Universidad de las Ciencias Informáticas

Facultad 5

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

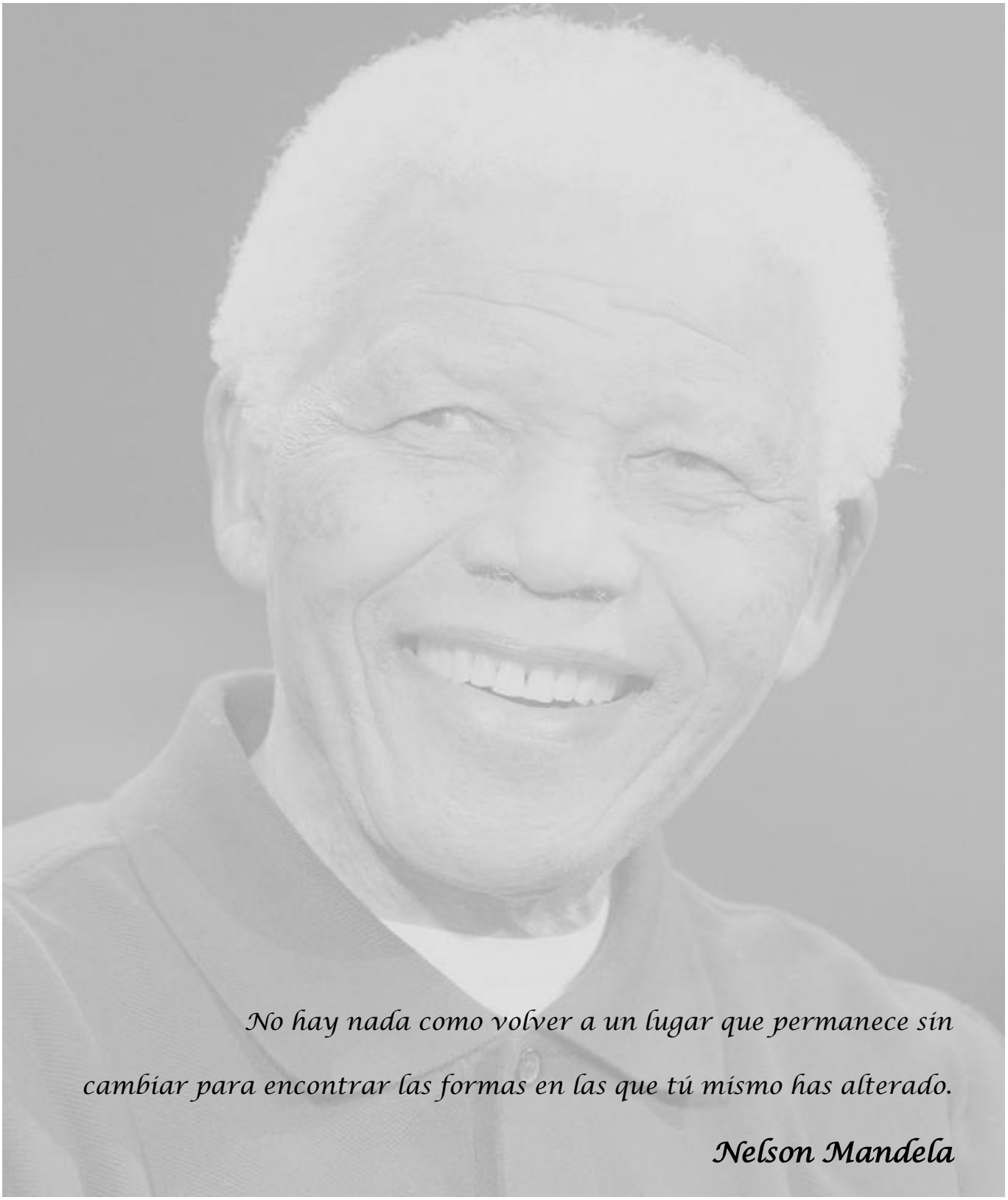
Desarrollo de paleta de componentes gráficos especializados del sector petrolero para el Sistema Integrado de Supervisión y Control de Procesos Guardián del ALBA

Autor: Julio López González

Tutor: Ing. Omar Martínez Díaz

"La Habana, 2015"

"Año 57 de la Revolución"



No hay nada como volver a un lugar que permanece sin cambiar para encontrar las formas en las que tú mismo has alterado.

Nelson Mandela

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Julio López González	**Ing. Omar Martínez Díaz**
_____	_____
Firma del autor	Firma del tutor

DATOS DE CONTACTO

Nombre y apellidos del tutor: Omar Martínez Díaz.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Correo electrónico: omarmd@uci.cu

Ingeniero en Ciencias Informáticas y especialista graduado en la UCI en el año 2010. Actualmente se desempeña en el Departamento de Desarrollo de Componentes del Centro de Informática Industrial (CEDIN).

AGRADECIMIENTOS

A todas aquellas personas que durante la realización de este trabajo me han apoyado de una forma u otra cuando lo he necesitado.

En especial a mis padres por darme fuerzas para seguir adelante y porque son los que me han convertido en lo que soy hoy.

A mi hermanita linda por ser tan especial y por darme a mis dos sobrinas que son como el Jing y el Jang pero las quiero con la vida.

A mis abuelos: tito, juanita y cucho.

A mi tios Miriam, René, Susana, Hugo y Baudilio.

A mi prima miri por ser un ejemplo a seguir y por darme una primita tan linda y tan sata.

En general a toda mi familia por su apoyo y amor incondicional.

A mis compañeros de estudio y en especial a un piquete llamado ¨los 300¨, nunca olvidaré todas las locuras, fiestas y actividades que compartimos, me siento orgulloso de haber estado en ese piquete.

A mi loca y pinareña novia Lazara por ser lo mejor que me ha pasado en la universidad, por cuidarme y engordarme en todo este tiempo, por aguantar mis pesadeces y malacrianzas, pero sobre todo por hacerme sentir y no es solamente la frase, el hombre más feliz de la tierra.

A mis amistades, que han compartido conmigo esta etapa tan linda de la vida. Quisiera mencionarlos a todos pero no me alcanzan las 80 hojas del documento, A Liuver y al negro por ayudarme en disímiles tareas durante estos cinco años, sin ellos hubiera pasado un poco más de trabajo.

A Sergio y Edward por ser como hermanos para mí,

Agradecimientos

Al rojo, tony, emoilio, reitel, el chino, el pushi, dayana,maiyara, rosmery, esmerida, alfonso, emilito, carlos, nayrobis, luis carlos, cajigal, sony, orlando y a todos los del grupo en general.

A lupita por hacer tantos dulces ricos y por ser tan buena amiga.

A ofelia, elizabeth, dayán, dania, dago, kirenia, jose, aray, midiala , juan carlos, eloy, costa, el pipi, yader, en fin a todos los que de una forma u otra han compartido un rato agradable conmigo en esta universidad,

A la gente de mi cuadra en holguin karina, thalia, ale, denis, juan luis, gabi, coco, chachi, kevin, jose, idania, gracias a todos por compartir tantos momentos inolvidables y los que faltan por compartir.

A mi tutor Omar, por su constante dedicación y ayuda en todo momento;

A todos los profesores que a lo largo de la carrera han contribuido en mi formación.

Y a nuestro Comandante en Jefe,

por concebir,

entre sus grandes proyectos e ideas,

este maravilloso sueño.

A todos, Muchas Gracias.

DEDICATORIA

Dedico el presente trabajo de diploma a mi familia, a mis padres Teresa y Jacinto,
por ser los mejores padres del mundo y por apoyarme en todo momento.
A mi tía fefa por ser como una madre para mí,
por preocuparse y quererme de la forma en que lo hace.
A mi abuela Zaida, que aunque no se encuentra entre
nosotros, siempre la tengo presente y la recordaré
por ser la persona más cariñosa que he conocido.
A toda mi familia y amigos más cercanos con los que sé
que puedo contar en todo momento.
A mi novia Lazara por quererme de la forma en que lo hace y por hacerme
pasar los mejores momentos de la universidad.
A mi hermana tere por ser la mejor hermana del mundo.
A todas esas personas, que a pesar de la distancia,
seguirán ocupando un lugar especial en
mi corazón.

RESUMEN

Los sistemas SCADA, acrónimo de Supervisory Control and Data Acquisition (Control, Supervisión y Adquisición de Datos) se utilizan principalmente en el entorno industrial. Una de las principales prestaciones dentro del módulo HMI es el ambiente de configuración. Este cuenta con una biblioteca gráfica donde se encuentran los diferentes componentes gráficos encargados de mostrar los valores recogidos del campo mediante los dispositivos, los cuales son utilizados para realizar los diferentes despliegues operacionales.

Este trabajo tiene como objetivo desarrollar componentes gráficos con diferentes funcionalidades, que puedan ser configurados según el propósito del operador y modificadas sus propiedades en el ambiente de configuración. El desarrollo de los componentes está sustentado por la metodología AUP, utilizando las técnicas de modelación establecidas por el Lenguaje Unificado de Modelado (UML) y el potente marco de desarrollo de interfaces gráficas de usuario Qt framework. Se aplicó el método de prueba de caja negra el cual permitió validar que se cumplieran los requerimientos funcionales establecidos.

PALABRAS CLAVE

SCADA, HMI, componentes gráficos, biblioteca gráfica, ambiente de configuración.

Índice de Contenidos

Índice de figuras

Índice de tablas

INTRODUCCIÓN

Los procesos industriales de la actualidad ocupan el mayor porcentaje de los desarrollos de aplicaciones de una empresa. Se observa un creciente interés en todas las áreas donde es crucial el control de las actividades, la supervisión en tiempo real, el registro de los eventos, la emisión de reportes y la preparación de los análisis dirigidos a la toma de decisiones y a la gerencia de una empresa exitosa. Por estas razones, hoy en día las empresas han aumentado el interés en emplear los sistemas de Supervisión, Control y Adquisición de Datos (SCADA por sus siglas en inglés).

Actualmente, los SCADA son parte estratégica de cualquier industria, bien sea eléctrica, petrolera, gasífera, aluminio, etc. Los sistemas SCADA originalmente se diseñaron para cubrir las necesidades de un sistema de control centralizado, sobre procesos o complejos industriales distribuidos sobre áreas geográficas muy extensas.

Un SCADA es un sistema que se utiliza para la supervisión, control y adquisición de datos del proceso de producción. Estos sistemas mejoran la eficacia del proceso de monitoreo y control proporcionando la información oportuna para poder tomar decisiones operacionales apropiadas (1). Además están compuestos por un software y por diferentes hardware de señales de entrada y salida, pantallas interfaces entre el hombre y las máquinas, bases de datos, redes, comunicaciones y controladores. Dentro de su estructura encontramos el módulo HMI, Human Machine Interface (HMI) o Interfaz Hombre-Máquina, que permite la interfaz entre la persona (operador) y la máquina (proceso) (2).

En el Centro de Informática Industrial (CEDIN) de la Facultad 5 se lleva a cabo el Proyecto SCADA Guardián del ALBA (GALBA) enfocada al sector petrolero. Se desarrolla en acuerdo con PDVSA, de la hermana República Bolivariana de Venezuela conocido en sus inicios como SCADA Nacional o SCADA PDVSA y a partir del 2008, en que fue presentado en la “Cumbre del ALBA”, se comienza a conocer como SCADA GALBA, donde se proyectó una futura instalación del sistema en los países integrantes de esta organización. Actualmente el módulo HMI en la versión Zamora R1 del SCADA Guardián del ALBA cuenta, entre sus principales elementos, con un editor de configuración. Dicho editor es el encargado de brindar la funcionalidad de configurar de manera sinóptica todo el proceso industrial, haciendo uso de una paleta de componentes agrupados en básicos como líneas, rectángulos, polígonos y similares. También cuenta con gráficos de control como son botones, deslizadores y otros afines a esta función, que unidos a otros componentes visualizadores encargados de mostrar los valores recogidos del campo mediante los

dispositivos, le dan la posibilidad al operador de crear los despliegues del proyecto. Estos despliegues son utilizados para guardar las configuraciones de los componentes asociados con los puntos, animaciones y modificaciones realizadas según el propósito del operador, permitiendo visualizarse en el ambiente de ejecución para supervisar y controlar los procesos industriales.

En el sector petrolero existen un conjunto de gráficos que se utilizan frecuentemente en la mayoría de los despliegues, ellos son: válvulas, enfriadores de crudo y gas, bombas y camiones cisternas. Si bien con la combinación de componentes gráficos existentes en el ambiente de configuración se pueden obtener representaciones que suplan a estos últimos, resulta costoso en tiempo y esfuerzo la elaboración de estas combinaciones. La creación de cada despliegue es un proceso que se complejiza según la cantidad de variables y componentes gráficos que tenga involucrado. Esto, sumada a la labor de ensamblar los componentes gráficos, hace que el tiempo de creación de dichos componentes se vea notoriamente penalizado.

A partir de la **situación problemática** planteada anteriormente, se propone como **problema de investigación** el siguiente: ¿Cómo disminuir el tiempo de creación de los componentes gráficos y enriquecer la paleta de componentes del ambiente de configuración del SCADA Guardián del ALBA?

Para lo cual el **objeto de estudio** lo constituye: los componentes gráficos de los sistemas SCADA y se precisa como **campo de acción**: los componentes gráficos para la visualización de procesos industriales en el Sistema Integrado de Supervisión y Control de Procesos Guardián del ALBA.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar una paleta de componentes gráficos especializados del sector petrolero para el Sistema Integrado de Supervisión y Control de Procesos Guardián del ALBA.

Este objetivo general se desglosa en los siguientes **objetivos específicos**:

- ✓ Realizar el marco teórico de la investigación.
- ✓ Definir las funcionalidades que deben tener los componentes.
- ✓ Diseñar los componentes que brinden solución al problema planteado.
- ✓ Implementar los componentes diseñados.
- ✓ Validar el correcto funcionamiento de los componentes desarrollados.

Para dar cumplimiento a los objetivos se realizaron esencialmente las siguientes **tareas**:

- ✓ Revisión bibliográfica para generar el marco teórico conceptual de la investigación que represente las tendencias actuales en la implementación de componentes gráficos de los sistemas SCADA.
- ✓ Estudio de las especificaciones de requisitos de los componentes gráficos a desarrollar para sentar las bases teóricas de la investigación.
- ✓ Análisis de las tecnologías y herramientas existentes para el desarrollo de componentes gráficos de las que se usarán en la implementación del sistema.
- ✓ Definición de la arquitectura del sistema a desarrollar y selección de la metodología de desarrollo.
- ✓ Realización del diseño para facilitar la implementación de la paleta de componentes.
- ✓ Implementación de los componentes gráficos como propuesta de solución para la visualización de los procesos industriales.
- ✓ Integración con el módulo HMI para probar las funcionalidades de los componentes.
- ✓ Validación mediante pruebas funcionales de los componentes gráficos implementados para lograr un correcto funcionamiento del sistema.

Durante el desarrollo de esta investigación se hizo necesario profundizar en el estudio de los temas abordados, por lo que se usaron varios **métodos teóricos** y **métodos empíricos** los cuales se describen a continuación:

Los **Métodos Teóricos** son aquellos que posibilitan profundizar en las relaciones esenciales y cualidades fundamentales de los procesos no observables directamente, son fundamentales para la comprensión de los hechos y para la formulación de la hipótesis de investigación:

- ✓ **Analítico-Sintético:** Se utiliza para realizar un análisis de la documentación referente a las tecnologías y herramientas utilizadas en la implementación de componentes gráficos para sistemas SCADA.
- ✓ **Análisis Histórico-Lógico:** Se utiliza para la comprensión de las tendencias actuales referidas a la evolución de los mecanismos de implementación de componentes gráficos en el módulo HMI de un SCADA.

Los **Métodos Empíricos** por otra parte le permiten al investigador, la recopilación de datos reales acerca del comportamiento de los hechos, fenómenos, objetos y procesos de la naturaleza y de la sociedad.

- ✓ **Consultas de fuentes de información:** Se utiliza para las consultas de fuentes bibliográficas durante el desarrollo de la investigación.

El documento está estructurado de la siguiente manera: resumen, introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se brindan las definiciones y conceptos que ayudan en el desarrollo y comprensión de los componentes gráficos. Además, se describen las principales funciones del SCADA, y los módulos que lo estructuran, haciendo énfasis en la visualización de los componentes. Se realiza un análisis de las herramientas y tecnologías utilizadas en el desarrollo y documentación de la aplicación, así como las bibliotecas gráficas y lenguajes de programación estudiados en función de la elaboración de la paleta de componentes gráficos.

