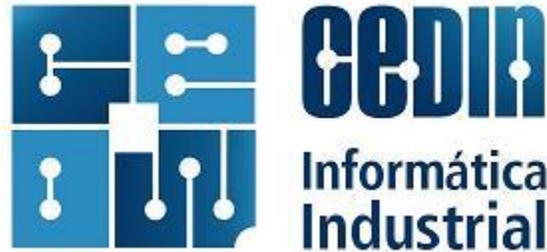




Universidad de las Ciencias Informáticas

Facultad 5



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

**Título: “Subsistema de bitácoras de operaciones para el SCADA
SAINUX”**

Autor: Luis Carlos Cruz Pérez

Tutores: Ing. Mariela Cepero Nuñez

Ing. José Antonio Aragón Cáceres

La Habana, 2015.

Año 57 de la Revolución.

DECLARACIÓN DE AUTORÍA:

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas como entidad con los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los __ días del mes de _____ del año 2015.

Luis Carlos Cruz Pérez

Autor

Ing. Mariela Cepero Nuñez

Tutor

Ing. José Antonio Aragón Cáceres

Tutor

DATOS DEL CONTACTO

Tutor: Mariela Cepero Nuñez

Categoría Científica: Máster en Ciencias

Especialidad: Ingeniero en Ciencias Informáticas

Correo Electrónico: mceperon@uci.cu

Categoría docente: Profesora Asistente

Años de experiencia: 7

Tutor: José Antonio Aragón Cáceres

Categoría Científica: -

Especialidad: Ingeniero en Ciencias Informáticas

Correo Electrónico: jaaragon@uci.cu

Categoría docente: Profesor Asistente

Años de experiencia: 5

Agradecimientos

A mis tutores Mariela y José Antonio que son los mejores, gracias por la ayuda y por ser pacientes conmigo.

A Valido, Miguel Ángel Socorro y Yusnier Sosa, gracias por la ayuda durante el desarrollo de la tesis.

A mis amigos: Lázaro Llanes, Javier, Israel, Jorge Bárbaro, Adrián, Addiel, Susana, Alexis, Ever, Gladys, Maiyara, Manuel, Liuver, Raudel, Tony, Julio, Ernesto Gutierrez, Gretel Naranjo, Ernesto Romero, Fernando, Eddiel y Diosmel que de una forma u otra contribuyeron en hacer posible este momento y en especial a Henry que ha sido un tutor más y me ayudó muchísimo; gracias mi hermano.

A mami Edita por siempre estar ahí para mí, por su apoyo y sus oraciones, por hacer posible lo que creíamos imposible, por haberme ayudado tanto y por siempre decirme "Niño no te preocupes, que todo va a salir bien".

A Cley: la niñita más linda del mundo, pues en momentos en los que rendirme estaba por convertirse en una opción, fue su recuerdo el que me hizo seguir adelante.

A toda mi familia, en especial a papi Chalo, a Yasi, a mis hermanos Roly y Reynaldito, a mi padre Rey y a tío Tomy. Ustedes no se imaginan cuanto me han ayudado: Gracias!

Dedicatoria

A toda mi familia, en especial a mami Edita, Cley, Yarel, papi Chalo, Yasi, Roly y Alex. En todo momento están en mis pensamientos y en mi corazón, y siempre los estoy extrañando.

Resumen

Un sistema SCADA (Supervisión, Control y Adquisición de Datos) es una aplicación de software, diseñada con la finalidad de controlar y supervisar procesos a distancia. Como parte del compromiso de la Universidad de las Ciencias Informáticas (UCI), dirigido a lograr la informatización de la sociedad y la soberanía tecnológica en el país, el Centro de Informática Industrial (CEDIN) se dio a la tarea de desarrollar un SCADA, con características propias y adaptables a la industria nacional para suplir las necesidades de nuestro país en materia de automatización: el SCADA SAINUX. El SCADA SAINUX está compuesto por varios módulos tales como: comunicaciones, adquisición de datos, interfaz hombre-máquina (HMI), seguridad y base de datos históricos. El objetivo de este trabajo es desarrollar un subsistema, como parte del módulo HMI, que permita registrar y consultar sucesos, mediante la utilización de bitácoras de operaciones almacenadas en la base de datos históricos. En las bitácoras se registra la información de los sucesos ocurridos en los turnos de guardia de los operadores, y que no son registrados automáticamente por el sistema, con el fin de tener constancia de lo ocurrido y que esta información pueda ser consultada, analizada y auditada en cualquier momento.

Palabras Clave: base de datos históricos, bitácoras, HMI, SCADA SAINUX, sucesos.

Índice

Resumen.....	V
Índice de Figuras.....	IX
Índice de Tablas	X
Introducción	1
Capítulo 1: Fundamentación teórica.....	4
1.1 Introducción.....	4
1.1.1 Automatización Industrial.....	4
1.1.2 SCADA	4
1.1.3 SCADA SAINUX	5
1.1.4 HMI	6
1.1.5 Bitácoras	6
1.1.6 Bitácoras en sistemas informáticos	7
1.2 Metodología, tecnologías y herramientas.....	9
1.2.1 Metodología de desarrollo de software.....	9
1.2.2 Lenguaje de modelado	11
1.2.3 Herramienta de modelado.....	11
1.2.4 Herramientas de desarrollo	11
1.2.5 Sistema gestor de bases de datos	12
1.2.6 Lenguaje de programación para la implementación de la solución.....	12
1.3 Conclusiones parciales del capítulo	12
Capítulo 2: Características y diseño del sistema.....	13
2.1 Introducción.....	13
2.2 Modelo conceptual.....	13
2.3 Especificación de los Requisitos de Software	14
2.3.1 Requisitos funcionales	14
2.3.2 Requisitos no funcionales	15
2.4 Propuesta de solución	16
2.5 Historias de Usuario	17
2.5.1 Descripción de las Historias de Usuario	17
2.6 Diseño de la arquitectura propuesta.....	31
2.6.1 Estilo arquitectónico	31
2.6.2 Arquitectura del subsistema	31
2.7 Patrones de diseño	32
2.7.1 Patrones GRASP	32
2.7.2 Patrones GOF.....	32
2.8 Diagrama de clases del diseño.....	34

2.8.1 Descripción de las clases del diseño	34
2.9 Conclusiones parciales del capítulo	35
Capítulo 3: Implementación y pruebas del sistema.....	36
3.1 Introducción.....	36
3.2 Diagrama de componentes.....	36
3.3 Diagrama de Despliegue.....	37
3.4 Estándar de Codificación	37
3.5 Pruebas de software.....	38
3.6 Planes de prueba y diseños de casos de prueba	39
3.6.1 Plan de prueba para la Historia de Usuario 1: Adicionar suceso	39
3.6.2 Diseño de caso de prueba 1: Adicionar Suceso.....	40
3.6.3 Plan de prueba para la Historia de Usuario 2: Agregar grupo de sucesos	40
3.6.4 Diseño de caso de prueba 2: Agregar grupo de sucesos.....	41
3.6.5 Plan de prueba para la Historia de Usuario 3: Modificar grupo de sucesos.....	42
3.6.6 Diseño de caso de prueba 3: Modificar grupo de sucesos	43
3.6.7 Plan de prueba para la Historia de Usuario 4: Eliminar grupo de sucesos.....	43
3.6.8 Diseño de caso de prueba 4: Eliminar grupo de sucesos	44
3.6.9 Plan de prueba para la Historia de Usuario 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones	44
3.6.10 Diseño de caso de prueba 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.....	45
3.6.11 Plan de prueba para la Historia de Usuario 6: Agregar tipo de sucesos	46
3.6.12 Diseño de caso de prueba 6: Agregar tipo de sucesos.....	47
3.6.13 Plan de prueba para la Historia de Usuario 7: Modificar tipo de sucesos	47
3.6.14 Diseño de caso de prueba 7: Modificar tipo de sucesos	48
3.6.15 Plan de prueba para la Historia de Usuario 8: Eliminar tipo de sucesos	48
3.6.16 Diseño de caso de prueba 8: Eliminar tipo de sucesos	49
3.6.17 Plan de prueba para la Historia de Usuario 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.....	49
3.6.18 Diseño de caso de prueba 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.....	50
3.6.19 Plan de prueba para la Historia de Usuario 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.....	51
3.6.20 Diseño de caso de prueba 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.....	52
3.7 Desarrollo de las iteraciones de pruebas	52
3.8 Resumen de no conformidades	53
3.9 Clasificación de las no conformidades	53
3.10 Conclusiones parciales del capítulo.....	53

Conclusiones generales	55
Recomendaciones	56
Referencias bibliográficas y bibliografía	57

Índice de Figuras

Figura 1: Bitácora del empleado (8)	7
Figura 2: Bitácora del sistema. (8)	8
Figura 3: Bitácora de procesos. (8)	8
Figura 4: Fases e Iteraciones de la metodología AUP variación UCI. (10)	10
Figura 5: Modelo Conceptual	13
Figura 6: Función Send Event.	16
Figura 7: Propuesta de solución	17
Figura 8: Correspondencia entre la Arquitectura Modelo-Vista y la Arquitectura de N capas. (21)	31
Figura 9: Diagrama de clases del diseño.	34
Figura 10: Diagrama de Componentes	36
Figura 11: Diagrama de Despliegue.	37
Figura 12: Estándar de codificación, ejemplo 1.	38
Figura 13: Estándar de codificación, ejemplo 2.	38
Figura 14: Clasificación de las no conformidades	53

Índice de Tablas

Tabla 1: Historia de Usuario Adicionar suceso.	18
Tabla 2: Historia de Usuario Agregar grupo de sucesos.	19
Tabla 3: Historia de Usuario Modificar grupo de sucesos.	20
Tabla 4: Historia de Usuario Eliminar grupo de sucesos.	22
Tabla 5: Historia de Usuario Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	23
Tabla 6: Historia de Usuario Agregar tipo de sucesos.	24
Tabla 7: Historia de Usuario Modificar tipo de sucesos.	25
Tabla 8: Historia de Usuario Eliminar tipo de sucesos.	26
Tabla 9: Historia de Usuario Visualizar tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	28
Tabla 10: Historia de Usuario Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.	30
Tabla 11: Plan de prueba para la Historia de Usuario 1: Adicionar suceso.	40
Tabla 12: Diseño de caso de prueba 1: Adicionar Suceso.	40
Tabla 13: Plan de prueba para la Historia de Usuario 2: Agregar grupo de sucesos.	41
Tabla 14: Diseño de caso de prueba 2: Agregar grupo de sucesos.	42
Tabla 15: Plan de prueba para la Historia de Usuario 3: Modificar grupo de sucesos.	43
Tabla 16: Diseño de caso de prueba 3: Modificar grupo de sucesos.	43
Tabla 17: Plan de prueba para la Historia de Usuario 4: Eliminar grupo de sucesos.	44
Tabla 18: Diseño de caso de prueba 4: Eliminar grupo de sucesos.	44
Tabla 19: Plan de prueba para la Historia de Usuario 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	45
Tabla 20: Diseño de caso de prueba 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	46
Tabla 21: Plan de prueba para la Historia de Usuario 6: Agregar tipo de sucesos.	47
Tabla 22: Diseño de caso de prueba 6: Agregar tipo de sucesos.	47
Tabla 23: Plan de prueba para la Historia de Usuario 7: Modificar tipo de sucesos.	48
Tabla 24: Diseño de caso de prueba 7: Modificar tipo de sucesos.	48
Tabla 25: Plan de prueba para la Historia de Usuario 8: Eliminar tipo de sucesos.	49
Tabla 26: Diseño de caso de prueba 8: Eliminar tipo de sucesos.	49
Tabla 27: Plan de prueba para la Historia de Usuario 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	50
Tabla 28: Diseño de caso de prueba 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	51

Tabla 29: Plan de prueba para la Historia de Usuario 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.....	52
Tabla 30: Diseño de caso de prueba 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.	52
Tabla 31: Resumen de no conformidades.....	53

Introducción

La informática industrial se enfoca en la aplicación de métodos y técnicas de las ciencias informáticas a los distintos ámbitos de la industria, mediante el tratamiento automático de la información proveniente de los procesos industriales, utilizando ordenadores que hacen más eficiente y precisa la correcta toma de decisiones. En la industria se usan regularmente los sistemas SCADA, dichos sistemas cuentan con las funcionalidades de control, supervisión y adquisición de datos. Recopilan datos de sensores, permiten comunicarse mediante señales de entrada y salida con los dispositivos de campo (sensores, actuadores, controladores) para controlar los procesos, los cuales son configurados por el usuario y pueden ser modificados con facilidad.

Es común en sistemas SCADA, tener una arquitectura distribuida en diferentes módulos o subsistemas funcionales, entre ellos se pueden encontrar los de adquisición y procesamiento de datos, comunicaciones, configuración, aplicaciones, seguridad, base de datos históricos, manejadores e interfaz hombre máquina (HMI, por sus siglas en inglés, *Human Machine Interface*). Debido a la naturaleza continua de los procesos que controlan los sistemas SCADA, estos deben de funcionar las 24 horas del día, los 7 días de la semana, ofreciendo una alta disponibilidad, como los utilizados en centrales nucleares o centrales eléctricas. La información gestionada por estos sistemas se archiva de forma digital en bases de datos históricos para su posterior estudio y análisis.

Como parte del compromiso de la Universidad con la Revolución han surgido varios centros de desarrollo especializados en diferentes dominios de aplicación, entre ellos el Centro de Informática Industrial (CEDIN). La misión principal del centro es desarrollar productos y servicios informáticos de automatización industrial, con un alto valor agregado y que cumplan las necesidades y expectativas de los clientes, potenciando la formación especializada y la investigación. (1) El mismo se encarga del desarrollo de proyectos para elevar los procesos de automatización de las empresas de todo el país y aportar productos informáticos de alto nivel para la exportación.

Dentro de los productos desarrollados por el CEDIN se encuentra el SCADA SAINUX, un software que integra las funcionalidades de alto nivel que permiten la solución de aplicaciones de supervisión y control de procesos, utilizando para ello una arquitectura distribuida por módulos que facilita la adición de nuevas funcionalidades, lo que contribuye a convertirse de manera incremental en una aplicación de gran envergadura que puede ser desplegada en la industria nacional. Como consecuencia de las necesidades existentes en empresas cubanas se desarrolla este producto que incorpora características propias y adaptables al sistema industrial nacional, soportado sobre tecnologías avanzadas en software libre, posibilitando así la soberanía tecnológica del país.

Como parte de la operación de los sistemas SCADA es importante llevar registros de los eventos que ocurren dentro y fuera del sistema, frecuentemente denominados registro de operaciones o bitácoras de operaciones. Frecuentemente los sistemas informáticos usan bitácoras para tener constancia de la información gestionada en dicho sistema, y de esta forma tener una base de conocimientos que brinde la

posibilidad de analizar patrones en los sucesos, mejorar el tiempo de respuesta ante eventualidades y facilitar la toma de decisiones al garantizar la homogeneidad, estructura y disponibilidad de dicha información.

Muchos sistemas SCADA brindan la posibilidad a los usuarios de registrar las acciones y medidas que se ejecutan durante el monitoreo y control de los procesos, de modo que queden almacenadas y puedan consultarse ante situaciones similares. Los sucesos a registrar se corresponden con aclaratorias o explicaciones de alguna eventualidad presentada y son adicionadas por los usuarios al sistema, en el registro de eventos o bitácora, contribuyendo así a la toma de decisiones futuras. El SCADA SAINUX actualmente no ofrece esta posibilidad, por lo que los operadores y mantenedores de manera opcional utilizan documentos, hojas de cálculo u otro soporte digital o en formato duro para registrar las bitácoras, propensos a pérdida o modificación de la información, sin una clasificación o formato específico, dificultando la consulta oportuna y dilatando el tiempo de respuesta ante eventualidades.

La bitácora resultante no se gestiona como parte de la información que maneja el sistema, por lo tanto, no se pueda utilizar como base de conocimientos para el tratamiento de situaciones similares, no se puede acceder a la información de forma rápida, esto induce una pérdida de tiempo considerable, afectando la velocidad de respuesta ante un suceso. La información no puede ser auditada y no se puede hacer un respaldo o backup en casos de mantenimiento o actualizaciones. En caso de que se repita un evento se debe analizar toda la situación desde cero provocando lentitud e inseguridad en la toma de decisiones.

La falta de estructura y clasificación de la información provocan que sea muy difícil para los supervisores consultar las bitácoras y entender con exactitud los eventos a los que hacen referencia los operadores, lo que puede ocasionar malentendidos y por consiguiente, tomar acciones incorrectas, lo que provocaría pérdidas y que se comprometan recursos innecesarios a dar respuesta a situaciones inexistentes o equivocadas. La falta de homogeneidad en la información procesada imposibilita realizar búsquedas por criterios o filtros específicos. Como la información se procesa en documentos aislados del sistema no se garantiza la integridad de la misma ya que puede ser modificada o eliminada con facilidad.

Teniendo en cuenta lo antes expuesto se define el siguiente **problema a resolver**: ¿Cómo garantizar la homogeneidad y disponibilidad de las bitácoras de operaciones en el SCADA SAINUX? Teniéndose como **objeto de estudio**: el proceso de gestión de bitácoras en sistemas informáticos y **campo de acción**: el proceso de gestión de bitácoras de operaciones en sistemas SCADA. Con el **objetivo general** de desarrollar un subsistema que permita la gestión de bitácoras de operaciones en el SCADA SAINUX.

