

DRIVER DE COMUNICACIONES ENTRE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC) LLAMADO SIBAS (SIEMENS BAHN AUTOMATION SYSTEM) Y UN SISTEMA SCADA (SISTEMA DE CONTROL Y ADQUISICIÓN DE DATOS) SOBRE PLATAFORMA DE SOFTWARE LIBRE, BASADO EN LOS ESTÁNDARES DE COMUNICACIONES INDUSTRIALES.

**CLAUDIA PATRICIA CASTILLO SIERRA
WILFRIDO DOMINGO LÓPEZ CUADRADO**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
BUCARAMANGA, SANTANDER, COLOMBIA.**

2012

DRIVER DE COMUNICACIONES ENTRE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC) LLAMADO SIBAS (SIEMENS BAHN AUTOMATION SYSTEM) Y UN SISTEMA SCADA (SISTEMA DE CONTROL Y ADQUISICIÓN DE DATOS) SOBRE PLATAFORMA DE SOFTWARE LIBRE, BASADO EN LOS ESTÁNDARES DE COMUNICACIONES INDUSTRIALES.

**CLAUDIA PATRICIA CASTILLO SIERRA
WILFRIDO DOMINGO LÓPEZ CUADRADO**

TESIS DE GRADO

Director del proyecto:

Daniel Arenas Seleey

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
BUCARAMANGA, SANTANDER, COLOMBIA.**

2012

NOTA DE ACEPTACIÓN:

FIRMA DEL PRESIDENTE DEL JURADO

FIRMA DEL JURADO

FIRMA DEL JURADO

Bucaramanga, mayo 4 de 2012

DEDICATORIA

A mis hijas Claudia Marcela y Laura Carolina

Claudia Patricia Castillo Sierra

DEDICATORIA

A mi esposa Claudia y mis Hijas Claudia Marcela y Laura Carolina por ser mi fuente de inspiración para seguir adelante.

Wilfrido López Cuadrado

AGRADECIMIENTOS

Nuestros mayores agradecimientos son para la multinacional Drummond, por el apoyo y la confianza ofrecida en la realización de este proyecto.

Agradecemos también a nuestras familias por el infinito apoyo y cariño brindado.

Gracias a todos los que esperaron y confiaron en la concreción de este trabajo.

CONTENIDO

INTRODUCCIÓN.....	14
1.1 DISEÑO METODOLÓGICO.....	16
1.1.1 Actividades	16
1.2 ESTRUCTURA DEL TRABAJO	18
2. ESTADO DEL ARTE Y MARCO TEORICO	19
2.1 ESTADO DEL ARTE.....	19
2.2 REVISIÓN DEL ESTADO DEL ARTE DE LOS SISTEMAS SCADAS	21
2.3 ANÁLISIS DEL ESTADO DEL ARTE DE LOS SISTEMAS SCADAS.....	23
2.4 PALAS ELÉCTRICAS	27
2.4.1 Operación de las palas Bucyrus.....	29
2.4.2 Sistema de monitoreo local.....	29
2.4.3 Control electrónico de las palas BUCYRUS 495HR.....	30
2.4.4 Movimientos de las palas.....	31
2.5 SIBAS.....	33
2.5.1 Procesador central ZR.....	34
2.6 COMUNICACIONES INDUSTRIALES DE LA PALA	35
2.6.1 Comunicaciones del SIBAS.....	39
2.7 SISTEMA SCADA.....	40
2.7.1 Funciones del sistema SCADA	41
2.7.2 Componentes Básicos de los Sistemas SCADAS.....	41
2.7.3 Interconexión general de sistema SCADA a un proceso automático.....	42
2.8 OPC – UA	43
2.9 AMBIENTES DE DESARROLLO	44
2.9.1 Características del Ignition	44
2.9.2 Módulos.....	45
2.9.3 Lenguaje de Programación Python.	46
2.10 CONSIDERACIONES PARA EL DISEÑO DE HMI	46
3. DISEÑO DE LA PLATAFORMA SCADA PARA MONITOREAR LAS PALAS ELÉCTRICAS 41, 42, 43 Y 44 EN LA COMPAÑÍA DRUMMOND LTD.	49

3.1 DEFINICIÓN DE REQUISITOS	49
3.2 ESPECIFICACIÓN DE REQUISITOS	50
3.3 DIAGRAMA UML	51
3.3.1 Diagrama de casos de usos.....	51
3.3.2 Diagrama de Actividad	56
3.3.3 Diagrama de Componentes.....	68
3.3.4 Diagrama de Despliegue	68
3.4 MODELO ENTIDAD RELACIÓN	69
3.4.1 Tablas.....	70
3.5 DISEÑO DE LA INTERFAZ DE USUARIO	73
3.5.1 Mapa de navegación del sistema SCADA	73
3.5.2 Descripción de las pantallas gráficas	74
4. IMPLEMENTACIÓN DE LA PLATAFORMA SCADA PARA MONITOREAR LAS PALAS ELÉCTRICAS 41, 42, 43 Y 44 EN LA COMPAÑÍA DRUMMOND LTD.	80
4.1 REQUERIMIENTOS.....	80
4.1.1 Requerimientos de hardware	80
4.1.2 Requerimientos de software:	81
4.2 HMI IMPLEMENTADO EN IGNITION.....	83
4.2.1 Configuración de Ignition gateway	83
4.3 ANIMACION DE LAS PALAS EN LA PANTALLA PRINCIPAL	99
4.4 DRIVER DE COMUNICACIÓN DEL PLC SIBAS	107
5. CONCLUSIONES	111
6. RECOMENDACIONES Y TRABAJOS FUTUROS	113
BIBLIOGRAFIA.....	114
ANEXO 1	115
ANEXO 2.....	119
ANEXO 3.....	125
ANEXO 4.....	132

LISTA DE FIGURAS

FIGURA 1: CICLO DE VIDA ITERATIVO	16
FIGURA 2: PALA ELÉCTRICA BUCYRUS	28
FIGURA 3. INTERFACE LOCAL	30
FIGURA 4: PLC SIBAS	34
FIGURA 5. RED DE PLC	36
FIGURA 6. PANEL DE COMUNICACIÓN DEL ROCKY	37
FIGURA 7. COMPONENTES DE UN SISTEMA SCADA.....	42
FIGURA 8. ESQUEMA GENERAL DE INTERCONEXIÓN DE UN SISTEMA SCADA Y EL PROCESO DE CONTROL.....	42
FIGURA 9. PALA BUCYRUS	49
FIGURA 10. CASOS DE USO DEL SISTEMA SCADA	52
FIGURA 11. CASO DE USO: RECONOCER ALARMAS ACTIVAS	56
FIGURA 12. CASO DE USO: INGRESAR AL SISTEMA SCADA.....	57
FIGURA 13. CASO DE USO: ACCESAR A HISTÓRICOS DE ALARMAS.....	58
FIGURAS 14. CASO DE USO: VISUALIZAR VISTA DE LOS SUBSISTEMAS DE LAS PALAS	59
FIGURA 15. CASO DE USO: VISUALIZAR DIAGRAMA UNIFILAR DE LA PALA.....	60
FIGURA 16. CASO DE USO: VISUALIZAR ESTADOS DEL PLC	61
FIGURA 17. CASO DE USO: VISUALIZAR ANIMACIÓN EN TIEMPO REAL DE LA PALA	62
FIGURA 18. CASO DE USO: VER TENDENCIAS.....	63
FIGURA 19. CASO DE USO: VER DOCUMENTACIÓN DE FALLAS.....	64
FIGURA 20. CASO DE USO: ADQUIRIR DATOS DEL PLC	65
FIGURA 21. CASO DE USO: CREAR USUARIOS	66
FIGURA 22. CASO DE USO: CAMBIAR PASSWORD	67
FIGURA 23. MODELO ENTIDAD RELACIÓN	69
FIGURA 24. MAPA DE NAVEGACIÓN.....	74
FIGURA 25. VENTANA DE NAVEGACIÓN.....	74
FIGURA 26. VENTANA PRINCIPAL	75
FIGURA 27. VENTANA DE OVERVIEW	75
FIGURA 28. VENTANA NETWORK PLC	76
FIGURA 29. NODO CROWD.....	76
FIGURA 30. ACTIVE ALARMS	77
FIGURA 31. HISTÓRICO DE ALARMAS	77
FIGURA 32. CABINA DEL OPERADOR	78
FIGURA 33: MOTOR TRENDS	79
FIGURA 34. CONFIGURACIÓN OPC –UA	84
FIGURA 35. CONFIGURACIÓN PLC SIEMENS.....	85
FIGURA 36. CONFIGURACIÓN AVANZADA DEL PLC SIEMENS	86
FIGURA 37. CONFIGURACIÓN DE LOS PQM.....	87
FIGURA 38. CONFIGURACIÓN AVANZADA.....	87

FIGURA 39. CONFIGURACIÓN SISTEMA CONTRA INCENDIO	88
FIGURA 40. CONFIGURACIÓN DE BASE DE DATOS.....	89
FIGURA 41. CONFIGURACIÓN DE ALARMAS	90
FIGURA 41: ALARMAS ACTIVAS	90
FIGURA 43: PROPIEDADES DEL OBJETO ALARMA.....	91
FIGURA 44. ALMACENAMIENTO DE ALARMAS	92
FIGURA 45. CONFIGURACIÓN DEL ALMACENAMIENTO DE ALARMAS.....	92
FIGURA 46. CONFIGURACIÓN DE NOTIFICACIÓN DE ALARMAS.....	94
FIGURA 47. FILTRADO DE ALARMAS	94
FIGURA 48. LISTADO DE EMAIL.....	95
FIGURA 49: DEFINICIÓN DE ROLES	96
FIGURA 50. PROPIEDADES ACTIVE DIRECTORY	98
FIGURA 51: POLÍTICAS DE SEGURIDAD PARA LOS COMPONENTES.....	99
FIGURA 52 .MOVIMIENTO DEL HOIST.....	100
FIGURA 53. MOVIMIENTO DEL CROWD.....	100
FIGURA 54. MOVIMIENTO DE PROPEL	101
FIGURA 55. ENCODER.....	101
FIGURA 56. MOVIMIENTO DEL HOIST.....	102
FIGURA 57. MOVIMIENTO DEL CROWD.....	103
FIGURA 58. MOVIMIENTO DE PROPEL	103
FIGURA 59. POSICIÓN 1 DEL BUNKED.....	104
FIGURA 60. POSICIÓN 2 DEL BUNKED.....	105
FIGURA 61. POSICIÓN 1 DEL BUNKED.....	105
FIGURA 62. POSICIÓN DE REPOSO DE LAS PALAS	106
FIGURA 63. SCRIPT DEL MARTILLO	106
FIGURA 64: VISTA DEL DATABLOCK DEL PLC SIBAS.....	107
FIGURA 65. CONFIGURACIÓN TIEMPO DE SCAN DEL PLC SIBAS.	109
FIGURA 66. CONFIGURACIÓN DE PARÁMETROS DEL DRIVER DE COMUNICACIÓN.....	110
FIGURA 67. DIAGRAMA DE DESPLIEGUE.....	125
FIGURA 68. PANTALLA INICIAL DEL SISTEMA SCADA	126
FIGURA 69. BOTONES PARA CONMUTAR	126
FIGURA 70. ACCESO A FUNCIONALIDADES DEL SISTEMA SCADA	127
FIGURA 71. RED DE PLC	128
FIGURA 72. DIAGRAMA UNIFILAR	128
FIGURA 73. TENDENCIAS	129
FIGURA 74. CALENDARIO DE EVENTOS	129
FIGURA 75. CAMBIO DE PASSWORD	130
FIGURA 76. HISTÓRICO DE ALARMAS	130
FIGURA 77. ALARMAS ACTIVAS.	131

LISTA DE TABLAS

TABLA 1. FABRICANTES SISTEMAS SCADAS	22
TABLA 2. CARACTERÍSTICAS SISTEMAS SCADAS	24
TABLA 3. PLAN DE COLORES	46
TABLA 4. RECONOCER ALARMAS ACTIVAS.....	53
TABLA 5. ACCESAR A HISTÓRICOS DE ALARMAS.....	53
TABLA 6. VER TENDENCIAS	53
TABLA 7. VISUALIZAR ANIMACIÓN EN TIEMPO REAL DE LAS PALAS ELÉCTRICAS.	53
TABLA 8. VISUALIZAR VISTA DE LOS SUBSISTEMAS DE LA PALA.	53
TABLA 9. VISUALIZAR DIAGRAMA UNIFILAR	54
TABLA 10. VISUALIZAR ESTADOS DE LA RED DE PLC	54
TABLA 11. ADQUIRIR DATOS.....	54
TABLA 12. INGRESAR AL SISTEMA SCADA	54
TABLA 13. VER DOCUMENTACIÓN DE FALLAS	54
TABLA 14. CREAR USUARIOS	55
TABLA 15. CAMBIAR PASSWORD	55
TABLA 16. ROLES	70
TABLA 17. USERS	70
TABLA 18. USER_ROLES_MAPPING	71
TABLA 19. SHOVEL	71
TABLA 20. ALERT_LOG	71
TABLA 21. SHIFTREPORT.....	72
TABLA 22. SCHEDULE.....	72
TABLA 23. SPARE_CODE.....	73

RESUMEN

La compañía Drummond LTD llegó a Colombia para desarrollar un proyecto minero de clase mundial en medio del departamento del Cesar, particularmente en el área comprendida entre los municipios de El Paso, La Jagua de Ibirico y Chiriguaná, zona a la que se le denominó Mina Pribbenow.

Para el desarrollo de sus actividades mineras en el país, Drummond Ltd. ha realizado la importación de nuevas tecnologías entre ellas 4 palas eléctricas del fabricante Bucyrus, maquinaria que facilita de la manera más eficiente la extracción del carbón de las diferentes capas del subsuelo. En la actualidad la mayoría de las empresas del sector industrial buscan aprovechar al máximo las bondades de la tecnología que ha incrementado sus beneficios haciendo que los procesos industriales puedan ser operados y visualizados por medio de un PC con la ayuda de aplicaciones de monitoreo y control, con las que se obtienen información basada en una programación orientada a objetos, eliminando las arquitecturas basadas en texto. El propósito de esta tesis es implementar un driver de comunicación que nos permita leer la información del multiprocesador SiBAS que hace parte de la red de PLC y el sistema SCADA para el monitoreo de las palas Bucyrus.

En el diseño del SCADA se seleccionó el software Ignition, porque ofrece un servidor OPC-UA, con diversos protocolos de comunicación capaces de trabajar en una plataforma de software libre como es Linux Debian, siendo importante porque la Pala dispone de un controlador de movimientos llamado SiBAS y un PLC Siemens S7-400 con una red Profibus que interconecta 9 nodos remotos más otros dispositivos de campo entre los cuales están el PQM de General Electric y el sistema contra incendio Fenwall.

Finalmente, se implementa la plataforma SCADA siguiendo las directrices del diseño propuesto y cumpliendo con cada uno de los requisitos funcionales. Como resultado, se obtuvo un sistema capaz de monitorear las variables del proceso, generar alarmas, almacenar la información en una base de datos, generar históricos, histogramas y permite la comunicación a través de diferentes protocolos con muchos dispositivos

de campo, todo esto de forma remota, sin importar la ubicación física de cada uno de los clientes.

El sistema SCADA de monitoreo remoto permitirá a la empresa disminuir el costo asociado de operación y mantenimiento e incrementando la producción en la Mina Pribbenow.

INTRODUCCIÓN

El presente trabajo se fundamenta en el diseño de un sistema de monitoreo y adquisición de datos para las palas Bucyrus en la multinacional Drummond LTD.

El mundo de la automatización se ha desarrollado a una velocidad vertiginosa, su aplicación ha contribuido al reconocimiento universal de sus ventajas y beneficios asociados al ámbito industrial, que es donde tiene una de sus mayores aplicaciones debido a la necesidad de controlar un gran número de variables, sumado esto a la creciente complejidad de los sistemas. El control automático de procesos se usa fundamentalmente porque reduce el costo asociado a la generación de bienes y servicios, incrementa la calidad y los volúmenes de producción, la eliminación de errores y un aumento en la seguridad personal de los procesos entre otros beneficios asociados con su aplicación.

Actualmente los fabricantes de hardware y software de automatización industrial han desarrollado programas que permiten las actividades de adquisición de datos, supervisión y control de procesos aplicables para la plataforma Windows con licencias propietarias muy costosas.

En la multinacional Drummond una de las problemáticas presentadas en las palas Bucyrus son las constates perdidas de comunicación local entre el SiBAS-PLC y la interface gráfica hombre maquina llamada "WinCC" de siemens, por tanto surge como necesidad implementar una alternativa de comunicación entre el SiBAS, el PLC y un sistema SCADA para monitoreo remoto de las cuatro palas Bucyrus 495HR, permitiendo minimizar el tiempo down o paro de estas máquinas, y así aumentar la productividad y rendimiento de estos equipos.

Es importante destacar que el sistema actual trabaja únicamente de forma local y es inestable, debido a las condiciones no favorables de trabajo de la máquina, adicionalmente se encuentran expuestos a vibraciones y condiciones ambientales no adecuadas para el buen funcionamiento de la estación de trabajo del software actual.

Este proyecto tiene como objetivo general desarrollar un driver de comunicaciones entre un controlador lógico programable (PLC) llamado SiBAS (SIEMENS BAHN AUTOMATION SYSTEM) [1], [2], y un sistema SCADA (Sistema de control y adquisición de datos) sobre plataforma de software libre, basado en los estándares de comunicaciones industriales.

Como objetivos específicos:

- Conocer el estado del arte en torno a las aplicaciones de software para sistemas SCADAS con el objeto de seleccionar la herramienta de desarrollo que nos brinde los mejores beneficios.
- Implementar un driver que permita comunicar un Controlador Lógico Programable, llamado SiBAS del fabricante Siemens Energy y un sistema SCADA para la visualización de datos en un entorno de software libre del fabricante Inductive Automation.
- Implementar un sistema de monitoreo o supervisión de variables análogas y digitales sobre una interfaz gráfica en tiempo real de los subsistemas que conforman las cuatro palas Bucyrus.

Este proyecto surge principalmente por motivaciones personales de los autores, de la experiencia, necesidades, restricciones, monopolio existente detectado en el software privativo y reglas establecidas para la regulación de las comunicaciones en los procesos de automatización industrial, hecho que tiende a revertirse mediante el uso de tecnologías libres.

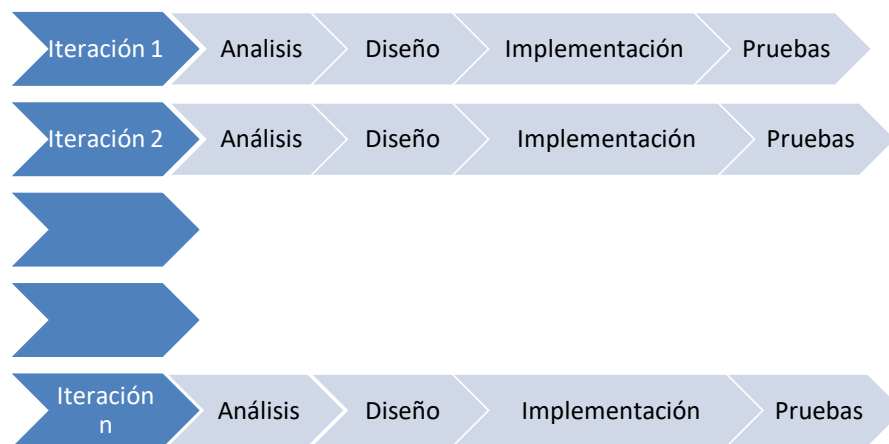
En el sistema SCADA que se propone se implementará en un entorno de software libre principalmente por las ventajas que nos ofrece este tipo de tecnologías entre ellas la estabilidad y robustez del sistema operativo Linux, la plataforma Ignition basada en Java, el gestor de base de datos MySQL, y el lenguaje de programación Python.

1.1 DISEÑO METODOLÓGICO

El desarrollo de este trabajo de grado está enmarcado en el enfoque de la investigación tecnológica aplicada, conocida también como investigación aplicada y desarrollo experimental orientado a la generación de conocimiento técnico y/o tecnológico cuya característica principal es la aplicación en la solución de una determinada problemática mediante el diseño de un proceso o producto.

En la propuesta metodológica para la gestión de este proyecto se ha elegido el ciclo de vida de tipo iterativo ver figura 1, considerando que este enfoque se caracteriza por que cada iteración es una parte del sistema parcialmente completo, probado, integrado y estable.

Figura 1: Ciclo de vida Iterativo



1.1.1 Actividades

El proceso para seguir el desarrollo del software nos fija una metodología estricta según el ciclo de vida iterativo, definiendo una serie de actividades que facilitarán la construcción de un sistema de mejor calidad.

La notación que se usará para modelar el sistema, es la definida por UML, de esta manera iniciará el ciclo de vida, capturando los requisitos y

culminará con el sistema funcionando, proporcionando así una visión completa y coherente del desarrollo de la aplicación.

Etapa 1. Estudio de Bibliografía.

Etapa 2. Desarrollo del driver, funcionalidades y sistema de supervisión

Etapa 2.1. Planificación y especificación de requisitos:

En esta etapa se hará la definición de requisitos y se establece aquellos que son críticos para el éxito del proyecto.

Definición de Requisitos.

Diagramas de Casos de Usos.

Etapa 2.2. Modelización del Sistema:

Diagrama de Actividades.

Diagrama de Despliegue

Diagrama de Componentes

Diseño de las Interfaces de Usuario

Etapa 3. Implementación

Construcción de Algoritmos.

Pruebas.

Integración

Etapa 4. Pruebas

Etapa 5. Documentación.

Etapa 6 Análisis de Resultados.

Etapa 7 Conclusiones.

1.2 ESTRUCTURA DEL TRABAJO

El documento está organizado en 6 capítulos en los cuales su contenido es el siguiente:

En el capítulo 2: están expuesto los conceptos teóricos de la Pala Eléctrica, red de PLC, el SiBAS y los sistemas SCADAS.

En el capítulo 3: se presenta el sistema desde un punto de vista funcional, especificación de requisitos, casos de usos, diagrama de actividad, diagrama de despliegue, diagrama de componentes y diseño de interfaces de usuario.

El capítulo 4 presenta la implementación del sistema SCADA, configuración del Gateway de Ignition y descripción del driver de comunicaciones.

En el capítulo 5 están expuestas las conclusiones del proyecto.

Y finalmente en el capítulo 6 alcances futuros del proyecto.

2. ESTADO DEL ARTE Y MARCO TEORICO

2.1 ESTADO DEL ARTE

Desde que surgieron los primeros autómatas programables o PLC en la década de los 70, se han generado dos grandes inquietudes, la primera hace referencia en la interacción de los PLC de diferentes fabricante; la segunda tiene como objetivo llevar y almacenar la información proveniente de los PLC (de las maquinas) a centros de cómputos ubicados en lugares remotos con sistemas de visualización e interfaces hombre máquina.

De lo anterior nacieron varias inquietudes, ¿quién y cómo solventar los problemas que surgiesen de un PLC específico?, para lo cual cada fabricante desarrollaba hasta aquel entonces su propia herramienta que ofreciera solución al problema puntual, adicionalmente, ¿cómo se establecerían las comunicaciones entre PLC de diferentes marcas, modelos y/o fabricantes?

Como respuesta, surge el estándar internacional IEC 61131 normalizando las características fundamentales de los controladores lógicos programables tanto en su parte de hardware como software, y en el año 1998, los fabricantes de PLC de la época y Microsoft como miembro principal crean la fundación OPC, una entidad encargada de generar un conjunto de reglas o estándares que permiten la regulación de un lenguaje de comunicaciones común entre PLC de diferentes fabricantes.

Hasta el año 2006 todas las especificaciones de la fundación OPC tenían como enfoque las tecnologías ofrecidas por Microsoft, soportadas en el uso del DCOM y sobre la plataforma del sistema operativo Windows, es decir, no había desarrollado de OPC servers para otros sistemas operativos, hasta ese momento Microsoft tenía todo el control.

De las diferentes limitantes y deficiencias detectadas del anterior OPC-DA basado en el DCOM de Microsoft, la Fundación OPC decidió un cambio total de las especificaciones del estándar al nuevo estándar llamado OPC-UA (OPC de arquitectura unificada)[3], y los fabricantes principales de PLC a nivel mundial, tales como Siemens, Schneider, Allen Bradley, General Electric, Schneider Electric, etc. Recientemente decidieron migrar al estándar OPC-UA.

Desligándose de la tecnología DCOM, la empresa Inductive Automation se hace miembro activo de la fundación OPC y pionera en el desarrollo de un OPC-UA multiplataforma.

El OPC-UA desarrollado por Inductive Automation ha roto los esquemas anteriores del estándar OPC basado en la tecnología DCOM de Microsoft, siendo esta empresa la primera a nivel mundial en construir un OPC-UA con herramientas de software libre y multiplataforma.

Referente a los sistemas SCADA (Sistema de Control y Adquisición de Datos) son líderes en el mercado mundial, los fabricantes de PLC como Siemens, Rockwell, General Electric, entre otros principalmente.

Siemens, posee un amplio SCADA propietario llamado WinCC quien deriva su desarrollo del lenguaje C y el cual solo funciona solo sobre plataforma Windows, con clientes limitados, para la conectividad con otros fabricantes requiere de un OPC Server externo, y conecta los requerimientos de su software SCADA mediante el cliente nativo OPC incorporado por siemens al WinCC.

Rockwell, por su lado posee su sistema SCADA llamado FactoryTalk Supervisory Edition, el cual funciona solo sobre plataforma Windows, recibe soporte de Microsoft con la cual tienen convenios. Su sistema SCADA incluye un OPC server propietario llamado RSLinx con el que conecta sus PLC, pero al momento de interactuar con otros fabricantes recibe soporte de la empresa Kepware, de quien deriva el nombre de su herramienta de software Kepserver (OPC Server externo a Rockwell fabricado por Kepware).

General Electric tiene el SCADA Proficy iFIX el cual es cliente OPC nativo, con la capacidad de poderse conectar a múltiples OPC Servers Externos de otros fabricantes de PLC y de los fabricantes de OPC reconocidos.

En cuanto al software libre son pocas las empresas que basan su modelo de negocio en este tipo de aplicaciones de código abierto con enfoque industrial, existen algunos proyecto como Lintouch [4], una implementación para la creación de interfaz hombre máquina, desarrollada como un software de código abierto con licencia GNU GPL otorgándoles a los usuarios el derecho de modificar el software acorde a sus necesidades.

En Venezuela fue implementado el proyecto Argos[5], con el propósito de ofrecer algunos componentes bajo la filosofía de software libre; promoviendo el núcleo del sistema SCADA (Supervisory Control and Data

Adquisition) e integrando diferentes componentes de software que permiten el control de un proceso desde un centro de cómputo.

Rscada[6] es un moderno sistema de control desarrollado por Raditex de control AB. rSCADA es un sistema basado en UNIX, basado en estándares abiertos y protocolos. El sistema está diseñado para el control, la regulación, la recogida de datos y seguimientos de los procesos. Implementaron la biblioteca libmodbus que cubre las partes más utilizadas del protocolo MODBUS. Tiene soporte integrado para TCP / IP, UDP / IP y el transporte de datos en serie.

Inductive Automation[7] en su plataforma de software Ignition no solo ha creado un OPC-UA con las características de los fabricantes antes descritas, adicionalmente integra una potente SCADA, también multiplataforma, a 32 y 64 bits, que sirve de interface hombre máquina, y que utiliza la máquina virtual de JAVA, con fácil integración de las diferentes bases de datos, como MySQL principalmente, SQL Server, Oracle, PostgreSQL, etc.

La segunda problemática quedo en el pasado, con el uso de las aplicaciones web, Inductive Automation y su sistema SCADA llamado Visión, con gráficos vectoriales, los cuales manejan bajo tamaño y alta resolución, hacen que cualquier máquina de computo, pueda tener desde un punto remoto la información en línea proveniente de los PLC o Maquinas.

2.2 REVISIÓN DEL ESTADO DEL ARTE DE LOS SISTEMAS SCADAS

Alrededor de los años sesenta la tendencia en automatización era la de que cada fabricante debía resolver sus problemas de control por sí solo. Quien se encontraba ante un problema de automatización desarrollaba un elemento electrónico específico para solventarlo. Incluían una serie de entradas y salidas fijas y utilizaban generalmente lenguajes de programación poco conocidos. Los años setenta ven aparecer una nueva generación de autómatas de la mano de fabricantes de equipos eléctricos como Siemens o Allen-Bradley. Implementaron autómatas capaces de controlar grandes cantidades de entradas y salidas, ideales para industrias. No se trataba de entornos amigables, por lo cual estos controles estaban diseñados para soportar las condiciones más severas y, por tanto, eran grandes, pesados y muy costosos. Otra de las consecuencias

de la evolución de la electrónica fue la reducción de los componentes, lo que permitió realizar una disminución progresiva de tamaño, peso y coste en todos los niveles industriales de control. Resultado de esto fue la introducción de los micro PLC, en los años ochenta. Permitían realizar controles modulares que se adaptaban a las necesidades del momento y venían provistos ahora de sistemas de programación genéricos (ladder o escalera), lo que les deparó un éxito inmediato en todo el ámbito industrial.

De una forma u otra, cada vez que se ha realizado el control de un sistema, grande o pequeño, ha sido necesario tener información visual de cómo está funcionando. Así, a medida que los sistemas de control han ido evolucionando y se han hecho cada vez más complejos, ha aumentado también la complejidad de los elementos que proporcionan la información al usuario. De un simple indicador de aguja, que representa una variable del proceso (por ejemplo, la presión de aire en una instalación neumática), se ha llegado a grandes paneles sinópticos que muestran el estado de grandes instalaciones.

La aparición de la informática permitió realizar este tipo de control de manera más sencilla. Los grandes cuadros de control empezaban a convertirse en monitores que podían mostrar la misma información. Vista la necesidad, varios fabricantes desarrollaron entonces paquetes de software capaces de comunicarse con los sistemas de control existentes y permitieron así una flexibilidad de uso no imaginada hasta el momento. Esta tendencia ha ido en aumento, de tal manera que hoy día las opciones existentes son numerosas (Ver tabla 1). Algunos de los más conocidos:

Tabla 1. Fabricantes Sistemas SCADAS

FABRICANTE	SOFTWARE SISTEMAS SCADAS
Intellution	IFIX
Omron	SCS
Siemens	Wincc
Rockwell Automation	Rswiew – Factory Talk View SE
Wonderware	Intouch
GeFanuc	Cimplicity

Con la incursión de Internet en el mundo de las comunicaciones industriales ahora es posible conectarse con un sistema de control situado en cualquier lugar del mundo gracias a la tecnología Web-Server: un ordenador dotado de un explorador y la dirección IP del sistema que queremos visualizar serán suficientes.

2.3 ANÁLISIS DEL ESTADO DEL ARTE DE LOS SISTEMAS SCADAS

Aunque diferentes en manejo y con pequeñas funciones específicas, los sistemas SCADA constan de funcionalidades comunes a todos ellos. En la siguiente lista se muestran las características generales comunes a los sistemas SCADA:

- Gráficos de tendencias, gestión de alarmas, etc.
- Generación de informes.
- Scripts para realizar funciones y rutinas muy utilizadas.
- Inclusión de controles Activex, COM/DCOM, OLE, etc.
- Unificación de supervisión en pantallas/ordenadores.
- Integración entre software de programación PLC y la aplicación SCADA.
- Biblioteca de símbolos y gráficos para la realización de pantallas.
- Soporte ADO/ODBC para leer o registrar datos en bases de datos.
- Comunicación con otras aplicaciones vía OPC.

Existen numerosos sistemas de supervisión y control, dado que cada fabricante importante de equipos de automatización proporciona un SCADA propietario, y adicionalmente, terceras empresas proporcionan software SCADA con diferentes funcionalidades, enfocados a sectores concretos, atendiendo a ciertas normativas, etc.

Este proyecto se maneja en un entorno heterogéneo, las palas Bucyrus posee variedad de dispositivos de distintos fabricantes tales como SIEMENS, ENERGY, PQM, los cuales han desarrollado su propio software como una tecnología propietaria, dificultando la integración de estos equipos de diferentes marcas en un solo sistema SCADA.

