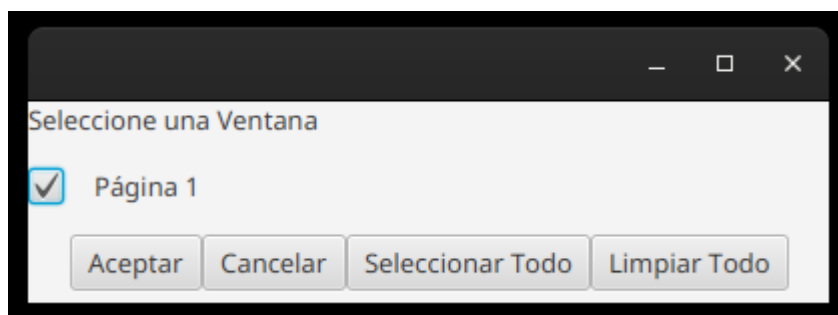


Figura 2.28 Ventana que permite mostrar las distintas páginas disponibles

2.7.4.4 Sprint Review

Tabla 2.11 Criterios de aceptación del Sprint 3

ID	Descripción	Criterio de Aceptación	Cumple
US-01	Como diseñador, quiero agregar representaciones gráficas de objetos del área industrial, para poder visualizar un área de trabajo simulada similar a la que se tiene en el entorno real.	La aplicación permite agregar distintas figuras predeterminadas que representan objetos del área industrial, además se pueden editar las propiedades de dichos objetos como el ángulo de rotación de la figura, si esta será reflejada de forma horizontal o vertical, además de poder editar las propiedades de color de esta.	SI
US-08	Como administrador, quiero crear cuentas de usuario identificadas por usuario y contraseña que pertenezcan a los Roles "Diseñador", "Operador", "Administrador", para poder brindar distintos tipos de privilegios según el tipo de usuario.	La aplicación permite a los usuarios de distintos privilegios autenticarse como tales, además se ha creado una base de datos donde se almacenarán los datos de los usuarios, junto con sus credenciales.	SI
US-10	Como diseñador, quiero organizar mis representaciones gráficas en distintas ventanas y poder moverme entre ellas desde la interfaz diseñada, para poder reducir la cantidad de objetos por pantalla.	La aplicación permite el manejo de varias páginas o ventanas, donde se pueden agregar distintas representaciones gráficas, cada página tiene su conjunto de propiedades características que se definen a través de una ventana al momento de su creación y pueden ser alteradas después.	SI

2.7.4.5 Sprint Retrospective

¿Qué salió bien en la iteración?

La aplicación permite agregar un conjunto de figuras predeterminadas además de poder alterar sus propiedades como el usuario requiera, además la aplicación ahora soporta

múltiples páginas por proyecto permitiendo mayor libertad al usuario para agregar las representaciones gráficas.

¿Qué se puede mejorar?

Es necesario que el usuario también pueda agregar imágenes desde el disco duro y que estas se muestren en la galería de selección de figuras predeterminadas.

2.7.5 Sprint 4

2.7.5.1 Sprint Planning

El objetivo de este Sprint busca implementar las siguientes opciones que son requeridas para el flujo de trabajo del usuario como agregar imágenes desde una fuente externa, guardar archivos que contengan todos los datos del proyecto. Además, se busca agregar la opción que permita la creación de Alarmas definidas por el usuario que verifiquen el estado actual de un tag y notifique al usuario al momento de que dicho estado cambie a uno no deseado. Finalmente se busca añadir personalización al usuario en cuanto a las cifras decimales mostradas dentro de un texto que lea un tag.

2.7.5.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.12.

Tabla 2.12 Backlog Sprint 4

ID	Descripción	Tareas
US-02	Como diseñador, quiero agregar nuevas representaciones gráficas de objetos del área industrial sin afectar la existencia de otros objetos, para poder tener nuevos objetos sin la necesidad de esperar una nueva versión del software.	<ul style="list-style-type: none"> • Agregar una opción dentro de la ventana para agregar figuras que permita importar una imagen externa. • Implementar opción para copiar la imagen importada hacia un directorio que la aplicación siempre leerá. • Implementar opción para siempre leer desde los directorios con sus respectivas categorías y agregar las imágenes importadas correspondientes.
US-09	Como diseñador, quiero definir alarmas, es decir, condiciones en las variables para que al momento de cumplir o no dichas condiciones se notifique al operador cuando se está ejecutando el HMI.	<ul style="list-style-type: none"> • Implementar ventana para crear Alarmas. • Implementar ventana para administrar las alarmas creadas. • Implementar ejecución de la verificación de las Alarmas en hilos ejecutados en segundo plano. • Implementar los estados de Alarma: <ul style="list-style-type: none"> ○ Activada

		<ul style="list-style-type: none"> ○ No activada ○ Reconocida ○ No Reconocida
US-14	Como diseñador, quiero guardar el proyecto en un archivo, para poder continuar con el diseño o que un operador lo pueda ejecutar.	<ul style="list-style-type: none"> ● Implementar la opción para generar un archivo que contenga las páginas del proyecto, los objetos que pertenecen a dichas páginas, así como todas sus propiedades. ● Implementar opción de “Cargar” que permitirá leer un archivo generado y regenerar los objetos dentro de la aplicación. ● Implementar opción de “Guardar” para guardar los cambios realizados en el proyecto dentro de un nuevo archivo si todavía no ha sido seleccionado o dentro del mismo si este ha sido previamente cargado. ● Implementar opción de “Guardar Como”, es decir, generar una copia del proyecto a partir del proyecto actual en uso. ● Implementar opción para verificar y mostrar al usuario si una o varias propiedades de uno o más objetos son erróneas o existieron problemas al regenerar el objeto.
US-17	Como diseñador, quiero definir la precisión de los datos del tipo flotante, para poder reducir o aumentar la visibilidad de los decimales.	<ul style="list-style-type: none"> ● Implementar campo que permita la definición de cuantas cifras decimales se podrán observar después de la coma, en los objetos de representación gráfica que los muestren como un texto o un Rectángulo Dinámico que puede mostrar el valor actual de porcentaje.

2.7.5.3 Ejecución del Sprint

Implementación de opción para importar imágenes

La aplicación requiere que el usuario pueda agregar imágenes de origen externo desde el sistema de archivos. Es entonces que se implementó dicha opción para agregar imágenes al lienzo, para ello en primer lugar se mostrará la ventana de la Figura 2.18, sin embargo existirá un cambio, que el botón “Seleccionar” estará habilitado, permitiendo seleccionar una imagen, luego de seleccionar la imagen el campo de texto se rellenará indicando que se determinó el origen de dicha imagen, y también se podrán aplicar las propiedades de imagen previamente descritas en la serie de figuras desde la Figura 2.19 hasta la Figura 2.24.

A continuación, se muestra la imagen externa agregada al lienzo en la Figura 2.29.



Figura 2.29 Imagen externa agregada al lienzo

Sin embargo, también es necesario que una imagen sea agregada a las imágenes predeterminadas, para que en futuras ejecuciones esta imagen permanezca junto con las demás y pueda ser utilizada sin tener que volver a buscar dicha imagen en el sistema de archivos, entonces se ha agregado un botón de “Importar” al final del conjunto de imágenes predeterminadas como se ilustra en la Figura 2.30, al dar clic en dicho botón se habilitará la ventana de selección de archivos.



Figura 2.30 Seleccionador de imágenes predeterminadas con el botón de "Importar"

A continuación, se ejemplifica como la nueva imagen se ha añadido al conjunto de imágenes predeterminadas en la Figura 2.31.



Figura 2.31 Seleccionador de imágenes predeterminadas con una nueva imagen añadida
Implementación de ventana para definir alarmas

La aplicación requiere la definición de alarmas, es decir, definir una serie de condiciones en base a un tag determinado, para ello se ha creado una ventana que permite definir estos parámetros, como se puede observar en la Figura 2.32, en primera instancia se tiene un campo de texto donde se pueden escribir las condiciones luego de agregar un tag con el botón “Añadir tag”, después existen otros campos como el de precisión de decimales y luego el nombre de la Alarma y su comentario.

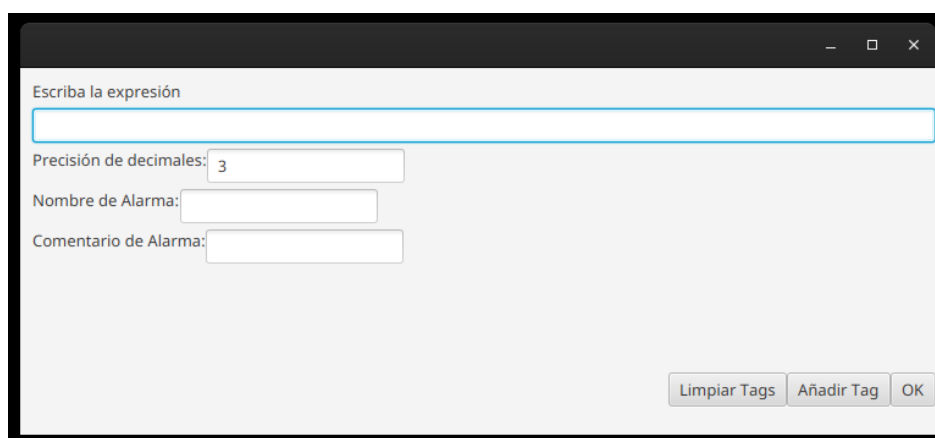
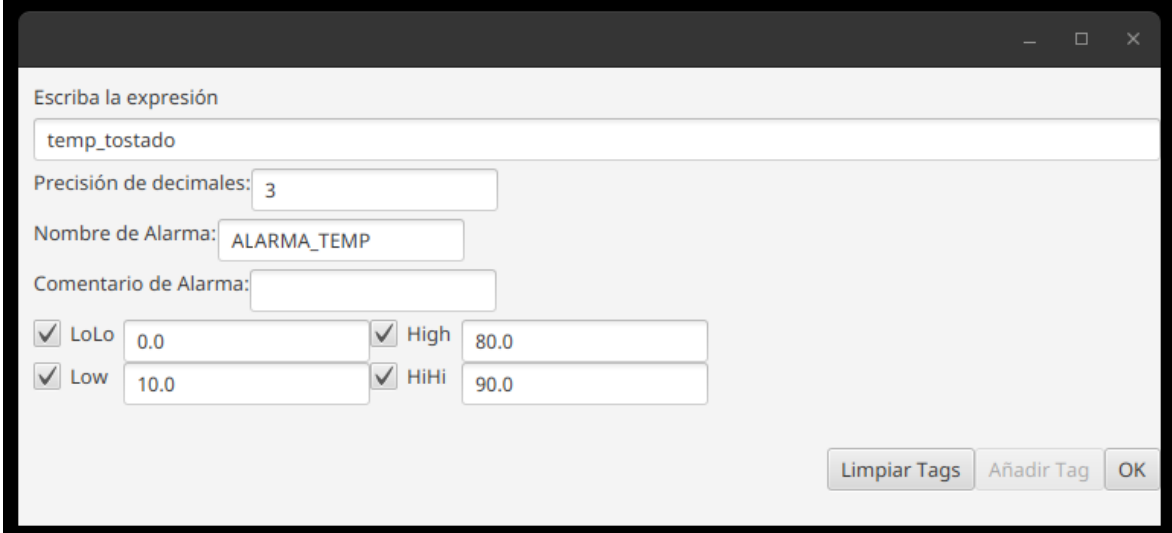


Figura 2.32 Ventana de creación de alarmas

Además, luego de agregar un tag se habilitan las opciones para la creación de condiciones de la Alarma dependiendo del tipo de tag seleccionado, por ejemplo, si se selecciona un tag del tipo flotante se habilitarán las siguientes opciones detalladas en la Figura 2.33, es

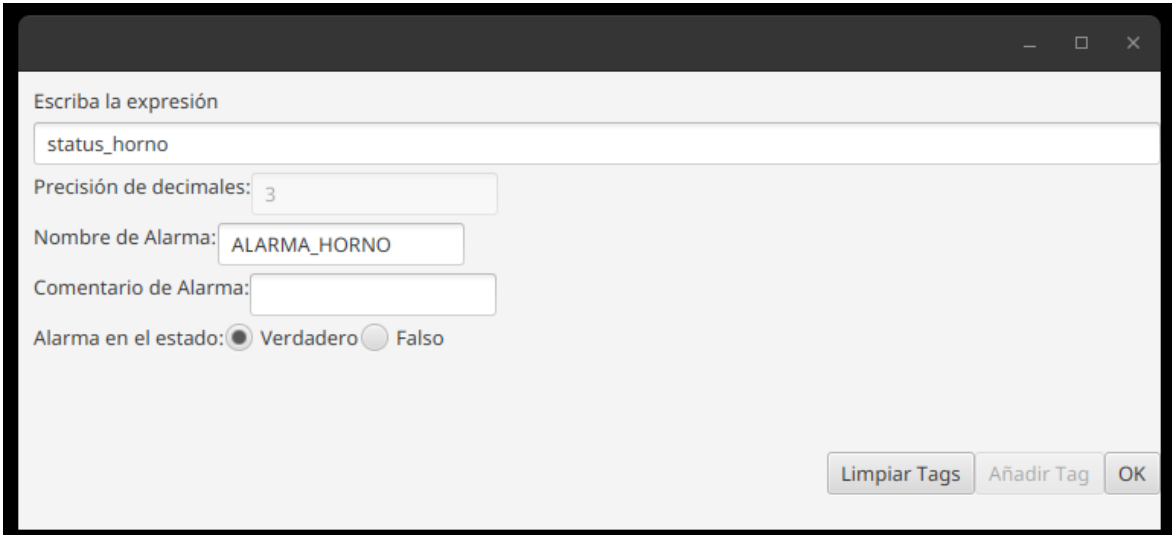
decir, las condiciones de LoLo(Bajo Bajo), Low(Bajo), High(Alto) y HiHi(Alto Alto) estas condiciones se ejecutarán en el momento que el valor actual del tag sea menor que el valor en Low y otra vez será ejecutada si es menor que LoLo, lo mismo aplica cuando el valor del tag supera el límite de High y luego nuevamente si supera el valor de HiHi.



The screenshot shows a window titled "Escriba la expresión" (Write the expression). The expression field contains "temp_tostado". Below it, the "Precisión de decimales" (Decimal precision) is set to 3. The "Nombre de Alarma" (Alarm name) is "ALARMA_TEMP". The "Comentario de Alarma" (Alarm comment) field is empty. There are four checked checkboxes for alarm conditions: "LoLo" with a value of 0.0, "Low" with a value of 10.0, "High" with a value of 80.0, and "HiHi" with a value of 90.0. At the bottom right, there are three buttons: "Limpiar Tags", "Añadir Tag", and "OK".

Figura 2.33 Ventana de creación de Alarmas con un tag del tipo flotante seleccionado

En cambio, si es un tag del tipo "Bool", se habilitarán las opciones de comprobar el estado en "Verdadero" o en "Falso" y cuando el tag cambie a dicho estado se activará la alarma, esto se puede observar en la Figura 2.34.



The screenshot shows a window titled "Escriba la expresión" (Write the expression). The expression field contains "status_horno". Below it, the "Precisión de decimales" (Decimal precision) is set to 3. The "Nombre de Alarma" (Alarm name) is "ALARMA_HORNO". The "Comentario de Alarma" (Alarm comment) field is empty. At the bottom, there is a section "Alarma en el estado:" (Alarm in the state:) with two radio buttons: "Verdadero" (True) which is selected, and "Falso" (False). At the bottom right, there are three buttons: "Limpiar Tags", "Añadir Tag", and "OK".

Figura 2.34 Ventana de creación de alarma con un tag del tipo "Bool"

Implementación de ventana para administrar alarmas

La aplicación requiere de una forma de administrar las distintas alarmas creadas por el usuario, para ello se ha implementado una ventana ejemplificada en la Figura 2.35, donde se muestran las distintas alarmas creadas con sus atributos correspondientes en una tabla de resumen, esta ventana es ilustrada en la Figura 2.35, y al igual que como se mostró en la ventana de la Figura 2.26 de administración de usuarios si se da clic derecho en una fila se habilitarán las opciones de “Nuevo” para crear una nueva alarma, “Editar” para editar la alarma seleccionada, y “Eliminar” para eliminar la alarma seleccionada.

#	Nombre	Alarma	Fecha y Hora	Tipo	Limite Alto	Limite Bajo	Valor	Estado	Reconocimiento
1	ALARMA_HORNO	status_horno&&true	-	-	-	-	-	Desactivada	
2	ALARMA_TEMP	temp_tostado*1	-		90.0	80.0	10.0	0.0	Desactivada

Figura 2.35 Ventana de administración de alarmas

Implementación de opción para guardar un archivo de proyecto

La aplicación requiere la opción para guardar el proyecto con todo el conjunto de figuras y objetos como alarmas dentro de un archivo, para ello se implementó la opción que ejecuta una ventana de selección de archivos del sistema operativo para guardar el proyecto actual en un archivo, como se puede observar a continuación en la Figura 2.36.