Se tiene como **idea a defender** que: Con la integración del subsistema de bitácoras de operaciones en el HMI del SCADA SAINUX, se podrán registrar los sucesos detectados por los operadores en dicho sistema y garantizar la homogeneidad y disponibilidad de la información.

Para dar cumplimiento al objetivo trazado en esta investigación, se proponen las siguientes tareas de investigación:

1. Investigación de sistemas informáticos que utilizan bitácoras para caracterizar sus funcionalidades.
2. Estudio de bibliografía referente a sistemas SCADA, bitácoras y HMI para definir conceptos básicos de la investigación.
3. Identificación de los requisitos funcionales y no funcionales para el diseño e implementación del subsistema.
4. Diseño del subsistema de bitácoras de operaciones según especificación de requisitos.
5. Implementación del subsistema de bitácoras de operaciones según diseño definido.
6. Documentación del código fuente siguiendo los estándares definidos para facilitar el mantenimiento y reutilización posterior del código.
7. Diseño y ejecución de pruebas para determinar errores en su funcionamiento.
8. Resolución de no conformidades para eliminar los errores encontrados en las pruebas.

Con el apoyo de los siguientes **métodos científicos**:

Métodos Teóricos

Método histórico-lógico: Para la comprensión de los antecedentes y las tendencias actuales de los sistemas SCADA.

Método analítico-sintético: Para la definición de las funcionalidades necesarias para el subsistema de bitácoras de operaciones del SCADA SAINUX.

Métodos Empíricos

Consulta de información en fuentes confiables: para la elaboración del marco teórico de la investigación.

Criterio de expertos: Para adquirir conocimientos referentes al tema y continuar con la investigación, mediante entrevistas a expertos y especialistas en el tema, cuya experiencia y opiniones, pueden ser de una valiosa contribución para el desarrollo del trabajo.

La investigación tendrá la siguiente estructura:

Capítulo 1: Fundamentación teórica.

En este capítulo se definen los conceptos y principios utilizados durante toda la investigación, y se presentan los argumentos teóricos que responden a las técnicas a emplear y el porqué de la necesidad de usar un subsistema de bitácora en el SCADA SAINUX.

Capítulo 2: Características y diseño del sistema.

En este capítulo se da a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo. Tomándose como base la metodología AUP para guiar el proceso de desarrollo.

Capítulo 3: Implementación y pruebas del sistema.

Este capítulo está dedicado a algunos aspectos de la etapa de implementación como el estándar de codificación y a la realización de pruebas al subsistema de bitácoras de operaciones para el SCADA SAINUX.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

El siguiente capítulo aborda temas relacionados con los conceptos más importantes abordados durante la investigación, así como el lenguaje y herramientas para el diseño y desarrollo del subsistema de bitácoras de operaciones.

1.1.1 Automatización Industrial

Se define como automatización industrial a la aplicación de distintas tecnologías para controlar y monitorizar un proceso, máquina, aparato o dispositivo que por lo regular cumple tareas o funciones repetitivas, haciendo que operen automáticamente, reduciendo al mínimo la intervención humana. Lo que se busca con la automatización industrial es generar la mayor cantidad de producto, en el menor tiempo posible, con el fin de reducir los costos y garantizar una uniformidad en la calidad. (2)

La automatización industrial se encuentra presente en muchos sectores de la economía, como en la fabricación de alimentos, productos farmacéuticos, productos químicos, en la industria gráfica, petrolera, automotriz, plásticos, telecomunicaciones, entre otros sectores los cuales generan grandes beneficios. No solo se aplica a máquinas o fabricación de productos, también se aplica la gestión de procesos, de servicios, a manejo de información y a mejorar cualquier proceso que conlleve un desempeño más eficiente.

1.1.2 SCADA

Un elemento fundamental dentro de la automatización industrial lo constituyen los sistemas SCADA. Un SCADA es una aplicación de software diseñada para el control y la supervisión de los procesos industriales a distancia, facilitando la retroalimentación en tiempo real de los dispositivos de campo y el control automático de los procesos. Provee de toda la información que se genera en un proceso productivo, convirtiéndose en un instrumento esencial para la toma de decisiones. (3)

Aunque en algunos casos la arquitectura puede ser centralizada, generalmente cuentan con una arquitectura distribuida posibilitando su utilización en áreas geográficamente distantes, por lo que las unidades de adquisición de datos y control se emplean de forma remota facilitando la retroalimentación entre los dispositivos de campo y el sistema. Para ello utilizan medios de comunicación diversos tales como: radios, satélites, líneas telefónicas, conexiones directas y redes LAN¹, MAN² y WAN³. También hacen uso de una base de datos centralizada para la gestión consecuente de toda la información procesada por el sistema. (3)

Entre las funciones básicas de un sistema SCADA se encuentran las siguientes: supervisión y control remoto de instalaciones, procesamiento de información, presentación de gráficos dinámicos, generación de

¹ *Local Area Network*: Red de área Local.

² *Metropolitan Area Network*: Red de área metropolitana.

³ *Wide Area Network*: Red de área amplia.

reportes, presentación de alarmas, almacenamiento de información histórica, presentación de gráficos de tendencias y programación de eventos. (3)

1.1.3 SCADA SAINUX

SAINUX es un SCADA para el despliegue en la industria nacional, basado en tecnologías libres para su desarrollo. Este software es desarrollado por el CEDIN e integra las características y funcionalidades de alto nivel que permiten la supervisión y control de procesos, utilizando para ello una arquitectura distribuida de módulos para su eficiente funcionamiento y desarrollo. Los módulos o subsistemas funcionales que permiten las actividades de adquisición, supervisión y control de los procesos en el SCADA SAINUX son los siguientes: adquisición, configuración, comunicaciones, base de datos históricos, seguridad y HMI (*Human Machine Interface*), estas son sus características principales: (4)

Adquisición: Encargado de la recolección y procesamiento de la información proveniente de los procesos productivos. Este módulo debido a su complejidad se divide en tres capas con tareas específicas las cuales son:

- **Capa de drivers:** Actúa de forma independiente en forma de bibliotecas dinámicas, permitiendo mediante diversos protocolos la comunicación con los dispositivos de campo (Autómatas, reguladores autónomos, sensores inteligentes, controladores, entre otros).
- **Capa de recolección:** Se encarga de crear un modelo de planificación y ejecutarlo, para esto se crean las listas de puntos a consultar según una frecuencia de muestreo. La función central del recolector es obtener la información del campo.
- **Capa de procesamiento:** Recibe, procesa y distribuye hacia otros niveles (ej. Base de datos históricos y la interfaz hombre máquina) toda la información recolectada. Entre las tareas que ejecuta están las de procesar las variables calculadas, trabajo con alarmas, verificación de la calidad de los datos y la linealización de la información recolectada entre otras.

Configuración: Es el encargado de almacenar, persistir, garantizar la consistencia y suministrar la configuración a los demás módulos del sistema. Dentro de los elementos de configuración que almacena el módulo se destacan los grupos operacionales de privilegios, usuarios, perfiles, grupos de transferencia a histórico, puntos, alarmas, dispositivos, canales de comunicación, tareas, despliegues y reportes, entre otros.

Comunicaciones: Es el encargado de la comunicación entre los módulos del sistema, implementa dos formas de comunicación: síncrona y asíncrona. Permite enviar y recibir comandos y sus respuestas así como datos complejos como alarmas, eventos, estados de la comunicación y bitácoras. Además soporta la comunicación simultánea de los módulos.

Base de datos históricos: Es el encargado de almacenar la información del sistema para que posteriormente pueda ser empleada, por ejemplo, en generación de reportes, tendencias, histórico de eventos o en gestión de producción. La base de datos histórica (BDH) contendrá la información persistente de los datos recolectados de los dispositivos.

Seguridad: Proporciona las funcionalidades necesarias para garantizar el trabajo autorizado de los usuarios y constituye una fuente detectora de ataques maliciosos o no autorizados, a través del control de acceso y la autenticación.

Interfaz Hombre Máquina (HMI): Se encarga de mostrar al usuario, de forma amigable, a través de imágenes, objetos animados y tablas los procesos que ocurren en el campo en tiempo real y los componentes implicados. Está compuesto por dos partes fundamentales: el ambiente de configuración y el ambiente de ejecución.

1.1.4 HMI

HMI por sus siglas en inglés "*Human Machine Interface*", es el dispositivo o sistema que permite la interacción entre la persona y la máquina. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores y comandos, tales como luces pilotos, indicadores digitales y analógicos, registradores, pulsadores, selectores y otros que se interconectaban con la máquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementados con controladores y otros dispositivos electrónicos que dejan disponibles puertas de comunicación, es posible contar con sistemas de HMI bastante poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o máquinas. (5)

Las HMI en los sistemas SCADA ofrecen herramientas para la configuración y la visualización, posibilitan la supervisión de los procesos al presentar al usuario información en diferentes formas, ya sea gráfica o tabular y también permite ejercer control sobre los procesos o ejecutar acciones en el sistema en tiempo real. Como resultado de la operación del usuario se generan eventos, algunos de los cuales se registran de manera automática, sin embargo, otros deben ser registrados manualmente reflejando las acciones ejecutadas por el operador ante determinado suceso o situación mientras operaba el sistema o situaciones fuera de este, lo que comúnmente se conoce como bitácora. (5)

1.1.5 Bitácoras

Una bitácora es un registro escrito de las acciones que se llevaron a cabo en cierto sistema, trabajo o tarea. Incluye todos los sucesos que tuvieron lugar, las fallas que se produjeron, los cambios que se introdujeron y los costos que ocasionaron. (6) Existen varios tipos de bitácoras, entre ellos se encuentran: bitácoras de usuarios, procesos, operaciones, sistema, entre otros. Generalmente se registra la información de usuarios, contraseñas, departamentos, permisos, dominios, supervisores, fecha, hora, lugar donde ocurrió el suceso, descripción de los sucesos, notas y otros detalles. (7)

Algunas ventajas del uso de las bitácoras en sistemas informáticos de forma general son:

- La información estructurada y homogénea: la información se presenta teniendo en cuenta una estructura para facilitar su consulta y clasificación.
- Disponibilidad de la información: la información procesada se encuentra disponible en todo momento para ser consultada y analizada en caso de auditorías o comprobaciones de seguridad.

- Facilita la toma de decisiones: en caso de ocurrencia de un evento se puede comprobar si anteriormente ocurrió un suceso parecido o que implique los mismos recursos y se toma una decisión para su solución basándose en la información brindada por la aplicación.
- Contribuye a mejorar el tiempo de respuesta ante eventualidades: si la información almacenada evidencia un resultado exitoso ante el tratamiento de eventualidades pueden repetirse las acciones frente a situaciones similares, de ese modo se minimiza el tiempo de análisis y respuesta.

Estos sistemas también contienen vulnerabilidades y existen desventajas en su uso, por ejemplo: La falta de flexibilidad: generalmente no se puede hacer modificaciones capitales en la plantilla de la aplicación. Problemas para migrar: por lo general hacer una migración es bastante engorroso.

1.1.6 Bitácoras en sistemas informáticos

En el trabajo publicado por el Mtro. Gustavo Reyes Hernández “Crear una bitácora en MySQL”, se utiliza fecha y hora, host origen, usuario y actividad realizada como principales parámetros en el registro de bitácoras. (8) En el sitio *Revolution en la nube* se encuentra publicado un artículo sobre las diferencias entre la bitácora del empleado, del sistema y de procesos. En la bitácora del empleado se muestran los cambios que se le han hecho a un empleado en particular y se filtra por fecha. Se puede apreciar el uso de los parámetros fecha, usuario, acción y observaciones. (9)

Fecha	Usuario	Acción	Observaciones
22/Ago/2011 17:04 p.m.	Admin	Modificar Fotografia	
11/Ago/2011 12:12 p.m.	Admin	Deshacer último cambio Crédito INFONAVIT	
11/Ago/2011 12:11 p.m.	Admin	Activar Crédito INFONAVIT	
05/Ago/2011 15:16 p.m.	Admin	Borrar Checada	
05/Ago/2011 15:16 p.m.	Admin	Modificar Checada	
05/Ago/2011 15:15 p.m.	Admin	Agregar Checada	
05/Ago/2011 08:51 a.m.	Admin	Calcular Promedio de Variables	
02/Ago/2011 10:25 a.m.	Admin	Cancelar Baja de Herramienta	
02/Ago/2011 10:25 a.m.	Admin	Registrar Baja de Herramienta	
02/Ago/2011 10:24 a.m.	Admin	Cancelar Baja de Herramienta	
02/Ago/2011 10:23 a.m.	Admin	Registrar Baja de Herramienta	
02/Ago/2011 10:22 a.m.	Admin	Cancelar Baja de Herramienta	
02/Ago/2011 10:22 a.m.	Admin	Registrar Baja de Herramienta	
02/Ago/2011 09:59 a.m.	Admin	Modificar Forma de Pago	
01/Ago/2011 16:09 p.m.	Admin	Modificar Forma de Pago	

Figura 1: Bitácora del empleado (9)

En la bitácora del sistema se muestra toda la actividad que los usuarios han realizado sobre el sistema. Se tienen como principales parámetros: fecha, usuario, resumen, objeto, verbo y empleado. Además, para el filtrado de la información se parte de una fecha inicial y una fecha final.

Bitácora de Sistema

Acciones ▾

Clase: Aplicación ▾

Fecha Inicial: 02/Ago/2010 [calendar] ←

Fecha Final: 06/Ago/2010 [calendar] ←

Tipo: Todos ▾

Usuario:

Empleado:

Resumen:

Objeto:

Verbo:

Filtrar

Tipo	Fecha	Usuario	Resumen	Objeto	Verbo	Empleado
Informativo	12/Ago/2010	Ana Aguirre	Consultar Bitácora de Sistema	Bitácora de Sistema	Consultar Bitácora de Sistema	Murillo Hernández, María
Informativo	12/Ago/2010	Ana Aguirre	Consultar Empresas	Empresa	Consultar Empresas	Murillo Hernández, María
Informativo	12/Ago/2010	Ana Aguirre	Consultar Lista de Nómina por Periodo	Nómina	Consultar Lista de Nómina por Periodo	Murillo Hernández, María
Informativo	12/Ago/2010	Ana Aguirre	Agregar Excepción de Prenómina	Excepción de Nómina	Agregar Excepción de Prenómina	Hernández Nuño, Dario

Figura 2: Bitácora del sistema. (9)

La bitácora de procesos permite consultar todos los procesos ejecutados por los usuarios de un sistema. Los parámetros empleados en este tipo de bitácora son: nombre del proceso, fecha de inicio, fecha final, estado, avance y usuario. (9)

Bitácora de Procesos

Acciones ▾

Fecha Inicial de: 01/Jun/2011 [calendar]

a: 08/Jun/2011 [calendar]

Proceso:

Mis procesos:

Usuario:

Estatus: Todos ▾

Filtrar

Proceso	Inicio	Final	Estatus	Avance	Usuario
Calcular Nómina	08/Jun/2011 09:46 a.m.	08/Jun/2011 09:46 a.m.	Terminado	100%	Admin
Calcular Nómina	08/Jun/2011 09:45 a.m.	08/Jun/2011 09:45 a.m.	Terminado	100%	Admin
Calcular Nómina	08/Jun/2011 08:40 a.m.	08/Jun/2011 08:40 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 17:01 p.m.	07/Jun/2011 17:01 p.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 16:56 p.m.	07/Jun/2011 16:56 p.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 16:53 p.m.	07/Jun/2011 16:53 p.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 11:32 a.m.	07/Jun/2011 11:32 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 11:29 a.m.	07/Jun/2011 11:29 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 11:16 a.m.	07/Jun/2011 11:16 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 11:08 a.m.	07/Jun/2011 11:09 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 11:05 a.m.	07/Jun/2011 11:05 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 10:51 a.m.	07/Jun/2011 10:51 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 10:24 a.m.	07/Jun/2011 10:24 a.m.	Terminado	100%	Admin
Calcular Nómina	07/Jun/2011 10:23 a.m.	07/Jun/2011 10:23 a.m.	Terminado	100%	Admin

Figura 3: Bitácora de procesos. (9)

En el sitio Foros del Web, en un post sobre bitácoras de usuario se registran las transacciones y operaciones realizadas por los usuarios mientras usan un sistema y se definen los siguientes parámetros en las bitácoras: fecha, id del usuario, módulo, acción, id del proyecto, detalles y la dirección IP de donde se accede al sistema. (10)

Luego de consultar las fuentes referenciadas se puede llegar a la conclusión de que en las bitácoras, sin especificar un tipo, generalmente son utilizados los parámetros fecha y hora, usuario que registra la bitácora, descripción del evento o detalles del mismo y el recurso o usuario del sistema implicado. Para el uso en el subsistema de bitácora de operaciones para el SCADA SAINUX, en el registro de sucesos, se definió de conjunto con los especialistas funcionales utilizar los parámetros: fecha y hora, usuario, descripción, recurso asociado, y para la clasificación de los sucesos: grupos y tipos de sucesos. Para la consulta de la información registrada en el subsistema se definió la utilización, primeramente, de un filtrado dado una fecha inicial y una fecha final, además de proporcionarle al usuario la posibilidad de filtrar esos resultados mediante los parámetros que poseen los sucesos.

1.2 Metodología, tecnologías y herramientas

A continuación se realiza una valoración de las principales tecnologías y herramientas que serán empleadas en la solución del problema planteado, definiendo sus características fundamentales. En algunos casos, la selección está fundamentada en su utilización por el proyecto SCADA SAINUX en la implementación de sus productos y funcionalidades.