Cabe destacar que para un sistema heterogéneo se deben revisar algunos aspectos para la elección del software SCADA:

Inclusión de drivers de comunicación para otros fabricantes de autómatas.

Visualización y control a través de un navegador web. Publicación web de las pantallas de monitorización y control.

OPC: El protocolo estándar para intercambio de datos.

La tabla 2 resume y recoge esquemáticamente las diferencias entre los diferentes fabricantes de software para sistemas SCADAS basados en los siguientes parámetros:

Tabla 2. Características Sistemas SCADAS

FABRICANTE	SIEMENS	ROCWELL	INDUCTIVE
SOFTWARE SCADA	WINCC	Factory View SE	Talk Ignition Vision
SISTEMA OPERATIVO	Windows	Windows	Multiplataforma Linux MAC Windows
OPC	SIMATIC server	NET OPC	RSLink Ignition OPC- UA Server
DRIVER	Productos SIEMENS	Productos ROCKWELL	SIEMENS ALLEN BRADLEY MODBUS
BD	Sybase SQL	SQL Server	MySQL SQL Server PostgreSQL Oracle
Licencia	Propietario	Propietario	Propietario

Sistema SCADA	VB-Scripting y C-Scripting	Visual Basic	Python
	Autenticación Roles	por Autenticación por Roles	Autenticación por Roles. Directorio Activo
Licencia OPC	Propietario	Propietario	Gratis
Acceso Simultaneo Clientes	Configuración mixta: 32 clientes + 3 clientes Web	5 Máximo	Ilimitado
	Configuración mixta: 50 clientes Web + 1 cliente de WinCC		

La capacidad de integrar distintas tecnologías de Autómatas para algunos fabricantes como SIEMENS, ROCKWELL, GE Fanuc, Wonderware entre otros, es imposible realizarla a través de su propio servidor OPC porque han sido desarrollados como simples programas que sólo pueden manejar un único tipo de dispositivo o protocolo. El resultado de esto es que son necesarios múltiples servidores OPC para manejar distintos tipos de dispositivos y/o protocolos. De esta necesidad han surgido proveedores especializados en driver de comunicación OPC para la conectividad industrial con los diferentes sistemas HMI. Matrikon y Kepware son líderes en este tipo de tecnología sin embargo esto implica comprar licencias adicionales para obtener una integración.

Ignition OPC-UA es un servidor OPC que tiene controladores y se configura para comunicar con PLCs, pero también es cliente OPC-UA, eso significa que puede conectarse a cualquier servidor OPC-UA. Ignition ha implementado drivers para PLCs Allen-Bradley, Siemens, Dispositivos Modbus Ethernet, TCP y UDP permitiendo utilizar múltiples drivers de comunicación de modo concurrente, también dispone de un driver API libre con el cual se puede escribir controladores y conectar cualquier dispositivo a la tecnología OPC de Ignition.

Una de las diferencias significativas entre los sistema de Supervisión analizados para esta investigación fue encontrar una aplicación compatible con múltiples plataforma facilitando a los usuarios finales usar diversos entornos, brindando la posibilidad de trabajar con tecnologías de software libre y principalmente porque podemos usar el sistema operativo Linux Debian en el servidor, proporcionando estabilidad, seguridad y confiabilidad para la aplicación.

2.3.1 Licencias

Con el fin de tener una flexibilidad que satisfaga todas las exigencias, WinCC está organizado modularmente. Se distingue entre licencias RC (Runtime y configuración para desarrollo) y meras licencias RT (sólo Runtime). Si en un PC solamente se quiere utilizar WinCC en Runtime, sin modificar desde allí la configuración, únicamente se necesita una licencia RT para ese PC.

Las licencias están clasificadas, además, atendiendo al número máximo de variables de proceso y de variables de fichero que se pueden utilizar en RT. Con la licencia "WinCC RC (2048)", se pueden emplear como máximo 2048 variables de proceso.

El sistema de Licenciamiento en el Factory Talk View SE atiende al número de pantallas: 25, 100, 250 o sin límite distinguiéndose licencias Runtime y Licencias Edition.

En Ignition no es necesario decidir qué licencia se utilizará y cuantos Usuarios y Variables se permiten; Eso significa licencia ilimitada para añadir todas las Variables que necesite y hacer llegar los Datos a todos los Usuarios.

2.3.2 Seguridad

En lo que respecta a la seguridad, FactoryTalk y WinCC ofrecen autenticación centralizada y control de acceso mediante la verificación de la identidad de cada usuario que accede a los sistemas críticos. Luego, concede o deniega su petición para realizar acciones específicas basándose en su nivel de acceso. En ese sentido Ignition ofrece adicionalmente autenticación en el directorio Activo a través del protocolo LDAP permitiendo integrar el esquema de autenticación corporativa

No existe un software universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Es necesario analizar el contexto del proyecto (recursos técnicos, humanos, tiempo de desarrollo, tipo de sistema, etc) para decidir cual se adapta y proporciona los beneficios que permitan no solo cumplir con los objetivos propuestos.

En la actualidad las organizaciones necesitan controlar sus procesos de forma eficiente integrando toda su arquitectura tecnológica en un sistema robusto, flexible, escalable y fácil de usar. También, la mayoría de los sistemas de supervisión y control de procesos han enfocado su diseño hacia aplicaciones Web permitiendo el acceso remoto desde cualquier ubicación por medio de una interfaz segura de monitorear, y de registrar en tiempo real, almacenando históricos y parámetros principales de procesos.

Considerando los requerimientos del sistema SCADA de las Palas Bucyrus y atendiendo al análisis realizado de las tecnologías existentes en el mercado para el desarrollo del proyecto elegimos la tecnología de Ignition de Inductive Automation para la implementación del sistema SCADA.

2.4 PALAS ELÉCTRICAS

La historia de Drummond en Colombia es reciente, llegó a Colombia en 1987 y solo en 1995 pudo iniciar sus operaciones después de los trámites y procedimientos legales para desarrollar un proyecto minero de clase mundial en el departamento del Cesar, particularmente en el área comprendida entre los municipios de El Paso, La Jagua de Ibirico y Chiriguáná, zona a la que se le denominó Mina **Pribbenow**, proyecto carbonífero La Loma.

Para el desarrollo de sus actividades mineras en el país (Plan Minero), Drummond Ltd. ha realizado la importación de nuevas tecnologías, maquinarias y conocimientos que permiten de la manera más eficiente la extracción del carbón de las diferentes capas del subsuelo.

Es así como el impulso de importantes proyectos de infraestructura de relevancia nacional, como la rehabilitación de la línea férrea entre Ciénaga y el centro del Cesar, y la construcción y operación del primer puerto privado que tuvo Colombia en virtud del Plan de expansión portuario y modernización del sector en 1991.

Básicamente el plan de minería es la programación de cada uno de los equipos dentro de las diferentes áreas de la mina de manera que se destape la mayor cantidad de carbón en una forma óptima y eficiente.

Se hace dos tipos de planes mineros:

A corto Plazo (3-6 meses)

A largo plazo > 6 meses

Variable que inciden en el plan minero:

Estadísticas de productividad de los equipos, disponibilidad mecánica y utilización.

Influencia del comportamiento geológico de los mantos que inciden sobre el método de minería a utilizar (Hace referencia a la pendiente del manto de carbón).

Comportamiento del Clima, análisis estadístico

Presupuesto preliminar de producción al año

Para cumplir con las metas de producción establecidas en el plan minero sobrepasando las adversidades climáticas y del terreno, Drummond Mina, considero la adquisición de cuatro palas Eléctricas de nombre Bucyrus (ver Figura 2), con las siguientes funcionalidades:

- Remover la capa de aluvial dejada por el avance de la Dragalina. (Equipos utilizados en desarrollo y avance de nuevas zonas de minado dada su gran capacidad).
- Extraer la roca de mantos de una altura entre 16 y 18 metros, como se ilustra a continuación:

Figura 2: Pala Eléctrica BUCYRUS



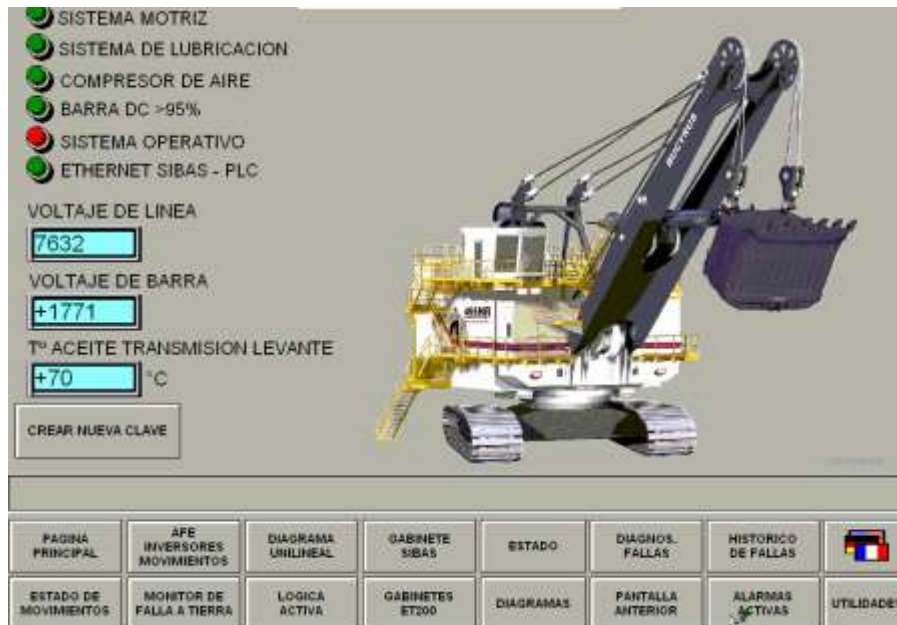
2.4.1 Operación de las palas Bucyrus. Antes del arranque, se realiza una inspección para asegurarse que esta lista para ser puesta en operación. Omitir esta revisión rutinaria podría generar pérdidas de tiempos innecesarios. Por ejemplo: una fuga de aceite no detectada podría resultar en una caja de transmisión seca, conduciendo eventualmente a un desgaste excesivo o a la destrucción de un engranaje, rodamientos trancados u otros problemas mecánicos y eléctricos.

Después del arranque y en operación se requiere de un monitoreo continuo de cada una de las partes y equipos internos que integran las palas Bucyrus, es así como surge la idea de los sistemas de monitoreo online local y online remoto.

2.4.2 Sistema de monitoreo local. De la complejidad de los subsistemas que conforman cada una de las partes de las palas, se hace necesario la implementación de una interface visual hombre-máquina capaz de describir el comportamiento y estado actual de las palas Bucyrus como se visualiza en la figura 3, la diferencia del local al remoto radica principalmente en el manejo seguro durante la operación de la máquina, excepto al operador en la cabina de mandos, nadie más puede estar dentro de la maquina cuando se está operando.

Cuando la maquina está detenida en fase de inspección o mantenimiento (Mecánico o Eléctrico), se utiliza la interface local, mientras que la maquina está en operación, se hace necesario de la interface remota para su monitoreo.

Figura 3. Interface Local



2.4.3 Control electrónico de las palas BUCYRUS 495HR

El control eléctrico y electrónico de las palas Bucyrus 495HR es realizado por dos grupos de controladores principales, el primero de ellos y el más importante, es llamado SiBAS [8] encargado del control motriz de la máquina. El segundo PLC SIEMENS[9] se encarga de todas las variables utilitarias a requerir que complementan el control motriz.

Funcionalidades del SiBAS vs PLC SIEMENS

Semejante a un control distribuido, el envío de señales de un grupo de procesadores al otro, y la ubicación de módulos remotos de entradas salidas distribuidos en diferentes puntos estratégicos de la pala, facilita y permite que cada grupo de controladores realice de la manera más clara y eficiente las siguientes funcionalidades:

Funcionalidades del SiBAS

A. Control del BUS DC, es una de las principales funciones del SiBAS, realizado mediante AFEs (Active Front End).

Los AFEs son fuentes activas de donde derivan su nombre, compuestas por capacitores y rectificadores trifásicos de alta potencia que convierten la energía AC a DC, se encargan de alimentar los inversores que controlan los motores para cada movimiento, y los IGBTs (Transistor Bipolar de Puerta Aislada) que conforman la etapa de rectificación son controlados por el SiBAS mediante PWM (modulación de ancho de pulso) con dos modos de operación, el primer modo como rectificador normal, y el segundo modo como inversor o popularmente llamado en electrónica de potencia "rectificador PWM", con este modo el AFE realiza devolución de energía a la red controlando:

El factor de potencia de la carga sostenida, en este caso de la pala.

Los armónicos generados por la máquina.

B. Control de todos los movimientos de la pala. Función llevada a cabo por los inversores de cada motor, también mediante PWM.

C. Protección de la sección de potencia.

Funcionalidades del PLC SIEMENS

A. Interface con el Operador.

B. Posicionamiento de los Límites de la Máquina.

C. Sistemas de Lubricación.

D. Sistemas Auxiliares, como la medición de temperaturas de los rodamientos, tacómetros, encoders, tablero de control de motores, frenos de propel, control neumático, escaleras de acceso, etc.

E. Protecciones de la Máquina.

F. Encendido y apagado del Bus DC y Movimientos.

2.4.4 Movimientos de las palas.

En conjunto con control del BUS DC, protecciones, control del factor de potencia y armónicos, el SiBAS compuesto por múltiples procesadores, se encarga de los movimientos principales de la máquina, como se describe a continuación [8]:

TRASLACION: Requerido para el desplazamiento de la máquina de un lugar a otro, también llamado Movimiento de Propel. Y es provisto por dos motores de 700HP/700V ubicados en la parte inferior de la máquina, y controlados por el SiBAS

GIRO: Controlado por el SiBAS, este movimiento lo realiza la maquina a 360 grado sobre su eje, normalmente es utilizado durante el cargue de material sobre los camiones, También es llamado Movimiento de Swing. Este movimiento es realizado por dos motores de 505 HP/700Vubicados en la parte media de la máquina, uno del lado derecho y otro en lado izquierdo.

SUBIDA BAJADA: También llamado movimiento de Hoist, lo realiza un solo motor con doble bobinado, de características especiales a 2240HP/1400V, su control es realizado principalmente por el SiBAS, se encarga de subir y bajar el balde o Bucket de la pala.

ATAQUE RETRACTAR: Es otro de los movimientos controlados por el SiBAS, Llamado también Crowd, lo realiza un potente motor de 700HP/1400V, como su nombre lo indica, este movimiento de ataque es utilizado para la recogida de material y retracción para posicionar el balde, y luego realizar el descargue del mismo.

IGBTs: Transistores Bipolares de Puerta Aislada, es una de las partes más importantes para el control de movimiento, este grupo de dispositivos, hacen de interface de potencia entre el SiBAS como unidades de procesamiento principal y los Motores que realizan los movimientos antes descritos.

2.5 SiBAS

Es una de las partes más importantes de la palas, con la cooperación del PLC se encarga de realizar el control de movimiento del Crowd, Hoist, Swing, Propel y AFEs, está compuesto por un procesador para los módulos de entradas y salidas, un procesador central llamado ZR, y cinco procesadores adicionales encargado de las protecciones electrónicas. A continuación los nombres dados por el fabricante para cada procesador:

ZR = Procesador Central o Procesador Principal, maneja toda la lógica de control del SiBAS, interactúa con los SIP y el procesador de módulos I/O.

Crowd SIP = Protecciones para el control del movimiento de empuje y retracción del balde.

Hoist SIP = Protección para el control de movimiento de subida y bajada del balde de la pala.

Swing SIP = Protecciones para el control del movimiento de Giro a 360 grados de las palas.

AFE 1,2 SIP = Cada pala posee cuatro AFE (Active Front End), básicamente son rectificadores trifásicos que convierten la energía AC a DC, compuestos por dispositivos de estado sólido (IGBT) de alta potencia, y encargados de alimentar los inversores que controlan los motores para cada movimiento. Las protecciones eléctricas del AFE 1 y 2 son coordinadas, en conjunto con el procesador central ZR.

AFE 3,4 SIP = Este procesador coordina las protecciones de los AFE 3 y 4, los cuales se interconectan a un barraje común de salida con el AFE 1 y 2, dando como resultado un voltaje DC rectificado de 1800 VDC aproximadamente.

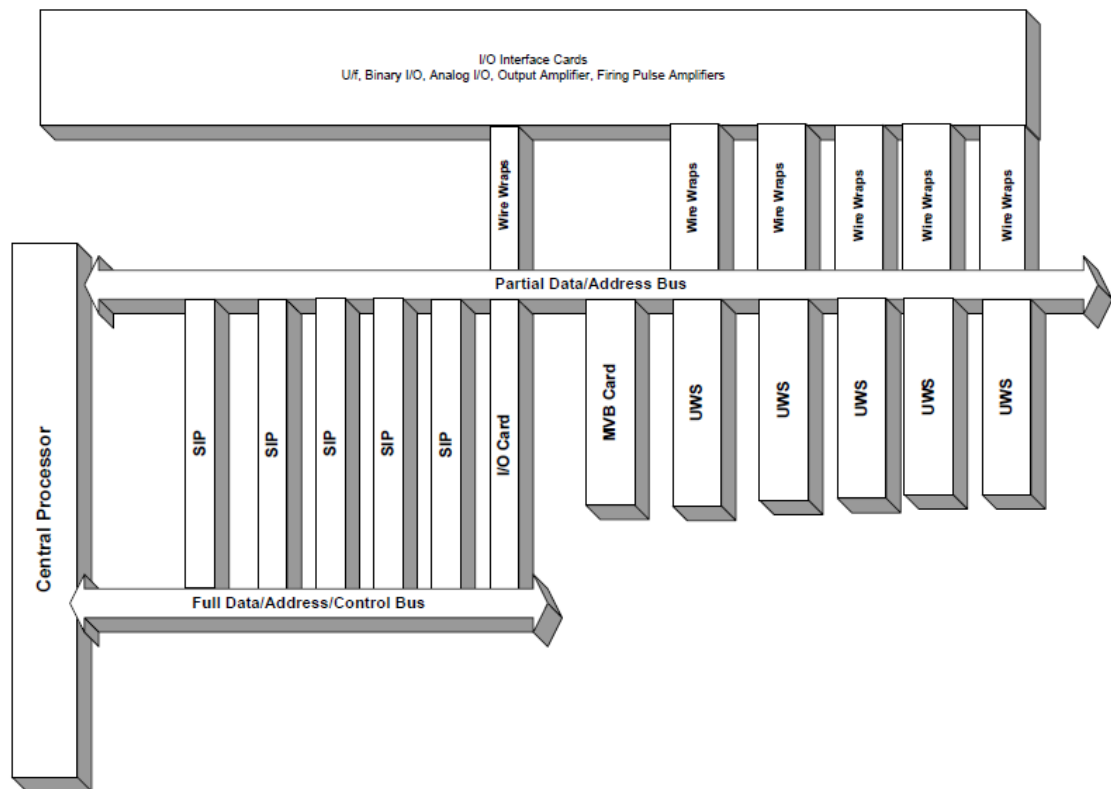
Procesador de Módulos I/O: Realiza el monitoreo y control de los módulos de entradas y salidas del SiBAS.

2.5.1 Procesador central ZR

Poca información se tiene de la lógica escrita en el procesador central, se conoce del funcionamiento del SiBAS en general y de las funciones asociadas a cada procesador, adicionalmente de que siemens posee un software de programación de este dispositivo llamado "SiBAS G".

En la figura 4 se visualiza el procesador central que se comunica con cada SIP y Procesador de módulos I/O mediante el bus de comunicaciones, de la misma manera con los módulos de entrada salidas.

Figura 4: PLC SiBAS



Otras de las funcionalidades de este procesador central, es la de enviar mediante el puerto de comunicaciones Ethernet, con protocolo http:/ 57 tags en código hexadecimal al PLC, donde este último, se encarga de procesarlo y posteriormente visualizar la información requerida en la interfaz gráfica local.

Se detalla la información del Datablock DB3502 en el Anexo 1 correspondiente a los Tags enviados por el Procesador Central del SiBAS, y recibida en el PLC S7 400 de Siemens.

2.6 COMUNICACIONES INDUSTRIALES DE LA PALA

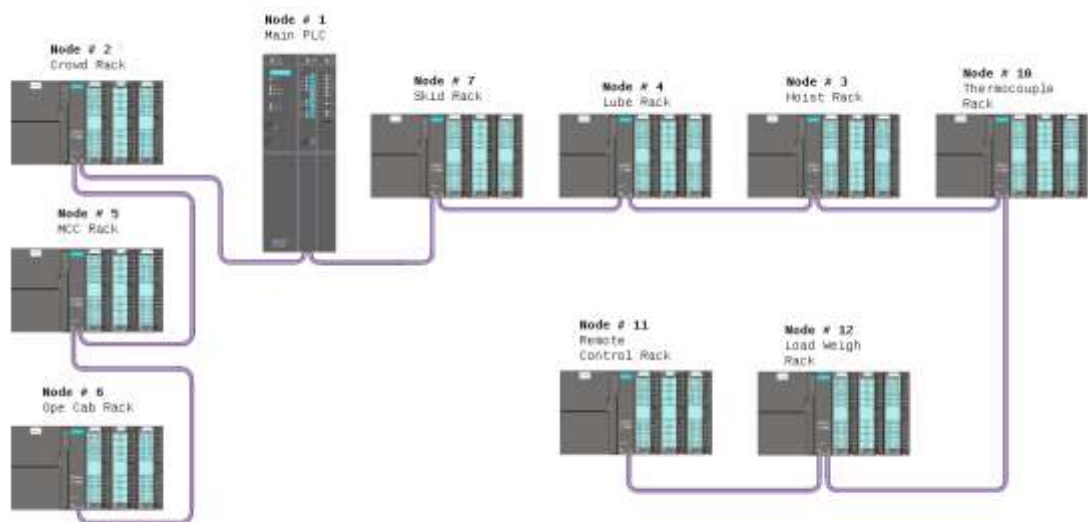
Básicamente se manejan dos tipos de comunicaciones dentro de la arquitectura de control de las palas Bucyrus 495HR.

La primera hace referencia al nivel de dispositivos de campo, llamada red Profibus-DP, y la segunda a nivel superior o de SCADA llamada Ethernet TCP/IP:

Red Profibus DP[8]: Permite máximo hasta 32 estaciones sin repetidor, cada pala Bucyrus 495HR maneja nueve estaciones ubicadas de manera estratégica alrededor de la pala, este protocolo es un estándar de comunicaciones para buses de campo y de dispositivos, la terminología DP hace referencia a periferia distribuida, cuando se comunica vía RS-485 utiliza un par de cobre trenzado apantallado con recubrimiento plástico violeta, permitiendo velocidades entre 9.6 kbps y 12 Mbps (Caso de las palas).

La red Profibus como se visualiza en la Figura 5 conecta los nodos siguientes con el procesador central o PLC S7400 de siemens:

Figura 5. Red de PLC



Nodo del Crowd: Encargado de manejar las señales de campo asociadas al movimiento de ataque y retracción.

Nodo del Hoist: Recibe las señales provenientes del movimiento de subida y baja del balde o bucket.

Nodo del Lubricación: Contiene los módulos remotos encargados de manejar las señales de los diferentes sistemas de lubricación.

Nodo MCC: Encargado del tablero de control de motores auxiliares.

Nodo Operador: Recibe las señales de los joysticks y controles de operador.

Nodo del Skid: Hace referencia al conjunto de módulos de entrada-salidas que hacen de interface física o cableado en duro (Ver Comunicaciones del SiBAS), entre el SiBAS y el PLC principal.

Nodo Thermocouple: Recibe el estado de temperaturas de los rodamientos de motores y reductores de la pala.

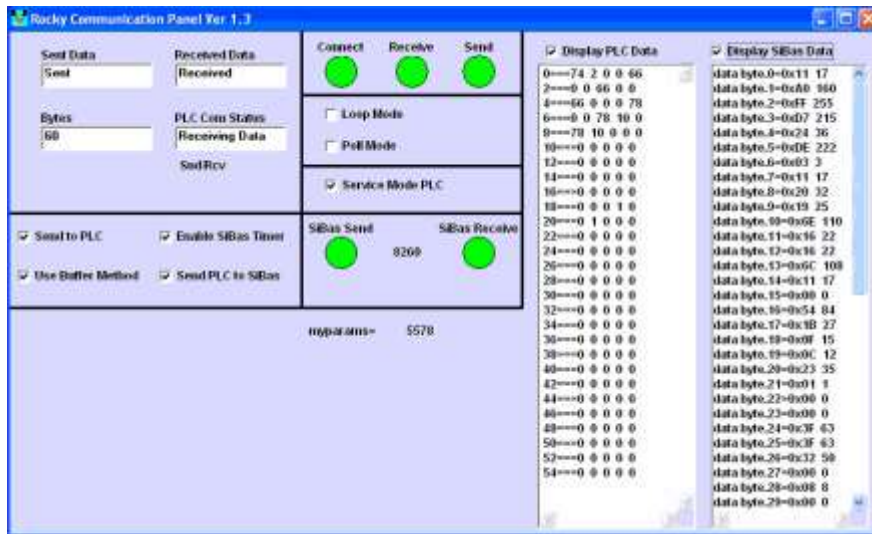
Nodo Remoto Control: Permite el control remoto de los movimientos del Hoist mediante una botonera localizada en la parte media de la pala.

Nodo Load Weigh: También llamado medidor de peso, se encarga de las señales de registro del tonelaje o carga movidas por la pala.

Red Ethernet: Con una velocidad de transmisión local de 100Mbps, en esta red se establecen las comunicaciones en el siguiente orden:

Entre el PLC principal S7400 con el SiBAS mediante la herramienta de software Rocky Simatic Server (ver panel de comunicación en la Figura 6). Para esta aplicación se requiere de un PC externo.

Figura 6. Panel de Comunicación del Rocky



Entre el PLC S7400 y el WinCC instalado en el PC Industrial Externo llamado Rocky (Estación de Mantenimiento)

Entre el PLC S7400 y el Computador del Operador ubicado en la cabina de mandos sobre la pala.

De los equipos adicionales que no vienen originalmente con la pala, como el medidor de energía PQM (Power Quality Meter) de General Electric y del Sistema Contraincendios Fenwall a la red de Drummond mediante la interconexión de radio.

Red de Radiofrecuencia: Esta red permite comunicar los PQM, Las redes Contraincendios, los PLC S7400, y los SiBAS de las cuatro palas con la red de Drummond, tiene 1Mbps de velocidad, su ancho de banda es compartido y depende del número de usuarios conectado, su tipo de transmisión es Mesh Inalámbrico y El radio hornet de las palas se conecta al MWR o IAP más cercano, el IAP se conecta al backhaul Canopy, y el Canopy al clúster de la torre del despacho, esta última se conecta al switch LAN de la red de Drummond.

- Una red MESH[11] es aquella que emplea uno o dos arreglos de conexión, una topología total o una parcial. En la total, cada nodo es conectado directamente a los otros. En la topología parcial los nodos están conectados solo a algunos de los demás nodos". El término "MESH" es a menudo usado como un sinónimo de "ad hoc" o red móvil. Obviamente combinando las dos características de la topología MESH y las capacidades de ad hoc.
- El MWR [12] es un enrutador inalámbrico usados para iniciar nuevas implementaciones de redes y asegurar suficiente cobertura mientras la población del cliente va creciendo. El MWR contiene dos radios de 802.11 (Wi-Fi) en conformidad con los estándares y dos de los más reconocidos radios móviles de banda ancha con Acceso de Movilidad Habilitado (MEA). El MWR dispone de dos puertos Ethernet de alta velocidad y soporta un total de seis direcciones de IP (tres direcciones por puerto). Esto permite que una red de dispositivos habilitados por IP sean directamente identificados, accedidos y administrados sobre una red Mesh de Banda Ancha de Motorola.

- El Sistema Canopy se basa en una tecnología de banda ancha inalámbrica que brinda acceso a Internet a alta velocidad y que ha sido diseñado para brindar un acceso a datos rentable, de alta velocidad y "de última milla" a clientes que antes no estaban suficientemente bien atendidos o que vivían en áreas donde no existía infraestructura. El Sistema Canopy utiliza redes Punto a Punto y Punto a Multipunto que pueden cubrir distancias que van de 3.2 a 16Km. (2 a 10 millas) en una configuración multipunto, y hasta 56 Km.

2.6.1 Comunicaciones del SiBAS.

Existen dos tipos de comunicaciones entre el PLC y el SiBAS, la primera, se realiza vía hardware, también llamada cableado en duro, y la segunda vía Ethernet.

El tipo de comunicación vía cableado en duro, es crítica para la operación de la máquina, maneja señales y lazos de control muy puntuales, cada lazo de control es monitoreado permanentemente, esto quiere decir, que si un lazo no se ejecuta en un tiempo determinado, automáticamente se genera una falla y por consiguiente paro de la máquina.

La comunicación vía Ethernet, no es considerada como crítica, la maquina puede operar incluso si el cable de comunicaciones Ethernet se rompe o se desconecta, pero la información de alarmas y el estado de las variables que inciden sobre una falla, le dan un nivel de importancia tal que se requiere de una conectividad permanente.

Siemens no tiene definido una comunicación directa de dispositivos, es decir entre el PLC S7400 y el SiBAS, para ello utiliza una interface de software que funciona sobre un PC y en plataforma Windows, esta interface o herramienta de software propietaria llamada Rocky Simatic Server, descrita anteriormente, realiza sus comunicaciones basada en el protocolo http

Con el Rocky Simatic Server, la comunicación entre el PLC S7 400 de Siemens y el SiBAS se realiza de manera bidireccional, luego el PLC la

procesa para ser visualizada en el sistema de monitoreo local por el WinCC.

WinCC

Este paquete de software, fue desarrollado por Siemens[13], bajo las normas de un sistema SCADA para la visualización y control de procesos industriales, entre sus características más importantes tenemos:

- Arquitectura de desarrollo abierta (Programación en lenguaje C).
- Soporte de tecnologías Active X.
- Manejo de comunicaciones OPC.
- Comunicación mediante drivers.
- Programación en línea, sin detener el runtime se puede actualizar las modificaciones en la aplicación.

Para el caso de las palas Bucyrus 495HR, el PLC Siemens S7 400 es el único dispositivo que envía y recibe datos de la interface gráfica local WinCC, no existe un driver definido dentro del WinCC capaz de manejar las comunicaciones industriales con el SiBAS, razón por lo cual el uso de la interface externa Rocky Simatic Server antes descrita.

2.7 SISTEMA SCADA

Los sistemas SCADA [14] (Supervisory Control And Data Acquisition) son aplicaciones de software, diseñadas con la finalidad de controlar y supervisar procesos a distancia. Proveen de una perspectiva integrada de todos los recursos de control e información de la planta, de esta manera, los usuarios pueden visualizar e interactuar con los procesos mediante una interfaces gráficas que muestra el comportamiento del proceso en tiempo real.

Estos sistemas actúan sobre los dispositivos de campos instalados en la planta, como son los controladores o autómatas, sensores, actuadores, motores, etc. Generalmente se vincula el software al uso de una computadora y de un PLC, la acción de control es realizada por los

controladores de campo, el cual toma las señales y las envía a las estaciones remotas usando un protocolo determinado sin embargo la comunicación del sistema con el operador generalmente se realiza mediante una computadora.

