

INTEROPERABILIDAD OPC PARA INTEGRACION DE SISTEMAS SCADA



**Álvaro Antonio Patiño
Dídier Restrepo Bravo**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN EN AUTOMATICA INDUSTRIAL
POPAYÁN
2007**

INTEROPERABILIDAD OPC PARA INTEGRACION DE SISTEMAS SCADA



**Monografía presentada como requisito parcial para optar al título de Ingenieros en
Automática Industrial**

**Álvaro Antonio Patiño
Dídier Restrepo**

**Director
Germán Mauricio Coral
Ingeniero en Electrónica y Telecomunicaciones**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN EN AUTOMATICA INDUSTRIAL
POPAYÁN
2007**

Nota de Aceptación

Director

Ing. Germán Mauricio Coral

Jurado _____
Ing.

Jurado _____
Ing.

Fecha de sustentación: Popayán, 2007

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Ing. Germán Mauricio Coral

Ing. Oscar Amaury Rojas

Ing. Carlos Cobos

Y demás personas que colaboraron con el desarrollo del presente trabajo.

*A Jesucristo mi Señor y Salvador, a mi compañero de trabajo, a mi familia, a mis amigos,
profesores y a la Universidad del Cauca*

DIDIER

*A Dios, a mi compañero de trabajo, a mi familia, a mis amigos y profesores y a la
Universidad del Cauca
ALVARO.*

CONTENIDO

| | | |
|--------|---|----|
| 1. | INTRODUCCIÓN..... | 1 |
| 1.1 | PANORAMA GENERAL DE LA INTEROPERABILIDAD A TRAVES DE OPC.... | 1 |
| 1.2 | INTEROPERABILIDAD EN LOS SISTEMAS DE INTEGRACION EN LOS NIVELES INFERIORES DE PLANTA..... | 2 |
| 1.3 | VISION GENERAL DEL PROYECTO REALIZADO | 3 |
| 2. | MARCO TEORICO | 4 |
| 2.1 | OPC | 4 |
| 2.2 | APLICACIONES DE OPC..... | 4 |
| 2.3 | ARQUITECTURA OPC..... | 5 |
| 2.4 | INTEGRACION OPC..... | 6 |
| 3. | GENERALIDADES DE LA ESPECIFICACION OPC DA AUTOMATION 2.02..... | 9 |
| 4. | DISEÑO E IMPLEMENTACION DEL SISTEMA UTILIZANDO EL MODELO MCS... | 12 |
| 4.1 | ESTUDIO DE PREFACTIBILIDAD | 12 |
| 4.1.1 | Declaración Inicial del Negocio | 12 |
| 4.1.2 | Características Esenciales | 13 |
| 4.1.3 | Personas y otros que Interactúan con el Sistema..... | 13 |
| 4.1.4 | Beneficios que se Logran..... | 13 |
| 4.1.5 | Criterios que Garantizan la Calidad del Sistema..... | 14 |
| 4.1.6 | Lista de Características del Sistema | 14 |
| 4.1.7 | Modelo actual del negocio | 15 |
| 4.1.8 | Modelo inicial de casos de uso del sistema. | 17 |
| 4.1.9 | Arquitectura Inicial del Sistema | 18 |
| 4.1.10 | Diagrama de Arquitectura..... | 19 |
| 4.1.11 | Listas de Riesgos | 19 |
| 4.1.12 | Lista De Prioridad De Casos De Uso Del Sistema..... | 20 |
| 4.1.13 | Revisión y Evaluación de la fase de Estudio de Prefactibilidad | 20 |
| 4.2 | FORMULACION DEL PROYECTO | 21 |
| 4.2.1 | Modelo Esencial De Casos De Uso Del Sistema | 21 |
| 4.2.2 | Descripción de los Escenarios de los casos de Uso esenciales | 21 |
| 4.2.3 | Revisión y Evaluación de los Casos de Uso Iniciales..... | 25 |
| 4.2.4 | Diagrama General de casos de Uso Mejorados..... | 27 |
| 4.2.5 | Diagramas generales de Secuencia. | 34 |
| 4.2.6 | Interfaces Requeridas | 39 |
| 4.2.7 | Contratos de las Operaciones Según cada Caso de Uso. | 44 |
| 4.2.8 | Diagramas de Colaboración Específicos. | 52 |
| 4.2.9 | Descripción de Paquetes y Clases de análisis Esenciales del sistema | 59 |
| 4.2.10 | Revisión y Evaluación de la fase de Formulación del Proyecto..... | 64 |
| 4.3 | EJECUCION DEL PROYECTO..... | 65 |
| 4.3.1 | Diagramas de Secuencia Específicos de Casos de uso de Diseño..... | 65 |
| 4.3.2 | Descripción de paquetes y Clases de Diseño de implementación del Sistema.... | 74 |
| 4.4 | VALIDACION DE LA SOLUCION..... | 85 |

