



Facultad 5

“Desarrollo de paleta de componentes gráficos del proceso de producción azucarera en el SCADA SAINUX.”

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autores

Alejandro Antonio Matamoros Vargas.

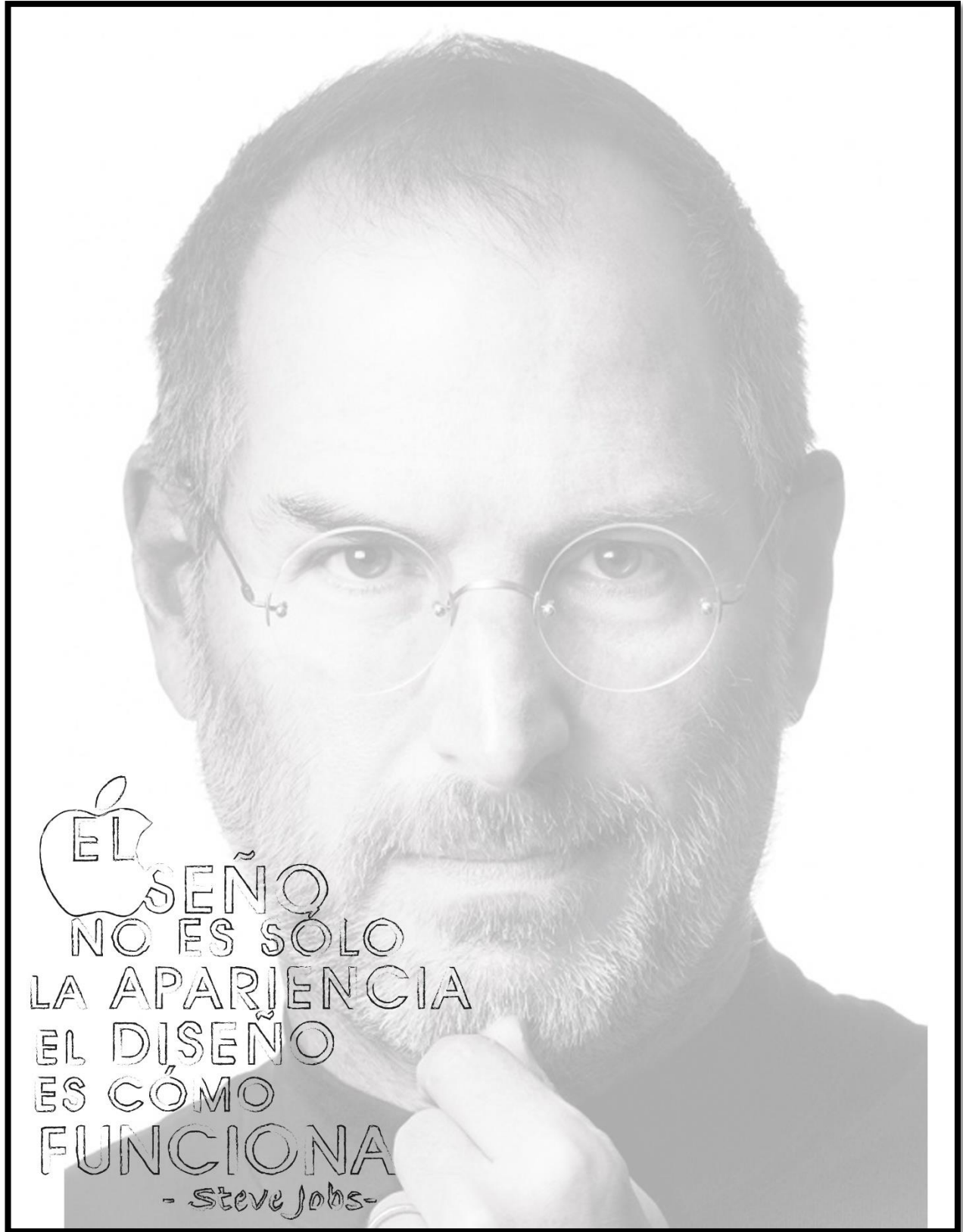
Tutores

Ing. Henry Marcelo Cabrera Robles.

Ing. Yordanis Bridón Danger.

La Habana, Cuba

Junio de 2016



EL
DISEÑO
NO ES SOLO
LA APARIENCIA
EL DISEÑO
ES CÓMO
FUNCIONA

- Steve Jobs -

Declaración de Autoría

Declaro que soy el único autor de la presente tesis y autorizo al Centro de Informática Industrial de 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 _____

Alejandro Antonio Matamoros Vargas

Firma del Autor

Ing. Henry Marcelo Cabrera Robles

Firma del Tutor

Ing. Yordanis Bridón Danger

Firma del Tutor

Datos de Contacto

Datos del Autor

Alejandro Antonio Matamoros Vargas

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: amatamoros@estudiantes.uci.cu

Datos del Tutor

Ing. Henry Marcelo Cabrera Robles

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: hmcrabrera@uci.cu

Datos del Tutor

Ing. Yordanis Bridón Danger

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: ybridon@uci.cu

Resumen

El controlar y supervisar procesos industriales a distancia se hace cada vez más necesario en la medida en que las tecnologías cambian y aumentan sus niveles de posicionamiento en el mercado internacional. En este sentido, los Sistemas de Supervisión, Control y Adquisición de Datos (SCADA) son los encargados de realizar estos procesos. En Cuba, el Centro de Informática Industrial (CEDIN) de la Universidad de las Ciencias Informáticas (UCI) está inmerso en el desarrollo de un Sistema de Automatización Industrial basado en GNU/LINUX (SAINUX) tema central al que tributa la investigación que se propone: “Desarrollo de paleta de componentes gráficos del proceso de producción azucarera en el SCADA SAINUX”. En tanto, hoy SAINUX no está orientado a un proceso industrial específico, los componentes actuales son los más genéricos, lo que el mantenedor utiliza otros componentes más básicos, como las imágenes, figuras simples y el nivel, trayendo consigo que al redimensionar la imagen deseada pierde calidad, afectando negativamente la representación gráfica de los elementos, además la utilización de figuras simples contribuye a realizar con mayor lentitud y un alto grado de complicación el proceso de configuración a supervisar, afectando en gran medida la usabilidad del sistema. Por consiguiente, la investigación se propone como objetivo desarrollar una solución integrada al SCADA SAINUX que permita elevar el grado de representación de la producción azucarera en el módulo de Interfaz Hombre Máquina (HMI) del SCADA SAINUX. Se utilizan los métodos teóricos, metodológicos y analíticos así como las herramientas y artefactos que complementan la estructura y organización de la propuesta investigativa. Los resultados avalados en la investigación se materializan a partir del desarrollo que posibilita elevar el grado de representación gráfica de manera rápida y organizada, eliminando los posibles errores subjetivos que se presentaban continuamente.

Palabras claves: Componentes gráficos, SAINUX, HMI, SCADA producción azucarera.

Índice de Contenido

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	13
1.1 Definiciones asociadas al proceso de producción azucarera	13
1.2 Sistemas de Supervisión, Control y Adquisición de Datos SCADA	16
1.2.1 Módulos del SCADA SAINUX	17
1.3 Análisis de sistemas existentes.....	19
1.3.1 Análisis de software nacional	19
1.4 Conclusiones Parciales.....	19
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN.....	21
2.1 Tecnologías y herramientas para la implementación.....	21
2.1.1 Metodología utilizada	22
2.1.2 Lenguaje de modelado.....	24
2.1.3 Herramienta CASE.....	24
2.1.4 Lenguaje de programación.....	24
2.1.5 Marco de trabajo	25
2.1.6 Entorno de desarrollo	25
2.1.7 Gráficos Vectoriales Escalables (SVG)	25
2.1.8 Editor Gráfico	26
2.1.9 Sistema Operativo.....	26
2.2 Propuesta de solución	27
2.2.1 Modelo de dominio.....	27
2.2.2 Descripción de clases del modelo de dominio	28
2.3 Requisitos del proceso de producción azucarera	29
2.3.1 Requisitos funcionales.....	29
2.3.2 Requisitos no funcionales.....	32
2.4 Descripción del sistema propuesto.....	33

Índice de Contenido

2.5	Planificación del desarrollo.....	34
2.5.1	Estimación de esfuerzos e iteraciones por historia de usuario.....	34
2.5.2	Plan de iteraciones.....	37
2.6	Conclusiones del capítulo.....	37
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....		38
3.1	Estilos arquitectónicos utilizados.....	38
3.2	Patrones de diseño.....	39
3.3	Modelo de diseño.....	41
3.3.1	Diagrama de secuencia.....	41
3.3.2	Diagrama de clases del diseño.....	42
3.4	Modelo de implementación.....	44
3.4.1	Diagrama de componentes.....	45
3.4.2	Estándar de codificación.....	46
3.5	Diagrama de Despliegue.....	47
3.5.1	Descripción del diagrama de despliegue.....	48
3.6	Pruebas.....	48
3.6.1	Pruebas de aceptación.....	49
3.7	Conclusiones del capítulo.....	51
CONCLUSIONES.....		52
Trabajos citados.....		53
Bibliografía.....		54

Índice de Tablas

<i>Tabla 1. Disciplinas de AUP-UCI.....</i>	<i>24</i>
<i>Tabla 2. Historia de Usuario Insertar Tacho.</i>	<i>32</i>
<i>Tabla 3. Estimación de esfuerzos.....</i>	<i>37</i>
<i>Tabla 4. Diagrama de secuencia de componentes gráficos.....</i>	<i>42</i>
<i>Tabla 5. Caso de Prueba 34.....</i>	<i>50</i>
<i>Tabla 6. Resultados de las pruebas.</i>	<i>50</i>
<i>Tabla 7. Historia de Usuario Insertar desfibradora.</i>	<i>56</i>
<i>Tabla 8. Historia de Usuario Eliminar desfibradora.....</i>	<i>57</i>
<i>Tabla 9. Historia de Usuario Configurar propiedades de desfibradora.....</i>	<i>57</i>
<i>Tabla 25. Historia de Usuario calentador doble.</i>	<i>58</i>
<i>Tabla 42. Historia de Usuario Insertar bomba de tornillo.</i>	<i>58</i>
<i>Tabla 52. Historia de Usuario Eliminar torre A90.....</i>	<i>59</i>
<i>Tabla 53. Historia de Usuario Configurar propiedades de torre A90.....</i>	<i>59</i>
<i>Tabla 54. Historia de Usuario Insertar torre B90.....</i>	<i>60</i>
<i>Tabla 55. Historia de Usuario Eliminar Insertar torre B90.....</i>	<i>60</i>
<i>Tabla 56. Historia de Usuario Configurar propiedades de torre B90.....</i>	<i>61</i>
<i>Tabla 57. Historia de Usuario Insertar torre C90.</i>	<i>62</i>
<i>Tabla 58. Historia de Usuario Modificar torre C90.</i>	<i>62</i>
<i>Tabla 59. Historia de Usuario Configurar propiedades de torre C90.....</i>	<i>63</i>
<i>Tabla 60. Prueba de aceptación Insertar desfibradora.</i>	<i>63</i>
<i>Tabla 61. Prueba de aceptación Eliminar desfibradora.</i>	<i>64</i>
<i>Tabla 62. Prueba de aceptación Configurar propiedades de desfibradora.</i>	<i>64</i>
<i>Tabla 81. Prueba de aceptación Insertar calentador triple.....</i>	<i>65</i>
<i>Tabla 82. Prueba de aceptación Eliminar calentador triple.....</i>	<i>66</i>
<i>Tabla 83. Prueba de aceptación Configurar propiedades de calentador triple.....</i>	<i>66</i>
<i>Tabla 84. Prueba de aceptación Insertar calentador quintuple.....</i>	<i>67</i>
<i>Tabla 85. Prueba de aceptación Eliminar calentador quintuple.....</i>	<i>68</i>
<i>Tabla 86. Prueba de aceptación Configurar propiedades de calentador quintuple.....</i>	<i>68</i>
<i>Tabla 89. Prueba de aceptación Configurar propiedades de clarificador.....</i>	<i>69</i>
<i>Tabla 90. Prueba de aceptación Insertar tanque clarificador.....</i>	<i>70</i>
<i>Tabla 91. Prueba de aceptación Eliminar tanque clarificador.....</i>	<i>70</i>
<i>Tabla 92. Prueba de aceptación Configurar propiedades de tanque clarificador.....</i>	<i>71</i>
<i>Tabla 93. Prueba de aceptación Insertar tacho.....</i>	<i>72</i>

Índice de Tablas

<i>Tabla 94 Prueba de aceptación Eliminar tachó.</i>	72
<i>Tabla 95. Prueba de aceptación Configurar propiedades de tachó.</i>	73
<i>Tabla 102. Prueba de aceptación Insertar tanque mezclador.</i>	74
<i>Tabla 103. Prueba de aceptación Eliminar tanque mezclador.</i>	74
<i>Tabla 104. Prueba de aceptación Configurar propiedades de tanque mezclador.</i>	75
<i>Tabla 105. Prueba de aceptación insertar torre A90.</i>	76

Índice de Figuras

<i>Figura 1. Etapas y equipos del proceso de producción de la caña.</i>	15
<i>Figura 2. Esquema básico de un sistema de Adquisición, supervisión y control.</i>	17
<i>Figura 3. Diagrama de clases del modelo de dominio.</i>	27
<i>Figura 4: Patrón de Arquitectura de Software Modelo-Vista.</i>	39
<i>Figura 5. Diagrama de paquetes.</i>	43
<i>Figura 6. Diagrama de Clases de Componentes Gráficos.</i>	44
<i>Figura 7. Diagrama de componentes.</i>	46
<i>Figura 8. Diagrama de despliegue.</i>	48
<i>Figura 9. Incidencias detectadas en el sistema.</i>	51
<i>Figura 10. Diagrama de secuencias Insertar clarificador.</i>	76
<i>Figura 11. Diagrama de secuencias Eliminar clarificador.</i>	77
<i>Figura 12. Diagrama de secuencias Eliminar molino.</i>	77
<i>Figura 13. Diagrama de secuencias Configurar propiedades de clarificador.</i>	77

INTRODUCCIÓN

La industria azucarera es una de las actividades económicas fundamentales que se manifiestan en la actualidad siendo la azúcar un producto básico, esencial y necesario en la dieta alimenticia. El proceso azucarero con el objetivo de alcanzar la mayor calidad está constituido por siete etapas, Cultivo y porte, Molienda, Clarificación, Evaporadores, Cristalización, Separación y Refinación. Algunas de las operaciones contenidas dentro de estas etapas del proceso azucarero son: desmenuzar la caña con cuchillas rotatorias y una desfibradora antes de molerla para facilitar la extracción del jugo. Estas operaciones requieren ser informatizadas para garantizar la supervisión y el control de cada uno de estos procesos; siendo el grupo empresarial AZCUBA el organismo rector del renglón azucarero en nuestro país, el cual requiere de un control riguroso para dichos procesos.

Para facilitar el trabajo de los especialistas y ayudar a la toma de decisiones se han desarrollado nuevas técnicas para la supervisión y control en el campo de la industria azucarera, desarrollando sistemas capaces de monitorear y controlar estos procesos que se extienden sobre una gran distancia (kilómetros / millas), garantizando la supervisión, el control y adquisición de datos centralizados; estos software se le denominan SCADA (Supervisión, Control y Adquisición de Datos). La instalación de un sistema SCADA necesita un hardware de señal de entrada y salida, sensores y actuadores, controladores, HMI, redes, comunicaciones, base de datos entre otros elementos. Cuba no se ha quedado al margen del desarrollo en la industria azucarera y ha desarrollado software que se especializan en la automatización de estos procesos como el sistema EROS. EROS es un software nacional que se encarga de trabajar acoplado a diversos sistemas de colección de datos, como elemento único o formando parte de una red industrial. EROS presenta varias desventajas este opera sobre Windows sistema operativo privado, además no cuenta con una amplia biblioteca de componente en la rama azucarera. Debido a esta dificultad los ingenieros del grupo empresarial AZCUBA deben diseñar y programar cada uno de los componentes necesario para un determinado despliegue mediante el uso de aplicaciones privadas como AUTOCAD y ADOBE PHOTOSHOP, trayendo como consecuencia un proceso largo y costoso. Con el objetivo de suplir las necesidades del sector empresarial surge el SAINUX, es un sistema SCADA distribuido, orientado a componentes y en proceso de desarrollo por especialistas CEDIN de la facultad 5 de UCI.