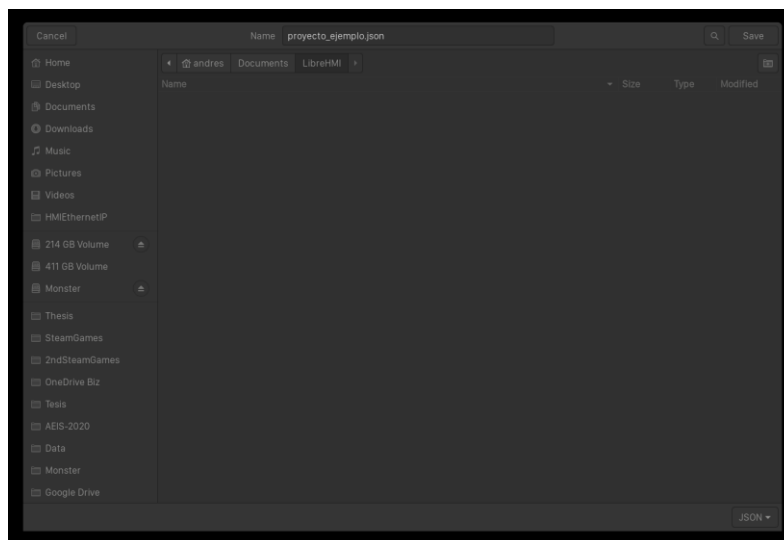


Figura 2.36 Ventana de Guardar Proyecto en sistema operativo Linux con el entorno “Gnome”

En el momento en el cual se guarda el proyecto, el título de la ventana cambia al del nombre del archivo concatenado con el nombre de la página actual como se ilustra en la Figura 2.37.



Figura 2.37 Título de la ventana de la aplicación

Implementación de opción para cargar un archivo de proyecto

De la misma forma se requiere que la aplicación pueda cargar un archivo de proyecto generando todos los objetos del proyecto, para ello también se utiliza la opción que llama a una ventana de selección de archivos del sistema operativo para cargar el proyecto como se puede ver en la Figura 2.38.

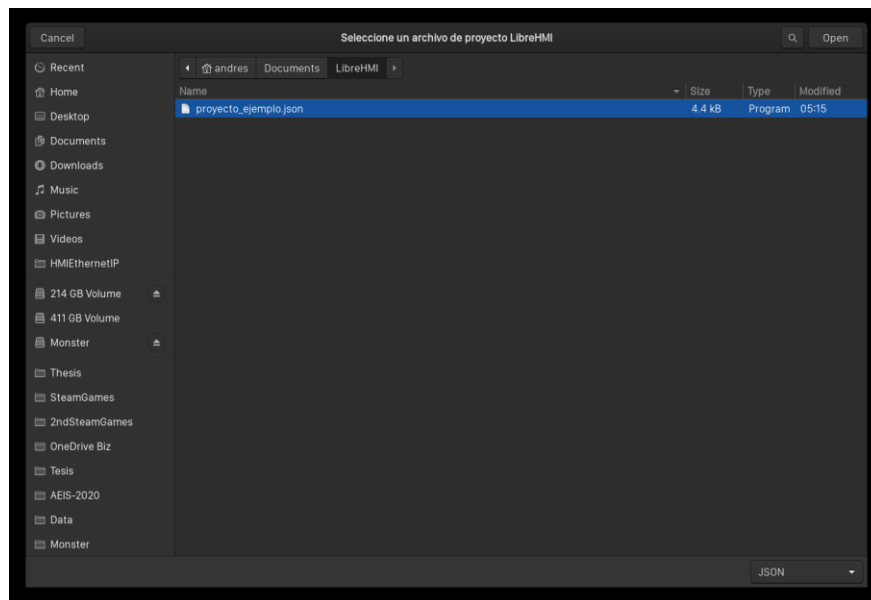


Figura 2.38 Ventana de carga de proyectos

Implementación de opción para definir el número de cifras en los textos

Como la aplicación realiza lecturas desde la base de datos de valores decimales estos pueden variar en el número de cifras decimales por lo que el usuario requiere poder definir el número de decimales que un texto pueda tener para ser mostrado, para ello al momento de definir un texto de representación gráfica existe el campo de "Precisión de decimales" donde se definirá el número de decimales requeridos por el usuario, en primer lugar se mostrará la ventana de definición de tags en la Figura 2.39.

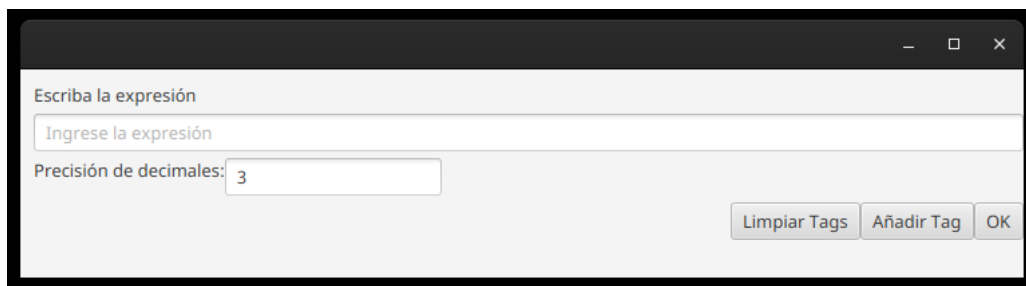


Figura 2.39 Ventana de asociación de una expresión para la representación gráfica de "texto"

Finalmente se ejemplifican las dos representaciones gráficas de texto una con el valor de precisión de decimales en 3 y el seleccionado en 2.



Figura 2.40 Dos representaciones gráficas de texto asociadas al mismo tag con el número de precisión de decimales en 3 y en 2.

2.7.5.4 Sprint Review

Tabla 2.13 Criterios de aceptación del Sprint 4

ID	Descripción	Criterio de Aceptación	Cumple
US-02	Como diseñador, quiero agregar nuevas representaciones gráficas de objetos del área industrial sin afectar la existencia de otros objetos, para poder tener nuevos objetos sin la necesidad de esperar una nueva versión del software.	La aplicación permite importar nuevas figuras dentro del lienzo, que podrán ser utilizadas en futuras ejecuciones, a través de la ventana para agregar figuras predeterminadas y dando clic en un ícono de "+" disponible en cada categoría, luego de importar la imagen esta se mostrará como si fuera una de las figuras predeterminadas.	SI
US-09	Como diseñador, quiero definir alarmas, es decir, condiciones en las variables para que al momento de cumplir o no dichas condiciones se notifique al operador cuando se está ejecutando el HMI.	La aplicación permite la creación de alarmas que verifican el estado de distintos tags, a través de una ventana donde se pueden definir dichos estados. Además, se pueden administrar las alarmas creadas para editarlas o eliminarlas. También, se ha creado un objeto de representación gráfica que puede agregarse al lienzo, y reaccionará cuando una alarma se active, permitiendo también cambiar el estado de una alarma "No	SI

		Reconocida” hacia el estado de “Reconocida”.	
US-14	Como diseñador, quiero guardar el proyecto en un archivo, para poder continuar con el diseño o que un operador lo pueda ejecutar.	La aplicación permite guardar un proyecto con todas las Representaciones Gráficas agregadas a las distintas páginas dentro de un archivo que permitirá luego regenerar estos objetos a través de la lectura de dicho archivo, es decir, utilizando la opción de “Cargar”. Además, se implementó la opción de “Guardar Como” que permite crear un nuevo archivo a partir de un proyecto previamente guardado.	SI
US-17	Como diseñador, quiero definir la precisión de los datos del tipo flotante, para poder reducir o aumentar la visibilidad de los decimales.	La aplicación permite definir cuantas cifras decimales después de la coma serán mostradas en un texto que se muestre en el lienzo.	

2.7.5.5 Sprint Retrospective

¿Qué salió bien en la iteración?

La aplicación permite agregar imágenes desde el disco local, además de ello se puede importar imágenes y que estén disponibles desde la galería de imágenes predeterminadas. Asimismo, ahora el usuario puede guardar sus proyectos con todas sus propiedades para utilizarlos más tarde.

¿Qué se puede mejorar?

La aplicación requiere de la implementación de los modos de edición y ejecución. Estos modos de aplicación permitirán al usuario interactuar de manera más profesional con la interfaz HMI, dentro del modo de ejecución solo se podrán observar las representaciones gráficas con tags asociados para su monitoreo, mientras que en el modo de edición se podrán agregar o quitar las representaciones y moverlas dentro del lienzo como el usuario lo requiera.

2.7.6 Sprint 5

2.7.6.1 Sprint Planning

El objetivo de este Sprint busca agregar las opciones que vuelven más interactiva la experiencia de usuario como son agregar una ventana de bienvenida cuando se ejecuta la aplicación, así como la implementación de distintos menús en la ventana de diseño que permiten acceder a las distintas opciones disponibles al usuario Diseñador. Siguiendo en este razonamiento, es clara la necesidad de implementar distintos modos de ejecución de

la aplicación, que se ven reflejados en el modo de “Diseño” y el modo de “Ejecución”. Finalmente, dentro de este Sprint se añadirán tres nuevos tipos de Representaciones Gráficas como son el “Resumen de Alarmas” que mostrará y permitirá interactuar con las alarmas y sus estados, así como dos figuras geométricas como son la Línea y la Elipse.

2.7.6.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.14.

Tabla 2.14 Backlog Sprint 5

ID	Descripción	Tareas
US-12	Como operador, quiero poder acceder a un "Resumen de Alarmas", para poder interactuar con un resumen de las alarmas actuales y pasadas, donde pueda reconocer, suprimir, deshabilitar, archivar o quitar las alarmas.	<ul style="list-style-type: none"> • Implementar un objeto que se pueda agregar al lienzo llamado Resumen de Alarmas, donde se puedan visualizar en tiempo real las alarmas que se activen y se pueda cambiar el estado de una alarma de “No Reconocida” a “Reconocida”. • Implementar opción para permitir que esta representación gráfica se guarde dentro del archivo de datos de proyecto. • Implementar opción para regenerar la representación gráfica con sus propiedades desde el archivo leído.
US-15	Como diseñador, quiero tener una ventana de inicio donde pueda acceder a los proyectos previamente creados y acceder a ellos, para facilitar el acceso a los proyectos previamente iniciados, así como iniciar proyectos nuevos.	<ul style="list-style-type: none"> • Implementar área donde se encuentren los botones para crear un nuevo proyecto y otro para abrir un proyecto. • Implementar área donde se muestren los proyectos recientes. • Implementar lógica para registrar los archivos recientes a partir de los últimos archivos abiertos o guardados. • Implementar dos botones para iniciar los modos de Edición y Ejecución en los proyectos recientes listados.
US-19	Como usuario, quiero un menú con las distintas representaciones gráficas en forma de íconos para poder agregarlas al lienzo.	<ul style="list-style-type: none"> • Agregar los íconos que representen cada una de las Representaciones Gráficas disponibles para agregar al lienzo. • Implementar un menú a modo de cuadrícula de botones con los íconos correspondientes para cada representación gráfica. • Implementar “tooltips” para cada botón que describa la representación gráfica a agregarse.

US-20	<p>Como usuario quiero un menú horizontal superior con las opciones de archivo, editar, alarmas, configurar y ventanas para permitir al usuario acceder a ellas.</p>	<ul style="list-style-type: none"> • Implementar un menú que contenga las opciones de "Archivo" como son: "Nuevo" para crear un proyecto, "Abrir" para abrir un archivo de proyecto, "Recientes" para ver los últimos archivos de proyecto utilizados, "Guardar" para guardar el archivo de proyecto actual, "Guardar Como" para crear un nuevo archivo de proyecto, "Salir" para cerrar la aplicación. • Implementar el menú de Edición con las opciones para "copiar" la representación gráfica seleccionada, "cortar" para realizar la acción de cortar la representación gráfica seleccionada, y "pegar" para pegar una representación gráfica desde el portapapeles hacia el lienzo. • Implementar el menú de Alarmas con las opciones para "Agregar Alarma" que despliega la ventana de creación de alarma, "Administrar Alarmas" para mostrar la ventana de administración de alarmas. • Implementar el menú de Configuración con las opciones de "Usuarios" para la administración de usuarios y de "Editar Propiedades de Conexión de la Base de datos" para mostrar una ventana que permita editar las opciones de conexión de la base de datos.
US-21	<p>Como usuario quiero un menú para poder acceder a las distintas pátinas creadas para poder a través de las distintas páginas creadas.</p>	<ul style="list-style-type: none"> • Implementar el menú de Ventanas para poder mostrar una ventana que permiten la selección de las distintas ventanas y mostrarlas.
US-22	<p>Como usuario quiero un modo de edición donde se puede modificar el archivo de proyecto y un modo de ejecución donde solo se podrá visualizar los objetos del proyecto, para poder utilizar la aplicación en estos distintos modos.</p>	<ul style="list-style-type: none"> • Implementar modo de ejecución que bloquee cualquier tipo modificaciones de los objetos del lienzo mientras la aplicación se encuentre en este modo. Además, se bloquearán los menús de "Editar", "Alarmas", "Configurar". • Implementar lógica de solicitud de inicio de sesión del tipo administrador cuando se dé clic en una representación gráfica de

		<p>entrada de valores, estos luego de un tiempo se bloquearán automáticamente.</p> <ul style="list-style-type: none"> • Dentro del menú de “Archivo” se mostrarán las opciones de cambiar de modo de “Ejecución” hacia el modo de “Edición” y viceversa.
US-24	<p>Como usuario quiero un área donde se muestren los distintos menús del lienzo que se posicionará en el lado izquierdo de la aplicación para se podrá mostrarla u ocultarla y así no perder espacio en la pantalla para agregar representaciones gráficas.</p>	<ul style="list-style-type: none"> • Implementar el menú de agregar objetos dentro de esta área. • Implementar el menú de manejo de ventanas en esta área. • Implementar un botón que permita mostrar u ocultar el menú al dar clic en él.
US-25	<p>Como usuario quiero poder agregar una figura geométrica como una línea y un círculo para poder utilizarlas en el proyecto.</p>	<ul style="list-style-type: none"> • Implementar la opción para agregar una línea y poder modificar sus propiedades. • Implementar la opción para agregar una elipse y poder modificar sus propiedades. • Implementar las opciones de redimensionamiento, de rotación y de modificación de color.

2.7.6.3 Ejecución del Sprint

Implementación de representación gráfica de resumen de alarmas”

La aplicación requiere una forma de administrar las distintas alarmas creadas por el usuario dentro del mismo lienzo como parte de las representaciones gráficas sin necesidad de acceder a un menú adicional, para ello se implementó un resumen de alarmas, que muestra una tabla de las alarmas y se irá agregando dinámicamente una fila con cada alarma que se active, con la opción de dar clic derecho y cambiar el estado de la alarma de “No Reconocida” a “Reconocida”, a continuación se muestra la representación gráfica sin ninguna alarma activada en la Figura 2.41, donde es evidente que es la misma tabla en cuanto a columnas disponibles que la tabla de la ventana de administración mostrada en la Figura 2.35, sin embargo, no se muestra ninguna alarma y es porque solo se mostrarán cuando una alarma sea activada.

de un triángulo con su punta hacia la derecha indicando que se iniciará el proyecto en el modo de ejecución, esta ventana se detalla en la Figura 2.45.

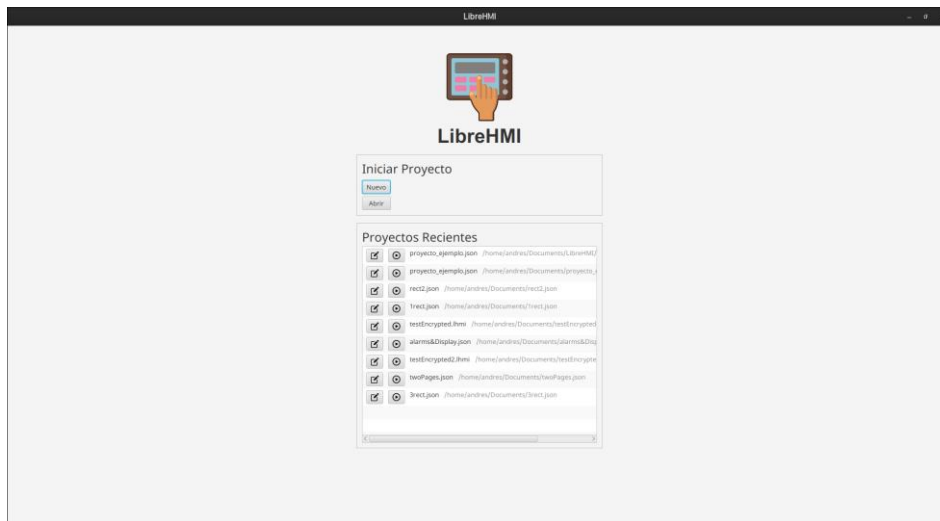


Figura 2.45 Ventana de bienvenida de la aplicación

Implementación de menú para agregar Representaciones Gráficas

La aplicación permite añadir una serie de representaciones gráficas al lienzo, por lo que es necesaria la creación de un menú que permita seleccionarlas, entonces se crearon una serie de 10 botones posicionados dentro de una cuadrícula de cuatro filas y tres columnas, con la última fila conteniendo un único botón, este menú se ejemplifica en la Figura 2.46.