| | | |
|-------|--|----|
| 4.4.1 | Desarrollo de Los programas en los PLCs a interoperar..... | 85 |
| 4.4.2 | Descripción del Proceso Utilizado en la Validación | 88 |
| 4.4.3 | Ejecución de Pruebas..... | 89 |
| 4.4.4 | Manuales de usuario | 92 |
| 5. | APORTES DEL PROYECTO REALIZADO | 93 |
| 6. | CONCLUSIONES | 95 |
| 7. | REFERENCIAS | 97 |
| 8. | BIBLIOGRAFIA..... | 98 |

INDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Estándares en la jerarquía de la automatización [2]..... | 2 |
| Figura 2. Arquitectura de información..... | 5 |
| Figura 3. Arquitectura de solución propuesta por OPC..... | 6 |
| Figura 4. Implementación propietaria. | 7 |
| Figura 5. Implementación de Solución OPC | 8 |
| Figura 6. Conjuntos de interfaces de OPC..... | 9 |
| Figura 7. Arquitectura Cliente Servidor OPC..... | 9 |
| Figura 8. Modelo de objetos de la interfaz de automatización..... | 10 |
| Figura 9. Ejemplo de lectura de Servidor OPC..... | 10 |
| Figura 10. Visión general del proceso de desarrollo MCS | 12 |
| Figura 11. Casos de uso generales del modelo actual. | 15 |
| Figura 12. Casos de uso generales planteados inicialmente | 17 |
| Figura 13. Arquitectura inicial del sistema | 19 |
| Figura 14. Casos de uso planteados inicialmente | 21 |
| Figura 15. Casos de uso mejorados..... | 27 |
| Figura 16. Secuencia Buscar Servidores | 35 |
| Figura 17. Secuencia Visualizar Browser | 36 |
| Figura 18. Secuencia Visualizar Propiedades..... | 37 |
| Figura 19. Secuencia Configurar Alarmas y Eventos | 38 |
| Figura 20. Secuencia Configurar Tendencias. | 39 |
| Figura 21. Interfaz frmPrincipal. | 40 |
| Figura 22. Interfaz Buscar Servidores | 40 |
| Figura 23. Interfaz Visualizar Browser | 41 |
| Figura 24. Interfaz Visualizar Propiedades | 41 |
| Figura 25. Interfaz Configurar Alarmas y Eventos | 42 |
| Figura 26. Interfaz Configurar Eventos | 42 |
| Figura 27. Interfaz Configurar Tendencia | 43 |
| Figura 28. Interfaz Visualizar Proceso..... | 43 |
| Figura 29. Interfaz Control Acceso..... | 44 |
| Figura 30. Diagrama de colaboración BuscarServidores()..... | 53 |
| Figura 31. Diagrama de colaboración SeleccionarServidores()..... | 53 |
| Figura 32. Diagrama de colaboración Browser(). | 54 |
| Figura 33. Diagrama de colaboración Mostrarramas()..... | 54 |
| Figura 34. Diagrama de colaboración VisualizarItems()..... | 55 |
| Figura 35. Diagrama de colaboración ActualizarValor()..... | 55 |
| Figura 36. Diagrama de colaboración EscribirValor(). | 56 |
| Figura 37. Diagrama de colaboración ConectarServidores()..... | 57 |
| Figura 38. Diagrama de colaboración ConfigurarAtributos(). | 57 |
| Figura 39. Diagrama de colaboración GuardarAlarmas()..... | 58 |
| Figura 40. Diagrama de Colaboración ConfigurarEvento() | 58 |
| Figura 41. Diagrama de colaboración GraficarTendencias(). | 58 |
| Figura 42. Diagrama de colaboración GuardarGrafica() | 59 |
| Figura 43. Diagrama de colaboración GenerarReportes() | 59 |