2.7.1 Funciones del sistema SCADA

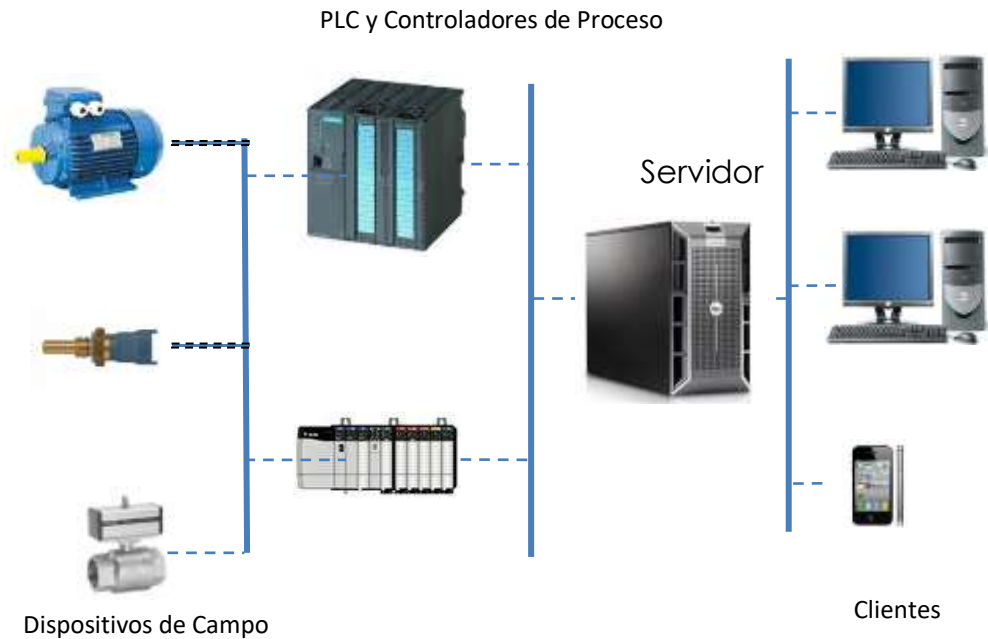
El sistema SCADA debe cumplir cinco funciones principales:

1. Adquisición de datos: Como su nombre lo indica, el sistema recoge, procesa y almacena la información recibida del proceso, en forma continua y confiable.
2. Transmisión de datos: La información recopilada debe ser enviada al lugar necesario para su tratamiento.
3. Presentación de datos: representación gráfica de los datos. Interfaz del Operador o HMI (Human Machine Interface).
4. Gestión de alarmas: Generar alarmas para que el operador realice las acciones de control necesarias para corregir las situaciones anómalas que se presenten.
5. Explotación de los datos adquiridos para gestión de la calidad, control estadístico, gestión de la producción, comportamiento de los subsistemas de la pala y gestión administrativa y financiera.

2.7.2 Componentes Básicos de los Sistemas SCADAS

La figura 7 muestra una arquitectura típica de un sistema SCADA para el monitoreo y control de procesos industriales:

Figura 7. Componentes de un sistema SCADA

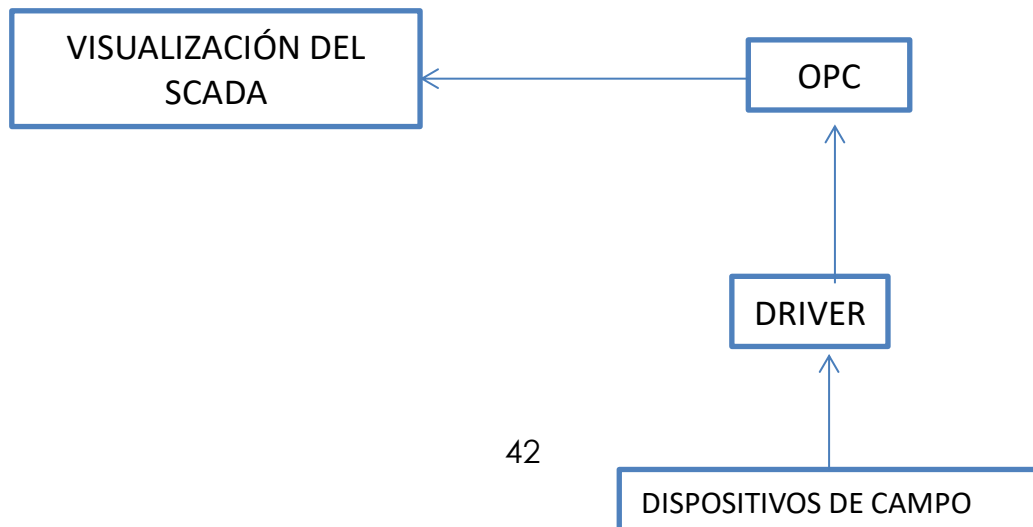


2.7.3 Interconexión general de sistema SCADA a un proceso automático.

La información que requiere un sistema SCADA debe recorrer las etapas vistas en Figura 8.

Esta información proviene de las variables del proceso que se generan mediante los sensores y actuadores, y que son controlados por un PLC o cualquier lazo de control. Estos PLC, por ejemplo, almacenan las variables a controlar en una memoria interna.

Figura 8. Esquema general de interconexión de un sistema SCADA y el proceso de control



Debido a la variedad de fabricantes de PLC, es imposible al sistema SCADA acceder a todos los modelos por sí solo. Al saber que ellos poseen arquitecturas y mapas de memoria diferentes, es necesario emplear drivers o controladores específicos.

Los Drivers [software controlador] permiten leer la memoria de los respectivos PLC de una marca y modelo específicos. Estos drivers pueden estar montados en el PC SCADA y pueden acceder a la memoria de un PLC en cualquier dirección que este en el bus de campo. Cada variable leída por el driver se le asigna un nombre. Se puede especificar la velocidad de muestreo de variables y su agrupación u organización.

Finalmente, las variables organizadas o agrupadas, etiquetadas, provenientes de diversos controladores lógicos programables son leídas por un programa especializado con herramientas y funciones específicas para procesar, visualizar, controlar, y realizar todas las acciones del sistema SCADA.

2.8 OPC – UA

El OLE para control de procesos (OPC)[3] es un estándar abierto para compartir datos entre dispositivos de campo y aplicaciones de ordenador. Permite a las aplicaciones leer y escribir valores de proceso y que los datos sean compartidos fácilmente en una red de ordenadores.

OPC Unified Architecture es la nueva generación tecnológica propuesta por la OPC Fundación (Entidad que regula las comunicaciones industriales para PLC de diferentes fabricantes, de manera que se hable un lenguaje común), El estándar OPC garantizan transmisión segura, fiable y neutra con respecto al fabricante de datos en bruto y/o información pre procesada entre los niveles de producción y los sistemas de planificación.

Con el OPC UA, toda la información deseada estará disponible para cualquier aplicación y usuario autorizado, en cualquier instante y en cualquier lugar. Esta funcionalidad será independiente del fabricante que haya desarrollado la aplicación, del lenguaje de programación seleccionado y del sistema operativo utilizado.

OPC Unified Architecture presenta dos importantes innovaciones en el mundo OPC. Por una parte, reemplaza el protocolo DCOM –específico de Windows por protocolos abiertos e independientes de la plataforma que incorporan, además, mecanismos de seguridad. Y, por otra, incorpora un modelo de información orientado a objetos que aglutina funcionalidades tradicionales de OPC como acceso a datos, históricos, alarmas y eventos e incorpora otras totalmente innovadoras como tipos de datos y métodos.

Como resultado de todo ello, no sólo se simplifica la incorporación de la interfaz OPC en cualquier sistema operativo y lenguaje de programación sino que, además se hace posible describir cualquier sistema complejo con el OPC UA.

2.9 AMBIENTES DE DESARROLLO

En el medio de la automatización industrial existen varios tipos de HMI (Human Machine Interface), para el desarrollo de la interfaz hombre – máquina se utilizó la plataforma de diseño Ignition.

El software Ignition tiene dos ambientes de trabajo conocidos como: Gateway y el Diseñador.

El primer ambiente nos permite configurar los dispositivos de campo, base de datos, las conexiones OPC, sistema de almacenamiento de Alarmas y la seguridad.

El diseñador es el ambiente necesario para la creación de la interface gráfica hombre – máquina, en él se crean ventanas, animaciones, reportes, tendencias, etc.

Esta adicionalmente incorpora un entorno de desarrollo y ejecución de Scripts utilizando el lenguaje Python, que permitió programar el driver de comunicación con el PLC SiBAS.

2.9.1 Características del Ignition

Ignition creado por Inductive Automation es un software servidor que permite crear Sistemas de Visualización, Control y Gestión que cubren todo el espectro entre HMI, SCADA y MES.

Ignition está basado en Tecnología Web. Tanto el acceso para Desarrollo como para Runtime se realizan a través de la Web empleando el innovador sistema Java Web Launch.

Los datos se almacenan siempre en un formato abierto y accesible. Posee un historizador de datos compatible con cualquier Base de Datos SQL. Ignition incluye comunicación nativa para almacenar datos en línea a Microsoft SQL Server, MySQL, Oracle, IBM DB2, Firebird, y PostgreSQL entre otros.

Las funcionalidades de Visualización y de creación de reportes pueden utilizar datos de múltiples Bases de Datos simultáneamente.

Ignition ofrece un servidor OPC-UA con posibilidad de comunicarse con diversos protocolos. Además integra un Cliente OPC-DA y OPC-UA que permite conectarse a cualquier Servidor OPC para recoger datos de campo. El Cliente OPC de Ignition asegura un alto rendimiento en la recolección de datos.

2.9.2 Módulos

La plataforma Ignition tiene una estructura modular. Se puede quitar y añadir módulos al instante con un mínimo de paro del sistema. En dependencia de los módulos instalados Ignition sirve para muchas funcionalidades [15].

- **MODULO OPC-UA:** Es un servidor OPC-UA programado 100% en Java con una interface abierta para añadir controladores.
- **Módulo SQL Bridge:** Realiza la conexión entre datos OPC y las bases de datos configuradas.
- **MÓDULO DE REPORTE:** El módulo crea reportes PDF dinámicos con un diseño exacto.
- **MÓDULO VISION:** Entorno Grafico de diseño de Pantallas para el sistema SCADA.

- **MÓDULO MOBILE:** Permite el acceso móvil a su sistema de control a través de un Smart Phone y/u otros teléfonos inteligentes tipo tableta. El módulo mobile da acceso instantáneo a cualquier proyecto HMI / SCADA creado con el Módulo Visión.

2.9.3 Lenguaje de Programación Python.

Python[16] es un lenguaje de programación dinámico y orientado a objetos que puede ser usado de muchas maneras en el desarrollo de software. Ofrece gran soporte e integración con otros lenguajes y herramientas, viene con una extensiva cantidad de librerías.

Python se puede ejecutar en Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, y teléfonos celulares Nokia. Python también ha sido portado para las máquinas virtuales de Java y .NET.

Python es distribuido bajo la licencia open source OSI que lo hace libre para ser usado inclusive en el desarrollo de productos comerciales.

2.10 CONSIDERACIONES PARA EL DISEÑO DE HMI

EL diseño del HMI juega un papel importante en procesos industriales porque determinar la eficiencia de la comunicación del usuario con las máquinas. La interfaz se puede convertir en una barrera debido a un pobre diseño y una escasa atención a los detalles de los objetivos del sistema, por lo tanto es necesario conocer las pautas de diseño, normas y recomendaciones para la elaboración de un HMI apropiado[14]:

De lo anterior, se considera que los colores hacen parte de los aspectos más importantes a la hora de diseñar una HMI ya que estos poseen un significado, el cual indica al operador cómo se está comportando el proceso y en muchas ocasiones sirve como aviso para evitar cualquier emergencia, por lo que en la Tabla 3, se muestra un plan de colores [14] que es recomendable a seguir para lograr tal objetivo.

TABLA 3. PLAN DE COLORES

PLAN DE COLORES		
Color	Significado General	Acción
Gris	Fondo	

Blanco		
Rojo	Emergencia	A) Parada B) Alarma de mayor prioridad C) Cerrado D) Apagado
Amarillo	Precaución	A) Condición Anormal B) Segunda alarma de mayor prioridad
Verde	Seguridad	A) Operación Normal B) Inicio C) Abierto D) Encendido
Negro	Datos Dinámicos	A) Medidas y Estado de la información B) Mensajes del sistema C) Tendencias D) Paso activos secuenciales

Además de los colores, las figuras o formas que se empleen en el HMI deben ir muy relacionadas con el proceso. Si la HMI se encuentra bien diseñada, el usuario final debe poder familiarizarse con ella rápidamente y facilitar su operación y controlar el proceso, para llegar a este resultado se recomienda seguir las siguientes recomendaciones [14]:

- Demasiada simetría puede hacer las pantallas difíciles de leer.
- Elementos de tamaño y color similares se perciben como pertenecientes a un grupo.
- Asumir errores en la entrada del usuario.
- Los gráficos de fondo deben usar tonos suaves, deben ser evitados los colores de alarma. Utilizar el negro para resaltar objetos.
- Texto: utilizar un tipo de letra estándar, que esté disponible en todos los PC.
- El estado del sistema de alarma debe ser visible en todas las pantallas.

- Los botones de navegación deben ser obvios y lo suficientemente grandes.
- Todas las páginas deben tener una visualización consistente. Los botones de navegación deben estar en el mismo lugar.
- Presentar al usuario sólo la información que necesita.
- Misma respuesta ante la misma acción. Los elementos estándar de la interface deben comportarse siempre de la misma forma.
- Consistencia de la apariencia estética (iconos, fuentes, colores, distribución de pantallas).

3. DISEÑO DE LA PLATAFORMA SCADA PARA MONITOREAR LAS PALAS ELÉCTRICAS 41, 42, 43 Y 44 EN LA COMPAÑÍA DRUMMOND LTD.

La figura 9 muestra la Pala eléctrica dedicada exclusivamente a la extracción de material estéril y carbón en Mina Pribbenow de la multinacional Drummond.

Dada la ubicación de las palas en puntos distantes de la mina, optimizar el control del estado de las máquinas en tiempo real y disminuir el tiempo Down con este objetivo se propone el diseño de la plataforma SCADA.

Figura 9. Pala Bucyrus



3.1 DEFINICIÓN DE REQUISITOS

Basado en las entrevistas mantenidas con los usuarios se pudieron identificar los requisitos del sistema.

Se debe crear un sistema SCADA que permita monitorear las palas Bucyrus identificadas como PALA 41, PALA 42, PALA 43 Y PALA 44. Estas

máquinas se encargan de remover material en las minas de carbón de Drummond ubicadas en el departamento del Cesar.

Las palas Bucyrus son controladas de manera automática por un PLC Siemens con nueve nodos remotos compuestos por módulos de entradas salidas (I/O) y un controlador de movimientos llamado SiBAS. Adicionalmente cuenta con elementos de protección compuesto por una red inteligente Contra incendios, y un medidor de energía y protección a sobre voltajes de la General Electric llamado PQM (Power Quality Meter). Toda esta información se debe visualizar en el sistema SCADA en tiempo real, utilizando mímicos que representan de manera visual el comportamiento y estado de las palas, sirviendo de interfaz hombre máquina. Es importante destacar que se desea monitorear información de dispositivos de campos de diferentes fabricantes y se necesita una plataforma de trabajo que permita integrar los distintos tipos de tecnología utilizadas en las palas eléctricas.

El sistema SCADA debe reportar las alarmas activas, histórico de alarmas con descripción detallada de cada falla generada (complementada con un archivo pdf) y otros reportes como listado de falla, tiempo Down entre otros.

En cuanto al PLC SiBAS (Siemens BUS Automation System) es necesario crear una aplicación que permita una comunicación con él y acceder a sus datos, considerando que el driver que se conecta al SiBAS, únicamente se ejecuta en el sistema operativo Windows.

3.2 ESPECIFICACIÓN DE REQUISITOS

El sistema SCADA visualiza interfaces gráficas de supervisión para observar la evolución de las variables del proceso en tiempo real.

El sistema SCADA debe disponer del conjunto de drivers necesario para recoger, procesar y almacenar la información recibida de los diversos dispositivos de campos.

El sistema debe almacenar datos históricos para hacer análisis de tendencias, generar reportes y buscar fallas.

La interfaz de Visualización debe contar con diferentes niveles que muestren el detalle de la máquina.

El sistema despliega un panel de alarmas activas para indicar una situación anormal.

3.3 DIAGRAMA UML

A continuación se describe la modelización de dichas funcionalidades. Los diagramas utilizados están basados en el Lenguaje Unificado de modelado UML[18].

3.3.1 Diagrama de casos de usos

3.3.1.1 Identificación de Actores

Los actores del sistema son los que se describen a continuación:

Usuario: Son todos los empleados autorizados que necesitan tener acceso al sistema SCADA como parte fundamental de su labor dentro de la empresa.

Operador: Es la persona encargada de monitorear las palas eléctricas.

Supervisor: Es la persona encarga de monitorear y tomar decisiones acerca de las máquinas.

PLC: Es el dispositivos de campo con la cual el sistema interactúa para recibir datos.

A continuación se describirán los casos de usos considerados para el proyecto:

3.3.1.2 Casos de Uso del Sistema SCADA

Figura 10. Casos de Uso del Sistema SCADA

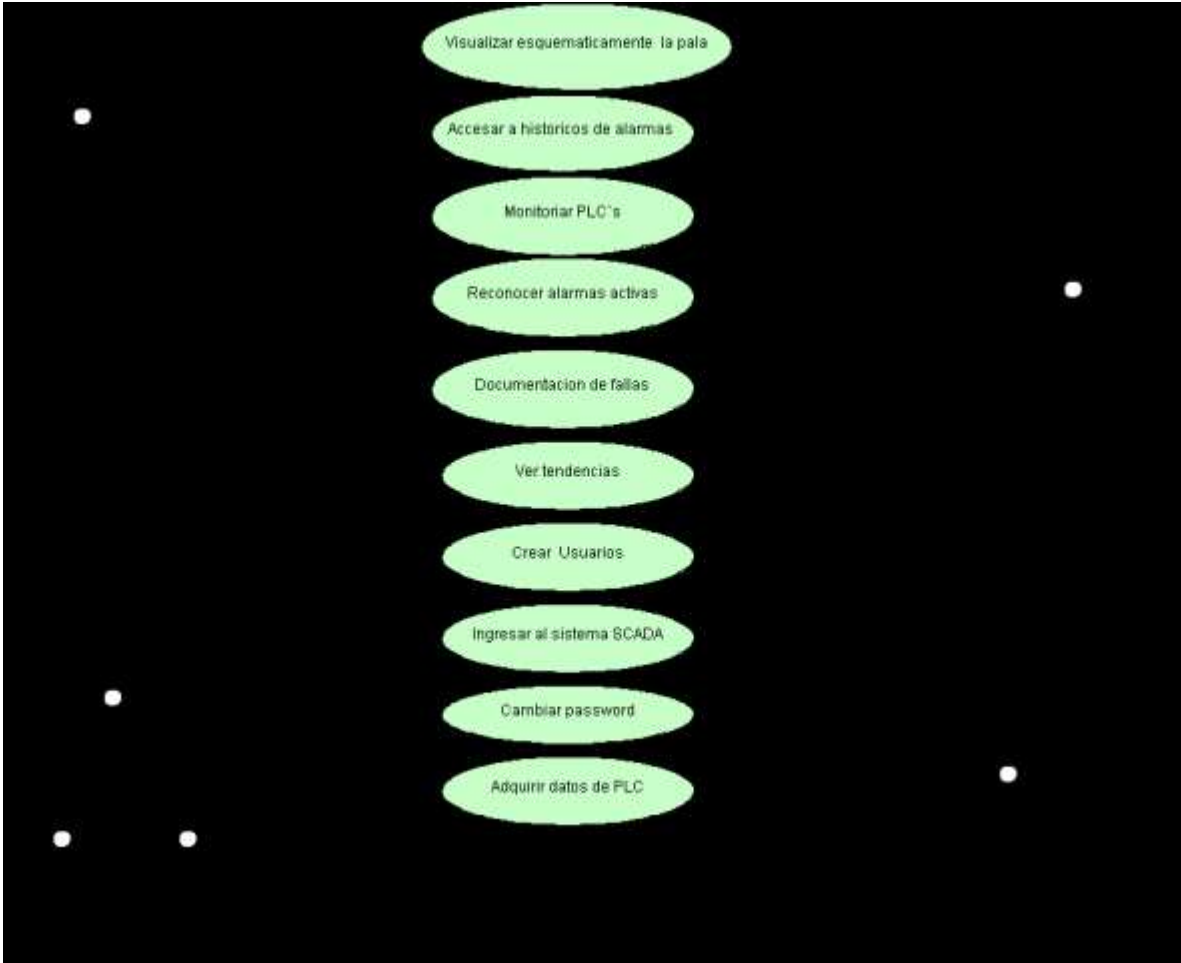


Tabla 4. Reconocer alarmas activas.

Nombre:	Reconocer alarmas activas.
Actores:	Operador
Descripción:	Las alarmas son notificadas en pantallas y pueden ser notificadas mediante correo electrónico de acuerdo al tipo de Falla.

Tabla 5. Accesar a Históricos de Alarmas

Nombre:	Accesar a Históricos de Alarmas
Actores:	Supervisor
Descripción:	Proporcionar a los usuarios el acceso al estadístico de Fallas filtrando de acuerdo a rango de fechas, tipo de falla o nombres de fallas especifica.

Tabla 6. Ver Tendencias

Nombre:	Ver tendencias
Actores:	Supervisor
Descripción:	Mostrar gráficas con relación a variables del proceso.

Tabla 7. Visualizar animación en tiempo real de las palas eléctricas.

Nombre:	Visualizar animación en tiempo real de las palas eléctricas.
Actores:	Operador
Descripción:	Proporcionar a los usuarios una vista general de las condiciones operativas de la máquina.

Tabla 8. Visualizar vista de los subsistemas de la pala.

Nombre:	Visualizar vista de los subsistemas de la pala.
Actores:	Supervisor
Descripción:	Proporcionar a los usuarios una vista de los subsistemas que conforman la pala.

Tabla 9. Visualizar diagrama Unifilar

Nombre:	Visualizar diagrama Unifilar.
Actores:	Supervisor
Descripción:	Proporcionar una vista grafica de la instalación eléctrica de la pala representada mediante una línea que indica un conjunto de conductores de un circuito.

Tabla 10. Visualizar estados de la Red de PLC

Nombre:	Visualizar estados de la Red de PLC
Actores:	Supervisor
Descripción:	Muestra una vista de la red de PLC

Tabla 11. Adquirir Datos

Nombre:	Adquirir datos
Actores:	PLC
Descripción:	Leer datos de los PLC' s

Tabla 12. Ingresar al Sistema SCADA

Nombre:	Ingresar al Sistema SCADA
Actores:	Operador – Supervisor.
Descripción:	Dar acceso a los usuarios del sistema mediante nombre de usuario y contraseña asignados previamente.

Tabla 13. Ver documentación de Fallas

Nombre:	Ver documentación de Fallas
Actores:	Operador - Supervisor
Descripción:	Permite obtener el catálogo de fallas.

Tabla 14. Crear Usuarios

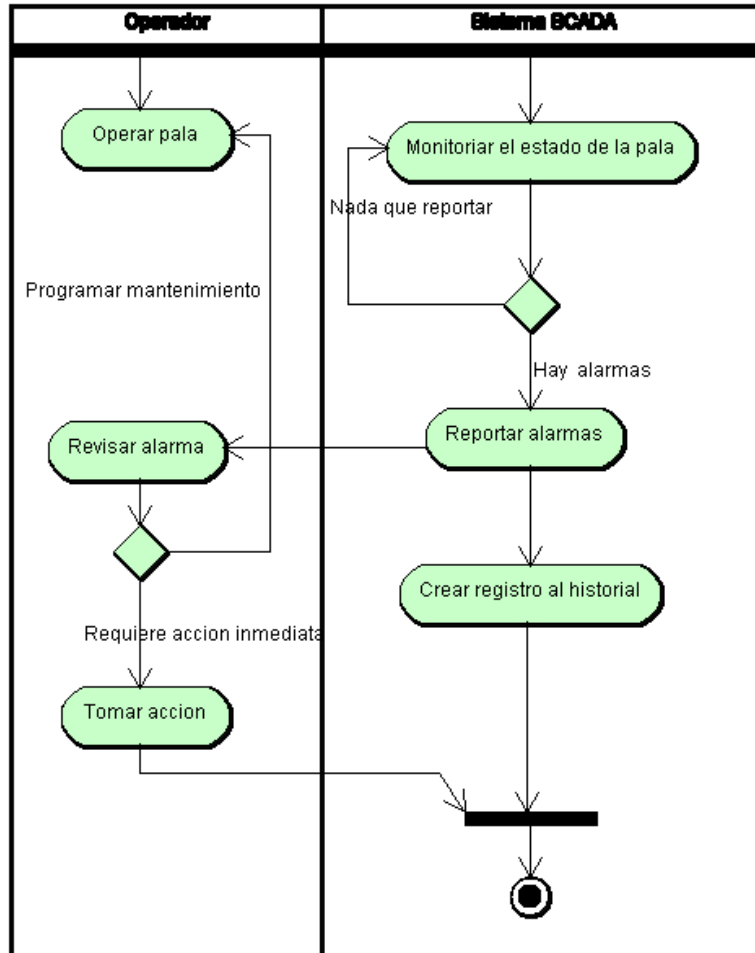
Nombre:	Crear Usuarios
Actores:	Supervisor
Descripción:	Ingresar la información de un usuario al sistema para que pueda tener acceso al mismo

Tabla 15. Cambiar Password

Nombre:	Cambiar Password
Actores:	Operador - Supervisor
Descripción:	Cambiar la clave de ingreso de un usuario al sistema.

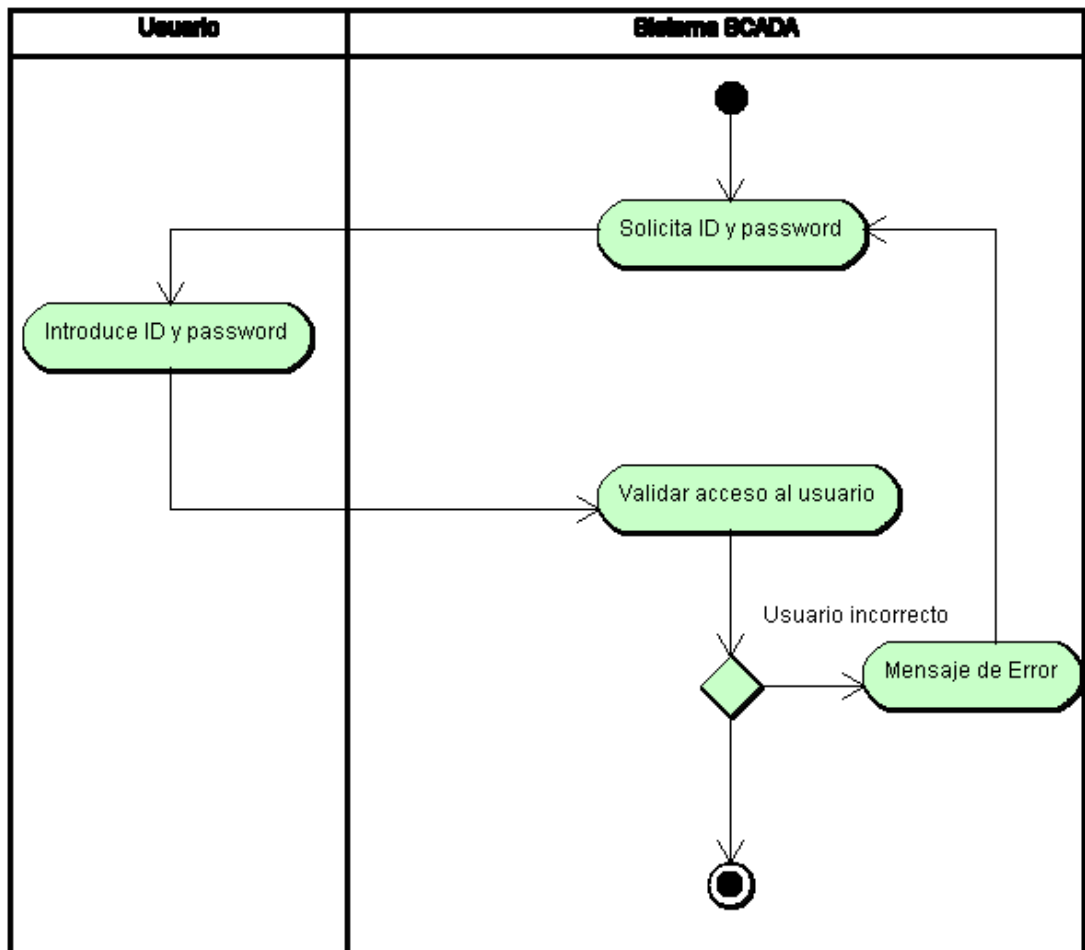
3.3.2 Diagrama de Actividad

Figura 11. Caso de Uso: Reconocer Alarmas Activas



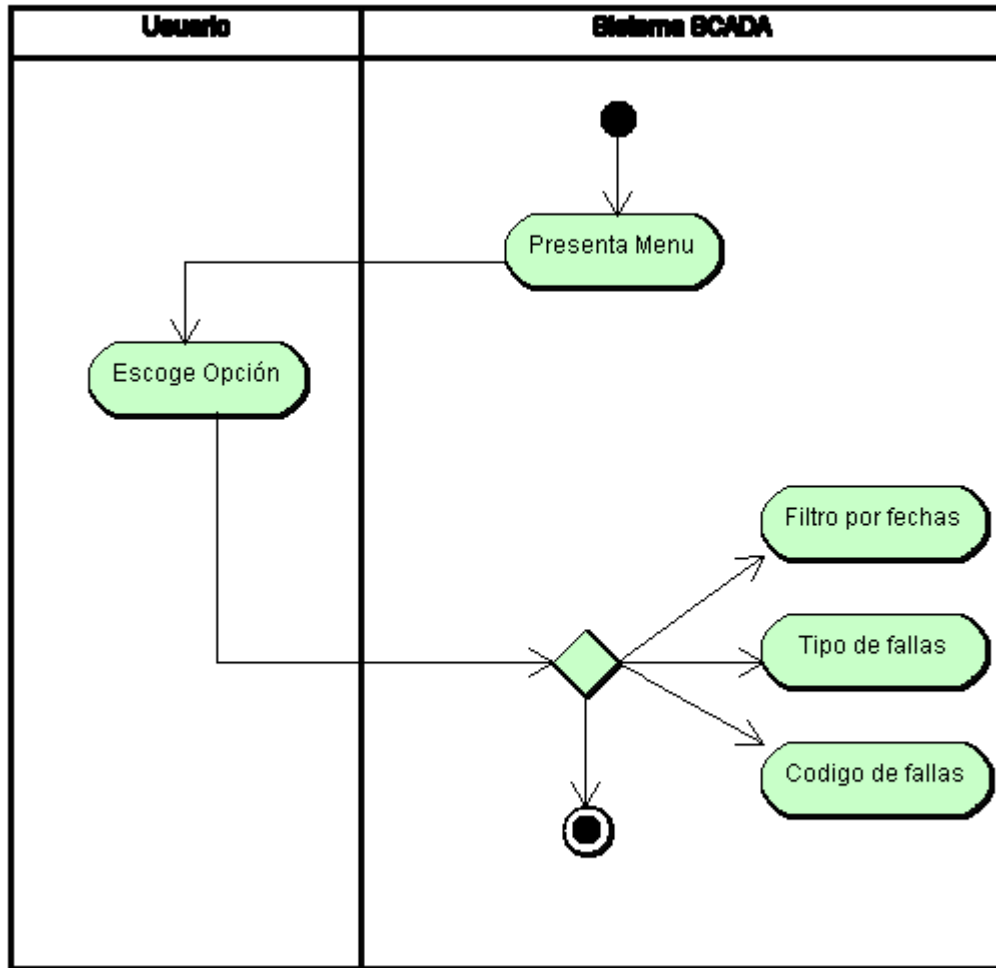
Una vez que el sistema realiza un monitoreo del estado de las palas, dispara una alarma cuando ocurre una falla y realiza un reporte automático en la pantalla de Alarmas Activas y a su vez un acceso a la base de datos para almacenarla en el histórico. El operador ingresa a esta pantalla mediante una label que indica activación de la alarma y determinara la acción a tomar.

Figura 12. Caso de Uso: Ingresar al sistema SCADA



El sistema presenta una interfaz para validar el acceso. El Usuario deberá introducir su nombre de usuario y contraseña, el sistema SCADA verificará los datos accediendo a la base de datos del sistema. Si el usuario no se encuentra registrado o existen un error en la validación saldrá un mensaje de error y se mostrará nuevamente el formulario de acceso al sistema.