Figura 2.46 Menú para agregar Representaciones Gráficas

Implementación de menú superior

La aplicación tiene una serie de opciones que el usuario requiere acceder como son las opciones de Archivo, Editar, Alarmas, Configurar, y Ventanas, este menú se posicionará en la parte superior de la aplicación, y se lo puede observar en la Figura 2.47.

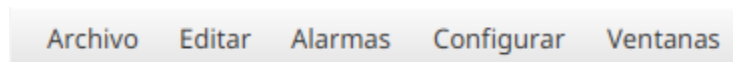


Figura 2.47 Menú de opciones que se posicionará en la parte superior de la ventana de la aplicación

Cada uno de estos menús, contendrán submenús que serán descritos a continuación, en primer lugar, se tiene el menú de “Archivos” como se representa en la Figura 2.48.

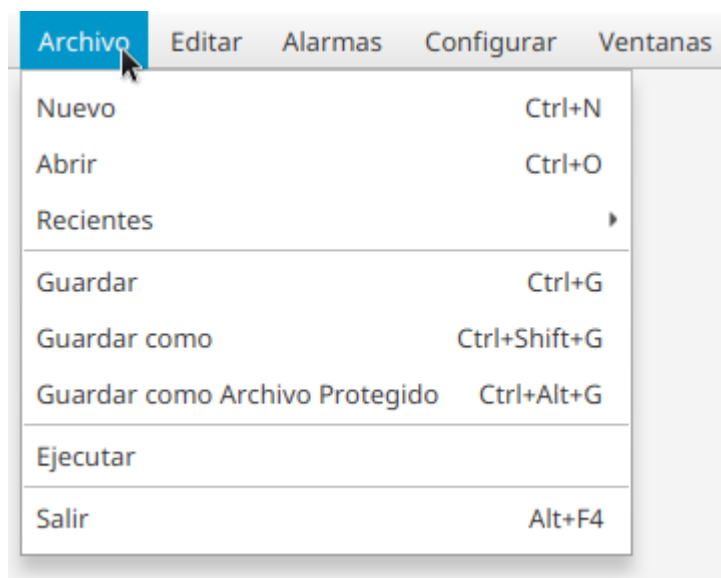


Figura 2.48 Menú de archivos con sus opciones disponibles

El comportamiento de las opciones del menú de “Archivos” se describirán a continuación:

- Nuevo, al dar clic en este submenú o utilizar la combinación de teclas “Ctrl + N”, se iniciará el procedimiento para iniciar la creación de un nuevo proyecto, es decir, limpiar la interfaz, verificando en primer lugar que no existan cambios, en el caso de existir algún cambio no guardado, se mostrará el siguiente cuadro de dialogo solicitando la confirmación como se ve en la Figura 2.49.

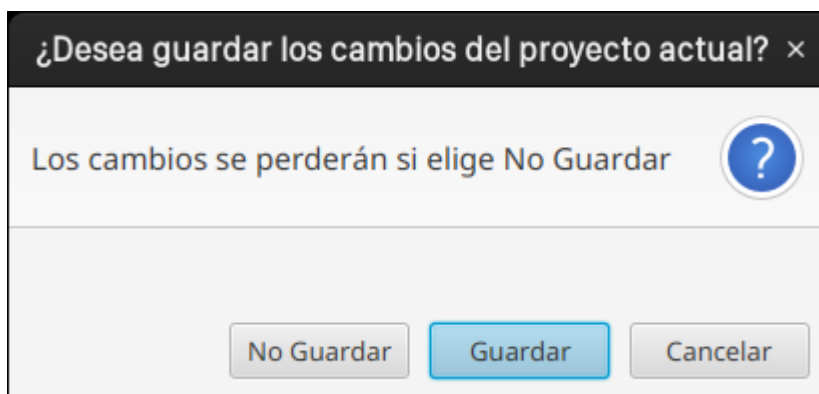


Figura 2.49 Cuadro de diálogo de confirmación para guardar cambios

- Abrir, esta opción iniciará la carga de un archivo de proyecto y una vez más se verificará en primer lugar que el proyecto actual no tenga ningún cambio en el caso de tenerlo se mostrará el cuadro de diálogo de la Figura 2.49.
- Recientes, en esta opción se mostrará en otro submenú los archivos recientes que también se muestran en la ventana de inicio.
- Guardar, con esta opción se ejecutará el procedimiento de guardado del proyecto actual.
- Guardar Como, con esta opción se ejecutará el procedimiento de generar una copia a partir del proyecto actual.
- Guardar Como Archivo Protegido, para futuras implementaciones iniciará el proceso de guardado de un archivo de proyecto cifrado con contraseña.
- Ejecutar o Editar, este menú permite cambiar entre los modos de ejecución que serán implementados más adelante indicado a través del texto descrito en el momento actual.
- Salir, este menú permitirá cerrar la aplicación y también iniciará el proceso de verificación que el proyecto actual no tenga ningún cambio en el caso de tenerlo se mostrará el cuadro de diálogo de la Figura 2.49.

Luego se tiene el menú de “Editar” que contiene una serie de opciones que estarán disponibles solamente en el modo de edición, como se pueden ver en la Figura 2.50.

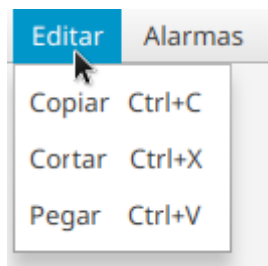


Figura 2.50 Menú de editar con sus opciones disponibles

El comportamiento de las opciones del menú de “Archivos” se describirán a continuación:

- Copiar, que permite iniciar el proceso de copiado de la representación gráfica seleccionada actualmente.
- Cortar, que permite iniciar el proceso de cortar de la representación gráfica seleccionada actualmente.

- Pegar, que permite iniciar el proceso de pegado de la representación gráfica desde el portapapeles hacia el lienzo.

Después se tiene el menú de “Alarmas” que contiene las opciones para la creación y administración de alarmas y estará habilitado en el modo de edición solamente, esto puede ser observado en la Figura 2.51.

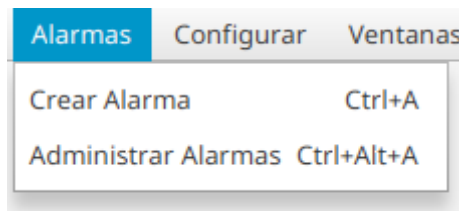


Figura 2.51 Menú de Alarmas con sus opciones disponibles

El comportamiento de las opciones del menú de “Alarmas” se describirán a continuación:

- Crear Alarma, que inicia el proceso de creación de alarmas mostrando la ventana de creación de alarmas de la Figura 2.32.
- Administrar Alarmas, que inicia la ventana de administración de alarmas de la Figura 2.35.

Ahora se tiene el menú de “Configurar” que contiene las opciones para definir una serie de opciones requeridas por el usuario para esto en primer lugar se mostrarán las opciones y sus submenús como se muestra en la Figura 2.52.

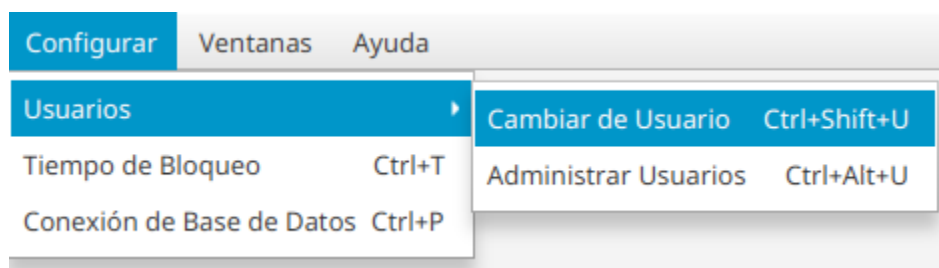


Figura 2.52 Menú de Configurar con sus opciones disponibles

El comportamiento de las opciones del menú de “Configurar” se describirán a continuación:

- Usuarios, que contiene un submenú con las opciones de usuario para cambiar del usuario hacia otro. También se tiene la opción de acceder a la ventana de Administrar Usuarios mostrada en la Figura 2.26.

- Tiempo de Bloqueo, iniciará un proceso de definición del tiempo de autobloqueo de las representaciones gráficas de entrada de valores, como el campo de texto, la slider, etc., que solicitan que el usuario inicie sesión como administrador para poder utilizarlos y luego de un tiempo en segundos se volverán a bloquear. Para esto se ha implementado una ventana de definición de este valor que se puede observar en la Figura 2.53.

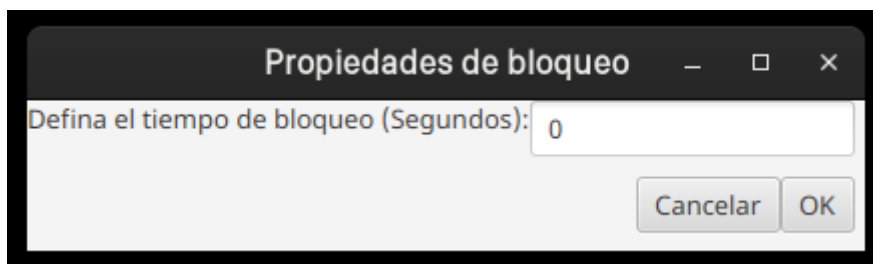


Figura 2.53 Ventana de definición del tiempo de bloqueo

- Conexión de Base de Datos, que iniciará el proceso para la definición de los parámetros necesarios para conectarse a la base de datos como son la dirección del host, puerto, usuario y contraseña, que serán almacenados en un archivo de texto plano dentro del sistema operativo, la ventana es ilustrada en la Figura 2.54.

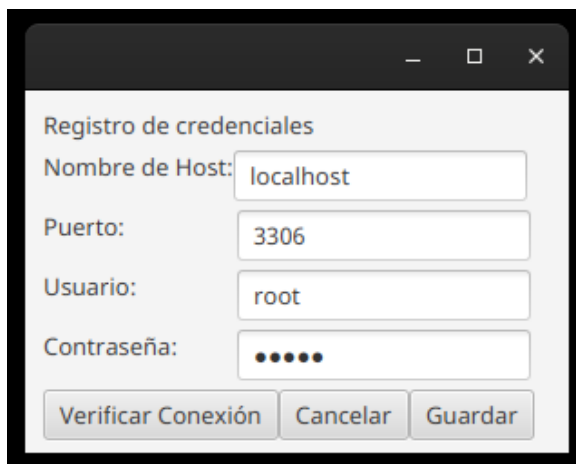


Figura 2.54 Ventana de definición de parámetros de conexión a la base de datos

Finalmente se tiene el menú de “Ventanas” que contiene las opciones para mostrar ventanas, así como otra para importar ventanas en el proyecto que será implementado más adelante, el menú con sus opciones se ilustrará en la Figura 2.55.



Figura 2.55 Menú de Ventanas con todas sus opciones

Implementación de menú para navegar entre las páginas

La aplicación requiere que en el modo de edición se pueda cambiar entre las distintas páginas en las cuales se está trabajando por lo que se ha implementado una lista en la cual se puede seleccionar y cambiar entre páginas rápidamente, este menú se mostrará en la parte izquierda inferior y se detalla de la siguiente forma en la Figura 2.56.



Figura 2.56 Lista de páginas disponibles para su navegación

Además, desde esta lista, si se selecciona una página y se da clic derecho se mostrarán las opciones que se detallan en la Figura 2.57.

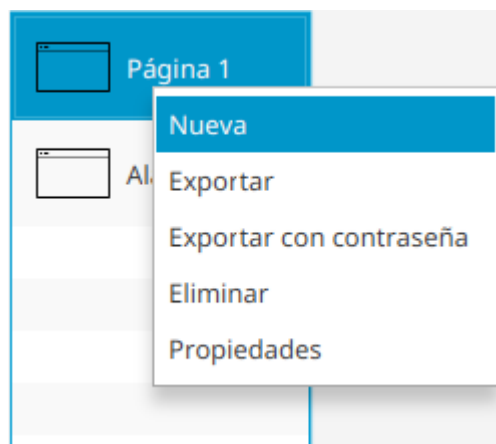


Figura 2.57 Opciones de página al dar clic derecho

El comportamiento de las opciones de página se describirá a continuación:

- Nueva, para crear una nueva página mostrando la ventana de la Figura 2.27.

- Exportar, para futuras implementaciones permitirá exportar una ventana en un archivo.
- Exportar con contraseña, para futuras implementaciones permitirá exportar una ventana en un archivo cifrado.
- Eliminar, iniciará un cuadro de dialogo de confirmación para eliminar la página del proyecto, como en la Figura 2.58.

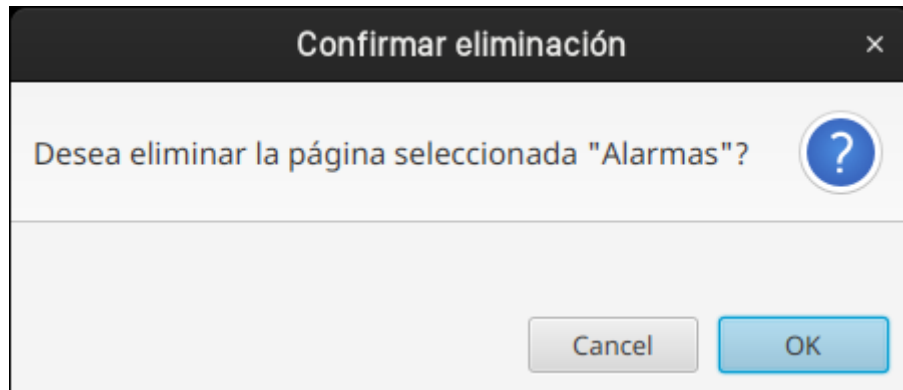


Figura 2.58 Cuadro de dialogo de confirmación de eliminación de página

- Propiedades, que permitirá editar las propiedades de la página y mostrará la ventana de la Figura 2.27.

Implementación de modo de edición

La aplicación como se ha comentado previamente requiere de dos modos de funcionamiento diferentes el primero a describirse será el modo de edición que permitirá agregar las representaciones gráficas o agregar alarmas, además de ofrecer las opciones de edición de “copiar”, “cortar” y “pegar”.

En este modo las representaciones gráficas de entrada de datos permanecen bloqueadas para permitir así arrastrar las representaciones gráficas a lo largo de todo el lienzo, y las demás representaciones gráficas que no son de entrada de datos se encuentran desbloqueadas para poder moverlas a lo largo de todo el lienzo, además en este modo se permite la “Selección” de las Representaciones Gráficas en la cual la representación gráfica añadirá un borde de color rojo y permitirá también las opciones de edición previamente descritas. Este comportamiento se ve en la Figura 2.59.