| | |
|---|----|
| Figura 44. Paquetes del sistema..... | 60 |
| Figura 45. Clases tipo interfaz..... | 62 |
| Figura 46. Clases de la Lógica de negocios | 63 |
| Figura 47. Casos de uso mejorados finales | 65 |
| Figura 48. Diagrama de secuencia Caso de uso No 1 Buscar Servidores | 66 |
| Figura 49. Diagrama de secuencia caso de uso No 2, Visualizar Browser | 67 |
| Figura 50. Diagrama de secuencia operación VisualizarItems()..... | 68 |
| Figura 51. Diagrama de secuencia operación conectarServidores(). | 68 |
| Figura 52. Diagrama de secuencia operación ActualizarValor() | 69 |
| Figura 53. Diagrama de secuencia operación EscribirValor() | 70 |
| Figura 54 Diagrama de secuencia operación ConfigurarAtributos() | 71 |
| Figura 55. Diagrama de secuencia operación ConfigurarEvento(). | 72 |
| Figura 56. Diagrama de secuencia operación GraficarTendencias(). | 73 |
| Figura 57. Diagrama de secuencia operación GenerarReportes(). | 74 |
| Figura 58. Diagrama de Componentes del sistema..... | 75 |
| Figura 59. Interfaces y clases de OPC Automation | 76 |
| Figura 60. Componentes generales de la arquitectura del sistema | 77 |
| Figura 61. Clases de implementación lógica de negocios | 78 |
| Figura 62. Clases tipo interfaz implementadas principales | 79 |
| Figura 63. Clases tipo interfaz implementadas | 80 |
| Figura 64. Clase tipo Servicios implementada | 80 |
| Figura 65. Modelo conceptual Servidores-Ítems. | 83 |
| Figura 66. Modelo Físico Servidor Ítems y Usuarios | 84 |
| Figura 67. Modelo de despliegue. | 85 |
| Figura 68. Programación del PLC con RSLogix 500 | 86 |
| Figura 69. RSLogix Emulate | 86 |
| Figura 70. Servidor RSLinx..... | 87 |
| Figura 71. Simulador para Servidor OPC Siemens (OPC Scout) | 88 |
| Figura 72. Interfaz HMI de operario para el Proceso | 89 |

INDICE TABLAS

| | |
|---|----|
| Tabla 1. Descripción del modelo de objetos OPC DA..... | 11 |
| Tabla 2. Características generales del sistema..... | 14 |
| Tabla 3. Descripción de los casos de uso del modelo actual | 16 |
| Tabla 4. Descripción de los casos de uso iniciales..... | 18 |
| Tabla 5. Descripción de riegos del proyecto | 19 |
| Tabla 6. Escenarios de los casos de uso esenciales..... | 25 |
| Tabla 7. Descripción de los escenarios de los casos de uso mejorados..... | 33 |
| Tabla 8. Funciones Básicas OPC..... | 45 |
| Tabla 9. Funciones de supervisión OPC..... | 45 |
| Tabla 10. Contratos de las operaciones de los casos de uso. | 51 |
| Tabla 11. Entidad Servidores..... | 81 |
| Tabla 12. Entidad Ítems..... | 82 |
| Tabla 13. Entidad Usuarios..... | 82 |
| Tabla 14. Pruebas de Interfaz..... | 91 |

RESUMEN

El presente trabajo de grado presenta el estudio y desarrollo de una aplicación software por medio de la implementación de la especificación OPC. Se abarcan los conceptos teóricos en que se basa la aplicación realizada, el diseño detallado sustentado en los requerimientos y la implementación final a través de las correspondientes herramientas software de programación. Se da a conocer en detalle cómo se alcanzó el objetivo de diseñar e implementar una herramienta software basada en OPC (Cliente OPC) que permite acceder a datos de diferentes PLCs y almacenarlos en una base de datos, presentando también características de sistema SCADA, al permitir visualizar, establecer alarmas y generar tendencias y reportes de manera esencial. Contribuyendo así al paradigma de la interoperabilidad, al realizar un cambio en las aplicaciones Cliente actuales, puesto que el sistema no solo permite la interconexión de diferentes PLCs si no que recibe, ordena, manipula y muestra los datos (no solo es un explorador como ocurre en muchas aplicaciones actuales), permitiendo su utilización en un sistema Supervisorio con sus respectivas funcionalidades. Con este proyecto se realiza un aporte en la interoperabilidad de sistemas a nivel local, nacional y en los modelos de integración actuales, además de los proyectos macro que se desarrollan dentro del grupo I+D en Automática Industrial en lo referente a sistemas Supervisorios y de integración empresarial.

1. INTRODUCCIÓN

En este capítulo se presentará el panorama general de la especificación OPC dentro de las empresas a nivel mundial, nacional y local y su vinculación dentro de los procesos de integración.

1.1 PANORAMA GENERAL DE LA INTEROPERABILIDAD A TRAVES DE OPC

OPC (OLE para control de procesos) fue desarrollado para estandarizar los sistemas propietarios de drivers de control, automatización de redes industriales y buses de campo de múltiples fabricantes, buscando interoperabilidad entre estos sistemas [1]. En esencia el problema ha sido que los datos en los sistemas de control, supervisión y sistemas distribuidos de diferentes fabricantes, tienen distintos protocolos de comunicación y diferentes formatos, además las redes industriales (SimaticNet, DeviceNet, etc.) y Buses de campo (Modbus, Profibus, etc.), tienen problemas de incompatibilidad en la comunicación, originando problemas en la integración e interoperabilidad con diversos fabricantes, ya que no hay ninguna garantía de que una solución propietaria interactúe de forma sólida y eficiente con otra de distinto fabricante, lo que trae consigo una duplicación de esfuerzo y un alto costo en software y hardware [1].