En la actualidad el sistema SAINUX no está orientado a un proceso industrial específico, los componentes actuales son los más genéricos posible para lograr un mayor alcance a la representación de procesos

Capítulo 1: Fundamentación teórica

industriales. La visión de SAINUX es garantizar paletas de componentes especializados por procesos industriales para poder reflejar lo más real posible un proceso industrial. Para solventar dicha situación en la actualidad el mantenedor utiliza otros componentes más básicos, como las imágenes, figuras geométricas y el nivel, trayendo consigo algunas desventajas:

- Al redimensionar la imagen deseada pierde calidad, afectando negativamente la representación gráfica de los elementos.
- La representación de figuras simples contribuye a realizar con mayor lentitud y un alto grado de complicación el proceso de configuración a supervisar, afectando en gran medida la usabilidad del sistema.

A partir de lo planteado anteriormente se ha determinado como **problema de investigación**: ¿Cómo contribuir en la especialización del módulo HMI del SCADA SAINUX para la producción azucarera?

El presente trabajo tiene como **objeto de estudio**: Representación de la producción azucarera en sistemas SCADA, definiendo como **campo de acción**: Representación de la producción azucarera en el sistema SCADA SAINUX.

Se define como **objetivo general**: Desarrollar una solución integrada al SCADA SAINUX que permita elevar el grado de representación de la producción azucarera en el módulo de HMI del SCADA SAINUX.

Para dar cumplimiento al objetivo general se definieron los siguientes **objetivos específicos**:

1. Caracterizar y evaluar los procesos de producción azucarera.
2. Diseñar y modelar el componente de representación gráfica a partir del módulo HMI.
3. Desarrollar el código fuente a partir del modelo arquitectónico obtenido.
4. Realizar pruebas de implementación al componente de representación gráfica.

Para cumplir el objetivo propuesto se definieron las siguientes **tareas de investigación**:

1. Elaboración del marco teórico de la investigación a través de la actualización del estado del arte que existe actualmente sobre el tema.
2. Identificación y caracterización de los principales elementos que están presentes en el proceso de producción azucarera.
3. Análisis y caracterización de las herramientas, tecnologías y metodología a utilizar en el desarrollo del componente.
4. Realización del levantamiento de requisitos funcionales y no funcionales.
5. Definición de la arquitectura para dar solución al desarrollo del componente.

Capítulo 1: Fundamentación teórica

6. Diseño del componente para obtener un modelo lógico del mismo.
7. Implementación de componentes gráficos brinde solución al problema planteado.
8. Realización de pruebas para validar el cumplimiento de los requerimientos.
9. Recopilación bibliográfica.

Para desarrollar estas tareas se utilizan los siguientes **Métodos Teóricos y Empíricos**:

Métodos Teóricos

1. **Analítico - Sintético**: Se utilizó en el análisis de módulo de HMI en el SCADA.
2. **Análisis Histórico – Lógico**: Se utilizó en la evaluación de los antecedentes, la evolución y el desarrollo que han tenido los sistemas SCADA, así como valorar las tendencias actuales.
3. **Modelación**: Se utilizó en la modelación de los diagramas de clases del análisis y del diseño, lo que permitió una mejor interpretación del componente.

Métodos Empíricos

1. **Método de Entrevista**: Se utilizó para realizar entrevistas a especialistas con el fin de obtener información relacionada al módulo de HMI en el SCADA, definir el problema a resolver y los procesos que se llevan a cabo en la presente investigación.

La presente investigación está estructurada por 3 capítulos:

Capítulo 1: Fundamentación Teórica. Se indican las bases teóricas fundamentales relacionadas con el dominio del problema, conceptos y técnicas del proceso de producción azucarera que serán utilizadas en el desarrollo del producto. Se caracterizan las principales herramientas existentes tanto el ámbito nacional e internacional y la posición del país ante el uso de una herramienta de esta magnitud.

Capítulo 2: Análisis y Diseño de la Solución. En este capítulo se definen las tecnologías y herramientas utilizadas en el marco de la presente investigación. Se realiza el modelado de negocio a través del modelo de dominio del proceso de producción azucarera. Se identifican los requisitos funcionales y no funcionales que debe poseer el componente para su correcto funcionamiento y cumplimiento de los objetivos propuestos, que a su vez son agrupados por los casos de usos del sistema.

Capítulo 3: Implementación y Prueba. En este capítulo se generan los artefactos relacionados con el diseño y la implementación del sistema generando los diagramas de clases del diseño, clases persistentes, entidad-relación, componente y despliegue. Estos artefactos implementan la solución

Capítulo 1: Fundamentación teórica

propuesta para la producción azucarera y luego se procede a realizar las pruebas utilizando técnicas específicas para comprobar el funcionamiento del sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se describirán los conceptos fundamentales para caracterizar y explicar los procesos asociados a la industria azucarera y los sistemas SCADA para dar solución al problema, así como el análisis y caracterización de las herramientas, tecnologías y metodología a utilizar en el desarrollo del componente. Se realizará un estado del arte a nivel nacional e internacional previendo el uso de los mismos y las ventajas y desventajas que aplican al problema en cuestión.

1.1 Definiciones asociadas al proceso de producción azucarera

Durante el proceso de producción de la caña de azúcar, existen una serie de etapas y equipos necesarios para que la misma adquiera la mayor calidad para luego ser exportada. Entre las principales etapas desde la perspectiva del control y monitoreo de los componentes son las siguientes: la **molienda** donde la caña es desmenuzada con cuchillas rotatorias y una desfibradora antes de molerla para facilitar la extracción del jugo que se hace pasándola en serie, entre los filtros, o mazas de seis molinos. Se utiliza agua en contracorriente para ayudar a la extracción que llega a 94 o 95% del azúcar contenida en la caña. El remanente queda en el bagazo residual que es utilizado como combustible en las calderas, así como materia prima para la fabricación de tableros de bagazo. La **clarificación**, consiste en calentar el jugo y decantarlo, dicha decantación se lleva a cabo en dos grandes clarificadores en los cuales las impurezas, en forma de barro, van al fondo y el jugo clarificado se extrae por la parte superior. El barro o cachaza, contiene todavía azúcar y requiere ser pasada por filtros rotativos al vacío de los cuales se recuperan una cantidad de jugo, que retorna al proceso y se retira una torta de cachaza que es devuelta al campo.

Otra de las etapas son los **evaporadores**, el jugo claro, pasa a los evaporadores en los cuales se elimina alrededor del 80% del agua contenida en el jugo, que con esta operación se convierte en madadura. Los evaporadores trabajan en múltiples efectos, que el vapor producido por la evaporación de agua en el primer efecto es utilizado para calentar el segundo y así sucesivamente, hasta llegar al quinto efecto que entrega sus vapores al condensador. El condensador es enfriado por agua en recirculación desde el estanque de enfriamiento. La **crystalización** es otra de las etapas, donde la meladura pasa a los tachos

Capítulo 1: *Fundamentación teórica*

donde continúa la evaporación de agua, lo que ocasiona la cristalización del azúcar. Una vez que se vaya eliminando el agua, llega un momento en el cual la azúcar disuelta en la meladura se deposita en forma de cristales de sacarosa. Los tachos trabajan con vacío para efectuar la evaporación a baja temperatura y evitar la caramelización del azúcar.

En la etapa de **separación** en los tachos se obtiene una masa denominada masa cocida, que es la mezcla de cristales de azúcar y miel. La separación se hace por centrifugación en las maquinas destinadas a dicha labor. De las centrifugas sale azúcar cruda y miel. La miel se retorna a los tachos para dos etapas adicionales de cristalización que termina con los conocimientos o melaza. El azúcar de tercera se utiliza como pie para la cristalización del segundo conocimiento y el azúcar de segunda para el conocimiento de primera. Y por último la etapa de **refinación** donde el azúcar de primera es refundida o redisuelta con agua; luego es aireado en un recipiente a presión y pasa a las clarificadoras donde las impurezas flotan y el licor clarificado es extraído por la parte inferior. El licor clarificado es pasado por los filtros de lecho profundo donde se eliminan el resto de las impurezas y de allí el filtrado es entregado a los tachos de refino. Igual que en los tachos de crudo, en estos tachos se elimina agua y se obtiene azúcar refinada cristalizada. La miel es retornada al conocimiento de crudo para mezclarse con la meladura y la azúcar húmeda de las centrifugas pasa a los secadores y de allí al envase. (1)

Capítulo 1: Fundamentación teórica

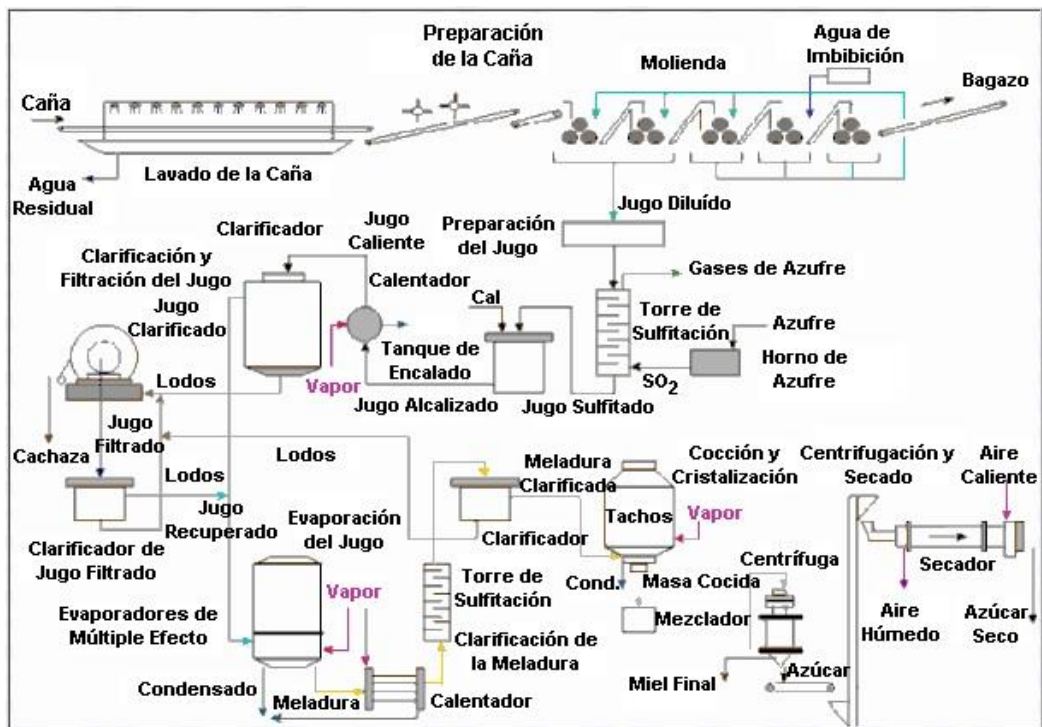


Figura 1. Etapas y equipos del proceso de producción de la caña.

Dentro de algunos de los procesos se utilizan una serie de elementos entre los que se encuentran:

Desfibradoras: Asegurar la alimentación a los molinos y preparar la caña, aplastándola para facilitar su toma y la extracción del jugo. Generalmente consta de dos cilindros, los cuales poseen una superficie diseñada para agarrar la caña, dicha superficie se debe construir de manera que rasgue y desfibre la caña. (1)

Molinos: Son cilindros horizontales cuyos centros forman un triángulo isósceles. Dos de ellas se encuentran a la misma altura y giran en la misma dirección, reciben el nombre de maza cañera (por donde entra la caña) y maza bagacera (por donde sale) o maza de descarga; la otra recibe el nombre de maza mayor o superior, la cual gira en dirección opuesta. (1)

Tacho: Deben reunir las condiciones para trabajar como un evaporador llevando el jarabe de 60° a la concentración correspondiente a su cristalización (95 a 98°), apropiados para la producción de cristales. (1)

Capítulo 1: Fundamentación teórica

Cristalizadores: Aparatos que reciben las masas de segunda y las de tercera o de agotamiento (masas B y C), permanecen más tiempo, durante el cual se somete la masa a un tratamiento para completar la cristalización y llevar el agotamiento de la masa al máximo. (1)

Clarificación: Proceso más eficiente para eliminar las partículas indeseables en suspensión que dejan la bebida turbia y perjudica una buena filtración, tapando los poros de los filtros. La clarificación consiste en adicionar algún producto clarificante en el licor, tales como la clara, albúmina secas, talco, gelatina incolora, bentonita, crema de leche, etc. Todos esos materiales son inofensivos para la salud y tienen un largo empleo en la fabricación de bebidas. (1)

Decantación: Representa la operación de decantar un líquido inclinando al recipiente. (1)

Tachos al vacío: Son equipos que se utilizan en la Industria Azucarera para la cocción de la meladura y las mieles provenientes de las centrifugas para obtener los granos de azúcar, proceso que se llama cristalización. La cristalización del azúcar es un proceso demorado que industrialmente se aumenta introduciendo al tacho unos granos de polvillo de azúcar finamente molido. (1)

1.2 Sistemas de Supervisión, Control y Adquisición de Datos SCADA

SCADA, (Supervisión, Control y Adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores) y controlando el proceso automáticamente. Provee de toda la información que se genera en el proceso productivo (supervisión, control calidad, control de producción, almacenamiento de datos) y permite su gestión e intervención.

El término SCADA usualmente se refiere a un sistema central que monitoriza y controla un proceso completo o una parte de un proceso que nos interesa controlar (el control puede ser sobre máquinas en general, depósitos, bombas) o finalmente un sistema que se extiende sobre una gran área (kilómetros/millas).

Entre las funciones de un sistema SCADA se encuentran:

- **Adquisición de datos:** significa que el sistema tiene la capacidad de obtener información desde el sistema de control, por ejemplo sobre valores de temperatura presión y a diferencia de la

Capítulo 1: Fundamentación teórica

“supervisión” los datos son registrados o almacenados para su posterior explotación.

- **Supervisión:** significa poder observar o monitorear aquello que sucede en el proceso industrial, el equipo o la maquinaria, por ejemplo conocer (generalmente de una forma gráfica) si un motor se encuentra encendido o apagado.
- **Control:** significa tener la posibilidad de ejecutar comandos; en otras palabras, poder enviar instrucciones hacia el sistema de control, por ejemplo ordenar al Controlador Lógico Programable (PLC siglas en inglés) que encienda o apague un motor.