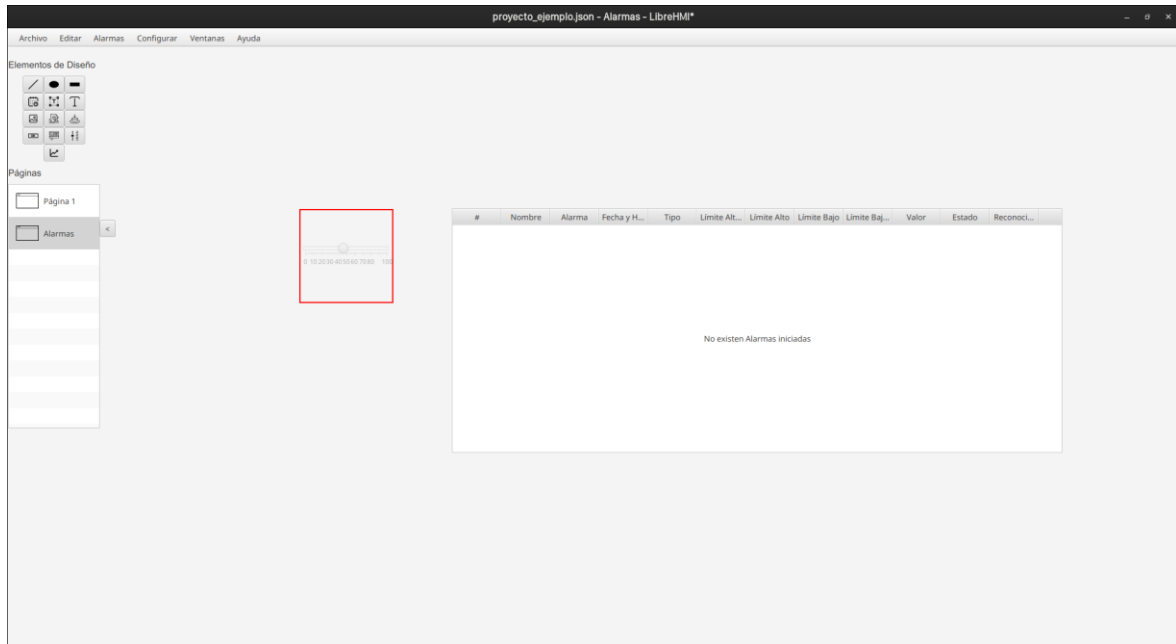


Figura 2.59 Funcionamiento de la aplicación en el modo de edición

Implementación de modo de ejecución

La aplicación también requiere pasar del modo de edición a un modo de ejecución, este modo es útil para el usuario del tipo operador, ya que aquí no se podrán editar las posiciones de las representaciones gráficas ni tampoco agregar nuevas, además en este modo ciertas opciones del menú superior se ven limitadas, finalmente se puede observar este comportamiento en la Figura 2.60

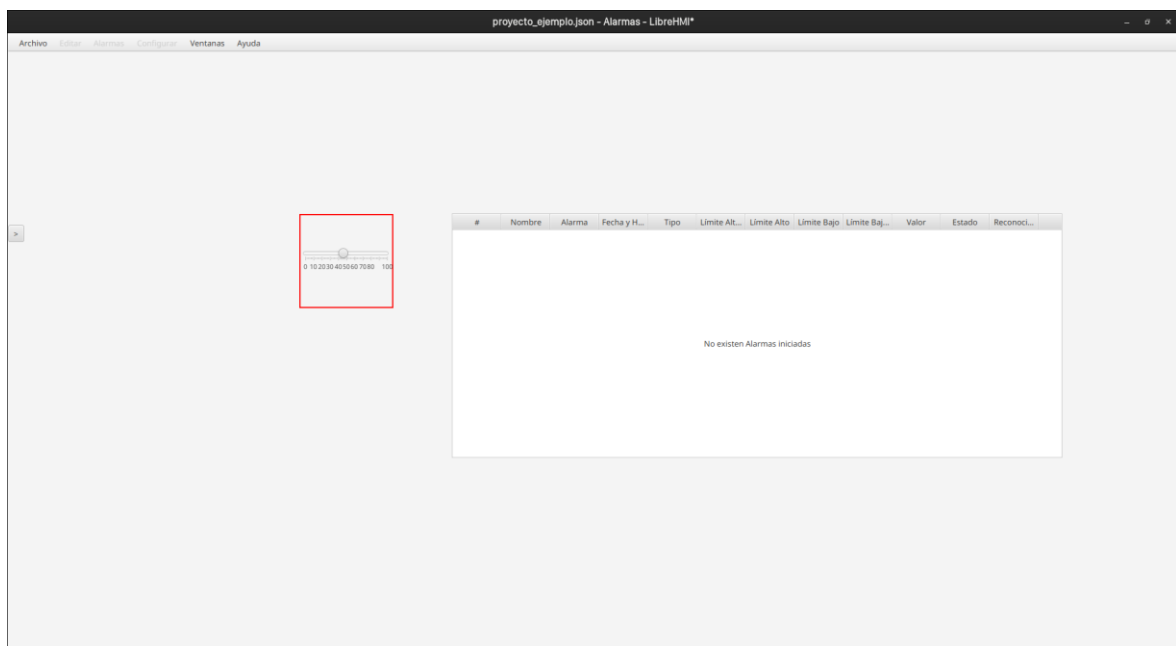


Figura 2.60 Funcionamiento de la aplicación en el modo de ejecución

Implementación de área donde se posicionarán los menús de navegación de ventanas y menú de agregar Representaciones Gráficas

Los menús para agregar representaciones gráficas, así como el menú para navegar entre las distintas páginas, se han posicionado en un área dinámica que puede mostrarse y ocultarse, como se puede observar en las figuras referenciadas anteriormente con dicha área mostrada en la Figura 2.59 y oculta en la Figura 2.60.

Implementación de representación gráfica de figura geométrica de línea

La aplicación requiere poder agregar una línea, para ello el usuario debe acceder a través del menú para agregar representaciones gráficas en el modo de edición y luego dar un clic para definir el principio de la línea y otro clic para definir el final de la línea, ejemplos de distintas líneas agregadas se podrán mostrar en la Figura 2.61.

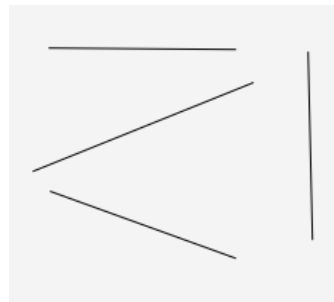


Figura 2.61 Diferentes líneas agregadas al lienzo

Implementación de Representaciones Gráficas de Elipse

La aplicación requiere poder agregar una elipse, para ello el usuario debe acceder desde el menú para agregar representaciones gráficas en el modo de edición y luego dar clic en alguna parte del lienzo, en dicho momento se agregará una elipse con propiedades predeterminadas, para poder modificar las propiedades de la figura se puede dar clic derecho sobre la figura y dar clic en la opción de “Propiedades”, esta figura agregada al lienzo y después de dar clic derecho se ilustra en la Figura 2.62.

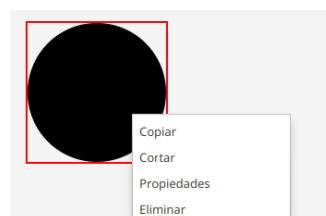


Figura 2.62 Figura de elipse después de dar clic derecho sobre ella

Después de dar clic en propiedades se mostrará una ventana que permite modificar los atributos de la figura, donde se cargarán los valores actuales de dicha figura y se permite modificarlos, como se detalla en la Figura 2.63.

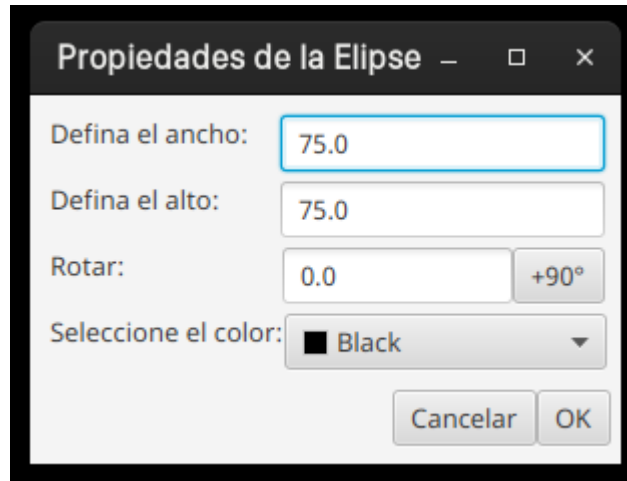


Figura 2.63 Ventana de propiedades de Elipse

Si se alteran las propiedades de la figura a un ancho de 150, luego define la figura a 45 grados y un color azul, se obtendrá un resultado ejemplificado en la Figura 2.64.

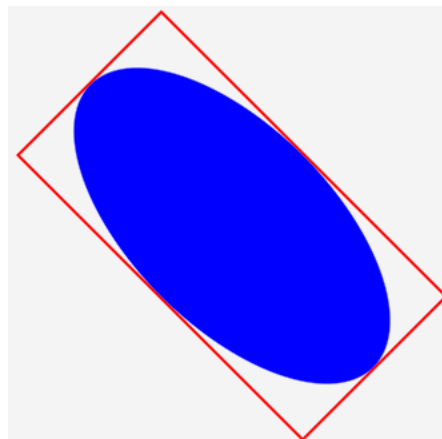


Figura 2.64 Figura de elipse con sus atributos modificados por el usuario

2.7.6.4 Sprint Review

Tabla 2.15 Criterios de aceptación Sprint 5

ID	Descripción	Criterio de Aceptación	Cumple
US-12	Como operador, quiero poder acceder a un "Resumen de Alarmas", para poder interactuar con un resumen de las alarmas actuales y pasadas, donde pueda reconocer, suprimir,	La aplicación permite agregar una representación gráfica de resumen de alarmas, que evaluará las alarmas registradas del proyecto cada segundo y actualizará la tabla con una nueva fila que muestre los	SI

	deshabilitar, archivar o quitar las alarmas.	detalles de dicha alarma, además permite la administración de las alarmas en cuanto a su estado, es decir, cambiarlas de “No Reconocida” a “Reconocida”.	
US-15	Como diseñador, quiero tener una ventana de inicio donde pueda acceder a los proyectos previamente creados y acceder a ellos, para facilitar el acceso a los proyectos previamente iniciados, así como iniciar proyectos nuevos.	La aplicación implementa la ventana de bienvenida que es la primera en mostrarse al usuario, y ofrece las opciones para crear nuevo proyecto o cargar uno previo, además de una lista de proyectos recientemente abiertos.	SI
US-19	Como usuario, quiero un menú con las distintas representaciones gráficas en forma de íconos para poder agregarlas al lienzo.	La aplicación cuenta con un menú que permite agregar las distintas representaciones gráficas al lienzo, en este menú se representan las 13 disponibles con un botón con un ícono para cada una.	SI
US-20	Como usuario quiero un menú horizontal superior con las opciones de archivo, editar, alarmas, configurar y ventanas para permitir al usuario acceder a ellas.	La aplicación cuenta con un menú superior que ofrece las distintas opciones para Archivo, Editar, Alarmas, Configurar y Ventanas.	SI
US-21	Como usuario quiero un menú para poder acceder a las distintas pántinas creadas para poder a través de las distintas páginas creadas.	La aplicación implementa una lista de ventanas disponibles que se muestra en una ventana para su selección y poder mostrarlas.	SI
US-22	Como usuario quiero un modo de edición donde se puede modificar el archivo de proyecto y un modo de ejecución donde solo se podrá visualizar los objetos del proyecto, para poder utilizar la aplicación en estos distintos modos.	La aplicación permite dos modos de funcionamiento, en el modo de edición se puede modificar las propiedades de los objetos, mientras que en el modo de ejecución todos los objetos quedan inmóviles.	SI
US-24	Como usuario quiero un área donde se muestren los distintos menús del lienzo que se posicionará en el lado izquierdo de la aplicación para se podrá mostrarla u ocultarla y así no perder espacio en la pantalla para agregar representaciones gráficas.	La aplicación implementa un área dinámica que agrega el menú para agregar representaciones al lienzo, así como una lista de páginas disponibles, está área tendrá un comportamiento dinámico a través de un botón que lo muestra y lo oculta.	SI
US-25	Como usuario quiero poder agregar una figura geométrica como una línea y un círculo	La aplicación permite agregar las representaciones gráficas de una línea y una elipse al lienzo, con las mismas propiedades comunes	SI

	para poder utilizarlas en el proyecto.	de las demás representaciones gráficas, así como sus propiedades específicas.	
--	--	---	--

2.7.6.5 Sprint Retrospective

¿Qué salió bien en la iteración?

La aplicación permite añadir la representación gráfica de resumen de alarmas donde se podrá administrar las distintas alarmas en cuestión de cambiarlas de estado “No reconocida” a “Reconocida”. También se implementó la ventana de bienvenida que se mostrará al momento de abrir la aplicación, así como los menús para acceder a las distintas opciones disponibles.

¿Qué se puede mejorar?

La aplicación tiene actualmente las figuras geométricas básicas como son rectángulo, elipse y línea, sin embargo, se podría habilitar la opción al usuario de implementar figuras geométricas más complejas como un polígono de N puntos, luego de que el usuario defina esa serie de puntos a través de dar clic.

2.7.7 Sprint 6

2.7.7.1 Sprint Planning

El objetivo de este Sprint busca implementar la representación del gráfico de tendencias y un conjunto de opciones requeridas, inicialmente se graficarán los valores actuales de un conjunto de tags seleccionados, posteriormente se podrán filtrar los datos graficados a partir de una fecha y hora de inicio y de fin. Además, para poder diferenciar entre los tags se añadirá una opción para definir el color de la línea asociada a cada tag. También, el usuario podrá definir los tiempos de actualización del conjunto de tags, Finalmente se implementará una opción para poder exportar los datos registrados de los tags.

2.7.7.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.16.

Tabla 2.16 Backlog Sprint 6

ID	Descripción	Tareas
US-14	Como Diseñador, Quiero Agregar gráficos de tendencias asociadas a una de las variables que pueden ser representadas con una ventana de X número de segundos hacia atrás, Para poder	<ul style="list-style-type: none"> • Implementar la representación gráfica de Gráfico de tendencias. • Implementar opción para hacer “Zoom”, es decir, filtrar los límites inferior y superior en términos de fecha y hora, mostrando solamente los datos pertenecientes a este rango.

	<p>monitorizar como se ha desarrollado el comportamiento de dicha variable</p>	<ul style="list-style-type: none"> • Implementar opción para hacer “Zoom Out” para regresar los datos al análisis de los datos nuevos en tiempo real. • Implementar opción para agregar uno o varios tags y leer su valor actual y representarlo en tiempo real dentro de una gráfica de líneas. • Implementar opción para definir un tiempo de muestreo de lectura de los tags. • Implementar opción para ocultar y mostrar un tag definido dentro del Gráfico de Tendencias. • Implementar opción para definir colores para cada línea que representa a un tag. • Implementar opción para exportar los datos en un archivo CSV.
--	--	---

2.7.7.3 Ejecución del Sprint

Implementación de gráfico de tendencias

La aplicación requiere de la implementación de un gráfico de tendencias, dentro de esta se enlazarán diferentes tags cuyos valores se irán leyendo a lo largo del tiempo cada X número de segundos permitiendo construir un gráfico de valores contra el tiempo. Para agregar la figura, en primer lugar, se dará clic en el botón correspondiente del área de menú y se mostrará una ventana para definir los parámetros necesarios de la figura, detallada en la Figura 2.65. En el primer campo mostrado sirve para definir el tiempo de muestreo en segundos, es decir, cada cuantos segundos se registrarán los valores de los tags asociados para poder graficarlos, después se tienen los campos de ancho y alto para definir el tamaño que el gráfico tendrá en el lienzo, luego se tiene un campo para definir la rotación, y finalmente se tiene el área donde se podrán asociar los tags a ser graficados, entonces, se muestra al principio un “Checkbox” para habilitar los campos, luego se tiene el campo para definir el nombre a ser mostrado que no necesariamente será siempre el mismo del tag, después se tiene el botón para definir una expresión, que generalmente será el valor directo de un tag aunque también puede aplicarse una operación matemática, al dar clic en dicho botón se mostrará la ventana de la Figura 2.39, y el último campo definirá el color de la línea a graficarse.