Por lo tanto la fundación OPC (OPC Foundation), que es una corporación sin ánimo de lucro, integrada por múltiples fabricantes [2], trabaja en la solución de estos problemas y en la propagación del estándar a nivel mundial, lo cual ha permitido que diversas empresas reconocidas como Siemens, Allen Bradley, Omron y muchas otras; hayan adoptado el estándar y realizado distintas aplicaciones que hoy en día son utilizadas con fines de interoperabilidad de dispositivos, sistemas SCADA y en general en el manejo de datos de planta. Sin embargo, debido a que las últimas especificaciones son relativamente recientes (2002-2006), y que las empresas tienden a ofrecer soluciones propietarias a nivel de integración e interoperabilidad, el estándar aun no ha alcanzado un uso realmente masivo, aunque tiende al crecimiento [2].

Generalmente el nivel de adopción de este estándar se ve claramente reflejado en la lista de miembros que publica la fundación OPC [2], donde se observa que la mayor parte de empresas miembros de OPC, son Europeas y Norte americanas. En Latinoamérica solo se encuentran hasta el momento, dos empresas Brasileñas, y en Colombia en la actualidad no hay ninguna empresa miembro de la fundación OPC, lo que evidencia la falta de nivel de desarrollo y aporte en el estándar. [2]. Por lo tanto es importante que en nuestro país y en especial en nuestro medio se adopte este tipo de estándares que ayuden a obtener, manejar, controlar y ordenar los datos y la información de nuestras empresas productivas.

1.2 INTEROPERABILIDAD EN LOS SISTEMAS DE INTEGRACION EN LOS NIVELES INFERIORES DE PLANTA.

Dentro de los niveles de jerarquía de automatización se encuentran varios estándares que facilitan la integración vertical, es el caso de las normas ISA S88 e ISA S95 para los niveles 2, 3 y 4 y en los niveles inferiores, 0, 1 y 2, se encuentra la especificación OPC [3]. Es en estos niveles inferiores donde ocurren los procesos productivos y se presentan tareas de control, adquisición de datos, supervisión de procesos y demás; tareas que substancialmente requieren que los dispositivos encargados de realizar dichas funciones trabajen de manera mancomunada en la consecución de los objetivos de la producción, aprovechando eficientemente la información que se genera en estos niveles para la posterior toma de decisiones en niveles superiores [3] .

En el caso especial de este proyecto, dando una visión general del aporte dentro de los procesos de integración, este se enmarca en los niveles 1 y 2 de la jerarquía como se muestra en la Figura 1. Sin embargo es considerable resaltar que la fundación OPC trabaja actualmente en la consecución de una norma unificada basada en protocolos de Internet y servicios Web para permitir la integración total de los datos a niveles superiores [3] .