1.2.1 Metodología de desarrollo de software

El papel de la metodología de desarrollo empleada en un proyecto es sin duda esencial, ya que se debe acoplar en el equipo de desarrollo, guiar y organizar actividades que conlleven a las metas trazadas por el grupo.

El Proceso Unificado Ágil (AUP, por sus siglas en inglés, *Agile Unified Process*), es una versión simplificada del Proceso de Desarrollo Unificado (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo: el desarrollo dirigido por pruebas (TDD por sus siglas en inglés, *Test Driven Development*), el modelado ágil, la gestión de cambios ágil, y la refactorización de base de datos para mejorar la productividad. (11)

Con el fin de mejorar la actividad productiva de la UCI se realizaron algunas variaciones en la metodología de desarrollo AUP, para ajustarla al entorno de producción de la universidad, por tanto, la metodología de desarrollo a usar en este trabajo es AUP en su variación UCI.

Fases de AUP en su variación UCI: (11)

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

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 esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente.

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

Las disciplinas de AUP en su variación UCI se muestran a continuación: (11)

Modelado de negocio (opcional): El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización.

Requisitos: Esta disciplina comprende el desarrollo de un modelo del sistema que se va a construir, además de la administración y gestión de los requisitos funcionales y no funcionales del producto.

Análisis y diseño: En esta disciplina, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema. Se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales.

Implementación: En la implementación, a partir de los resultados del Análisis y Diseño se desarrolla el sistema.

Pruebas internas: En esta disciplina se verifica el resultado de la implementación probando cada parte desarrollada. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible, componentes de pruebas ejecutables para automatizar las pruebas.

Pruebas de liberación: Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, antes de ser entregados al cliente para su aceptación.

Pruebas de aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Despliegue (opcional): Constituye la instalación, configuración, adecuación, puesta en marcha de soluciones informáticas y entrenamiento al personal del cliente.

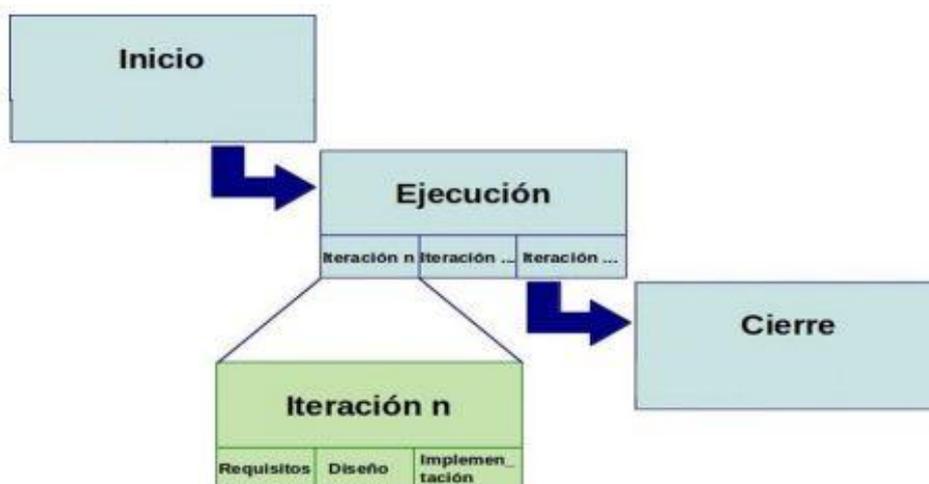


Figura 4: Fases e Iteraciones de la metodología AUP variación UCI. (11)

1.2.2 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. (12)

Está compuesto por varios elementos gráficos que se combinan mediante reglas para conformar los diferentes diagramas. Permite además la modelación de sistemas con tecnología orientada a objetos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que el modelo UML describe lo que supuestamente hará el sistema, pero no dice cómo implementar dicho sistema. (12)

Por todo lo anteriormente mencionado y teniendo en cuenta además, que para el desarrollo de la aplicación propuesta en la solución se seguirá el paradigma orientado a objeto; se ha decidido que el lenguaje de modelado a utilizar sea el UML.

1.2.3 Herramienta de modelado

En la actualidad, las tecnologías ayudan considerablemente a solucionar muchos detalles que en años previos retrasaban el desarrollo. Entre estas tecnologías se encuentran las herramientas de modelado, éstas constituyen el medio donde se modela el sistema que se desea, rigiéndose por una metodología y utilizando un lenguaje de modelado.

Una herramienta CASE (*Computer Aided Software Engineering*) es un conjunto de métodos, utilidades y técnicas que proporcionan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en algunas de sus fases. Estas herramientas engloban todos los pasos del desarrollo de software, y también aquellas actividades generales que se aplican a lo largo del proceso. (13)

Se selecciona para el modelado del sistema la herramienta CASE *Visual Paradigm* para UML 8.0, dado que es de fácil uso, con soporte multiplataforma, y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Se puede encontrar en varios idiomas, apoya un gran número de lenguajes en la generación de código y la ingeniería inversa en Java, C++, PHP, entre otros, es fácil de instalar y de actualizar y posee compatibilidad entre ediciones. Esta herramienta que está diseñada para usuarios interesados en sistemas de software de gran escala con el uso de la aproximación orientado a objeto. (13)

1.2.4 Herramientas de desarrollo

Se seleccionan las siguientes herramientas de desarrollo para la implementación del subsistema por ser las herramientas utilizadas en el desarrollo del sistema SCADA SAINUX y así lograr una mejor integración de la solución con el producto final.

Framework de desarrollo Qt 4.8.2, Ambiente de Desarrollo Integrado (IDE) Qt Creator 2.5.0

Qt es un *framework* multiplataforma de desarrollo de aplicaciones en C++, basado en la librería Qt. Se utiliza fundamentalmente para desarrollar aplicaciones con interfaz gráfica, gracias al conjunto de controles

independientes de la plataforma que ofrece. Como todos los *framework*, Qt cuenta con características que lo identifican en su uso: contiene un editor avanzado para C++, tiene integrado un diseñador de formularios de interfaces gráficas de usuario (GUI), contiene herramientas para la administración y construcción de proyectos, facilidad de completado automático, y un depurador visual. (14)

1.2.5 Sistema gestor de bases de datos

PostgreSQL: Sistema de gestión de base de datos relacional orientada a objetos y libre. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. (15) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de la última modificación. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos, lo que constituye una gran ventaja en su uso.

1.2.6 Lenguaje de programación para la implementación de la solución

El lenguaje de programación C++ está considerado por muchos como el lenguaje de programación más potente, dado que permite el trabajo tanto a alto nivel como a bajo nivel, logrando gran eficiencia en los tiempos de ejecución y bajo consumo de memoria en los programas desarrollados, aspectos que muchas aplicaciones requieren, siendo una opción factible como lenguaje a utilizar en sistemas que necesitan un alto rendimiento. Entre las principales características que brinda C++ se puede mencionar la programación orientada a objetos. La posibilidad de orientar la programación a objetos le permite al programador, diseñar aplicaciones desde un punto de vista más cercano a la vida real. Además de permitir reutilizar el código de una manera más lógica y productiva. (16)

1.3 Conclusiones parciales del capítulo

- Para dar solución a la situación problemática se investigó sobre las principales características y funcionalidades de los sistemas SCADA, los HMI y las bitácoras para comprender la necesidad de incorporar las bitácoras de operaciones en el SCADA SAINUX.
- Se investigaron estos sistemas para identificar las características y funcionalidades que puedan incorporarse en el subsistema de bitácoras de operaciones.
- También se investigaron las tecnologías y herramientas a emplear en la fase de desarrollo de la aplicación, para hacer uso de las más adecuadas para el desarrollo de este tipo de aplicación y que a la vez cumpla con las normas y objetivos de la universidad, dirigidos a la migración a software libre.

Capítulo 2: Características y diseño del sistema.

2.1 Introducción

En el presente capítulo se da a conocer la propuesta del sistema y los diagramas para apoyar la comprensión del funcionamiento del mismo. Se utiliza la metodología AUP en su variante UCI para guiar el proceso de desarrollo. Además, se presenta la arquitectura utilizada en la implementación del subsistema de bitácoras de operaciones para el SCADA SAINUX.

2.2 Modelo conceptual

Un modelo conceptual es un conjunto de conceptos y de reglas destinados a representar de forma global los aspectos lógicos de los diferentes tipos de elementos existentes en la realidad que está siendo analizada. Es una representación figurada de una experiencia empírica, que tiene como objetivo ayudar a comprender la realidad. (17)

A continuación se presenta el modelo conceptual correspondiente al subsistema de bitácoras de operaciones, así como la descripción de los conceptos involucrados.

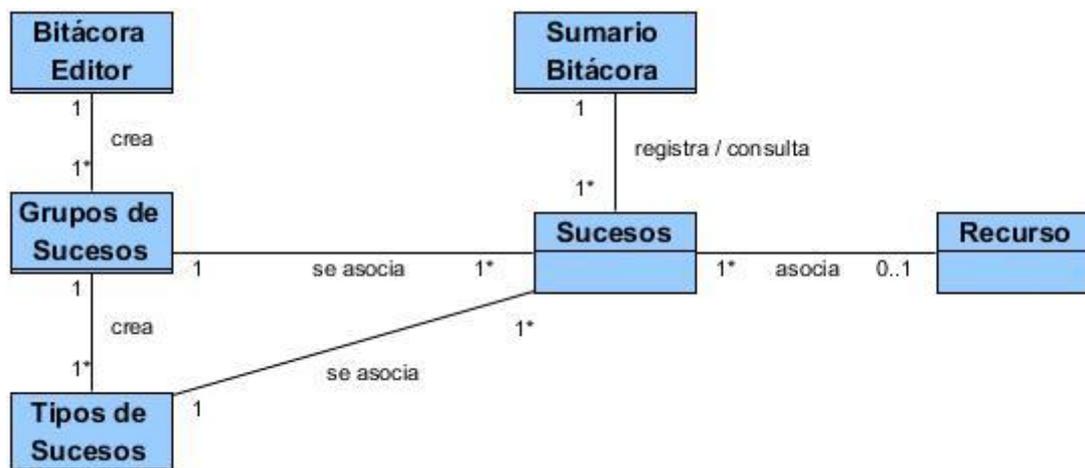


Figura 5: Modelo Conceptual.

Bitácora Editor: hace referencia al entorno de edición del módulo HMI donde se agregan, modifican y eliminan grupos y tipos de sucesos.

Sumario Bitácora: hace referencia al entorno de visualización del módulo HMI donde se registran y consultan sucesos, a los cuales se asocia un tipo y un grupo de sucesos y puede asociarse un recurso.

Recurso: un suceso puede, o no, tener asociado un recurso, algunos ejemplos de recursos son: puntos analógicos, puntos digitales, dispositivos, subcanales, reportes y alarmas.

Sucesos: un suceso tiene asociado varios componentes los cuales lo definen, estos son: grupo de sucesos, tipo de sucesos, recurso y descripción, campo en el cual el operador debe dejar como constancia una descripción completa del suceso y en caso de haber una respuesta inmediata al mismo, también debe incluirla.

Grupos de sucesos: define grupos que tienen asociados tipos de sucesos para la clasificación general de los mismos, generalmente estos grupos definen un departamento o área.

Tipos de sucesos: cada tipo de sucesos pertenece a un grupo de sucesos. Al igual que los grupos de suceso, los tipos son creados por los mantenedores en el entorno de edición del HMI y son esenciales para la clasificación de los sucesos.

2.3 Especificación de los Requisitos de Software

La especificación de requisitos comprende la descripción del comportamiento del sistema que se va a desarrollar constituyendo una base arquitectónica que con posterioridad describirá los procesos que realizará la futura aplicación que se propone. Además, de definir el funcionamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que este pueda poseer. (18)

2.3.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. En algunos casos, los requisitos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. (18) Estos son los requisitos funcionales definidos y sus descripciones:

1. Crear sucesos: el subsistema de bitácoras de operaciones debe permitir la creación de sucesos en la interfaz principal de la aplicación para lo cual se necesita una descripción del suceso y la respuesta al mismo, un grupo y un tipo de sucesos al cual pertenece el evento al que se hace referencia y se debe brindar la posibilidad de seleccionar un recurso relacionado al suceso, entre ellos se encuentran: puntos analógicos, puntos digitales, dispositivos, subcanales, despliegues y alarmas.
2. Crear grupos de sucesos: se debe permitir la creación de grupos de sucesos en el editor del SCADA, los cuales se facilitarán la clasificación y la forma de presentar los tipos de sucesos al usuario, (los grupos de sucesos son creados por los mantenedores).
3. Modificar grupos de sucesos: se debe permitir que los mantenedores modifiquen los grupos de sucesos en el editor del SCADA.
4. Eliminar grupos de sucesos: se debe permitir que los mantenedores eliminen los grupos de sucesos en el editor del SCADA.
5. Visualizar los grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones: se debe permitir la visualización de los grupos de sucesos creados con anterioridad en el editor del SCADA en la parte derecha de la interfaz principal del subsistema.
6. Crear tipos de sucesos: se debe permitir la creación de tipos de sucesos por parte de los mantenedores en el editor del SCADA, se debe relacionar cada tipo de sucesos a un grupo de sucesos creado con anterioridad en el editor.
7. Modificar tipos de sucesos: se debe permitir la modificación de los tipos de sucesos en el editor del SCADA.

8. Eliminar tipos de sucesos: se debe permitir la eliminación de los tipos de sucesos en el editor del SCADA.
9. Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones: se debe permitir la visualización de los tipos de sucesos creados con anterioridad en el editor del SCADA en la parte derecha de la interfaz principal del subsistema en forma de árbol.
10. Permitir la consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado: el subsistema debe permitir consultar todos los sucesos creados en un rango para su estudio y análisis en la interfaz principal del subsistema.

2.3.2 Requisitos no funcionales

Son aquellos requisitos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interfaz del sistema. (18) Estos son los requisitos no funcionales definidos según su clasificación:

Interfaces de usuarios

- El diseño del subsistema debe ser sencillo, funcional, fácil de utilizar, intuitivo y de rápida adaptación para los usuarios.
- Forma de Interacción de la Interfaz: Basada en ventanas.

Seguridad

- El subsistema solo será usado por usuarios autorizados a interactuar con el módulo HMI del SCADA SAINUX.

Software

- El subsistema debe ejecutarse sobre el sistema operativo GNU/Linux en su distribución Debian 7.

Hardware

- El ordenador para el uso del subsistema debe contar con no menos de 1 GB de memoria RAM.
- Procesador Intel Core i3 multinúcleo con velocidad igual o superior a 2.0 Ghz.
- Disco duro interno, de al menos 80 GB SATA 3.0 Gb/s 7200 RPM.
- 2 Tarjetas de Red PCIe Ethernet Gigabit de 10/100/1000 Mbps. (19)

Funcionamiento

- Tiempo para consultas de bitácoras almacenadas en base de datos:
 - Consulta sencilla de 1 a 100 registros: hasta 2 segundos.
 - Consulta media de 101 a 9999 registros: hasta 10 segundos.
 - Consultas complejas de 10.000 registros en adelante: mayor a 10 segundos.

Implementación

- En el desarrollo del subsistema deben utilizarse tecnologías libres.

- El sistema debe regirse por el siguiente estándar internacional:
 - *Institute of Electrical and Electronic Engineers (IEEE): IEEE C37.1-1994 "Standard Definition, Specification and Analysis of Systems used for Supervisory Control, Data Acquisition and Automatic Control"* (ANSI/IEEE).

Usabilidad

- El subsistema debe integrarse al módulo HMI del SCADA SAINUX.

2.4 Propuesta de solución

La propuesta de solución consiste en un subsistema que permite registrar la información de los sucesos ocurridos dentro y fuera del sistema de forma dinámica e intuitiva, para que los operadores puedan dejar constancia de lo ocurrido en sus turnos de guardia y que la información esté estructurada, sea homogénea y esté disponible en cualquier momento que se necesite.

La información se almacena en la base de datos históricos del SCADA SAINUX usando la interfaz para el envío de eventos que provee la capa de comunicaciones:

```
int sendEvent(Event * data /**<arreglo de Eventos a enviar*/,  
             size_t arraySize /**<cantidad de objetos Event en el arreglo*/,  
             ChannelId channelId /**canal por el que se enviará la data*/);
```

Figura 6: Función Send Event.

Para acceder a la información procesada por el subsistema, almacenada en la base de datos históricos (BDH) del SCADA SAINUX, se hace uso de la biblioteca libqt4-sql-psql del framework Qt que permite acceder a información almacenada en gestores de bases de datos PostgreSQL.

Los sucesos tienen asociados el usuario que lo agregó al sistema, la fecha y hora en que se registró el suceso, una descripción de lo ocurrido, un tipo y un grupo de sucesos y puede tener asociado un recurso. Para la clasificación de los sucesos serán gestionados tipos y grupos de sucesos en el entorno de edición, los cuales serán presentados a los operadores en la interfaz principal del subsistema en el entorno de visualización, haciendo uso del módulo de configuración para lograr esta interacción.

El subsistema permite la consulta de bitácoras (sucesos) dado una fecha inicial y final, así como palabras clave y otros parámetros relacionados directamente con los sucesos para facilitar el filtrado de la información. Para un mejor entendimiento de la propuesta de solución descrita se muestra la siguiente figura.