1.2 Sistemas de Supervisión, Control y Adquisición de Datos (SCADA)

1.2.1 Definición de SCADA

SCADA proviene de las siglas Supervisory Control and Data Acquisition (Supervisión, Control y Adquisición de Datos). Los sistemas SCADA son aplicaciones de software diseñadas con la finalidad de controlar y supervisar procesos a distancia. Se basan en la adquisición de datos de procesos remotos. Este tipo de sistema es diseñado para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde una computadora. Además, envía la información generada en el proceso productivo a diversos usuarios, tanto del mismo nivel como hacia otros supervisores dentro de la empresa, es decir, que permite la participación de otras áreas, como por ejemplo: control de calidad, supervisión, mantenimiento, etc. (3).

1.2.2 Requisitos básicos de los sistemas SCADA

- ✓ Todo sistema debe tener arquitectura abierta, es decir, debe permitir su crecimiento y expansión, así como, deben poder adecuarse a las necesidades futuras del proceso y de la planta.
- ✓ La programación e instalación no debe presentar mayor dificultad, debe contar con interfaces gráficas que muestren un esquema básico y real del proceso.
- ✓ Deben permitir la adquisición de datos de todo equipo, así como la comunicación a nivel interno y externo (redes locales y de gestión).
- ✓ Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware, y fáciles de utilizar, con interfaces amigables para el usuario (3).

1.2.3 Funciones principales

Dentro de las funciones principales realizadas por el sistema SCADA están las siguientes:

- ✓ **Supervisión:** El operador podrá observar desde el monitor la evolución de las variables de control, como cambios que se produzcan en la operación diaria de la planta, lo que permite dirigir las tareas de mantenimiento y estadística de fallas.
- ✓ **Control:** Mediante el sistema se puede activar o desactivar los equipos remotamente (por ejemplo abrir válvulas, activar interruptores, prender motores, etc.), de manera automática y también manual. El operador puede ejecutar acciones de control y podrá modificar la evolución del proceso en situaciones irregulares que se generen.
- ✓ **Adquisición de datos:** Recolectar, procesar, almacenar y mostrar la información recibida en forma continua desde los equipos de campo.

1.2.4 Módulos de un sistema SCADA

Los módulos o bloques software que permiten las actividades de adquisición, supervisión y control son los siguientes:

- ✓ **Configuración:** permite al usuario definir el entorno de trabajo de su SCADA, adaptándolo a la aplicación particular que se desea desarrollar.
- ✓ **Interfaz Hombre-Máquina:** proporciona al operador las funciones de control y supervisión de una planta. El proceso se representa mediante sinópticos gráficos almacenados en el ordenador de proceso y generados desde el editor incorporado en el SCADA o importados desde otra aplicación durante la configuración del paquete.
- ✓ **Módulo de proceso:** ejecuta las acciones de mando pre-programadas a partir de los valores actuales de variables leídas.
- ✓ **Gestión y archivo de datos:** se encarga del almacenamiento y procesado ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.
- ✓ **Comunicaciones:** se encarga de la transferencia de información entre la planta y la arquitectura hardware que soporta el SCADA, y también entre ésta y el resto de elementos informáticos de gestión (4).

1.3 Módulo Interfaz Hombre-Máquina (HMI)

Una interfaz Hombre-Máquina o HMI, Human Machine Interface, por sus siglas en inglés, es un sistema que presenta datos a un operador y a través del cual éste controla un determinado proceso (5). Permite

que el operador, en ciertas circunstancias, vaya más allá del manejo de la máquina y observe el estado del equipo e intervenga en el proceso. La información se proporciona por medio de paneles de control con señales luminosas, campos de visualización o botones, o por medio de software que utiliza un sistema de visualización que se ejecuta en una terminal (6).

1.3.1 Interfaz Hombre Máquina en sistemas SCADA

Entre los sistemas SCADA que se estudiaron se encuentran Movicon[1] y WinCC[2]. Estos sistemas aunque fueron desarrollados por empresas diferentes, sus HMI comparten elementos comunes. Cada uno provee herramientas que permiten diseñar y configurar los procesos que se desean supervisar y controlar, entre ellas están las barras de herramientas para la creación y modificación de objetos gráficos, paleta o biblioteca de componentes que contienen un grupo de objetos ya predefinidos y configurados.

1.3.1.1 HMI Movicon

La plataforma SCADA/HMI Movicon™11 tiene un ambiente de desarrollo para crear aplicaciones SCADA, HMI y análisis estadístico de los datos de producción, permitiendo así a los proyectistas poder reducir drásticamente los tiempos de desarrollo, y a los usuarios disponer soluciones potentes y flexibles (7).

Las características principales de Movicon son:

- ✓ Editor Gráfico Vectorial y tecnología basada en Gráficos Vectoriales Escalables (SVG en sus siglas en inglés).
- ✓ Programación realmente a objetos y símbolos, así como a objetos complejos.
- ✓ Visualización en árbol de todos los sinópticos, con selección intuitiva y configuración de propiedades de cada elemento individual, incluso elementos complejos, sin necesidad de descomponer y recomponer.

[1] Movicon es uno de los sistemas líderes a nivel mundial utilizado para el control, supervisión y adquisición de datos del proceso de producción, desarrollado por Progea que es la empresa líder en soluciones software Scada/HMI. Está presente desde 1990 en el mercado del software para la automatización de procesos industriales.

[2] WinCC es uno de los sistemas líderes a nivel mundial utilizado para el control, supervisión y adquisición de datos del proceso de producción, desarrollado por la empresa Siemens en 1996.

- ✓ Potente animación gráfica, con 16 funciones dinámicas de animación configurables en las propiedades de los objetos, además de la extensión mediante las correspondientes API gráficas VBA para cada objeto.
- ✓ Barra de herramienta con diferentes objetos gráficos funcionales (botones, medidores, etc.).
- ✓ Objetos gráficos que pueden ser inspeccionados en su estructura XML tanto internamente como por programas y editores externos. Los objetos complejos pueden así ser editados en las propiedades también en modo “textual” para acelerar las operaciones de “copiar-pegar” o “buscar-reemplazar”.
- ✓ Biblioteca de visualizadores analógicos integrada con apariencia real (7).

Figura 1: Editor gráfico del SCADA Movicon

Este HMI provee un ambiente en el cual pueden realizarse la configuración del proyecto y los diseños de despliegues haciendo uso de componentes gráficos o bibliotecas de símbolos, los cuales son generalmente agrupados por conceptos de negocios asociados a los procesos industriales de gas y petróleo, domótica y otros. Movicon brinda varias facilidades para la edición de propiedades gráficas de los objetos en cuanto a color, tipos de bordes, utilización de gradientes, así como configuración de animaciones, gestión de comandos y eventos.

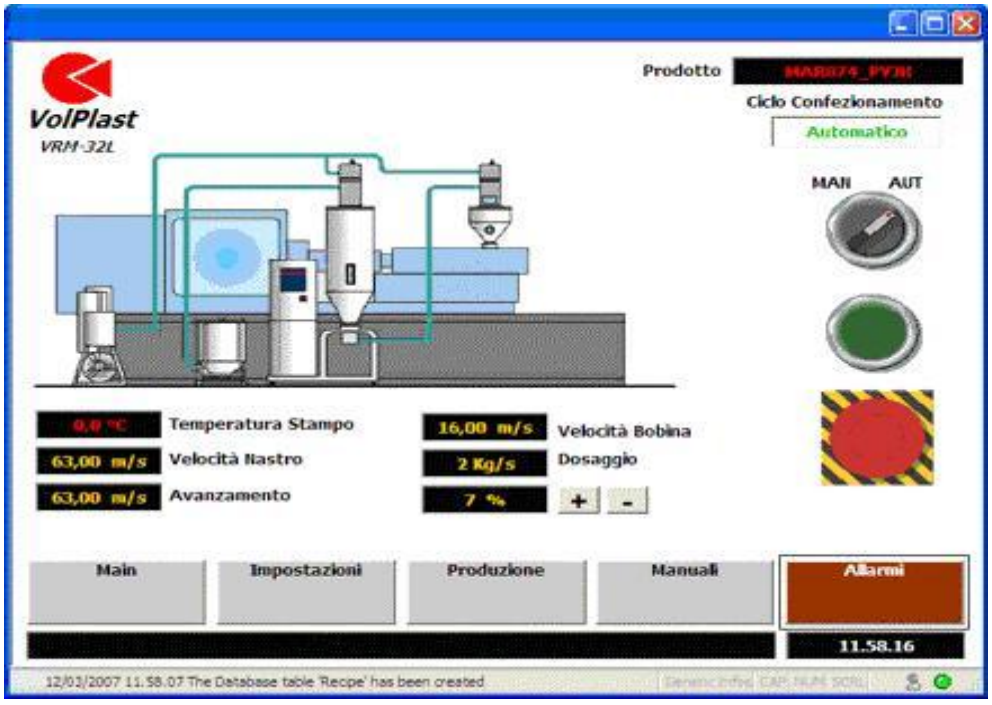


Figura 2: Ejemplo de gráficos disponibles

1.3.1.2 HMI WinCC

Este HMI permite la supervisión, adquisición y tratamientos de datos que provienen de un proceso. Al igual que Movicon brinda las herramientas necesarias para el diseño de despliegues como barra de herramientas, área de edición, paleta de estilos, controles y una paleta de objetos gráficos que está compuesta por objetos estándar (línea, polígono, línea poligonal, elipse, círculo, segmento elíptico, segmento circular, arco, rectángulo, texto estático, conector), objetos inteligentes (campos de entrada/salida) y objetos de ventana (botones, casillas de verificación). Entre las funciones que cuenta este HMI están:

- ✓ Visualización totalmente gráfica del desarrollo y el estado de los procesos.
- ✓ Manejo de la máquina o instalación desde una interfaz de usuario que se puede personalizar, con menús y barras de herramientas propios.
- ✓ Señalización y confirmación de eventos.
- ✓ Archivo de valores medidos y avisos en una base de datos del proceso.
- ✓ Protocolización de los datos actuales del proceso y de datos de archivo registrados.
- ✓ Administración de usuarios y sus permisos de acceso (8).

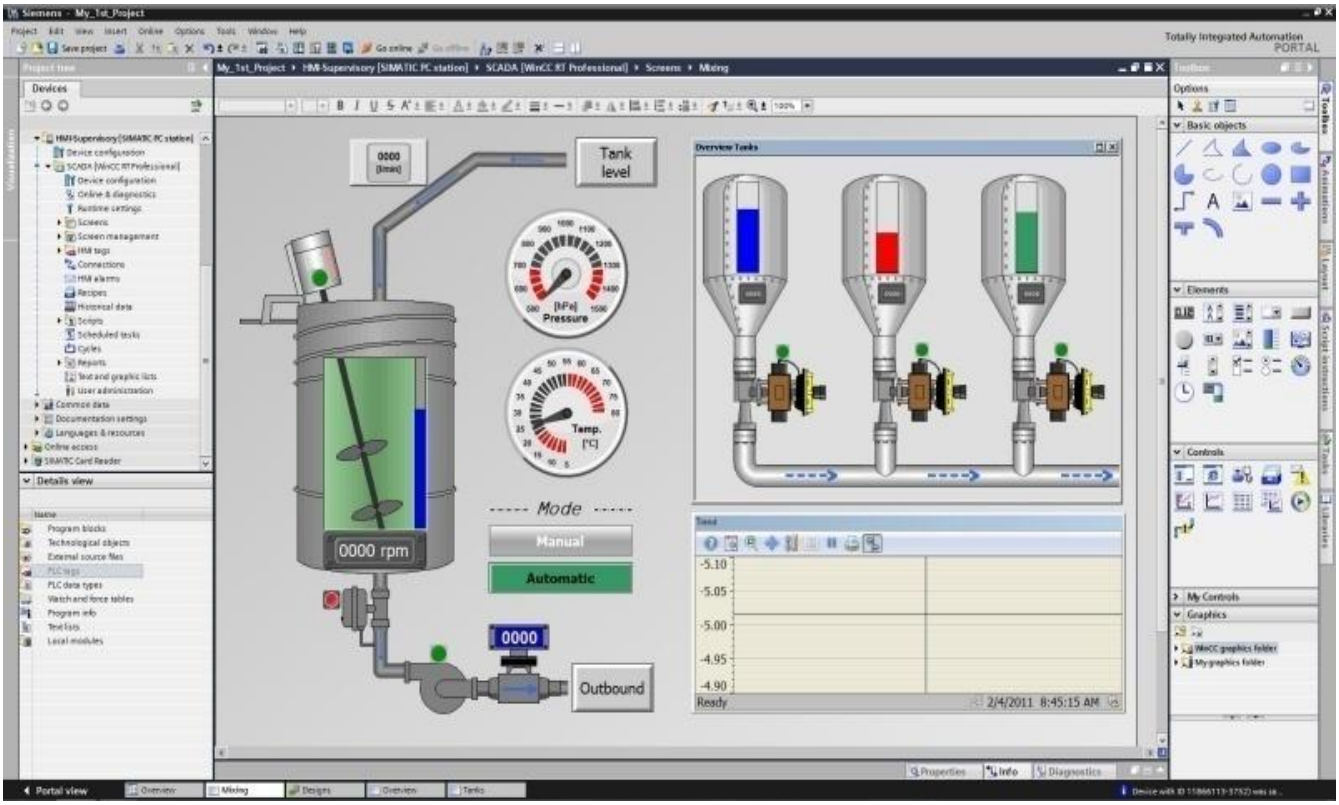


Figura 3: Editor gráfico del SCADA WinCC

Dichos objetos presentan propiedades de fuente para los textos, tamaño de caracteres, grosor de línea, ajustes del borde, color de fondo y color de fuente, así como propiedades de geometría (posición, ancho, alto). Permite además agrupar varios objetos en un grupo, brindando propiedades comunes de relleno, estilos e intermitencia para estos grupos. Los mismos pueden ser activados y desactivados con animaciones de colores y pueden ser seleccionados de manera dinámica para facilitar el control en el ambiente de ejecución. Este HMI también maneja la edición de alarmas que permite tener los datos para el bloque de sistema como fecha, hora y número de alarma programada; además permite los datos para el bloque de texto, que funciona como aviso indicando la razón de la ocurrencia de la alarma.

1.4 SCADA Guardián del ALBA

El SCADA GALBA es un sistema de supervisión, control y adquisición de datos, desarrollado por la UCI en el CEDIN para la Empresa Socialista Mixta de Venezuela que surgió con el objetivo primordial de lograr el desarrollo e independencia tecnológica de PDVSA y dar solución a todas sus necesidades en las materias de automatización informática, y telecomunicaciones. Este SCADA se ha adecuado a las necesidades de otros sistemas en la rama de la automatización, como son el sistema SCADA Eléctrico y el SCADA

SAINUX (Sistema SCADA para empresas cubanas), además está compuesto por un conjunto de módulos que funcionan basados en una arquitectura distribuida:

- ✓ **Bases de Datos Históricos:** Es el encargado de almacenar la información recibida desde el campo, así como la sucesión de alarmas y eventos. Esta información es de vital importancia para realizar cualquier tipo de análisis posteriores como diagnósticos o reportes. Debe permitir a los operarios y ejecutivos el análisis rápido de información histórica, la que debe estar organizada según patrones de comportamiento de las plantas y dar datos relevantes de todo el proceso industrial.
- ✓ **Comunicaciones o Middleware:** Este módulo representa la capa de software encargada de la comunicación entre los diferentes procesos distribuidos, de mediano y alto nivel. Enlaza todos los componentes del sistema y tiene relación directa con el canal de comunicación. Su arquitectura es compleja en la mayoría de los casos, pero presenta pequeñas interfaces que garantizan rapidez y robustez en la transmisión de mensajes. Es uno de los módulos horizontales al sistema y sus cambios suelen afectar el funcionamiento general del mismo.
- ✓ **Seguridad:** Permite a los usuarios autenticarse en el sistema, y de esta forma acceder sólo a los recursos que tiene asignado. Posee herramientas para la protección ante ataques piratas, fallos eléctricos, problemas de red, entre otros. Su objetivo es garantizar la accesibilidad, integridad y confidencialidad de la información manejada por el sistema.
- ✓ **Adquisición de Datos:** Supervisa y recoge la información del resto de las subestaciones, bien sean otros ordenadores conectados (en sistemas complejos) a los instrumentos de campo o directamente sobre dichos instrumentos, algunas de las funciones que cumple son:
    - Interrogar de forma periódica a la unidad de terminal remota (RTU), y transmitirle consignas; siguiendo usualmente un esquema maestro-esclavo.
    - Actuar como interfaz al operador, incluyendo la presentación de información de variables lo más cercano al tiempo real, la administración de alarmas y la recolección.
- ✓ **Interfaz Hombre-Máquina o HMI:** Interfaz que comunica al sistema con sus operadores, está compuesto por dos ambientes, el de configuración que permite administrar la configuración de varios procesos con sus respectivos despliegues diseñados, y el de ejecución que permite ejercer control, supervisar y visualizar los despliegues configurados.