Estas acciones se evidencian a la hora de interrogar de forma periódica a las Unidades de Terminales Remotas (RTU) y transmitirle consignas; siguiendo usualmente un esquema maestro-esclavo; actuar como interfaz al operador, incluyendo la presentación de información de variables en tiempo real, la administración de alarmas, y la recolección y presentación de información historizada; así como la manera de ejecutar software especializado que cumple funciones específicas asociadas al proceso supervisado por el SCADA. (2)

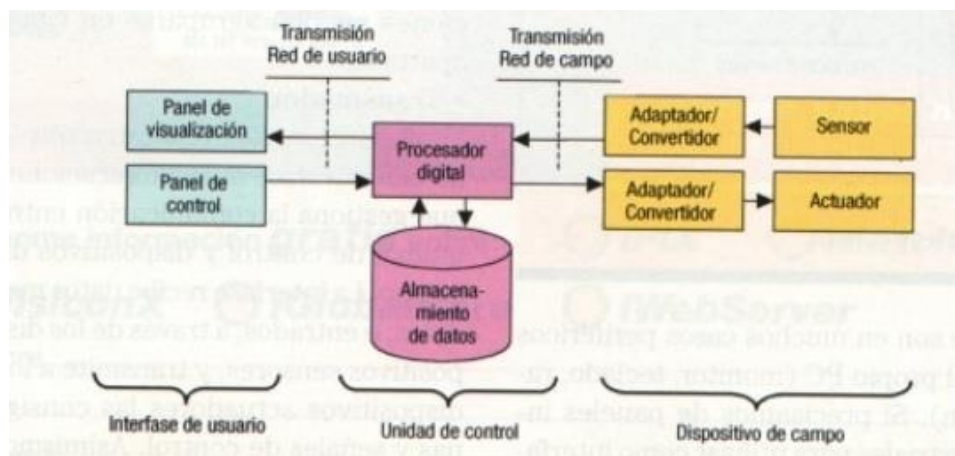


Figura 2. Esquema básico de un sistema de Adquisición, supervisión y control.

1.2.1 Módulos del SCADA SAINUX

- **Comunicación:** Este módulo representa la capa de software encargada de la comunicación entre los diferentes procesos distribuidos, de mediano y alto nivel.
- **Configuración:** Módulo encargado de almacenar, persistir y suministrar, la información base para el funcionamiento de los demás módulos.

Capítulo 1: Fundamentación teórica

- **Seguridad:** Permite a los usuarios autenticarse en el sistema, y de esta forma poder acceder sólo a los recursos que tiene asignado su rol. Posee herramientas para la protección ante ataques piratas, fallos eléctricos y problemas de red.
- **Bases de Datos Históricas:** Es el que permite almacenar la sucesión de alarmas y eventos así como la información recibida desde los dispositivos que se encuentran en el campo. Esta información es de vital importancia para realizar cualquier tipo de análisis posterior como diagnósticos o reportes.
- **Adquisición o Procesamiento de Datos:** Supervisa y almacena la información del resto de las subestaciones, las cuales pueden ser ordenadores conectados a los dispositivos de campo o directamente sobre dichos dispositivos, algunas de las funciones que realiza son:
 - ✓ Actuar como interfaz al operador, incluyendo la presentación de información de variables en tiempo real, la administración de alarmas y la recolección y presentación de la información presente.
 - ✓ Puede ejecutar software especializado que cumple funciones específicas asociadas al proceso supervisado por el SCADA.
- **Interfaz Hombre-Máquina (HMI):** es un mecanismo que le permite a un operador humano interactuar con una máquina o proceso y determinar el estado (prendido / apagado) o magnitud de los dispositivos y/o variables físicas que están presentes en una planta o proceso industrial. La Interfaz HMI es la encargada de representar, en un ordenador, los procesos que ocurren en el campo. A su vez muestra los componentes implicados, los sensores, las estaciones remotas y el sistema de comunicación, permitiendo que el operador posea el control sobre los procesos que intervienen en la industria. Este módulo logra que el operario este en contacto directo con el sistema para así realizar la supervisión del proceso en general. Está compuesto por dos partes fundamentales:
 - **Entorno de Configuración (EC) o Editor:** Permite al mantenedor configurar la información específica del área que se desea supervisar y diseñar los despliegues, los cuales, haciendo uso de los componentes gráficos, permiten simular todos los procesos del campo.
 - **Entorno de Visualización (EV) o Visualizador:** Es donde el operador puede supervisar y controlar toda la información que fue configurada en el EC, interactuando con los componentes gráficos para asegurar el control sobre el sistema, monitorear los cambios de estado de las variables, gestionar alarmas y generar reportes.

1.3 Análisis de sistemas existentes

1.3.1 Análisis de software nacional

Entre los sistemas SCADA más avanzados y reconocidos a nivel nacional e internacional se puede mencionar el Sistema de Supervisión y Control de Procesos EROS, se desarrolló en 1993 e instaló por primera vez en la empresa niquélela Ernesto Che Guevara, Moa. Es un sistema para la Supervisión y Control de Procesos de carácter nacional, diseñado y desarrollado en la División de Automatización, Empresa de Servicios Técnicos de Computación, Comunicaciones y Electrónica (SERCONI).

EROS se comunica con los dispositivos por medio de manejadores (Drivers), para la adaptación a nuevos entornos. Actualmente se conecta con servidores OLE for Process Control (OPC), redes MODBUS (protocolo de comunicaciones), autómatas, reguladores autónomos, analizadores de redes e interruptores, sistemas de pesaje y otros dispositivos. Posee una capa de red a través del protocolo Transmission Control Protocol e Internet Protocol (TCP-IP), que habilita la comunicación entre diferentes computadoras que ejecutan el EROS. Soporta el protocolo OPC v. 2.0 para la Integración con otros SCADA y es compatible con el Remote Desktop Protocol RDP para la administración remota y los servicios de postventa.

EROS no se ajusta a la dinámica de los procesos azucareros y no ha podido ser completada para cubrir todas las funcionalidades necesarias para la producción azucarera. EROS presenta varias desventajas debido a que opera sobre Windows, sistema operativo privado, además no cuenta con una amplia biblioteca de componente en la rama azucarera. Debido a estas dificultades los ingenieros del grupo empresarial AZCUBA diseñan y programan los componentes que requieren para un determinado despliegue mediante el uso de aplicaciones privadas, trayendo como consecuencia el desarrollo de un proceso largo y costoso.

1.4 Conclusiones Parciales

Luego de realizado un análisis de todos los conceptos y las relaciones que intervienen en el proceso de producción azucarera, así como las diferentes técnicas que son necesarios para garantizar el control y la supervisión de los procesos de producción azucarera, se hace necesario realizar una herramienta que permita garantizar una paleta de componentes especializados por procesos industriales que pueda reflejar lo más real posible un proceso industrial. Para el estudio y el reconocimiento de las necesidades que debe suplir la herramienta el autor se apoyó en el producto de software reconocido a nivel nacional EROS, utilizando las funcionalidades que más se ajustaron al dominio del problema.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

Para llevar a cabo la representación visual de los procesos de producción azucarera y lograr que exista una semejanza entre los objetos reales y los diseñados se requiere de diferentes tecnologías y herramientas que facilitan el entorno de trabajo para el desarrollo de dicha aplicación. En este capítulo son recogidos los principales conceptos relacionados al dominio del problema explicando los flujos por los que transita la caña de azúcar desde que comienza el proceso de producción azucarera, apoyándose en la realización del modelo de dominio para identificar con mayor claridad los conceptos y sus relaciones. La realización de un componente que incluya las características y funcionalidades necesarias para determinar el buen funcionamiento y aprovechamiento de la misma. A continuación son definidos los requisitos funcionales y no funcionales que debe poseer el componente para cumplir con los objetivos planteados y agrupados los requisitos funcionales en casos de uso del sistema para centrar la atención en las principales tareas del usuario y los diferentes requerimientos que debe cumplir el componente.

2.1 Tecnologías y herramientas para la implementación

La selección de dichas herramientas y tecnologías está fundamentada por su utilización en el proyecto SCADA SAINUX, por lo que para el desarrollo de la paleta de componentes gráficos especializados en la producción azucarera se hace necesario la utilización de un marco de trabajo (framework, en inglés). Un framework sirve de base para la organización y desarrollo del software. Puede incluir soporte de programas, bibliotecas, un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. También se requiere de un entorno de desarrollo integrado (IDE, por sus siglas en inglés). Un IDE puede denominarse como un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI, por sus siglas en inglés). Además se utiliza un lenguaje de programación, que es una herramienta que permite comunicarse e instruir a la computadora para que realice una tarea específica. Para modelar el sistema se utiliza una herramienta que permite representar sus principales características mediante diagramas. Asimismo se utiliza una metodología de desarrollo de software para estructurar, planificar y controlar todo el proceso de desarrollo del proyecto.

Capítulo 2: Análisis y diseño de la solución

2.1.1 Metodología utilizada

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos. Permiten indicar paso a paso todas las actividades a realizar para lograr el producto informático deseado, delimitando qué personas deben participar en el desarrollo de las actividades y qué papel juegan dentro del mismo. *“Se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán”* (3). El objetivo del uso de metodologías de desarrollo es elevar la calidad del *software* a través de una mayor transparencia y control sobre el proceso.

Para lograr una buena planificación, organización y control del desarrollo de la paleta de componentes gráficos del proceso de producción azucarera en el SCADA SAINUX se decidió utilizar como metodología de desarrollo AUP (Proceso Unificado Ágil). AUP es una versión simplificada del RUP (Proceso Unificado de Desarrollo). En él se describe métodos sencillos, fáciles de entender para el desarrollo de software de aplicaciones empresariales, utilizando técnicas ágiles y conceptos de RUP. Constituye la metodología estándar más utilizada en la universidad, además de ser la metodología utilizada por el proyecto en el cual se realiza esta investigación. Esta metodología abarca siete flujos de trabajo, cuatro de ellos son ingenieriles y los otros tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. Dentro de las fases que propone AUP se encuentran:

- **Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. 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.
- **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 se libera el producto. Durante esta fase el 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.

Capítulo 2: Análisis y diseño de la solución

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

Roles AUP-UCI

- **Administrador de proyecto:** Maneja a los miembros construye relaciones con los clientes, planea, maneja y asigna los recursos.
- **Ingeniero de procesos:** Desarrolla, adapta y apoya sus materiales del proceso del software.
- **Desarrollador:** Escribe, testea y construye software.
- **Administrador de Base de Datos:** Diseña, prueba, desarrolla y apoya los esquemas de la BD.
- **Modelador ágil:** Crea y desarrolla modelos, bosquejos o los archivos de la herramienta CASE, de una manera evolutiva y de colaboración.
- **Administrador de la configuración:** Dirige las actividades relacionadas con la administración de la configuración en el proyecto.
- **Cliente:** Responsable de revisar los compromisos externos que no han sido cumplidos y están en riesgo de no cumplirse.
- **Administrador de pruebas:** Responsables del éxito de las pruebas, incluyendo el planeamiento, la defensa para las pruebas y las actividades de la calidad.
- **Probador:** Encargado de ejecutar las pruebas.

Disciplinas	Objetivos
Modelado de negocio (opcional)	Comprender los procesos de negocio.
Requisitos	Administración y gestión de los requisitos funcionales y no funcionales.
Análisis y diseño	Modelar el sistema.
Implementación	Construcción del sistema.
Pruebas internas	Verificar el resultado de la implementación.
Pruebas de liberación	Diseñar y ejecutar pruebas por una entidad externa certificadora de la calidad.
Pruebas de aceptación	Verificar que el software está listo.

Capítulo 2: Análisis y diseño de la solución

Despliegue (Opcional)	Instalación, configuración, adecuación, y puesta en marcha.
Gestión y soporte	Conjunto de operaciones que se realizan para dirigir y administrar una empresa con el objetivo de sostener o mantener el negocio.

Tabla 1. Disciplinas de AUP-UCI.

2.1.2 Lenguaje de modelado

El lenguaje de modelado que se utiliza es UML 2.1, *“es el más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos propuestos por RUP”* (4). UML ofrece un estándar para guiar los pasos a seguir para realizar un sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación y esquemas de bases de datos. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas para la modelación de la presente investigación como diagramas de clases, de componentes, de despliegue, entre otros.

2.1.3 Herramienta CASE

Como herramienta CASE¹ para el modelado se utiliza el *Visual Paradigm 8.0*, utilizando UML como lenguaje de modelado, definida como *“la aplicación de métodos y técnicas que dan utilidades a los programas, por medio de otros, procedimientos y su respectiva documentación”* (5). *Visual Paradigm 8.0*: Está diseñada para construir sistemas fiables con el uso del paradigma orientado a objetos, incluyendo actividades de la ingeniería de *software*, análisis de sistemas y análisis de negocios (6). Presenta un diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad. Utiliza un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación y la representación de los diagramas imprescindibles para el negocio de la investigación.

2.1.4 Lenguaje de programación

Como lenguaje de programación orientado a objetos se utiliza C++ de propósito general. Brinda la

¹ CASE: Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering).

Capítulo 2: Análisis y diseño de la solución

posibilidad de crear clases, plantillas, sistema de espacios de nombres y funciones en línea, posee un mecanismo para el manejo de excepciones. Permite la sobrecarga de operadores y los utiliza para el manejo de memoria, brinda soporte para la programación genérica. Los códigos escritos en C++ ocupan menos memoria y son más rápidos en comparación con otros lenguajes en tiempo de ejecución.

2.1.5 Marco de trabajo

La utilización de un marco de trabajo (framework, en inglés), como objetivo principal ofrecer una funcionalidad definida, auto contenida, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta. El framework utilizado es Qt en su versión 4.8.2, el cual es multiplataforma para el desarrollo de aplicaciones, las cuales pueden ser con o sin interfaz gráfica. Para el desarrollo del presente trabajo se hace uso de este framework debido a que utiliza el lenguaje de programación C++ de forma nativa y es precisa para realizar todas las interfaces gráficas necesarias. Qt incluye un amplio conjunto de componentes gráficos que proporcionan las funcionalidades estándar de interfaz gráfica de usuario, introduce una innovadora alternativa para la comunicación entre objetos, llamados señales y ranuras. Puede soportar las funcionalidades de interfaz de usuario que requieren las aplicaciones modernas, tales como menús, menús contextuales, arrastrar, soltar, y barras de herramientas. Propone el patrón de diseño modelo/vista para la representación gráfica de los datos, con la separación de las funcionalidades introducidas por esta arquitectura.

2.1.6 Entorno de desarrollo

Se utiliza **Qt Creator** como entorno de desarrollo integrado (IDE) (IDE) en su versión 4.8.2, teniendo en cuenta las facilidades que ofrece para la edición de código en C++ y para el diseño de las interfaces.

2.1.7 Gráficos Vectoriales Escalables (SVG)

Los objetos gráficos pueden ser agrupados, transformados y compuestos en objetos previamente renderizados, y pueden recibir un estilo común. El texto puede estar en cualquier espacio de nombres Lenguaje de Marcas Extensible (XML) admitido por la aplicación, lo que mejora la posibilidad de búsqueda y la accesibilidad de los gráficos SVG. El juego de características incluye las transformaciones anidadas, las rutas recortadas, las máscaras alfa, los filtros de efectos, las plantillas de objetos y la extensibilidad.

Capítulo 2: Análisis y diseño de la solución

El modelo objeto del documento (DOM siglas en inglés) para SVG, que incluye el DOM XML completo, permite animaciones de gráficos vectoriales sencillos y eficientes mediante ECMAScript o SMIL. Debido a su compatibilidad y relación con otras normas Web, características como el scripting pueden ser aplicadas a elementos SVG y a otros elementos XML desde distintos espacios de nombre XML. (8)

SVG es el único formato vectorial capaz de ser interpretado por los navegadores web. Una gran ventaja de la implementación de estos archivos es que nos permitirá tomar aquellos gráficos SVG que cuenten con licencia de uso libre (Creative Commons, etc) presentes en las páginas web y usarlos en nuestros diseños.

Otro punto a favor de la utilización de este formato se debe a que SVG es una especificación para generar imágenes en lenguaje XML que puede tener embebido gráficos vectoriales. El formato SVG no es binario y es perfectamente editable con un simple editor de texto como son Notepad, Gedit, Sublime Text sin necesidad de tener una herramienta profesional de edición de gráficos vectoriales.