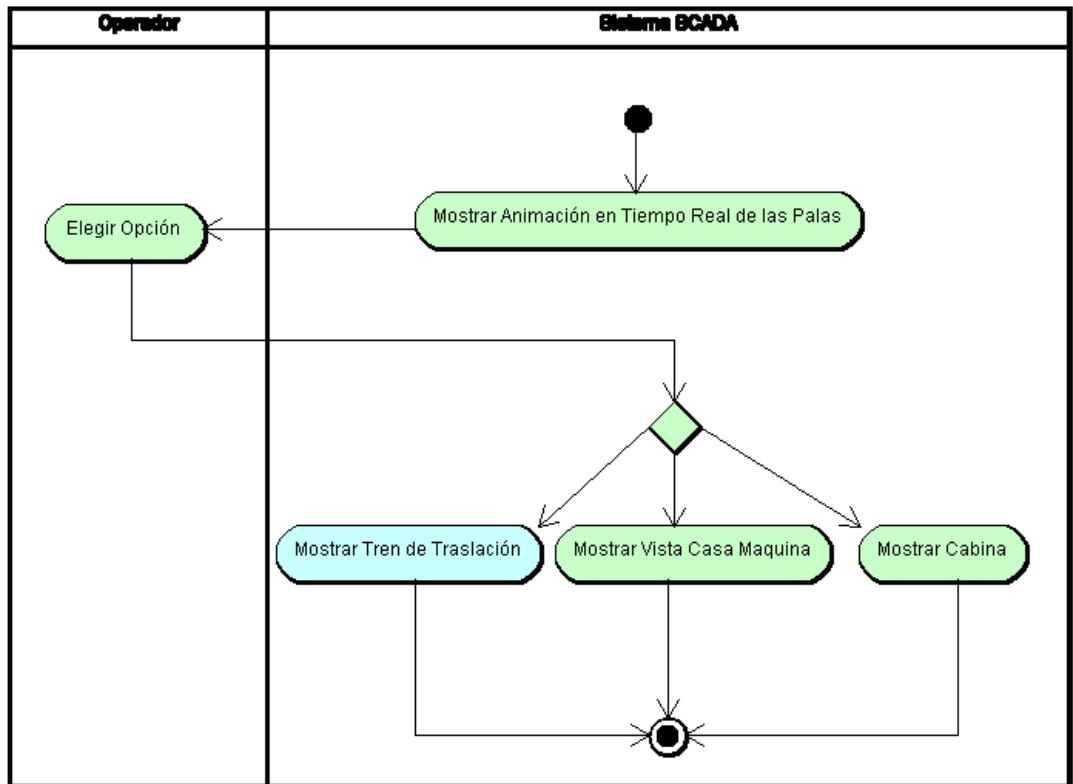
Figura 13. Caso de Uso: Accesar a Históricos de Alarmas



Una vez que el usuario se valida, el sistema SCADA le presenta una interfaz en la que puede escoger entre distintas opciones, al seleccionar Acceso a Históricos de Alarmas tiene la posibilidad de filtrar la información de acuerdo:

- Introducir código de Alarma
- Rango de Fecha
- Tipo de Alarma

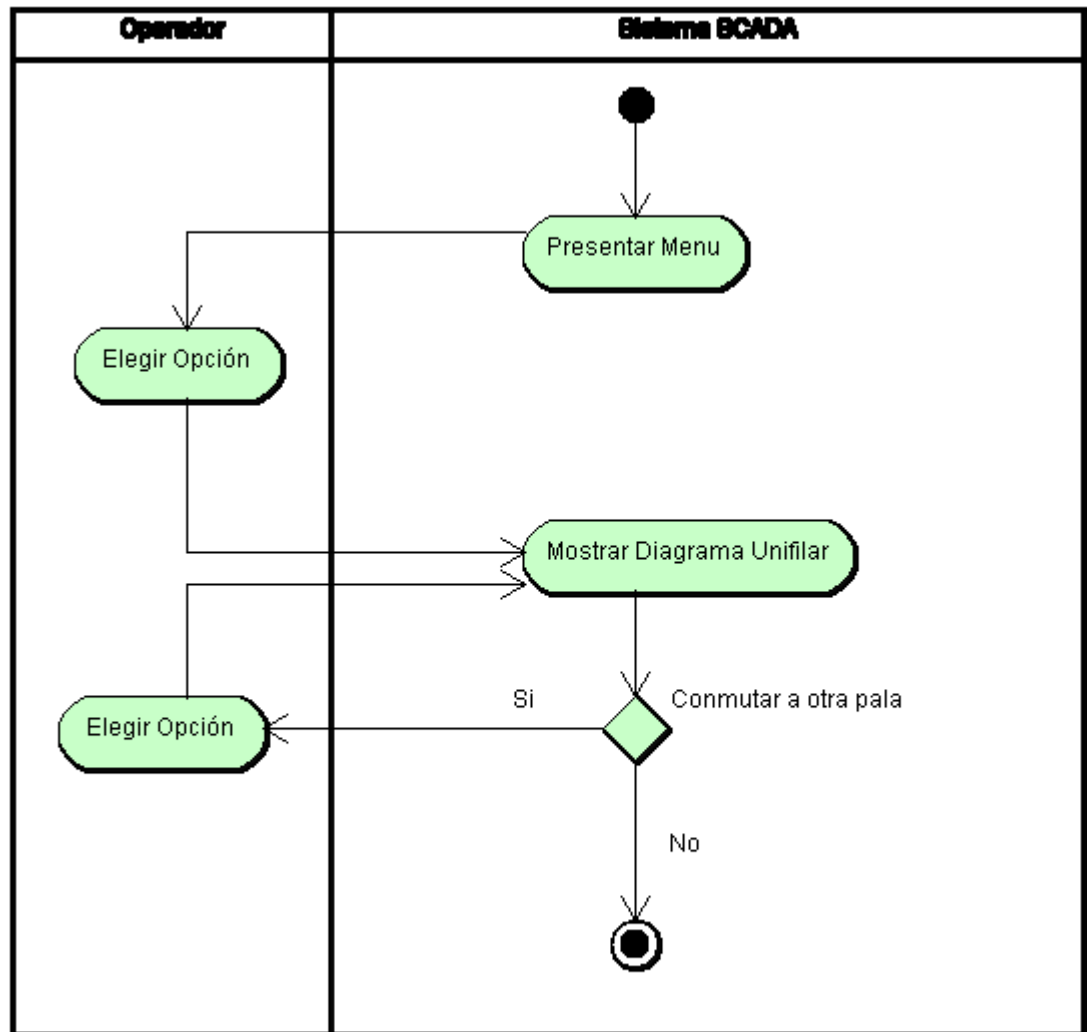
Figuras 14. Caso de Uso: Visualizar Vista de los subsistemas de las Palas



El sistema SCADA en su pantalla principal visualiza la pala mostrando sus movimientos en tiempo real y a su vez proporciona el acceso a los distintos subsistemas que conforman la pala:

- Tren de Traslación
- Casa Maquina
- Cabina

Figura 15. Caso de Uso: Visualizar Diagrama Unifilar de la Pala.



El sistema proporciona una interfaz que le permite seleccionar el diagrama Unifilar de cualquiera de las pala y a su vez ofrece conmutar éntrelos diagramas de las palas 41, 42, 43 y 44.

Figura 16. Caso de Uso: Visualizar estados del PLC

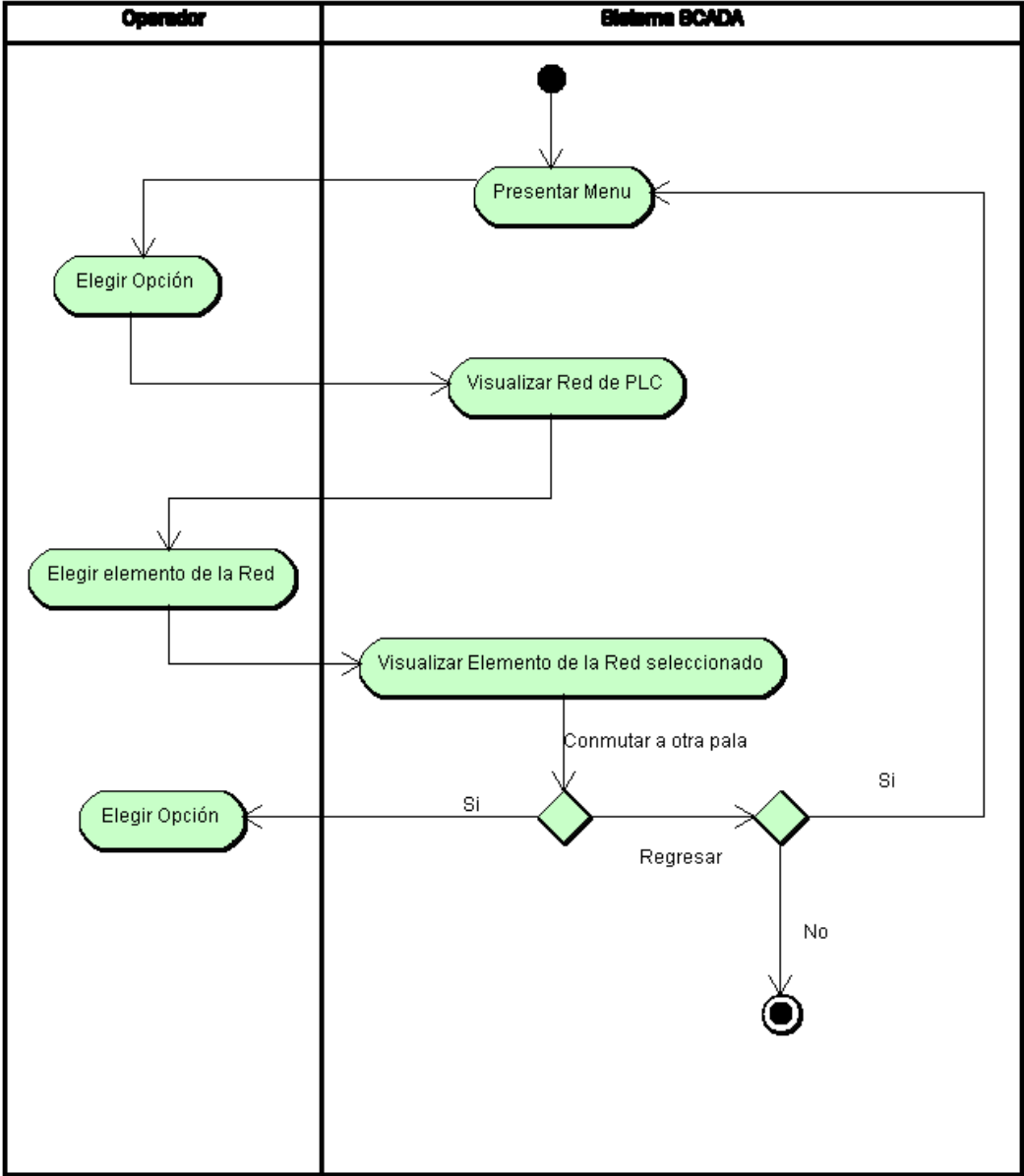
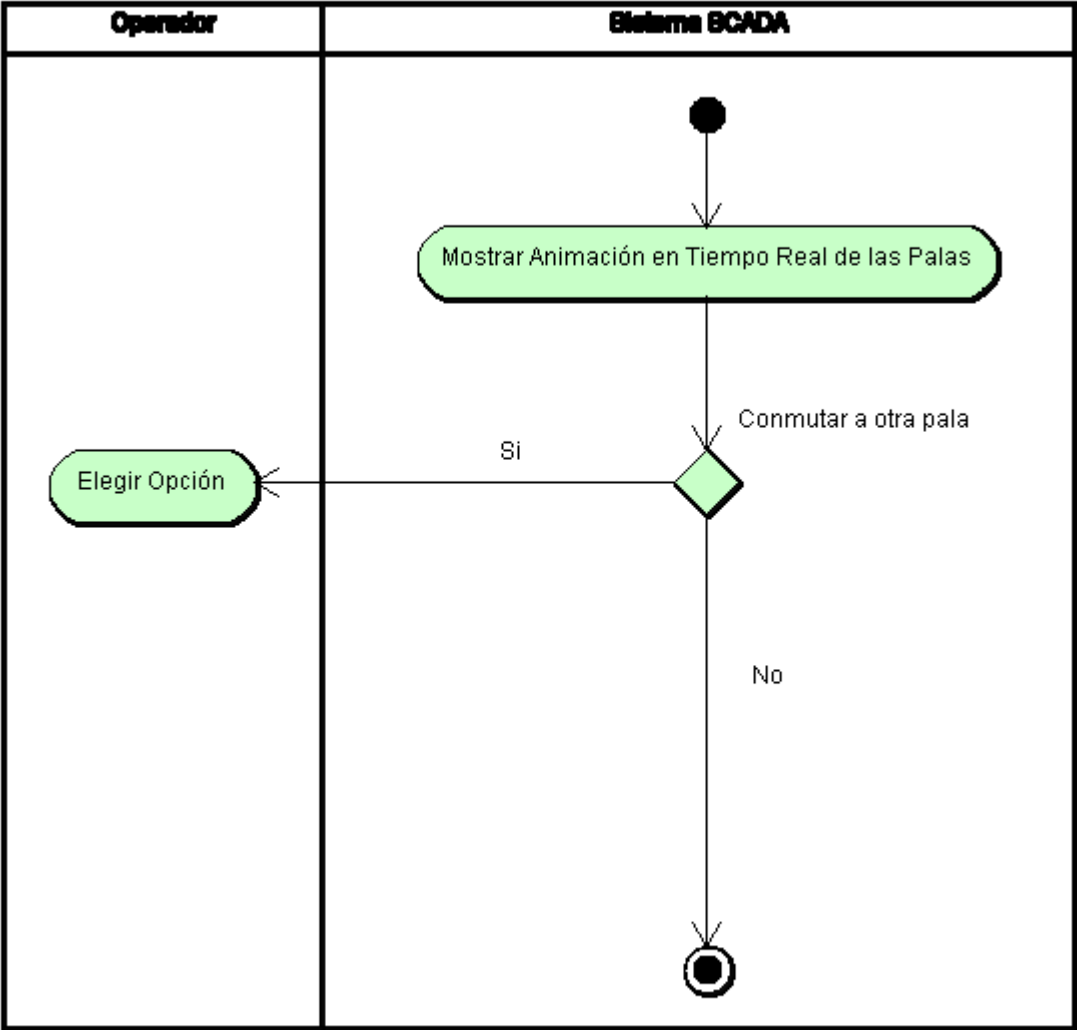


Figura 17. Caso de Uso: Visualizar Animación en tiempo real de la Pala



El sistema muestra en la pantalla principal las variables operativa de la Pala seleccionada y ofrece la opción de conmutar a las demás palas ofreciendo la vista actual.

Figura 18. Caso de Uso: Ver Tendencias

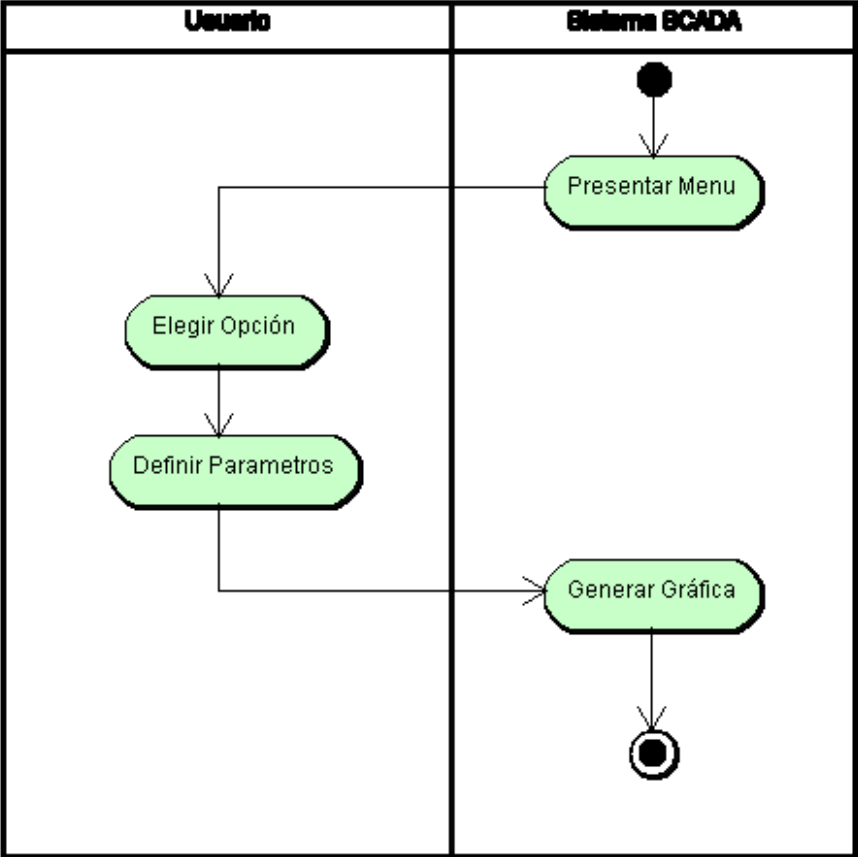


Figura 19. Caso de Uso: Ver documentación de fallas.

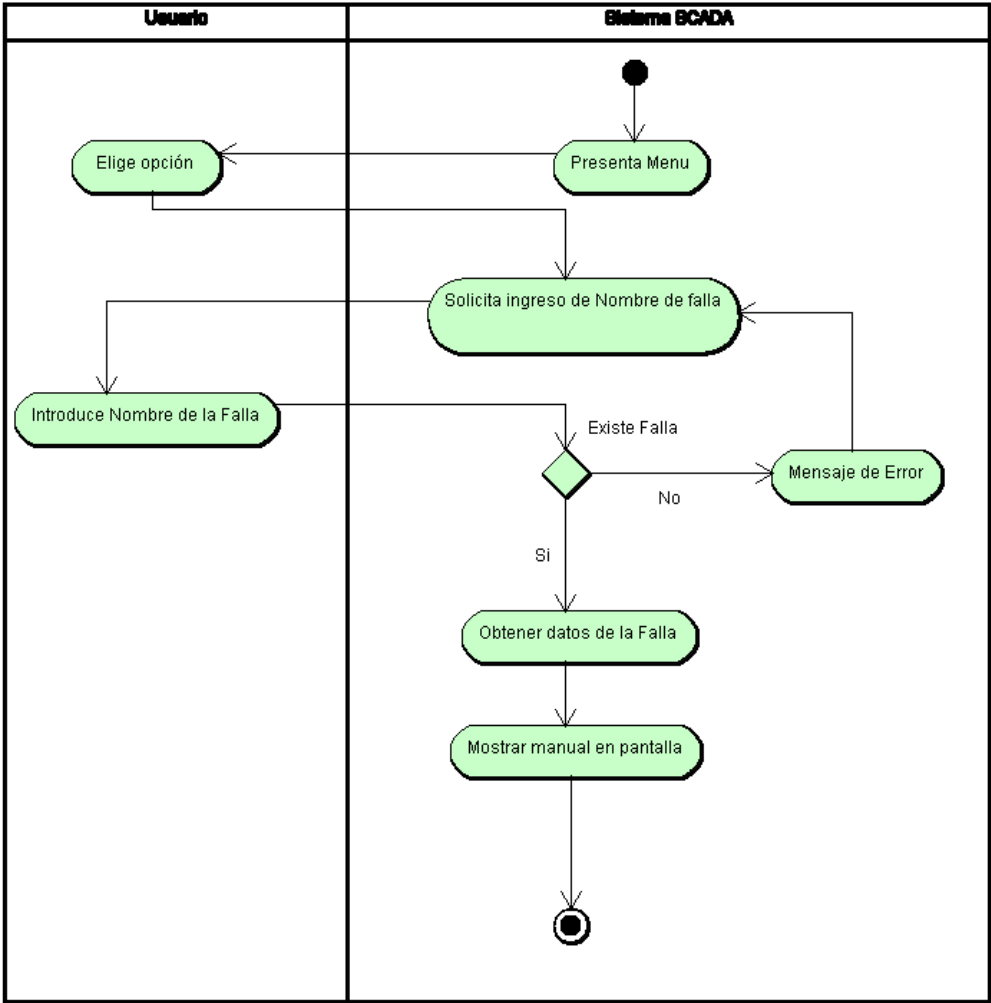
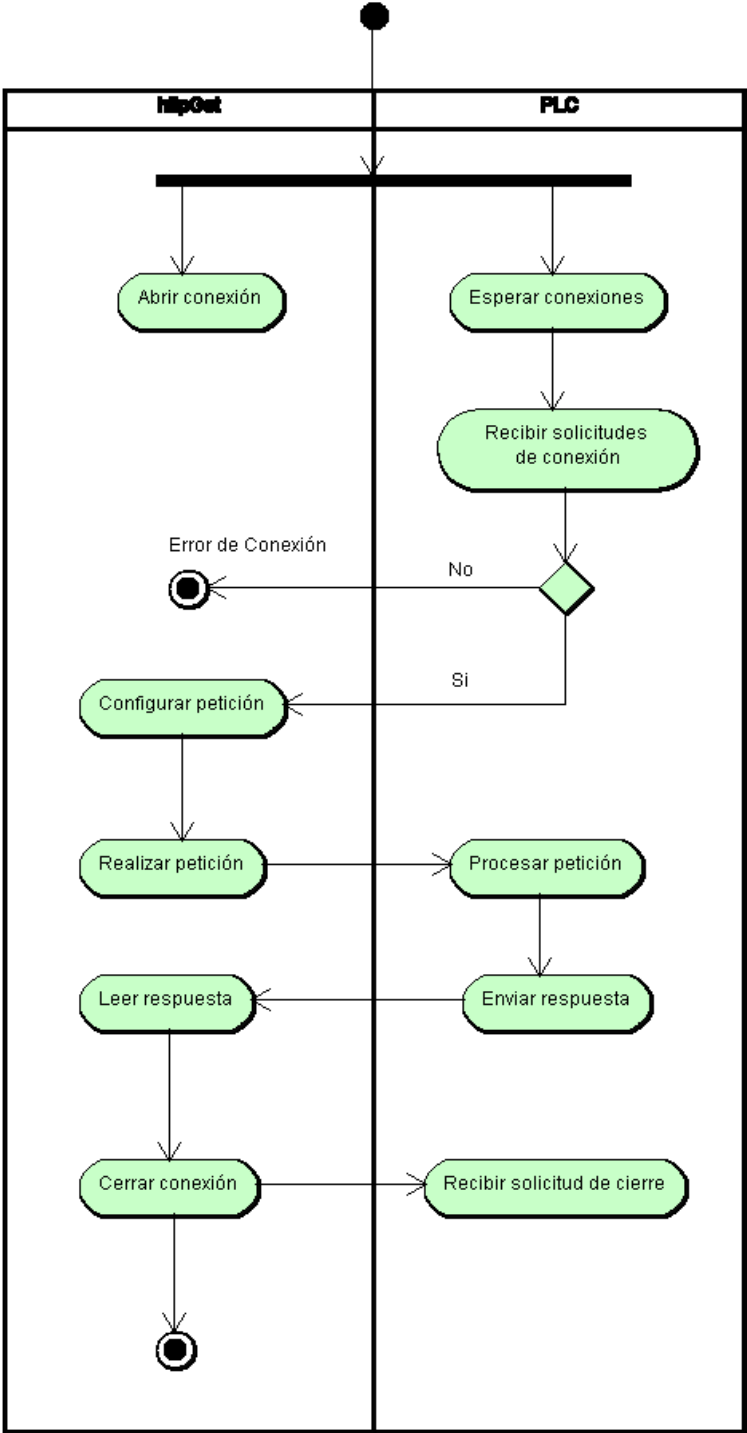


Figura 20. Caso de Uso: Adquirir datos del PLC



Este caso abre una conexión con el PLC SiBAS utilizando la dirección IP asignada. Una vez establecida la conexión, se configura la petición y se envía la petición HTTP GET, leemos un recurso de la URL, el PLC SiBAS devuelve la respuesta (datos) Y finalmente se cierra la conexión.

Figura 21. Caso de Uso: Crear usuarios

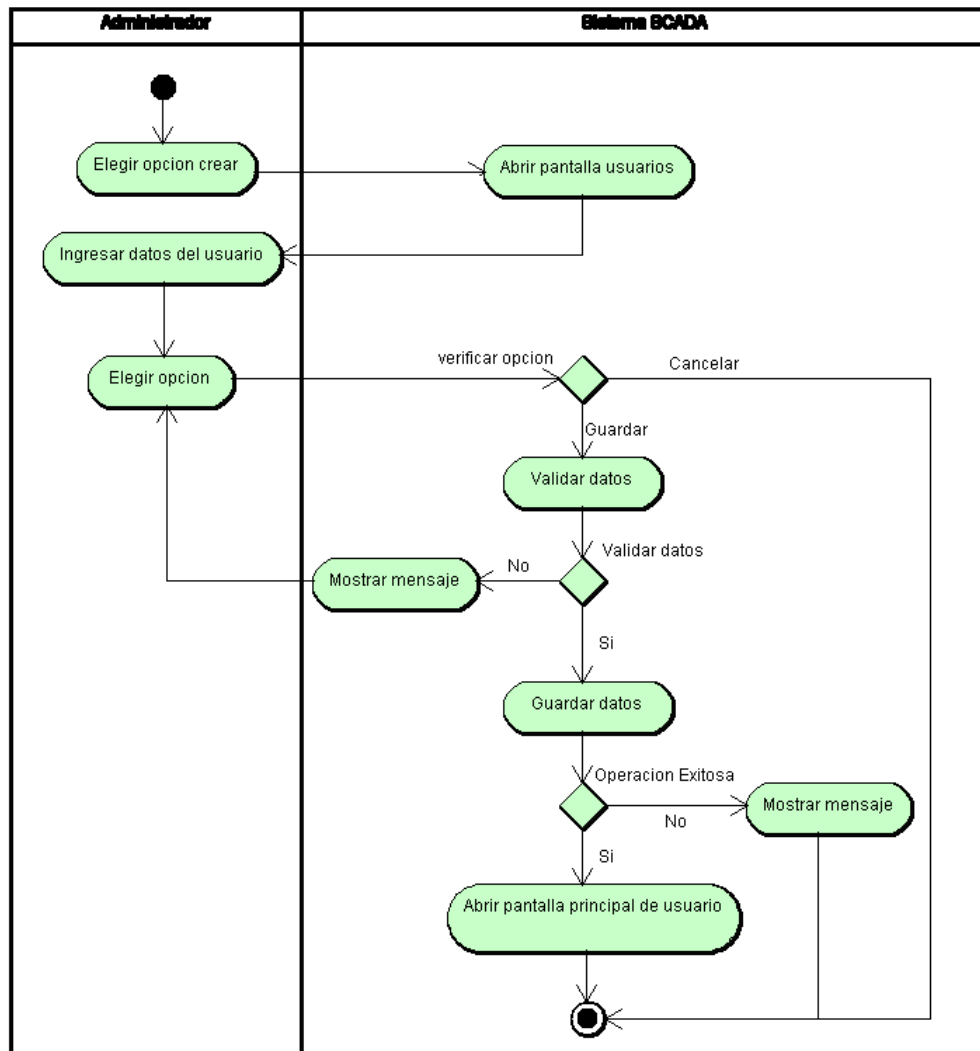
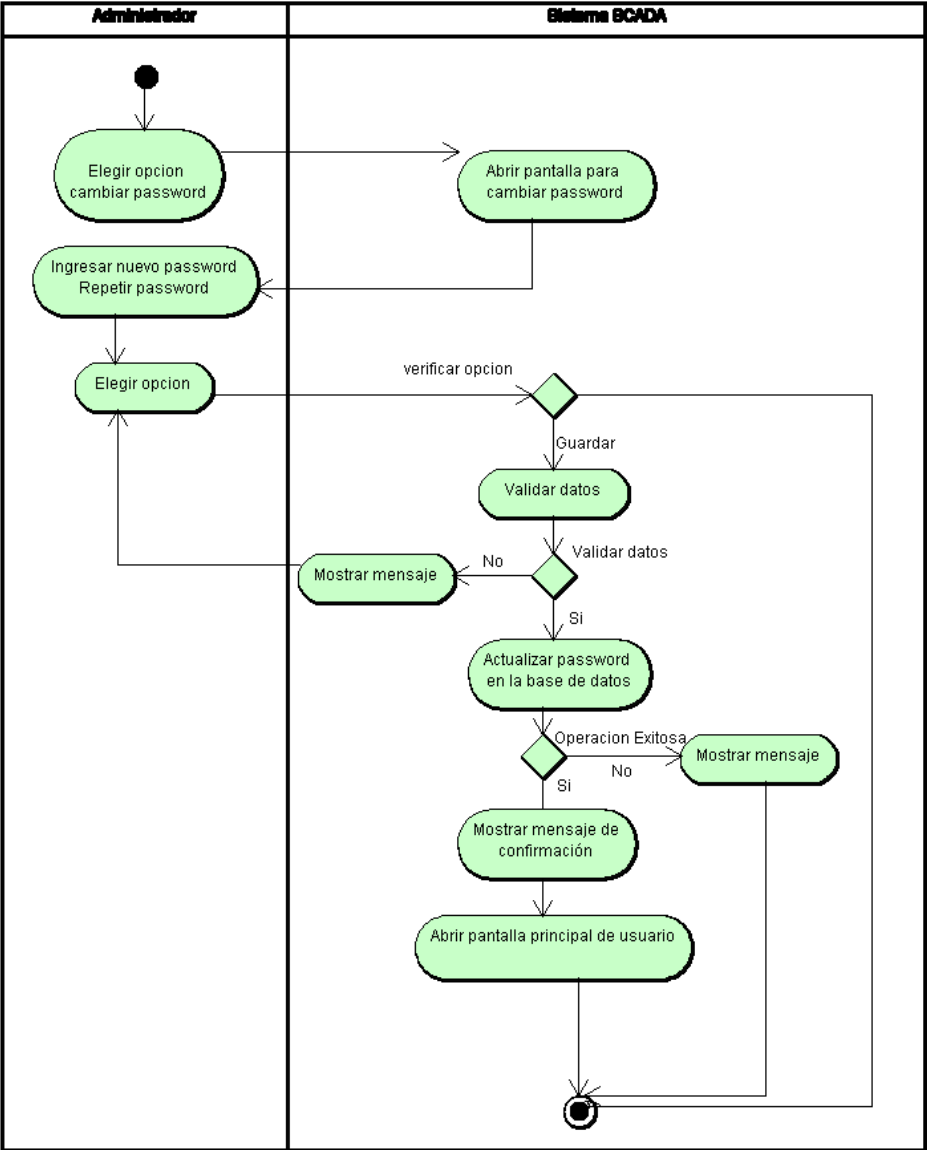
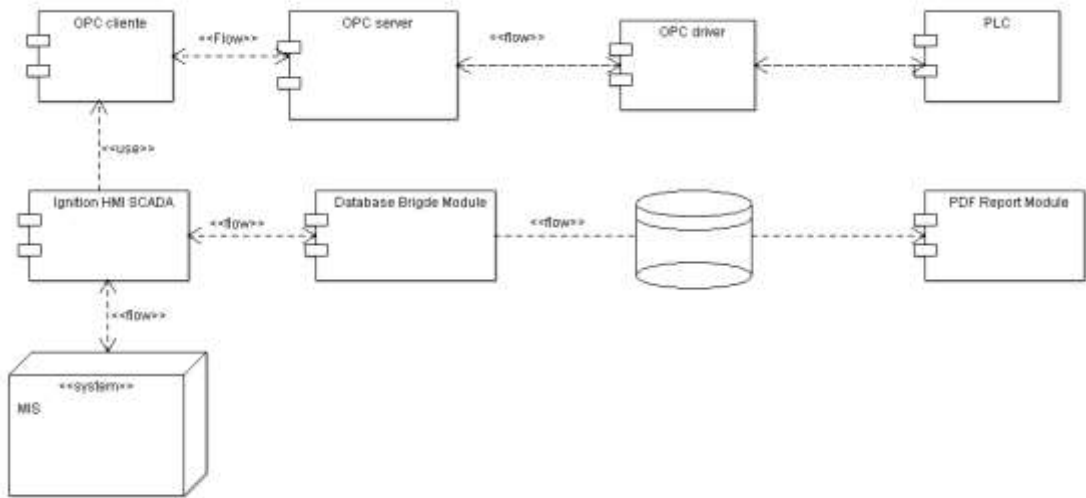