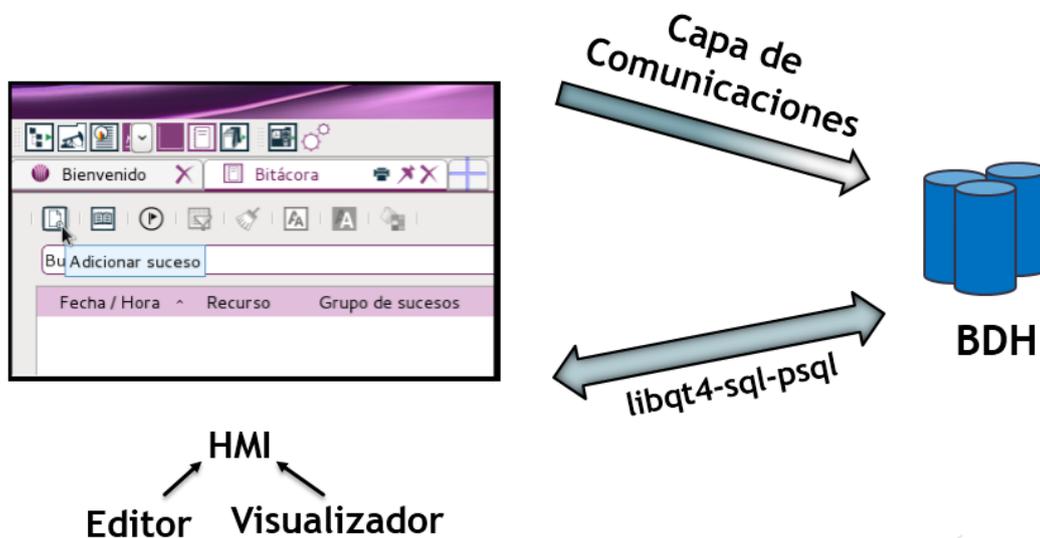


Figura 7: Propuesta de solución.

2.5 Historias de Usuario

Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos, son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, también permiten responder rápidamente a los requisitos cambiantes. (20) Estas son las historias de usuario definidas en la fase de análisis, para el diseño e implementación de la solución:

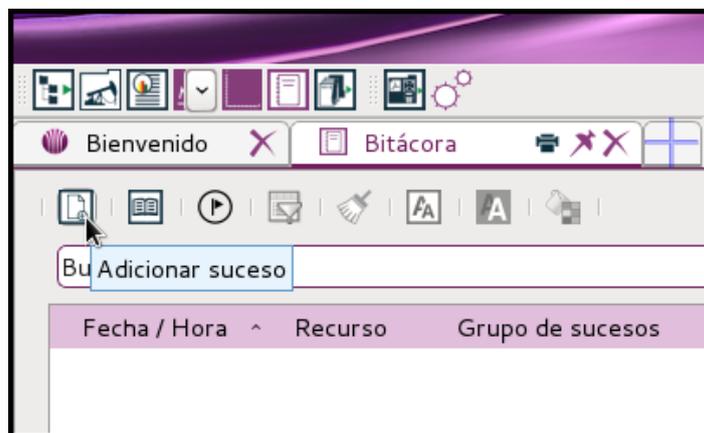
2.5.1 Descripción de las Historias de Usuario

Historia de Usuario		
Número: HU1	Nombre del requisito: Adicionar suceso.	
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1	
Prioridad: Alta	Tiempo Estimado: 2 semanas.	
Riesgo en Desarrollo: Problemas eléctricos. Problemas técnicos.	Tiempo Real: 2 semanas.	
	Referencias: HU2, HU6.	
Descripción: El operador será capaz de adicionar un suceso al subsistema de bitácoras de operaciones a través del botón Adicionar suceso. Se debe seleccionar grupo de sucesos, tipo de sucesos y redactar una descripción del suceso. Se puede seleccionar un recurso asociado al suceso (opcional). Luego presionar el botón Aceptar y se almacena la siguiente información.		
Campo	Tipo de dato	Reglas o restricciones
Fecha y hora	date	No nulo
Id del usuario	int	No nulo

Id del grupo de sucesos	int	No nulo
Id del tipo de sucesos	int	No nulo
Descripción del suceso	string	No nulo Máximo 256 caracteres
Id del recurso asociado	int	-

Observaciones:

Prototipo de interfaz:



Adicionar suceso

Seleccione grupo de sucesos Tiene recurso asociado

Seleccione un grupo de sucesos ...

Seleccione tipo de suceso
 Seleccione un tipo de sucesos ...

Descripción: (256 caracteres)

Recursos

- ▷ Puntos analógicos
- ▷ Puntos digitales
- ▷ Dispositivos
- ▷ Subcanales
- ▷ Despliegues
- ▷ Reportes
- ▷ Alarmas

Recurso seleccionado

Tabla 1: Historia de Usuario Adicionar suceso.

Historia de Usuario	
Número: HU2	Nombre del requisito: Agregar grupo de sucesos.

Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: Problemas eléctricos. Problemas técnicos.	Tiempo Real: 1 semana Referencias: -

Descripción: El mantenedor será capaz de adicionar un grupo de sucesos en el editor del SCADA SAINUX dentro del ítem Bitácoras en el módulo HMI. El grupo de sucesos va a tener asociados tipos de sucesos. Para crear un grupo de sucesos se debe presionar el botón secundario del ratón sobre el ítem Bitácoras y seleccionar la opción Agregar Grupo de sucesos, luego aparece una ventana con los campos para crear un grupo de sucesos, se debe escribir el nombre del grupo de sucesos y redactar una descripción del mismo (opcional), al presionar el botón Aceptar se almacena la siguiente información.

Campo	Tipo de dato	Reglas o restricciones
Id del grupo de sucesos	int	No nulo
Nombre del grupo de sucesos	string	No nulo No repetido
Descripción del grupo de sucesos	string	Máximo 256 caracteres

Observaciones:

Prototipo de interfaz:

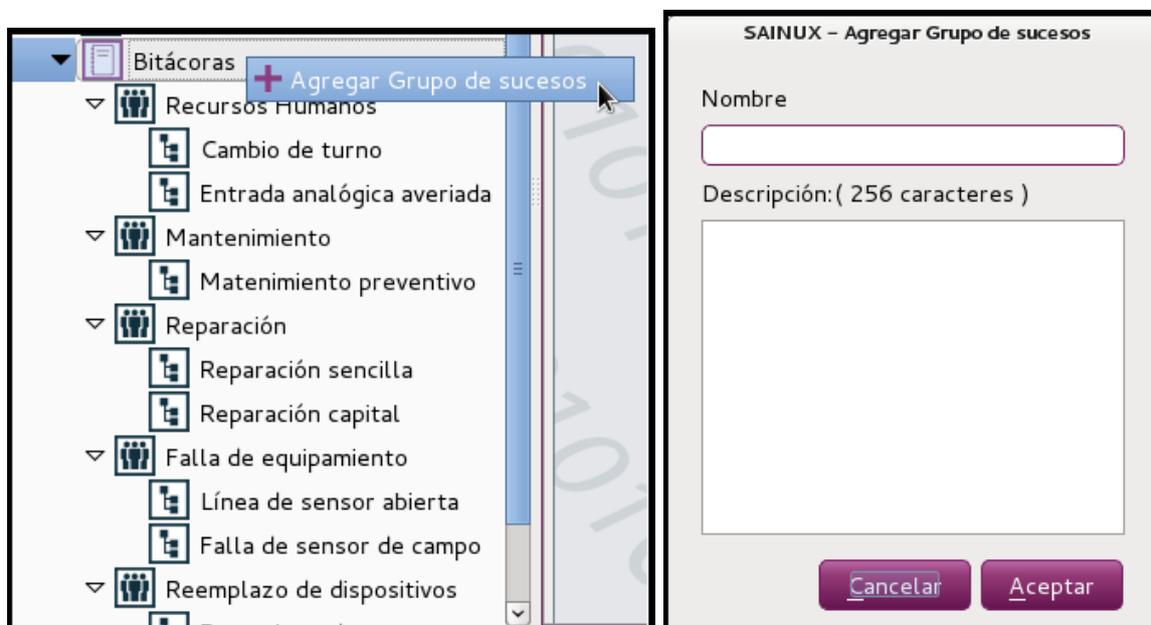


Tabla 2: Historia de Usuario Agregar grupo de sucesos.

Historia de Usuario										
Número: HU3	Nombre del requisito: Modificar grupo de sucesos.									
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1									
Prioridad: Media	Tiempo Estimado: 1 día									
Riesgo en Desarrollo:	Tiempo Real: 12 horas									
Problemas eléctricos.	Referencias: HU2.									
Problemas técnicos.										
<p>Descripción: El mantenedor será capaz de modificar un grupo de sucesos en el editor del SCADA SAINUX. Al presionar el botón secundario del ratón encima de un grupo de sucesos se presentan tres opciones (Agregar tipo de sucesos, Propiedades y Eliminar), el mantenedor debe escoger la opción Propiedades, luego aparece una ventana con los campos Nombre y Descripción para su edición. Al presionar el botón Aceptar y se almacena la siguiente información.</p> <table border="1"> <thead> <tr> <th>Campo</th> <th>Tipo de dato</th> <th>Reglas o restricciones</th> </tr> </thead> <tbody> <tr> <td>Nombre del grupo de sucesos</td> <td>string</td> <td>No nulo No repetido</td> </tr> <tr> <td>Descripción del grupo de sucesos</td> <td>string</td> <td>Máximo 256 caracteres</td> </tr> </tbody> </table>		Campo	Tipo de dato	Reglas o restricciones	Nombre del grupo de sucesos	string	No nulo No repetido	Descripción del grupo de sucesos	string	Máximo 256 caracteres
Campo	Tipo de dato	Reglas o restricciones								
Nombre del grupo de sucesos	string	No nulo No repetido								
Descripción del grupo de sucesos	string	Máximo 256 caracteres								

Observaciones:

Prototipo de interfaz:

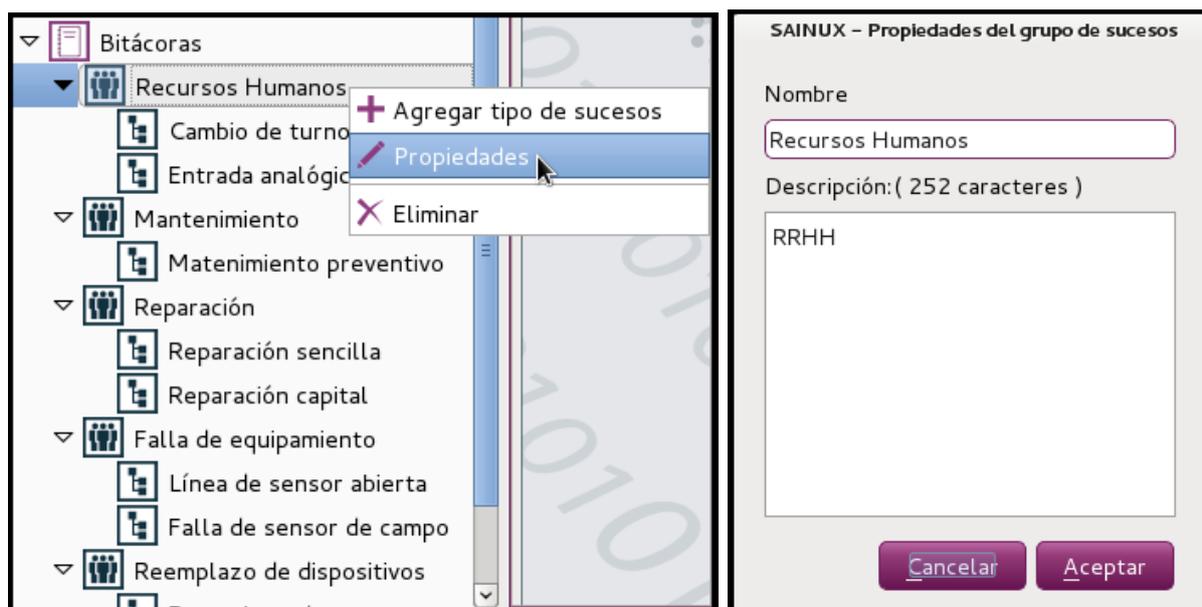


Tabla 3: Historia de Usuario Modificar grupo de sucesos.

Historia de Usuario	
Número: HU4	Nombre del requisito: Eliminar grupo de sucesos.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo: Problemas eléctricos. Problemas técnicos.	Tiempo Real: 12 horas
	Referencias: HU2.
Descripción: El mantenedor será capaz de eliminar un grupo de sucesos en el editor del SCADA SAINUX. Al presionar el botón secundario del ratón encima de un grupo de sucesos se presentan tres opciones (Agregar tipo de sucesos, Propiedades y Eliminar), el mantenedor debe escoger la opción Eliminar, luego aparece una ventana con un mensaje para confirmar la eliminación. Al presionar el botón Aceptar se elimina el grupo de sucesos.	
Observaciones:	
Prototipo de interfaz:	

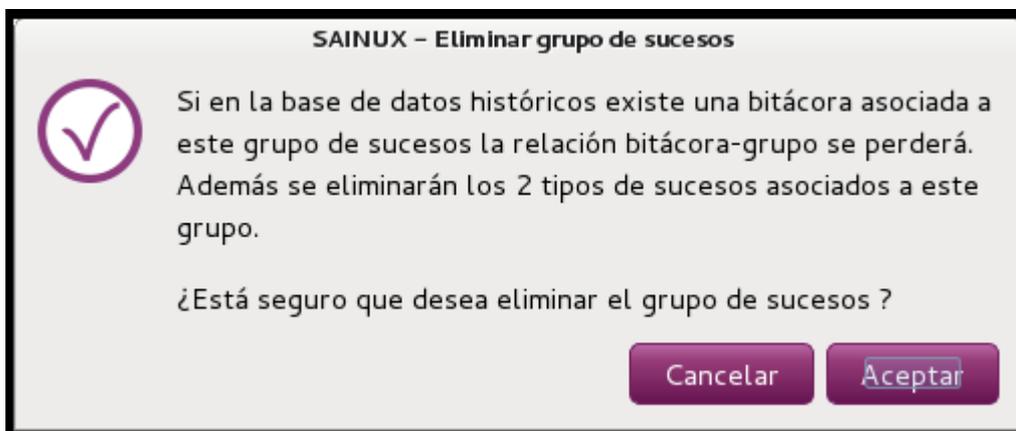
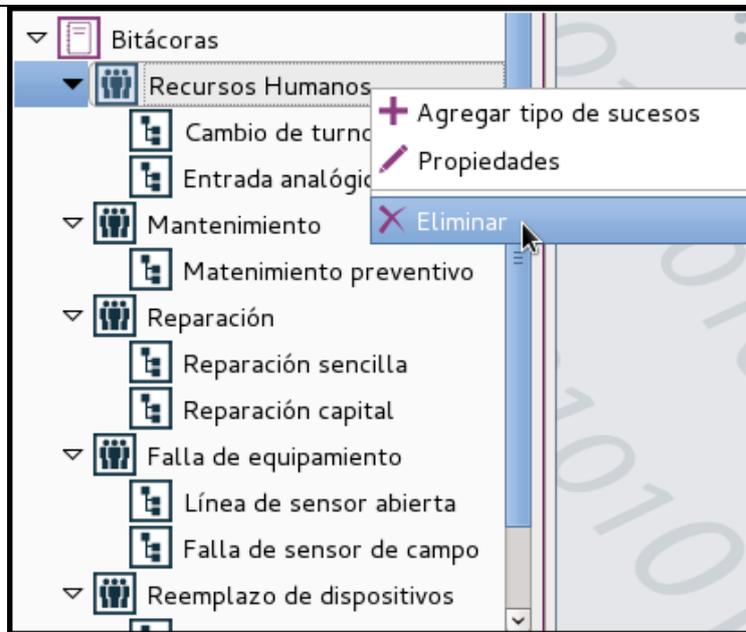


Tabla 4: Historia de Usuario Eliminar grupo de sucesos.

Historia de Usuario	
Número: HU5	Nombre del requisito: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: Problemas eléctricos. Problemas técnicos.	Tiempo Real: 1 semana Referencias: HU2.
Descripción: se debe permitir la visualización de los grupos de sucesos creados con anterioridad en el editor del SCADA en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones	

en forma de árbol. Mediante el uso del botón “Mostrar / Ocultar árbol de grupos y tipos de sucesos” en la parte superior izquierda de la interfaz principal se tiene la posibilidad de ocultar y mostrar el árbol de grupos y tipos de sucesos para un mejor aprovechamiento de la interfaz principal para la consulta de bitácoras.