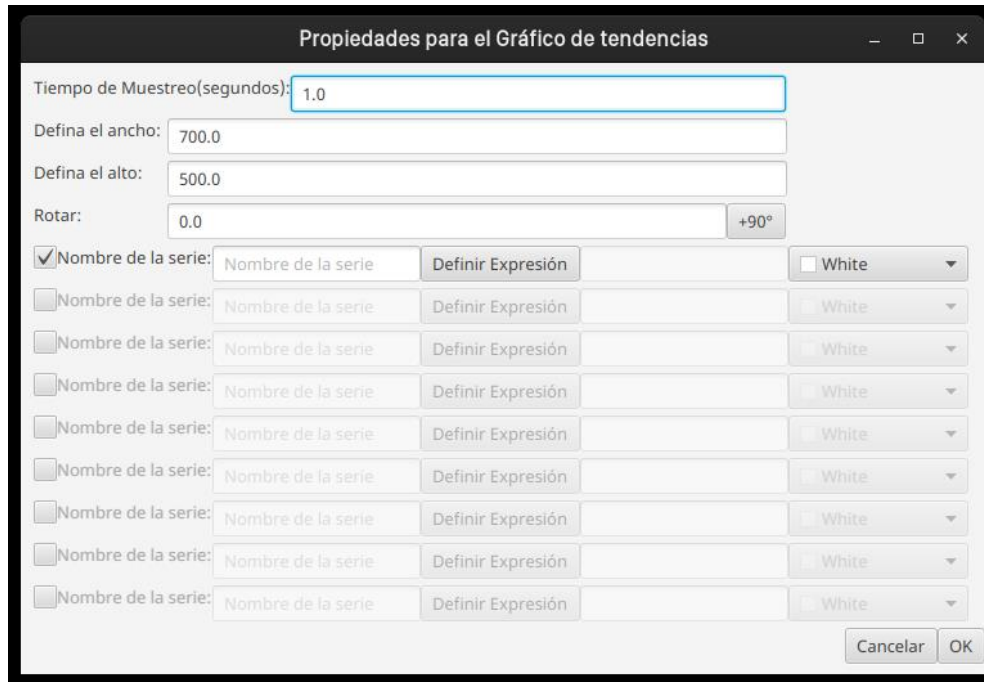


Figura 2.65 Ventana de definición de parámetros para el gráfico de tendencias

A continuación, se mostrará en la Figura 2.66 el resultado de un gráfico de tendencias con dos tags asociados además se tiene una serie de opciones que serán detalladas más adelante, como son el nombre de cada una de las líneas junto con dos checkboxes, luego debajo de estas se tiene una serie de campos como un para definir una fecha y hora con un slider, y finalmente en la parte más inferior se tienen cuatro botones.

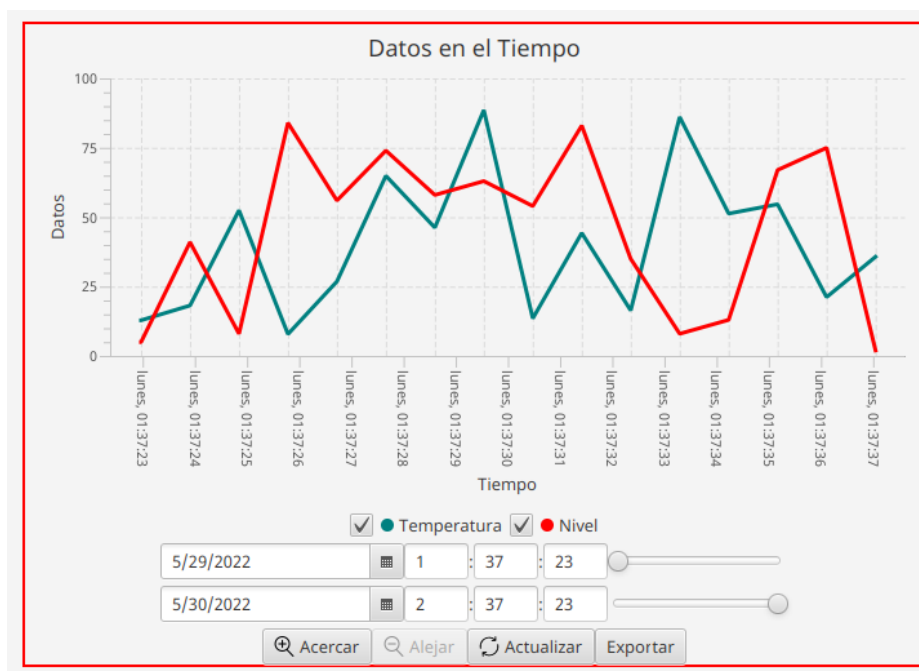


Figura 2.66 Gráfico de tendencias con dos tags asociados

Implementación de Zoom In

El gráfico de tendencias requiere poder filtrar los datos recopilados, para ello se ha implementado dos áreas de campos que permiten definir una fecha y hora, asimismo se podrá definirlo desde la slider, ejemplificaremos este comportamiento en la Figura 2.67, donde se puede ver que el botón “Alejar” se encuentra habilitado, además también se puede hacer más “Zoom In”, es decir, filtrar aún más los datos recopilados.

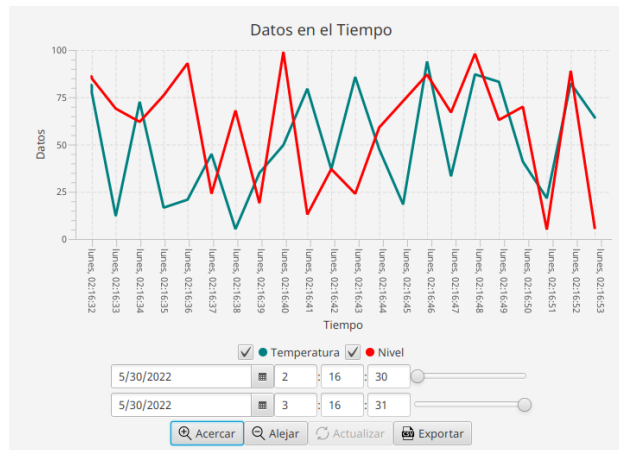


Figura 2.67 Gráfico de tendencias con datos filtrados

Implementación de Zoom Out

Luego de que se ha hecho un “Zoom In”, el usuario requiere también regresar al estado inicial del gráfico, es decir, con sus datos actualizándose segundo a segundo, por lo que para ello se ha implementado a través del botón “Alejar” luego de dar clic se regresa al principio con los datos de fecha y hora actualizados, esto puede observarse en la Figura 2.68.

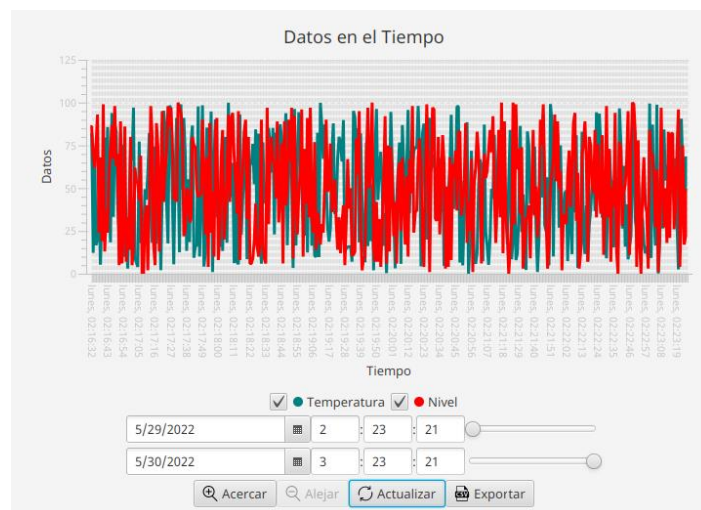


Figura 2.68 Gráfico de tendencias actualizado luego de haber hecho un "Zoom Out"

Implementación de ocultar un tag definido dentro del gráfico de tendencias

La aplicación requiere que el usuario tenga la opción de poder ocultar un tag que está siendo graficado, para ello basta deseleccionar el “checkbox” que esta debajo del gráfico de tendencias, este comportamiento es posible realizarlo también con los datos filtrados para ejemplificarlo se mostrará el Gráfico de tendencias ocultado con la serie “Nivel” luego de hacer “Zoom In” en la Figura 2.69.

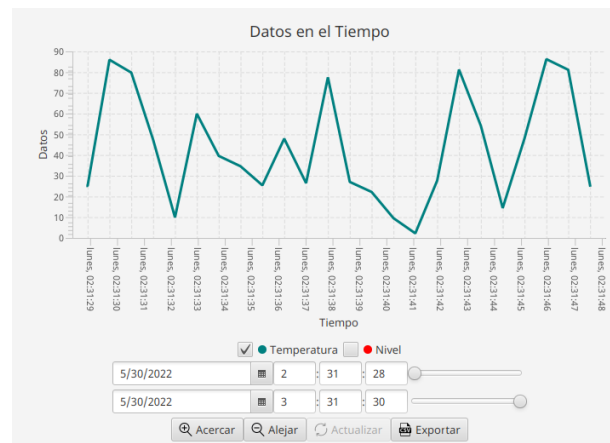


Figura 2.69 Gráfica de tendencias con una serie oculta y realizado un "Zoom In"

Implementación de opción para exportar los datos del gráfico hacia un archivo CSV

Los datos recopilados por el Gráfico de tendencias pueden ser requeridos por el usuario, por lo que se ha implementado una opción para poder exportar los datos en un archivo CSV, para ello basta con dar clic en el botón “Exportar” y se mostrará una ventana de selección de archivos para generar un archivo donde se guardarán los datos, esta ventana se ejemplifica en la Figura 2.70.

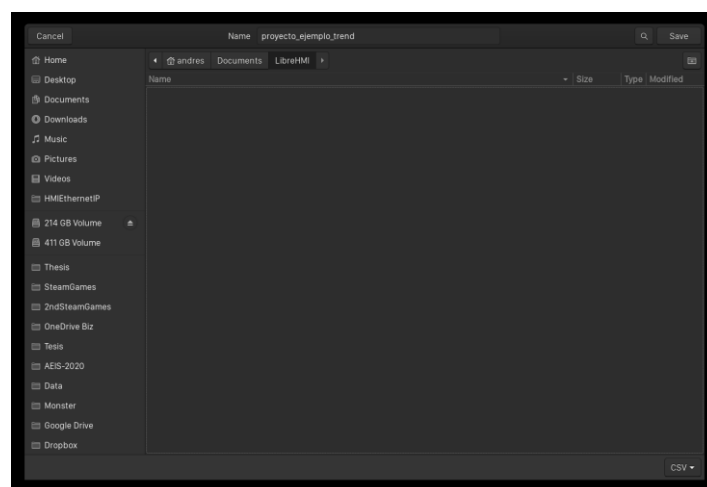


Figura 2.70 Ventana de exportar datos del gráfico de tendencias

Al momento de abrir el archivo exportado en un editor de texto, se puede observar su contenido, en el cual se presentarán en primer lugar la columna “fecha y hora” y después se tendrá una columna para cada tag asociado, este contenido es ilustrado en la

```

1 "Fecha y Hora", "Temperatura", "Nivel"
2 "2022/05/30 02:16:32", "81.4499", "86.0"
3 "2022/05/30 02:16:32", "77.4729", "85.0"
4 "2022/05/30 02:16:33", "12.1793", "69.0"
5 "2022/05/30 02:16:34", "72.623", "62.0"
6 "2022/05/30 02:16:35", "16.6166", "76.0"
7 "2022/05/30 02:16:36", "20.8498", "93.0"
8 "2022/05/30 02:16:37", "44.9781", "24.0"
9 "2022/05/30 02:16:38", "5.26871", "68.0"
10 "2022/05/30 02:16:39", "35.0017", "19.0"
11 "2022/05/30 02:16:40", "49.6945", "99.0"
12 "2022/05/30 02:16:41", "79.5161", "13.0"
13 "2022/05/30 02:16:42", "36.9337", "37.0"
14 "2022/05/30 02:16:43", "85.7509", "24.0"
15 "2022/05/30 02:16:44", "47.5622", "59.0"
16 "2022/05/30 02:16:45", "18.351", "73.0"
17 "2022/05/30 02:16:46", "93.965", "87.0"
18 "2022/05/30 02:16:47", "33.2282", "67.0"
19 "2022/05/30 02:16:48", "87.1105", "98.0"
20 "2022/05/30 02:16:49", "83.1763", "63.0"
21 "2022/05/30 02:16:50", "40.9796", "70.0"
22 "2022/05/30 02:16:51", "21.6097", "5.0"
23 "2022/05/30 02:16:52", "82.3916", "89.0"
24 "2022/05/30 02:16:53", "64.3991", "6.0"
25 "2022/05/30 02:16:54", "75.3412", "26.0"
26 "2022/05/30 02:16:55", "9.82511", "86.0"
27 "2022/05/30 02:16:56", "19.9791", "7.0"
28 "2022/05/30 02:16:57", "3.03001", "33.0"
29 "2022/05/30 02:16:58", "40.818", "14.0"
30 "2022/05/30 02:16:59", "78.1148", "80.0"
31 "2022/05/30 02:17:00", "66.2692", "5.0"
32 "2022/05/30 02:17:01", "97.0669", "15.0"
33 "2022/05/30 02:17:02", "9.09222", "62.0"
34 "2022/05/30 02:17:03", "6.55708", "54.0"
35 "2022/05/30 02:17:04", "4.00087", "53.0"

```

Figura 2.71 Datos exportados de un gráfico de tendencias en un archivo CSV

2.7.7.4 Sprint Review

Tabla 2.17 Criterio de aceptación del Sprint 6

ID	Descripción	Criterio de Aceptación	Cumple
US-14	Como Diseñador, Quiero Agregar gráficos de tendencias asociadas a una de las variables que pueden ser representadas con una ventana de X número de segundos hacia atrás, Para poder monitorizar como se ha desarrollado el comportamiento de dicha variable	La aplicación permite agregar un gráfico de tendencias, donde se pueden asociar distintos tags con sus respectivas líneas de color que irán variando acorde a los valores leídos desde la base de datos. Además, este gráfico de tendencias permite filtrar los datos representados en una fecha y hora de inicio y una de fin. Finalmente se pueden exportar los datos recopilados por el gráfico de tendencias en un archivo CSV.	SI

2.7.7.5 Sprint Retrospective

¿Qué salió bien en la iteración?

La aplicación permite agregar un gráfico de tendencias de una serie de tags que el usuario puede asociar, este gráfico también cuenta con las opciones para filtrar, definir los colores de las líneas y exportar los datos recopilados por el gráfico.

¿Qué se puede mejorar?

El gráfico de tendencias en ejecución puede consumir demasiados recursos al graficar continuamente cada segundo a lo largo del tiempo, la aplicación requiere que este gráfico se ejecute por periodos de tiempo de más de 24 horas, por lo que es necesario optimizarlo para que pueda ser utilizado en dicho tiempo sin que la memoria del sistema se llene por completo.

2.7.8 Sprint 7

2.7.8.1 Sprint Planning

El objetivo de este Sprint busca añadir las últimas funcionalidades requeridas como son la definición de contraseñas para proteger los archivos de proyecto, así como exportar e importar solamente páginas o ventanas de un proyecto hacia otro. Además, en este sprint se implementarán las opciones para permitir a los usuarios definir el intervalo de tiempo de lectura de los tags utilizados, también se busca la implementación de un tag local que no dependa de un valor leído desde la base de datos del controlador.

2.7.8.2 Sprint Backlog

Se seleccionaron las siguientes historias de usuario, mostradas en la Tabla 2.18.

Tabla 2.18 Backlog Sprint 7

ID	Descripción	Tareas
US-04	Como Diseñador, Quiero Definir los tiempos de actualización del estado de las variables, Para poder diferenciar entre las representaciones que requieran más prioridad en su monitoreo	<ul style="list-style-type: none">• Añadir campo de texto para definir el tiempo de actualización en las ventanas de asociación de Tags.• Implementar lógica para leer el atributo definido en el campo, y actualizar los hilos en el tiempo definido.
US-16	Como Diseñador, Quiero Definir contraseñas maestras en los HMI desarrollados, Para Poder Controlar quien puede ejecutar y editar los diseños	<ul style="list-style-type: none">• Implementar la encriptación y desencriptación de un archivo en base a una contraseña maestra.• Implementar ventana para definir contraseñas al archivo de proyecto.
US-18	Como usuario quiero agregar un tag que se ejecuta localmente sin necesidad de la base de datos	<ul style="list-style-type: none">• Implementar tag local que se comporte de la misma forma que un tag leído desde la base de datos.• Implementar ventana de creación de tag local

		<ul style="list-style-type: none"> • Implementar ventana de administración de tag locales creados.
US-23	Como usuario quiero poder exportar e importar solamente las ventanas hacia otro proyecto	<ul style="list-style-type: none"> • Implementar la exportación de una ventana en un archivo. • Implementar la opción para exportar una ventana con contraseña. • Implementar la importación de una ventana al proyecto. • Implementar la importación de una ventana con contraseña.

2.7.8.3 Ejecución del Sprint

Implementación de ventana para definir contraseña de un archivo de proyecto

La aplicación requiere una opción para cifrar un archivo de proyecto, para ello se ha implementado una ventana de definición de contraseñas, para esto el usuario debe dirigirse hacia el menú superior y dar clic en el menú de “Archivos” y luego dar clic en “Guardar como Archivo Protegido”, es entonces que se mostrará la ventana de para la selección de archivos del sistema operativo de la Figura 2.72.