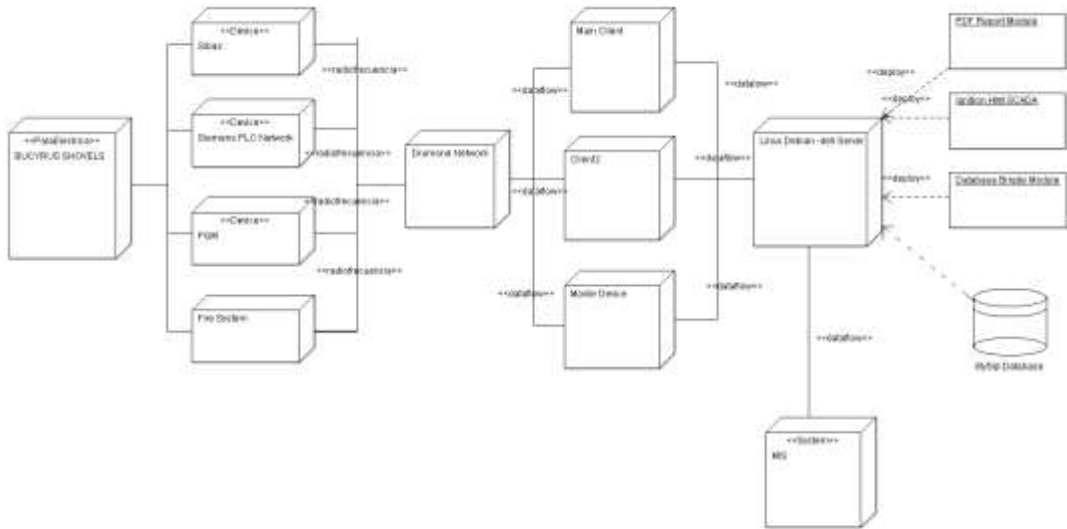
Figura 22. Caso de Uso: Cambiar password



3.3.3 Diagrama de Componentes



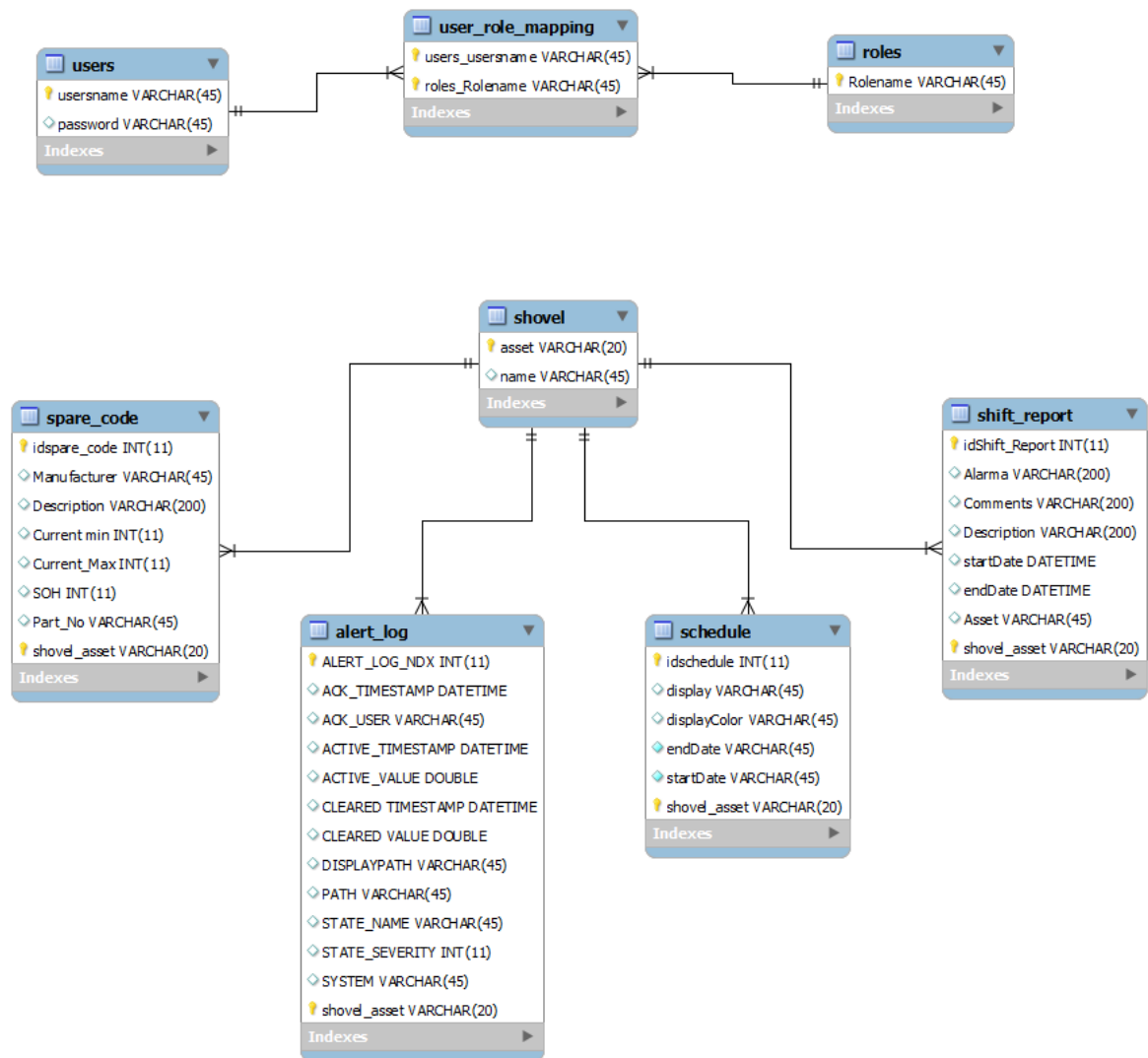
3.3.4 Diagrama de Despliegue



3.4 MODELO ENTIDAD RELACIÓN

En la figura 23 se muestra el diagrama entidad relación del sistema SCADA, como se observa las tablas users, roles y user_role_mapping forman parte del modelo de seguridad que es implementado en la aplicación.

Figura 23. Modelo Entidad Relación



3.4.1 Tablas

A continuación se describen las tablas mostradas en la figura 22:

Tabla de Roles

Esta tabla almacena los roles que son manejados en la aplicación, el nombre de los roles es único.

Tabla 16. Roles

Campo	Tipo de Dato
! Rolename	Varchar(45)

Tabla Users

Está tabla almacena la información relacionada con la creación de los usuarios en la aplicación.

Tabla 17. Users

Campo	Tipo de dato
! Username	Varchar(45)
Password	Varchar(45)

Tabla User Roles_Mapping

Esta tabla se utiliza para hacer la relación de roles y usuarios

Tabla 18. User_Roles_Mapping



Campo	Tipo de dato
 Username	Varchar(45)
 Rolename	Varchar(45)

Tabla Shovel

Esta tabla almacena la identificación de cada pala

Tabla 19. Shovel


Campo	Tipo de dato
 Asset	Varchar(20)
Name	Varchar(45)

Tabla Alert_Log

Esta tabla almacena las alarmas generadas por la pala

Tabla 20. Alert_Log



Campo	Tipo de Dato
 Alert_Log_NDX	Int
Ack_Timestamp	Datetime
Ack_User	Varchar(45)
Active_timestamp	Datetime
Activevalue	Double
Cleared_timestamp	Datetime
Cleared_Value	Double
Displaypath	Varchar(45)
Path	Varchar(45)
State_Name	Varchar(45)
State_Severity	int
System	Varchar(45)
 Asset	Varchar(20)

Tabla ShiftReport

Almacena información relacionada con fallas que generan tiempo de paro de las máquinas.

Tabla 21. ShiftReport



Campo	Tipo de dato
 IdShifReport	int
Alarm	Varchar(200)
Comments	Varchar(200)
Description	Varchar(200)
startDate	Datetime
endDate	Datetime
 Asset	Varchar(20)

Tabla Schedule

Almacena información relacionada con las actividades de mantenimiento asociada al equipo.

Tabla 22. Schedule



Campo	Tipo de Dato
 idschedule	int
Display	Varchar(45)
displayColor	Varchar(20)
startDate	Datetime
endDate	Datetime
 Asset	Varchar(20)

Tabla Spare_code

Esta tabla almacena un listado de las cantidades repuestos críticos que se encuentran en bodega.

Tabla 23. Spare_Code

Campo	Tipo de Dato
Idspare_code	int
Manufacturer	Varchar(45)
Description	Varchar(200)
Current_Min	Int
Current_Max	int
SOH	int
Part_No	Varchar(45)
Asset	Varchar(20)

3.5 DISEÑO DE LA INTERFAZ DE USUARIO

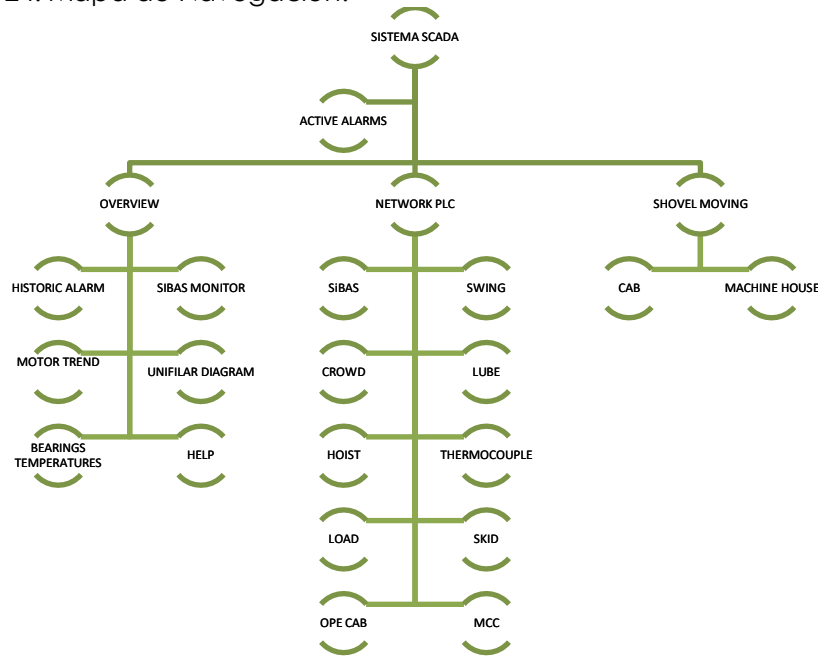
La función del HMI (Interfaz Hombre-Máquina) es proporcionar una vista del proceso mediante gráficos los cuales representan cada uno de los dispositivos y subsistemas del mismo.

En la elaboración de las pantallas de visualización se tuvieron consideraciones en colores, símbolos y formas los cuales son importantes para reflejar un diseño parecido a la máquina.

3.5.1 Mapa de navegación del sistema SCADA

A continuación en la figura 23 se presenta el mapa de navegación de la interfaz del usuario a lo largo del sistema SCADA; por medio de las opciones de menú que presentará la aplicación se logrará la funcionalidad descrita para el sistema.

Figura 24. Mapa de Navegación.



3.5.2 Descripción de las pantallas gráficas

Todas las pantallas del sistema SCADA mostrarán en la parte superior una ventana de navegación que proporciona un componente Ficha para los comandos más frecuentes permitiendo simplificar el acceso desde cualquier ventana.

Esta ventana de navegación como se muestra en la figura 25 visualiza los indicadores de alarmas suministrando el número de alarmas activas de cada pala y que a su vez permite el acceso a la ventana de visualización de las mismas.

Los botones de Shovel 41 hasta Shovel 44 permiten conmutar los datos de cada pala.

Figura 25. Ventana de Navegación



La pantalla principal muestra la animación de la pala en tiempo real con movimientos que representan el estado actual de la misma (ver Figura 26).

Figura 26. Ventana Principal



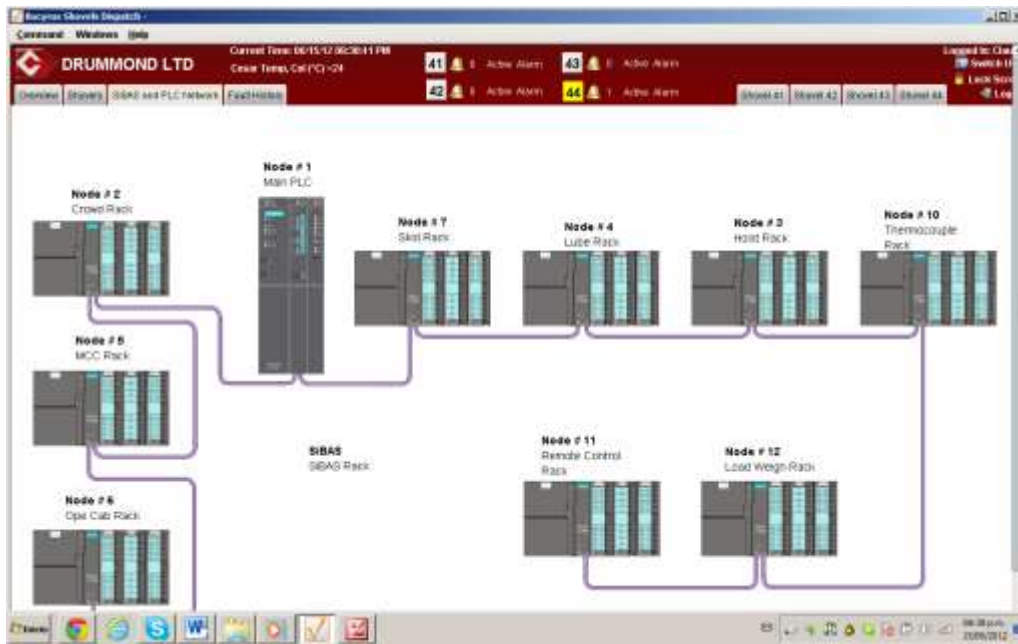
PANTALLA DE OVERVIEW: Esta ventana ofrece el acceso a varias opciones a través de los pictogramas o iconos acompañados de etiquetas dándole al usuario una representación visual de los aspectos del sistema SCADA.

Figura 27. Ventana de Overview



SHOVEL NETWORK: En esta pantalla se visualiza en la figura 28 y muestra todos los elementos de la Red de PLC de la Pala y haciendo clic sobre cada elemento se observa el estado de las señales de entrada y salida de cada nodo de la red.

Figura 28. Ventana Network PLC



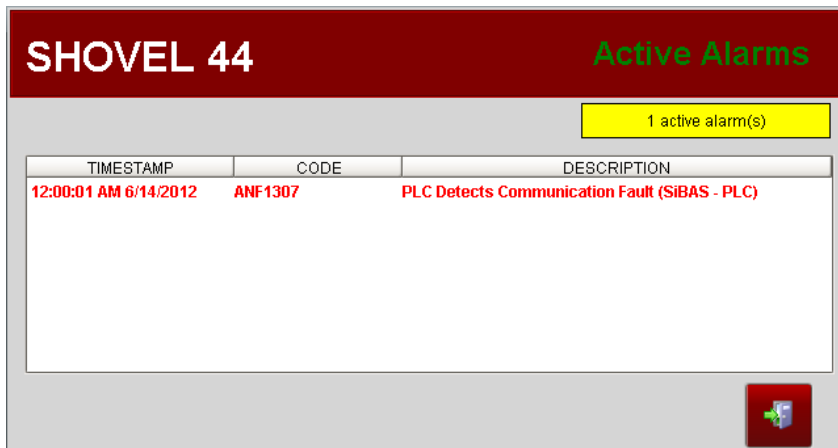
Ingresando en un nodo en particular se visualiza el detalle de las señales. Ver Figura 29.

Figura 29. Nodo Crowd



ACTIVE ALARMS: En esta pantalla como se visualiza en la figura 30 van apareciendo todas las señales que ingresen en el nivel de alarmas.

Figura 30. Active Alarms



HISTORY ALARMS: Muestra los datos históricos de las alarmas que se encuentran almacenados en la base de datos. La información se puede filtrar de acuerdo a un rango de fecha, un tipo de falla o por nombre de la falla. Ver Figura 31

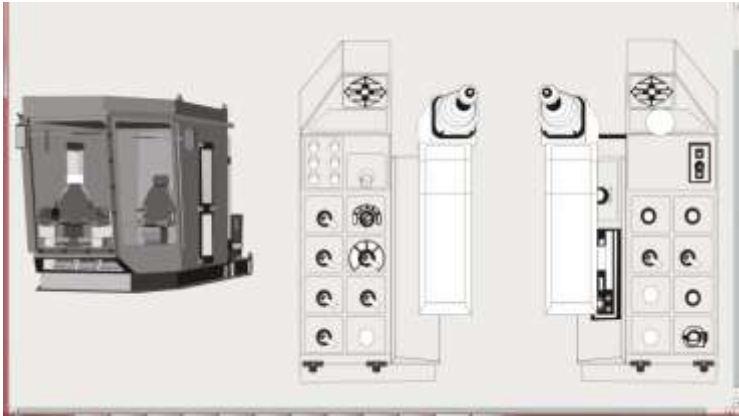
Figura 31. Histórico de Alarmas



UNIFILAR DIAGRAM: Representación unilineal de las conexiones eléctricas de la pala. Las imágenes del objeto real ayudan a la comprensión del diagrama y los interruptores son animados, esto es, dependiendo del estado de operación en que se encuentre el sistema.

CABIN:

Figura 32. Cabina del Operador

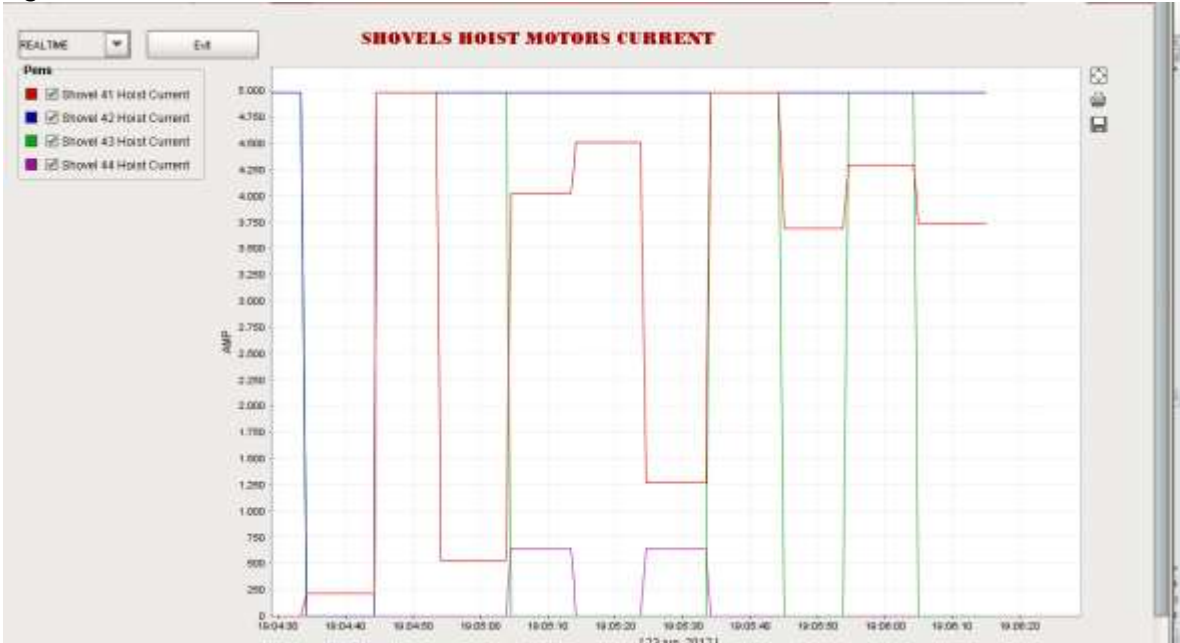


GRÁFICOS DE TENDENCIA:

(TRENDS MOTOR): Como se observa la figura 33, muestran la evolución de los 4 parámetros eléctricos más importantes de los motores: Tensión, Corriente, Potencia real y Potencia aparente. Además, se incluye un componente selector del tipo de tendencia (en tiempo real o histórico).

MECHANICAL TREND: Muestran la evolución de las Variables mecánicas de la máquina tales como temperatura, ángulo de inclinación, vibración.

Figura 33: MOTOR TRENDS



4. IMPLEMENTACIÓN DE LA PLATAFORMA SCADA PARA MONITOREAR LAS PALAS ELÉCTRICAS 41, 42, 43 Y 44 EN LA COMPAÑÍA DRUMMOND LTD.

Una vez considerado el análisis y diseño del sistema SCADA, se describirán los aspectos tenidos en cuenta para la implementación del proyecto.

4.1 REQUERIMIENTOS

4.1.1 Requerimientos de hardware

Acorde al Diagrama de Despliegue o Deployment Diagram, se estima que para el buen funcionamiento del sistema SCADA remoto, se requiere de la siguiente infraestructura de Hardware:

Red inalámbrica: Debido a las grandes distancias (más de 10 kilómetros) y ubicación de las maquinas en lugares remotos de la mina, Drummond ha implementado con anterioridad a este proyecto, sistemas de comunicación por radiofrecuencia que permiten obtener información manual y puntual del estado de las máquinas, sin embargo acorde a las exigencias del sistema SCADA, se prevé como mejora a la infraestructura actual (en segunda fase), la implementación de nuevos radios, pasando de una transmisión de 1Mbps a 54Mbps por cada pala Bucyrus 495HR, esto nos permitirá mejorar los tiempos de Scan de las variables o tags analógicos, mejor conectividad del sistema SCADA y de otros usuarios al tiempo.

Red LAN: En una torre de transmisión central se interconectan la Redes Inalámbricas de las palas Bucyrus 495HR a la red LAN corporativa de Drummond (Existente), y mediante fibra óptica se llega al cuarto de servidores, donde será instalado la aplicación.

Servidor: Requerido para la implementación del sistema SCADA posee características especiales, consideradas acorde a las exigencias del software Ignition (Plataforma de la aplicación SCADA), dentro de las más críticas, se encuentra el uso de la memoria física del servidor cuando múltiples clientes (Máximo = 200 Clientes) están conectados, razón de los 32Gbytes de RAM del Servidor.

PC Cliente: Estación fija, conectada permanentemente al servidor, será manipulada por un operador con conocimiento técnico cualificado de las palas Bucyrus, encargado principalmente de los reportes de alertas y alarmas a los supervisores de mantenimiento eléctrico y mecánico mientras la pala este en operación.

4.1.2 Requerimientos de software:

Sistema Operativo Linux Debian Squeeze x 64: Instalado en el Servidor, seleccionado por la gran estabilidad que ofrece sobre tecnologías Windows, rápido, funcional, con un sinnúmero de herramientas para monitorear la ejecución de procesos, Libre de antivirus evitando el consumo de recursos del servidor, con la mayoría de las aplicaciones desarrolladas como software libre.

Base de Datos MySQL: También instalado en el Servidor, la selección de MySQL como motor de base de datos para el sistema SCADA de este proyecto, obedece a la gran compatibilidad con el sistema Linux Debian, a que es tecnología de software libre (donde su costo es cero), y a que Siemens no utiliza un Browser para almacenamiento de los tags de PLC, de donde se hace necesario instalar un motor de base que realice la funcionalidad de almacenar temporalmente los tag provenientes del PLC y declarados en el sistema SCADA como SQLtags.

Adicional a lo anterior, es de notar que MySQL es la base de datos más usada a nivel mundial, con un 55% más que Oracle, 61% más que SQL, y 57% sobre PostgreSQL, ofreciendo ventajas como la de ser multiplataforma, a diferencia de otras bases de datos, MySQL es más rápido en cuanto a búsqueda e indexaciones, con un mejor manejo de las tablas ahorrando espacio en el disco y procesos en la CPU del servidor, adicionalmente viene en una sola versión, donde todos sus módulos adicionales como php, Python, perl, etc son gratis y fáciles de instalar.

Software SCADA:

El software del sistema SCADA utilizado para este proyecto es llamado Ignition, de la empresa HECHTMAN ENTERPRISES, INC. de INDUCTIVE AUTOMATION, ubicada en Folsom California.

Este potente sistema SCADA instalado en el servidor, está basado en las nuevas tecnologías de JAVA, es multiplataforma y compatible con MySQL, aunque permite conectividad con múltiples bases de datos (Simultaneas).

Sobre Ignition se desarrollara un Script de programa en Python, que permitirá leer directamente del SiBAS los 57 tags de información que se envían al PLC.

Java JRE 6u32

Antes que la base de datos, es el primer pre-requisito a ser instalado en el servidor después de instalar Linux Debian Squeeze x64 bits, se requiere para la instalación y el buen funcionamiento del SCADA Ignition, realmente Ignition puede funcionar con otras versiones de JRE (Ejemplo JRE7u4, la última), pero hemos decidido utilizar la versión 6U32 por la gran estabilidad que ofrece.

El JRE de java es el único requisito exigido para los clientes del sistema SCADA, los clientes pueden instalar cualquier versión de JAVA JRE, excepto la JRE 6u26, con la que Ignition funciona pero con algunos inconvenientes.

Sistema Operativo del PC Cliente Permanente:

Por ser Ignition multiplataforma, el PC cliente puede tener cualquier sistema operativo, Windows, Linux, Mac, Apple. El único requisito después de instalar el sistema operativo, es el mencionado anteriormente, Java JRE en sus diferentes versiones, debido a que los clientes y diseñadores se soportan en Java Web Start.

Browser de Internet:

Es de vital importancia, debido a que se utiliza para realizar la configuración del Software Ignition y el acceso de los clientes a la aplicación.

4.2 HMI IMPLEMENTADO EN IGNITION

4.2.1 Configuración de Ignition gateway

La principal característica de un sistema SCADA es la adquisición de datos. Para iniciar cualquier transmisión es necesaria la configuración del servidor de datos OPC-UA y cada uno de los dispositivos conectados a la red.

4.2.1.1 Configuración del OPC – UA

Inductive Automation como miembros activos de la fundación OPC son pioneros en el desarrollo del estándar OPC UA (Unified Architecture), desarrollado para la reglamentación de las comunicaciones industriales a nivel de PLC.

Una conexión OPC- UA hace posible el acceso a datos, históricos, alarmas, eventos y mejora capacidad de comunicación a través de Internet.

Para la configuración de una nueva conexión (ver figura 34) son necesarios los siguientes parámetros:

- Nombre de la conexión
- Descripción
- Host: corresponde a la dirección IP
- Port: Como valor predeterminado para la conexión del OPC- UA se usa el puerto 4096.

Figura 34. Configuración OPC –UA

Authentication	
Authentication Profile	opcua-module <small>The name of the authentication profile to authenticate against.</small>
Allowed Roles	ReadWrite <small>Separate multiple roles with a comma (ex: Administrator,user,manager)</small>
Allow Anonymous Access	<input type="checkbox"/> <small>Allow users to connect whether or not they possess authentication credentials. (Default: false)</small>

Server	
Server Port	4096 <small>The port on the local machine the server will run on. Requires a module restart to take effect.</small>
Endpoint Address	172.16.149.1 <small>This is the local address that the Ignition UA server will bind to. It is also the address that will be used in a GetEndpointResponse, so it is important that this be an address reachable by any clients that wish to connect. Requires a module restart to take effect.</small>
Minimum Sampling Interval	100 <small>The fastest rate (in milliseconds) that the server will use to sample its underlying data sources. Requires a module restart to take effect. (Default: 100)</small>

Discovery Server	
Discovery Server Enabled	<input checked="" type="checkbox"/> <small>True if the local discovery server should be enabled. Requires a module restart to take effect. (Default: true)</small>

Other	
Stale Threshold	5 <small>The multiplier by which the server determines that updates from a driver have become stale. This period will be calculated as the fastest sampling rate for that node multiplied by this settings value. (Default: 5)</small>

4.2.1. 2 Configuración de los Dispositivos

Ignition en su página de configuración web, coloca a disposición del usuario el Driver que permite ser configurado con el PLC a conectar, actualmente han creado drivers para PLC Siemens, Allen Bradley, Modbus, UDP y conexiones TCP.

De los módulos desarrollados por Ignition, para el sistema SCADA solo se ha requerido del Driver Siemens para conectar con los PLCs principal de cada pala, y del Driver modbus para los medidores de energía PQM de General Electric y el sistema Contraincendios Fenwal.

El uso de estos drivers se realiza mediante declaraciones múltiples, en la que se colocan los parámetros de comunicación declarados por el fabricante y usuario final en cada dispositivo.

Para el caso de los PLC S7400 de Siemens, Ejemplo: Pala 41 como sigue:

Se le da un nombre al Dispositivo o PLC a conectar, en este caso Shovel41PLC.

Se colocan los diferentes valores de tiempo timeout, considerando que sean a lo menos tres veces más que los tiempos de scan utilizados para la lectura de tags individuales.

El Hostname, que corresponde a la dirección IP = 10.102.100.12 de la tarjeta Ethernet del PLC S7400 en la pala 41, ver figura 35:

Figura 35. Configuración PLC Siemens S7-400

General	
Device Name	<input type="text" value="Shovel41PLC"/> Name of this device.
Group	<input type="text" value="Default Group"/> An arbitrary named group for this device to be a part of. This is primarily used as a tool for organizing the list of installed devices.
Browse Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a browse operation times out.
Read Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a read operation times out.
Write Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a write operation times out.
Enable Device	<input checked="" type="checkbox"/> Enable this device?

Connectivity	
Hostname	<input type="text" value="10.102.100.12"/> The hostname or IP address the driver should try to connect to.
Communication Timeout	<input type="text" value="1000"/> The time in mSec to wait for a device to respond.

En las opciones avanzadas del Driver Siemens de Ignition, se declara el número del Rack = 0, posición física de la CPU = 3 en el Slot o Rack de soporte, y el puerto 102 de comunicaciones Ethernet definido por siemens como se visualiza en la figura 36.

Figura 36. Configuración avanzada del PLC Siemens

Advanced	
Port	<input type="text" value="102"/> The TCP port to connect on.
PDU Size	<input type="text" value="240"/> The PDU size to start at when negotiating with the device.
Rack Number	<input type="text" value="0"/> Rack this device is located in.
CPU Slot	<input type="text" value="3"/> Slot number the CPU is located in.
Reconnect After Consecutive Timeouts	<input checked="" type="checkbox"/> Force a reconnect after 3 consecutive timeouts.

Los drivers Modbus utilizados para declarar los medidores de energía "PQM" y la red contraincendios "Fenwall" de cada pala Bucyrus, son configurados semejantes al driver de Siemens, con la diferencia a que el protocolo Modbus TCP utiliza el puerto 502, y los holding registers, que no son más que posiciones de memoria donde se almacenan la información de los tags a requerir por el sistema SCADA, como se ilustra en la figura 37:

Figura 37. Configuración de los PQM

Modbus v2

General	
Device Name	<input type="text" value="Shovel41PQM"/> Name of this device.
Group	<input type="text" value="Default Group"/> An arbitrary named group for this device to be a part of. This is primarily used as a tool for organizing the list of installed devices.
Browse Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a browse operation times out.
Read Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a read operation times out.
Write Timeout	<input type="text" value="3000"/> Amount of time (in milliseconds) before a write operation times out.
Enable Device	<input checked="" type="checkbox"/> Enable this device?

Connectivity	
Hostname	<input type="text" value="10.102.100.16"/> Hostname/IP address of the Modbus device.
Port	<input type="text" value="502"/> Port to connect to.
Communication Timeout	<input type="text" value="1000"/> Maximum amount of time to wait for a response.

Ejemplo 1: de los registros de memoria (Holding Register) de los medidores de energía PQM de General Electric en la pala 41, como se aprecia en la figura 38, las lecturas de la posiciones de memoria 640 a la 800, y de la 1134 a la 1147. Adicionalmente el tipo de datos enteros "INT16" a ser leídos.