Observaciones:

Prototipo de interfaz:

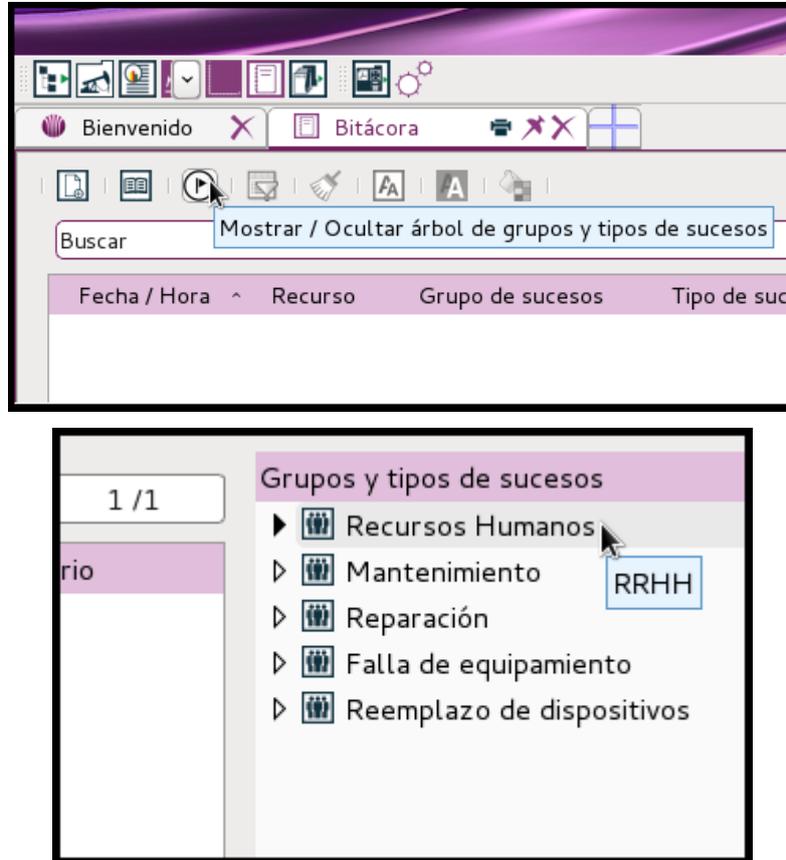


Tabla 5: Historia de Usuario Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

Historia de Usuario	
Número: HU6	Nombre del requisito: Agregar tipo de sucesos.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo:	Tiempo Real: 1 semana
Problemas eléctricos.	Referencias: HU2.
Problemas técnicos.	

Descripción: El mantenedor será capaz de adicionar un tipo de sucesos en el editor del SCADA dentro de un grupo de sucesos, los cuales se encuentran dentro del ítem Bitácoras en el módulo HMI. El tipo de sucesos va a estar asociado a un grupo de sucesos para su clasificación. Para crear un tipo de sucesos se debe presionar el botón secundario del ratón sobre un grupo de sucesos y seleccionar la opción Agregar tipo de sucesos, luego aparece una ventana con los campos Nombre y Descripción, al presionar el botón Aceptar y se almacena la siguiente información.

Campo	Tipo de dato	Reglas o restricciones
Id del tipo de sucesos	int	No nulo
Nombre del tipo de sucesos	string	No nulo No repetido en el mismo grupo
Descripción del tipo de sucesos	string	Máximo 256 caracteres

Observaciones:

Prototipo de interfaz:

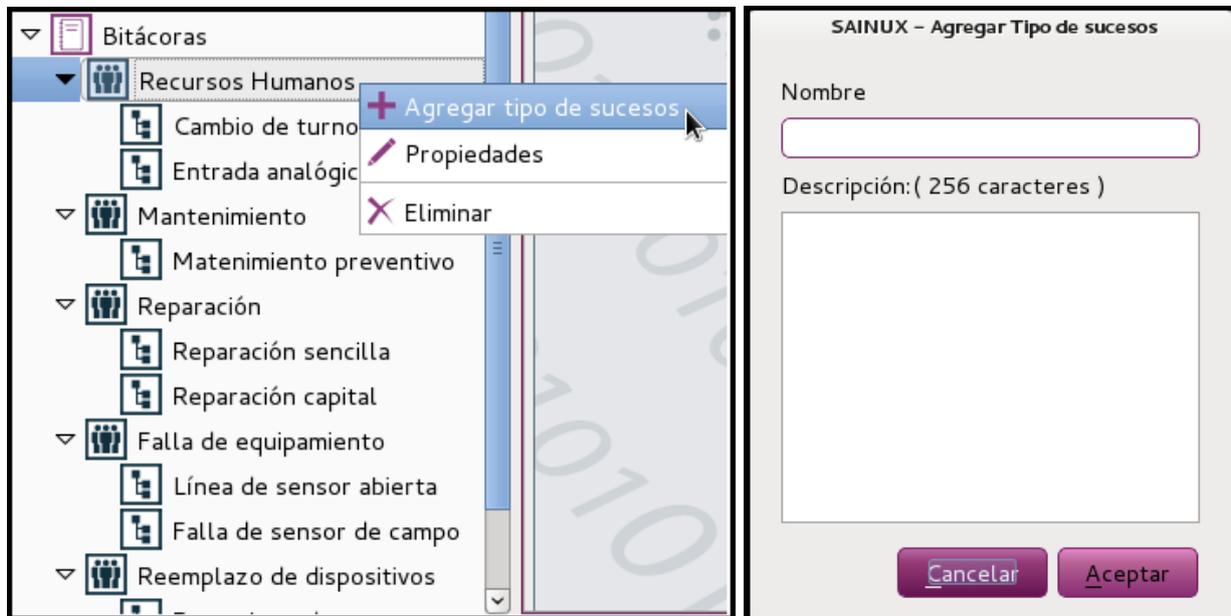


Tabla 6: Historia de Usuario Agregar tipo de sucesos.

Historia de Usuario	
Número: HU7	Nombre del requisito: Modificar tipo de sucesos.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo:	Tiempo Real: 12 horas

Problemas eléctricos.

Referencias: HU6.

Problemas técnicos.

Descripción: El mantenedor será capaz de modificar un tipo de sucesos en el editor del SCADA SAINUX. Al presionar el botón secundario del ratón encima de un tipo de sucesos se presentan dos opciones (Propiedades y Eliminar), el mantenedor debe escoger la opción Propiedades donde se muestran las propiedades del tipo de sucesos en una ventana con los campos Nombre y Descripción para su edición. Al presionar el botón Aceptar y se almacena la siguiente información.

Campo	Tipo de dato	Reglas o restricciones
Nombre del tipo de sucesos	string	No nulo No repetido en el mismo grupo
Descripción del tipo de sucesos	string	Máximo 256 caracteres

Observaciones:

Prototipo de interfaz:

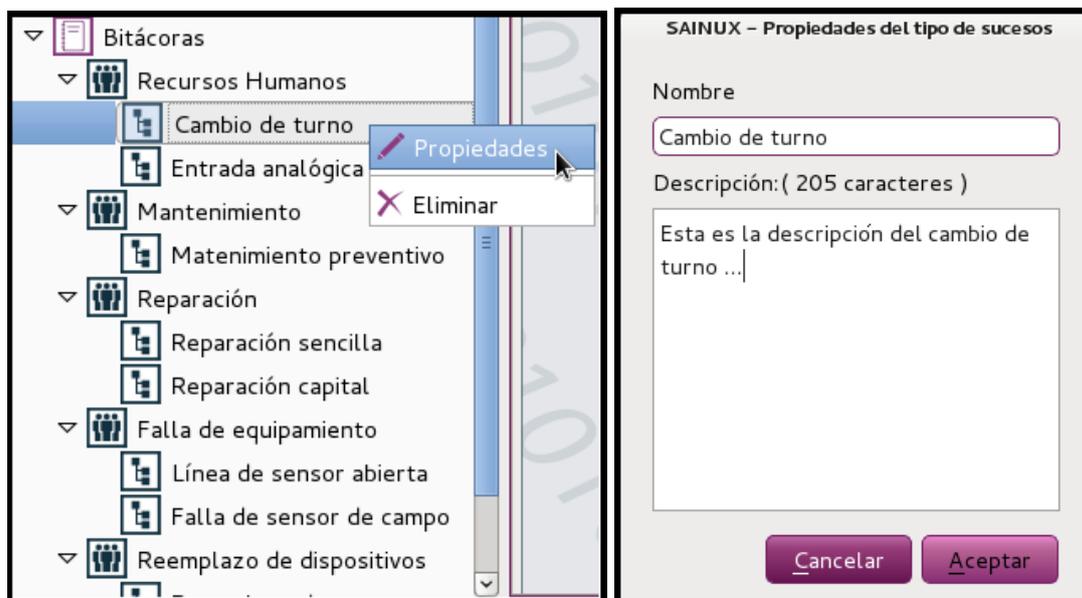


Tabla 7: Historia de Usuario Modificar tipo de sucesos.

Historia de Usuario	
Número: HU8	Nombre del requisito: Eliminar tipo de sucesos.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 1 día
Riesgo en Desarrollo:	Tiempo Real: 12 horas

Problemas eléctricos.

Referencias: HU6.

Problemas técnicos.

Descripción: El mantenedor será capaz de eliminar un tipo de sucesos en el editor del SCADA SAINUX. Al presionar el botón secundario del ratón encima de un tipo de sucesos se presentan dos opciones (Propiedades y Eliminar), el mantenedor debe escoger la opción Eliminar, luego aparece una ventana con un mensaje para confirmar la eliminación, al presionar el botón Aceptar se elimina el tipo de sucesos.

Observaciones:

Prototipo de interfaz:

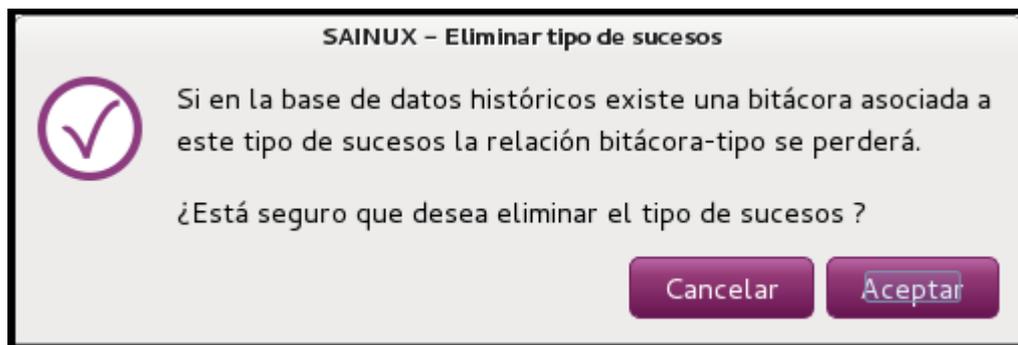
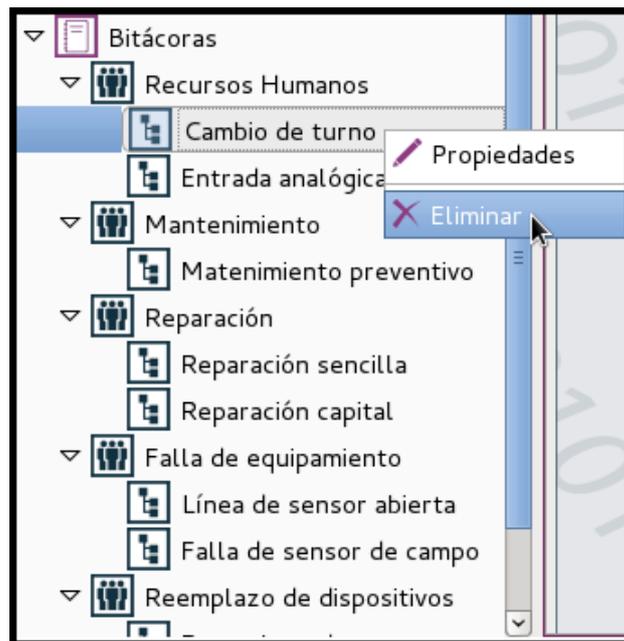


Tabla 8: Historia de Usuario Eliminar tipo de sucesos.

Historia de Usuario	
Número: HU9	Nombre del requisito: Visualizar tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: Problemas eléctricos. Problemas técnicos.	Tiempo Real: 1 semana Referencias: HU2, HU6.
Descripción: se debe permitir la visualización de los tipos de sucesos creados con anterioridad en el editor del SCADA en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones en forma de árbol. Mediante el uso del botón “Mostrar / Ocultar árbol de grupos y tipos de sucesos” en la parte superior izquierda de la interfaz principal se tiene la posibilidad de ocultar y mostrar el árbol de grupos y tipos de sucesos para un mejor aprovechamiento de la interfaz principal para la consulta de bitácoras.	
Observaciones:	
Prototipo de interfaz:	

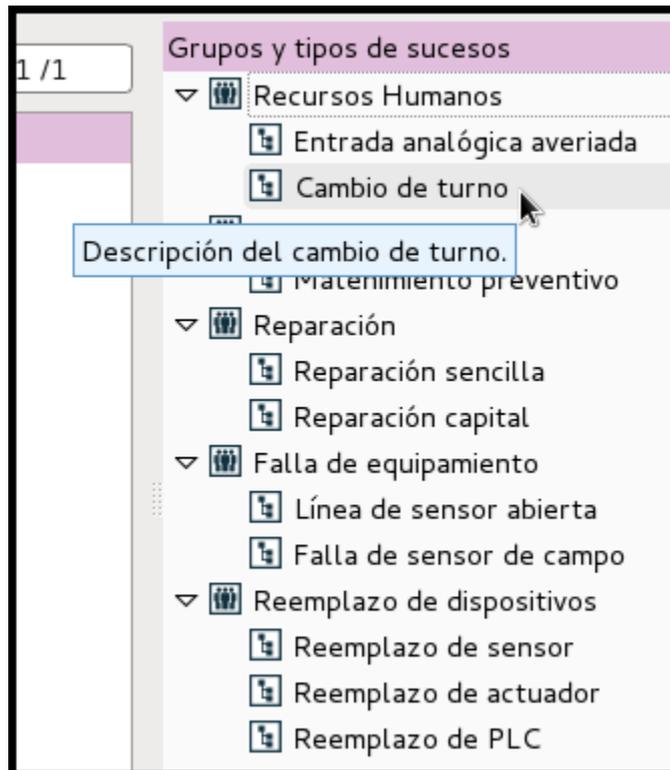
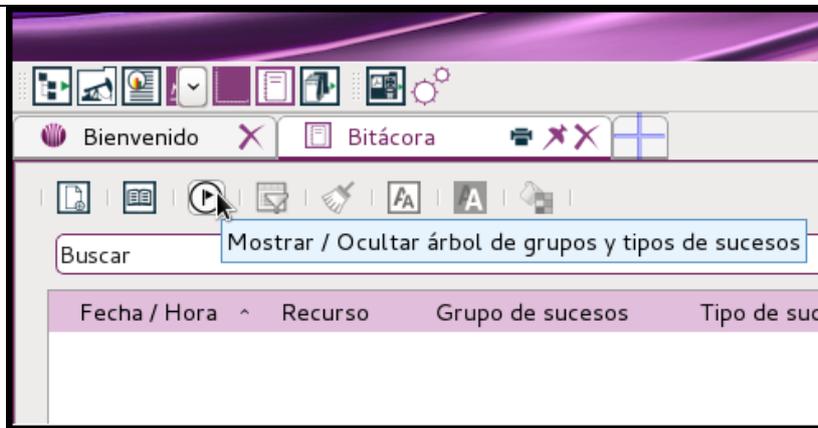


Tabla 9: Historia de Usuario Visualizar tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

Historia de Usuario	
Número: HU10	Nombre del requisito: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.
Programador: Luis Carlos Cruz Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3 semanas
Riesgo en Desarrollo:	Tiempo Real: 3 semanas

Problemas eléctricos.

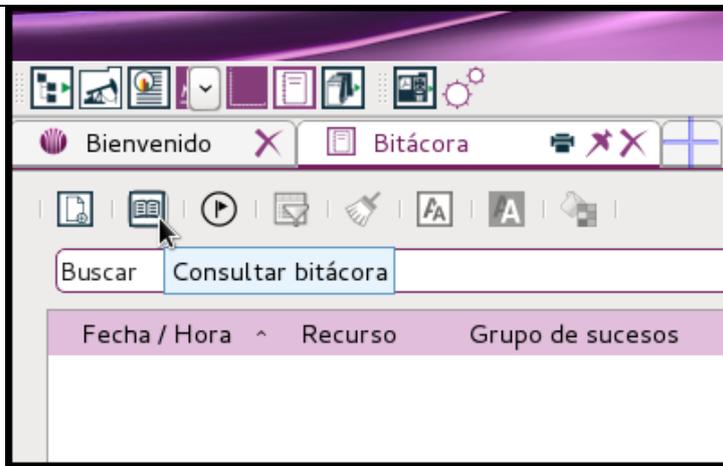
Referencias: HU1.

Problemas técnicos.

Descripción: El operador será capaz de consultar sucesos (bitácoras) registrados en el subsistema al seleccionar la opción Consultar bitácora en la parte superior izquierda de la interfaz principal del subsistema. Se debe seleccionar un rango de fechas en la ventana Configuración de la consulta, además, al presionar el botón Filtro se muestra la ventana Filtro de Consulta, donde se pueden configurar otros parámetros para el filtrado de la información (usuario, descripción, tipo de sucesos, grupo de sucesos, recurso). Al seleccionar los datos para el filtrado y presionar el botón Aceptar se presentarán los resultados en la interfaz principal en forma de sumario.

Observaciones:

Prototipo de interfaz:



Configuración de la consulta

Desde

Fecha Hora

Hasta

Fecha Hora

Filtro de Consulta

Criterio: Campo Alfanumérico

Operador

Tabla 10: Historia de Usuario Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.