1.4.1 Interfaz Hombre Máquina del SCADA GALBA

El módulo HMI del SCADA GALBA se encuentra dividido en dos ambientes, el ambiente de configuración y ambiente de ejecución.

1.4.1.1 Ambiente de configuración

Este ambiente permite configurar varios procesos o partes de ellos, posibilitando al usuario definir el entorno de trabajo de su aplicación, según la disposición de pantallas requeridas y los niveles de acceso para los distintos usuarios. Dentro del ambiente de configuración el usuario define las pantallas gráficas o de texto que va a utilizar, se configuran los diversos canales de comunicación que permitirán el enlace con los elementos de campo, entre otros elementos importantes en la configuración del proceso. Estas configuraciones son las que más tarde utilizará el visualizador para poner en ejecución la representación gráfica de la planta.

Para el diseño de los despliegues son utilizados los componentes gráficos, que son la representación de los instrumentos de campo utilizados en el control y supervisión de los procesos industriales. Los componentes gráficos también son llamados widgets u objetos. En programación, un widget es un elemento de una interfaz (interfaz gráfica de usuario o GUI) que muestra información con la cual el usuario puede interactuar. Por ejemplo: ventanas, cajas de texto, checkboxs, listbox, entre otros. La característica definitoria de un widget es proporcionar un punto de interacción con el usuario para la manipulación directa de un tipo de dato dado. En otras palabras, los widgets son bloques básicos y visuales de construcción que, combinados en una aplicación, permiten controlar los datos y la interacción con los mismos.

Actualmente en el ambiente de configuración existen un conjunto de componentes gráficos que se utilizan frecuentemente en la mayoría de los despliegues, ellos son: visualizador, medidor vertical, slider, gráfico de barra, rectángulo, imagen, etc. Si bien con la combinación de componentes gráficos existentes se pueden obtener representaciones que suplan a otros componentes que no están implementados como: válvulas, enfriador de crudo y gas, bomba y camión cisterna, resulta costoso en tiempo y esfuerzo la elaboración de estas combinaciones. Por lo que se hace necesario implementar dichos componentes para disminuir su tiempo de creación y enriquecer la paleta de componentes gráficos del ambiente de configuración.

Variables

El flujo principal de información en los sistemas SCADA lo constituyen las variables (puntos). Estas variables pueden representar innumerables indicadores como son: presión, temperatura, flujo, potencia,

peso, intensidad de corriente, voltaje, potencial hidrógeno, densidad, carga, resistencia o capacitancia entre otros. Las variables son adquiridas mediante instrumentación utilizando sensores conectados a autómatas y equipos de control. Después de convertidas a señales eléctricas, estas variables pasan a ser estructuras que contienen datos, los que pueden ser de tipos simples (entero, flotante, cadenas, y otros) o de tipos complejos. La información de estas variables permite conocer el estado del sistema y su historia (9).
Las variables contienen un conjunto de atributos, generalmente se destacan:

- ✓ **Nombre y/o Identificador:** Es el atributo que identifica y hace única a la variable en el sistema.
- ✓ **Valor:** Es el atributo que contiene el estado (valor) de la variable en un instante de tiempo.
- ✓ **Marca de tiempo:** Representa el instante de tiempo en que se adquirió o calculó la variable.
- ✓ **Calidad:** Representa la calidad del valor de la variable (9).

Los componentes gráficos a desarrollar utilizan los puntos calculados para visualizar los valores de los dispositivos en el campo. Un punto calculado es definido como una entidad de software que reside en la base de datos lo más cercano al tiempo real, la cual permite hacer operaciones aritméticas/lógicas y cuyos operandos pueden ser Valores de Puntos de Telemetría (Entrada/Salida) u otros puntos calculados. Los puntos calculados no poseen salida de campo y son de tipo Analógico o Digital dependiendo de la naturaleza de la operación a procesar:

- ✓ **Analógico:** El Valor procesado por el punto es de tipo Real y Entero.
- ✓ **Digital:** El Valor procesado por el punto es de tipo Entero (INT).

Animaciones

A un objeto se le pueden adjudicar varias animaciones a la vez. Cada acción irá asociada a una condición y necesitará que se le definan algunos parámetros tales como: tipo de animación, comparación, estado y punto asociado. Una animación es un tipo de comportamiento que puede tomar un determinado objeto que según sus características se puede animar de varias formas:

- ✓ Opacidad.
- ✓ Mover.
- ✓ Rotar.
- ✓ Escalar.
- ✓ Fondo.
- ✓ Nivel.
- ✓ Texto.

Una medición se encarga de recoger el valor de un punto en ese instante de tiempo para asignársela a un componente, cuando se establece una medición a un componente asociado a puntos digitales se crea una animación por defecto con dos formas de animación, opacidad y escalabilidad. Estos componentes válvula automática, enfriador de crudo y gas, válvula motorizada, bomba y válvula manual utilizan los puntos digitales y se creará esta animación por defecto con un estado de alarma de activa no reconocida. En casos de los componentes camión cisterna y válvula de control por posicionamiento que utilizan los puntos analógicos se crea una animación por defecto con una forma de animación de tipo color, para los estados de alarmas de activa reconocida o activa no reconocida.

Estados de las Alarmas

El estado de una alarma indica al sistema y al operador cómo se encuentran los atributos de la alarma, los mismos resumen la condición de alarma y la operación que han sido realizadas sobre ella. A continuación se resumen los estados de alarma utilizados por los componentes a desarrollar:

- ✓ **Activa Reconocida (AR).** El operador reconoce la ocurrencia de la alarma, pero la condición de alarma se mantiene activa. Cesa el parpadeo y la señal sonora.
- ✓ **Activa y No Reconocida (ANR).** Se activó la condición de alarma y pasa al estado de no reconocida. Previamente esta alarma fue rearmada para activarse. Se activa una señal sonora correspondiente con su prioridad, área o grupo de alarma. Se le coloca un color correspondiente con su prioridad y se parpadea para indicar al operador la ocurrencia.

Eventos

En términos de informática, particularmente en el lenguaje de los sistemas, evento se denomina a los sucesos generados por el sistema. Según su tipo pueden ser procesados en su totalidad por el mismo sistema o a través de mecanismos que corren bajo su propio control en forma de mensajes. En este sentido, los mensajes son interpretados por el mecanismo como una petición para que ejecute ciertas acciones. El almacenamiento de los eventos del SCADA además de ser almacenados en la Base de Datos Históricos (BDH) se debe guardar en el Log de eventos globales. Así, en caso de que falla la BDH el usuario pueda acceder a los eventos a través del Log de eventos globales (10).

Tipos de evento

Alarma: Evento generado por un dispositivo o una función que señala la existencia de una condición anormal a través de un cambio discreto audible o visible, o ambas, que requiere su atención inmediata.

Comando: Evento generado por una acción o mandato que el usuario proporciona al sistema, desde una interfaz que haya sido previamente definida para ello, por ejemplo, botones desde un despliegue o líneas de comando. De cualquier manera, este tipo de acción invoca la ejecución de una secuencia de instrucciones programada previamente en el sistema que a su vez son generadores de eventos, por ejemplo la confirmación de envío, recepción y ejecución del comando.

Sesión: Evento generado por el usuario al iniciar o cerrar una sesión en el sistema.

Configuración: Evento generado por el usuario al ejecutar cambios en la configuración del sistema, por ejemplo, modificación de variables, despliegues, usuarios, entre otros (10).

Área de herramientas gráficas del Ambiente de configuración del HMI

El editor configuración cuenta con un grupo de componentes gráficos que se clasifican según su funcionalidad y características, entre ellos están los Básicos y Avanzados.

- ✓ **Básicos:** en este grupo se incluyen componentes básicos como: línea, rectángulo, elipse, polígono, pastel, entre otros. Estas formas presentan propiedades visuales estáticas y dinámicas que son asociadas a expresiones de variables del SCADA.
- ✓ **Avanzados:** son componentes gráficos que presentan todas las funcionalidades de un componente básico, pero pueden ejecutar funcionalidades específicas del SCADA. Algunos ejemplos son:
  - **Controles:** lista desplegable, cuadro de número (spinbox), etiqueta (label), control deslizante (slider), botón pulsable, entre otros.
  - **Medidores:** visualizador, medidor vertical y sumario.
  - **Gráficos:** tendencia, gráfico xy y gráfico de barra.

Algunas de las vistas del ambiente de configuración son basadas en extensiones las cuales se detallan a continuación:

Inspector de propiedades: es la paleta que permite listar las propiedades de los componentes gráficos que se deseen configurar. Cualquier modificación realizada en una propiedad, se actualiza inmediatamente el estado del objeto.

Despliegue: es el área de edición donde se adicionan y configuran los diferentes componentes para guardar las configuraciones realizadas, permitiendo visualizarlas en el ambiente de ejecución.

Paleta de componentes: es la paleta donde se encuentran todos los componentes utilizados en el diseño de despliegues los cuales están separados por grupos.

Barra de herramientas: es el área donde se encuentran las diferentes herramientas para realizar acciones sobre el proyecto como: agregar/crear proyecto, abrir, guardar, cortar, copiar, pegar, eliminar, deshacer, rehacer, alineación, girar, agrupar, desagrupar, acercar, alejar, reajustar, modo selección y modo manual.

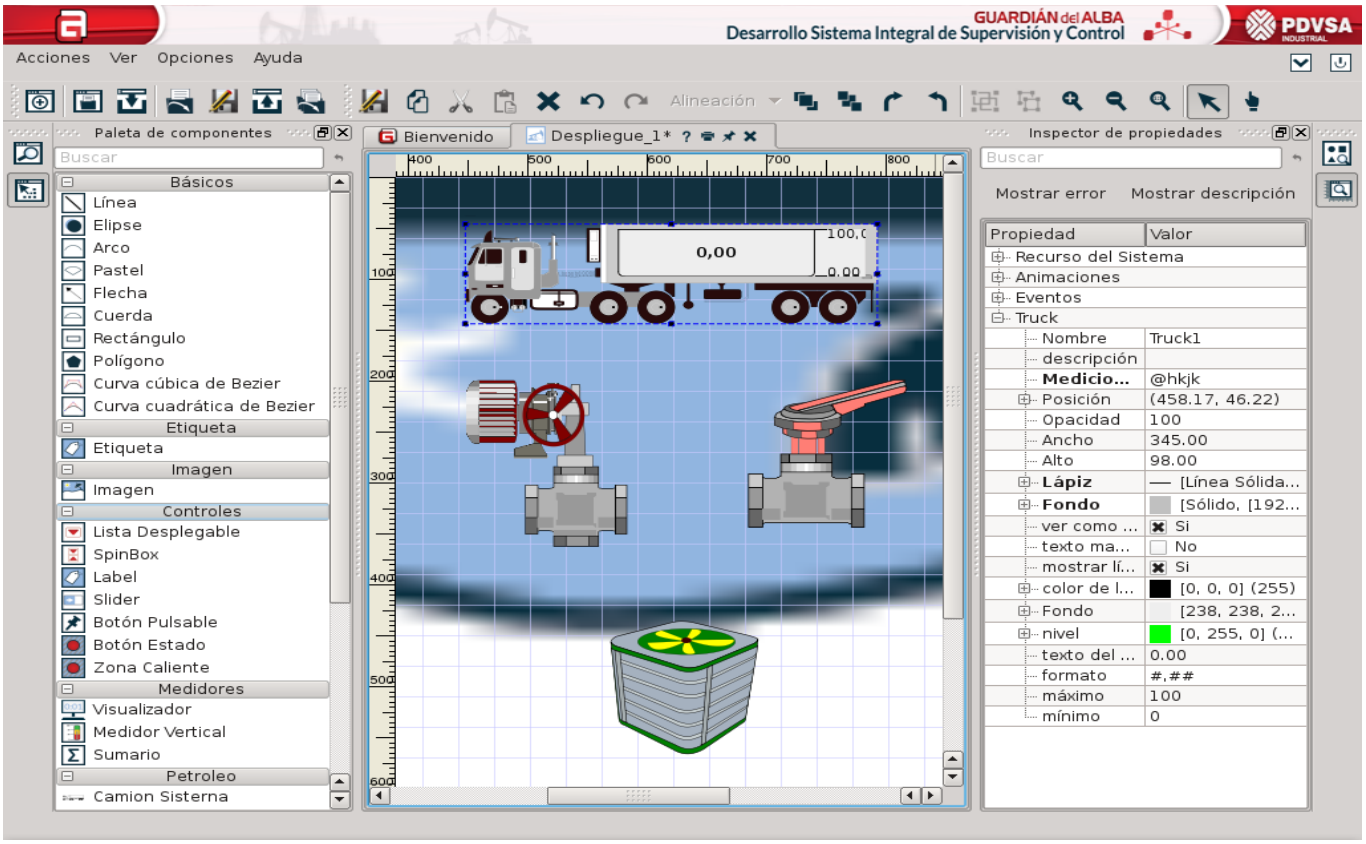


Figura 4: Ambiente de Configuración del HMI GALBA

1.4.1.2 Ambiente de ejecución

El ambiente de ejecución, también conocido como Runtime, se encarga de visualizar las animaciones y los objetos definidos en el editor. Muestra lo que está ocurriendo en el campo lo más cercano al tiempo real y es el que envía los comandos a las estaciones remotas, quién recibe los valores de las variables, interactúa con la mayoría de los operadores pues se emplea para supervisar el proceso de manera directa. Al ser el módulo que se encarga de brindar el control total sobre el proceso de producción, la interfaz de usuario brinda un conjunto de funcionalidades primarias, entre ellas la generación de reportes, impresión, análisis de variables, visualización de la tendencia de indicadores, configuración de los manejadores para la comunicación y acceso a las alarmas (11).

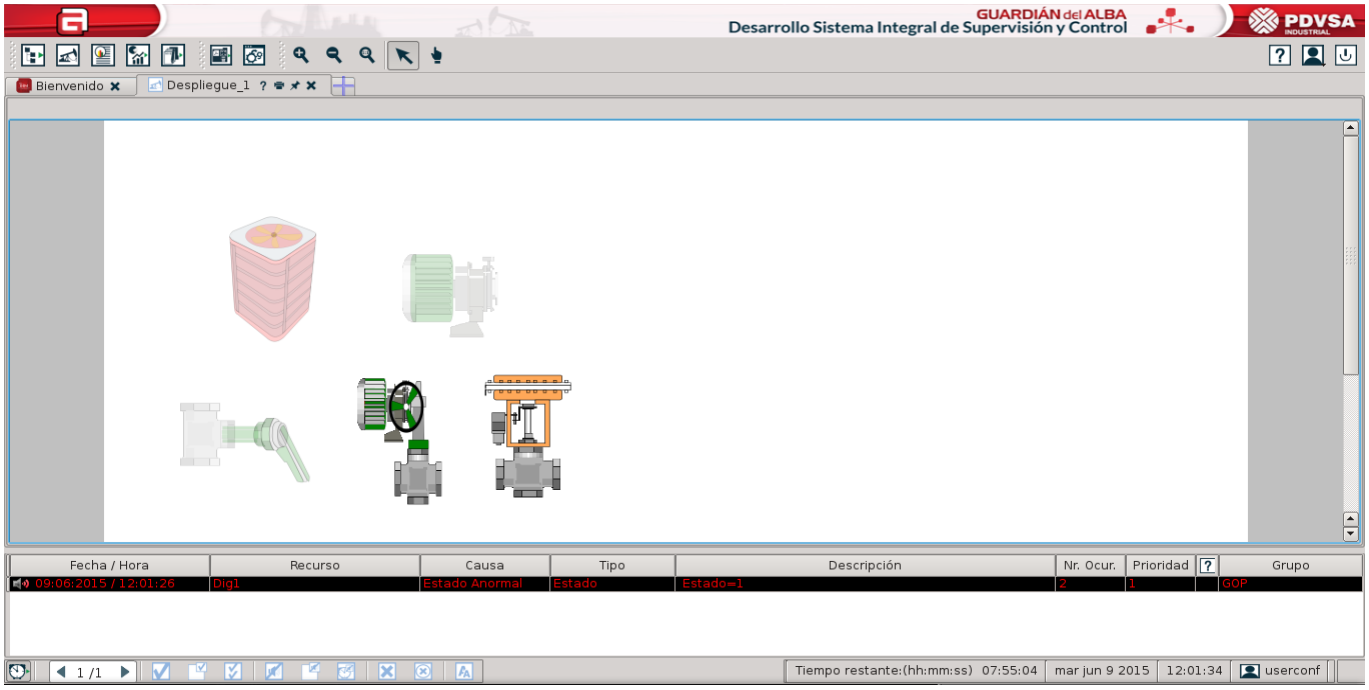


Figura 5: Ambiente de ejecución del HMI GALBA

1.5 Selección de herramientas y tecnologías

Seguidamente se realiza una valoración de las principales tecnologías y herramientas que serán empleadas en la solución del problema planteado, definiendo las características fundamentales que las hacen relevantes con respecto a las otras del mismo tipo. Como este trabajo es parte del proyecto SCADA GALBA algunas de las herramientas utilizadas fueron definidas por dicho proyecto como son: distribución Debian 7.0, Qt Creator v2.8 como entorno de desarrollo, marco de trabajo Qt framework v4.8 y lenguaje de programación C++ v98, las restantes herramientas fueron seleccionadas para apoyar el desarrollo de la paleta de componentes gráficos.