2.1.8 Editor Gráfico

InkScape es un editor de gráficos vectoriales de código abierto que usa el formato de archivo SVG y opera sobre sistema operativo GNU/Linux. El objetivo principal de dicho editor es crear una herramienta de dibujo compatible con los estándares XML, SVG y CSS. (9)

2.1.9 Sistema Operativo

El sistema operativo seleccionado es Debian, en su versión 7. Debian o Proyecto Debian es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre. El sistema se encuentra precompilado, empaquetado y en un formato para múltiples arquitecturas de computadoras y para varios núcleos. El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, por lo que es llevado adelante por los usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuirlo comercialmente mientras se respete su licencia.

2.2 Propuesta de solución

2.2.1 Modelo de dominio

El modelo de dominio (MD) proporciona una visión estructural que puede ser completada por los modelos de casos de uso. En él se describen las distintas entidades, atributos, papeles y relaciones, además de las restricciones que rigen el dominio del problema (9). El MD define un vocabulario y es útil como herramienta de comunicación, ayuda al entendimiento del negocio entre los equipos, añadiendo precisión y enfoque en el tema a tratar. En la figura 3 se muestra el diagrama de clases del modelo de dominio asociado a la investigación

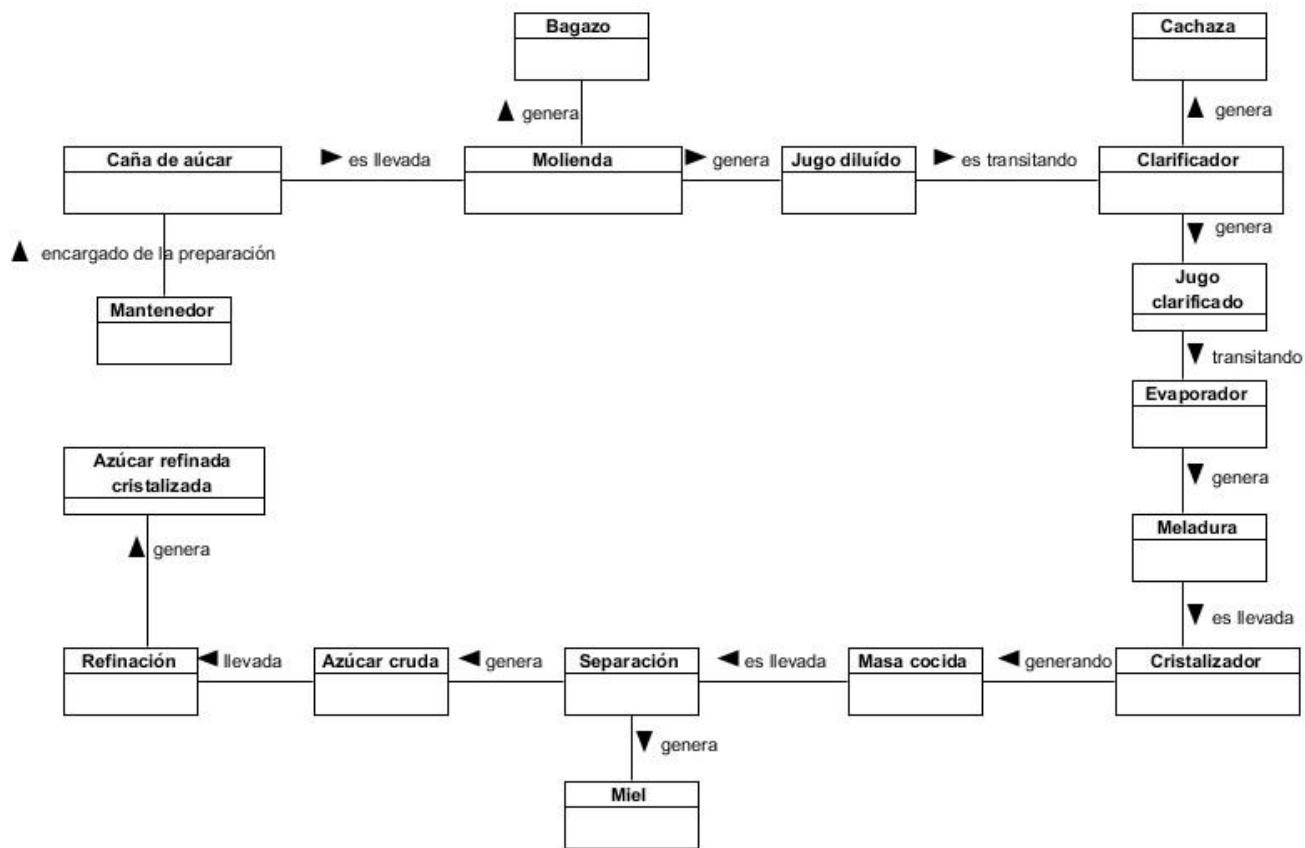


Figura 3. Diagrama de clases del modelo de dominio.

Capítulo 2: Análisis y diseño de la solución

2.2.2 Descripción de clases del modelo de dominio

Con el objetivo de lograr un mejor entendimiento del sistema por parte de los especialistas y desarrolladores, se verán reflejados los principales conceptos asociados al modelo de dominio, empleando para ello un glosario de términos.

- **Mantenedor:** adjetivo/nombre masculino y femenino [persona] Que mantiene.
- **Caña de azúcar:** Planta de tallo leñoso, hueco, flexible, de hojas anchas, algo ásperas y flores agrupadas en panojas muy ramosas; puede alcanzar hasta 4 m de altura.
- **Molienda:** Proceso que consiste en triturar una materia hasta reducirla a trozos muy pequeños o a polvo.
- **Bagazo de caña:** Residuo orgánico resultante del prensado de la caña de azúcar. Puede usarse como combustible y causar contaminación atmosférica por emisión de humos.
- **Jugo diluido:** Consiste en remojar el bagazo con el jugo diluido producto de la imbibición.
- **Clarificador:** El jugo ya calentado es descargado en un tanque conocido como tanque flash, con la finalidad de llevar el líquido a una velocidad laminar, separando el vapor y gases que vienen con el jugo por un auto evaporación debido a la alta temperatura y baja de presión.
- **Cachaza:** Este desecho es producto de la clarificación, está compuesto por: residuo de jugo, lodos, bagacillo y materia extraña separada en la clarificación.
- **Jugo clarificado:** Es el jugo claro que se obtiene después del proceso de clarificación y que se alimenta a los evaporadores.
- **Evapoardor:** Diseñado para concentrar azúcar proveniente del jugo de la caña de azúcar.
- **Meladura:** El jugo concentrado obtenido de la evaporación aún contiene muchas impurezas, por lo que se somete a una nueva clarificación tratándolo con ácido fosfórico, solución de cal, agua oxigenada, floculante y aumento de temperatura.
- **Cristalización:** Esta operación se lleva a cabo en equipos de transferencia térmica que trabajan al vacío, conocidos como tachos, calentando con el vapor proveniente de los evaporadores.
- **Masa cocida:** Su semilla proviene de la manera antes descrita y se alimenta con la miel primera, hasta completare el cocimiento.
- **Separación:** Consiste en la separación de los cristales en masas cocidas de la miel madre utilizando máquina rotativas llamadas centrifugas, aplicando lavados para eliminar las capas de miel que recubren dichos cristales.

Capítulo 2: Análisis y diseño de la solución

- **Miel:** La melaza o miel de caña es un producto líquido y espeso derivado de la caña de azúcar, y en menor medida de la remolacha azucarera, obtenido del residuo restante en las cubas de extracción de los azúcares.
- **Azúcar cruda:** El azúcar moreno, negro, terciado o rubio es un azúcar de sacarosa que tiene un color marrón, turbinado y demerara se obtienen por cristalización del jugo de azúcar crudo, que se centrifuga para retirar el agua y algunas impurezas.
- **Refinado:** Mediante la refinación, se eliminan los colorantes o inorgánicas que el licor pueda contener. El azúcar disuelto se trata con ácido y sacarato de calcio para formar un compuesto que arrastra las impurezas, las cuales se retiran fácilmente en el clasificador. El licor resultante se concentra, se cristaliza de nuevo en un tacho y se pasa a las centrifugas, para eliminar el jarabe.
- **Azúcar refinada cristalizada:** El azúcar refinado se lava con condensado de vapor, se seca con aire caliente, se clasifica según el tamaño del cristal.

2.3 Requisitos del proceso de producción azucarera

2.3.1 Requisitos funcionales

A continuación siguiendo las pautas que propone la metodología AUP-UCI, se encapsularon los requisitos funcionales del sistema en historias de usuarios (HU). Los requisitos funcionales define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Las historias de usuario son una técnica utilizada en AUP-UCI para especificar los requisitos del software. Se realizó una por cada funcionalidad del sistema, fueron empleadas para hacer estimaciones de tiempo y para el plan de iteraciones. Las historias de usuario se definieron lo más comprensible posible para que pueda ser implementada por el desarrollador en pocas semanas. (8)

Historias de Usuario específicas para la etapa de molienda en la producción azucarera.

HU 1: Insertar desfibradora.

HU 2: Eliminar desfibradora.

HU 3: Configurar propiedades de desfibradora.

HU 4: Insertar cuchilla.

HU 5: Eliminar cuchilla.

Capítulo 2: Análisis y diseño de la solución

HU 6: Configurar propiedades de cuchilla.

HU 7: Insertar primera estera.

HU 8: Eliminar primera estera.

HU 9: Configurar propiedades de primera estera.

HU 10: Insertar segunda estera.

HU 11: Eliminar segunda estera.

HU 12: Configurar propiedades de segunda estera.

HU 13: Insertar tercera estera.

HU 14: Eliminar tercera estera.

HU 15: Configurar propiedades de tercera estera.

HU 16: Insertar molino.

HU 17: Eliminar molino.

HU 18: Configurar propiedades de molino.

Historias de Usuario específicas para la etapa de clarificación en la producción azucarera.

HU 19: Insertar calentador doble.

HU 20: Eliminar calentador doble.

HU 21: Configurar propiedades de calentador doble.

HU 22: Insertar calentador triple.

HU 23: Eliminar calentador triple.

HU 24: Configurar propiedades de calentador triple.

HU 25: Insertar calentador quíntuple.

HU 26: Eliminar calentador quíntuple.

HU 27: Configurar propiedades de calentador quíntuple.

HU 28: Insertar clarificador.

HU 29: Eliminar clarificador.

HU 30: Configurar propiedades de clarificador.

HU 31: Insertar tanque clarificador.

HU 32: Eliminar tanque clarificador.

HU 33: Configurar propiedades de tanque clarificador.

Historias de Usuario específicas para la etapa de cristalización en la producción azucarera.

HU 34: Insertar tacho.

Capítulo 2: Análisis y diseño de la solución

HU 35: Eliminar tacho.

HU 36: Configurar propiedades de tacho.

Historias de Usuario específicas para la etapa de evaporación en la producción azucarera.

HU 37: Insertar bomba de tornillo.

HU 38: Eliminar bomba de tornillo.

HU 39: Configurar propiedades de bomba de tornillo.

HU 40: Insertar rallador.

HU 41: Eliminar rallador.

HU 42: Configurar propiedades de rallador.

HU 43: Insertar tanque mezclador.

HU 44: Eliminar tanque mezclador.

HU 45: Configurar propiedades de tanque mezclador.

Historias de Usuario específicas para la etapa de separación y refinación en la producción azucarera.

HU 46: Insertar torre A90.

HU 47: Eliminar torre A90.

HU 48: Configurar propiedades de torre A90.

HU 49: Insertar torre B90.

HU 50: Eliminar torre B90.

HU 51: Configurar propiedades de B90.

HU 52: Insertar torre C90.

HU 53: Eliminar torre C90.

HU 54: Configurar propiedades de C90.

Historia de Usuario	
Número: 34	Nombre del requisito: Insertar Tacho.
Programador: Alejandro Antonio Matamoros Vargas.	Iteración Asignada:

Capítulo 2: Análisis y diseño de la solución

Prioridad: Alta.	Tiempo Estimado: 40 horas.
Riesgo en Desarrollo: Riesgo definido en el GESPRO.	Tiempo Real: 40 horas.
Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar. El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.	

Tabla 2. Historia de Usuario Insertar Tacho.

Las restantes historias de usuario se encuentran en el Anexo 1.

Se definieron los siguientes requisitos no funcionales para el correcto funcionamiento del sistema, los cuales son propiedades o cualidades que el producto debe cumplir. Se deben asumir como propiedades o argumentos que hacen que el producto sea atractivo y aceptado, no definen el éxito general del producto, pero si influyen en la evaluación del cliente

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son atributos de calidad que debe poseer el sistema, especifica criterios que pueden usarse para juzgar la operación del mismo. Son aquellos requisitos no funcionales son atributos de calidad que debe poseer el sistema, especifica criterios que pueden usarse para juzgar la operación del mismo. Son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este. Define propiedades como rendimiento, disponibilidad, seguridad, accesibilidad, usabilidad, estabilidad, interfaz, entre otros. Para la Paleta de Componentes Gráficos del Proceso de Producción Azucarera se acordaron los siguientes requisitos no funcionales.

Requisitos de usabilidad:

El componente de diseño debe ser fácil de usar por cualquier persona con experiencia básica, media o alta sobre el proceso de producción azucarera. Las funcionalidades principales del componente se verán

Capítulo 2: Análisis y diseño de la solución

representadas por íconos para un mejor reconocimiento por parte de los usuarios que trabajaran con la aplicación. Debido que dicha aplicación es un componente que será utilizado en el producto SCADA y las diferentes herramientas que son utilizadas como C++ y *framework* Qt se requiere desarrollar una aplicación de tipo Escritorio.

Requerimientos de interfaz externa:

El componente presenta una interfaz gráfica sencilla que incluye una barra de herramientas de fácil acceso a los principales comandos para el diseño de la producción azucarera. Contiene un panel para la gestión de proyectos y otro panel para representar y editar las propiedades visuales de los objetos. El componente tendrá un ambiente agradable y sencillo, permitiendo que el usuario se adapte con facilidad.

Requisitos de *hardware*:

Para el correcto funcionamiento de la aplicación es necesario que la máquina de la estación de trabajo presente memoria RAM de 2G o superior y microprocesador Dual Core a 1.5 GHz, para garantizar la agilidad de la aplicación en un corto tiempo.

Requisitos de *software*:

La aplicación debe funcionar en Sistema Operativo Debian, en su versión 7.

2.4 Descripción del sistema propuesto

Para darle cumplimiento a los objetivos planteados en el proceso de producción azucarera se decidió utilizar Las Historias de Usuario, permitiendo describir las funcionalidades que deberán realizarse para llevar a cabo dicho proceso.

Capítulo 2: Análisis y diseño de la solución

2.5 Planificación del desarrollo

2.5.1 Estimación de esfuerzos e iteraciones por historia de usuario.

La siguiente tabla muestra estimación de esfuerzo para cada una de las historias de usuarios definidas en el desarrollo de la solución propuesta, el plan de iteraciones realizado y la duración total de las mismas.