2.6 Diseño de la arquitectura propuesta

2.6.1 Estilo arquitectónico

Sistemas basados en llamada y retorno: Este estilo arquitectónico permite al diseñador del software (arquitecto del sistema) construir una estructura de programa relativamente fácil de modificar y ajustar a escala. (21)

2.6.2 Arquitectura del subsistema

El framework Qt proporciona un conjunto de clases (vistas) que, basado en la arquitectura Modelo-Vista permite la fácil manipulación de los datos y la forma en que se presentan al usuario. Al hacer uso del framework Qt para el desarrollo de la aplicación se hereda la arquitectura Modelo-Vista, empleada por el mismo para el manejo de los datos. (22)

Para el desarrollo de la solución se decidió utilizar la arquitectura Modelo-Vista dada la posibilidad que brinda Qt de permitir la fácil manipulación de los datos. La arquitectura modelo-vista de Qt se puede clasificar como una arquitectura de N capas por la forma en que se separan los diferentes aspectos del desarrollo, tales como las cuestiones de vista, lógica de negocio y acceso a datos, lo que permite que las distintas partes de la aplicación se puedan modificar sin que se afecte las demás capas implementadas en el desarrollo de la aplicación.

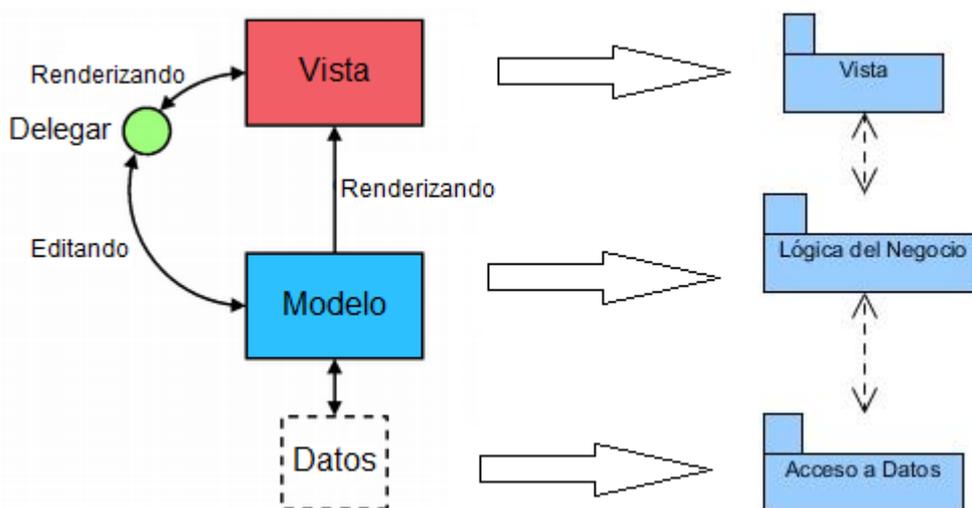


Figura 8: Correspondencia entre la Arquitectura Modelo-Vista y la Arquitectura de N capas. (22)

Vista: es la que presenta el sistema al usuario, le comunica la información y captura la información introducida por éste, tiene la característica de ser amigable, entendible y fácil de usar, en sí, es donde se encuentran las vistas de la aplicación y se encarga de interactuar con la capa de lógica de negocio.

Lógica del negocio: es responsable de recibir los eventos de entrada desde la vista y presentar los resultados, y se comunica con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Acceso a datos: es donde están los Objetos de Acceso a Datos, el Sistema de Gestión de Base de Datos que realiza todo el almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.7 Patrones de diseño

Un patrón de diseño es un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos. Existen tres clasificaciones para los mismos: de creación, estructurales y de comportamiento. (23) Entre los que se usaron en el desarrollo del subsistema de bitácoras de operaciones se encuentran dos patrones GRASP y tres GOF.

2.7.1 Patrones GRASP

GRASP, Acrónimo de *General Responsibility Assignment Software Patterns* (Patrones de Software para la Asignación General de Responsabilidad). Estos patrones describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. (24)

Patrón Experto: El propósito del patrón Experto es asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. La responsabilidad de la creación de un objeto o la implementación de un método debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). (24)

Dicho patrón se evidencia en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información que manejan. Específicamente en las clases de cada una de las entidades, cada una de ellas será la responsable de implementar la lógica del elemento que representa (Grupos de sucesos, Tipos de Sucesos).

Patrón Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de su uso es encontrar un creador que debe conectar con el objeto producido en cualquier evento. (24)

El patrón creador se percibe en las clases que representan una entidad que tiene la responsabilidad de crear otra, tal es el caso de la clase *LogGroup* que se encarga de crear instancias de la clase *OperatorLogType*, esto se logra debido a que *LogGroup* conoce los datos necesarios para crear los tipos de sucesos.

2.7.2 Patrones GOF

Los patrones *Gang-of-Four* ("pandilla de los cuatro") o comúnmente llamados GOF, descritos en el libro *Design Patterns: Elements of Reusable Object Oriented Software* (Gama 1995) definen un catálogo con 23 patrones básicos. (25)

Estos patrones se clasifican en 3 grandes categorías basadas en su propósito: (25)

- De Creación: abstraen el proceso de creación de instancias.

- Estructurales: se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
- De Comportamiento: atañen a los algoritmos y a la asignación de responsabilidades entre objetos.

Para la implementación del mecanismo en cuestión se utilizaron los patrones: *Singleton*, *Observer* y *Facade*, los cuales se describen a continuación.

Singleton: El patrón de diseño *Singleton* (instancia única) como patrón de creación, está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Este patrón se implementa creando en una clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado). (25)

El uso de este patrón se puede observar en la clase *Workbench*, esta clase contiene instancias de los módulos principales del sistema y solo se instancia una vez cuando inicia la aplicación, ya sea en ambiente de edición o de ejecución, y persiste hasta que se cierra la aplicación, logrando tener una instancia global de esta clase.

Observer: La mayoría de librerías para el desarrollo de aplicaciones con GUI como Qt, está diseñado para ser dirigidos por eventos, de ahí el famoso sistema de señales y ranuras (*signals and slots*) de Qt que no es más que un patrón de diseño *Observer* bastante fortalecido. Señales y ranuras es una construcción del lenguaje introducido en Qt, lo que hace que sea fácil de implementar el patrón *Observer*, evitando código repetitivo. (25)

La idea principal detrás del patrón de comportamiento *Observer* es que existe una entidad con estados cambiantes y una o más entidades observándola. Los observadores esperan a que la entidad observada les informe de un cambio de estado a través de un evento que puede ser de su interés, por lo que los observadores se registran con la entidad observada. Cuando ocurre un evento, la entidad observada mira en su lista de observadores y notifica a aquellos que se registraron para recibir eventos de ese tipo. Los observadores dejaron instrucciones detalladas de cómo puede la entidad observada ponerse en contacto con ellos para recibir los eventos. (25)

Este patrón se pone de manifiesto de forma general en toda la aplicación, con el uso de los *signals* y *slots* que propone la biblioteca Qt, logrando actualizar las vistas cuando cambia el estado en alguna entidad y de forma específica en la vista del árbol de proyectos en el editor, para actualizar los grupos y tipos de sucesos.

Facade: Es un patrón estructural a nivel de objetos, su propósito es proporcionar una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas. (25)

Su uso es recomendado para situaciones donde se pretende proporcionar una interfaz simple para un subsistema complejo, cuando existen muchas dependencias entre los clientes y las clases que implementan una abstracción. Una “fachada” proporciona al subsistema independencia y portabilidad. (25)

El uso de este patrón se pone de manifiesto en la clase del módulo de configuración *EditionConfigurationAgent* y la clase *EditionConfigurationClient* de las cuales una es fachada de la otra, o sea, *EditionConfigurationClient* es la encargada de recibir y devolver los datos de configuración y es usada por los otros módulos del SCADA para este propósito, mientras que *EditionConfigurationAgent* es la que gestiona el proceso como tal en el módulo configuración.

2.8 Diagrama de clases del diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos. El diagrama de clases del subsistema de bitácoras de operaciones se dividió en tres capas o componentes de para representar la arquitectura N capas (Vista, Lógica del Negocio y Acceso a Datos).

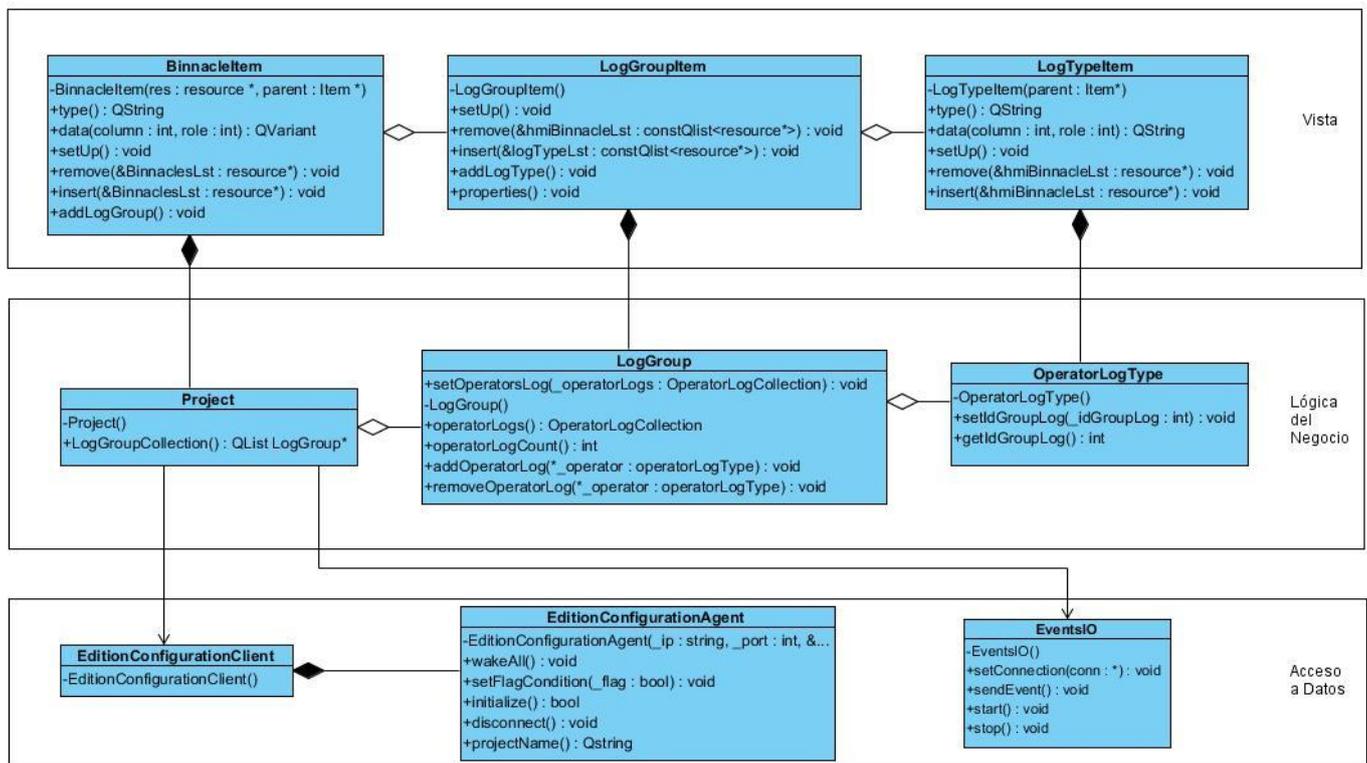


Figura 9: Diagrama de clases del diseño.

2.8.1 Descripción de las clases del diseño

Vista

BinnacleItem: Clase de la vista que garantiza que se muestre el comportamiento del ítem Bitácoras en el árbol de proyectos del editor del SCADA y se pueda seleccionar la opción Agregar grupos de sucesos en el mismo.

LogGroupItem: Clase de la vista que presenta al usuario los grupos de sucesos creados, dentro del ítem Bitácoras en el árbol de proyectos del editor del SCADA y se asegura de que se pueda seleccionar la opción Agregar tipo de sucesos en el mismo.

LogTypeItem: Clase de la vista que se asegura de que se muestren los tipos de sucesos dentro de los grupos de sucesos, que a su vez se encuentran dentro del ítem Bitácoras en el árbol de proyectos del editor del SCADA.

Modelo

Project: Clase del modelo que se encarga de almacenar la lista de grupos de sucesos del ítem Bitácoras en el Editor de SCADA y los atributos y reglas en la gestión de los mismos.

LogGroup: Clase del modelo que se encarga de almacenar los datos de la lógica del negocio de los grupos de sucesos y contiene la lista de los tipos de sucesos.

OperatorLogType: Clase del modelo que se encarga de almacenar los datos de la lógica del negocio de los tipos de sucesos, sus atributos y reglas.

Acceso a Datos

EditionConfigurationClient: Clase que recibe y devuelve los datos de configuración del SCADA al resto de los módulos, se encarga de solicitar y proporcionar los datos a la clase *EditionConfigurationAgent* para que esta gestione el proceso en el servidor de configuración.

EditionConfigurationAgent: Esta clase se encarga de gestionar directamente los datos de configuración que recibe de la clase *EditionConfigurationClient* y devuelve los datos a esta misma clase cuando los solicita.

EventsIO: Esta clase se encarga de enviar los datos de los sucesos al servidor del módulo base de datos históricos a través de la capa de comunicaciones haciendo uso de la función *SendEvent* y se encarga de solicitar y devolver los datos requeridos por el operador haciendo uso de filtros a través de la biblioteca libqt4-sql-psql.

2.9 Conclusiones parciales del capítulo

- En este capítulo se utilizaron varias herramientas para facilitar el proceso de diseño de la aplicación y la definición de sus funcionalidades y características. Entre los artefactos generados por las mismas se encuentran, el modelo conceptual para entender mejor el contexto de la solución, las historias de usuario que permiten una mayor comprensión de los requisitos funcionales del sistema y el diagrama de clases para describir a un nivel más detallado el desarrollo del subsistema.
- Se hizo uso de los patrones de diseño para dar solución a problemas típicos y recurrentes que se encuentran a la hora de desarrollar una aplicación.

Capítulo 3: Implementación y pruebas del sistema.

3.1 Introducción

Este capítulo está dedicado a la realización de pruebas al subsistema de bitácoras de operaciones para el SCADA SAINUX. Las pruebas son una de las fases más importantes en el desarrollo de software ya que en ellas se detectan las anomalías del producto y se corrigen con el objetivo de garantizar un producto con calidad. En este capítulo se generan varios artefactos, entre los que se encuentran: el diagrama de componentes para modelar la vista estática del subsistema y las dependencias lógicas entre los componentes de software y el diagrama de despliegue para modelar la disposición física del subsistema de bitácoras de operaciones mediante su relación con el resto del sistema SCADA SAINUX.

3.2 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre los mismos. Se utiliza para modelar la vista estática del sistema y muestra la organización y las dependencias lógicas entre los componentes de software, sean estos de código fuente, binarios o ejecutables. (26)

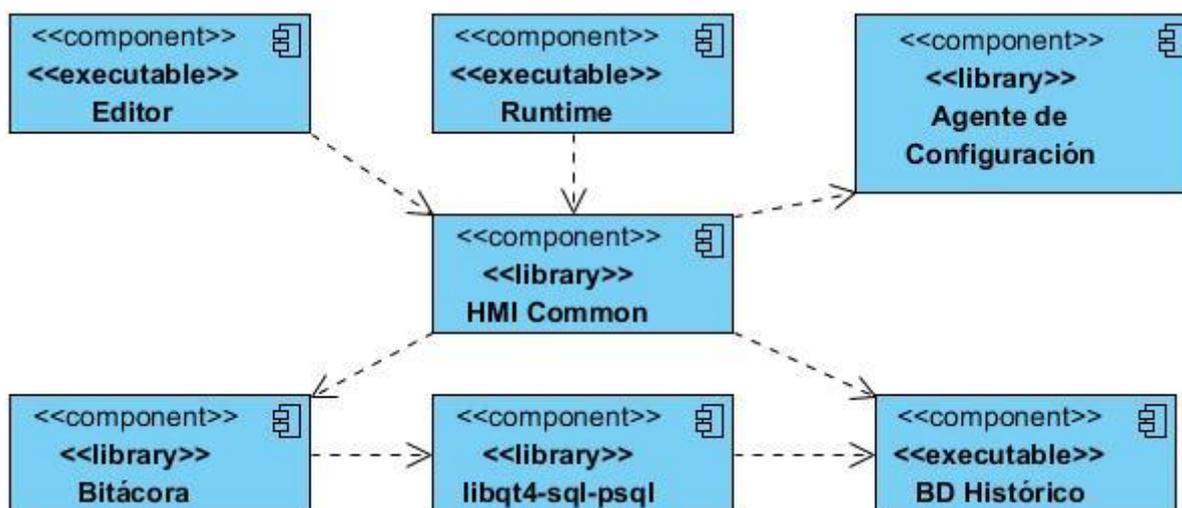


Figura 10: Diagrama de Componentes.

Agente de Configuración: es el que permite a la bitácora obtener los grupos y tipos de sucesos previamente configurados en el HMI-Editor.

BD Histórico: encapsula al servidor de datos históricos y su base de datos asociada los que permiten la persistencia de las bitácoras creadas.

Bitácora: se refiere al sumario que forma parte del HMI-Ejecución sobre el cual se crean y consultan las bitácoras de operaciones.

HMI Common: encapsula todas las funcionalidades comunes del módulo HMI de manera general.

Editor: se refiere al entorno de edición del módulo HMI.

Runtime: se refiere al entorno de visualización del módulo HMI.

Libqt4-sql-psql: biblioteca que permite acceder a información almacenada en gestores de bases de datos postgresQL.