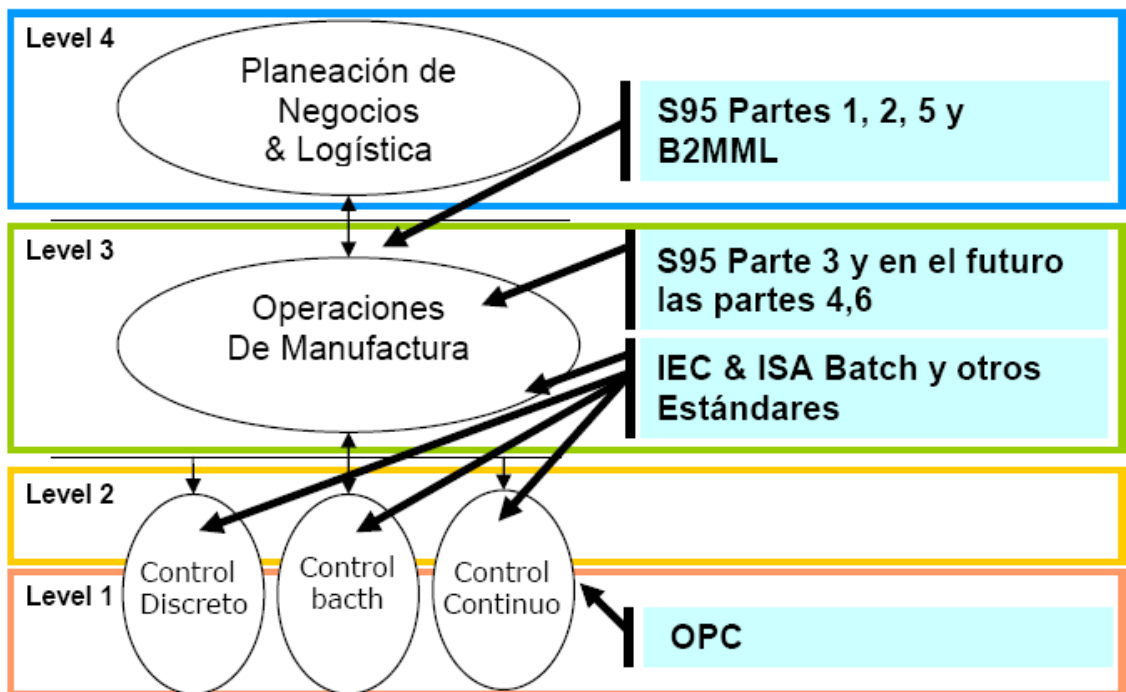


Figura 1. Estándares en la jerarquía de la automatización [2].

1.3 VISION GENERAL DEL PROYECTO REALIZADO

La aplicación realizada tiene como uno de sus objetivos la interoperabilidad entre dos dispositivos de diferentes fabricantes, por lo cual se trabajó especialmente con la especificación de la fundación OPC, Interfaz de Automatización acceso a datos 2.02 (se tratará más adelante).

En primera instancia, se realizó el modelado a través de UML (Lenguaje Modelado Unificado) para captar los requerimientos de la aplicación software a realizar y a partir de ellos desarrollar todo el sistema, enmarcado dentro del modelo propuesto MCS (Modelo de Construcción de Soluciones) que explicará todo el contenido, los errores y mejoras, hasta lograr la implementación final pasando por las fases de diseño y pruebas.

Por último se codificaron todos los modelos finales a través de la herramienta software Visual Basic.NET con sus respectivas bases de datos en SQL Server 2005 y diferentes funcionalidades teniendo en cuenta una arquitectura software ordenada y definida.

2. MARCO TEORICO

Lograr que interoperen diferentes sistemas, que intercambien datos, que se pueda realizar una supervisión del proceso, independiente del fabricante del dispositivo o de la solución propietaria, es casi un paradigma y más concretamente un reto para muchas empresas, ya que el problema que sucede en muchos procesos productivos es la proliferación de las llamadas “islas automatizadas”. Se presentan a continuación los conceptos fundamentales que sustentan este proyecto de interoperabilidad a través de OPC; los cuales apoyarán y darán un mejor entendimiento del desarrollo logrado y de los aportes realizados.

2.1 OPC

OPC (*OLE for Process Control*) [4], es una especificación técnica no propietaria definida por la Fundación OPC (*OPC Foundation* es una entidad sin fines de lucro, encargada de administrar la especificación OPC) y consiste básicamente en un Sistema de Interfaces Estándar basado en OLE/COM y DCOM de Microsoft. Con OPC es posible interoperar dispositivos industriales con sistemas de información o aplicativos de escritorio. En otras palabras, OPC permite desarrollar de una manera muy práctica y eficiente aplicaciones que pretendan comunicarse con equipos industriales controlados por PLCs [4].

2.2 APLICACIONES DE OPC

Mediante OPC se puede comunicar a numerosas fuentes de datos uniformemente, sin importar el tipo, más exactamente dispositivos en el piso de la fábrica, o en una base de datos en una habitación de control. La arquitectura de la información para la Industria del Proceso mostrada en la Figura 2 , implica los niveles siguientes:

Administración de Campo: a través de OPC se puede acceder a la información de los dispositivos de campo, como sus parámetros de configuración, variables de proceso etc. y ser llevada al usuario o a alguna aplicación que necesite usarla y manipularla.

Procesos de administración: los sistemas distribuidos DCS y los sistemas SCADA para controlar y supervisar datos del proceso de fabricación, necesitan recoger los datos de todas las fuentes disponibles, para lograr realizar estos procedimientos de forma eficiente.

Administración de Negocio: los fabricantes necesitan recoger los datos de planta e integrarlos a su sistema de negocios. Los Software de Supervisión y bases de datos deben satisfacer estas necesidades mediante una arquitectura abierta de comunicación basada en el acceso a datos y no en el tipo de datos, que pueda de manera eficiente llevarlos a la parte de la empresa que maneja los procesos financieros.