Figura 38. Configuración avanzada
Address Configuration

No se el... archivo [Import Configuration](#)

[Export Configuration](#)

Prefix	Start	End	Step	Unit ID	Modbus Type	Modbus Address	
<input type="text" value="0"/>	<input type="text" value="640"/>	<input type="text" value="800"/>	<input type="checkbox"/>	<input type="text" value="0"/>	Holding Register (Int16) ▼	<input type="text" value="640"/>	[delete]
<input type="text" value="0"/>	<input type="text" value="1134"/>	<input type="text" value="1147"/>	<input type="checkbox"/>	<input type="text" value="0"/>	Holding Register (Int16) ▼	<input type="text" value="1134"/>	[delete]

Radix

[Add Row](#)

Ejemplo 2: como se ilustra en la figura 39 de los registros de memoria (Holding Register) del sistema contraincendios de la pala 41. De donde solo es leído el registro 2, y los registros de la posición de memoria 10 a la 27, adicionalmente el tipo de dato a ser leído.

Figura 39. Configuración Sistema contra Incendio

Address Configuration

No se el... archivo [Import Configuration](#)

[Export Configuration](#)

Prefix	Start	End	Step	Unit ID	Modbus Type	Modbus Address
<input type="text" value="0"/>	<input type="text" value="002"/>	<input type="text" value="002"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	Holding Register (Int16) ▼	<input type="text" value="002"/> [delete]
<input type="text" value="0"/>	<input type="text" value="010"/>	<input type="text" value="027"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	Holding Register (Int16) ▼	<input type="text" value="010"/> [delete]

Radix

[Add Row](#)

4.2.1.3 Configuración de la Base de Datos

Muchas de las características de Ignition como el historiadador, alarmas requieren una conexión a una base de datos; Para esta aplicación utilizaremos el sistema de gestión de base de datos MYSQL por su condición de Open Source, por su gran adaptación a diferentes entornos de desarrollo y su integración en distintos sistemas operativos.

En la interfaz web del Gateway de Ignition se crea una conexión a la base de datos MYSQL a través del driver JDBC.

El formato de la conexión MySQL tiene los siguientes parámetros:

`jdbc:mysql:/host/puerto/base de datos`

`jdbc:mysql://localhost:3306/mysql`

Host: El nombre de host o dirección IP del servidor de base de datos.

Puerto: El puerto del servidor de base de datos. El puerto de MYSQL por defecto es 3306.

Base de Datos: Nombre de la base de datos.

Además para conectarse se debe pasar los parámetros del nombre de usuario y la contraseña de la base de datos.

En la figura 40 se visualiza la configuración actual de la Base de Datos

Figura 40. Configuración de Base de Datos.

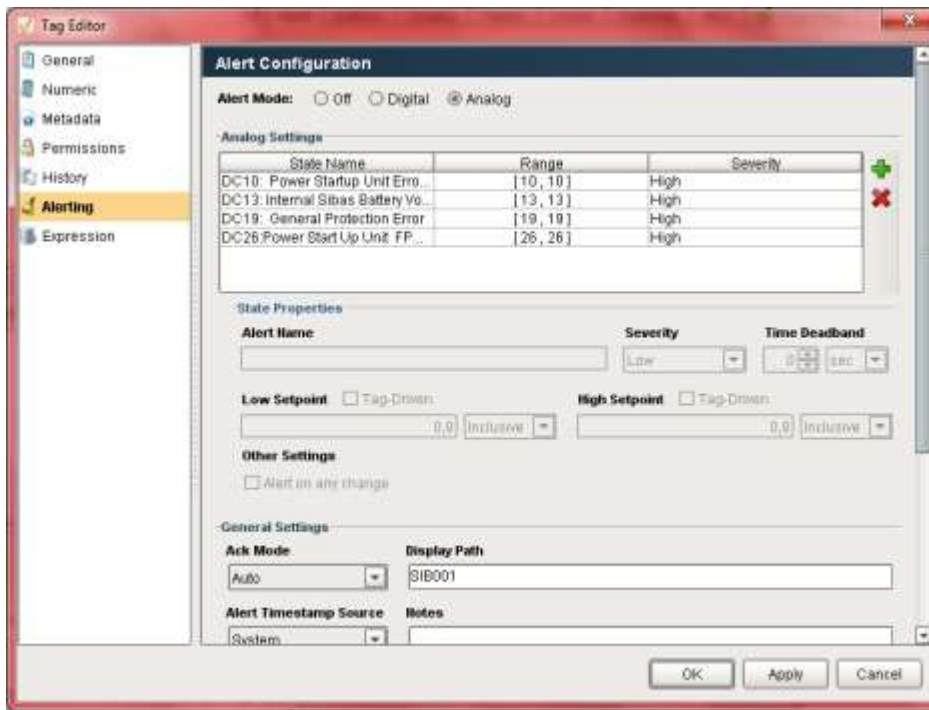
The screenshot shows the 'Edit Database Connection' window with the following configuration:

Main Properties	
Name	MySQL <small>Warning: Changing the name of a database connection is risky. Any projects that refer to this connection by name (instead of referring to their project default) will start causing errors trying to connect to a connection that no longer exists. Please verify that no projects refer to this connection by name, and update the ones that do.</small>
Description	Connection to Ignition BI495HR Shovels Database
JDBC Driver	MySQL Connector/J <small>The JDBC driver dictates the type of database that this connection can connect to. It cannot be changed once created.</small>
Connect URL	jdbc:mysql://localhost:3306/mysql <small>The Connect URL is JDBC-driver specific. It usually contains the address of the machine that the database is running on. The format of the MySQL connect URL is: jdbc:mysql://host:port/database With the three parameters (in bold) host: The host name or IP address of the database server. port: The port that the database server is running on. MySQL default port is 3306. database: The name of the logical database that you are connecting to on the MySQL server.</small>
Username	root
Change Password?	<input type="checkbox"/> Check this box to change the existing password.
Password	[Redacted]
Password	[Redacted] <small>Re-type password for verification.</small>
Extra Connection Properties	zeroDateTimeBehavior=convertToNull <small>There is an extensive list of extra connection properties available for MySQL Connector/J. See the documentation for a table describing all connection properties.</small>

4.2.1.4 Configuración Alarmas

Las alarmas es un estado de falla detectado a partir de condiciones que se evalúan con respecto a un determinado punto. A los Tags considerados indicadores de alarma se debe asignar una propiedad llamada Alerting que permite que sus atributos se pasen a una situación de alarmas en condiciones preestablecidas en esta propiedad por ejemplo una sobre temperatura de los reductores de giro y se establece la alarma cuando sobrepasa los 90°C (ver figura 41). Cuando se produce una alarma, se registra la hora y la fecha, prioridad y estado mientras se encuentre activa.

Figura 41. Configuración de Alarmas



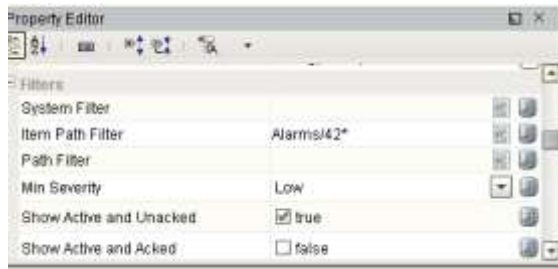
A nivel gráfico, el Ignition visión proporciona un objeto de alarma como el mostrado en la figura 42. A nivel funcional, esta ventana servirá para visualizar todas las alarmas de nivel sin reconocer. Hasta que los valores que han hecho disparar las alarmas no vuelvan a la normalidad, no desaparecerán de la tabla.

Figura 41: Alarmas Activas



Cada pala Bucyrus tiene un objeto de alarma asociado configurado por el Item Path de cada máquina (ver figura 43) y un contador de alarmas. El ítem Path viene determinado por la ubicación de los Tags de alarmas en el sql Tags.

Figura 43: Propiedades del Objeto Alarma



Contador de alarmas

```
runScript("system.alert.queryAlertStatus(active=1,path='"+{Root container.Label 3.tagPath} + "').rowCount", 1000)
```

Para crear un registro histórico de eventos de alerta se crean y configurar tablas para almacenamiento de las alarmas en la base de datos dentro del Gateway de Ignition.

En la configuración del almacenamiento de las alarmas se determinan los siguientes parámetros (ver Figura 44 y 45):

- Nombre de la base de Datos
- Nombre de la Tabla que se utilizará para el almacenamiento de los eventos de alerta.
- Periodo: El periodo indica el tiempo que estará almacenada el registro en la base de datos.

- Filtro: Se puede filtrar las alarmas de acuerdo a la prioridad, ruta y estados.

Figura 44. Almacenamiento de Alarmas

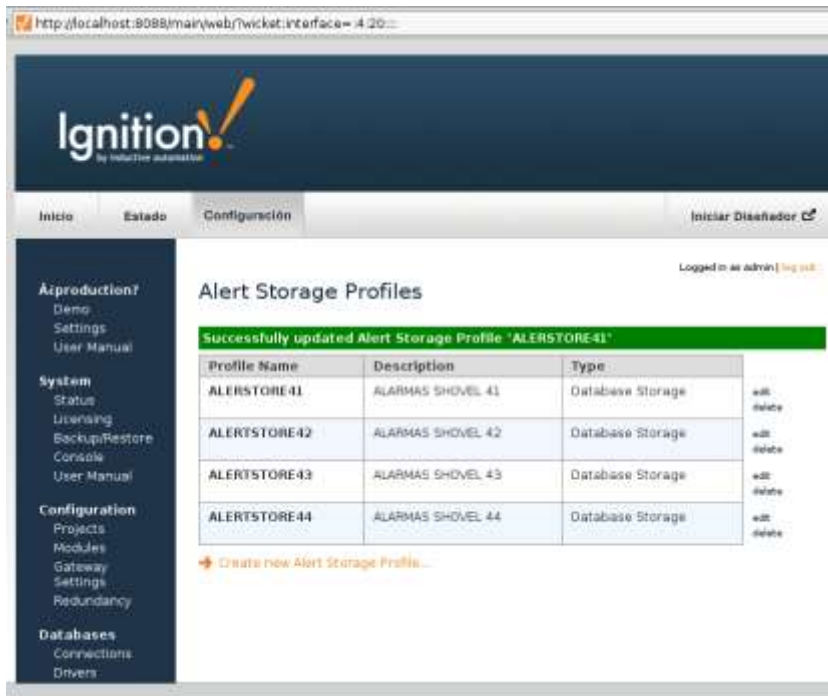


Figura 45. Configuración del almacenamiento de Alarmas



Para visualizar el histórico de alarmas se utiliza un componente Tabla con datos conmutables para las 4 palas que realiza una consulta a la base de datos.

```
SELECT ACTIVE_TIMESTAMP, DISPLAYPATH, STATE_NAME FROM {Root
Container.Tabla}
WHERE
{Root Container.CLAUSULA}
ORDER BY ACTIVE_TIMESTAMP DESC
```

Donde CLAUSULA es:

```
"ACTIVE_TIMESTAMP >= " + {Root Container.INICIO.date} +" ' "+
" AND ACTIVE_TIMESTAMP <= " +{Root Container.FIN.date}+" "' +
if (len({Root Container.Container 1.Text Field.text})>0,
"AND DISPLAYPATH = "+{Root Container.Container 1.Text Field.text}+" ",
""
) +
if ({Root Container.Container.Radio Button.selected},
"AND DISPLAYPATH LIKE " + 'ANF'+ "%",
if ({Root Container.Container.Radio Button 1.selected},
"AND DISPLAYPATH LIKE " + 'CDA'+ "%",
if ({Root Container.Container.Radio Button 2.selected},
"AND DISPLAYPATH LIKE " + 'SiB'+ "%"; ""))
)
```

Configuración para notificación de Alertas.

Las alertas serán notificadas por medio de mensaje vía correo electrónico a una lista de destinatarios teniendo en cuenta la prioridad asociada según la gravedad de la incidencia ocasionada. En función de su importancia y su procedencia se debe decidir si la alarma ha de ser notificada o no. En nuestro sistema, para que una alarma sea notificada deberá superar un filtro en el cual la alarmas tiene prioridad Alta y que implica un paro de las palas.

Para la configuración es necesario disponer de un servidor de correo electrónico

Los parámetros de configuración son los siguientes:

- Servidor de Correo Electrónico utilizado para enviar las alertas
- Puerto a través del cual se debe realizar la comunicación

- Si el servidor de correo precisa autenticación, se activa la casilla de verificación e indique un nombre de usuario y contraseña válidos para el servidor de correo.

Las propiedades avanzadas permiten configurar los criterios para filtrar los tipos de fallas que serán reportados a través del correo electrónico, para el caso solo serán notificadas las alarmas que ocasionan un paro en el equipo y que son representadas con prioridad alta.

En la figura 46 y 47 se visualizada la configuración.

Figura 46. Configuración de notificación de alarmas

The screenshot displays the 'Edit Alert Notification Profile' configuration page. On the left is a navigation menu with categories like System, Configuration, Databases, OPC Connections, OPC UA, Mobile, SQL Tags, and Security. The main content area contains the following fields:

- Profile Name:** Shovel's Alarms
- Description:** Shovel's Alarms Reports
- Mail Server:** 192.168.180.232
- Port:** 25
- From Address:** Shovel'sAlarm@summondtd.com
- Use Authentication:**
- Username:** [Redacted]
- Change Password?:**
- Password:** [Redacted]
- Password:** [Redacted]
- Use SSL:**


Figura 47. Filtrado de alarmas

Show advanced properties?

Pre-Process Filter Criteria

System	<input type="text"/>	A filter string used to filter notification of alerts by their originating System. Can use an asterisk (*) and the question mark (?) to match any or exactly one character. Example: 'SQLTags.*' would match alerts coming from the SQLTags system (any provider), but not other alerts.
Path	<input type="text"/>	A filter string used to filter notification of alerts by their path.
State Name	<input type="text"/>	A filter string used to filter notification of alerts by their state name.
Minimum Severity	High	If specified, only alerts with greater or equal severity will be processed.
Maximum Severity	Selecione uno	If specified, only alerts with lesser or equal severity will be processed.

Ignition by Inductive Automation. Copyright © 2003-2012. All rights reserved. | [View license](#)



Por último se debe crear el listado de los destinatarios de los correos electrónicos. Ver figura 48

Figura 48. Listado de Email

Logg

System

- Status
- Licensing
- Backup/Restore
- Console
- User Manual

Configuration

- Projects
- Modules
- Gateway Settings
- Redundancy

Manage Recipients for Profile 'Shovels Alarms'

Email Addresses	
cpcs_76@yahoo.es	delete
wlopez1@drummondLtd.com	delete

4.2.1.5 Configuración de usuarios

Ignition usa el concepto de seguridad basados en Roles. Las directivas de seguridad se definen en términos de estos roles en vez de definirse para usuarios específicos. Esto permite a los usuarios ser agregado, reasignado y eliminado, sin afectar a la lógica de la política de seguridad.

Dentro de las políticas de seguridad implementadas para el sistema SCADA se definieron 5 roles (ver Figura 49):

- **Administrador:** Implementa la configuración del Gateway y tiene acceso al diseñador.
- **Diseñador:** Se restringe el acceso a la configuración del Gateway.
- **Operadores - Supervisores:** Solo tiene acceso al runtime y se restringe su acceso para manipular un componente o pantalla.

UserDrummond: Solo tiene acceso al runtime, los usuarios de este rol no trabajan directamente con la maquina pero hacen parte del personal directivo de la empresa. Estos usuarios existen en la base de datos del Directorio Activo.

Figura 49: Definición de Roles

Users		Roles
Roles		
Role Name	Description	
IgnitionDesigners		edit delete
Ignition_Administrator		edit delete
Operators		edit delete
Supervisors		edit delete
UsersDrummond		edit delete

Para la gestión de los usuarios se crearon 2 métodos de autenticación con el objeto de permitir el acceso a todos los empleados de la compañía.

El primer método utiliza una conexión a una Base de Datos externa donde se almacena los roles e información de los usuarios y la autorización es administrada con esta solución.

En el siguiente código se muestra el botón de creación de usuarios desde un controlador de eventos:

```

username = event.source.parent.getComponent('Username').text
password = event.source.parent.getComponent('Password1').text
password2 = event.source.parent.getComponent('Password2').text
role = "Ignition" + event.source.parent.getComponent('Dropdown').selectedStringValue

if password != password2:
    system.gui.errorBox('Passwords do not match.')
elif len(username)==0:
    system.gui.errorBox('Please enter a username.')
else:
    probe = system.db.runScalarQuery("SELECT Username FROM USERS WHERE
Username='%s'%username)
    if probe != None:
        system.gui.errorBox("User '%s' already exists"%username)
    else:
        insertUserQuery = "INSERT INTO USERS (Username>Password) VALUES
('%s','%s')"%username,password)
        system.db.runUpdateQuery(insertUserQuery)
        system.db.runUpdateQuery("INSERT INTO
USER_ROLE_MAPPING(Username,Rolename VALUES ('%s','%s')"% (username, role))
        system.gui.messageBox("User '%s' has been created" % username)
        event.source.parent.getComponent('Username').text = ""
        event.source.parent.getComponent('Password1').text = ""
        event.source.parent.getComponent('Password2').text = ""

```

El siguiente script muestra el cambio de password. Sólo es posible cambiar el password del usuario que se encuentra conectado.

```

password = event.source.parent.getComponent('Password2').text
password2 = event.source.parent.getComponent('Password3').text
username = event.source.parent.getComponent('Username').text
if password != password2:
    system.gui.errorBox('Passwords do not match.')
else:
    insertUserQuery = "UPDATE USERS SET Password = '%s' WHERE Username =
'%s'%password,username)
    system.db.runUpdateQuery(insertUserQuery)
    system.gui.messageBox("User '%s' has been modified" % username)
    event.source.parent.getComponent('Password2').text = ""
    event.source.parent.getComponent('Password3').text = ""

```

En el siguiente el evento permite el cambio de usuario

```

uname = event.source.parent.getComponent("Username").text
pwd = event.source.parent.getComponent("Password").text
success = system.security.switchUser(uname,pwd,event)
if not success:
    event.source.parent.getComponent("Username").requestFocusInWindow()

```

En el segundo perfil se configura la integración de los usuarios corporativos del servidor Active Directory a través del protocolo LDAP a la plataforma de Ignition, especificando los siguientes parámetros (ver figura 50):

Dominio del servidor del Active Directory, en este caso el dominio de la multinacional Drummond es `laloma.drummondco.net`.

Especificar el host que actúa como controlador de dominio principal. También puede utilizar un controlador de dominio secundario en caso de que el principal no está disponible.

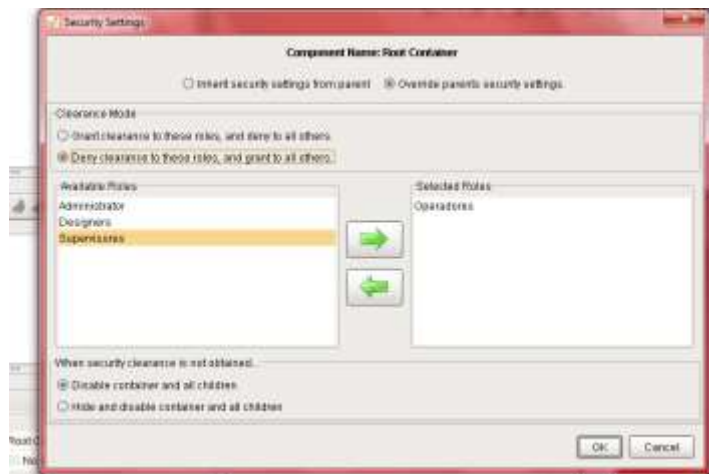
Puerto LDAP: 389

Figura 50. Propiedades Active Directory

Active Directory Properties	
Domain	<input type="text" value="laloma.drummondco.net"/> <small>The Windows domain for this Active Directory server. Examples: "MyCompany.com" or "SuperCorp.local". If you aren't sure of your domain, ask your network administrator. Leave blank to set advanced properties manually.</small>
Gateway Username	<input type="text" value="services"/> <small>The login name for the gateway to use when querying Active Directory. Only used for retrieving the entire list of roles for the Designer. Not strictly necessary for authentication. (default:)</small>
Change Password?	<input type="checkbox"/> <small>Check this box to change the existing password.</small>
Password	<input type="password"/> <small>The password for the above username.</small>
Password	<input type="password"/> <small>Re-type password for verification.</small>
Primary Domain Controller Host	<input type="text" value="192.168.180.213"/> <small>The IP address or hostname of your primary domain controller. Example: "192.168.1.4" or "MainServer"</small>
Primary Domain Controller Port	<input type="text" value="389"/> <small>The port number for the primary domain controller's LDAP interface. (default: 389)</small>
Secondary Domain Controller Host	<input type="text" value="192.168.180.215"/> <small>The IP address or hostname of your secondary domain controller (optional). Example: "192.168.1.4" or "MainServer"</small>
Secondary Domain Controller Port	<input type="text" value="389"/> <small>The port number for the secondary domain controller's LDAP interface. (default: 389)</small>
<input type="checkbox"/> Show advanced properties?	
<input type="button" value="Save Changes"/>	

Se establece para el rol de los operadores y supervisores restricciones en el acceso a pantallas como se observa en la Figura 51.

Figura 51: Políticas de seguridad para los componentes.



4.3 ANIMACION DE LAS PALAS EN LA PANTALLA PRINCIPAL

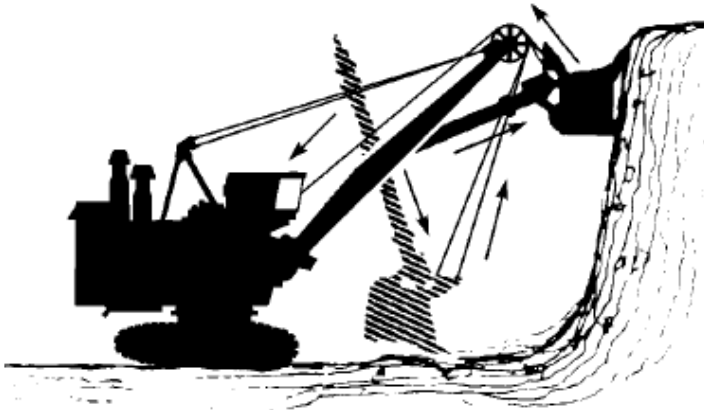
En el sistema hay características de la máquina que son imprescindibles mostrar en la interfaz gráfica como son los movimientos de Propel, Hoist y Crowd dada que solo la observación de estos atributos permite conocer el estado general de la Pala.

La palas Bucyrus para el control de movimiento asocian a sus motores eléctricos sensores de posición a los que se les llaman "Encoders", que le indican al PLC la acción que está ejecutando la pala en un momento dado.

Mediante la lectura de información de los tags provenientes de cada encoder se ha de construir la animación en tiempo real del estado de movimiento de las cuatro palas en la pantalla principal, como se indica en la figura 51, 52, 53:

Movimiento del Hoist: Es un movimiento angular, realizado desde -270° Grados a 0° Grados, permite subir y bajar el balde de la pala para la recogida y posterior descargue de material en los camiones de transporte.

Figura 52 .Movimiento del Hoist



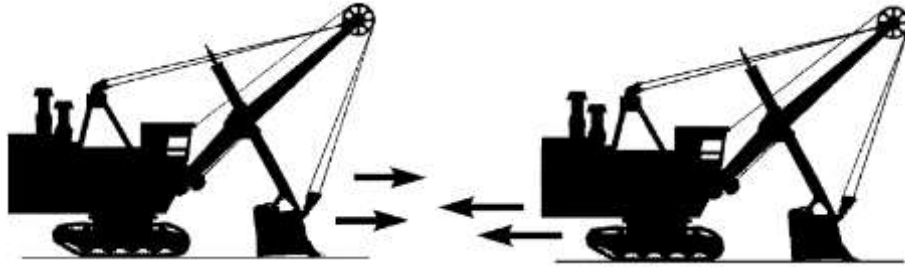
Movimiento del Crowd: Es un movimiento lineal, realizado sobre el eje de las X, su desplazamiento está definido por los límites de operación preestablecidos para la correcta operación de la máquina.

Figura 53. Movimiento del Crowd



Movimiento de Propel: Se realiza también sobre el eje de las X, con la variante de que si una de las orugas se queda estática mientras que la otra se mueve, se genera un movimiento de giro sobre su eje, para toda la pala, paralelo al desplazamiento hacia adelante y hacia atrás.

Figura 54. Movimiento de Propel



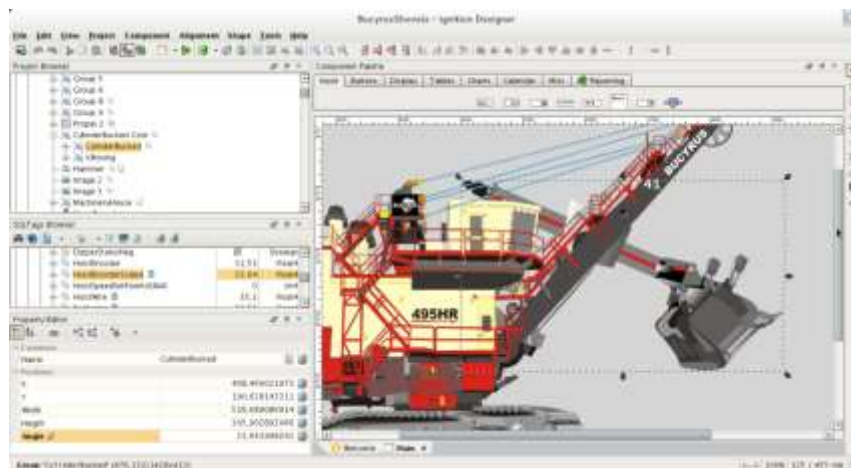
Tags para los Movimientos:

Cada valor de tag para cada movimiento, antes de ser asociado a la gráfica sobre la cual realizara el efecto del movimiento, es procesado, y escalizado, el procesamiento de señal se realiza sobre el PLC y la escalización en el SCADA Ignition, de manera que se den las proporcionalidades de la acción a ejecutar, así como sigue:

Tag del Hoist:

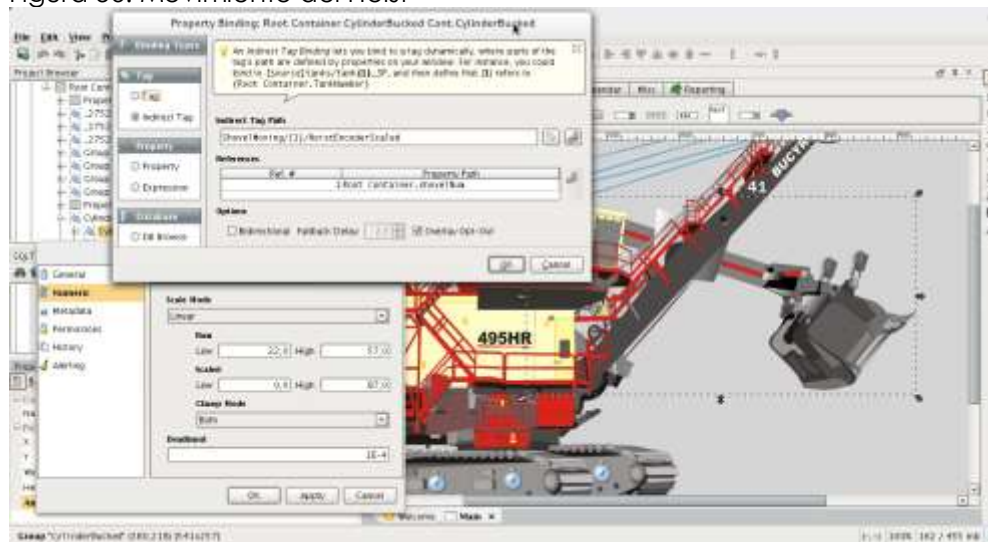
En Diseñador de Ignition en la figura 55, se aprecia el valor del Tag de HoistEncoderScaled = Angle, es el resultado de una operación matemática, ecuación lineal o escalización del tag HoistEncoder, el valor del tag en esta imagen es 23,64 grados respecto a la horizontal, dentro del rango de 0 a 87 grados, su variable es un numero de coma flotante de orden 4, float4 (que corresponde a cuatro palabras de 8 bits, para un total de 32 bits).

Figura 55. Encoder



El concepto más importante manejado en el SCADA Ignition es el Binding o enlace, para el Angle, un valor lineal proveniente del enlace HoistEncoder, es escalizado de 22 a 0 grados, y de 57 a 87 grados. Posteriormente se carga este valor a la gráfica punteada con el enlace ShovelMoving/{1}/HoistEncoderScaled, donde {1} es un conmutador que adquiere el valor de la pala seleccionada. Ver la figura 56

Figura 56. Movimiento del Hoist



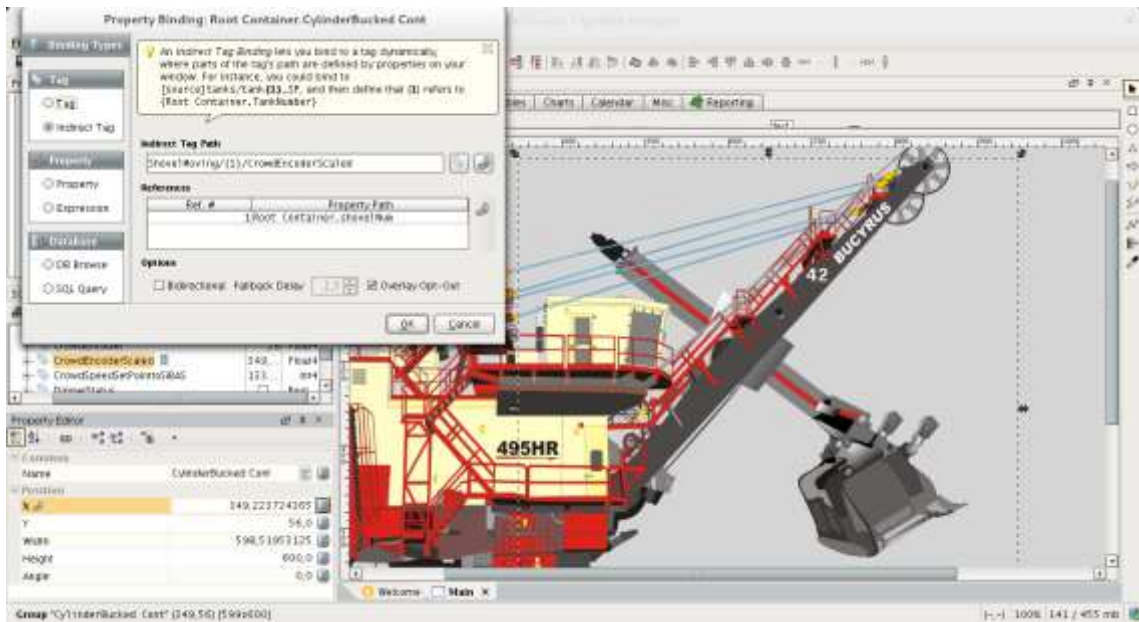
Tag del Crowd:

La animación anterior CylinderBucked se coloca dentro de un contenedor, nombrado CylinderBucked Cont (Figura 57 punteada a continuación), el contenedor se mueve sobre el eje de las X, hacia adelante y hacia atrás, arrastrando y desplazando la gráfica CylinderBucked que puede estar a su vez realizando el movimiento angular del Hoist.

En la gráfica siguiente, se aprecia el valor escalizado de CrowdEncoder, el cual es llamado CrowdEncoderScaled = X = 362,17 que corresponde a una coordenada de la pantalla gráfica.