Historias de Usuario	Puntos de Estimación	Iteración	Duración total de las iteraciones (semanas)
Insertar desfibadora.	0.4 semanas	1	3
Eliminar desfibadora.	0.6 semanas		
Configurar propiedades de desfibadora.	1 semanas		
Insertar cuchilla.	0.3 semanas		
Eliminar cuchilla.	0.5 semanas		
Configurar propiedades de cuchilla.	0.2 semanas		
Insertar primera estera.	0.5 semanas	2	3
Eliminar primera estera	0.5 semana		
Configurar propiedades de primera estera.	0.5 semanas		
Insertar segunda estera.	0.3 semanas		
Eliminar segunda estera.	0.2 semana		
Configurar propiedades de segunda estera.	1 semanas		
Insertar tercera estera.	0.3 semanas	3	3
Eliminar tercera estera.	0.1 semanas		
Configurar propiedades	0.5 semanas		

Capítulo 2: Análisis y diseño de la solución

de tercera estera.			
Insertar molino.	0.1semanas		
Eliminar molino.	1 semanas		
Configurar propiedades de molino.	1 semanas		
Insertar calentador doble.	0.8 semana	4	3
Eliminar calentador doble.	0.2 semana		
Configurar propiedades de calentador doble.	1 semana		
Insertar calentador triple.	0.2 semanas		
Eliminar calentador triple.	0.4 semana		
Configurar propiedades de calentador triple.	0.2 semana		
Insertar calentador quintuple.	0.2 semana		
Eliminar calentador quintuple.	0.2 semana	5	3
Configurar propiedades de calentador quintuple.	0.5 semana		
Insertar clarificador.	0.3 semana		
Eliminar clarificador.	0.5 semana		
Configurar propiedades de clarificador.	0.5 semanas		
Insertar tanque clarificador.	0.4 semana		
Eliminar tanque clarificador.	0.2 semanas		
Configurar propiedades de tanque clarificador.	0.4 semana		

Capítulo 2: Análisis y diseño de la solución

Insertar tacho.	1 semana	6	3
Eliminar tacho.	0.8 semana		
Configurar propiedades de tacho.	0.2 semanas		
Insertar bomba de tornillo.	0.1 semana		
Eliminar bomba de tornillo.	0.1 semana		
Configurar propiedades de bomba de tornillo.	0.8 semana		
Insertar rallador.	0.2 semanas	7	3
Eliminar rallador.	1 semana		
Configurar propiedades de rallador.	0.8 semana		
Insertar tanque mezclador.	0.4 semana		
Eliminar tanque mezclador.	0.4 semana		
Configurar propiedades de tanque mezclador.	0.2 semana		
Insertar torre A90.	0.2 semanas	8	3
Eliminar torre A90.	1 semana		
Configurar propiedades de torre A90.	0.2 semana		
Insertar torre B90.	0.4 semana		
Eliminar torre B90.	0.2 semana		
Configurar propiedades de B90.	0.2 semana		
Insertar torre C90.	0.2 semana		
Eliminar torre C90.	0.3 semana		

Capítulo 2: Análisis y diseño de la solución

Configurar propiedades de C90.	0.3 semanas		
--------------------------------	-------------	--	--

Tabla 3. Estimación de esfuerzos.

2.5.2 Plan de iteraciones

Luego de definidas las Historias de Usuarios (HU) y estimado el esfuerzo propuesto para su realización, se realizó el sistema en ocho iteraciones, las cuales se describen detalladamente a continuación:

Iteración I, II y III: Iteraciones que tiene como objetivo realizar las HU 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 y 18 las cuales se encargan del desarrollo de los componentes gráficos para la etapa de molienda en la producción azucarera.

Iteración IV y V: Iteraciones que tiene como objetivo realizar las HU 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 y 33 las cuales se encargan del desarrollo de los componentes gráficos para la etapa de clarificación en la producción azucarera.

Iteración VI y VII: Iteraciones que tiene como objetivo realizar las HU 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44 y 45 las cuales se encargan del desarrollo de los componentes gráficos para las etapas de cristalización y evaporación en la producción azucarera.

Iteración VIII: Iteración que tiene como objetivo realizar las HU 46, 47, 48, 49, 50, 51, 52, 53 y 54 las cuales se encargan del desarrollo de los componentes gráficos para las etapas de separación y refinación en la producción azucarera.

2.6 Conclusiones del capítulo

Las herramientas y tecnologías escogidas para elaborar el componente propiciaron mayor facilidad en el diseño e implementación del mismo, contribuyendo a la independencia tecnológica por la que aboga el país. Fue descrita la propuesta de solución de la paleta de componentes gráficos del proceso de producción azucarera a través del modelo de dominio que permitió comprender con mayor claridad los conceptos y las relaciones existentes entre el especialista y las diferentes transiciones de la azúcar. Los requisitos funcionales y no funcionales que debe cumplir la aplicación permitieron definir de las características y funcionalidades de la misma tanto en *hardware* como en *software*, ubicando al equipo de desarrollo en los objetivos de la solución y proporcionando una guía para su desarrollo y posterior validación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Para realizar la etapa de diseño e implementación del componente propuesto se procede a utilizar la arquitectura de debe poseer el mismo, siendo la misma la principal base para desarrollar un sistema. Fueron utilizados diferentes estilos arquitectónicos para garantizar la estructura y organización de los componentes del *software*, sus propiedades y las conexiones entre ellos, así como los patrones utilizados para asignar responsabilidades y mantener la organización en la implementación de dicha aplicación. A partir de la arquitectura y los patrones utilizados en el componente se procede a diseñar mediante diagramas de clases de diseño y diagramas de componente para encapsular los diferentes aspectos que interviene en cada uno de estos diagramas, ayudando a la organización del código y funcionalidades del componente.

3.1 Estilos arquitectónicos utilizados

La arquitectura de software proporciona una visión global del sistema a construir, describe la estructura y la organización de los componentes del *software*, sus propiedades y las conexiones entre ellos. Los componentes de *software* incluyen varios módulos y representan los datos que serán manipulados por el programa. El logro efectivo del diseño de una aplicación que persigue el objetivo de cumplir con los requisitos propuestos, está guiado por una arquitectura que defina los diferentes patrones que brindan un esquema de referencia útil para guiar el desarrollo de *software*. Todo esto posibilita que los integrantes del equipo de proyecto trabajen sobre la misma línea y exista una compatibilidad entre ellos para alcanzar los objetivos trazados. (10)

Los estilos arquitectónicos son también patrones de construcción y se le conoce como la descripción de una categoría del sistema que contiene: un conjunto de componentes que realiza una función requerida por el sistema; un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes; restricciones que definen como se puede integrar los componentes que forman el sistema; y modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema.

En el desarrollo del producto se empleó la arquitectura de software Modelo-Vista de Qt.

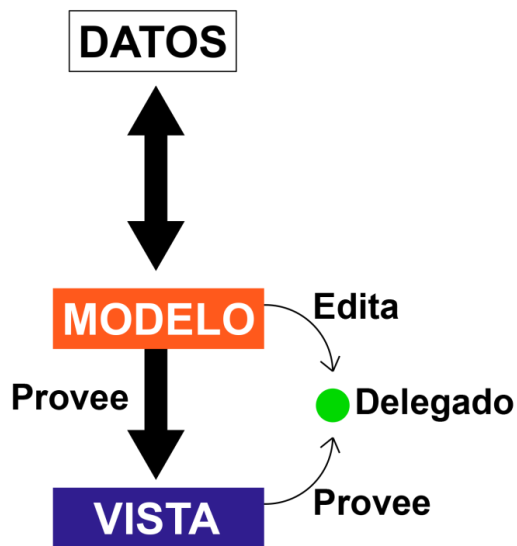


Figura 4: Patrón de Arquitectura de Software Modelo-Vista.

La arquitectura de software Modelo-Vista de Qt, tiene como objetivo que las clases modelo se comunican con una fuente de datos, proporcionando una interfaz para los otros componentes en la arquitectura, dependiendo del tipo de fuente de datos y la forma en que se implementa el modelo. La clase vista obtiene referencias a objetos de datos. Las clases modelo / vista son separadas en: modelos, vistas y delegados. Cada uno de estos componentes se define por clases abstractas que proporcionan interfaces comunes.

3.2 Patrones de diseño

Los patrones de diseño son la base para buscar soluciones a problemas en el diseño de interacción o interfaces dentro del desarrollo de un *software*. Un conjunto de patrones arquitectónicos permiten que el ingeniero del *software* reutilice los conceptos a nivel de diseño. Los patrones de diseño permiten al diseñador crear la arquitectura de diseño integrando componentes reusables. (11) Los patrones no expresan nuevas formas de realizar el diseño, todo lo contrario, sino que pretenden codificar conocimiento, estilos y principios existentes que ya han sido probados y son válidos. A continuación se muestran los patrones utilizados en el diseño de la propuesta de solución

Capítulo 3: Implementación y prueba

1. **Patrones GRASP²:** Describen los principios fundamentales para asignar responsabilidades a los objetos. Los autores de esta investigación definieron los siguientes patrones que fueron utilizados.

Experto: Se utiliza para asignar responsabilidades a los objetos, con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Se evidencia en las clases BladeModel expertas en los datos que serán utilizados por cada clase en los cuales cada una tiene datos que solo ella puede manejar.

Creador: Es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Se evidencia en la presente investigación en las clases *Estera* encargada de guardar todos los datos referentes a las esteras.

Bajo Acoplamiento: Se encarga de que exista poco dependencia entre las clases. En la presente investigación se ve evidenciado la relación de bajo acoplamiento y alta cohesión al separar los elementos. En todo el proyecto de manera general.

Alta Cohesión: Es el encargado de que cada elemento debe realizar una tarea única dentro de la aplicación y se ve evidenciado en la investigación ya que cada clase tiene su función definida dentro del sistema. En todo el proyecto de manera general.

Controlador: Se encarga de gestionar un evento de entrada al sistema. Se evidencia en la clase que controla todo el flujo de entradas del especialista, la clase Blade.

2. **Patrones GOF³:** describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. Los patrones GOF se clasifican en tres grandes categorías basadas en su propósito: creacionales para la abstracción de cómo crear un objeto, estructurales para indicar cómo están compuestas las clases y de comportamiento para la asignación de responsabilidades en las clases.

Observador u *Observer*: Se encarga de velar por el comportamiento entre las clases y objetos, si uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Se evidencia en las clases correspondientes a las vistas de los componentes y la clase Measurement, las vistas estas suscritas y

² GRASP: Patrones de Asignación de Responsabilidades (General Responsibility Assignment Software Patterns).

³ GOF: Banda de cuatro (Gang of Four).

Capítulo 3: Implementación y prueba

cuando ocurre un cambio en la medición estas se actualizan mediante la función `updateState()`.

Método Plantilla: Se encarga de una vez implementado en una clase abstracta el código común que será usado por las clases que heredan de ella, permitirle que implementen el comportamiento que varía mediante la reescritura (total o parcial) de determinados métodos. (11) Este patrón se ve reflejado en los métodos `paintRuntime ()`, `paintEdition ()` y `loadSVG ()`.

3.3 Modelo de diseño

El modelo de diseño de un sistema completo incorpora las representaciones del *software* en función de los datos, arquitectura, interfaz y procedimiento. (11) Es un diagrama que permite optimizar los detalles que lleva la etapa de implementación, resaltando la estructura y el estilo arquitectónico, los componentes que residen dentro de la arquitectura y las interfaces entre los componentes y el mundo exterior.

3.3.1 Diagrama de secuencia

El diagrama de secuencia representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores. Los diagramas de secuencia en UML, son los encargados de representar la forma en que los objetos se comunican entre sí al transcurrir el tiempo. (11)

Capítulo 3: Implementación y prueba

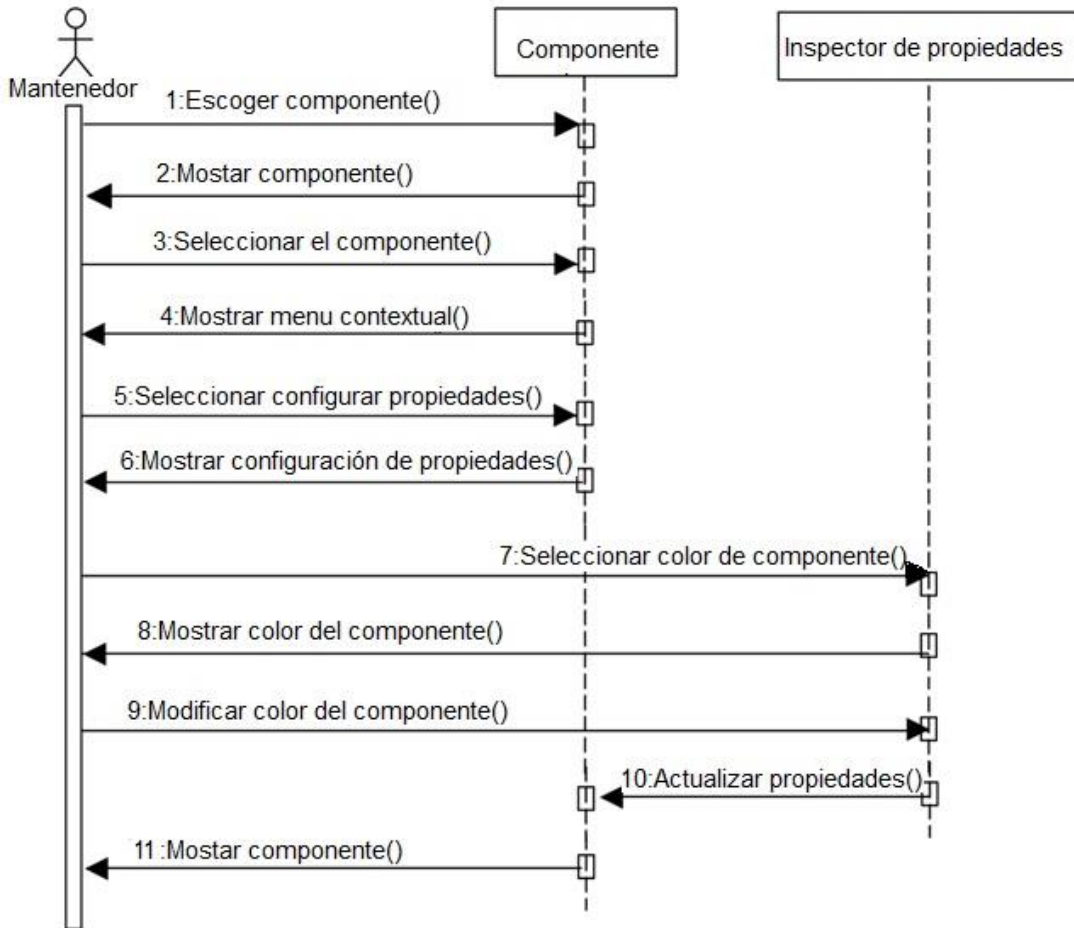


Tabla 4. Diagrama de secuencia de componentes gráficos.

3.3.2 Diagrama de clases del diseño

Los diagramas de clases se utilizan para modelar la visión estática de un sistema, describen las estructuras de un sistema mostrando sus clases, orientados a objetos. Presenta las clases del sistema con sus relaciones estructurales y de herencia, incluyendo conjuntos de operaciones y propiedades que son implementadas para una interfaz gráfica. (3)

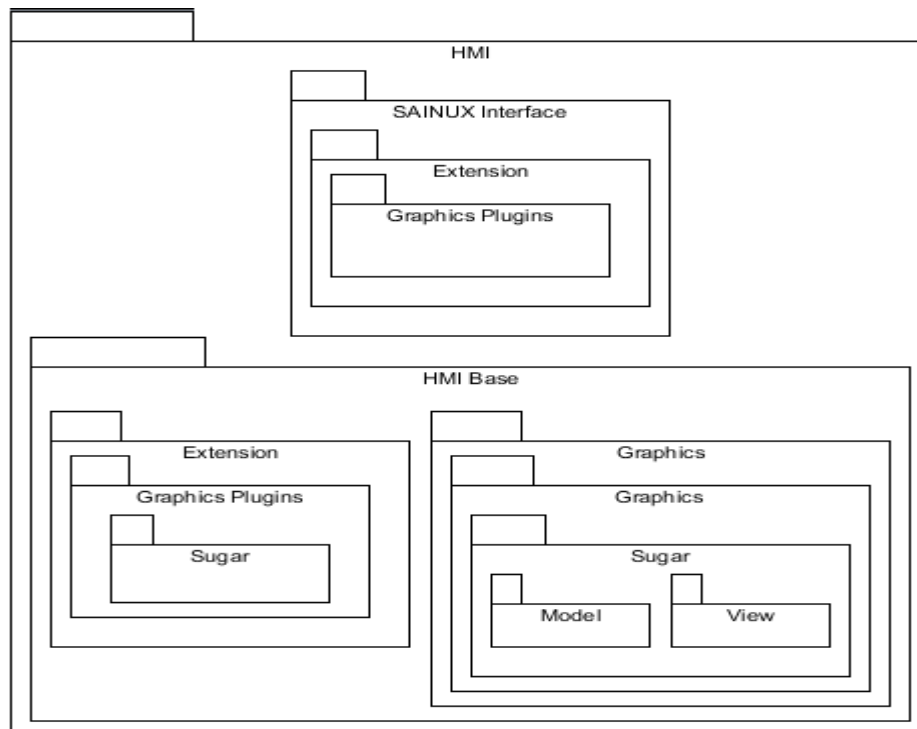


Figura 5. Diagrama de paquetes.