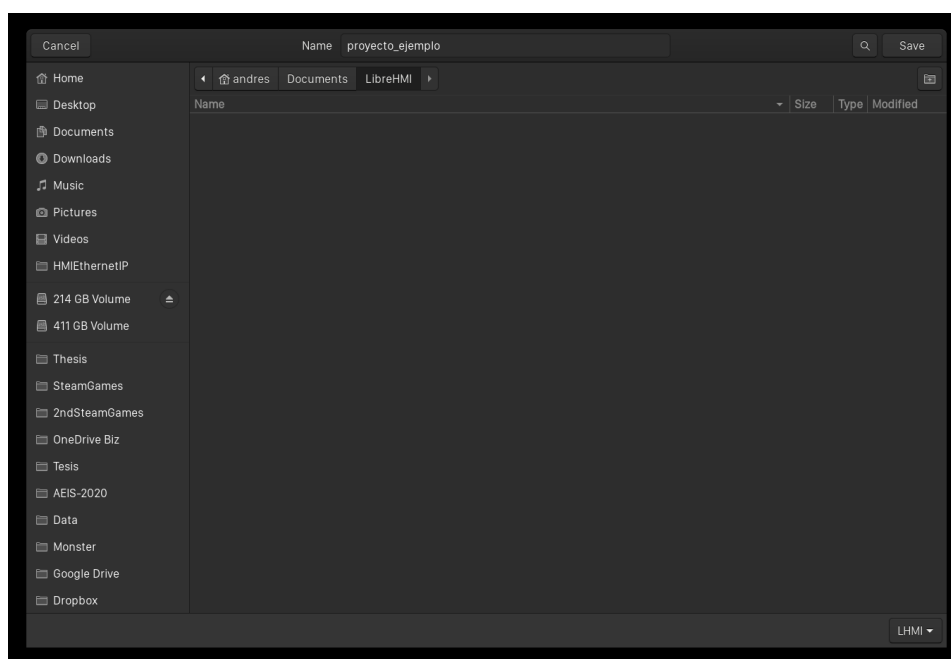


Figura 2.72 Ventana de Guardado de proyecto cifrado

Después de seleccionado un archivo y dar clic en el botón de guardar de la ventana de la Figura 2.72, se mostrará la ventana para definir la contraseña del archivo como se ilustra en la Figura 2.73, y se creará un archivo cifrado.

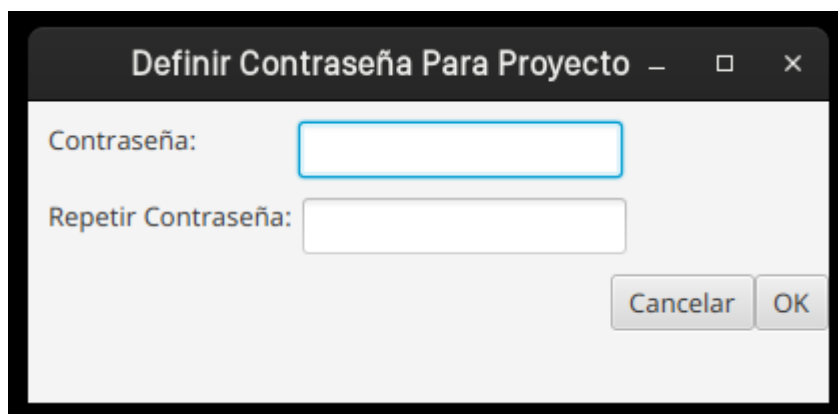


Figura 2.73 Ventana de definición de contraseña para archivo de proyecto cifrado
Implementación de ventana para verificar la contraseña de un archivo de proyecto

Luego de que se ha cifrado un archivo de proyecto es cargarlo, para ello se ha implementado otra ventana que solicitará la contraseña, y dicha ventana se muestra en la Figura 2.74.

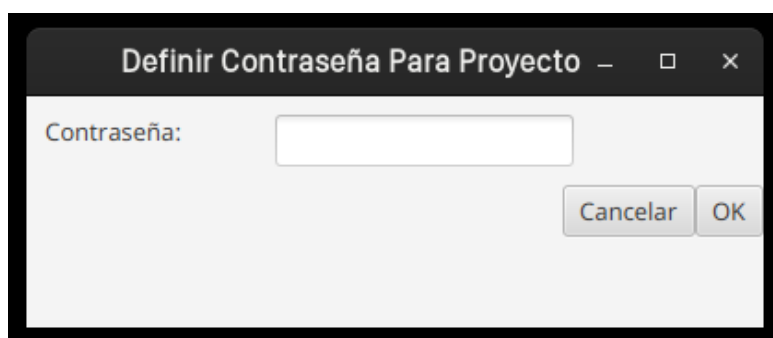


Figura 2.74 Ventana para definir contraseña de descifrado del archivo de proyecto
Implementación de opción de exportar página a un archivo de proyecto

La aplicación requiere habilitar al usuario la opción de exportar las páginas del proyecto para ser importadas más tarde en otro proyecto, para ello se ha implementado dicha opción al acceder a la lista de navegación de páginas, seleccionar una de ellas y dar clic derecho en ella, donde se ejecuta el menú contextual de la Figura 2.57, entonces se puede dar clic en la opción de Exportar, y se mostrará la ventana para la selección de archivos que se ilustra en la Figura 2.75, también se puede dar clic en la opción de Exportar con Contraseña, donde sucederá un procedimiento similar al de “Exportar Archivo de Proyecto con Contraseña”.

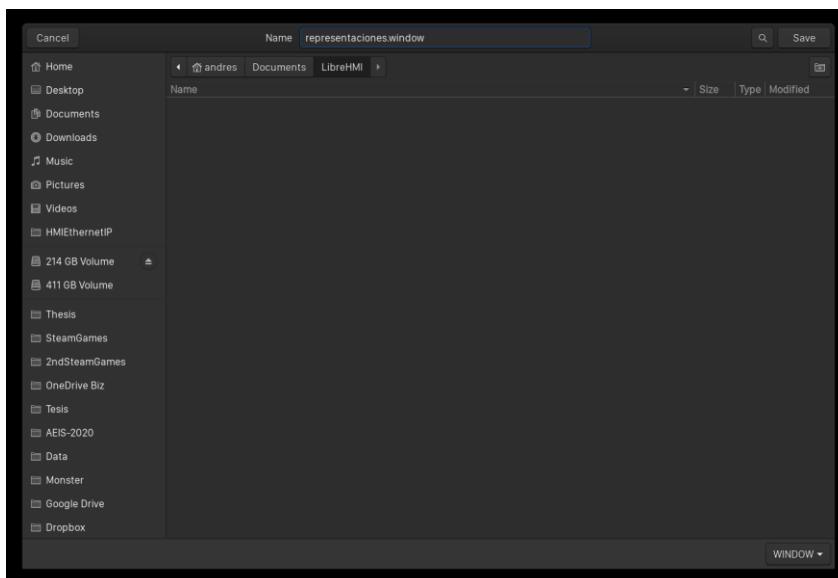


Figura 2.75 Ventana para exportar una ventana

Implementación de opción para importar página a un archivo de proyecto

Evidentemente al poder exportar páginas se debe habilitar una opción para importarlas, para ello el usuario debe acceder al menú superior y en el menú de “Ventanas” en la opción de “Importar Ventana” que mostrará una ventana para seleccionar archivos como se detalla en la Figura 2.76.

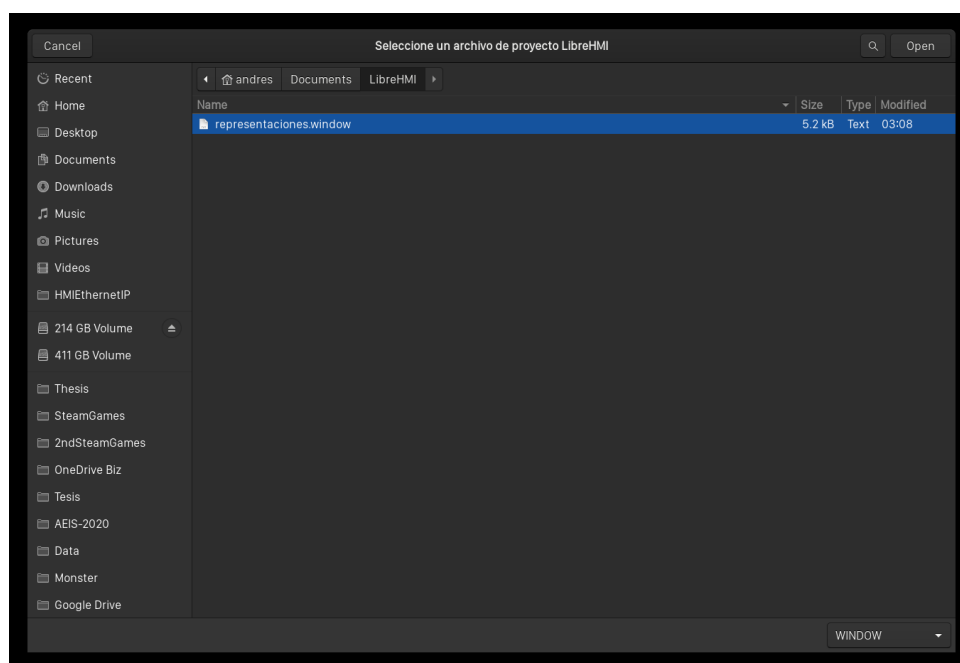


Figura 2.76 Ventana para importar un archivo de página

Implementación de ventana de creación de tag local

La aplicación requiere de un tag que no dependa de la base de datos, es decir, que actúe localmente dentro del proyecto en ejecución solamente, para esto se ha implementado una ventana de creación de tag locales, que permite la definición de los distintos atributos pertenecientes a un tag, tales como, el nombre, el tipo como flotante, entero o bool, y la acción como escritura o lectura como se muestra en la Figura 2.77.

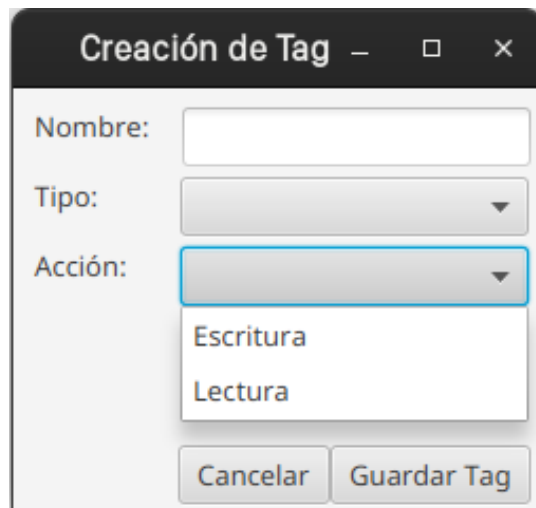


Figura 2.77 Ventana de creación de tag

Al momento de seleccionar un tipo se habilitará la opción para definir el valor inicial del tag, si este es seleccionado como del tipo numérico, es decir, entero o flotante, se habilitará un campo de texto donde se podrá escribir el valor del tag, se puede observar este comportamiento en la Figura 2.78.

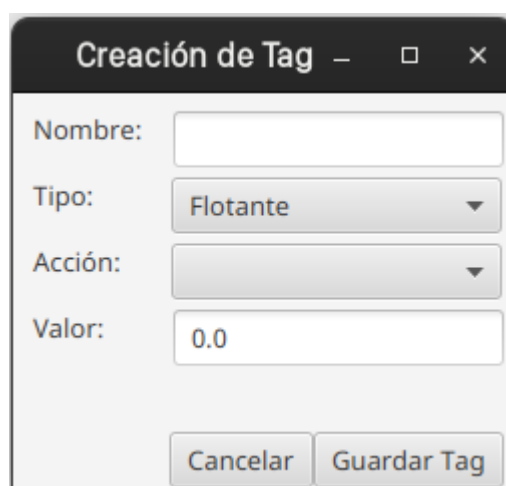


Figura 2.78 Ventana de creación de tag local para el tipo Numérico

Si en cambio se selecciona un tag del tipo bool se habilitarán las opciones para seleccionar el valor en verdadero o falso, como se puede ejemplificar en la Figura 2.79.

Figura 2.79 Ventana de creación de tag Local para el tipo Bool

Luego de que se ha creado un tag local, este se mostrará en la ventana de selección de tags cuando se requiera enlazarlo a una representación gráfica, como por ejemplo en la siguiente ventana para asociar un tag de la Figura 2.80.

Nombre del PLC	IP	Grupo	Nombre del Tag	Tipo	Dirección	Acción
plc1	192.168.1.11	tostado	status_molino1	Bool	B3:0/2	Escritura
plc1	192.168.1.11	tostado	status_horno	Bool	B3:0/0	Escritura
plc1	192.168.1.11	tostado	sp_temp_tostado	Flotante	F8:3	Escritura
plc2	192.168.1.12	mezclado	salida_leche	Bool	B3:0/2	Escritura
plc2	192.168.1.12	mezclado	salida_azucar	Bool	B3:0/1	Escritura
plc2	192.168.1.12	mezclado	sp_vel_mezclador	Entero	N7:0	Escritura
plc2	192.168.1.12	mezclado	status_mezclador	Bool	B3:0/0	Escritura
LibreHMI	localhost	Local	TAG_LOCAL_CREADO	Flotante	Local	Escritura

Figura 2.80 Ventana de selección de tag con un tag Local disponible

Implementación de ventana para administrar tag Locales

Los tags locales creados requieren de una opción para poder editar su valor predeterminado, así como eliminarlos del proyecto, para ello se ha implementado una ventana donde se muestran los tags locales con una opción al dar clic derecho para crear uno nuevo, editar el tag seleccionado y eliminarlo como se ilustra en la Figura 2.81.

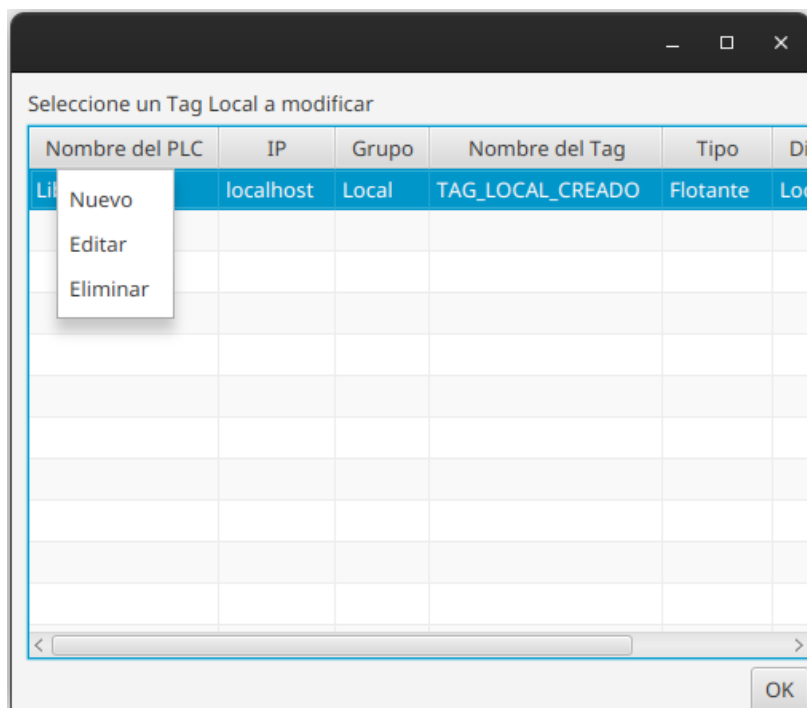


Figura 2.81 Ventana de Administración de tags Locales

Implementación de lógica para los tiempos de actualización de los tags

La aplicación requiere que el usuario pueda personalizar el tiempo de actualización de las distintas representaciones gráficas que se pueden enlazar a un tag, para ello en la ventana de asociación de tag se agregó un campo de texto para definir este tiempo de actualización en segundos, como se detalla en la Figura 2.82. A partir de ahora las representaciones gráficas se actualizarán dependiendo del valor de dicho campo.