En la Figura 2 se observa que OPC proporciona los siguientes tipos de datos:

- Datos en tiempo real de sensores – temperatura, presión, flujo.
- Parámetros de control – Abierto, cerrado, run, stop
- Información de estado:
 - Estado de la conexión con el Hardware
 - Estado del software y subsistema local
- Datos disponibles, como información de máquina

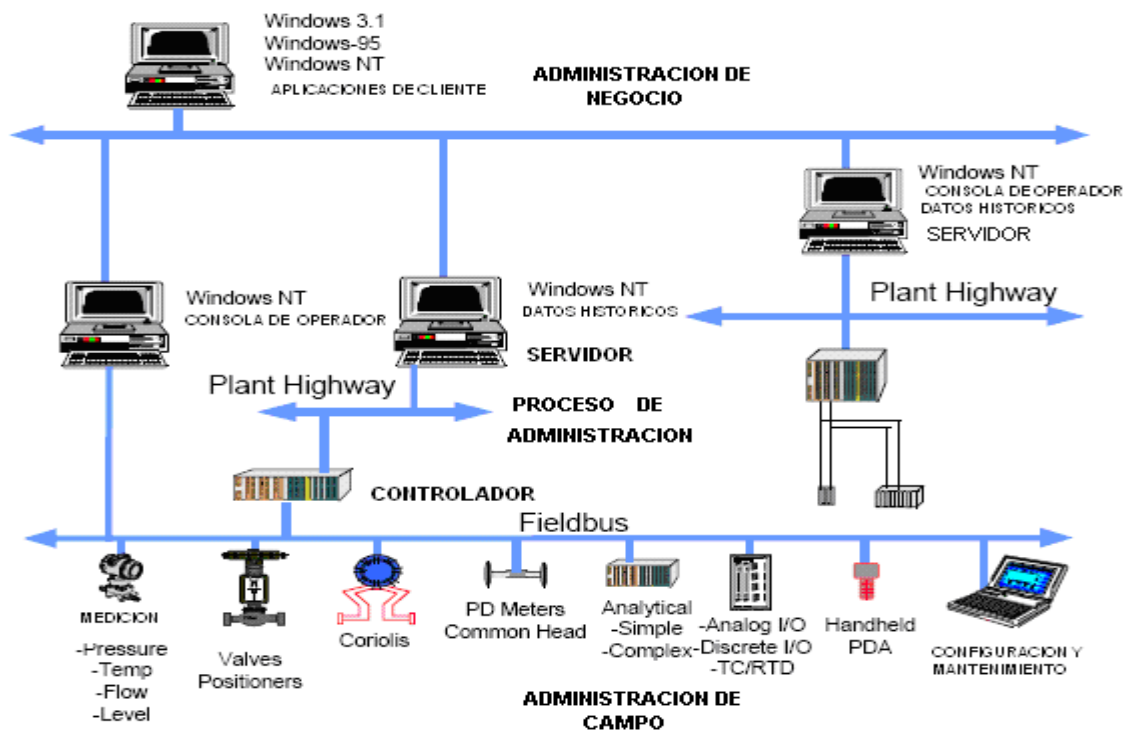


Figura 2. Arquitectura de información

2.3 ARQUITECTURA OPC

OPC propone un modelo cliente servidor, donde distintas aplicaciones tienen una **base común** de acceder a los datos. El servidor OPC se comunica con el cliente OPC, garantizando una comunicación eficaz, donde cada vendedor no necesitará un drive para comunicarse con cada aplicación, si no que OPC le proporciona la capacidad de poder leer y escribir a cada dispositivo, con cada una de las distintas aplicaciones, como se observa en la Figura 3 [5].

Solución OPC

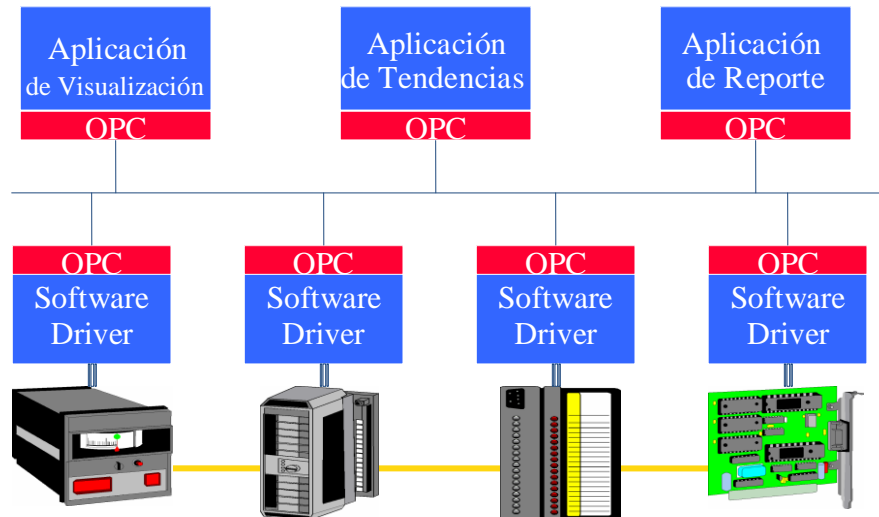


Figura 3. Arquitectura de solución propuesta por OPC

2.4 INTEGRACION OPC

La arquitectura y el diseño de OPC, logran que se pueda configurar un Servidor OPC que permita a una aplicación cliente acceder a datos de distintos Servidores OPC.