Para una mejor comprensión visual, se diseñó el siguiente esquema, al cual representa nuestro diagrama de clase de los componentes gráficos, en el empaquetado Sugar se encuentra todas las clases de nuestra solución, estas son de tipo Model (por ejemplo la clase BladeModel) o de tipo View (por ejemplo la clase Blade) las clases view heredan de la clase GrphicsSVGObject y las clases model heredan de la clase GrphicsSVGObjectModel.

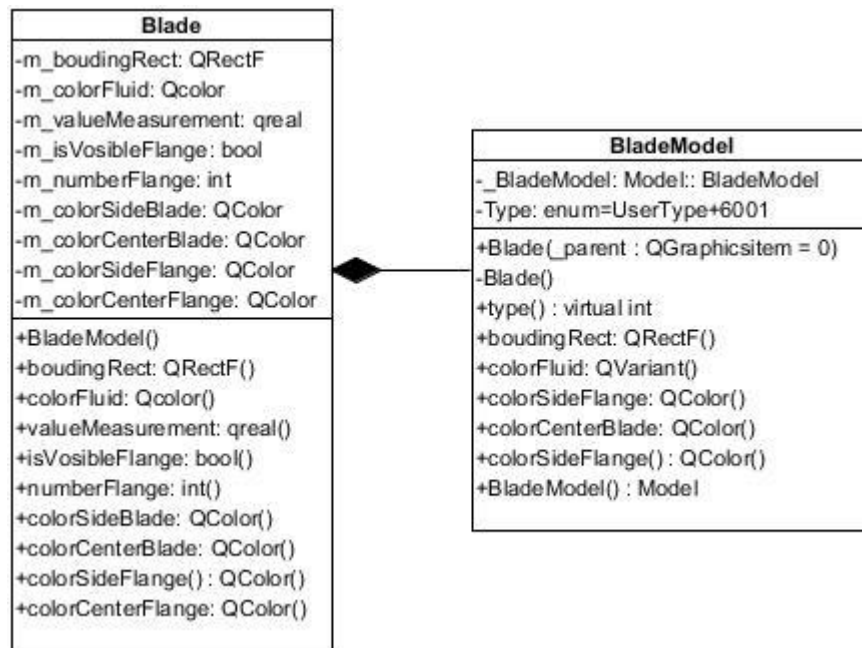
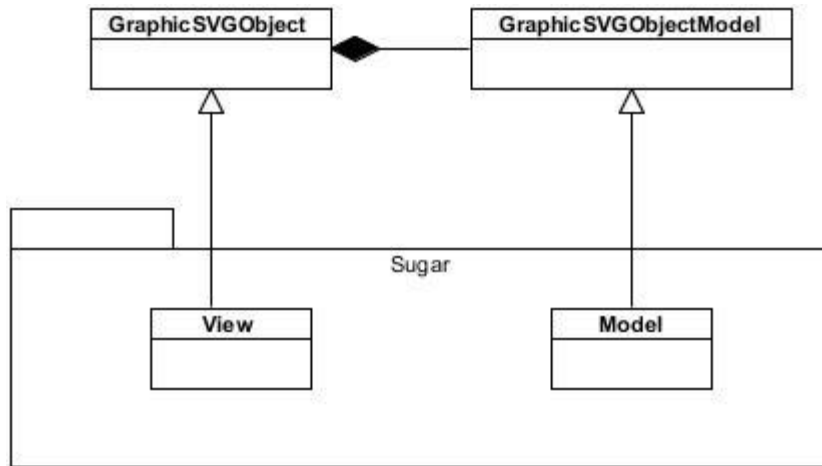


Figura 6. Diagrama de Clases de Componentes Gráficos.

3.4 Modelo de implementación

“El Modelo de Implementación es el proceso de convertir una especificación del sistema en un sistema ejecutable”. (10)El Modelo de Implementación describe cómo los elementos de diseño se implementan en

Capítulo 3: Implementación y prueba

componentes, estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y otros tipos de ficheros necesarios para la implementación y el despliegue del sistema. En este modelo se describen las relaciones entre los paquetes y las clases del diseño con los diferentes subsistemas y componentes físicos.

3.4.1 Diagrama de componentes

Un componente es *“una parte modular, desplegable y reemplazable de un sistema que encapsula implementación y expone un conjunto de interfaces”*. (10) Un diagrama de componentes muestra la organización, relaciones y dependencias entre los componentes de un sistema. Se utiliza para modelar la vista estática de un sistema y muestra la organización y las dependencias lógicas entre un conjunto de componentes de *software*.

En el Diagrama de Componentes el módulo depende físicamente de las siguientes librerías:

- GraphicsPlugins: Permite el almacenamiento de Plugins de los objetos gráficos.
- Graphics: Permite agrupar las entidades encargadas de representar los componentes gráficos, que luego serán utilizados en la representación.
- SXGraphic Plugins: Es la interfaz específica de la biblioteca Graphics Plugins que es utilizada para los componentes de SAINUX.
- ResourceManager: Biblioteca encargada de gestionar los recursos del sistema.
- Editor: Es una aplicación de escritorio encargada de configurar todos los recursos asociados al proceso de supervisión en el SCADA SAINUX.
- Visualizador: Encargado de visualizar, monitorear y controlar los recursos asociados al proceso de supervisión en el SCADA SAINUX.

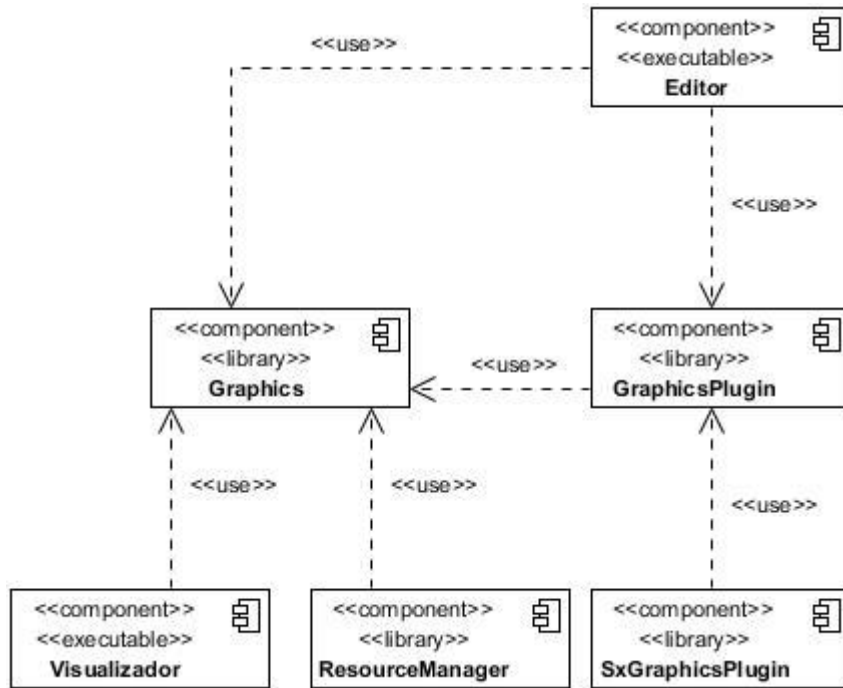


Figura 7. Diagrama de componentes.

3.4.2 Estándar de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo único, logrando uniformidad entre los programadores. (11) El componente de la presente investigación está sujeto a los estándares utilizados en el proyecto por lo que fueron definidos los siguientes:

- Los atributos de las clases deben comenzar con la letra m seguido de guión bajo y a continuación el nombre del atributo, si existen atributos compuestos la segunda palabra debe comenzar con mayúscula.

Ejemplo: `m_isColorFlange`

- Las funciones no deben exceder de las 200 líneas de código.
- Las secciones `public`, `protected` y `private` serán declaradas en este orden.
- El código será escrito en inglés y la documentación en español.

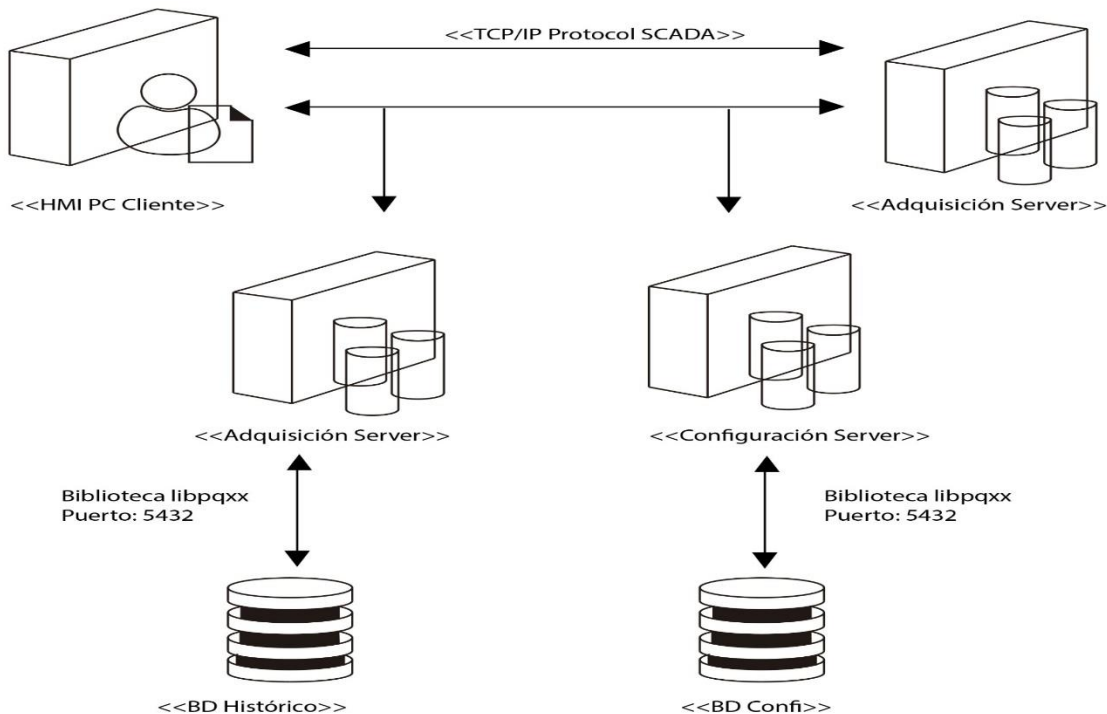
Capítulo 3: Implementación y prueba

- Para especificar el nombre del autor, en la documentación, se utilizan los comandos @autor.
- Las variables y funciones comienzan con letra minúscula, cada palabra consecutiva en el nombre comienza con letra mayúscula.
- Los valores de los numerativos deben ser con letras mayúsculas.
- Las funcionalidades y atributos para documentar deben cumplir el siguiente formato @brief Nombre del método.
- Los parámetros que recibe una función deben comenzar con guion bajo.
- Las expresiones regulares en el SVG se declaran con dos @@

Ejemplo: @@bladecolor@@

3.5 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los nodos que componen un sistema y la distribución de los componentes sobre dichos nodos. (11)



Capítulo 3: Implementación y prueba

Figura 8. Diagrama de despliegue.

3.5.1 Descripción del diagrama de despliegue

La PC-Cliente es el ordenador donde se configurará el módulo HMI el cual es el nodo donde se encuentra nuestra solución, donde pueden estar instalados uno o los dos entornos. Los módulos adquisición, configuración y base de datos históricos se ejecutan en una PC, haciendo función de servidor para cada uno por separado, la conexión entre estos módulos y la PC-Cliente se realiza utilizando comunicación TCP/IP con el protocolo SCADA desarrollado por el centro. El módulo de configuración y base de datos históricos se conectan a las bases de datos por la biblioteca libpqxx utilizando el puerto 5432.

3.6 Pruebas

Las pruebas de software son elementos críticos para garantizar la calidad de la aplicación, representando una revisión final de las especificaciones, del diseño y de la codificación (9). Las pruebas son flujos de trabajo donde se verifica el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema para luego ser entregadas a terceros. (12)

Tipos de pruebas de software

- Unitarias: Se encargan de ejecutar cada módulo, particionar y definir los casos de pruebas y comparar el resultado.
- Regresión: Se encargan de identificar errores introducidos por la combinación de programas probados unitariamente y determina como será cargada la base de datos de prueba.
- Integridad: Se asegura que los métodos de acceso y proceso funcionen adecuadamente y sin provocar corrupción de datos, determina como será cargada la base de datos de prueba y asegura que la implementación ha sido exitosa.
- Aceptación: Se encarga de verificar por parte de cliente que el software cumpla con sus expectativas. Evalúa las entradas que se realizan sobre el software y las salidas que produce, sin preocuparse por su comportamiento interno.
- Seguridad y control de acceso: Se encarga de verificar que un actor solo pueda acceder a las

Capítulo 3: Implementación y prueba

funciones y datos que su usuario tiene permitido y que solo los actores con acceso al sistema y a la aplicación están habilitados para accederla.

En el desarrollo de paleta de componentes gráficos del proceso de producción azucarera en el SCADA SAINUX será utilizada la prueba de aceptación, siendo esta la prueba más idónea por la metodología de desarrollo empleada, Para la realización de la misma se requiere la participación del cliente y del equipo de desarrollo, generando como artefacto los Casos de Pruebas (CP).

3.6.1 Pruebas de aceptación

Las pruebas de aceptación son realizadas por el cliente, son aplicadas al sistema en general y se apoyan en la especificación de requisitos y en el manual de usuario y son aplicadas una vez terminado e integrado el producto. A continuación se muestra un CP correspondiente a las pruebas de aceptación realizadas.

Caso de prueba

Prueba de Aceptación	
Número: 34	Historia de usuario: 34
Nombre: Insertar Tacho	
Descripción: Permite insertar el clarificador al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico Tacho este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.	
Resultado esperado: Permite insertar el tacho y se muestra en el área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Capítulo 3: Implementación y prueba

Tabla 5. Caso de Prueba 34.

Una vez realizadas las tres iteraciones de pruebas se obtuvieron los siguientes resultados:

Sistema	HU	Iteración	NC	Cerrada	No Procede
Componente azucarero.	54	1ra	10	8	2
		2da	8	8	0
		3ra	0	0	0

Tabla 6. Resultados de las pruebas.

El Componente de Desarrollo de paleta de componentes gráficos del proceso de producción azucarera en el SCADA SAINUX fue comprobado a través de las pruebas de aceptación en tres iteraciones, que se describen a continuación:

Iteración 1: Se encontraron un total de 10 no conformidades relacionadas con la visualización de la información que se desea mostrar. Fueron arreglados los errores encontrados y se procede a la segunda iteración.