1.5.1 Gráficos Vectoriales Escalables (SVG)

Los Gráficos Vectoriales Escalables (SVG siglas en inglés) son una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados estos últimos con ayuda de lenguaje de integración multimedia sincronizada (SMIL siglas en inglés), en formato de lenguaje de marcas extensible (XML siglas en inglés) que consiste en un conjunto de reglas para representar información en una forma fácilmente procesable por un ordenador. SVG se convirtió en una recomendación del World Wide Web Consortium (W3C) en Septiembre de 2001, por lo que ya ha sido incluido de forma nativa en el navegador web del W3C Amaya (12).

SVG permite tres tipos de objetos gráficos:

- ✓ Elementos geométricos vectoriales.
- ✓ Imágenes de mapa de bits /digitales.
- ✓ Texto.

Algunas de las ventajas de SVG:

- ✓ Es un formato de gráficos vectoriales, con lo que los gráficos resultan editables, admiten curvas Bézier, transparencias, suavizados, rastrillados, etc.
- ✓ Produce gráficos escalables, que pueden aumentar o disminuir de tamaño sin pérdida de calidad, lo que los hace adaptarse sin problemas a cualquier resolución de pantalla.
- ✓ Admite gestión avanzada del color, manejando 24 bits de profundidad, pudiendo además usarse en su definición cualquiera de los sistemas estándar: RGB, CMYK, etc.
- ✓ Se pueden incluir en un gráfico SVG sonidos y etiquetas explicativas.
- ✓ Permite la creación de animaciones en escala de tiempo.
- ✓ Admite diferentes tipos de filtros, consiguiendo resultados sorprendentes.
- ✓ Posiciona el gráfico de acuerdo con el puntero del ratón.
- ✓ Es una tecnología de código libre, no propietario, con las ventajas que eso representa para los desarrolladores (13).

1.5.1.1 Inkscape

Es un editor de gráficos vectoriales de código abierto, con capacidades similares a otros editores gráficos como Illustrator, Freehand o CorelDraw. Sigue el estándar de la W3C: el formato de archivo Scalable Vector Graphics (SVG). Las características soportadas incluyen: objetos, trazos, texto, marcadores, clones, mezclas de canales alfa, transformaciones, gradientes, patrones y agrupamientos (13).

Algunas de las ventajas de Inkscape:

- ✓ Puede importar formatos como Postscript, JPEG, PNG, y TIFF y exporta PNG así como muchos formatos basados en vectores.
- ✓ Totalmente compatible con los estándares XML, SVG y CSS.

- ✓ Herramienta multiplataforma, funciona en Windows, Mac OS X y otros sistemas derivados de Unix.

1.5.2 GIMP (GNU Image Manipulation Program)

GIMP es un programa libre y gratuito de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Permite el tratado de imágenes en capas, para poder modificar cada objeto de la imagen en forma totalmente independiente a las demás capas en la imagen. También pueden subirse o bajarse de nivel las capas para facilitar el trabajo de la imagen. La imagen final puede guardarse en el formato original xcf de GIMP que soporta capas, o en un formato plano sin capas, que puede ser PNG, BMP, JPG, GIF, PDF, etc. También importa imágenes vectoriales en formato SVG creadas, por ejemplo, con Inkscape.

1.5.3 Marco de trabajo

Un marco de trabajo es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado (14). Típicamente, puede incluir soporte de programas, bibliotecas, un lenguaje de programación, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

1.5.3.1 Qt Framework

Qt es un marco de trabajo multiplataforma en C++ de desarrollo de aplicaciones. Se utiliza fundamentalmente para desarrollar aplicaciones con interfaz gráfica, gracias al conjunto de controles independientes de la plataforma que ofrece, aunque también es usado para crear herramientas de línea de comando o consolas de gestión para servicios. Qt está disponible para sistemas tipo UNIX (Linux, BSD, UNIX, etc.) con servidor gráfico X WindowSystem, Apple Mac OS X, Microsoft Windows y sistemas Linux embebidos. Además se puede hacer uso de la librería desde lenguajes diferentes a C++ gracias al empleo de bindings: Python, Java, Ruby, Ada, Pascal, Perl, PHP, Haskell, Lua, D, .NET, etc. (15).QGraphicsView es el framework dentro de Qt que permite la creación e interacción de elementos gráficos 2Dy un widget de vista para visualizar los objetos, con soporte para el zoom y rotación.

1.5.4 Entorno de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Además pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Ofrecen un marco de

trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (16).

1.5.4.1 Qt-Creator

QtCreator es un IDE multiplataforma que se ajusta a las necesidades de los desarrolladores Qt. Además se centra en proporcionar características que ayudan a los nuevos usuarios de Qt a aprender y comenzar a desarrollar rápidamente. Posee como principales características un editor de código con soporte para C++, QML y ECMAscript, control estático de código y estilo a medida que se escribe. También consta con herramientas para la rápida navegación del código y una ayuda sensible al contexto integrada (17).

1.5.5 Lenguaje de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C y le agrega la capacidad de abstraer tipos como en Smalltalk. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma (18).

Características generales:

- ✓ C++ es un lenguaje híbrido.
- ✓ Es multiplataforma, orientado a objetos e imperativo.
- ✓ Usa tipos de datos fuertes y estáticos.
- ✓ Manejo de memoria por parte del programador, lo que permite un mejor control de esta y buena administración de recursos de computadora.
- ✓ Posibilidad de redefinir los operadores (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales.
- ✓ C++ permite trabajar tanto a alto como a bajo nivel (19).

1.5.6 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML, en sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado visual de propósito general orientado a objetos. Este lenguaje se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. UML cubre las diferentes vistas de la arquitectura de un sistema mientras evoluciona a través del ciclo de vida del desarrollo de software.

1.5.6.1 Herramienta para el modelado

Visual Paradigm es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Algunas ventajas que proporciona Visual Paradigm son: dibujo para facilitar el modelado de UML, ya que proporciona herramientas específicas para ello, reutilización porque dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos y generación de código dado que permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software (20).

1.5.7 Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software (21). Cubren todo el ciclo de desarrollo del producto, estableciendo etapas y controles a aplicar en cada momento. Estas recopilan un conjunto de técnicas y procedimientos en cada una de las fases que las componen. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software.

1.5.7.1 Metodología de desarrollo de software AUP

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera fácil la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles como son:

- ✓ Desarrollo Dirigido por Pruebas (Test Driven Development - TDD en inglés).
- ✓ Modelado ágil.
- ✓ Gestión de Cambios ágil.
- ✓ Refactorización de Base de Datos para mejorar la productividad (22).

Esta metodología cuenta con las siguientes fases:

1- Inicio: En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

2- Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y libera el producto, este producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.

3- Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (22).

1.6 Conclusiones parciales.

El estudio del estado del arte y las principales soluciones existentes de los componentes gráficos en los diferentes SCADA, permitieron sentar las bases teóricas para la futura implementación de la solución propuesta. Las herramientas y tecnologías que se escogieron para el desarrollo de dicha solución son de código abierto y multiplataforma, esta selección permitió que cumpliera con las políticas de migración al software libre establecidas en el país. Se seleccionó la metodología de desarrollo AUP teniendo en consideración que es la más usada en los proyectos de la UCI.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

2.1 Introducción

Este capítulo tiene como objetivo describir las actividades desarrolladas durante todo el proceso de análisis y diseño de la solución. Se identifican los requerimientos funcionales y no funcionales que se deben tener en cuenta para la elaboración de la propuesta de solución. Se analizan los conceptos fundamentales del modelo de dominio, se hace una representación de los actores, casos de uso del sistema y se describen las principales clases y componentes.

2.2 Descripción de la propuesta de solución

Se propone la implementación de una paleta de componentes gráficos especializados del sector petrolero para el módulo HMI del SCADA Guardián del ALBA. Los componentes del sistema podrán ser creados, modificados o eliminados en el despliegue, además se le podrán definir animaciones y mediciones según el propósito del operador. Estos componentes usarán los puntos calculados de tipo analógico y digital, entre los que usarán los puntos analógicos están camión cisterna y válvula de control por posicionamiento, los restantes componentes válvula automática, enfriador de crudo y gas, válvula motorizada, bomba y válvula manual utilizarán los puntos digitales. Al establecerle una medición a un componente asociado a un punto analógico se añadirá una animación por defecto de color de fondo para los estados de alarma activa reconocida y activa no reconocida. Por otra parte si se le establece una medición a un componente asociado a un punto digital se añadirá una animación por defecto de opacidad y escalabilidad con estado de alarma de activa no reconocida. Además soportarán las funcionalidades básicas de los componentes existentes en el ambiente de configuración. En la descripción de los casos de uso se muestran las imágenes de los componentes de la propuesta de solución, para esto es necesario consultar los anexos. También se deben integrar los componentes desarrollados a la paleta de componentes del módulo HMI lo que permitirá disminuir la creación de dichos componentes.

2.3 Modelo de Dominio

El Modelo de Dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real en lugar de componentes de software (23).

2.3.1 Diagrama de clases del dominio

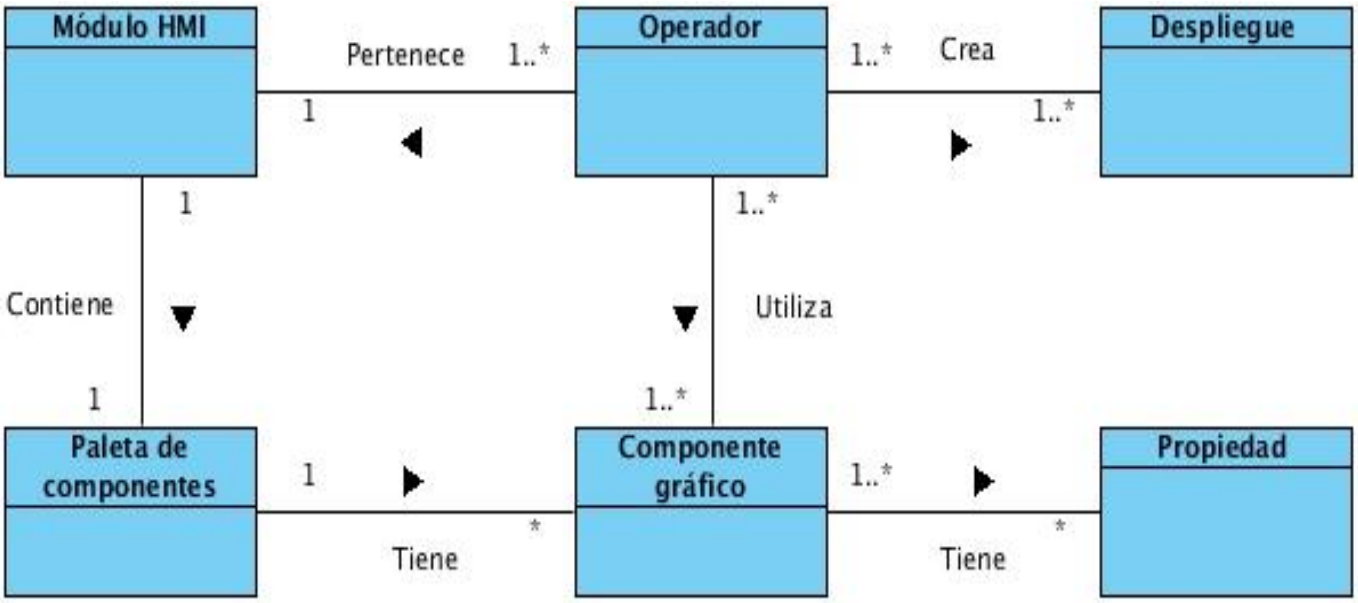


Figura 6: Modelo de Dominio

2.3.1.1 Descripción de las clases conceptuales del Modelo de Dominio

El **Módulo HMI** es el encargado de solicitar o modificar la configuración mediante el ambiente de configuración.

La **Paleta de componentes** es la vista que tiene agrupados los diferentes componentes gráficos del módulo HMI.

Los **Componentes gráficos** son componentes visualizadores encargados de mostrar los valores recogidos del campo mediante los dispositivos.

El **Operador** es el que utiliza y configura los componentes gráficos de la paleta de componentes para la creación de los despliegues.

El **Despliegue** es el área de edición donde se adicionan y configuran los diferentes componentes para guardar las configuraciones realizadas, permitiendo visualizarlas en el ambiente de ejecución.

Las **Propiedades** son los diferentes atributos que tienen y se le configuran a los componentes gráficos como son las animaciones, mediciones, eventos, entre otras.

2.4 Requisitos funcionales y no funcionales del sistema.

Durante el desarrollo de la etapa de análisis y diseño se definen como requisitos funcionales y no funcionales los que a continuación se describen, teniendo como prefijo los requisitos funcionales las letras RF y de los no funcionales las letras RNF:

2.4.1 Requerimientos Funcionales

Los requerimientos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir, se dividieron en requerimientos generales y específicos:

Tabla 1: Requisitos funcionales

Requisitos funcionales	**Descripción**
Generales	Los requisitos generales son aquellos que deben cumplir todos los componentes.
RF 1 Crear componente en el despliegue.	Permitir crear el componente arrastrándolo de la paleta al despliegue.
RF 2 Eliminar componente del despliegue.	Permitir eliminar los diferentes componentes del despliegue.
RF 3 Soportar funcionalidades básicas de los componentes existentes:	Permitir las funcionalidades básicas definidas en los componentes existentes en el ambiente de configuración.
✓ **RF 3.1** Definir nombre del componente.	Establecer el nombre apropiado del componente gráfico.
✓ **RF 3.2** Establecer rotación del componente.	Establecer una rotación del componente rotando siempre por el centro.
✓ **RF 3.3** Definir descripción del componente.	Establecer una descripción asociada a la funcionalidad o utilidad del componente.
✓ **RF 3.4** Definir mediciones del	Establecer las mediciones asociadas a puntos

componente.	genéricos o reales del componente.
✓ **RF 3.5** Definir opacidad del componente.	Establecer el nivel de opacidad del componente
✓ **RF 3.6** Definir posición del componente.	Definir la posición del componente determinada por las coordenadas X, Y.
✓ **RF 3.7** Definir tamaño del componente.	Establecer ancho y alto del componente.
✓ **RF 3.8** Definir color del componente.	Permitir modificar el color de fondo de la parte dinámica del componente.
✓ **RF 3.9** Definir animaciones sobre el componente.	Establecer las diferentes animaciones sobre el componente. Seleccionar la variable que define el cambio del color. Editar correspondencia entre los valores discretos de la variable con los colores deseados para cada valor. Definir si habilitar un parpadeo (blink).
✓ **RF 3.10** Visualizar componente en el ambiente de ejecución.	Permitir visualizar las configuraciones realizadas en el despliegue.
Específicos	Los requisitos específicos son aquellos que deben cumplir un componente determinado.
Camión cisterna	
RF 4 Definir atributos del nivel.	Modificar color del fondo, borde y relleno de la parte dinámica del nivel a través de una paleta de colores.