A pesar de que OPC está diseñado principalmente para acceder a datos de un servidor en red, las interfaces OPC se pueden usar para distintas aplicaciones. En el nivel más bajo se puede tomar datos de aparatos físicos y llevarlos a SCADA o DCS, o de un servidor SCADA o DCS a una aplicación [5].

Por esta razón, para entender la necesidad de estandarizar un proceso de control industrial y así mostrar la importancia de OPC y de esta clase de proyectos, se presenta el siguiente ejemplo de estudio, de una empresa Petroquímica, donde se observa la solución propietaria para la integración en contraste con la solución OPC.

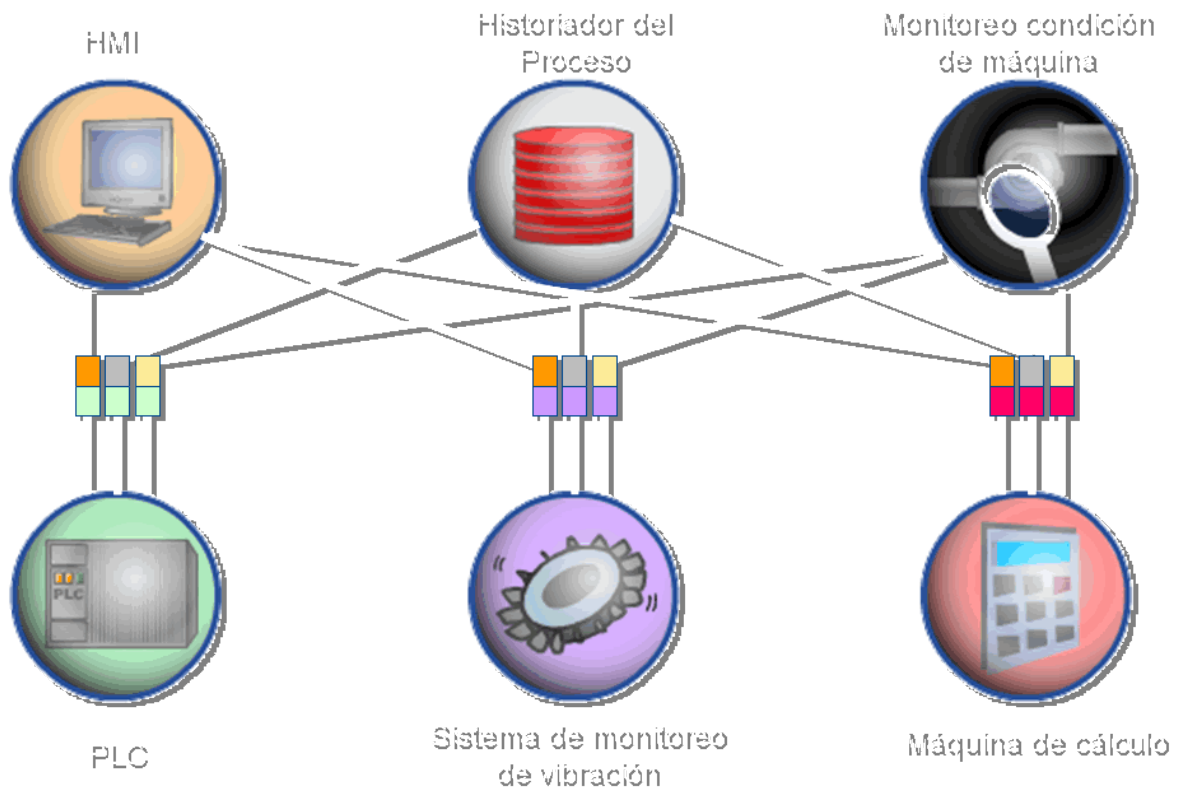


Figura 4. Implementación propietaria.