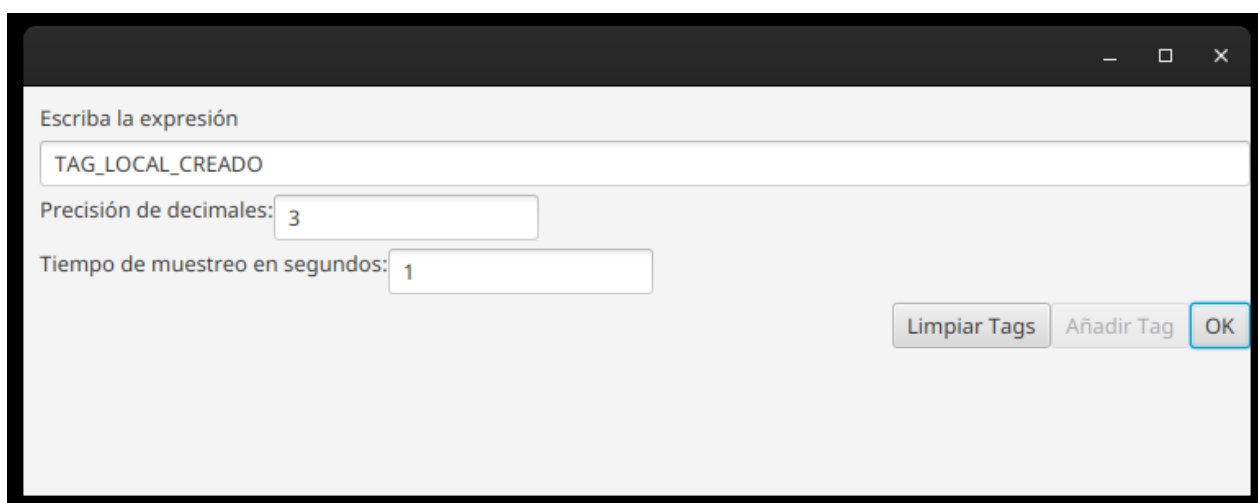


Figura 2.82 Ventana de asociación de tag con el campo de Tiempo de muestreo

2.7.8.4 Sprint Review

Tabla 2.19 Criterios de aceptación Sprint 7

ID	Descripción	Criterio de Aceptación	Cumple
US-04	Como Diseñador, Quiero Definir los tiempos de actualización del estado de las variables, Para poder diferenciar entre las representaciones que requieran más prioridad en su monitoreo.	La aplicación permite definir distintos tiempos de actualización del valor de un tag asociado a una representación gráfica.	SI
US-16	Como Diseñador, Quiero Definir contraseñas maestras en los HMI desarrollados, Para Poder Controlar quien puede ejecutar y editar los diseños	La aplicación permite encriptar un archivo de proyecto a través de una contraseña definida por el usuario desde la aplicación, de la misma forma se puede descifrar el archivo con la contraseña definida desde la aplicación.	SI
US-18	Como usuario quiero agregar un tag que se ejecuta localmente sin necesidad de la base de datos	La aplicación permite la creación de tags locales que se comportan de la misma forma que un tag leído desde la base de datos, para su creación y administración existen distintas ventanas disponibles para el usuario.	SI
US-23	Como usuario quiero poder exportar e importar solamente las ventanas hacia otro proyecto	La aplicación permite exportar una ventana en un archivo de texto plano o cifrado, así mismo se puede agregar dicha ventana exportada a un proyecto distinto.	SI

2.7.8.5 Sprint Retrospective

¿Qué salió bien en la iteración?

La aplicación ahora permite crear tags locales que son independientes de la base de datos del controlador. Además, se pueden exportar tanto proyectos enteros como ventanas únicamente, así como cifrarlos a través de una contraseña.

¿Qué se puede mejorar?

La aplicación debe impedir la eliminación de un tag local que este asociado a una representación gráfica, y mostrar al usuario las representaciones que se encuentren asociadas a dicho tag.

2.8. Retrospectiva

A lo largo del desarrollo de los sprints del proyecto se han identificado varios problemas que han podido solventarse para poder satisfacer las necesidades del usuario, en esta sección se resumirán estos problemas, además de comentar el porqué de la utilidad de la metodología SCRUM.

Luego de desarrollar el primer sprint, se presenta la necesidad de personalizar el tiempo de actualización de los tags y la disponibilidad de creación de tags locales, es entonces que se agregaron dos historias de usuario que implementaban esta funcionalidad y fueron realizadas en los sprint siguientes.

Mientras se desarrollaba el segundo sprint, se pensaba implementar la representación gráfica con animación de cambio de estado a través de un objeto de barra de progreso de la librería JavaFX, sin embargo, no cumplía con las características para mantener la coherencia interna con el sistema, es por eso que se implementó una animación a través de dos rectángulos que varían en base al tag asociado simulando una barra de progreso.

Después del desarrollo del tercer sprint, es evidente que se pueden agregar imágenes al lienzo, entonces es necesario que el usuario tenga la opción para agregar sus propias imágenes más allá de las predeterminadas, para eso se agregó una nueva tarea para la historia de usuario US-02 del siguiente sprint que realizaría una opción muy similar.

Cuando el cuarto sprint finalizó, se determinó que es necesario un modo de edición y de ejecución que permita al usuario administrar y personalizar las pantallas de trabajo que desarrollará, entonces se agregó una nueva historia de usuario para este propósito.

En el desarrollo del quinto sprint, se requería implementar una representación gráfica de resumen de alarmas, esta requería de consultar las alarmas y agregarlas para que el usuario pueda observarlas, para realizar esto se reutilizó la tabla de administración de alarmas donde se cambiaron las opciones al dar clic derecho y a través de un hilo propio hace las consultas sobre si una alarma se ha activado o desactivado.

A continuación, en el desarrollo del sexto sprint, mientras se implementaba el gráfico de tendencias utilizando un objeto de JavaFX este tenía sus opciones de personalización en cuanto a color y dinamismo muy limitadas es entonces que fue necesaria una sobreescritura de ciertos métodos del gráfico para permitir así mostrar u ocultar líneas graficadas en tiempo real, así como poder cambiar su color.

Finalmente, para poder realizar la implementación de los tag locales, se generó el problema de que estos serían siempre estáticos por lo que se implementó un hilo para que realice la

actualización de valores si estos se ven afectados por una representación de entrada de datos.

3. RESULTADOS Y DISCUSIÓN

3.1. LibreHMI Plataforma de software supervisorio

En este capítulo se detalla el producto final en funcionamiento, utilizando todas sus opciones disponibles, para ello se utilizará como base el proceso simulado en el Wonderware® InTouch® HMI de preparación y producción de chocolate de Ambato que se muestra en la Figura 3.1. y en la Figura 3.2.

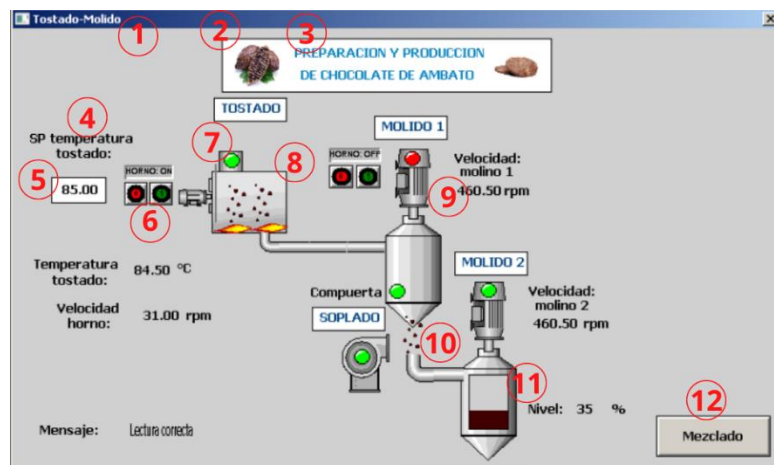


Figura 3.1 Pantalla de proceso de tostado y molido [25], con elementos numerados para su posterior comparación

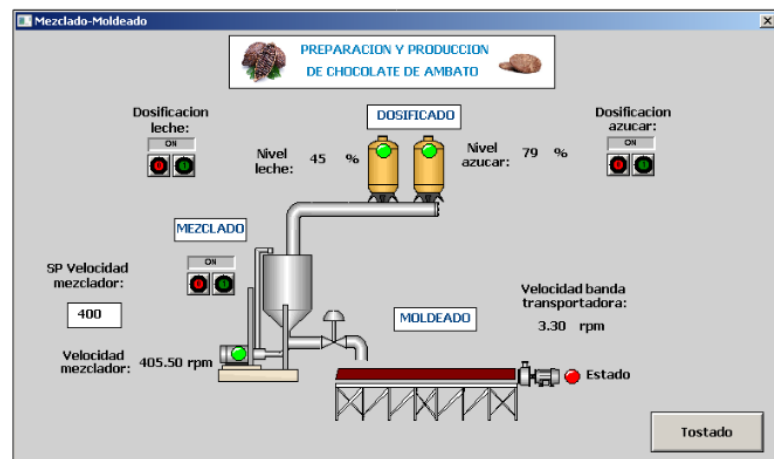


Figura 3.2 Pantalla de proceso de mezclado y moldeado [25]

Para replicar la simulación en la aplicación se seguirán los siguientes procedimientos que se mostrarán a continuación. En primer lugar, se iniciará la aplicación para mostrar la

ventana de la Figura 2.45, entonces se debe dar clic en el botón “Nuevo” y la aplicación solicitará que el usuario ingrese sus credenciales mostrando la ventana de la Figura 3.3.

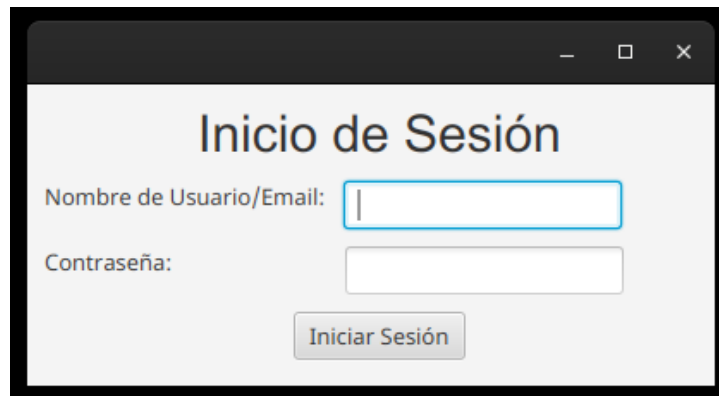


Figura 3.3 Ventana de inicio de sesión

Luego de iniciar sesión se mostrará la ventana de la Figura 2.59 pero con el lienzo en blanco, llegado a este punto se agregarán las representaciones gráficas análogas de la Figura 3.1, obteniendo un resultado como el que se mostrará a continuación en la Figura 3.4.



Figura 3.4 Pantalla de proceso de tostado y molido en LibreHMI, con elementos numerados para su posterior comparación

A continuación, se agregará una nueva página a través del proceso como en la Figura 2.57 y en la ventana de la Figura 2.27 se escribirá el nombre de la página “Mezclado-Moldeado” para después volver al proceso de agregar las representaciones gráficas análogas de la Figura 3.5.

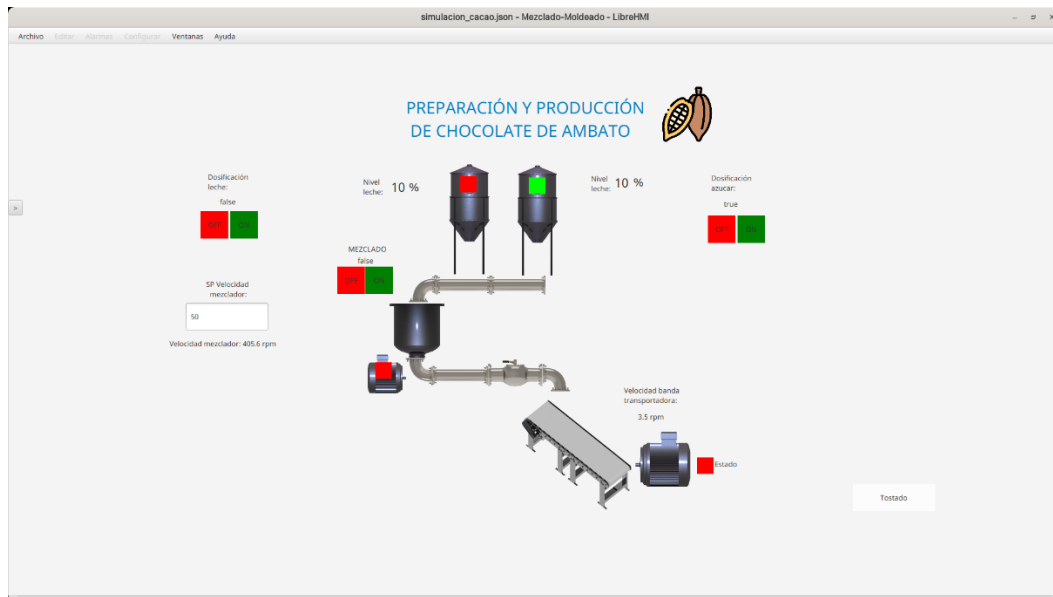


Figura 3.5 Pantalla de proceso de mezclado y moldeado en LibreHMI

Para terminar esta sección se muestra en la Tabla 3.1 una comparativa de funcionalidades entre el software privativo y el desarrollado en el trabajo de titulación, a través de los elementos señalados en los círculos numerados en la Figura 3.1 y Figura 3.4.

Tabla 3.1 Comparación de similitud de opciones entre una aplicación HMI profesional y la aplicación desarrollada.

Wonderware® InTouch® HMI	LibreHMI
1. Título de la página en la barra superior de la ventana.	1. Se muestra el título de la página en la barra superior de la ventana.
2. Imagen agregada desde el sistema de archivos.	2. Se puede agregar una imagen desde el sistema de archivos.
3. Etiquetas de texto con tamaño de texto grande y color azul.	3. y 4. La aplicación desarrollada también permite agregar etiquetas de texto y personalizarlo en cuanto a tamaño, color y fuente.
4. Etiquetas de texto con tamaño de texto mediano y color negro.	
5. Cuadro de entrada de texto, asociado a un tag.	5. Se permite agregar un cuadro de entrada de texto que se puede asociar a un tag.
6. Representación gráfica de switch, que envía un valor booleano al tag.	6. La aplicación no implementa la representación gráfica de switch, pero se puede simular su comportamiento a través de dos botones pulsadores.

7. Representación gráfica de luz, que cambia su valor de rojo a negro en función de una variable.	7. La aplicación desarrollada no implementa específicamente una representación gráfica de luz, sin embargo, el comportamiento puede simularse a través de un rectángulo con la animación de relleno asociada a una variable.
8. Representación gráfica de figura del área industrial.	8. Se implementa una galería de figuras del área industrial por defecto que se pueden agregar al lienzo.
9. Texto asociado a un valor desde un tag.	9. Se puede agregar texto que refleje el valor de un tag.
10. Animación de visibilidad en base a tag.	10. Se implementa la opción definir una animación de visibilidad, es decir, mostrar u ocultar una representación gráfica en base a un tag.
11. Cuadrado con opción de animación de relleno.	11. Existe la opción para definir una animación de relleno dentro de un cuadrado.
12. Botón para moverse entre las distintas páginas.	12. Se puede agregar un botón que permite navegar entre las distintas páginas.

Con lo expuesto en la Tabla 3.1 se puede comentar que las funcionalidades implementadas en el desarrollo de la aplicación del presente trabajo de titulación son similares por los puntos mostrados anteriormente como para recrear una simulación de un HMI profesional dentro de la misma.

3.2. PRUEBAS DE USABILIDAD

3.2.1 Usabilidad y Facilidad de Uso

La facilidad de uso suele ser confundida con el término usabilidad, a continuación, se presenta la diferencia entre ellos. Como se lee en [29] la usabilidad es esencial para la experiencia de usuario, es una medida de que tan bien un usuario dentro de un contexto puede lograr efectivamente un objetivo específico a través de la utilización de un producto. Por otro lado en [30] se dice que la facilidad de uso es una noción central para la usabilidad, describiendo que tan fácil se puede utilizar un producto, es decir, con que facilidad un usuario puede aprender a través de la relación de los elementos de la experiencia de usuario.

Finalmente, como se referencia en [31] una forma en la cual se puede medir de forma eficiente la usabilidad de un producto es a través de la Escala de Usabilidad del Sistema (SUS, por sus siglas en inglés) que es una herramienta utilizada para medir la facilidad de uso percibida aplicada a varios diseños y servicios digitales de experiencias de usuario permitiendo identificar problemas con el diseño. En síntesis la SUS calcula una medida de la usabilidad en general [31], entonces se detallará en la siguiente sección como se utiliza esta herramienta.

3.2.2 Escala de Usabilidad del Sistema

La SUS es una forma de medición de la usabilidad de un proyecto considerada confiable y rápida, está basada en un cuestionario de 10 preguntas cada una con cinco opciones de selección en un rango del 1 al 5, siendo el valor mayor estar completamente de acuerdo y el menor completamente desacuerdo, como respuesta por parte de los encuestados [32]. Las preguntas se detallan en la Tabla 3.2.

Tabla 3.2 Cuestionario de preguntas SUS

Nro.	Pregunta	VALORACIÓN				
		1 Completamente desacuerdo	2	3	4	5 Completamente de acuerdo
1	Creo que me gustaría usar este sistema con frecuencia.					
2	Encontré el sistema innecesariamente complejo.					
3	Pensé que el sistema era fácil de usar.					
4	Creo que necesitaría el apoyo de un técnico para					

	poder utilizar este sistema.					
5	Descubrí que las diversas funciones de este sistema estaban bien integradas.					
6	Pensé que había demasiada inconsistencia en este sistema.					
7	Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.					
8	Encontré el sistema muy difícil de usar.					
9	Me sentí muy seguro usando el sistema.					
10	Necesitaba aprender muchas cosas antes de poder ponerme en marcha con este sistema.					