De la misma manera el tag asociado al contenedor, con el path o ruta: ShovelMoving/{1}/CrowdEncoderScaled

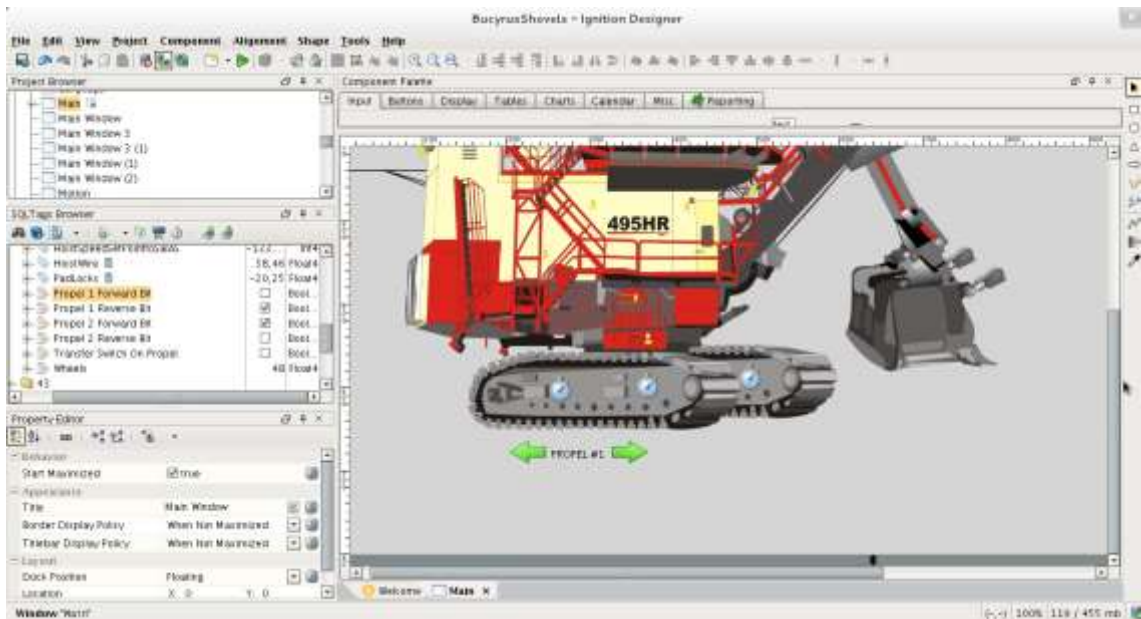
Figura 57. Movimiento del Crowd



Tags de Propel

Los movimientos de Propel poseen dirección y velocidad, esta animación se encuentran en fase de desarrollo, se observa el grupo de señales digitales o booleanas para el direccionamiento forward (Hacia adelante) y reverse (Hacia atrás), posteriormente tendrán un segundo enlace, que asociara el cambio de velocidad con un cambio de colores, indicando el desplazamiento y dirección de cada oruga como se observa en la figura 58.

Figura 58. Movimiento de Propel



Estados del Movimiento: Se detalla a continuación en las figuras 59, 60 y 61, movimientos captados de la pantalla principal, esta animación representa movimientos en tiempo real de las cuatro palas.

Figura 59. Posición 1 del Bunked



Figura 60. Posición 2 del Bunked



Figura 61. Posición 1 del Bunked



Pala en posición de reposo.

En la figura 62 se visualiza la pala 44, en ventana de mantenimiento mecánico, la animación registra la posición de la escalera de ingreso a la pala (lado derecho) y la el balde en un punto de trabajo seguro.

Figura 62. Posición de reposo de las palas

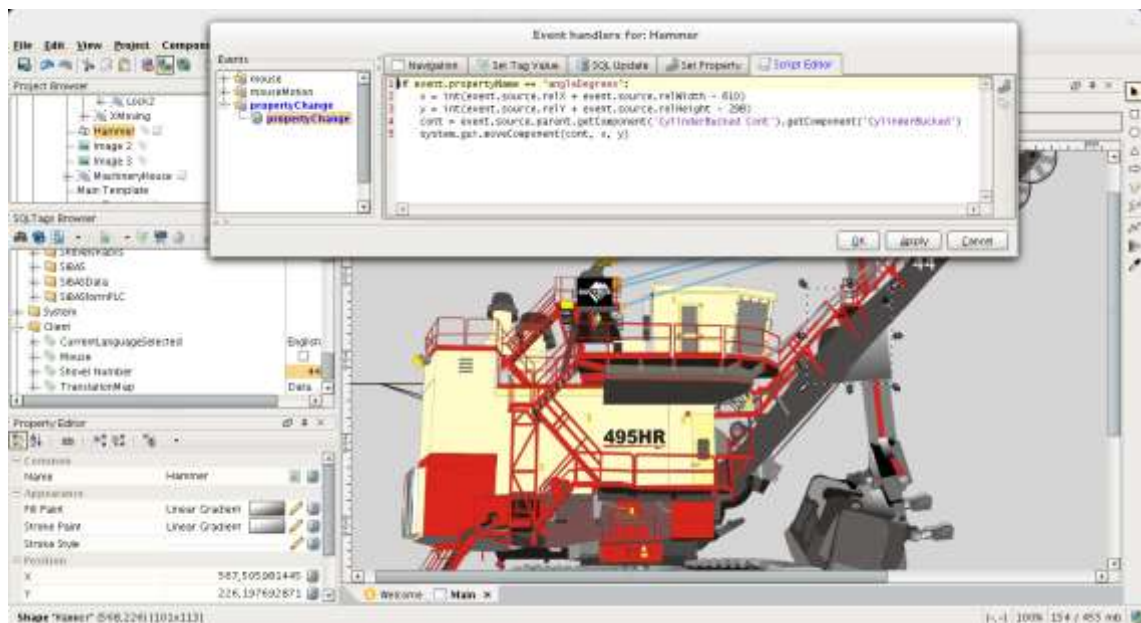


Script del martillo

Como fue descrito, el cilindro del balde se desplaza dentro de un soporte en forma de martillo, ver segmento punteado en la figura 63, una forma de juntar dos gráficas, es mediante una resta de coordenadas, la cual consiste en restarle de las coordenadas del CylinderBucked Cont la posición de las coordenadas del martillo, pero esto se realiza dentro de las propiedades de la gráfica martillo, y hace que una gráfica enlace y mantenga a la otra dentro de un rango de trabajo.

```
if event.propertyName == "angleDegrees":
    x = int(event.source.relX + event.source.relWidth - 610)
    y = int (event.source.relY + event.source.relHeight - 298)
    cont = event.source.parent.getComponent('CylinderBucked
Cont').getComponent('CylinderBucked')
    system.gui.moveComponent(cont, x, y)
```

Figura 63. Script del Martillo



4.4 DRIVER DE COMUNICACIÓN DEL PLC SiBAS

El driver de comunicación en entornos industriales es un programa que permite la comunicación entre los dispositivos de campo y el OPC Server, cada dispositivo a través utiliza un protocolo que “traduce” la señal que envía y recibe.

En el PLC SiBAS se definen tres tipos de datos:

BOOL: Tamaño en bit 1. Rango True o False

BYTE: Tamaño en Bit 8. Rango de 0 a 255

INT: Tamaño en Bit 16. Rango -32.768 a 32.767

Para el intercambio de datos el PLC SiBAS utiliza el Protocolo HTTP y mediante esta conexión se realiza la lectura de 57 registros o tags de información y en donde cada registro de memoria o tags se encuentra plenamente identificado. Ver Figura 64

Figura 64: Vista del datablock del PLC SiBAS

Address	Name	Type	Initial value	Comment
*0.0	HDSE	BOOL	FALSE	HOIST DRIVE ON: Hoist status; Sibas to PLC
*0.1	HDSD	BOOL	FALSE	HOIST DRIVE OFF: Hoist status; Sibas to PLC
*0.2	HDSF	BOOL	FALSE	HOIST DRIVE FAULT: Hoist status; Sibas to PLC
*0.3	HDSW	BOOL	FALSE	HOIST DRIVE WARNING: Hoist status; Sibas to PLC
*0.4	HDMC	BOOL	FALSE	HOIST MINIMUM CURRENT: Hoist status; Sibas to PLC
*0.5	HDSR	BOOL	FALSE	HOIST DRIVE READY TO START: Hoist status; Sibas to PLC
*0.6	HDSB	BOOL	FALSE	HOIST/PROPEL: DRIVE BRAKE SET: Hoist status; Sibas to PLC
*0.7	HDVSpd	BOOL	FALSE	Hoist Overspeed
*1.0	HDSOM1	BOOL	FALSE	HOIST OPERATING MODE BIT 1: Hoist status; Sibas to PLC
*1.1	HDSOM2	BOOL	FALSE	HOIST OPERATING MODE BIT 2: Hoist status; Sibas to PLC
*1.2	HDSOM3	BOOL	FALSE	HOIST OPERATING MODE BIT 3: Hoist status; Sibas to PLC
*1.3	HDTF1	BOOL	FALSE	HOIST TEMP FAULT INV/1: Hoist status; Sibas to PLC
*1.4	HDTF2	BOOL	FALSE	HOIST TEMP FAULT INV/2: Hoist status; Sibas to PLC
*1.5	HDNP	BOOL	FALSE	HOIST NORMAL PARAMETER SET/DIG: Hoist status; Sibas to PLC
*1.6	HDAP	BOOL	FALSE	HOIST ALTERNATE PARAMETER SET/PROPEL: Hoist status; Sibas to PLC
*1.7	HDS	BOOL	FALSE	HOIST SPEED < 1.5m (5%): Hoist status; Sibas to PLC
*2.0	HDSF	BYTE	#16#0	HOIST SPEED FEEDBACK: Hoist status; Sibas to PLC
*3.0	HDTF	BYTE	#16#0	HOIST TORQUE FEEDBACK: Hoist status; Sibas to PLC
*4.0	HDCF	BYTE	#16#0	HOIST CURRENT FEEDBACK: Hoist status; Sibas to PLC
*5.0	HDACT	BYTE	#16#0	HOIST ACTIVE CURRENT FEEDBACK: Hoist status; Sibas to PLC
*6.0	MS HOIST	BYTE	#16#0	HOIST Master Switch Status
*7.0	CDSE	BOOL	FALSE	CROWD DRIVE ON: Crowd status; Sibas to PLC
*7.1	CDSD	BOOL	FALSE	CROWD DRIVE OFF: Crowd status; Sibas to PLC
*7.2	CDWF	BOOL	FALSE	CROWD DRIVE FAULT: Crowd status; Sibas to PLC
*7.3	CDWW	BOOL	FALSE	CROWD DRIVE WARNING: Crowd status; Sibas to PLC
*7.4	CDMC	BOOL	FALSE	CROWD MINIMUM CURRENT: Crowd status; Sibas to PLC
*7.5	CDSR	BOOL	FALSE	CROWD DRIVE READY TO START: Crowd status; Sibas to PLC
*7.6	CDMS	BOOL	FALSE	CROWD/PROPEL: DRIVE BRAKE SET: Crowd status; Sibas to PLC
*7.7	CDVSpd	BOOL	FALSE	CROWD Overspeed
*8.0	CDSON1	BOOL	FALSE	CROWD OPERATING MODE BIT 1: Crowd status; Sibas to PLC
*8.1	CDSON2	BOOL	FALSE	CROWD OPERATING MODE BIT 2: Crowd status; Sibas to PLC
*8.2	CDSON3	BOOL	FALSE	CROWD OPERATING MODE BIT 3: Crowd status; Sibas to PLC
*8.3	CDTF1	BOOL	FALSE	CROWD TEMP FAULT INV/1: Crowd status; Sibas to PLC
*8.4	CDTF2	BOOL	FALSE	CROWD TEMP FAULT INV/2: Crowd status; Sibas to PLC
*8.5	CDNP	BOOL	FALSE	CROWD NORMAL PARAMETER SET/DIG: Crowd status; Sibas to PLC
*8.6	CDAP	BOOL	FALSE	CROWD ALTERNATE PARAMETER SET/PROPEL: Crowd status; Sibas to PLC

El driver de comunicación es un script desarrollado en el lenguaje de programación Python e integrado a la interfaz de la plataforma de Ignition enlazándolo al OPC Server para el acceso a los datos.

El script al ejecutarse realiza las siguientes acciones:

Lee los bytes provenientes del PLC SIBAS, a través de la capa HTTP y los escribe en los tags de la ruta especificada para tal fin.

El método principal del driver es **parseShovelData** que recibe dos parámetros: La IP del SiBAS y la ruta en la que se escribirán los tags.

La información suministrada por el PLC viene en el siguiente formato:

ADDRESS_DEL_TAG 0xNUMERO_HEXADecimal

Donde ADDRESS_DEL_TAG es la Id del tag en el que se debe escribir el dato encontrado en la línea y 0Xnumero_HEXADecimal es un número hexadecimal cualquiera precedido por 0x. A continuación un ejemplo de cómo luciría un archivo del PLC:

HDSE 0x0

HDSD 0x2

HDSF 0x1

HDSW 0x98

El método `parseShovelData` lee el archivo de origen y obtiene de este una secuencia de cadenas de números hexadecimales, cada elemento representa un byte de información, esta secuencia se obtiene por medio del método auxiliar `readNumbers`.

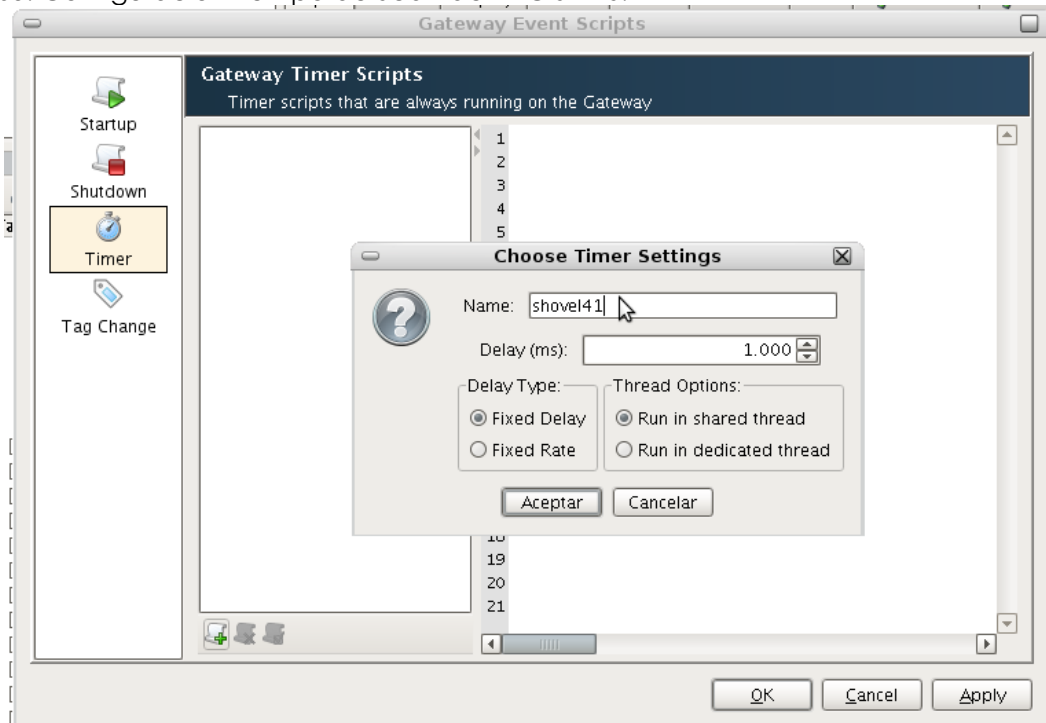
Se recorre la lista determinando el tipo de dato del valor basándose en la Id del tag en el que se va a escribir. Esto se logra al consultar en una lista que contrasta la Id del tag con el tipo de dato que este acepta, la lista se obtiene del método `getTags`.

El método `getTags` utiliza una estructura de datos que almacena los nombres y el tipo de dato de los Tags.

Habiendo determinado el tipo de dato, el valor es transformado al tipo adecuado para escribirlo en el tag.

Finalmente para mantener los tag actualizados, es decir, en tiempo es necesario realizar un scan de los datos del SiBAS ver Figura 65, para ello se activa un temporizador que permite la ejecución de forma cíclica del Script cada 500 milisegundos.

Figura 65. Configuración tiempo de Scan del PLC SiBAS.

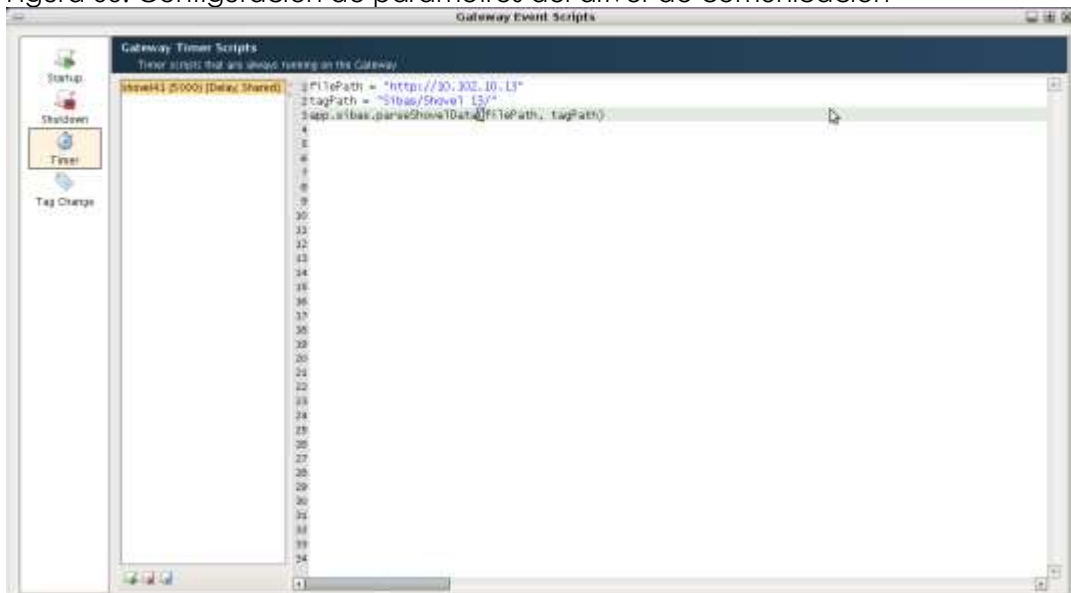


Para la ejecución del script se definen los siguientes parámetros como se observa en la figura 66:

filepath: Dirección IP del SIBAS de la Pala

tagPath: Ubicación de los Tags en la estructura de Ignition.

Figura 66. Configuración de parámetros del driver de comunicación



Cabe anotar que el temporizador del script se debe realizar 4 veces, uno para cada pala.

Ver Anexo 2: Código fuente del Driver de Comunicación.

5. CONCLUSIONES

Como resultado de este trabajo se obtiene el diseño e implementación de una plataforma SCADA que cumple con los requerimientos funcionales del sistema monitoreo de las palas Bucyrus.

El diseño realizado está basado en una potente plataforma de desarrollo como lo es Ignition. Debido a esto, el diseño es de fácil escalabilidad y más aún, permite que funcionalidades como la integración de nuevos protocolos, nuevos driver sea posible.

En cuanto a la implementación se realiza la comunicación entre el sistema SCADA y el PLC SiBAS a través del protocolo HTTP logrando leer 57 variables o Tags. Es importante destacar el desarrollo de muchas funciones en la plataforma de Ignition que facilitaron la implementación del driver de comunicación.

Se realizó una revisión bibliográfica del estado del arte en torno a sistemas SCADA. Esta revisión da sustento a lo planteado en este trabajo, puesto que las publicaciones referenciadas reflejan las discusiones actuales en torno al tema, y a la vez permitiendo seleccionar el software adecuado para la aplicación propuesta.

Siendo Ignition tecnología de software propietario pero desarrollada con herramientas de software libre, su costo sigue siendo supremamente inferior al costo del software del fabricante, siendo este último 25 veces más costoso, Drummond LTD planea un cambio 100% de la plataforma de software actual a Ignition, y cerrando el contrato de soporte técnico de siemens.

La incorporación de otras maquinarias como las dragalinas en el Gateway de Ignition es posible debido al potencial de este software y específicamente estas máquinas poseen tecnologías Allen Bradley y el OPC -UA Server de Ignition contiene los drivers gratuitos para la

comunicación, lo que significa que no habría un costo adicional por incluirlas .

6. RECOMENDACIONES Y TRABAJOS FUTUROS

Uno de los objetivos de este trabajo de tesis es diseñar la interfaz gráfica o sistema SCADA, en futuras versiones estarán disponibles otras funcionalidades asociadas a un sistema de mantenimiento preventivo entre ellas generación de órdenes de trabajo, el acceso a la Base de datos de almacén (Peoplesoft) para generar requisiciones de materiales y repuestos, etc.

A corto plazo se iniciará la integración de 3 dragalinas, eso implica que el flujo de información y número de clientes será mayor, por tanto es necesario aumentar la capacidad de la memoria RAM del servidor, debido a que los clientes de Ignition hacen uso de esta memoria. |

Esperamos contar con el apoyo de la UNAB para darle continuidad al desarrollo de este proyecto, en la actualidad existe una herramienta del fabricante llamada SiBAS Monitor, funciona solo en Windows y el protocolo es RS232, con ella se puede acceder a la totalidad de la memoria de programa del controlador de movimiento o PLC SiBAS, esperamos desarrollar un segundo driver que nos permita la visualización en línea de todos los datos del SiBAS.

BIBLIOGRAFIA

- [1] http://www.exif.cz/tde/open_drive/afe-tde.pdf
- [2] http://www3.sea.siemens.com/step/templates/lesson.mason?ac_drives:4:1:8
- [3] WOLFGANG, Mahnke. OPC Unified Architecture. 1ªed. Germany. Editorial Springer, 2009.337p. ISBN 978-3-540-8898-3.
- [4] <http://lintouch-ng.sourceforge.net/>
- [5] <http://www.cintal.com.ve/tecnologia/argos/>
- [6] <http://www.freescada.com/>
- [7] <http://www.inductiveautomation.com>
- [8] Pala para minería. Manual de mantención & Operación. Revisión de 2010. Wisconsin, USA: Bucyrus International.
- [9] <http://www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/controladores/Pages/S7400.aspx>
- [10] <http://www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/controladores/Pages/S7400.aspx>
- [11] http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/13_es_red_es_mesh_guia_v02.pdf
- [12] <http://www.motorola.com/>
- [13] www.automation.siemens.com
- [14] RODRÍGUEZ, Penin, Aquilino. Sistemas SCADAS. 2ª ed. Barcelona. Editorial Marcombo, 2011. 448 p. ISBN: 8426714501.
- [15] Ignition, HMI, SCADA. User manual. Revisión de 2012. Folsom, USA: Inductive Automation.
- [16] <http://www.coplec.org>
- [17] <http://www.epsevg.upc.edu/hcd/material/lecturas/pantalla.pdf>
- [18] SCHMULLER, Joseph. Aprendiendo UML en 24 Horas.3ª ed. México. Editorial Prentice Hall, 2010.404p.

ANEXO 1

Address	Name	Type	Comment
0	HDSE	BOOL	HOIST DRIVE ON: Hoist status; SiBAS to PLC
0.1	HDSO	BOOL	HOIST DRIVE OFF: Hoist status; SiBAS to PLC
0.2	HDSF	BOOL	HOIST DRIVE FAULT: Hoist status; SiBAS to PLC
0.3	HDSW	BOOL	HOIST DRIVE WARNING: Hoist status; SiBAS to PLC
0.4	HDMC	BOOL	HOIST MINIMUM CURRENT: Hoist status; SiBAS to PLC
0.5	HDSR	BOOL	HOIST DRIVE READY TO START: Hoist status; SiBAS to PLC
0.6	HDBS	BOOL	HOIST/PROPEL#1 DRIVE BRAKE SET: Hoist status; SiBAS to PLC
0.7	HOvrSpd	BOOL	Hoist Overspeed
1	HDSOM1	BOOL	HOIST OPERATING MODE BIT 1: Hoist status; SiBAS to PLC
1.1	HDSOM2	BOOL	HOIST OPERATING MODE BIT 2: Hoist status; SiBAS to PLC
1.2	HDSOM3	BOOL	HOIST OPERATING MODE BIT 3: Hoist status; SiBAS to PLC
1.3	HDTFI1	BOOL	HOIST TEMP FAULT INV/1: Hoist status; SiBAS to PLC
1.4	HDTFI2	BOOL	HOIST TEMP FAULT INV/2: Hoist status; SiBAS to PLC
1.5	HDNP	BOOL	HOIST NORMAL PARAMETER SET/DIG: Hoist status; SiBAS to PLC
1.6	HDAP	BOOL	HOIST ALTERNATE PARAMETER SET/PROPEL: Hoist status;
1.7	HDN5	BOOL	HOIST SPEED < 2.5Hz (5%): Hoist status; SiBAS to PLC
2	HDNF	BYTE	HOIST SPEED FEEDBACK: Hoist status; SiBAS to PLC
3	HDTF	BYTE	HOIST TORQUE FEEDBACK: Hoist status; SiBAS to PLC
4	HDCF	BYTE	HOIST CURRENT FEEDBACK: Hoist status; SiBAS to PLC
5	HDACF	BYTE	HOIST ACTIVE CURRENT FEEDBACK: Hoist status;
6	MS_HOIST	BYTE	HOIST Master Switch Scaled
7	CDSE	BOOL	CROWD DRIVE ON: Crowd status; SiBAS to PLC
7.1	CDSO	BOOL	CROWD DRIVE OFF: Crowd status; SiBAS to PLC
7.2	CDSF	BOOL	CROWD DRIVE FAULT: Crowd status; SiBAS to PLC
7.3	CDSW	BOOL	CROWD DRIVE WARNING: Crowd status; SiBAS to PLC
7.4	CDMC	BOOL	CROWD MINIMUM CURRENT: Crowd status;
7.5	CDSR	BOOL	CROWD DRIVE READY TO START: Crowd status;
7.6	CDBS	BOOL	CROWD/PROPEL#2 DRIVE BRAKE SET: Crowd status;
7.7	COvrSpd	BOOL	CROWD Overspeed
8	CDSOM1	BOOL	CROWD OPERATING MODE BIT 1: Crowd status; SiBAS to PLC
8.1	CDSOM2	BOOL	CROWD OPERATING MODE BIT 2: Crowd status; SiBAS to PLC

8.2	CDSOM3	BOOL	CROWD OPERATING MODE BIT 3: Crowd status; SiBAS to PLC
8.3	CDTF1	BOOL	CROWD TEMP FAULT INV/1: Crowd status; SiBAS to PLC
8.4	CDTF2	BOOL	CROWD TEMP FAULT INV/2: Crowd status; SiBAS to PLC
8.5	CDNP	BOOL	CROWD NORMAL PARAMETER SET/DIG: Crowd status;
8.6	CDAP	BOOL	CROWD ALTERNATE PARAMETER SET/PROPEL: Crowd status;
8.7	CDN5	BOOL	CROWD SPEED < 2.5Hz (5%): Crowd status; SiBAS to PLC
9	CDNF	BYTE	CROWD SPEED FEEDBACK: Crowd status; SiBAS to PLC
10	CDTF	BYTE	CROWD TORQUE FEEDBACK: Crowd status; SiBAS to PLC
11	CDCF	BYTE	CROWD CURRENT FEEDBACK: Crowd status; SiBAS to PLC
12	CDACF	BYTE	CROWD ACTIVE CURRENT FEEDBACK: Crowd status;
13	MS_CRD	BYTE	CROWD Master Switch Scaled
14	SDSE	BOOL	SWING DRIVE ON: Swing status; SiBAS to PLC
14.1	SDSD	BOOL	SWING DRIVE OFF: Swing status; SiBAS to PLC
14.2	SDSF	BOOL	SWING DRIVE FAULT: Swing status; SiBAS to PLC
14.3	SDSW	BOOL	SWING DRIVE WARNING: Swing status; SiBAS to PLC
14.4	SDMC	BOOL	SWING MINIMUM CURRENT: Swing status; SiBAS to PLC
14.5	SDSR	BOOL	SWING DRIVE READY TO START: Swing status; SiBAS to PLC
14.6	SDBS	BOOL	SWING DRIVE BRAKE SET: Swing status; SiBAS to PLC
14.7	SOvrSpd	BOOL	Swing Overspeed
15	SDSOM1	BOOL	SWING OPERATING MODE BIT 1: Swing status; SiBAS to PLC
15.1	SDSOM2	BOOL	SWING OPERATING MODE BIT 2: Swing status; SiBAS to PLC
15.2	SDSOM3	BOOL	SWING OPERATING MODE BIT 3: Swing status; SiBAS to PLC
15.3	SDTF1	BOOL	SWING TEMP FAULT INV/1: Swing status; SiBAS to PLC
15.4	SDTF2	BOOL	SWING TEMP FAULT INV/2: Swing status; SiBAS to PLC
15.5	SDNP	BOOL	SWING NORMAL PARAMETER SET/DIG: Swing status; SiBAS to
15.6	SDAP	BOOL	SWING NORMAL PARAMETER SET/PROPEL: Swing status; SiBAS to PLC
15.7	SDN5	BOOL	SWING SPEED < 2.5Hz (5%): Swing status; SiBAS to PLC
16	SDNF	BYTE	SWING SPEED FEEDBACK: Swing status; SiBAS to PLC
17	SDTF	BYTE	SWING TORQUE FEEDBACK: Swing status; SiBAS to PLC
18	SDCF	BYTE	SWING CURRENT FEEDBACK: Swing status; SiBAS to PLC
19	SDACF	BYTE	SWING ACTIVE CURRENT FEEDBACK: Swing status; SiBAS to PLC
20	MS_SWG	BYTE	SWING Master Switch Scaled
21	AFEE	BOOL	AFE ON: AFE status; SiBAS to PLC
21.1	AFER	BOOL	AFE READY: AFE status; SiBAS to PLC
21.2	AFEF	BOOL	AFE FAULT: AFE status; SiBAS to PLC
21.3	AFEW	BOOL	AFE WARNING: AFE status; SiBAS to PLC
21.4	BPCE	BOOL	BUS PRECHARGED: AFE status; SiBAS to PLC
21.5	DPCE	BOOL	DPC CLOSED: AFE status; SiBAS to PLC
21.6	CHCNZ	BOOL	CHOPPER CURRENT > 0: AFE status; SiBAS to PLC

21.7	AFETF	BOOL	AFE TEMPERATURE FAULT: AFE status; SiBAS to PLC
22	AFEF1	BOOL	AFE FAULT #1: AFE status; SiBAS to PLC
22.1	AFEF2	BOOL	AFE FAULT #2: AFE status; SiBAS to PLC
22.2	AFEF3	BOOL	AFE FAULT #3: AFE status; SiBAS to PLC
22.3	AFEF4	BOOL	AFE FAULT #4: AFE status; SiBAS to PLC
22.4	CH1TF	BOOL	CHOPPER 1 TEMP FAULT: AFE status; SiBAS to PLC
22.5	CH2TF	BOOL	CHOPPER 2 TEMP FAULT: AFE status; SiBAS to PLC
22.6	CH1F	BOOL	CHOPPER 1 FAULT: AFE status; SiBAS to PLC
22.7	CH2F	BOOL	CHOPPER 2 FAULT: AFE status; SiBAS to PLC
23	CH3TF	BOOL	CHOPPER 3 TEMP FAULT: AFE status; SiBAS to PLC
23.1	CH4TF	BOOL	CHOPPER 4 TEMP FAULT: AFE status; SiBAS to PLC
23.2	CH3F	BOOL	CHOPPER 3 FAULT: AFE status; SiBAS to PLC
23.3	CH4F	BOOL	CHOPPER 4 FAULT: AFE status; SiBAS to PLC
23.4	HIV	BOOL	HIGH LINE INPUT VOLTAGE FAULT: AFE status; SiBAS to PLC
23.5	LIV	BOOL	LOW LINE INPUT VOLTAGE FAULT: AFE status; SiBAS to PLC
23.6	LLW	BOOL	LOW LINE INPUT WARNING: AFE status; SiBAS to PLC
23.7	PFnotREF	BOOL	POWER FACTOR NOT EQUAL PF REFERENCE: AFE status;
24	DCVF	BYTE	DCV BUS FEEDBACK: AFE status; SiBAS to PLC
25	DCIF	BYTE	DCI BUS FEEDBACK: AFE status; SiBAS to PLC
26	VLINE	BYTE	LINE VOLTAGE: AFE status; SiBAS to PLC
27	PF	BYTE	POWER FACTOR: AFE status; SiBAS to PLC
28	BROM1	BOOL	OPERATING MODE BIT 1: AFE status; SiBAS to PLC
28.1	BROM2	BOOL	OPERATING MODE BIT 2: AFE status; SiBAS to PLC
28.2	BROM3	BOOL	OPERATING MODE BIT 3: AFE status; SiBAS to PLC
28.3	SEQBIT1	BOOL	SEQUENCE BIT #1: AFE status; SiBAS to PLC
28.4	SEQBIT2	BOOL	SEQUENCE BIT #2: AFE status; SiBAS to PLC
28.5	SEQBIT3	BOOL	SEQUENCE BIT #3: AFE status; SiBAS to PLC
28.6	DGF	BOOL	DRIVES SYSTEM GROUND FAULT: AFE status; SiBAS to PLC
28.7	AFEGB	BOOL	AFE GATE BLOCKED: AFE status; SiBAS to PLC
29	HPGB	BOOL	HOIST GATE BLOCKED: General status; SiBAS to PLC
29.1	CGB	BOOL	CROWD GATE BLOCKED: General status; SiBAS to PLC
29.2	SGB	BOOL	SWING GATE BLOCKED: General status; SiBAS to PLC
29.3	HTF	BOOL	HOIST TACH FAULT: General status; SiBAS to PLC
29.4	HTW	BOOL	HOIST TACH WARNING: General status; SiBAS to PLC
29.5	CTF	BOOL	CROWD TACH FAULT: General status; SiBAS to PLC
29.6	CTW	BOOL	CROWD TACH WARNING: General status; SiBAS to PLC
29.7	STF	BOOL	SWING TACH FAULT: General status; SiBAS to PLC
30	STW	BOOL	SWING TACH WARNING: General status; SiBAS to PLC
30.1	HPREGEN	BOOL	HOIST IN REGENERATION: Hoist status; SiBAS to PLC