Iteración 2: Se comprobaron nuevamente todos los elementos de la iteración anterior, obteniéndose 8 no conformidades relacionadas con las funcionalidades que debía cumplir el sistema. Fueron arreglados los errores encontrados y se procede a la tercera iteración.

Iteración 3: Fueron comprobados nuevamente los elementos, no encontrándose no conformidades. Concluyendo así el proceso de prueba de integración al componente.

A continuación el grafico de barra se muestra los resultados de las pruebas:

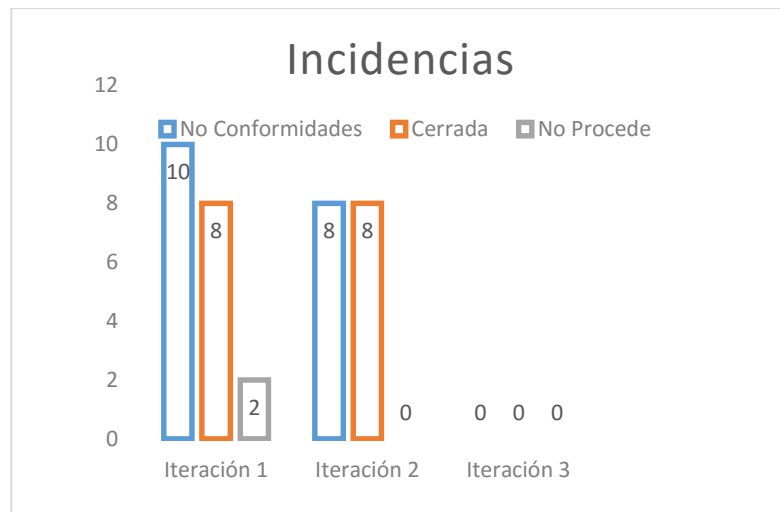


Figura 9. Incidencias detectadas en el sistema.

3.7 Conclusiones del capítulo

La arquitectura identificada y los patrones de diseño implementados garantizan una mayor organización en el diseño e implementación del componente, los cuales están adecuados a la propuesta de diseño elaborada en el proyecto SCADA XAINUX, promoviendo un producto de software estandarizado. Para comprobar el correcto funcionamiento del componente fueron realizadas las pruebas de aceptación, la cual se considera que fueron cumplidos los objetivos por los cuales fue aplicada.

CONCLUSIONES

Con el desarrollo de la presente investigación se logró organizar y guiar el trabajo hacia el cumplimiento del objetivo general perseguido. Luego de estudiar las metodologías de desarrollo de software se selecciona la metodología ágil AUP-UCI implantada en el proyecto, para guiar el proceso de diseño y desarrollo de la solución. El diagnóstico realizado sirvió de base para la caracterización de la propuesta de solución, permitiendo definir y describir en un nivel de detalle significativo las funcionalidades y clases a implementar. Además posibilitó planificar, estimar el tiempo de entrega y duración del proyecto. También fue posible mediante el uso de AUP-UCI, determinar las pruebas a realizar al código, a las funcionalidades desde el entorno de desarrollo para validar y certificar el correcto funcionamiento del subsistema. Al concluir el presente trabajo se obtuvo como resultado:

- Una paleta de componentes gráficos especializada en la producción azucarera e integrada al módulo HMI del SCADA SAINUX.
- Las soluciones nacionales existentes analizadas no responden en su totalidad al problema de la presente investigación, no incluyen dentro de sus funcionalidades la posibilidad de redimensionar la imagen deseada sin perder calidad, afectando negativamente la representación gráfica de los elementos.
- El uso de herramientas que permitan visualizar el proceso azucarero garantizando con alto valor de precisión la supervisión de dicho proceso garantiza la eficiencia, la calidad del software y el trabajo de los especialistas en esta área.
- La utilización de herramientas y tecnologías libres, permitió que el componente sea extensible y contribuya a la independencia tecnológica por la que aboga el país, creando herramientas de carácter nacional que permitan agilizar el proceso de producción azucarera.
- La arquitectura identificada y los patrones de diseño implementados garantizan una mayor organización en el diseño e implementación del componente, promoviendo un producto de software estandarizado y relacionado el producto SCADA SAINUX.
- Las pruebas aplicadas se consideran aceptadas y garantizan que el componente funcione correctamente permitiendo visualizar los procesos de producción azucarera.

Referencias bibliográficas

Trabajos citados

1. **Ferrer, F.A. López.** Manual Práctico de Maquinaria y Aparato en los ingenios de azúcar de caña. [aut. libro] pepe azucar. *Manual Práctico de Maquinaria y Aparato en los ingenios de azúcar de caña.* La Habana : s.n., 1969.
2. **Dagoberto Monteros, David B. Barrantes y José M. Quirós.** *Introducción a los sistemas de control, supervisión y adquisición de datos (SCADA).* 2004.
3. **Quintero, Juan Bernardo, y otros.** "Un estudio comparativo de herramientas para el modelado con UML." revista universidad eafit 41, no. 137 (2012): 60-76. 2012.
4. **Microsoft.** Diagramas de clases de UML: Instrucciones. [En línea] [Citado el: 14 de marzo de 2016.] <https://msdn.microsoft.com/es-es/library/dd409416.aspx>.
5. **Larman, Craig.** *Applying UML and Patterns.* Prentice Hall. : s.n., 1998.
6. **Mad, Freepress S. Coop.** Freepress S. Coop. Inkscape-software-libre-para-diseño-vectorial. [En línea] [Citado el: 2 de diciembre de 2015.] <http://www.freepress.coop/recursos/inkscape-software-libre-para-diseno-vectorial/>.
7. **Pressman, Roger.** *Ingeniería del software: Un enfoque práctico. Sexta Edición.* New York : McGrawHill : s.n., 2005.
8. **Jacobson, Ivar , Booch , Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* 2000.
9. **Arquitectura de software. Diagrama de componentes.** [En línea] [Citado el: 29 de marzo de 2016.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
10. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque Práctico.* [En línea] [Citado el: 3 de febrero de 2016.] <http://bibliodoc.uci.cu/pdf/8448111869.pdf>.
11. **Estándares de Codificación .Net v.1.0.0.0.** [En línea] [Citado el: 15 de abril de 2016.] <http://serk.kualtus.com/codigo.htm>.
12. **Alarcón, Andrea y Alarcón, Érica.** *Herramientas CASE para Ingeniería de Requisitos.* 2008.
13. **QTCreator.** *QT Creator and Tools.* 2013.
14. **ALEGSA.com.ar.** DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. Definición de SVG. [En línea] [Citado el: 17 de enero de 2016.] <http://www.alegsa.com.ar/Dic/svg.php>.
15. **Buschman, Regine Meunier.** *La Guía de Arquitectura. Version 2.0. .* 2009.
16. **Paradigm, Visual.** "Visual paradigm for uml. *Visual Paradigm for UML-UML tool for software application development*". . 2010.
17. **RAE.** <http://www.rae.es>. *RAE.* [En línea] Diccionario de la Lengua Española. Vigésima segunda edición., 2013. [Citado el: 05 de 01 de 2014.]
18. **Penin, A.R.** *Sistemas SCADA.* s.l. : Marcombo, 2012. ISBN 9788426716477.
19. **CORPORATION, NOKIA.** Qt. [En línea] 2008. [Citado el: 2 de diciembre de 2015.] <http://qt.nokia.com/products/developer-tools/>.
20. **Ingeniería de Software.** [En línea] [Citado el: 15 de marzo de 2016.] <http://clases3gingsof.wikifoundry.com/page/Proceso+de+Despliegue+de+RUP>.
21. **UML y Patronos.** [aut. libro] Craig Larman. *UML y Patronos. Modelo de Dominio.* . Prentice Hall : s.n., 2003.
22. **Qt Documentation.** [En línea] [Citado el: 16 de marzo de 2016.] <http://doc.qt.io/qt-4.8/model-view-programming.html>.
23. **Funciones del Sistema SCADA.** [En línea] [Citado el: 4 de octubre de 2015.] http://www.oocities.org/gabrielordonez_ve/FUNCIONES_DEL_SISTEMA_SCADA.htm.
24. **Relaciones entre clases: Diagramas de clases UML.** [En línea] [Citado el: 10 de marzo de 2016.] <http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf>.

Bibliografía Consultada

Bibliografía

1. **Ferrer, F.A. López.** Manual Práctico de Maquinaria y Aparato en los ingenios de azúcar de caña. [aut. libro] pepe azucar. *Manual Práctico de Maquinaria y Aparato en los ingenios de azúcar de caña*. La Habana : s.n., 1969.
2. **Dagoberto Monteros, David B. Barrantes y José M. Quirós.** *Introducción a los sistemas de control, supervisión y adquisición de datos (SCADA)*. 2004.
3. **Quintero, Juan Bernardo, y otros.** "Un estudio comparativo de herramientas para el modelado con UML." revista universidad eafit 41, no. 137 (2012): 60-76. 2012.
4. **Microsoft.** Diagramas de clases de UML: Instrucciones. [En línea] [Citado el: 14 de marzo de 2016.] <https://msdn.microsoft.com/es-es/library/dd409416.aspx>.
5. **Larman, Craig.** *Applying UML and Patterns*. Prentice Hall. : s.n., 1998.
6. **Mad, Freepress S. Coop.** Freepress S. Coop. Inkscape-software-libre-para-diseño-vectorial. [En línea] [Citado el: 2 de diciembre de 2015.] <http://www.freepress.coop/recursos/inkscape-software-libre-para-diseno-vectorial/>.
7. **Pressman, Roger.** *Ingeniería del software: Un enfoque práctico. Sexta Edición*. New York : McGrawHill : s.n., 2005.
8. **Jacobson, Ivar , Booch , Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. 2000.
9. **Arquitectura de software. Diagrama de componentes.** [En línea] [Citado el: 29 de marzo de 2016.] <http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
10. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque Práctico*. [En línea] [Citado el: 3 de febrero de 2016.] <http://bibliodoc.uci.cu/pdf/8448111869.pdf>.
11. **Estándares de Codificación .Net v.1.0.0.0.** [En línea] [Citado el: 15 de abril de 2016.] <http://serk.kualtus.com/codigo.htm>.
12. **Alarcón, Andrea y Alarcón, Érica.** *Herramientas CASE para Ingeniería de Requisitos*. 2008.
13. **QTCreator.** *QT Creator and Tools*. 2013.
14. **ALEGSA.com.ar.** DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA. Definición de SVG. [En línea] [Citado el: 17 de enero de 2016.] <http://www.alegsa.com.ar/Dic/svg.php>.
15. **Buschman, Regine Meunier.** *La Guía de Arquitectura. Version 2.0.* . 2009.
16. **Paradigm, Visual.** "Visual paradigm for uml. Visual Paradigm for UML-UML tool for software application development". . 2010.
17. **RAE.** <http://www.rae.es>. *RAE*. [En línea] Diccionario de la Lengua Española. Vigésima segunda edición., 2013. [Citado el: 05 de 01 de 2014.]
18. **Penin, A.R.** *Sistemas SCADA*. s.l. : Marcombo, 2012. ISBN 9788426716477.
19. **CORPORATION, NOKIA.** Qt. [En línea] 2008. [Citado el: 2 de diciembre de 2015.] <http://qt.nokia.com/products/developer-tools/>.
20. **Ingeniería de Software.** [En línea] [Citado el: 15 de marzo de 2016.] <http://clases3ggingsof.wikifoundry.com/page/Proceso+de+Despliegue+de+RUP>.
21. **UML y Patrones.** [aut. libro] Craig Larman. *UML y Patrones. Modelo de Dominio.* . Prentice Hall : s.n., 2003.
22. **Qt Documentation.** [En línea] [Citado el: 16 de marzo de 2016.] <http://doc.qt.io/qt-4.8/model-view-programming.html>.
23. **Funciones del Sistema SCADA.** [En línea] [Citado el: 4 de octubre de 2015.] http://www.oocities.org/gabrielordonez_ve/FUNCIONES_DEL_SISTEMA_SCADA.htm.
24. **Relaciones entre clases: Diagramas de clases UML.** [En línea] [Citado el: 10 de marzo de 2016.] <http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf>.

Glosario de términos

Desfibradoras: Asegurar la alimentación a los molinos y preparar la caña, aplastándola para facilitar su toma y la extracción del jugo. Generalmente consta de dos cilindros, los cuales poseen una superficie diseñada para agarrar la caña, dicha superficie se debe construir de manera que rasgue y desfibre la caña.

Molinos: Son cilindros horizontales cuyos centros forman un triángulo isósceles. Dos de ellas se encuentran a la misma altura y giran en la misma dirección, reciben el nombre de maza cañera (por donde entra la caña) y maza bagacera (por donde sale) o maza de descarga; la otra recibe el nombre de maza mayor o superior, la cual gira en dirección opuesta.

Tacho: Deben reunir las condiciones para trabajar como un evaporador llevando el jarabe de 60°Brix a la concentración correspondiente a su cristalización (95 a 98°Brix), apropiados para la producción de cristales.

Cristalizadores: Aparatos que reciben las masas de segunda y las de tercera o de agotamiento (masas B y C), permanecen más tiempo, durante el cual se somete la masa a un tratamiento para completar la cristalización y llevar el agotamiento de la masa al máximo.

Clarificación: Proceso más eficiente para eliminar las partículas indeseables en suspensión que dejan la bebida turbia y perjudica una buena filtración, tapando los poros de los filtros. La clarificación consiste en adicionar algún producto clarificante en el licor, tales como la clara, albúmina secas, talco, gelatina incolora, bentonita, crema de leche, etc. Todos esos materiales son inofensivos para la salud y tienen un largo empleo en la fabricación de bebidas.

Decantación: Representa la operación de decantar un líquido inclinando al recipiente.

Tachos al vacío: Son equipos que se utilizan en la Industria Azucarera para la cocción de la meladura y las mieles provenientes de las centrifugas para obtener los granos de azúcar, proceso que se llama cristalización. La cristalización del azúcar es un proceso demorado que industrialmente se aumenta introduciendo al tacho unos granos de polvillo de azúcar finamente molido.

Anexos

Anexo 1. Entrevista a Rafael Antonio Matamoros Verdecia

¿Cuáles son las etapas del proceso azucarero?

¿En qué consiste cada una de las etapas del proceso azucarero?

¿Cuáles son los principales métodos de producción y en qué consisten?

A continuación se muestra las descripciones de las Historias de Usuarios representadas en el desarrollo de la paleta de componente, cada requerimiento representa un insertar, modificar, eliminar y configurar las propiedades de un determinado componente sin necesidad de variar en sus funcionalidades, mostrando en los anexos dos ejemplos que esclarecen las actividades que se ejecutan para cada componente con estas cuatro funcionalidades, así mismo para los casos de pruebas asociados a los ejemplos que se muestran a continuación:

Historia de Usuario	
Número: 1	Nombre del requisito: Insertar desfibrador.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 1.
Prioridad: Alta.	Tiempo Estimado: 16 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 16 horas.
Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar. El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.	

Tabla 7. Historia de Usuario Insertar desfibrador.

Historia de Usuario	
Número: 2	Nombre del requisito: Eliminar desfibrador.

Anexos

Programador: Alejandro Antonio Matamoros	Iteración Asignada: 1.
Prioridad: Alta.	Tiempo Estimado: 24 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 24 horas.
Descripción: Posibilita al mantenedor eliminar un componente gráfico en el despliegue. El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente a eliminar ejecutar la tecla Supr (suprimir).	

Tabla 8. Historia de Usuario Eliminar desfibradora.

Historia de Usuario	
Número: 3	Nombre del requisito: Configurar propiedades de desfibradora.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 1.
Prioridad: Alta.	Tiempo Estimado: 40 horas.
Riesgo en Desarrollo: Alta.	Tiempo Real: 40 horas.
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades. El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.	