2.4.2 Requerimientos No Funcionales

Un requisito no funcional (RNF) especifica los criterios que se deben usar para juzgar el funcionamiento de un sistema, en lugar de un comportamiento específico. En general, los requisitos no funcionales verifican cómo un sistema debería de ser (24). Para la paleta de componentes propuesta se definen los siguientes requerimientos no funcionales:

- ✓ **Requerimientos de Seguridad:**

RNF 1: La paleta de componentes propuesta podrá ser usada por cualquier persona autorizada que pertenezca al módulo HMI del SCADA GALBA.

- ✓ **Requerimientos de Software**

RNF 2: Como sistema operativo GNU/Linux, recomendándose utilizar la distribución Debian 7.0 wheezy.

- ✓ **Restricciones del diseño y la implementación**

RNF 3: Para el diseño de la interfaces visuales Qt.

RNF 4: Como lenguaje de programación C++.

- ✓ **Requerimientos de Confiabilidad**

RNF 5: La herramienta debe ser exacta en la información que le suministra al usuario para evitar cualquier tipo de error.

- ✓ **Requerimientos Legales**

RNF 6: Se utilizan herramientas de software libre de las cuales se posee licencia para su uso y distribución.

2.5 Descripción de los actores

Los actores del sistema son todas las entidades externas al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo.

Se identificó como único actor del sistema al operador.

Tabla 2: Actores del sistema

Actor	Descripción

Operador	Puede ser un operador o cualquier otro especialista asociado a la rama, con los que el sistema interactúa dentro de las entidades de configuración.

2.6 Diagrama de Casos de Uso del Sistema

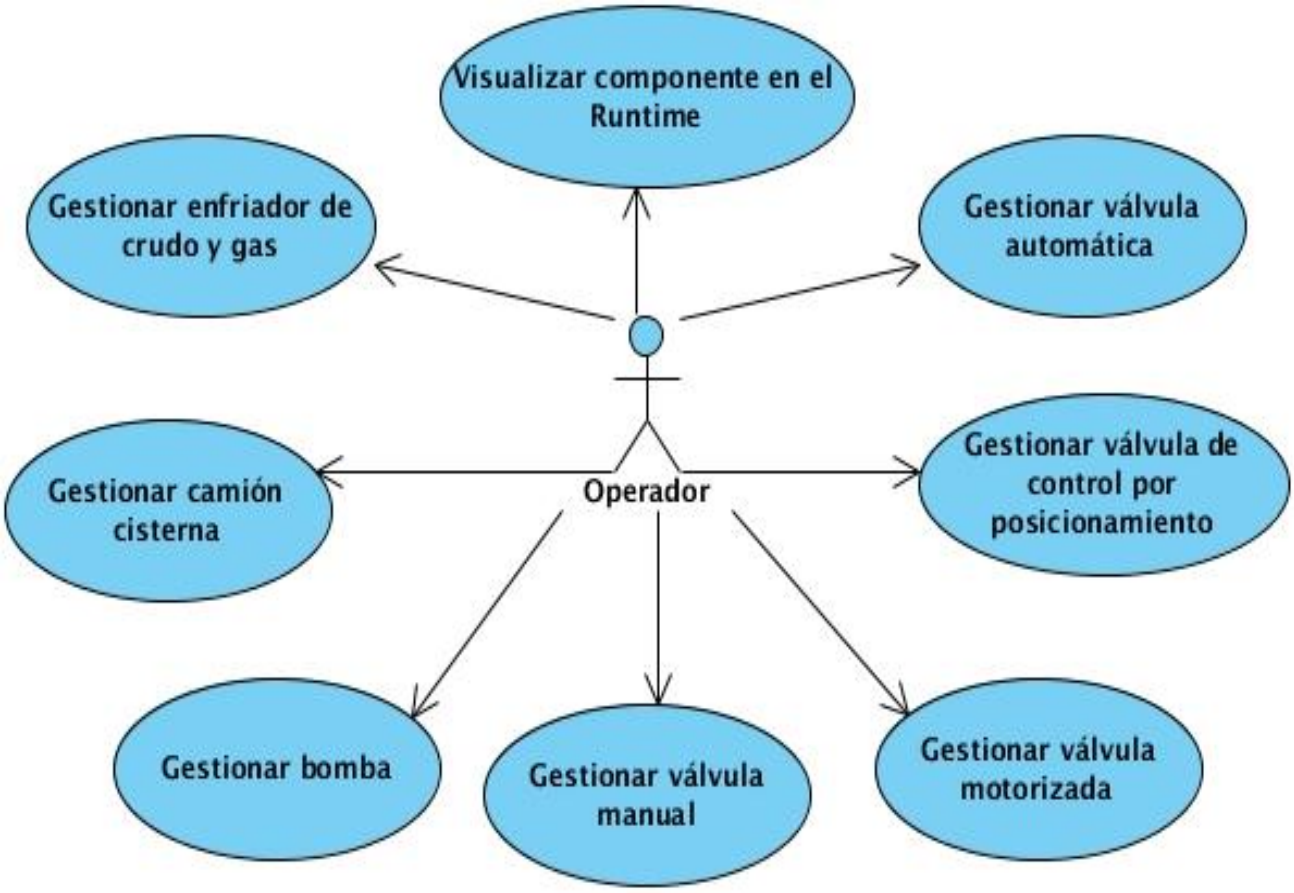


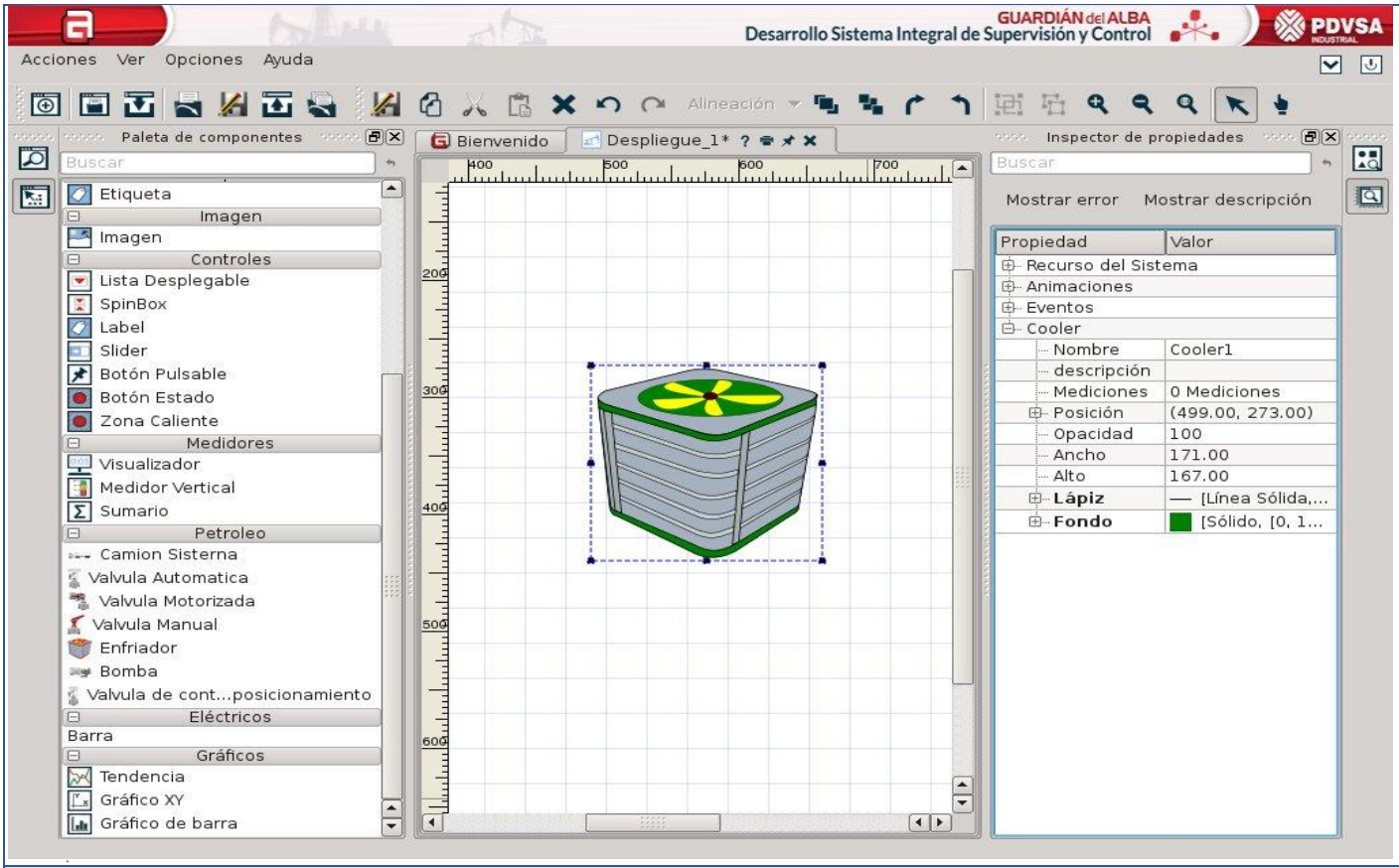
Figura 7: Diagrama de Casos de Uso

2.6.1 Descripción textual de los Casos de Usos del Sistema (CUS).

A continuación se presenta la descripción textual del CUS Gestionar enfriador de crudo y gas, la descripción de los casos de uso restantes puede encontrarse en el Anexo 1 de la presente investigación. Los prototipos de los CUS mostrados a continuación y en los anexos son resultados de la integración de la propuesta de solución con el editor de configuración del módulo HMI.

Tabla 3: Descripción CUS Gestionar enfriador de crudo y gas

Caso de uso	Gestionar enfriador de crudo y gas
Actor	Operador
Resumen	El caso de uso se inicia cuando el operador decide adicionar, eliminar o modificar el componente gráfico en un despliegue.
Referencias	RF1, RF2, RF3.1, RF3.2, RF3.3, RF3.4, RF3.5, RF3.6, RF3.7, RF3.8 y RF3.9.
Precondiciones	Debe existir un despliegue.
Prioridad	Crítico.
Flujo de eventos	
Flujo básico	
Actor	**Sistema**
1. Desea realizar las siguientes acciones: • Si decide adicionar el componente gráfico, ir a la sección, **"Adicionar componente gráfico".** • Si decide eliminar el componente gráfico, ir a la sección, **"Eliminar componente gráfico".** • Si decide modificar el componente gráfico, ir a la sección, **"Modificar componente gráfico".**	
Sección "Adicionar componente gráfico"	
Flujo básico Adicionar componente gráfico	
Actor	**Sistema**
1. Selecciona en la Paleta de componentes el componente Enfriador y lo arrastra hasta el área de edición del despliegue.	2. Verifica que el operador arrastra el componente al área de edición del despliegue. 3. Adiciona el componente gráfico al despliegue. 4. Termina el caso de uso.
Flujo alterno de eventos "No es el área de edición"	
Flujo básico No es el área de edición	
Actor	**Sistema**
	2.1 No se adiciona el componente en el área de edición del despliegue.
Prototipo de interfaz de Usuario	



Sección “Modificar componente gráfico”	
Flujo básico Modificar componente gráfico	
Actor	**Sistema**
1. Selecciona sobre el área de edición el componente gráfico que desea modificar y escoge el inspector de propiedades.	2. Muestra en el Inspector de propiedades las siguientes opciones: • Recurso del Sistema. • Animaciones. • Eventos. • Cooler.
3. Selecciona una de las opciones: Animaciones o Cooler.	4. Ejecuta una de las siguientes acciones: • Si selecciona la opción Animaciones, ir a la sección **“Animaciones”.** • Si selecciona la opción Cooler, ir a la sección **“Cooler”.**
Sección “Animaciones”	
Flujo básico Animaciones	
Actor	**Sistema**
1. Despliega el campo Animaciones.	2. Muestra el campo Umbrales y el botón “…”.
3. Selecciona el botón “…”.	4. Muestra una ventana con los campos requeridos para crear una o varias animaciones.

5. Ingresa los datos requeridos y presiona el botón “Aceptar”.	6. Verifica que la colección de condiciones no está vacía. 7. Verifica que las condiciones no tengan mediciones asociadas. 8. Crea la animación correctamente. 9. Termina el caso de uso.
Flujo alterno de eventos “Colección de condiciones vacía”	
Flujo básico Colección de condiciones vacía	
Actor	**Sistema**
	6.1 Muestra el mensaje “Se debe crear al menos una condición para adicionar un umbral”.
Flujo alterno de eventos “Condición sin mediciones”	
Flujo básico Condición sin mediciones	
Actor	**Sistema**
	7.1 Muestra el mensaje “No puede quedar una condición vacía”.
Sección “Cooler”	
Flujo básico Cooler	
Actor	**Sistema**
1. Despliega el campo Cooler.	2. Muestra una lista con los siguientes campos: • Nombre. • Descripción. • Mediciones. • Posición. • Opacidad. • Ancho. • Alto. • Fondo.
3. Selecciona el campo deseado a modificar.	4. Ejecuta una de las siguientes acciones: • Si decide modificar los campos nombre, descripción, opacidad, ancho o alto, ir a la sección **“Modificar propiedades”.** • Si decide modificar mediciones, ir a la sección **“Mediciones”.** • Si decide modificar posición, ir a la sección **“Posición”.** • Si decide modificar Fondo, ir a la sección **“Fondo”.**
Sección “Modificar propiedades”	
Flujo básico Modificar propiedades	
Actor	**Sistema**
	1. Muestra el valor actual del campo.
2. Modifica los datos del campo y presiona la tecla “Enter”.	3. Verifica que el campo no está vacío. 4. Modifica la propiedad correctamente.

	5. Termina el caso de uso.
Flujo alterno de eventos "Campo vacío"	
Flujo básico Campo vacío	
Actor	**Sistema**
	4.1 No modifica el campo y muestra el valor anterior.
Sección "Mediciones"	
Flujo básico Mediciones	
Actor	**Sistema**
	1. Muestra el valor actual del campo mediciones y los siguientes botones: • "…" • "<--"
2. Selecciona uno de los botones.	3. Ejecuta una de las siguientes acciones: • Si selecciona el botón "…", ir a la sección **"Crear Medición"**. • Si selecciona el botón "<--", ir a la sección **"Eliminar Medición"**.
Sección "Crear Medición"	
Flujo básico Crear Medición	
Actor	**Sistema**
	1. Muestra una ventana con los campos requeridos para crear una medición.
2. Ingresa los datos requeridos y presiona el botón "Aceptar".	3. Verifica que el operador selecciona un punto. 4. Crea la medición correctamente. 5. Establece una animación por defecto. 6. Termina el caso de uso.
Flujo alterno de eventos "Punto no seleccionado"	
Flujo básico Punto no seleccionado	
Actor	**Sistema**
	3.1 Muestra el mensaje "No ha seleccionado ningún recurso".
Sección "Eliminar Medición"	
Flujo básico Eliminar Medición	
Actor	**Sistema**
	1. Elimina la medición satisfactoriamente.
Sección "Posición"	
Flujo básico Posición	
Actor	**Sistema**
	1. Muestra el valor actual del campo posición.
2. Presiona doble click sobre el campo posición.	3. Muestra los campos "X" y "Y" con sus valores actuales.