30.2	CPREGEN	BOOL	HOIST IN REGENERATION: Hoist status; SiBAS to PLC
30.3	SPREGEN	BOOL	HOIST IN REGENERATION: Hoist status; SiBAS to PLC
30.4	Sibas13	BOOL	SIBAS I/O BIT 13: General status; SiBAS to PLC
30.5	Sibas14	BOOL	SIBAS I/O BIT 14: General status; SiBAS to PLC
30.6	Sibas15	BOOL	SIBAS I/O BIT 15: General status; SiBAS to PLC
30.7	Sibas16	BOOL	SIBAS I/O BIT 16: General status; SiBAS to PLC
31	GNDFLTV	BYTE	GROUND FAULT CIRCUIT VOLTAGE FROM SIBAS
32	PELEKZH	BYTE	Hoist Power at Terminals PWRH
33	spare1	BYTE	Spare
34	H_Number	BYTE	Hoist Parameter Number
35	HRMSI	BYTE	Hoist RMS Current
36	ISYWRZH	BYTE	Torque Building Current PWRH
37	UNISTZH	BYTE	Hoist - Line Voltage PWRH
38	PELEKZC	BYTE	Crowd - Power at Terminals PWRC
39	HDPARCB	BOOL	Checkback HOIST in Default
39.1	CDPARCB	BOOL	Checkback CROWD in Default
39.2	SDPARCB	BOOL	Checkback SWING in Default
39.3	AFPARCB	BOOL	Checkback AFE in Default
39.4	Sparebit01	BOOL	Default AFE Parameters in Progress
39.5	Sparebit02	BOOL	
39.6	Sparebit03	BOOL	
39.7	Sparebit04	BOOL	
40	C_NUMBER	BYTE	Crowd Parameter Number
41	CRMSI	BYTE	Crowd RMS Current
42	ISYWRZC	BYTE	Crowd - Torque Building Current
43	UNISTZC	BYTE	Crowd - Line Voltage
44	PELEKZS	BYTE	Swing - Power at terminals
45	GFTOMVB	BYTE	Ground Fault Info to Comm.
46	S_NUMBER	BYTE	Swing Parameter Number
47	IL1PWRS	BYTE	Swing - Phase Current L1
48	MUXCOUNT	BYTE	Temperature MUX Count
49	TEMPVALS	BYTE	Black Temperature Value
50	PFIST	BYTE	Actual Power Factor
51	ACTPSUM	BYTE	Active Power Sum
52	AFE_Number	BYTE	AFE - Parameter Number
53	fix_0_A	BYTE	Fix Value 0 A
54	XDIAGNR_H	INT	Diagnostic Code
56	spare	BYTE	
57	spare2	BYTE	

ANEXO 2

Código Fuente de Driver de Comunicación

```
global app, system, Integer, Byte
import app, system
from java.lang import Integer
from java.lang import Byte

def parseShovelData(filePath, tagPath):
    import time, sys

    bytes = app.sibas.readNumbers(filePath)
    tags = app.sibas.getTags()
    tagNum = 0
    byteNum = 0
    while tagNum < len(tags) and byteNum < len(bytes):
        if tags[tagNum][1] == "BYTE":
            try:
                system.tag.writeToTag(tagPath+tags[tagNum][0],
app.sibas.unsignedToSigned(Integer.parseInt(bytes[byteNum], 16)))

            except:
                pass
            tagNum += 1
            byteNum += 1
        elif tags[tagNum][1] == "BOOL":
            binNum = app.sibas.bin(bytes[byteNum]).zfill(8)
            for i in range(8):
                try:
                    system.tag.writeToTag(tagPath+tags[tagNum][0],
binNum[i])

                except:
                    pass
                tagNum += 1
                byteNum += 1
        elif tags[tagNum][1] == "INT":
            byte1 = bytes[byteNum]
            byteNum += 1
            byte2 = bytes[byteNum]
            print "byte1:", byte1, "byte2:", byte2
            try:
                system.tag.writeToTag(tagPath+tags[tagNum][0],
(Integer.parseInt(byte1, 16)*256+(Integer.parseInt(byte2, 16)))
                #system.tag.writeToTag(tagPath+tags[tagNum][0],
Byte.parseByte(byte1).intValue())
            except:
                pass
            tagNum += 1
```

```

        byteNum += 1
    time = time.localtime()
    timestring = "%s-%2d-%2d %2d:%2d:%2d" % (time[0], time[1], time[2], time[3], time[4],
time[5])
    system.tag.writeToTag("%s_Last_Update"%tagPath, timestring)

def readNumbers(filePath):
    if filePath[0:4] == "http":
        file = app.sibas.httpGet(filePath, 3000)
    else:
        file = system.file.readFileAsString(filePath)
    nums = []
    pos = 0
    while 1:
        pos = file.find("0x",pos)
        if pos == -1:
            break
        pos = pos + 2
        nums.append(file[pos:pos+2])
    return nums

def bin(s):
    return Integer.toBinaryString(Integer.parseInt(s,16))

def getTags():
    tags = [{"HDSE","BOOL"},
["HDS D","BOOL"},
["HDS F","BOOL"},
["HDS W","BOOL"},
["HDM C","BOOL"},
["HDS R","BOOL"},
["HDB S","BOOL"},
["HOvrSpd","BOOL"},
["HDSOM1","BOOL"},
["HDSOM2","BOOL"},
["HDSOM3","BOOL"},
["HDTFI1","BOOL"},
["HDTFI2","BOOL"},
["HDNP","BOOL"},
["HDAP","BOOL"},
["HDN5","BOOL"},
["HDNF","BYTE"],
["HDTF","BYTE"],
["HDCF","BYTE"],
["HDACF","BYTE"],
["MS_HOIST","BYTE"],
["CDSE","BOOL"],
["CDS D","BOOL"],
["CDS F","BOOL"],
["CDS W","BOOL"],

```


["CDMC","BOOL"],
["CDSR","BOOL"],
["CDBS","BOOL"],
["COvrSpd","BOOL"],
["CDSOM1","BOOL"],
["CDSOM2","BOOL"],
["CDSOM3","BOOL"],
["CDTF1","BOOL"],
["CDTF2","BOOL"],
["CDNP","BOOL"],
["CDAP","BOOL"],
["CDN5","BOOL"],
["CDNF","BYTE"],
["CDTF","BYTE"],
["CDCF","BYTE"],
["CDACF","BYTE"],
["MS_CRD","BYTE"],
["SDSE","BOOL"],
["SDSD","BOOL"],
["SDSF","BOOL"],
["SDSW","BOOL"],
["SDMC","BOOL"],
["SDSR","BOOL"],
["SDBS","BOOL"],
["SOvrSpd","BOOL"],
["SDSOM1","BOOL"],
["SDSOM2","BOOL"],
["SDSOM3","BOOL"],
["SDTF1","BOOL"],
["SDTF2","BOOL"],
["SDNP","BOOL"],
["SDAP","BOOL"],
["SDN5","BOOL"],
["SDNF","BYTE"],
["SDTF","BYTE"],
["SDCF","BYTE"],
["SDACF","BYTE"],
["MS_SWG","BYTE"],
["AFEE","BOOL"],
["AFER","BOOL"],
["AFEF","BOOL"],
["AFEW","BOOL"],
["BPCE","BOOL"],
["DPCE","BOOL"],
["CHCNZ","BOOL"],
["AFETF","BOOL"],
["AFEF1","BOOL"],
["AFEF2","BOOL"],
["AFEF3","BOOL"],
["AFEF4","BOOL"],

["CH1TF","BOOL"],
["CH2TF","BOOL"],
["CH1F","BOOL"],
["CH2F","BOOL"],
["CH3TF","BOOL"],
["CH4TF","BOOL"],
["CH3F","BOOL"],
["CH4F","BOOL"],
["HIV","BOOL"],
["LIV","BOOL"],
["LLW","BOOL"],
["PFnotREF","BOOL"],
["DCVF","BYTE"],
["DCIF","BYTE"],
["VLIN","BYTE"],
["PF","BYTE"],
["BROM1","BOOL"],
["BROM2","BOOL"],
["BROM3","BOOL"],
["SEQBIT1","BOOL"],
["SEQBIT2","BOOL"],
["SEQBIT3","BOOL"],
["DGF","BOOL"],
["AFEGB","BOOL"],
["HPGB","BOOL"],
["CGB","BOOL"],
["SGB","BOOL"],
["HTF","BOOL"],
["HTW","BOOL"],
["CTF","BOOL"],
["CTW","BOOL"],
["STF","BOOL"],
["STW","BOOL"],
["HPREGEN","BOOL"],
["CPREGEN","BOOL"],
["SPREGEN","BOOL"],
["Sibas13","BOOL"],
["Sibas14","BOOL"],
["Sibas15","BOOL"],
["Sibas16","BOOL"],
["GNDFLTV","BYTE"],
["PELEKZH","BYTE"],
["spare1","BYTE"],
["H_Number","BYTE"],
["HRMSI","BYTE"],
["ISYWRZH","BYTE"],
["UNISTZH","BYTE"],
["PELEKZC","BYTE"],
["HDPARCB","BOOL"],
["CDPARCB","BOOL"],

```

["SDPARCB","BOOL"],
["AFPARCB","BOOL"],
["Sparebit01","BOOL"],
["Sparebit02","BOOL"],
["Sparebit03","BOOL"],
["Sparebit04","BOOL"],
["C_NUMBER","BYTE"],
["CRMSI","BYTE"],
["ISYWRZC","BYTE"],
["UNISTZC","BYTE"],
["PELEKZS","BYTE"],
["GFTOMVB","BYTE"],
["S_NUMBER","BYTE"],
["IL1PWRS","BYTE"],
["MUXCOUNT","BYTE"],
["TEMPVALS","BYTE"],
["PFIST","BYTE"],
["ACTPSUM","BYTE"],
["AFE_Number","BYTE"],
["fix_0_A","BYTE"],
["XDIAGNR_H","INT"]
return tags

```

```

def httpGet(address, timeout):
    import sys
    sys.add_package("org.apache.commons.io")
    from java.net import HttpURLConnection, URL
    from org.apache.commons.io import IOUtils
    from java.io import IOException

    try:
        url = URL(address)
        con = url.openConnection()

        try:
            con.setRequestMethod("GET")
        except ProtocolException:
            pass

        con.setDoOutput(0)
        con.setDoInput(1)

        con.setConnectTimeout(timeout)
        con.setReadTimeout(timeout)

        data = IOUtils.toString(con.getInputStream(), None)

        con.disconnect();

    return data;

```

```
    except IOError:
        raise

def unsignedToSigned(intVal):
    if intVal > 127:
        intVal = 0 - (256 - intVal)
    return intVal
```

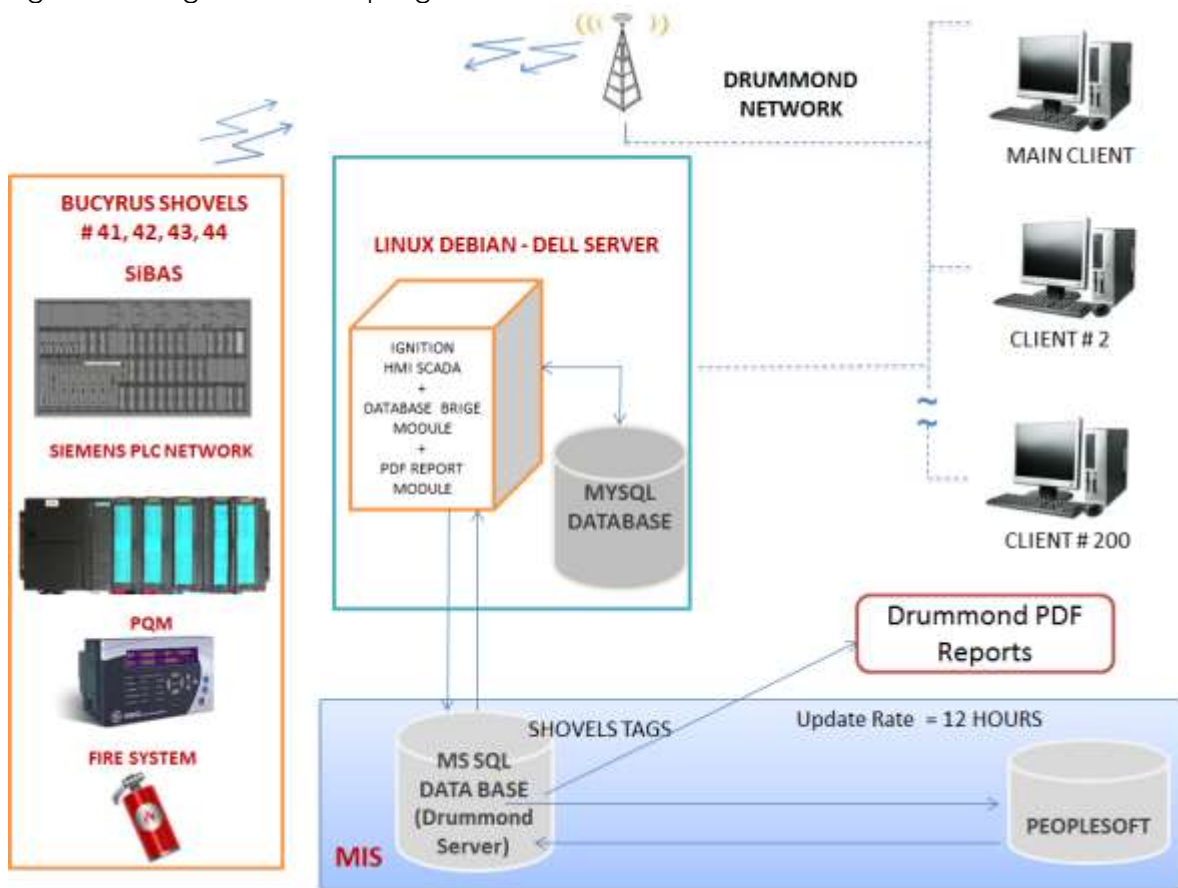
ANEXO 3

MANUAL DEL USUARIO

En este Anexo se presentará un manual de usuario de la aplicación más detallado para el uso correcto de la aplicación.

Antes de iniciar la construcción del software se realizó un diagrama de despliegue, donde se indica la interconectividad del software con sus diferentes herramientas de trabajo que influyen en la lectura, escritura, manejo de la información, y manejo de las comunicaciones como se indica, figura 67:

Figura 67. Diagrama de Despliegue



Como es de notar en la figura 67, Ignition realiza sus comunicaciones con 4 máquinas al tiempo, cada una de ellas posee cuatro dispositivos o

controladores de los cuales se obtiene información en línea del estado de la máquina.

La primera pantalla que aparecerá será la que se muestra en la Figura 68

Figura 68. Pantalla inicial del sistema SCADA



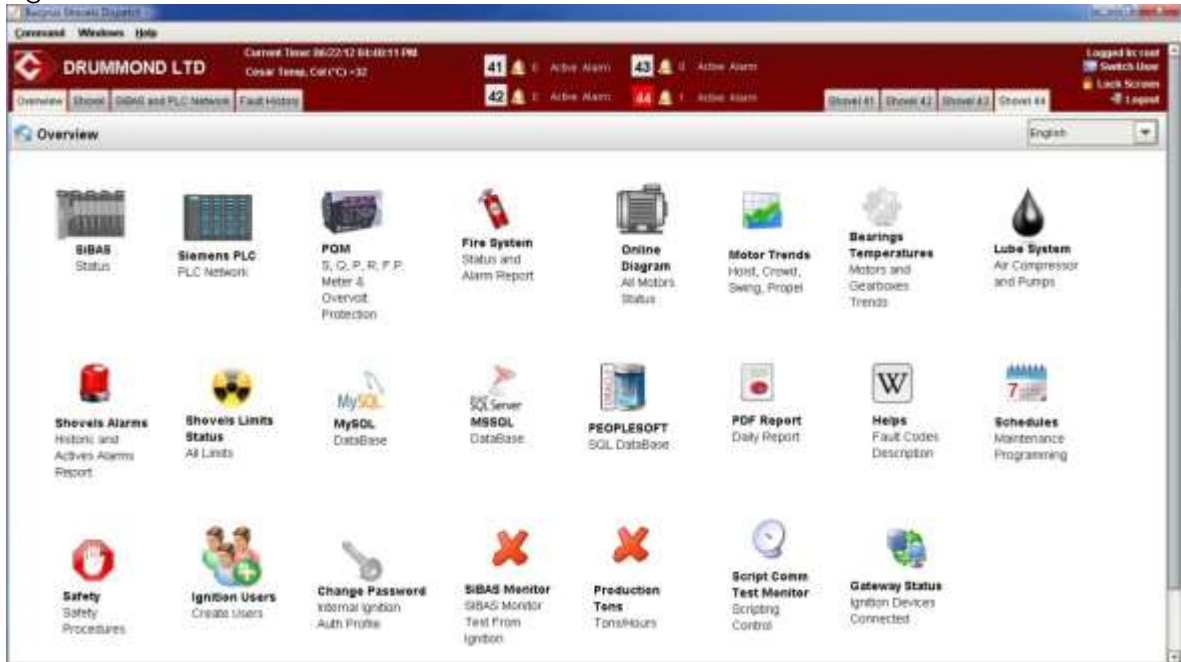
Debido a la semejanza de cada máquina, se construyó un menú de selección o switch encargado de conmutar sobre una mismo objeto, la información proveniente de la maquina seleccionada, así como el estado (refiriéndose a posición o cambio en el tiempo de una variable), este menú se ubica al lado derecho de la ventana principal y se muestra en la Figura 69.

Figura 69. Botones para conmutar



Durante la construcción del software siempre se tuvo a consideración tener acceso múltiple desde cualquier ubicación, sin olvidar de que existe una ruta principal, desde la cual se puede tener una vista general (ver figura 70) de todo el sistema SCADA.

Figura 70. Acceso a funcionalidades del sistema SCADA



La vista general de la figura 70 es llamada Overview, contiene una serie de iconos que brindan al usuario o cliente de la aplicación un sin número de facilidades para acceder a los diferentes acciones que se pueden realizar en el sistema SCADA, divididas por diferentes secciones.

A continuación, se especificarán las acciones posibles seleccionables a través del menú de la pantalla Overview.

Red de PLC siemens, desde el icono Siemens PLC (PLC Network), Figura 71

Figura 71. Red de PLC

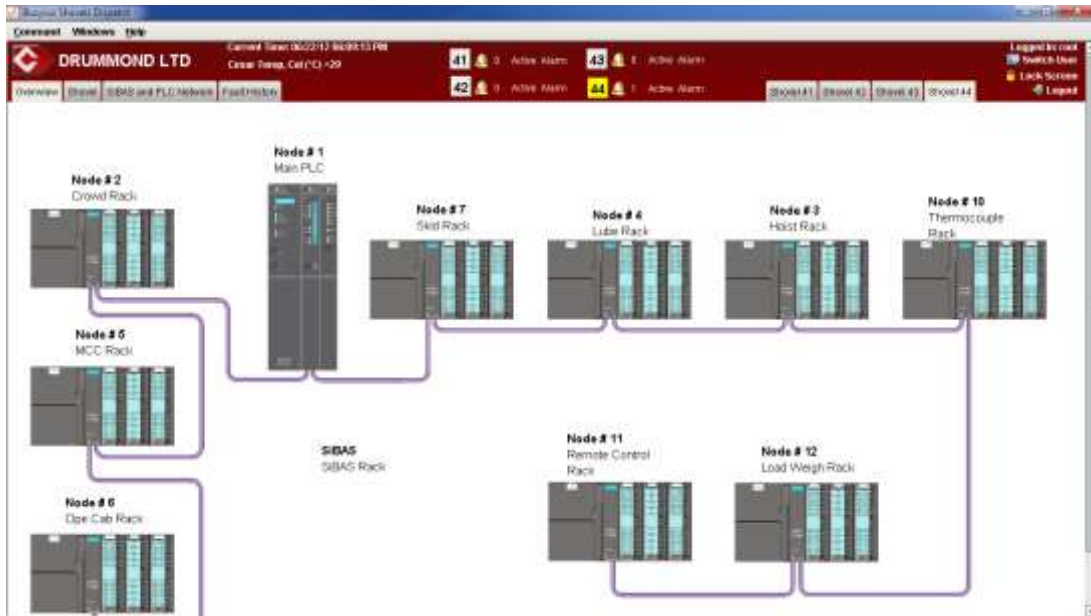
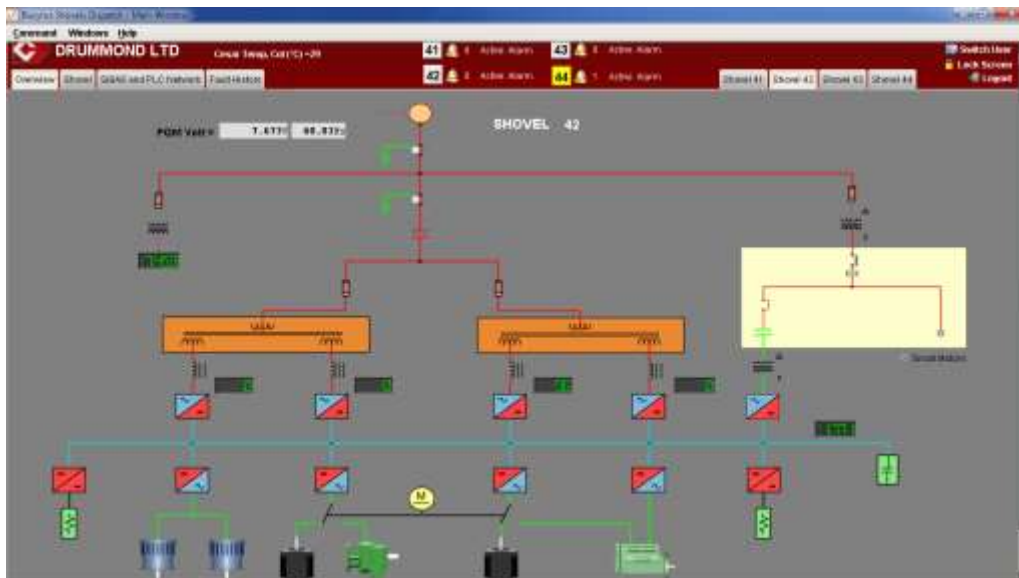


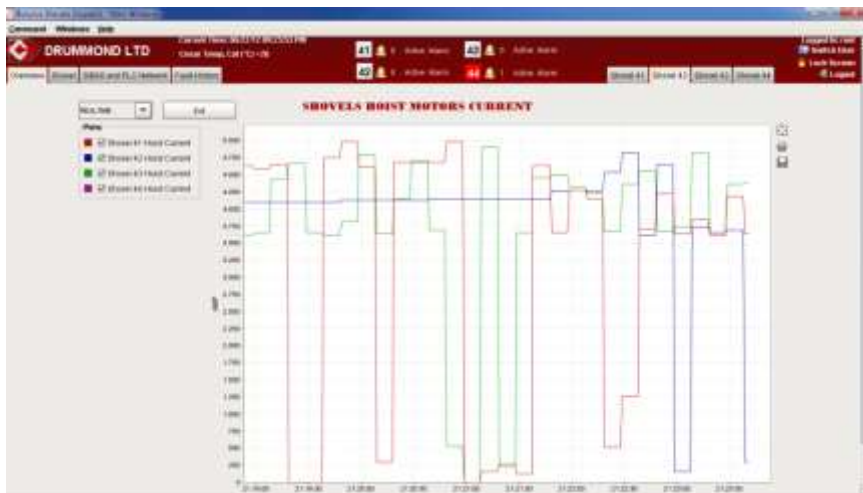
Diagrama Unifilar, se accede desde el icono Online Diagram (All Motors Status). Ver Figura 72

Figura 72. Diagrama Unifilar



Para visualizar Tendencias en tiempo real e Histórico, hacer clic en el Icono Motor Trends, Ver Figura 73

Figura 73. Tendencias



Para la planificación de tareas de mantenimiento se debe hacer clic en el Icono Schedules, esta opción permite crear y borrar eventos, también tiene la opción de imprimir el calendario de eventos. Ver figura 74

Figura 74. Calendario de eventos



La aplicación permite crear nuevos usuarios y realizar cambio de password para el usuario, desde el icono Change Password.

Figura 75. Cambio de Password



Histórico de Fallas y/o Fault History: Esta opción se puede visualizar desde la pantalla del Overview o desde la Ficha Fault History.

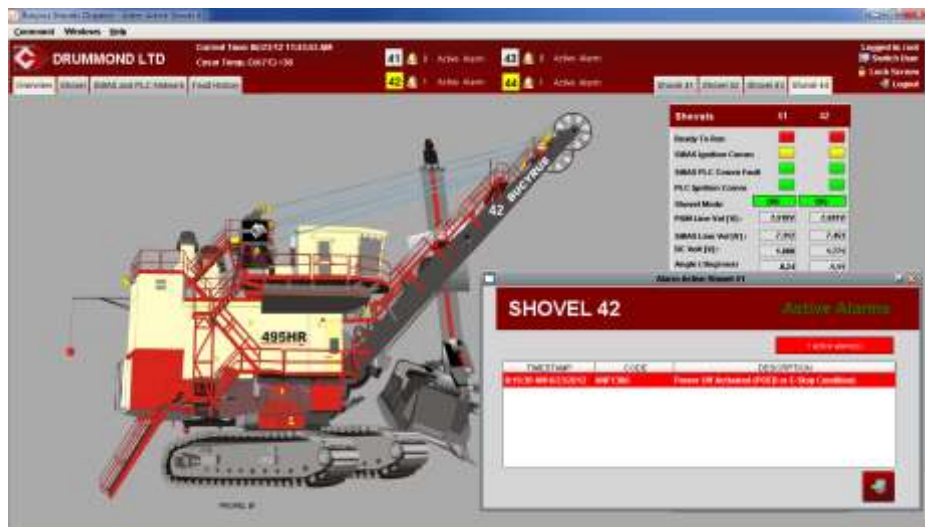
Figura 76. Histórico de Alarmas



El histórico tiene la opción, de seleccionar cualquier alarma dentro del histórico, y acceder a la descripción de la misma en un archivo pdf que detalla las posibles soluciones a la falla descrita.

Alarmas Activas: Cada pala tiene una indicación de falla en el Docked , para visualizarla se debe hacer clic en el icono respectivo a cada pala, sin embargo este icono cambian de color para indicar que hay una falla presente en la máquina.

Figura 77. Alarmas activas.



Ejemplo: en la imagen de la figura 77, se observa la pala 42 en posición de reposo, debido a un mantenimiento mecánico, como alarma, se tiene una parada de emergencia accionada, la cual detiene todos los movimientos, siendo parte de los bloqueos y procedimientos de seguridad antes de iniciar una actividad.

ANEXO 4

En el anexo 3 se muestra el script utilizado para la creación de tablas

```
CREATE TABLE `mysql`.`shovel` (  
  `asset` VARCHAR(20) NOT NULL ,  
  `name` VARCHAR(45) NULL ,  
  PRIMARY KEY (`asset` ) );
```

```
.....  
  
CREATE TABLE `mysql`.`ALERT_LOG` (  
  `ALERT_LOG_NDX` INT NOT NULL AUTO_INCREMENT ,  
  `ACK_TIMESTAMP` DATETIME NULL ,  
  `ACK_USER` VARCHAR(45) NULL ,  
  `ACTIVE_TIMESTAMP` DATETIME NULL ,  
  `ACTIVE_VALUE` DOUBLE NULL ,  
  `CLEARED_TIMESTAMP` DATETIME NULL ,  
  `CLEARED_VALUE` DOUBLE NULL ,  
  `DISPLAYPATH` VARCHAR(45) NULL ,  
  `PATH` VARCHAR(45) NULL ,  
  `STATE_NAME` VARCHAR(45) NULL ,  
  `STATE_SEVERITY` INT NULL ,
```

```
`SYSTEM` VARCHAR(45) NULL ,  
`Asset` VARCHAR(45) NULL ,  
PRIMARY KEY (`ALERT_LOG_NDX`),  
INDEX `Assett` (`Asset` ASC);
```

```
CREATE TABLE `mysql`.`schedule` (  
  `idschedule` INT NOT NULL AUTO_INCREMENT ,  
  `display` VARCHAR(45) NULL ,  
  `displayColor` VARCHAR(45) NULL ,  
  `endDate` VARCHAR(45) NOT NULL ,  
  `startDate` VARCHAR(45) NOT NULL ,  
  `Asset` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`idschedule`),  
  INDEX `Assett` (`Asset` ASC));
```

```
CREATE TABLE `mysql`.`spare_code` (  
  `idspare_code` INT NOT NULL ,  
  `Manufacturer` VARCHAR(45) NULL ,  
  `Description` VARCHAR(200) NULL ,
```

```
`Current_min` INT NULL ,  
`Current_Max` INT NULL ,  
`SOH` INT NULL ,  
`Part_No` VARCHAR(45) NULL ,  
`Asset` VARCHAR(45) NOT NULL ,  
PRIMARY KEY (`idspare_code`),  
INDEX `Asset` (`Asset` ASC) );
```

```
CREATE TABLE `mysql`.`Shift_Report` (  
  `idShift_Report` INT NOT NULL AUTO_INCREMENT ,  
  `Alarma` VARCHAR(200) NULL ,  
  `Comments` VARCHAR(200) NULL ,  
  `Description` VARCHAR(200) NULL ,  
  `startDate` DATETIME NULL ,  
  `endDate` DATETIME NULL ,  
  `Asset` VARCHAR(45) NULL ,  
  PRIMARY KEY (`idShift_Report`),  
  INDEX `Asset` (`Asset` ASC) );
```

```
CREATE TABLE `mysql`.`users` (  
  `username` VARCHAR(45) NOT NULL ,  
  `password` VARCHAR(45) NULL ,  
  PRIMARY KEY (`username` ) );
```

.....

```
CREATE TABLE `mysql`.`Roles` (  
  `Rolename` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`Rolename` ) );
```

.....

```
CREATE TABLE `mysql`.`USER_ROLE_MAPPING` (  
  `Rolename` VARCHAR(45) NOT NULL ,  
  `Username` VARCHAR(45) NOT NULL ,  
  INDEX `Username` (`Username` ASC) );
```