Tabla 9. Historia de Usuario Configurar propiedades de desfibradora.

Historia de Usuario	
Número:	Nombre del requisito: Insertar calentador doble.

19		
Programador: Alejandro Antonio Matamoros		Iteración Asignada: 4.
Prioridad: Alta.		Tiempo Estimado: 32 horas.
Riesgo en Desarrollo: Medio.		Tiempo Real: 32 horas.
<p>Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar.</p> <p>El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.</p>		

Tabla 10. Historia de Usuario calentador doble.

Historia de Usuario		
Número: 37	Nombre del requisito: Insertar bomba de tornillo.	
Programador: Alejandro Antonio Matamoros		Iteración Asignada: 6.
Prioridad: Alta.		Tiempo Estimado: 4 horas.
Riesgo en Desarrollo: Medio.		Tiempo Real: 4 horas.
<p>Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar.</p> <p>El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.</p>		

Tabla 11. Historia de Usuario Insertar bomba de tornillo.

Historia de Usuario

Anexos

Historia de Usuario	
Número: 47	Nombre del requisito: Eliminar torre A90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 40 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 40 horas.
Descripción: Posibilita al mantenedor eliminar un componente gráfico en el despliegue. El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente a eliminar ejecutar la tecla Supr (suprimir).	

Tabla 12. Historia de Usuario Eliminar torre A90.

Historia de Usuario	
Número: 48	Nombre del requisito: Configurar propiedades de torre A90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 8 horas.
Riesgo en Desarrollo: Alta.	Tiempo Real: 8 horas.
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades. El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.	

Tabla 13. Historia de Usuario Configurar propiedades de torre A90.

Anexos

Historia de Usuario	
Número: 49	Nombre del requisito: Insertar torre B90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 12 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 12 horas.
Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar. El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.	

Tabla 14. Historia de Usuario Insertar torre B90.

Historia de Usuario	
Número: 50	Nombre del requisito: Eliminar torre B90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 8 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 8 horas.
Descripción: Posibilita al mantenedor eliminar un componente gráfico en el despliegue. El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente a eliminar ejecutar la tecla Supr (suprimir).	

Tabla 15. Historia de Usuario Eliminar Insertar torre B90.

Anexos

Historia de Usuario	
Número: 51	Nombre del requisito: Configurar propiedades torre B90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 8 horas.
Riesgo en Desarrollo: Alta.	Tiempo Real: 8 horas.
<p>Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.</p> <p>El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.</p>	

Tabla 16. Historia de Usuario Configurar propiedades de torre B90.

Historia de Usuario	
Número: 52	Nombre del requisito: Insertar torre C90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 8 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 8 horas.
<p>Descripción: Posibilita al mantenedor incorporar un componente gráfico al despliegue que desea diseñar.</p> <p>El mantenedor debe hacer doble clic primario sobre el componente o clic primario sostenido sobre el componente arrastrándolo hacia el área de trabajo, que está ubicado en la paleta especializa de componentes azucareros.</p>	

Tabla 17. Historia de Usuario Insertar torre C90.

Historia de Usuario	
Número: 53	Nombre del requisito: Eliminar torre C90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 8 horas.
Riesgo en Desarrollo: Medio.	Tiempo Real: 8 horas.
<p>Descripción: Posibilita al mantenedor eliminar un componente gráfico en el despliegue.</p> <p>El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente a eliminar ejecutar la tecla Supr (suprimir).</p>	

Tabla 18. Historia de Usuario Modificar torre C90.

Historia de Usuario	
Número: 54	Nombre del requisito: Configurar propiedades torre C90.
Programador: Alejandro Antonio Matamoros	Iteración Asignada: 8.
Prioridad: Alta.	Tiempo Estimado: 12 horas.
Riesgo en Desarrollo: Alta.	Tiempo Real: 12 horas.
<p>Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.</p> <p>El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último</p>	

seleccionar el color deseado y aceptar para que los cambios se guarden.

Tabla 19. Historia de Usuario Configurar propiedades de torre C90.

Prueba de Aceptación	
Número: 1	Historia de usuario: 1
Nombre: Insertar Desfibradora	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.	
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 20. Prueba de aceptación Insertar desfibradora.

Prueba de Aceptación	
Número: 2	Historia de usuario: 2
Nombre: Eliminar Desfibradora	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	

Anexos

Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• Presionar la tecla Supr (suprimir).
Resultado esperado: Permite eliminar el componente del área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 21. Prueba de aceptación Eliminar desfibradora.

Prueba de Aceptación	
Número: 3	Historia de usuario: 3
Nombre: Configurar propiedades de desfibradora.	
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.	
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 22. Prueba de aceptación Configurar propiedades de desfibradora.

Prueba de Aceptación

Anexos

Número: 22	Historia de usuario: 22
Nombre: Insertar calentador triple	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.	
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 23. Prueba de aceptación Insertar calentador triple.

Prueba de Aceptación	
Número: 23	Historia de usuario: 23
Nombre: Eliminar calentador triple.	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• Presionar la tecla Supr (suprimir).	

Anexos

Resultado esperado: Permite eliminar el componente del área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 24. Prueba de aceptación Eliminar calentador triple.

Prueba de Aceptación	
Número: 24	Historia de usuario: 24
Nombre: Configurar propiedades de calentador triple.	
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.	
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 25. Prueba de aceptación Configurar propiedades de calentador triple.

Prueba de Aceptación	
Número: 25	Número: 25
Nombre: Insertar calentador quíntuple	

Anexos

Descripción: Permite insertar el componente al área de trabajo.
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 26. Prueba de aceptación Insertar calentador quintuple.

Prueba de Aceptación	
Número: 26	Historia de usuario: 26
Nombre: Eliminar calentador quintuple.	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• Presionar la tecla Supr (suprimir).	
Resultado esperado: Permite eliminar el componente del área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no	

conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 27. Prueba de aceptación Eliminar calentador quintuple.

Prueba de Aceptación	
Número: 27	Historia de usuario: 27
Nombre: Configurar propiedades de calentador quintuple.	
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none"> • Seleccionar el componente gráfico en el área de trabajo. • El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden. 	
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 28. Prueba de aceptación Configurar propiedades de calentador quintuple.

Prueba de Aceptación	
Número: 28	Historia de usuario: 28
Nombre: Insertar clarificador	
Descripción: Permite insertar el componente al área de trabajo.	

Anexos

Prueba de Aceptación	
Número: 30	Historia de usuario: 30
Nombre: Configurar propiedades de clarificador.	
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.	
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 29. Prueba de aceptación Configurar propiedades de clarificador.

Prueba de Aceptación	
Número: 31	Número: 31
Nombre: Insertar tanque clarificador	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	

Anexos

Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 30. Prueba de aceptación Insertar tanque clarificador.

Prueba de Aceptación	
Número: 32	Historia de usuario: 32
Nombre: Eliminar tanque clarificador.	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• Presionar la tecla Supr (suprimir).	
Resultado esperado: Permite eliminar el componente del área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 31. Prueba de aceptación Eliminar tanque clarificador.

Prueba de Aceptación	
Número: 33	Historia de usuario: 33

Anexos

Nombre: Configurar propiedades de tanque clarificador.
Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 32. Prueba de aceptación Configurar propiedades de tanque clarificador.

Prueba de Aceptación	
Número: 34	Historia de usuario: 34
Nombre: Insertar tacho	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción	

Anexos

azucarera.
<ul style="list-style-type: none">• Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 33. Prueba de aceptación Insertar tachó.

Prueba de Aceptación	
Número: 35	Historia de usuario: 35
Nombre: Eliminar tachó.	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• Presionar la tecla Supr (suprimir).	
Resultado esperado: Permite eliminar el componente del área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 34 Prueba de aceptación Eliminar tachó.

Prueba de Aceptación	
Número: 36	Historia de usuario: 36
Nombre: Configurar propiedades de tachó.	

Anexos

Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none"> • Seleccionar el componente gráfico en el área de trabajo. • El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 35. Prueba de aceptación Configurar propiedades de tachó.

Prueba de Aceptación	
Prueba de Aceptación	
Número: 43	Historia de usuario: 43
Nombre: Insertar tanque mezclador.	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none"> • Seleccionar el componente gráfico en la paleta especializada de la producción 	

<p>azucarera.</p> <ul style="list-style-type: none"> Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.
<p>Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.</p>
<p>Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.</p>

Tabla 36. Prueba de aceptación Insertar tanque mezclador.

Prueba de Aceptación	
Número: 44	Historia de usuario: 44
Nombre: Eliminar tanque mezclador.	
Descripción: Permite eliminar el componente del área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.	
Entradas/ Pasos de Ejecución:	
<ul style="list-style-type: none"> Seleccionar el componente gráfico en el área de trabajo. Presionar la tecla Supr (suprimir). 	
Resultado esperado: Permite eliminar el componente del área de trabajo.	
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.	

Tabla 37. Prueba de aceptación Eliminar tanque mezclador.

Prueba de Aceptación	
Número: 45	Historia de usuario: 45
Nombre: Configurar propiedades de tanque mezclador.	

Anexos

Descripción: Posibilita al mantenedor configurar las propiedades existente en el inspector de propiedades.
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en el área de trabajo.
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en el área de trabajo.• El mantenedor debe hacer clic primario sobre el componente para seleccionarlo, una vez seleccionado el componente el sistema muestra una ventana (inspector de propiedades), ir a la opción de cambiar el color haciendo clic primario, por último seleccionar el color deseado y aceptar para que los cambios se guarden.
Resultado esperado: Permite modificar las propiedades del componente y se los cambios de la modificación.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 38. Prueba de aceptación Configurar propiedades de tanque mezclador.

Prueba de Aceptación	
Número: 46	Historia de usuario: 46
Nombre: Insertar torre A90.	
Descripción: Permite insertar el componente al área de trabajo.	
Condiciones de Ejecución: El usuario debe haber ejecutado el editor del SCADA-SAINUX y comprobar que el componente gráfico este en la paleta de componentes gráficos azucarera.	
Entradas/ Pasos de Ejecución: <ul style="list-style-type: none">• Seleccionar el componente gráfico en la paleta especializada de la producción azucarera.	

<ul style="list-style-type: none">Hacer doble clic primario (clic izquierdo) sobre el componente gráfico o mantener el clic primario y arrastrarlo hacia el área de trabajo.
Resultado esperado: Permite insertar el componente y se muestra en el área de trabajo.
Evaluación de la prueba: Se realiza la primera iteración y fueron identificadas algunas no conformidades, se solucionan, se realizar otra iteración y da como resultado satisfactoria.

Tabla 39. Prueba de aceptación insertar torre A90.

A continuación se muestran los diagramas de secuencias de los principales componentes, los restantes diagramas son semejantes cambiando solo el componente gráfico.

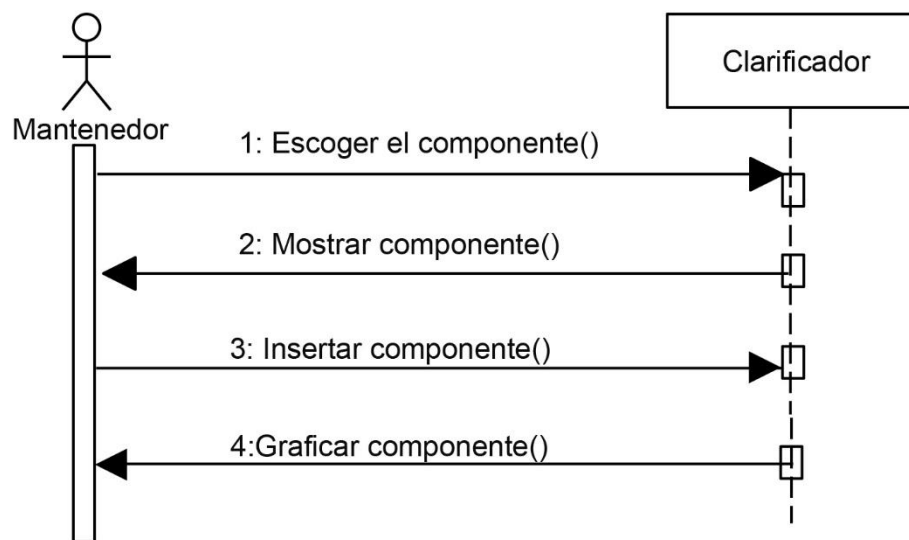


Figura 10. Diagrama de secuencias Insertar clarificador.

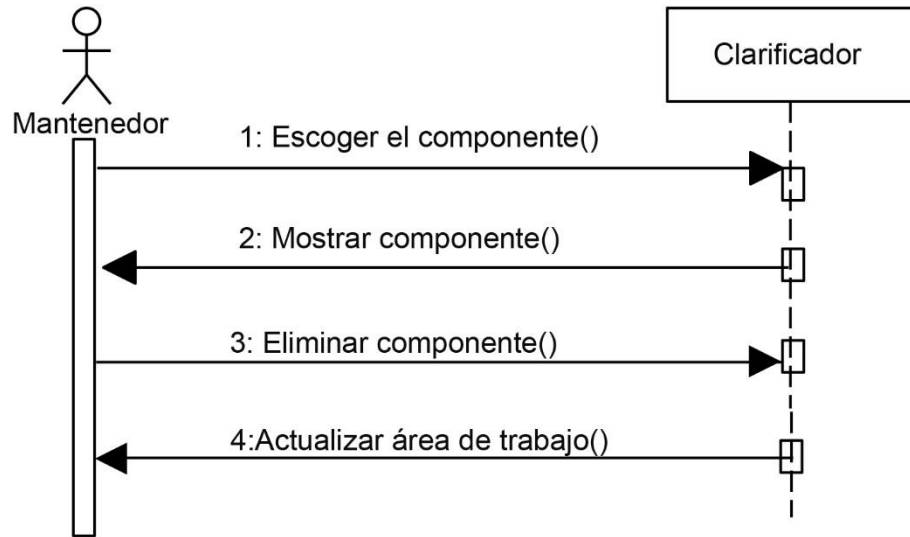


Figura 11. Diagrama de secuencias Eliminar clarificador.

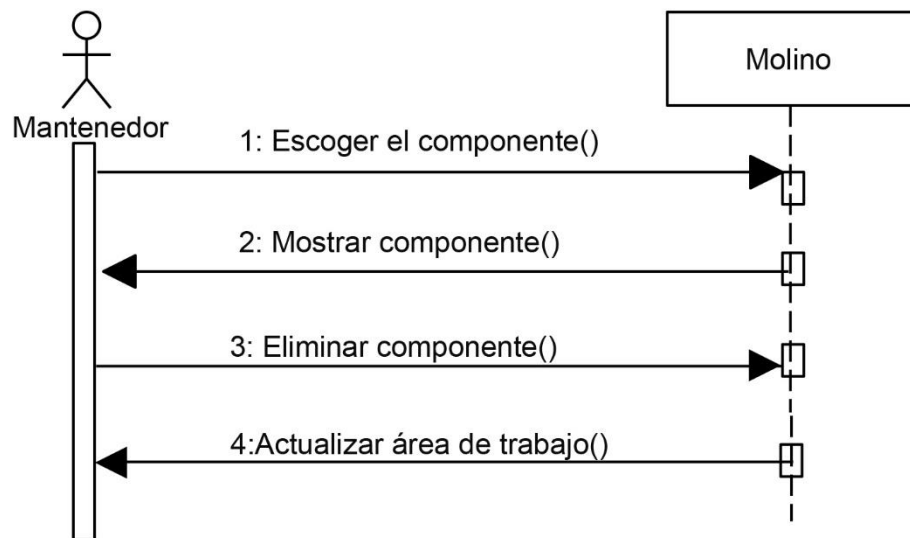


Figura 12. Diagrama de secuencias Eliminar molino.

Figura 13. Diagrama de secuencias Configurar propiedades de clarificador.