4. Ingresa los valores deseados en los campos para obtener una nueva posición y presiona la tecla “Enter”.	5. Verifica que los campos no están vacíos (Ver Flujo alterno de eventos “Campo vacío”). 6. Modifica la posición correctamente. 7. Termina el caso de uso.
Sección “Fondo”	
Flujo básico Fondo	
Actor	**Sistema**
	1. Muestra el valor actual del campo fondo.
2. Presiona doble click sobre el campo fondo.	3. Despliega los campos “Estilo“ y “Color“ con sus valores actuales.
4. Selecciona el campo “Color“.	5. Muestra el valor actual del campo color y el botón “…”.
6. Presiona el botón “…”.	7. Muestra una ventana con una paleta de colores y diferentes parámetros a modificar.
8. Selecciona los parámetros deseados y presiona el botón “OK”.	9. Modifica el color satisfactoriamente. 10. Termina el caso de uso.
Flujo alterno de eventos “Cancelar”	
Flujo básico Cancelar	
Actor	**Sistema**
8.1 Selecciona los parámetros deseados y presiona el botón “Cancelar“.	8.2 Mantiene los valores anteriores.
Sección “Eliminar componente gráfico”	
Flujo básico Eliminar componente gráfico	
Actor	**Sistema**
1. Selecciona en el área de edición el componente gráfico a eliminar y presiona click derecho sobre el componente.	2. Muestra las acciones siguientes: • Copiar. • Cortar. • Eliminar. • Traer al frente. • Enviar al fondo. • Girar a la derecha. • Girar a la izquierda. • Propiedades.
3. Selecciona la opción Eliminar.	4. Verifica que el operador selecciona la opción “Eliminar”. 5. Elimina el componente gráfico seleccionado. 6. Termina el caso de uso.
Flujo alterno de eventos “Eliminado incorrectamente”	
Flujo básico Eliminado incorrectamente	
Actor	**Sistema**
	4.1 No se elimina el componente gráfico del despliegue.

2.7 Diagrama de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de agregación. Una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica; mostrando un conjunto de elementos que son estáticos, como las clases y tipos junto con sus contenidos y relaciones (25).

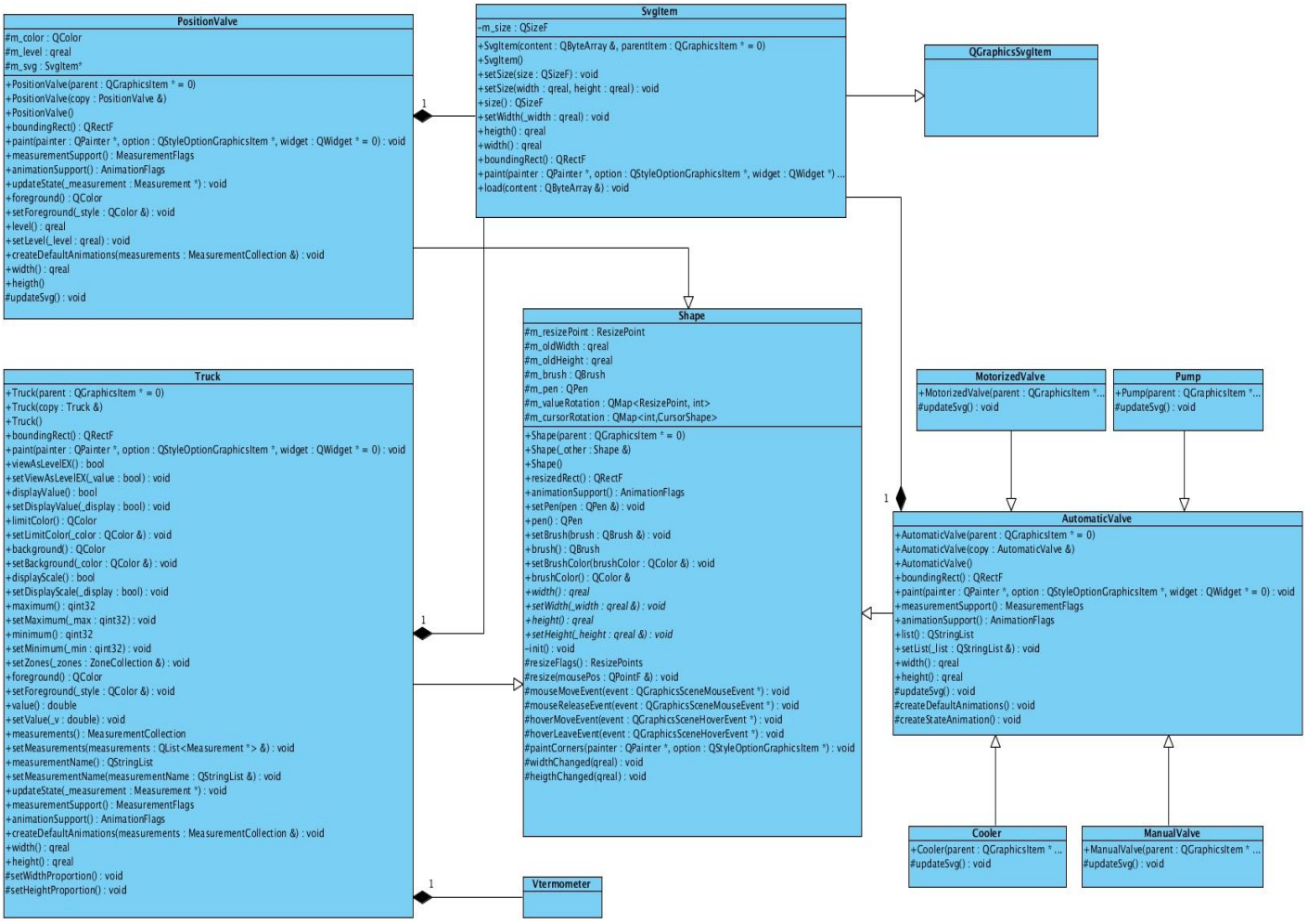


Figura 8: Diagrama de clases

2.7.1 Descripción de las clases más relevantes

Una clase en la implementación del sistema tiene propiedades (atributos), comportamientos (métodos) y relaciones con otras clases.

Tabla 4: Descripción de la clase SvgItem

Nombre	SvgItem
Descripción	La clase SvgItem hereda de QGraphicsSvgItem, por lo que posee todas las funcionalidades de esta. Se encarga de renderizar los contenidos de los ficheros SVG.
Atributos	m_size
Principales métodos	
Nombre	SvgItem
Descripción	Constructor de la clase, es el encargado de inicializar los atributos.
Nombre	paint
Descripción	Es el encargado de pintar el contenido de un elemento en coordenadas locales.
Nombre	setSize
Descripción	Esta propiedad tiene el tamaño del dibujo SVG generado. Por defecto esta propiedad se establece en qsize (-1, -1), lo que indica que el generador no debe emitir el ancho y alto de los atributos del elemento SVG.

Tabla 5: Descripción de la clase Shape

Nombre	Shape
Descripción	Esta clase contiene el ancho, altura, pluma (pen) y brocha (brush) de un componente. La implementación por defecto se llama boundingRect () para devolver una forma rectangular simple, pero las subclases pueden reimplementar esta función para devolver una forma más precisa para los componentes no rectangulares.
Atributos	m_resizePoint, m_oldWidth, m_oldHeight, m_brush, m_pen, m_valueRotation, m_cursorRotation
Principales métodos	
Nombre	animationSupport

Descripción	Posee las diferentes animaciones que se le pueden realizar a los componentes.
Nombre	resize
Descripción	Permite cambiar el ancho y alto del componente cuando es modificado mediante el mouse.
Nombre	setBrushColor
Descripción	Establece el color de la brocha (brush) con que se va a pintar las formas.
Nombre	mouseMoveEvent
Descripción	Es utilizado para recibir eventos de movimiento del mouse.
Nombre	resizedRect
Descripción	Es utilizado para actualizar cuando el área actual ha cambiado de tamaño o medida.

2.8 Arquitectura del Sistema

La arquitectura de software define la estructura del sistema. Esta estructura se constituye de componentes o piezas de código que nacen de la noción de abstracción, cumpliendo funciones específicas e interactuando entre sí con un comportamiento definido (26). Puede considerarse entonces como el "puente" entre los requisitos del sistema y la implementación.

Qt está basado en la arquitectura modelo-vista, también aplicada al trabajo con gráficos. Cuenta con un conjunto de clases fundamentales para este propósito como son:

- ✓ ***QGraphicsView*** es la clase encargada de visualizar en una ventana desplazable el contenido de la escena. Permite además la interacción con los elementos de la escena mediante eventos de teclado y mouse.
- ✓ ***GGraphicsScene*** es la clase que representa al modelo y es la encargada de almacenar los elementos gráficos. Proporciona de manera eficiente la posición de los elementos gráficos 2d gracias a la técnica *BSP tree* (*Binary Space Partitioning*) que almacena los ítems en dependencia de su ubicación espacial. Determina que elementos de la escena deben ser actualizados, cuáles son visibles y cuáles no, y detecta colisiones entre ellos.

- ✓ La clase abstracta ***QGraphicsItem*** es la base de todos los elementos gráficos de una escena. Incluye las definiciones de la geometría, detección de colisiones, la implementación de su pintado y la interacción de estos con los eventos de teclado y mouse.
- ✓ **QGraphicsObject** es la clase base para todos los elementos gráficos que requieren signals, slots y propiedades.
- ✓ **QGraphicsSvgItem** puede ser utilizada para renderizarlos contenidos de ficheros SVG.
- ✓ **QGraphicsWidget** es la clase base para todos los elementos widget en una QGraphicsScene (27).

2.8.1 Descripción de la Arquitectura Modelo-Vista en Qt.

El patrón Arquitectónico MVC se divide en tres componentes esenciales:

- **Modelo:** Es el encargado de comunicarse con la fuente de datos, proporciona una interfaz para los otros componentes en la arquitectura. La naturaleza de la comunicación depende del tipo de fuente de datos y la forma en que se implementa el modelo.
- **Vista:** Obtiene los índices del modelo, estos constituyen referencias a los elementos de datos. Mediante los índices del modelo la vista puede recuperar los elementos de datos de la fuente de datos.
- **Delegado:** En una vista estándar un delegado representa los elementos de datos, cuando un elemento es editado el delegado es el encargado de comunicarse directamente con el modelo usando los índices del mismo (28).

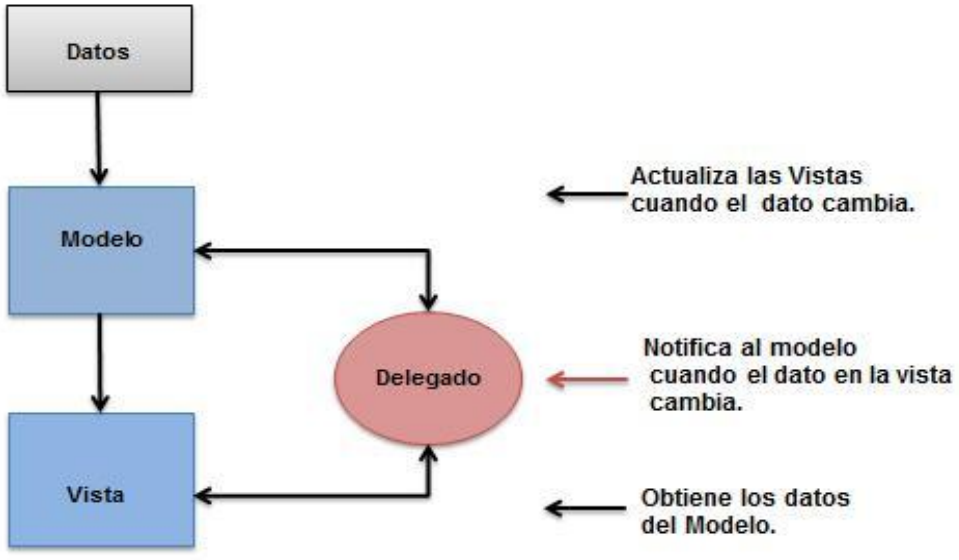


Figura 9: Arquitectura Modelo-Vista en Qt

2.8.1.1 Beneficios de usar Modelo Vista Controlador.

La aplicación del patrón MVC en la creación de un software brinda a sus desarrolladores un número de beneficios que permiten agilizar la implementación, dentro de ellos se pueden encontrar:

- Facilita y flexibiliza la estructuración del código.
- Clara separación de responsabilidades entre interfaz, lógica de negocio y de control.
- Ofrece sencillez para crear distintas representaciones de los mismos datos.
- Permite la reutilización de los componentes.
- Tiene facilidad para desarrollar prototipos rápidos (28).

2.9 Patrones de Diseño

Un patrón de diseño nombra, abstrae e identifica los aspectos fundamentales de una estructura común de diseño que la hacen útil para la creación de un diseño orientado a objetos reutilizable (29). Estos patrones facilitan el aprendizaje de los diseñadores compendiando conocimientos ya existentes, estandarizan el modo de realizar el diseño, proporcionan elementos reusables en el diseño de software y establecen un vocabulario común entre los diseñadores.

Los principales patrones de diseño son los conocidos como GoF (del inglés Gang Of Four, en español grupo de los cuatro). En este grupo de patrones se encuentran los:

- Creacionales (Abstract Factory, Builder, Factory Method, Prototype, Singleton).
- Estructurales (Adapter, Bridge, Decorator, Facade).
- Patrones de comportamiento (Command, Observer, State, Visitor, Composite, Template Method).

También están los patrones de diseño orientado a objeto para la asignación de responsabilidades o comúnmente denominados patrones GRASP (de inglés General Responsibility Assignment Software Patterns), que como su nombre lo indica son los encargados de realizar responsabilidades según la función de cada uno de estos.

Los patrones de diseño GRASP son los siguientes:

- Experto en Información.
- Creador.
- Alta Cohesión.
- Bajo Acoplamiento.
- Controlador:

2.9.1 Patrones de Diseño utilizados

2.9.1.1 Experto

Se encarga de asignar responsabilidades al experto en información, o sea, aquella clase que cuenta con la información necesaria para el cumplimiento de las mismas (30). Este patrón se pone de manifiesto en la clase Shape pues cuenta con la información necesaria para configurar la forma en que se dibujan los componentes.

2.9.1.2 Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos (30). Este patrón se evidencia en la clase Truck el cual cuenta con los parámetros de inicialización para crear instancia de la clase Vtermometer.

2.9.1.3 Método de plantilla

Permite que ciertos pasos de un algoritmo definido en una operación de una superclase, sean redefinidos en las subclases sin necesidad de tener que sobrescribir la operación entera (31). Este patrón se evidencia en la clase Shape y sus derivaciones.

2.10 Conclusiones Parciales

En este capítulo se identificaron los requerimientos funcionales y no funcionales de la solución propuesta, lo cual permitió establecer las pautas para la futura implementación del sistema. Para el cumplimiento del objetivo planteado se obtuvo el modelo de domino y se realizó la especificación de casos de uso así como el diagrama de casos de uso del sistema. Se definió la propuesta de solución y se describieron la arquitectura de software y los patrones de diseños utilizados con el objetivo de optimizar su desarrollo.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

Como resultado de la descripción de la solución del sistema, se procede a la implementación y prueba de la propuesta de solución. En el presente capítulo se exponen los diagramas de componentes para estructurar el modelo de implementación y el diagrama de despliegue donde se muestran las relaciones físicas entre los componentes hardware y software en el sistema final.

3.2 Modelo de Implementación

En la etapa de implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes. El modelo de implementación está compuesto por los diagramas de despliegue y componentes; este describe los componentes a construir y su organización en nodos físicos en los que funcionará el sistema, componentes tales como: ficheros ejecutables, código fuente y otros tipos de ficheros que sean necesarios para la implementación y despliegue del sistema.

3.2.1 Diagrama de componentes

El diagrama de componentes muestra los elementos de diseño de un sistema de software. Permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces (32). Este diagrama muestra el resultado de la integración de la propuesta de solución con el editor de configuración del módulo HMI.

El lenguaje de modelado UML define cinco estereotipos estándar que se aplican a los componentes:

- Executable: Especifica un componente que se puede ejecutar en un nodo.
- Library: Especifica una biblioteca de objetos estática o dinámica.
- Table: Especifica un componente que representa una tabla de una base de datos.
- File: Especifica un componente que representa un documento que contiene código fuente o datos.
- Document: Especifica un componente que representa un documento (32).