3.3 Diagrama de Despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar la disposición física de los artefactos software en nodos (usualmente plataforma de hardware). La mayoría de las veces el modelado de la vista de despliegue implica modelar la topología del hardware sobre el que se ejecuta el sistema. (27)

A continuación se muestra la relación entre el subsistema de bitácoras de operaciones dentro del módulo HMI y el sistema SCADA SAINUX comunicándose a través del módulo comunicaciones mediante el protocolo TCP.

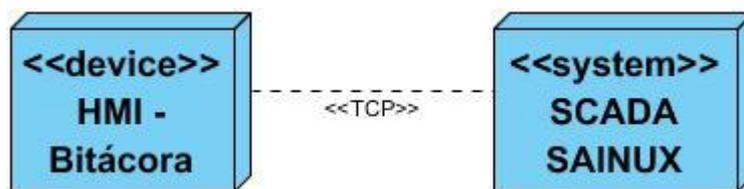


Figura 11: Diagrama de Despliegue.

3.4 Estándar de Codificación

Los estándares de codificación, también conocidos como estilos de programación o convenciones de código, son convenios para escribir código fuente en ciertos lenguajes de programación. Permiten que el código en consecuencia sea mantenible y que todos los participantes lo puedan entender en un menor tiempo. (28)

Para la implementación del subsistema de bitácoras de operaciones es necesario utilizar el Estándar de codificación de C++ definido para su uso en el CEDIN. (29)

Algunas de las pautas que define el estándar utilizado son:

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

A continuación se presentan dos ejemplos del estándar de codificación usado:

```
/**@brief Funcionalidad para crear la barra de botones de la bitácora.
 *
 * @author Luis Carlos Cruz lccruz@estudiantes.uci.cu
 * @date 09/02/2015
 */
void createToolButton();
```

Figura 12: Estándar de codificación, ejemplo 1.

```
/**@brief Devuelve true si la cadena pasada por parámetros es vacía o contiene solamente espacio
 *
 * @return true si la cadena es vacía o contiene solo espacios
 * @param _string QString cadena de texto
 * @author Luis Carlos Cruz lccruz@estudiantes.uci.cu
 * @date 20/02/2015
 */
bool stringEmpty(QString _string);
```

Figura 13: Estándar de codificación, ejemplo 2.

3.5 Pruebas de software

La fase de pruebas de los sistemas, tiene como objetivo la verificación de diversos procesos para comprobar si éstos cumplieron con los requisitos. Las pruebas de software, según Pressman, son el último bastión para la evaluación de la calidad y donde de manera más pragmática se realiza la detección de errores. Las pruebas se clasifican en dos vertientes fundamentales; pruebas de caja blanca y pruebas de caja negra. (30)

Las **pruebas de caja blanca** se realizan en un contexto donde se conoce el funcionamiento interno del producto. Donde se realizan pruebas para asegurarse de que todas las piezas encajan, en otras palabras, que las operaciones internas se realizan de acuerdo con las especificaciones. Estas pruebas son enfocadas a un ambiente procedural, que prueban todas las rutas lógicas del programa y la colaboración entre componentes. (30)

Las **pruebas de caja negra** se desarrollan en un contexto donde se conocen las funciones específicas para las cuales se diseñó el producto. Se realizan pruebas para demostrar que cada función es plenamente operacional. Se aplican generalmente a la interfaz del software y se concentran en los requisitos funcionales del software. (30)

Una vez terminada la fase de implementación del subsistema de bitácoras de operaciones se procede a ejecutar las pruebas a cada una de las historias de usuario definidas en el Capítulo 2. Las pruebas que se le realizarán al subsistema serán de caja negra lo que permitirá encontrar discrepancias entre el software y la especificación funcional. Estas pruebas permitirán validar los procesos y reglas de negocio y requisitos funcionales establecidos.

Para garantizar la calidad del subsistema y el correcto cumplimiento de los requisitos funcionales, se realizarán iteraciones de pruebas hasta que la cantidad de errores críticos sea cero. Se consideran críticos

aquellos errores que atentan contra el correcto funcionamiento del subsistema, dado que se encuentran en las funcionalidades más importantes.

3.6 Planes de prueba y diseños de casos de prueba

Se definieron planes de pruebas y se diseñaron casos de prueba para cada una de las historias de usuario para asegurar que cada funcionalidad cumpliera con los requisitos definidos en la fase de análisis.

3.6.1 Plan de prueba para la Historia de Usuario 1: Adicionar suceso

Descripción General:

Se encarga de la adición de sucesos al subsistema de bitácoras de operaciones.

Condiciones de Ejecución:

El operador debe seleccionar la opción Adicionar suceso en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU1: Adicionar suceso	Escenario 1.1: Suceso agregado satisfactoriamente.	Se insertan los siguientes datos: 1. Seleccionar: grupo de sucesos, tipo de sucesos y recurso (en caso de que exista un recurso asociado al suceso). 2. Ingresar descripción del suceso.	1. Seleccionar la opción Adicionar suceso en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones. 2. Aparece la ventana Adicionar suceso donde se deben ingresar los datos del suceso correctamente. 3. Accionar el botón Aceptar.
	Escenario 1.2: Datos incorrectos.	En caso de que no se seleccione alguno de los datos (grupo de sucesos, tipo de sucesos) o no se agregue una descripción.	1. Seleccionar la opción Adicionar suceso en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones. 2. Aparece la ventana Adicionar suceso donde se deben ingresar los datos del suceso. 3. Al accionar el botón Aceptar: 3.1 Si se omite la selección del grupo de sucesos aparecerá el mensaje “Debe escoger un grupo de sucesos” y se rellenará el campo de color rojo para

			<p>facilitar la identificación del error por parte del operador.</p> <p>3.2 Si se omite la selección del tipo de sucesos aparecerá el mensaje “Debe escoger un tipo de sucesos” y se rellenará el campo de color rojo para facilitar la identificación del error por parte del operador.</p> <p>3.3 En caso de que falte la descripción se mostrará el mensaje “Descripción vacía” y se rellenará el campo de color rojo para facilitar la identificación del error por parte del operador.</p>
--	--	--	---

Tabla 11: Plan de prueba para la Historia de Usuario 1: Adicionar suceso.

3.6.2 Diseño de caso de prueba 1: Adicionar Suceso

C: Datos correctos, V: Campo vacío.

Id de la sección	Escenario	Variable 1 Grupo de sucesos	Variable 2 Tipo de sucesos	Variable 4 Recurso	Variable 5 Descripción	Respuesta del sistema	Resultado de la prueba
P.HU1	Crear Suceso	C	C	C	C	Se adiciona el suceso	Prueba satisfactoria
		V	C	C	C	Notificación de Error	
		C	V	C	C	Notificación de Error	
		C	C	C	V	Notificación de Error	

Tabla 12: Diseño de caso de prueba 1: Adicionar Suceso.

3.6.3 Plan de prueba para la Historia de Usuario 2: Agregar grupo de sucesos

Descripción General:

Se encarga de la creación de grupos de sucesos dentro del ítem Bitácoras en el módulo HMI en el editor del SCADA SAINUX.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el ítem Bitácoras dentro del módulo HMI en el editor del SCADA y seleccionar la opción Agregar Grupo de sucesos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
----------------------	--------------------------	---------------------------------	---------------

P.HU2: Agregar Grupo de sucesos.	Escenario 2.1: Grupo de sucesos agregado satisfactoriamente.	Se insertan los siguientes datos: Nombre y Descripción (opcional) del grupo de sucesos.	<ol style="list-style-type: none"> 1. Presionar el botón secundario del ratón sobre el ítem Bitácoras y seleccionar la opción Agregar Grupo de sucesos. 2. Aparece la ventana Agregar Grupo de sucesos donde se deben ingresar los datos del grupo a agregar. 3. Accionar el botón Aceptar.
	Escenario 2.2: Datos incorrectos.	En caso de que no se ingrese un nombre o ya exista un grupo de sucesos registrado con el mismo nombre del grupo que se desea crear.	<ol style="list-style-type: none"> 1. Presionar el botón secundario del ratón sobre el ítem Bitácoras y seleccionar la opción Agregar Grupo de sucesos. 2. Aparece la ventana Agregar Grupo de sucesos donde se deben ingresar los datos del grupo a agregar. 3. Al accionar el botón Aceptar: <ol style="list-style-type: none"> 3.1 Si no se ha ingresado un nombre se muestra el mensaje “Nombre vacío” y se rellena de color rojo el campo para el nombre del grupo para facilitar la identificación del error por parte del mantenedor. 3.2 Si se ha ingresado un nombre que ya existe registrado se muestra el mensaje “Nombre Repetido” y se rellena de color rojo el campo para el nombre del grupo para facilitar la identificación del error por parte del mantenedor.

Tabla 13: Plan de prueba para la Historia de Usuario 2: Agregar grupo de sucesos.

3.6.4 Diseño de caso de prueba 2: Agregar grupo de sucesos

C: Datos correctos, V: Campo vacío, R: Nombre repetido.

Id de la sección	Escenario	Variable 1 Nombre	Variable 2 Descripción	Respuesta del sistema	Resultado de la prueba
P.HU2	Agregar Grupo de sucesos	C	C	Se crea el grupo de sucesos	Prueba satisfactoria
		V	C	Notificación de Error	

		R	C	Notificación de Error	
--	--	---	---	-----------------------	--

Tabla 14: Diseño de caso de prueba 2: Agregar grupo de sucesos.

3.6.5 Plan de prueba para la Historia de Usuario 3: Modificar grupo de sucesos

Descripción General:

Se encarga de la modificación de los datos de un grupo de sucesos.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el grupo que desea modificar y seleccionar la opción Propiedades.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU3: Modificar grupo de sucesos.	Escenario 3.1: Grupo de sucesos modificado satisfactoriamente.	Se modifican los siguientes datos: Nombre o Descripción de un grupo de sucesos creado previamente.	1. Presionar el botón secundario del ratón sobre el grupo de sucesos que se desea modificar y seleccionar la opción Propiedades. 2. Aparece la ventana Propiedades del grupo de sucesos donde se pueden modificar los datos del grupo a modificar. 3. Accionar el botón Aceptar.
	Escenario 3.2: Datos incorrectos.	En caso de que se deje vacío el campo del nombre o ya exista registrado un grupo con el mismo nombre.	1. Presionar el botón secundario del ratón sobre el grupo de sucesos que se desea modificar y seleccionar la opción Propiedades. 2. Aparecerá la ventana Propiedades del grupo de sucesos donde se pueden modificar los datos del grupo a modificar. 3. Al accionar el botón Aceptar: 3.1 Si no se ha ingresado un nombre se muestra el mensaje "Nombre vacío" y se rellena de color rojo el campo para el nombre del grupo para facilitar la identificación del error por parte del mantenedor. 3.2 Si se ha ingresado un nombre que ya existe registrado se muestra el mensaje "Nombre Repetido" y se rellena de color rojo el campo para el nombre del grupo para

			facilitar la identificación del error por parte del mantenedor.
--	--	--	---

Tabla 15: Plan de prueba para la Historia de Usuario 3: Modificar grupo de sucesos.

3.6.6 Diseño de caso de prueba 3: Modificar grupo de sucesos

C: Datos correctos, V: Campo vacío, R: Nombre Repetido.

Id de la sección	Escenario	Variable 1 Nombre	Variable 2 Descripción	Respuesta del sistema	Resultado de la prueba
P.HU3	Modificar Grupo de Sucesos	C	C	Se modifica el grupo de sucesos	Prueba satisfactoria
		V	C	Notificación de Error	
		R	C	Notificación de Error	

Tabla 16: Diseño de caso de prueba 3: Modificar grupo de sucesos.

3.6.7 Plan de prueba para la Historia de Usuario 4: Eliminar grupo de sucesos

Descripción General:

Se encarga de la eliminación de un grupo de sucesos.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el grupo que desea eliminar y seleccionar la opción Eliminar.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU4: Eliminar grupo de sucesos.	Escenario 4.1: Grupo de sucesos eliminado satisfactoriamente.	Se elimina un grupo de sucesos creado previamente.	<ol style="list-style-type: none"> 1. Presionar el botón secundario del ratón sobre el grupo de sucesos que se desea eliminar y seleccionar la opción Eliminar. 2. Aparecerá una ventana para confirmar la eliminación. En caso de que el grupo que se desea eliminar contenga al menos un tipo de sucesos asociado en el mensaje que se muestra para confirmar la eliminación se le advertirá al usuario que si se elimina el grupo se eliminarán los tipos de sucesos que este contenga. 3. Accionar el botón Aceptar.

	Escenario 4.2: Cancelación de Eliminación.	Se cancela la eliminación de un grupo de sucesos.	<ol style="list-style-type: none"> 1. Presionar el botón secundario del ratón sobre el grupo de sucesos que se desea eliminar y seleccionar la opción Eliminar. 2. Aparecerá una ventana para confirmar la eliminación. En caso de que el grupo que se desea eliminar contenga al menos un tipo de sucesos asociado en el mensaje que se muestra para confirmar la eliminación se le advertirá al usuario que si se elimina el grupo se eliminarán los tipos de sucesos que este contenga. 3. Accionar el botón Cancelar.
--	--	---	--

Tabla 17: Plan de prueba para la Historia de Usuario 4: Eliminar grupo de sucesos.

3.6.8 Diseño de caso de prueba 4: Eliminar grupo de sucesos

A: Botón accionado.

Id de la sección	Escenario	Variable 1 Botón Aceptar	Variable 2 Botón Cancelar	Respuesta del sistema	Resultado de la prueba
P.HU4	Eliminar grupo de sucesos	A	-	Se elimina el grupo de sucesos.	Prueba satisfactoria
		-	A	Se cancela la eliminación del grupo de sucesos.	

Tabla 18: Diseño de caso de prueba 4: Eliminar grupo de sucesos.

3.6.9 Plan de prueba para la Historia de Usuario 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones

Descripción General:

Se encarga de mostrar y ocultar los grupos de sucesos en el árbol de grupos y tipos de sucesos en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones.

Condiciones de Ejecución:

Para que los operadores sean capaces de visualizar en la interfaz principal del subsistema de bitácoras de operaciones los grupos de sucesos, estos deben ser agregados previamente por los mantenedores en el editor del SCADA SAINUX.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	Escenario 5.1: Grupos de sucesos mostrados satisfactoriamente.	Se muestra en la interfaz principal del subsistema de bitácoras de operaciones, en el entorno de visualización, los grupos de sucesos creados previamente en el editor del SCADA SAINUX.	1. Los grupos de sucesos son creados por los mantenedores en el editor del SCADA SAINUX y se presentan a los operadores en forma de árbol en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones en el entorno de visualización o Runtime del SCADA.
	Escenario 5.2: Grupos de sucesos ocultos satisfactoriamente.	Se oculta el árbol de grupos y tipos de sucesos.	1. Mediante el uso del botón "Mostrar / Ocultar árbol de grupos y tipos de sucesos" en la parte superior izquierda de la interfaz principal se tiene la posibilidad de ocultar el árbol de grupos y tipos de sucesos para un mejor aprovechamiento de la interfaz principal para la consulta de bitácoras.

Tabla 19: Plan de prueba para la Historia de Usuario 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

3.6.10 Diseño de caso de prueba 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones

C: Datos correctos.

Id de la sección	Escenario	Variable 1 Nombre del Grupo de Sucesos	Respuesta del sistema	Resultado de la prueba
P.HU5	Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	C	Se muestra el árbol de grupos y tipos de sucesos.	Prueba satisfactoria
		C	Se oculta el árbol de grupos y tipos de sucesos.	

Tabla 20: Diseño de caso de prueba 5: Visualizar grupos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

3.6.11 Plan de prueba para la Historia de Usuario 6: Agregar tipo de sucesos

Descripción General:

Se encarga de la creación de un tipo de sucesos dentro de un grupo de sucesos que a su vez se encuentra dentro del ítem Bitácoras en el módulo HMI en el editor del SCADA SAINUX.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el grupo de sucesos dentro del cual desea agregar un tipo de sucesos y seleccionar la opción Agregar tipo de sucesos. En caso de que no exista al menos un grupo de sucesos creado no se puede crear un tipo de sucesos.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU6: Agregar tipo de sucesos.	Escenario 6.1: Tipo de sucesos agregado satisfactoriamente.	Se insertan los siguientes datos: Nombre y Descripción del tipo de sucesos que se desea crear.	1. Presionar el botón secundario del ratón sobre el grupo de sucesos dentro del cual desea agregar un tipo de sucesos y seleccionar la opción Agregar tipo de sucesos. 2. Aparecerá la ventana Agregar tipo de sucesos donde se deben ingresar los datos del tipo de sucesos a agregar. 3. Accionar el botón Aceptar.
	Escenario 6.2: Datos incorrectos.	En caso de que no se ingrese un nombre o ya exista un tipo de sucesos registrado con el mismo nombre en ese grupo de sucesos.	1. Presionar el botón secundario del ratón sobre el grupo de sucesos dentro del cual desea agregar un tipo de sucesos y seleccionar la opción Agregar tipo de sucesos. 2. Aparece la ventana Agregar tipo de sucesos donde se deben ingresar los datos del tipo de sucesos a agregar. 3. Al accionar el botón Aceptar: 3.1 Si no se ha ingresado un nombre se muestra el mensaje "Nombre vacío" y se rellena de color rojo el campo para el nombre para facilitar la identificación del error por parte del mantenedor. 3.2 Si se ha ingresado un nombre que ya existe registrado en ese grupo de sucesos se muestra el mensaje "Nombre Repetido" y se rellena de color rojo el campo para el nombre

			para facilitar la identificación del error por parte del mantenedor.
--	--	--	--

Tabla 21: Plan de prueba para la Historia de Usuario 6: Agregar tipo de sucesos.