Para la interpretación de las puntuaciones obtenidas luego de encuestar a los participantes se deben realizar los siguientes cálculos [33]:

- Para las preguntas impares, se restará uno de la respuesta [33].
- Para las preguntas pares, se restará la respuesta de 5 [33].
- Luego se sumarán los resultados de cada pregunta y se multiplicará por 2.5 [33].

Como se lee en [32], si se obtiene un resultado superior a 68 se considerará la usabilidad por encima del promedio, por otro lado cualquier valor menor estará por debajo del promedio.

3.2.3 Resultados de encuesta SUS

Para poder obtener un resultado que refleje la opinión del público al que se dirige este sistema, se ha encuestado a 9 personas del DACI y también se encuestó a la dueña del producto, los resultados y la tabla calculada se podrán observar a detalle en el Anexo A.

A continuación, se presenta en la Tabla 3.3, donde se han realizado los cálculos previamente definidos en la sección anterior en cada pregunta.

Tabla 3.3 Tabla de resultados SUS

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	RESULTADO SUS
Encuestado 1	3	2	4	4	4	1	2	2	3	4	72.5
Encuestado 2	2	3	3	3	3	3	3	4	3	2	72.5
Encuestado 3	4	3	3	3	2	4	2	3	4	4	80
Encuestado 4	2	4	3	3	4	3	3	3	3	4	80
Encuestado 5	3	3	4	4	3	3	4	3	3	3	82.5
Encuestado 6	3	4	4	3	2	3	4	4	4	4	87.5
Encuestado 7	4	3	3	2	3	4	3	4	3	3	80
Encuestado 8	3	4	4	3	3	4	4	3	4	4	90
Encuestado 9	3	3	2	4	4	4	3	4	2	2	77.5
Encuestado 10	4	3	3	2	3	4	3	3	3	3	77.5
Promedios	3.1	3.2	3.3	3.1	3.1	3.3	3.1	3.3	3.2	3.3	80

Después se procederá a mostrar en primer lugar un gráfico de resumen del puntaje promedio de las preguntas luego del cálculo SUS, este puede observarse en la Figura 3.6.

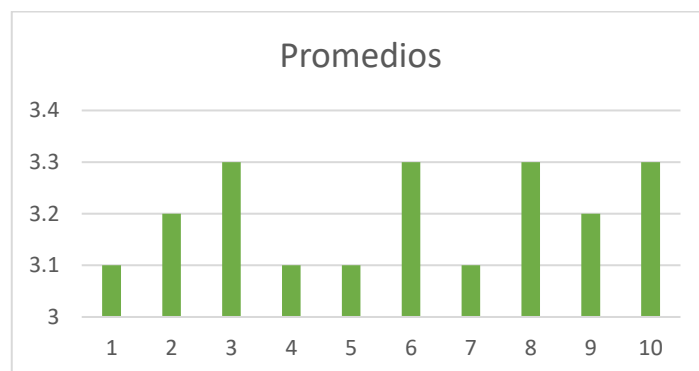


Figura 3.6 Gráfica de promedios de resultados SUS por cada pregunta

A continuación, se muestra el gráfico de resumen de los puntajes totales SUS de cada encuestado, como se ilustra en la Figura 3.7.



Figura 3.7 Gráfico de resumen de los resultados SUS de los encuestados

Finalmente, el puntaje final que se obtuvo de promedio de las encuestas SUS es de 80 puntos sobre 100, que como se detalló en la anterior sección es un resultado por encima del promedio, es decir, la satisfacción de usabilidad de los usuarios es alta.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se determinaron los requerimientos funcionales del sistema para la implementación de software supervisorio, a través de una serie de entrevistas a la dueña del producto que permitieron generar el backlog del proceso SCRUM, se llegó a la conclusión de organizar los requerimientos del software en módulos como el lienzo, tags, usuarios, exportación, alarmas, representaciones gráficas y páginas como se puede referenciar en la Tabla 2.2 estableciendo como alcance que el producto entregable del proyecto sea una aplicación de escritorio donde se puedan agregar distintas representaciones gráficas con el fin de simular dinámicamente un proceso industrial.
- El software supervisorio desarrollado permite a los usuarios implementar pantallas HMI siguiendo los estándares profesionales, a través de los distintos objetos que se añaden al lienzo de trabajo, tal y como se pudo detallar en la Tabla 3.1, donde

se demostró que al ser muy similar en opciones comparado el software privativo Wonderware® InTouch® HMI el usuario puede tener una experiencia satisfactoria.

- La arquitectura de la aplicación se diseñó y definió mediante un modelo de tres capas en primer lugar la capa de presentación que se encargará de manejar las distintas representaciones propias de un HMI, luego la capa de aplicación que contiene toda la lógica de objetos HMI y el nivel de datos donde se consultarán los distintos tags que se asocian a las representaciones. Estas características en conjunto permiten que la aplicación tenga funcionalidades propias de un software HMI profesional.
- La aplicación permite su interoperabilidad en distintos sistemas operativos gracias a que fue implementada en el lenguaje de programación Java, así como no depender de ninguna librería o framework que sea dependiente del sistema operativo, además gracias al controlador Ethernet/IP™, es posible conectarse a sistemas SCADA que utilicen este protocolo, de forma que se puede simular un ambiente de trabajo con varias instancias de la aplicación en distintos computadores para cada área de trabajo cada uno con los elementos de la vida real de sus respectivas áreas para poder monitorizar los diversos estados de dichos elementos.
- En el presente trabajo se aplicó la metodología SCRUM y gracias el refinamiento del *Product Backlog* permitió su flexibilidad para agregar nuevas historias de usuario mientras se iban realizando los sprint, es decir que varias historias no estuvieron escritas desde el principio, pero a lo largo del desarrollo se fueron agregando, este comportamiento dinámico permitió la eficiencia del proyecto en cuanto a su entendimiento por todos los miembros del *Scrum Team* sin necesidad de entender todo desde el principio del proyecto como se referencia en [34].
- Se ha comprobado la calidad del producto a través de las pruebas SUS, donde los encuestados respondieron una serie de preguntas acerca de cómo el software les parece fácil y útil de usar, obteniendo un resultado de 80 sobre 100 que significa que el software tiene una calificación por encima del promedio.
- La implementación de la animación de relleno porcentual se realizaría a través de asociar un tag con un objeto propio de JavaFX, una barra de progreso, sin embargo, este objeto era limitado en cuanto a las opciones y estilo requerido para mantener una coherencia interna con el sistema, entonces fue necesaria la implementación

manual de una barra de progreso superponiendo dos rectángulos y variando sus tamaños en base al valor de un tag.

- Los objetos de área industrial se representan a modo de imágenes predeterminadas dentro de la aplicación, sin embargo, para permitir que el usuario agregue nuevos objetos se habilita importar una imagen de origen externo que se copia a un directorio del sistema para que sea tomada en cuenta como parte del conjunto de imágenes predeterminadas.
- No existen explícitamente las figuras de círculo y de cuadrado, sin embargo, estas pueden añadirse a través de definir la elipse con dos radios iguales y el rectángulo con sus valores de ancho y alto iguales.
- Algunas figuras geométricas deben ser definidas por el usuario como es el caso del círculo y el cuadrado, que no existen explícitamente, sin embargo, pueden definirse a través de la elipse con dos radios iguales y el rectángulo con sus valores de ancho y alto iguales.
- Las opciones de gráfico de tendencias de JavaFX permitían una limitada personalización de color y comportamiento de las líneas representadas, entonces se requirió sobrescribir dicho comportamiento para permitir definir el color y también poder ocultar una línea de representación en tiempo real mientras se está ejecutando.

4.2. RECOMENDACIONES

- Agregar más representaciones gráficas que no han sido consideradas dentro del alcance de esta versión, como por ejemplo un checkbox asociado a un tag, un combobox que permita seleccionar distintos valores y asociarlos a un tag, además de otros elementos de entrada que son propios de software HMI.
- La aplicación puede integrar las alarmas a los objetos de forma que un objeto refleje también que una alarma se ha activado, más allá de tener el objeto de resumen de alarmas, además dicho resumen podría integrarse como una ventana independiente que este siempre accesible al usuario y sea independiente al lienzo.
- La aplicación puede agregar más opciones a la administración de alarmas como la opción de archivar, silenciar, etc. También pueden añadirse más objetos que notifiquen estados de un tag que no sean necesariamente alarmas, como por ejemplo eventos y notificaciones.

- La aplicación puede agregar más opciones para gestionar las representaciones gráficas dentro del lienzo, como por ejemplo una cuadrilla de fondo, alinear los objetos automáticamente, una opción gráfica que indique que dos objetos están alineados, la posición actual de un objeto y otras más que permitan al usuario una experiencia más cómoda al momento de diseñar HMIs.
- La aplicación puede integrar opciones de accesibilidad como permitir realizar zoom a los elementos dentro del lienzo para permitir que sea utilizada por más usuarios que pueden tener problemas de visión.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] “Ingeniería en Electrónica y Control - Facultad de Ingeniería Eléctrica y Electrónica - FIEE”. <https://fieee.epn.edu.ec/index.php/oferta-academica/pregrado/ingenieria-en-electronica-y-control> (consultado el 12 de diciembre de 2020).
- [2] “Wonderware California”. <https://california.wonderware.com/Licenses/LicenseInfo.aspx> (consultado el 12 de diciembre de 2020).
- [3] F. Adamo, F. Attivissimo, G. Cavone, y N. Giaquinto, “SCADA/HMI Systems in Advanced Educational Courses”, *IEEE Trans. Instrum. Meas.*, vol. 56, núm. 1, pp. 4–10, feb. 2007, doi: 10.1109/TIM.2006.887216.
- [4] C. Coro y C. Fabricio, “Importancia del uso del sistema scada para el desarrollo empresarial”, ago. 2019, Consultado: el 15 de marzo de 2021. [En línea]. Disponible en: <http://dspace.uniandes.edu.ec/bitstream/123456789/10386/1/PIUSDSIS024-2019.pdf>
- [5] J. Colomer, J. Ayza, y J. Meléndez, “Sistemas de Supervisión”. CEA-IFAC, 2000.
- [6] D. Pliatsios, P. Sarigiannidis, T. Lagkas, y A. G. Sarigiannidis, “A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics”, *IEEE Commun. Surv. Tutor.*, vol. 22, núm. 3, pp. 1942–1976, 2020, doi: 10.1109/COMST.2020.2987688.
- [7] V. Schiffer, “COMMON INDUSTRIAL PROTOCOL (CIPTM) AND THE FAMILY OF CIP NETWORKS”. ODVA, Inc, febrero de 2016. [En línea]. Disponible en: https://www.odva.org/wp-content/uploads/2020/06/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf
- [8] M. D. Sheet Khattab, “DESIGN AND IMPLEMENTATION OF USER INTERFACE; STRATEGIES FOR EFFECTIVE HUMAN COMPUTER INTERFACE”, University of Technology, Iraq - Baghdad, 2007. Consultado: el 15 de noviembre de 2021. [En línea]. Disponible en: <https://uotechnology.edu.iq/ce/english/thesis/thesisMsc2007/2007%20Muna%20Dhia%20Sheet%20Khattab.pdf>
- [9] Rockwell Automation, Ed., “Rockwell Automation - Process HMI Style Guide”. Rockwell Automation, mayo de 2019. Consultado: el 15 de noviembre de 2021. [En línea]. Disponible en: https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/proces-wp023_-en-p.pdf
- [10] “IntelliJ IDEA overview | IntelliJ IDEA”, IntelliJ IDEA Help. <https://www.jetbrains.com/help/idea/discover-intellij-idea.html> (consultado el 17 de diciembre de 2021).
- [11] Andrew Pomponio, “MySQL Overview: Key Features, Benefits, and Use Cases”, OpenLogic by Perforce. <https://www.openlogic.com/blog/mysql-overview> (consultado el 21 de diciembre de 2021).
- [12] “MySQL :: MySQL Connector/J 8.0 Developer Guide :: 1 Overview of MySQL Connector/J”. <https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-overview.html> (consultado el 17 de diciembre de 2021).
- [13] “1 JavaFX Overview (Release 8)”. <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm> (consultado el 17 de diciembre de 2021).
- [14] Atlassian, “Learn Git- Git tutorials, workflows and commands | Atlassian Git Tutorial”, Atlassian. <https://www.atlassian.com/git> (consultado el 18 de diciembre de 2021).

- [15] “What Is GitHub? A Beginner’s Introduction to GitHub”, Kinsta®. <https://kinsta.com/knowledgebase/what-is-github/> (consultado el 20 de diciembre de 2021).
- [16] “How to Prioritize Product Backlog Using MoSCoW Method”. <https://www.visual-paradigm.com/scrum/prioritize-backlog-with-moscow/> (consultado el 6 de enero de 2022).
- [17] “What is Scrum?”, Scrum.org. <https://www.scrum.org/resources/what-is-scrum> (consultado el 21 de diciembre de 2021).
- [18] “What is a Sprint in Scrum?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-sprint-in-scrum> (consultado el 21 de diciembre de 2021).
- [19] “What is Sprint Planning?”, Scrum.org. <https://www.scrum.org/resources/what-is-sprint-planning> (consultado el 21 de diciembre de 2021).
- [20] “What is a Daily Scrum?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-daily-scrum> (consultado el 3 de enero de 2022).
- [21] “What is a Sprint Review?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-sprint-review> (consultado el 3 de enero de 2022).
- [22] “What is a Sprint Retrospective?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-sprint-retrospective> (consultado el 4 de enero de 2022).
- [23] “What is a Product Backlog?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-product-backlog> (consultado el 4 de enero de 2022).
- [24] “What is a Sprint Backlog?”, Scrum.org. <https://www.scrum.org/resources/what-is-a-sprint-backlog> (consultado el 4 de enero de 2022).
- [25] A. Moya, “DESARROLLO DE UN DRIVER ETHERNET/IP PARA COMUNICACIÓN DE PLCs A INTERFACES DE OPERADOR”, Proyecto Integrador, Escuela Politécnica Nacional, Quito, 2022.
- [26] “What is Three-Tier Architecture”, el 16 de septiembre de 2021. <https://www.ibm.com/cloud/learn/three-tier-architecture> (consultado el 16 de junio de 2022).
- [27] “Design Pattern - Overview”. https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm (consultado el 16 de junio de 2022).
- [28] “What is MVC Design Pattern | How it Works | Skills | Scope | Advantages”, EDUCBA, el 26 de abril de 2019. <https://www.educba.com/what-is-mvc-design-pattern/> (consultado el 16 de junio de 2022).
- [29] “What is Usability?”, The Interaction Design Foundation. <https://www.interaction-design.org/literature/topics/usability> (consultado el 23 de junio de 2022).
- [30] “What is Ease of Use?”, The Interaction Design Foundation. <https://www.interaction-design.org/literature/topics/ease-of-use> (consultado el 23 de junio de 2022).
- [31] “The System Usability Scale & How it’s Used in UX | Adobe XD Ideas”, Ideas. <https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/> (consultado el 23 de junio de 2022).

- [32] A. S. for P. Affairs, "System Usability Scale (SUS)", el 6 de septiembre de 2013. <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (consultado el 16 de junio de 2022).
- [33] "System Usability Scale". <https://uxls.org/methods/system-usability-scale/> (consultado el 16 de junio de 2022).
- [34] "Product Backlog Refinement easily explained", Vollcom Digital, el 8 de marzo de 2022. <https://www.vollcom-digital.com/blog/agility-scrum/product-backlog-refinement-easily-explained/> (consultado el 18 de julio de 2022).

ANEXOS

ANEXO A. Resultados de pruebas de usabilidad y cálculo de resultados SUS

A través del siguiente enlace se puede acceder a los resultados de las pruebas:

https://docs.google.com/spreadsheets/d/1KhfxdjftkRIUnGLkcyHtanvhmcLx_JwUVjaBSsJYm1o/edit?usp=sharing

ANEXO B. Manual de usuario de aplicación LibreHMI

A través del siguiente enlace se puede acceder al manual de usuario de la aplicación

https://drive.google.com/file/d/1BDX_-xuXlnWJp7UZka-gH_t33fcv7nFu/view?usp=sharing

ANEXO C. Repositorio con código fuente de LibreHMI en GitHub

A través del siguiente enlace se podrá acceder al código fuente del proyecto.

<https://github.com/andr3slelouch/LibreHMI>

ORDEN DE EMPASTADO