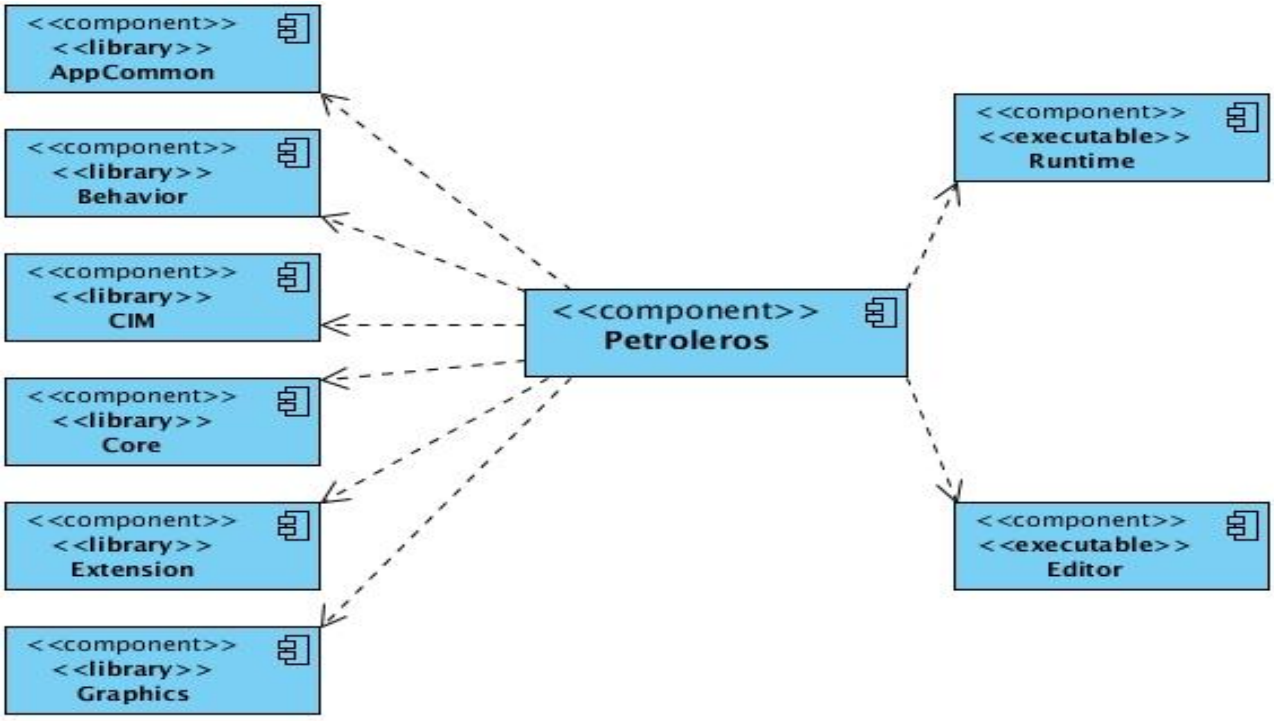


Figura 10: Diagrama de componentes

3.2.2 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Este muestra la configuración de los elementos de hardware (nodos) y cómo los elementos y artefactos del software se trazan en esos nodos (32). A continuación se muestra el diagrama de despliegue modelado para la aplicación a desarrollar, ver **Figura 11.**

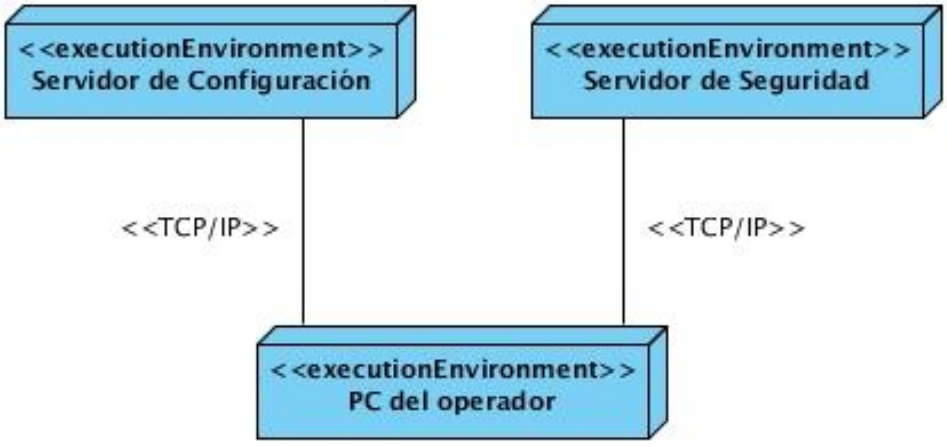


Figura 11: Diagrama de despliegue

3.3 Descripción de Nodos.

- ✓ **Servidor de Configuración:** Este nodo es el encargado de poseer toda la configuración del sistema.
- ✓ **Servidor de Seguridad:** Este nodo es utilizado para autenticarse cuando el operador necesita cargar o descargar una configuración en el sistema, además de restringir permisos a algunos usuarios de visualizar determinada información en el Runtime.
- ✓ **PC del operador:** Este nodo representa una computadora (pc) de un operador que necesita configurar o visualizar componentes gráficos en el SCADA Guardián del ALBA, mediante el editor de configuración y el Runtime que también pueden estar situados en diferentes computadoras.

3.4 Estándares de codificación

Un estándar de codificación define el formato que el programador debe usar al escribir un programa. El uso de un buen estándar de codificación mejora la legibilidad del código y facilita la comunicación entre programadores (33). Además ayuda a mejorar el proceso de codificación haciéndolo eficiente y en muchos casos reutilizables.

Como la solución propuesta en este trabajo es parte del sistema SCADA GALBA el estándar de codificación utilizado fue definido por el proyecto:

- Se adopta el estilo de bloques de documentación de JavaDoc, el cual consiste de un bloque de comentario de estilo C.
- Para hacer una descripción breve se adopta el uso del comando @brief.
- Especificar el nombre del autor y la fecha de creación de cualquier estructura en un código, para ello se utilizan los comandos @autor y @date.
- Para hacer referencia a otras clases utilizar el comando @see.
- El código será escrito en inglés y la documentación en español.
- Las variables y funciones comienzan con letra minúscula. Cada palabra consecutiva en el nombre comienza con letra mayúscula.

3.5 Pruebas de software

Las pruebas son una actividad en la cual un software o uno de sus componentes se ejecutan en circunstancias previamente planificadas con el objetivo básico de encontrar errores, los resultados son observados y registrados, donde se evalúa algún aspecto del sistema o componente (34). Para que las pruebas tengan éxito es necesario realizar casos de pruebas y utilizar técnicas que nos guíen el proceso de

la prueba. En la presente investigación se utiliza para el diseño de casos de prueba el método de prueba de caja negra.

3.5.1 Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, entendiendo por interfaz las entradas y salidas de dicho software (35). Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.
Estas pruebas permiten encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Una vez terminada la fase de implementación de los componentes gráficos se procede a ejecutar las pruebas a cada uno de los casos de uso definidos en el Capítulo 2. Estas pruebas permitirán validar que se cumplan los requerimientos funcionales establecidos. Para garantizar la calidad y el correcto cumplimiento de los requisitos funcionales, se realizarán iteraciones de pruebas hasta que la cantidad de errores encontrados sea cero.

3.5.1.1 Técnica Partición de Equivalencia

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. Descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar (36).

3.5.2 Casos de Prueba

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada.

- ✓ V: válido
- ✓ I: inválido
- ✓ N/A: no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

A continuación se muestran los Diseños de Casos de Pruebas de los CU Gestionar camión cisterna, Gestionar bomba, Gestionar enfriador de crudo y gas, Gestionar válvula manual, Gestionar válvula automática, Gestionar válvula motorizada, Gestionar válvula de control por posicionamiento, el resto de los Diseños de Casos de Pruebas pueden encontrarse en el Anexo 2 del presente trabajo.

3.5.2.1 Diseño de Caso de Prueba para los CU descritos anteriormente.

Descripción general:

Se encarga de adicionar, modificar o eliminar un componente.

Condiciones de ejecución:

- ✓ El ambiente de configuración debe estar en ejecución.
- ✓ Debe estar abierto un despliegue de un proyecto determinado.

Tabla 6: Diseño de Caso de Prueba " Crear componente en el despliegue".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Crear componente en el despliegue.	Esc. 1.1: Componente adicionado satisfactoriamente.	Se arrastra y adiciona el componente gráfico en el despliegue.	1. Presiona el botón primario del ratón sobre el componente gráfico y lo arrastra hacia el despliegue. 2. Se muestra el despliegue donde se debe adicionar el componente.
	Esc. 1.2: Se adiciona incorrectamente.	En caso de que se arrastra el componente fuera del área del despliegue.	1. Presiona el botón primario del ratón sobre el componente gráfico y

			lo arrastra hacia el despliegue. 2. Se muestra el despliegue donde se debe adicionar el componente. 2.1 Si no se ha arrastrado para el área del despliegue, no se muestra el componente adicionado.

Diseño de caso de prueba: Crear componente en el despliegue.

Id de la sección	Escenario	Variable	Respuesta del sistema	Resultado de la prueba
SC 1	Componente adicionado satisfactoriamente	N/A	Se adiciona el componente en el despliegue.	Prueba satisfactoria
	Se adiciona incorrectamente	N/A	No se adiciona el componente en el despliegue.	

Tabla 7: Diseño de Caso de Prueba "Definir posición del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 2: Definir posición del componente.	Esc. 1.1: Posición modificada correctamente.	Se selecciona y modifica la posición del componente.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del

			componente y presiona doble click sobre el campo posición. 3. Se muestran los campos X y Y donde se debe ingresar los nuevos valores y presiona el botón aceptar.
	Esc. 1.2: Campo vacío.	En caso de que se deje el campo vacío.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y presiona doble click sobre el campo posición. 3. Se muestran los campos X y Y donde se debe ingresar los nuevos valores y presiona el botón aceptar. 3.1 Si no se ha ingresado un nuevo valor y se deja el campo vacío, se muestra el valor anterior.

Diseño de caso de prueba: Definir posición del componente.

Id de la sección	Escenario	Variable 1 X	Respuesta del sistema	Resultado de la prueba
SC 2	Posición modificada correctamente	V	Se modifica la posición del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica la posición del componente, se muestra el valor anterior.	

Id de la sección	Escenario	Variable 2 Y	Respuesta del sistema	Resultado de la prueba
SC 2	Posición modificada correctamente	V	Se modifica la posición del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica la posición del componente, se muestra el valor anterior.	

Tabla 8: Diseño de Caso de Prueba "Definir mediciones del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 3: Definir mediciones del componente.	Esc. 1.1: Mediciones establecidas correctamente.	Se selecciona el componente y establecen las mediciones.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del

			componente y selecciona el campo mediciones. 3. Se muestra una ventana donde se deben llenar los campos requeridos y presiona el botón aceptar.
	Esc. 1.2: Se establecen las mediciones incorrectamente.	En caso de que no se seleccionan los datos requeridos.	1. Selecciona el componente deseado y presiona el inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo mediciones. 3. Se muestra una ventana donde se deben llenar los campos requeridos y presiona el botón aceptar. 3.1 Si no se ha seleccionado un recurso, se muestra el mensaje ¨No ha seleccionado ningún recurso¨.

Diseño de caso de prueba: Definir mediciones del componente.

Id de la sección	Escenario	Variable 1 punto	Respuesta del sistema	Resultado de la prueba

SC 3	Mediciones establecidas correctamente	V	Se establecen las mediciones del componente y se añade una animación por defecto.	Prueba satisfactoria
	Se establecen las mediciones incorrectamente.	I	No se establecen las mediciones del componente, se muestra el mensaje ¨No ha seleccionado ningún recurso¨.	

Tabla 9: Diseño de Caso de Prueba "Modificar nombre del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 4: Modificar nombre del componente.	Esc. 1.1: Nombre modificado correctamente.	Se selecciona y modifica el nombre del componente.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo nombre. 3. Se muestra el campo nombre donde se debe ingresar el nuevo nombre y presiona el botón aceptar.
	Esc. 1.2: Campo vacío.	En caso de que se deje el campo vacío.	1. Selecciona el componente deseado y

			presiona el inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo nombre. 3. Se muestra el campo nombre donde se debe ingresar el nuevo nombre y presiona el botón aceptar. 3.1 Si no se ha ingresado un nuevo nombre y se deja el campo vacío, se muestra el nombre anterior.

Diseño de caso de prueba: Modificar nombre del componente.

Id de la sección	Escenario	Variable 1 nombre	Respuesta del sistema	Resultado de la prueba
SC 4	Nombre modificado correctamente	V	Se modifica el nombre del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica el nombre del componente, se muestra el nombre anterior.	

Tabla 10: Diseño de Caso de Prueba "Definir animaciones sobre el componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 5: Definir animaciones sobre el componente.	Esc. 1.1: Animaciones adicionadas correctamente.	Se selecciona y adicionan las animaciones del componente.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo animaciones. 3. Se muestra el campo umbrales y se selecciona dicho campo. 4. Se muestra una ventana donde se deben llenar los campos requeridos y presiona el botón aceptar.
	Esc. 1.2: Animaciones adicionadas incorrectamente.	En caso de que no se seleccionan los datos requeridos.	1. Selecciona el componente deseado y presiona el inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo animaciones. 3. Se muestra el campo umbrales y se

			selecciona dicho campo. 4. Se muestra una ventana donde se deben llenar los campos requeridos y presiona el botón aceptar. 4.1 Si no se ha creado al menos una condición, se muestra el mensaje ¨La colección de condiciones está vacía. Se debe crear al menos una condición para adicionar un umbral¨. 4.2 Si no se ha asociado a un punto, se muestra el mensaje ¨Condición sin medición asociada. No puede quedar una condición vacía¨.

Diseño de caso de prueba: Definir animaciones sobre el componente.

Id de la sección	Escenario	Variable 1 nombre de la animación	Variable 2 punto	Respuesta del sistema	Resultado de la prueba
SC 5	Animaciones adicionadas correctamente	V	V	Se adicionan las animaciones del componente	Prueba satisfactoria

	Animaciones adicionadas incorrectamente	I	V	No se adicionan las animaciones del componente, se muestra el mensaje ¨La colección de condiciones está vacía. Se debe crear al menos una condición para adicionar un umbral¨.	
		V	I	No se adicionan las animaciones del componente, se muestra el mensaje ¨Condición sin medición asociada. No puede quedar una condición vacía¨.	
		I	I	No se adicionan las animaciones del componente	

Tabla 11: Diseño de Caso de Prueba "Modificar descripción del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 6: Modificar descripción del componente.	Esc. 1.1: Descripción modificada correctamente.	Se selecciona y modifica la descripción del componente.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo descripción. 3. Se muestra el campo descripción donde se debe ingresar la nueva descripción y presiona

			el botón aceptar.
	Esc. 1.2: Campo vacío.	En caso de que se deje el campo vacío.	1. Selecciona el componente deseado y presiona el inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo descripción. 3. Se muestra el campo descripción donde se debe ingresar la nueva descripción y presiona el botón aceptar. 3.1 Si no se ha ingresado una nueva descripción y se deja el campo vacío, se muestra la descripción anterior.

Diseño de caso de prueba: Modificar descripción del componente.

Id de la sección	Escenario	Variable 1 descripción	Respuesta del sistema	Resultado de la prueba
SC 6	Descripción modificada correctamente	V	Se modifica la descripción del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica la descripción del componente, se muestra la	

			descripción anterior.	

Tabla 12: Diseño de Caso de Prueba "Definir color del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 7: Definir color del componente.	Esc. 1.1: Color modificado correctamente.	Se selecciona y modifica el color del componente.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo fondo con doble click. 3. Se muestra el campo color y se selecciona. 4. Se muestra una ventana con una paleta de colores, donde se debe modificar el color y presiona el botón OK.
	Esc. 1.2: Campos vacíos.	En caso de que se deje un campo vacío.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo

			fondo con doble click. 3. Se muestra el campo color y se selecciona. 4. Se muestra una ventana con una paleta de colores, donde se debe modificar el color y presiona el botón OK. 4.1 Si no se ha ingresado un nuevo valor y se deja el campo vacío, se muestra el color anterior.

Diseño de caso de prueba: Definir color del componente.

Id de la sección	Escenario	Variable 1 color	Respuesta del sistema	Resultado de la prueba
SC 7	Color modificado correctamente	V	Se modifica el color del componente	Prueba satisfactoria
	Campos vacíos.	-	No se modifica el color del componente, se muestra el color anterior.	

Tabla 13: Diseño de Caso de Prueba "Definir tamaño del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 8: Definir tamaño del componente.	Esc. 1.1: Tamaño modificado	Se selecciona y modifica el tamaño del componente.	1. Selecciona el componente deseado y

	correctamente.		presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y los campos ancho y alto. 3. Selecciona el campo donde se debe ingresar los nuevos valores y presiona el botón aceptar.
	Esc. 1.2: Campo vacío.	En caso de que se deje un campo vacío.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y los campos ancho y alto. 3. Selecciona el campo donde se debe ingresar los nuevos valores y presiona el botón aceptar. 3.1 Si no se ha ingresado un nuevo valor y se deja el campo vacío, se muestra el valor anterior.