3.6.12 Diseño de caso de prueba 6: Agregar tipo de sucesos

C: Datos correctos, V: Campo vacío, R: Nombre Repetido.

Id de la sección	Escenario	Variable 1 Nombre	Variable 2 Descripción	Respuesta del sistema	Resultado de la prueba
P.HU6	Agregar tipo de sucesos	C	C	Se crea el tipo de sucesos	Prueba satisfactoria
		V	C	Notificación de Error	
		R	C	Notificación de Error	

Tabla 22: Diseño de caso de prueba 6: Agregar tipo de sucesos.

3.6.13 Plan de prueba para la Historia de Usuario 7: Modificar tipo de sucesos

Descripción General:

Se encarga de la modificación de los datos de un tipo de sucesos.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el tipo de sucesos que desea modificar y seleccionar la opción Propiedades.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU7: Modificar tipo de sucesos.	Escenario 7.1: Tipo de sucesos modificado satisfactoriamente.	Se modifican los siguientes datos: Nombre o Descripción de un tipo de sucesos creado previamente.	1. Presionar el botón secundario del ratón sobre el tipo de sucesos que se desea modificar y seleccionar la opción Propiedades. 2. Aparece la ventana Propiedades del tipo de sucesos donde se pueden modificar los datos del tipo de sucesos. 3. Accionar el botón Aceptar.
	Escenario 7.2: Datos incorrectos.	En caso de que se deje vacío el campo del nombre o ya exista registrado un tipo de sucesos con el mismo	1. Presionar el botón secundario del ratón sobre el tipo de sucesos que se desea modificar y seleccionar la opción Propiedades.

		nombre en ese grupo de sucesos.	<p>2. Aparece la ventana Propiedades del tipo de sucesos donde se pueden modificar los datos del tipo de sucesos.</p> <p>3. Al accionar el botón Aceptar:</p> <p>3.1 Si no se ha ingresado un nombre se muestra el mensaje “Nombre vacío” y se rellena de color rojo el campo para el nombre para facilitar la identificación del error por parte del mantenedor.</p> <p>3.2 Si se ha ingresado un nombre que ya existe registrado en ese grupo de sucesos se muestra el mensaje “Nombre Repetido” y se rellena de color rojo el campo para el nombre para facilitar la identificación del error por parte del mantenedor.</p>
--	--	---------------------------------	--

Tabla 23: Plan de prueba para la Historia de Usuario 7: Modificar tipo de sucesos.

3.6.14 Diseño de caso de prueba 7: Modificar tipo de sucesos

C: Datos correctos, V: Campo vacío, R: Nombre Repetido.

Id de la sección	Escenario	Variable 1 Nombre	Variable 2 Descripción	Respuesta del sistema	Resultado de la prueba
P.HU7	Modificar tipo de sucesos	C	C	Se modifica el tipo de sucesos	Prueba satisfactoria
		V	C	Notificación de Error	
		R	C	Notificación de Error	

Tabla 24: Diseño de caso de prueba 7: Modificar tipo de sucesos.

3.6.15 Plan de prueba para la Historia de Usuario 8: Eliminar tipo de sucesos

Descripción General:

Se encarga de la eliminación de un tipo de sucesos.

Condiciones de Ejecución:

El mantenedor debe presionar el botón secundario del ratón sobre el tipo de sucesos que desea eliminar y seleccionar la opción Eliminar.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU8: Eliminar tipo de sucesos.	Escenario 8.1: Tipo de sucesos eliminado satisfactoriamente.	Se elimina un tipo de sucesos creado previamente.	1. Presionar el botón secundario del ratón sobre el tipo de sucesos que se desea eliminar y seleccionar la opción Eliminar. 2. Aparece una ventana para confirmar la eliminación. 3. Accionar el botón Aceptar.
	Escenario 8.2: Cancelación de eliminación.	Se cancela la eliminación de un tipo de sucesos.	1. Presionar el botón secundario del ratón sobre el tipo de sucesos que se desea eliminar y seleccionar la opción Eliminar. 2. Aparece una ventana para confirmar la eliminación. 3. Accionar el botón Cancelar.

Tabla 25: Plan de prueba para la Historia de Usuario 8: Eliminar tipo de sucesos.

3.6.16 Diseño de caso de prueba 8: Eliminar tipo de sucesos

A: Botón accionado.

Id de la sección	Escenario	Variable 1 Botón Aceptar	Variable 2 Botón Cancelar	Respuesta del sistema	Resultado de la prueba
P.HU8	Eliminar tipo de sucesos	A	-	Se elimina el tipo de sucesos.	Prueba satisfactoria
		-	A	Se cancela la eliminación del tipo de sucesos.	

Tabla 26: Diseño de caso de prueba 8: Eliminar tipo de sucesos.

3.6.17 Plan de prueba para la Historia de Usuario 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones

Descripción General:

Se encarga de mostrar y ocultar los tipos de sucesos en el árbol de grupos y tipos de sucesos en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones.

Condiciones de Ejecución:

Para que los operadores sean capaces de visualizar en la interfaz principal del subsistema de bitácoras de operaciones los grupos y tipos de sucesos, estos deben ser agregados previamente por los Mantenedores en el editor del SCADA SAINUX.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU9: Visualizar tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	Escenario 9.1: Tipos de sucesos mostrados satisfactoriamente.	Se muestra en la interfaz principal del subsistema de bitácoras de operaciones los tipos de sucesos creados previamente en el editor del SCADA SAINUX.	1. Los tipos de sucesos son creados por los mantenedores en el editor del SCADA SAINUX y se presentan a los operadores en la parte derecha de la interfaz principal del subsistema de bitácoras de operaciones en el entorno de visualización o Runtime del SCADA en forma de árbol; cada tipo de sucesos dentro del grupo de sucesos al que pertenece.
	Escenario 9.2: Tipos de sucesos ocultados satisfactoriamente.	Se oculta el árbol de grupos y tipos de sucesos.	1. Mediante el uso del botón “Mostrar / Ocultar árbol de grupos y tipos de sucesos” en la parte superior izquierda de la interfaz principal se tiene la posibilidad de ocultar el árbol de grupos y tipos de sucesos para un mejor aprovechamiento de la interfaz principal para la consulta de bitácoras.

Tabla 27: Plan de prueba para la Historia de Usuario 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

3.6.18 Diseño de caso de prueba 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones

C: Datos correctos.

Id de la sección	Escenario	Variable 1 Nombre del tipo de sucesos	Respuesta del sistema	Resultado de la prueba
P.HU9	Visualizar tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.	C	Se muestra el árbol de grupos y tipos de sucesos.	Prueba satisfactoria
		C	Se oculta el árbol de grupos y tipos de sucesos.	

Tabla 28: Diseño de caso de prueba 9: Visualizar los tipos de sucesos en la interfaz principal del subsistema de bitácoras de operaciones.

3.6.19 Plan de prueba para la Historia de Usuario 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado

Descripción General:

Se encarga de mostrar en la interfaz principal en forma de sumario, los datos de los sucesos (bitácoras) que fueron creados en el rango de fechas seleccionado por el operador. El operador puede usar filtros para clasificar la información.

Condiciones de Ejecución:

El operador debe seleccionar la funcionalidad Consultar bitácora en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU10: Consultar registros históricos de bitácoras de operaciones.	Escenario 10.1: Sucesos (bitácoras) consultados satisfactoriamente.	Se muestra en la interfaz principal del subsistema de bitácoras de operaciones los sucesos filtrados.	<ol style="list-style-type: none"> 1. Seleccionar la funcionalidad Consultar bitácora en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones. 2. Aparece la ventana Configuración de la consulta para seleccionar el rango de fechas y el botón Filtro que al ser presionado muestra las opciones de filtrado para ser editadas en la ventana Filtro de consulta. 3. Accionar el botón Aceptar.
	Escenario 10.2: No se encontraron sucesos que cumplan con los filtros definidos.	En el caso de que no se encuentren sucesos que cumplan con los filtros definidos.	<ol style="list-style-type: none"> 1. Seleccionar la funcionalidad Consultar bitácora en la parte superior izquierda de la interfaz principal del subsistema de bitácoras de operaciones. 2. Aparece la ventana Configuración de la consulta para seleccionar el rango de fechas y el botón Filtro que al ser presionado muestra la ventana Filtro de consulta donde se encuentran las opciones de filtrado. 3. Al accionar el botón Aceptar: Si no se encuentran sucesos que cumplan con los filtros definidos no se

			mostrará ningún resultado en la interfaz principal.
--	--	--	---

Tabla 29: Plan de prueba para la Historia de Usuario 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.

3.6.20 Diseño de caso de prueba 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado

C: Datos correctos, I: Datos Incorrectos.

Id de la sección	Escenario	Variable 1 Fecha Inicial	Variable 2 Fecha Final	Variable 3 Filtros	Respuesta del sistema	Resultado de la prueba
P.HU10	Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.	C	C	C	Se muestra el resultado de la consulta en la interfaz principal.	Prueba satisfactoria
		I	C	C	Notificación de Error	
		C	I	C	Notificación de Error	
		C	C	I	Notificación de Error	

Tabla 30: Diseño de caso de prueba 10: Consulta de registros históricos de bitácoras de operaciones en un rango de fechas determinado.

3.7 Desarrollo de las iteraciones de pruebas

Los planes y diseños de prueba definidos se ejecutaron sobre el subsistema de bitácoras en busca de no conformidades sobre todas las funcionalidades del sistema. En cada iteración se probaron todas las funcionalidades, desarrollándose 3 de ellas, cuyos resultados se presentan a continuación:

Iteración #1

Durante las pruebas en esta iteración se encontraron 10 no conformidades, las cuales fueron resueltas inmediatamente.

Iteración #2

Durante las pruebas en esta iteración se encontraron 6 no conformidades, las cuales fueron resueltas inmediatamente.

Iteración #3

Durante las pruebas en esta iteración no se encontraron no conformidades.

3.8 Resumen de no conformidades

Caso de Prueba	Cantidad de no conformidades en la Iteración 1	Cantidad de no conformidades en la Iteración 2	Cantidad de no conformidades en la Iteración 3
CP1	1	1	0
CP2	3	1	0
CP3	0	0	0
CP4	1	1	0
CP5	1	1	0
CP6	3	0	0
CP7	0	0	0
CP8	0	0	0
CP9	1	1	0
CP10	0	1	0
Total	10	6	0

Tabla 31: Resumen de no conformidades.

3.9 Clasificación de las no conformidades

Se encontraron un total de 16 no conformidades: 3 errores ortográficos, 5 mensajes incorrectos o incompletos, 6 errores de visualización y 2 de validación.



Figura 14: Clasificación de las no conformidades.

3.10 Conclusiones parciales del capítulo

- En el desarrollo del capítulo se presentó la distribución física y lógica del sistema y sus componentes mediante los diagramas de componentes y despliegue.

- Finalmente, se valida el subsistema desarrollado mediante pruebas de caja negra, realizando casos de prueba a todas las funcionalidades. Este proceso permitió detectar las no conformidades que presentaba el subsistema y dar solución a las mismas.

Conclusiones generales

El presente trabajo tuvo como base la investigación y el desarrollo científico encaminados a la elaboración e integración de un subsistema de bitácoras de operaciones al módulo HMI del SCADA SAINUX, concluyéndose que:

- Como resultado del presente trabajo se da cumplimiento al objetivo general, lográndose el desarrollo de un subsistema que permite registrar y consultar bitácoras de operaciones en el SCADA SAINUX.
- El subsistema contribuye a garantizar la integridad, estructura y disponibilidad de la información registrada al ser almacenada en la base de datos históricos del SCADA SAINUX.
- Este se encuentra integrado a la rama en desarrollo de la primera versión del SCADA SAINUX, posibilitando a los usuarios del software la consulta de información relacionada con las bitácoras de operaciones registradas por los operadores, de modo que pueden ser analizadas y utilizadas en la toma de decisiones.

Recomendaciones

Con el objetivo de incrementar las funcionalidades que presenta el subsistema, para mejorar los servicios que el mismo pueda prestar, se plantean las siguientes recomendaciones:

- Incorporar funcionalidades que permitan el envío automático de bitácoras mediante correo electrónico.
- Utilizar la interfaz para la consulta de bitácoras que ofrece la biblioteca libHDA para acceder a la información almacenada en la base de datos históricos del SCADA SAINUX.

Referencias bibliográficas y bibliografía

1. CEDIN Informática Industrial. *gespro. Suite de Gestión de Proyectos*. [En línea] [Citado el: 20 de 4 de 2015.] <http://gespro.cedin.prod.uci.cu/>.
2. Crespo, William. WordPress.com. [En línea] 9 de Febrero de 2011. <https://automatizacionindustrial.wordpress.com/2011/02/09/queeslaautomatizacionindustrial/>.
3. ecured. [En línea] 2008. http://www.ecured.cu/index.php/Sistema_SCADA.
4. Hernández, Adrián Carmona. *Manejador para la comunicación con dispositivos mediante protocolo SNMP*. La Habana : s.n., 2014.
5. Romero, Ing. Diego M. *Sistemas de Interfaz Humano Máquina (HMI)*. 2007.
6. alpemosa. Buenas Tareas. [En línea] 2 de 3 de 2010. <http://www.buenastareas.com/ensayos/Bitacora/146397.html>.
7. Bitacoraweb. [En línea] 2006. <http://www.bitacoraweb.net/info/bitacora.html>.
8. Hernandez, Mtro. Gustavo Reyes. Taller de Base de Datos. *Crear una bitácora en MySQL*. [En línea] 9 de 6 de 2010. <http://tavoberry.com/blog/crear-una-bitacora-en-mysql/>.
9. Revolution en la nube. [En línea] 14 de 8 de 2014. <http://aprendo.tressrevolution.com/article/AA-00366/0/Pregunta-Cu%C3%A1-es-la-diferencia-entre-la-bit%C3%A1cora-del-empleado-del-sistema-y-de->.
10. bacdavi. Foros del Web. *Bitácora de usuarios*. [En línea] 16 de 12 de 2008. <http://www.forosdelweb.com/f86/bitacora-usuarios-654694/>.
11. Rodríguez, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2014.
12. DocIRs. [En línea] 2010. <http://www.docirs.com/uml.htm>.
13. ecured. [En línea] 2010. http://www.ecured.cu/index.php/Visual_Paradigm.
14. Torres, Jesús. Sistema de videovigilancia from scratch. [En línea] 29 de enero de 2013. <http://ull-etsii-sistemas-operativos.github.io/videovigilancia-blog/proyecto-qt-framework-de-desarrollo-de-aplicaciones.html>.
15. ecured. [En línea] 2009. <http://www.ecured.cu/index.php/PostgreSQL>.
16. Space, David Blanchad. blanchardspace. [En línea] 6 de Mayo de 2013. <https://blanchardspace.wordpress.com/2013/05/06/introduccion-a-c-que-es/>.
17. Buenas Tareas. [En línea] 5 de Octubre de 2009. <http://www.buenastareas.com/ensayos/Modelos-Conceptuales/23455.html>.
18. Metodología Gestión de Requerimientos. [En línea] 1 de Marzo de 2010. <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>.
19. *Sistema de Supervisión y Control Guardián del ALBA. Especificaciones Técnicas Generales*. 2013.
20. q-vision technologies. [En línea] 12 de Marzo de 2012. <http://www.qvision.us/es/servicios/gestion-de-requerimientos-de-software/generacion-y-verificacion-de-historia-de-usuario-scrum>.
21. ecured. [En línea] 2006. http://www.ecured.cu/index.php/Estilos_arquitectonicos.
22. Torrão, Graciano. Graciano Torrão's Deblog. [En línea] 15 de 1 de 2014. <http://gracianotorrao.com/2014/01/15/db02-exemplos-de-escrita-de-uma-aplicacao-para-acesso-a-base-de-dadostemp/>.

23. Fernández, Ruben. genbeta: dev desarrollo y software. [En línea] 14 de 7 de 2014. <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>.
24. Grosso, Andrés. Prácticas de software. [En línea] 21 de 3 de 2011. <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
25. geek the planet. [En línea] 11 de 5 de 2011. <http://geektheplanet.net/5462/patrones-gof.shtml>.
26. ALTOVA. [En línea] 2014. <http://www.altova.com/es/umodel/uml-component-diagrams.html>.
27. SPARX systems. [En línea] 2007. http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.
28. Universidad Nacional Autónoma de México. [En línea] 2010. <http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/>.
29. Lorenzo, Ariel Chávez. *Programa de mejora. Estándares de codificación y documentación para C++*. La Habana : s.n., 2010.
30. Pressman, R.S. *Ingeniería de Software, Un enfoque práctico*. s.l. : Mc Graw Hill., 2007.
31. Proyecto SCADA nacional. [En línea] 2011. <http://arcangel.belug.org.ve/proyecto-scada-nacional-nombre-codigo-galba/comment-page-1/>.
32. aie. [En línea] <http://www.aie.cl/files/file/comites/ca/abc/hmi.pdf>.
33. Geeks & Linux Atelier. [En línea] 15 de 5 de 2009. <http://glatelier.org/2009/05/15/qt-creator-desarrollando-aplicaciones-rapidamente/>.
34. Hipertexto. [En línea] <http://www.hipertexto.info/documentos/uml.htm>.