Diseño de caso de prueba: Definir tamaño del componente.

Id de la sección	Escenario	Variable 1 ancho	Respuesta del sistema	Resultado de la prueba
SC 8	Tamaño modificado correctamente.	V	Se modifica el tamaño del componente.	Prueba satisfactoria
	Campo vacío.	-	No se modifica el tamaño del componente, se muestra el valor anterior.	

Id de la sección	Escenario	Variable 2 alto	Respuesta del sistema	Resultado de la prueba
SC 8	Tamaño modificado correctamente	V	Se modifica el tamaño del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica el tamaño del componente, se muestra el valor anterior.	

Tabla 14: Diseño de Caso de Prueba "Establecer rotación del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 9: Establecer rotación del componente.	Esc. 1.1: Rotación establecida correctamente	Se rota el componente siempre por el centro	1. Presiona el botón secundario del ratón sobre el componente gráfico y selecciona la opción de rotación deseada. 2. Se muestra el despliegue donde se debe rotar el

			componente.
	Esc. 1.2: Rotación establecida incorrectamente	En caso de que no se seleccionan las opciones para rotar el componente	1. Presiona el botón secundario del ratón sobre el componente gráfico y selecciona la opción de rotación deseada. 2. Se muestra el despliegue donde se debe rotar el componente. 2.1 Si no se selecciona una de las opciones de rotación, se muestra el componente con la misma rotación.

Diseño de caso de prueba: Establecer rotación del componente.

Id de la sección	Escenario	Variable	Respuesta del sistema	Resultado de la prueba
SC 9	Rotación establecida correctamente	N/A	Se rota el componente en el despliegue	Prueba satisfactoria
	Rotación establecida incorrectamente	N/A	No se rota el componente en el despliegue	

Tabla 15: Diseño de Caso de Prueba "Definir opacidad del componente".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 10: Definir	Esc. 1.1: Opacidad	Se selecciona y modifica la	1. Selecciona el

opacidad del componente.	modificada correctamente.	opacidad del componente.	componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo opacidad. 3. Se muestra el campo opacidad donde se debe ingresar el nuevo valor y presiona el botón aceptar.
	Esc. 1.2: Campo vacío.	En caso de que se deje un campo vacío.	1. Selecciona el componente deseado y presiona el botón inspector de propiedades. 2. Muestra las propiedades del componente y selecciona el campo opacidad. 3. Se muestra el campo opacidad donde se debe ingresar el nuevo valor y presiona el botón aceptar. 3.1 Si no se ha ingresado un nuevo valor y se deja el campo

			vacío, se muestra el valor de la opacidad anterior.

Diseño de caso de prueba: Definir opacidad del componente.

Id de la sección	Escenario	Variable 1 descripción	Respuesta del sistema	Resultado de la prueba
SC 10	Opacidad modificada correctamente	V	Se modifica la opacidad del componente	Prueba satisfactoria
	Campo vacío.	-	No se modifica la opacidad del componente, se muestra el valor de la opacidad anterior.	

Tabla 16: Diseño de Caso de Prueba "Eliminar componente del despliegue".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 11: Eliminar componente del despliegue.	Esc. 1.1: Componente eliminado satisfactoriamente.	Se elimina el componente gráfico del despliegue.	1. Selecciona en el área de edición el componente gráfico a eliminar. 2. Presiona click derecho sobre el componente y selecciona la opción ¨Eliminar¨.
	Esc. 1.2: Se elimina incorrectamente.	En caso de que no se selecciona la opción correcta.	1. Selecciona en el área de edición el componente gráfico a eliminar.

			2. Presiona click derecho sobre el componente y selecciona la opción ¨Eliminar¨. 2.1 Si no se ha seleccionado la opción ¨Eliminar¨, no se elimina el componente.

Diseño de caso de prueba: Eliminar componente del despliegue.

Id de la sección	Escenario	Variable	Respuesta del sistema	Resultado de la prueba
SC 11	Componente eliminado satisfactoriamente	N/A	Se elimina el componente en el despliegue.	Prueba satisfactoria
	Se elimina incorrectamente	N/A	No se elimina el componente en el despliegue.	

3.6 Resultados de las pruebas

Se realizaron 13 casos de prueba para validar la aplicación, con diferentes datos de entradas para la ejecución de los mismos, identificándose resultados satisfactorios e insatisfactorios, evaluando las funcionalidades en cada una de las iteraciones. Los resultados insatisfactorios fueron corregidos en el final de la iteración correspondiente, para de esta manera poder pasar a la próxima iteración y que el sistema evolucionara de manera estable. Se comienza por la primera iteración encontrando 6 no conformidades relacionadas con errores de validación de datos de 13 requisitos funcionales. Luego se prosigue a una segunda iteración disminuyendo en esta a 2 relacionadas con errores de interfaz. Posteriormente se ejecuta la tercera y última iteración quedando el sistema libre de no conformidades. Luego de comprobadas cada una de las funcionalidades se procedió a corroborar que el módulo HMI pudiera utilizar los componentes gráficos creados en la paleta de componentes. Para ello se creó un nuevo proyecto en el

ambiente de configuración, se crearon varios componentes con las configuraciones necesitadas y se inició el proceso de visualización en el Runtime. Realizadas estas acciones se pudo observar la actualización de los valores mostrados en los componentes y sus animaciones lo más cercano al tiempo real, demostrando así que la paleta de componentes desarrollada da cumplimiento a la problemática planteada en este trabajo. A continuación se muestra la **Figura #12** con el análisis de las no conformidades encontradas en cada iteración de las pruebas realizas al sistema.

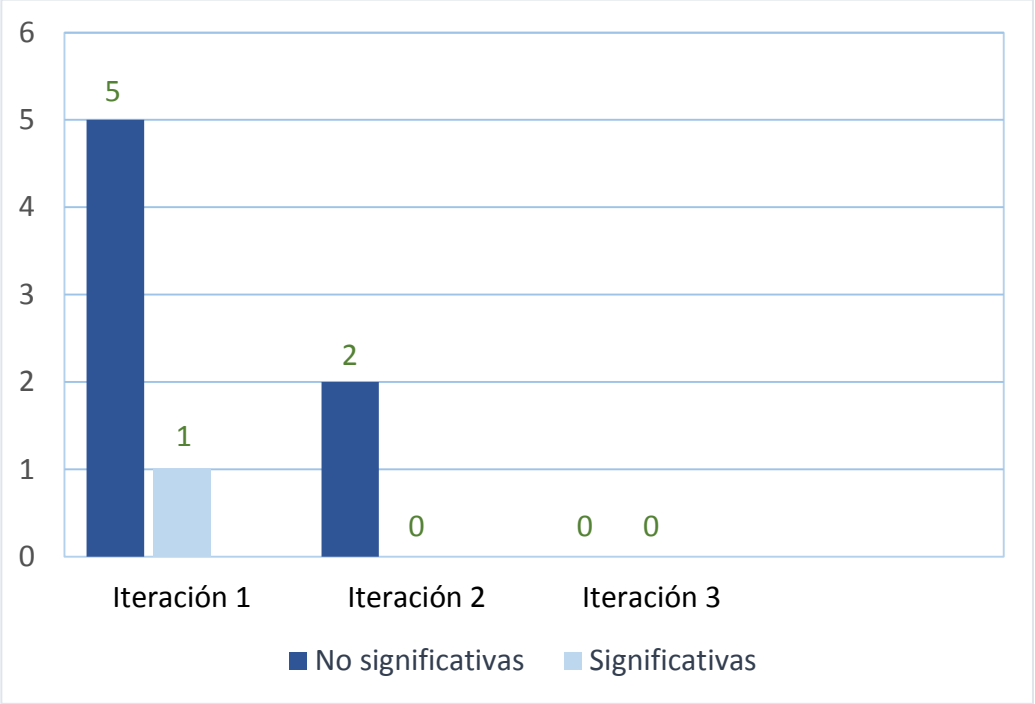


Figura 12: No conformidades

3.7 Conclusiones parciales

Durante el desarrollo de este capítulo se expusieron los principales artefactos del modelo de implementación, lo que permitió mostrar los componentes del sistema y sus relaciones a través del diagrama de componentes. Mediante el diagrama de despliegue se modeló la arquitectura en tiempo de ejecución y se mostró los vínculos de comunicación entre los nodos. Las pruebas de caja negra realizadas sobre la solución propuesta lograron corregir las deficiencias funcionales detectadas en el sistema. Se implementaron los diferentes componentes gráficos de la propuesta de solución lo cual permitirán a los operadores disminuir la creación de dichos componentes en el módulo HMI del SCADA GALBA. Se integró la paleta de componentes gráficos en el módulo HMI lo cual permitió dar solución al problema de la investigación planteado al principio del presente trabajo.

CONCLUSIONES GENERALES

Luego de culminada la investigación cuyo producto lo constituye la paleta de componentes gráficos, se llegan a las siguientes conclusiones:

- ✓ El estudio del estado del arte y las principales soluciones existentes de los componentes gráficos en los diferentes SCADA, permitieron sentar las bases teóricas para la futura implementación de la solución propuesta.
- ✓ Se definieron los requerimientos funcionales y no funcionales de la solución propuesta, lo cual permitió establecer las pautas para el desarrollo de la implementación del sistema.
- ✓ Las herramientas y tecnologías que se escogieron para el desarrollo de la solución propuesta son de código abierto y multiplataforma, esta selección permitió que la propuesta de solución fuera fácil de integrar con el ambiente de configuración del módulo HMI y que cumpliera con las políticas de migración al software libre establecidas en el país.
- ✓ El uso de la metodología de desarrollo AUP generó una gran documentación que sirve de guía para proyectos que tengan similitudes con los componentes gráficos realizados.
- ✓ Las pruebas de caja negra realizadas sobre la solución propuesta lograron corregir las deficiencias funcionales detectadas en el sistema.
- ✓ Se implementaron e integraron los diferentes componentes gráficos de la propuesta de solución lo cual permitirán a los operadores disminuir la creación de dichos componentes en el módulo HMI del SCADA GALBA.

RECOMENDACIONES

- ✓ Evaluar la posibilidad de prescindir del componente medidor vertical de la clase Vtermometer y utilizar el formato SVG para simular el nivel del camión cisterna.

REFERENCIAS BIBLIOGRÁFICAS

1. Slideshare. [En línea] 2013. [Citado el: 23 de Diciembre de 2014.] http://www.slideshare.net/nestorcusco/sistema-scada-24902242.

2. ADR Formación. [En línea] [Citado el: 23 de Diciembre de 2014.] http://www.adrformacion.com/cursos/scadawin/leccion1/tutorial1.html.

3. **Hernández Cevallos, María Isabel y Ledesma Marcalla, Denis Alejandro.** *Desarrollo de un sistema SCADA para la medición de voltajes con sistemas embebidos para el laboratorio de Mecatrónica de la Facultad de Mecánica.* 2010.

4. EcuRed. [En línea] 2011. [Citado el: 8 de Enero de 2015.] http://www.ecured.cu/index.php/Sistema_SCADA.

5. Electro Industria. [En línea] Noviembre de 2007. [Citado el: 24 de Diciembre de 2014.] http://www.emb.cl/electroindustria/articulo.mvc?xid=837.

6. COPADATA. [En línea] 2013. [Citado el: 24 de Diciembre de 2014.] http://www.copadata.com/es/productos/product-features/interfaz-hombre-maquina-hmi.html.

7. **Movicon 11.** [En línea] 2010. [Citado el: 20 de Diciembre de 2014.] http://www.progea.com/downloads/Inf_Es_Movicon11.pdf.

8. **Siemens AG.** *SIMATIC WinCC.* 2008.

9. **Aragón Caceres, José Antonio y Llanes Jiménez, Beatriz.** *Servicio de Integración con Terceros para el Acceso a Variables del sistema SCADA Guardián del ALBA.* 2009.

10. **Villareal, Elizabeth y Raed, Charrouf.** *Desarrollo SCADA Guardián del ALBA - Especificación de Eventos.* 2011.

11. **Hernández, Judeli.** *Implementación de objetos gráficos para el desarrollo de despliegues operacionales en el sector comercio y suministro del SCADA Nacional del PDVSA.* Mérida, Venezuela : s.n., 2008.

12. EcuRed. [En línea] [Citado el: 20 de Marzo de 2015.] http://www.ecured.cu/index.php/Scalable_Vector_Graphics#Caracter.C3.ADsticas.

13. Observatorio tecnológico. [En línea] [Citado el: 22 de Enero de 2015.] http://recursostic.educacion.es/observatorio/web/es/software/software-general/325-inkscape.

14. **Osorio Cintra, Yordano Yunior.** Sistema para la Gestión del Banco de Problemas de la UCI. [En línea] Septiembre de 2013. [Citado el: 8 de Enero de 2015.] http://semanatecnologica.fordes.co.cu/sites/default/files/public/p61_0.pdf.

15. **Torres, Jesús.** Qt Framework. [En línea] Enero de 2013. [Citado el: 9 de Enero de 2015.] http://jmtorres.webs.ull.es/me/2013/01/proyecto-qt-framework-de-desarrollo-de-aplicaciones/.

16. IDE de Programación. [En línea] [Citado el: 9 de Enero de 2015.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.

17. Qt Creator. [En línea] [Citado el: 10 de Enero de 2015.] http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado.

18. Qué es C++. [En línea] [Citado el: 13 de Enero de 2015.] http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Introducci%C3%B3n.

19. **Padrón Avila, Mario Ernesto.** *Herramienta de pruebas de rendimiento a los servicios de obtención de variables, alarmas y eventos del Servidor de Comunicación con Terceros del SCADA "Guardián del ALBA".* Junio, 2013.

20. Descripción de la herramienta. Visual Paradigm. [En línea] [Citado el: 13 de Enero de 2015.] http://www.ie.inf.uc3m.es/grupo/docencia/reglada/Is1y2/PracticaVP.pdf.

21. Metodología de Desarrollo de Software. [En línea] [Citado el: 14 de Enero de 2015.] http://es.scribd.com/doc/12983329/Metodologia-de-Desarrollo-de-Software#scribd.

22. **Rodríguez Sánchez, Tamara.** *Metodología de desarrollo para la actividad productiva de la UCI.* Noviembre, 2014.

23. Modelo de dominio. [En línea] [Citado el: 5 de Febrero de 2015.] http://issuu.com/hmorenop/docs/9_modelo_de_dominio.

24. Requisitos No-Funcionales. [En línea] [Citado el: 7 de Febrero de 2015.] http://www.softqanetwork.com/requisitos-no-funcionales-nfr.

25. Diagrama de Clases. [En línea] [Citado el: 7 de Febrero de 2015.] http://www.ecured.cu/index.php/Diagrama_de_Clase#Diagrama_de_Clases.

26. [En línea] Abril de 2004. [Citado el: 20 de Febrero de 2015.] http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf.

27. *Qt Documentation.* 2012.

28. **Nicolau González, Frank Enrique.** *Interfaz Hombre-Máquina para el cliente OPC DA Almiquí.* 2013.

29. Developer Network. [En línea] [Citado el: 2015 de Enero de 27.] https://msdn.microsoft.com/es-es/library/bb972242.aspx.

30. **Gamma, E., y otros.** *Design Patterns. Elements of Reusable Object-Oriented . s.l. : Addison Wesley.* 1994.

31. **E. Gamma, R. Helm, R. Johnson and J. Vlissides.** *Design Patterns: elements of reusable object-oriented software.* 1994.

32. Diagramas UML: Componentes y despliegue. [En línea] 26 de Septiembre de 2010. [Citado el: 11 de Febrero de 2015.] http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue.

33. [En línea] [Citado el: 5 de Febrero de 2015.] https://sistemas.uniandes.edu.co/~miso4203/dokuwiki/lib/exe/fetch.php?media=principal:temas:estimacionprobe.pdf.

34. [En línea] [Citado el: 4 de Marzo de 2015.] http://analidiseorienobjet.wikispaces.com/file/view/pruebas+de+software.pdf.

35. [En línea] [Citado el: 7 de Marzo de 2015.] http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI444/Pruebas.pdf.

36. EcuRed. [En línea] [Citado el: 15 de Marzo de 2015.] http://www.ecured.cu/index.php/Pruebas_de_caja_negra.