Como se puede observar en la Figura 4. Implementación propietaria., el PLC, el sensor de vibración, y el motor de cálculo, trabajan sobre diferentes protocolos de comunicación – TSAA, Modbus, y DDE. Por eso la aplicación Software que requiera datos de estos dispositivos también necesita ser entendida por los protocolos TSAA, Modbus, y DDE. Adicionalmente cada dispositivo necesita atender las tres aplicaciones. Los Históricos de Proceso, la aplicación GUI (que proveen una interfaz Humano – máquina) y el monitor de Condición de Máquina, todos se comunican con todos los dispositivos, resultando algo totalmente complejo. Este despliegue toma a la compañía alrededor de 10 días y su costo es aproximadamente de US \$ 50.000

Se observa en la Figura 5 como se reduce la complejidad. Ahora todas las aplicaciones software hablan solamente con el Servidor OPC y todos los dispositivos responden solamente al Servidor OPC, el cual proporciona la “caja negra” para aplicaciones software hacia el hardware.

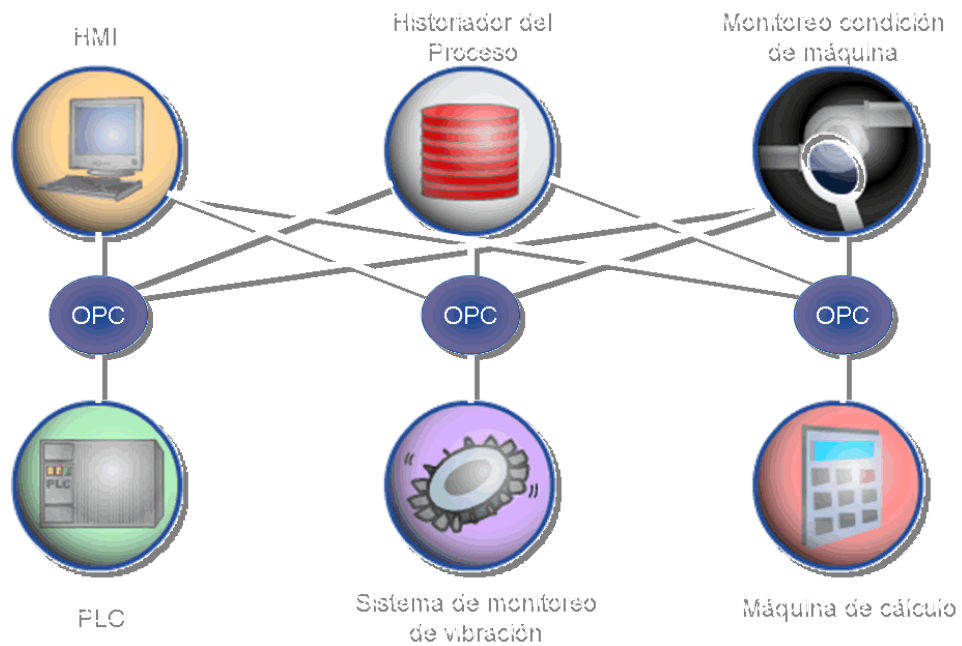


Figura 5. Implementación de Solución OPC

La implementación OPC toma alrededor de 2 días para su despliegue con un costo aproximado de US \$10.000. Se podría eliminar incluso dos Servidores OPC e implementar los drivers correctos dentro de un simple Servidor OPC. Esto produce un equilibrio y deja que la plataforma de los servidores OPC pueda soportar protocolos estándar como DDE, Modbus y TSAA [6].

Algunos de los beneficios son:

- Reducir la carga en el dispositivo.
- La escalabilidad de los sistemas es incrementada.
- El Servidor OPC proporciona datos por caché.
- Las aplicaciones Cliente no necesitan conocer detalles de los protocolos.
- Se incrementa la vida del dispositivo (no necesita atender a múltiples clientes).
- Interoperabilidad (Unix/Linux y Windows – ambas plataformas son soportadas por OPC)
- Estandarización.

3. GENERALIDADES DE LA ESPECIFICACION OPC DA AUTOMATION 2.02

Esta especificación Automation Interface Data Acces 2.02 [7], es una interfaz para desarrollar diferentes aplicaciones OPC de acceso a datos. El propósito es proveer una interfaz funcional de automatización para acceder a los datos del servidor. Esto quiere decir que define una forma de comunicación para aplicaciones que necesiten acceder a una fuente de datos de un dispositivo. Es importante entender que la especificación define dos conjuntos de interfaz, la OPC interfaz Custom y la OPC interfaz de automatización, la primera para aplicaciones realizadas en C++ y la segunda para aplicaciones realizadas en Visual Basic. Para estas últimas aplicaciones la Fundación OPC (OPC Foundation) provee una librería de enlace dinámico (dll, conocido como Wrapper) que permite comunicar la aplicación Cliente realizada utilizando la interfaz de automatización, con cualquier servidor OPC. Esto se muestra en la Figura 6. Se debe clarificar que la especificación proporciona las interfaces OLE/COM (Objets Linking Embebed) no para implementar una aplicación en particular.

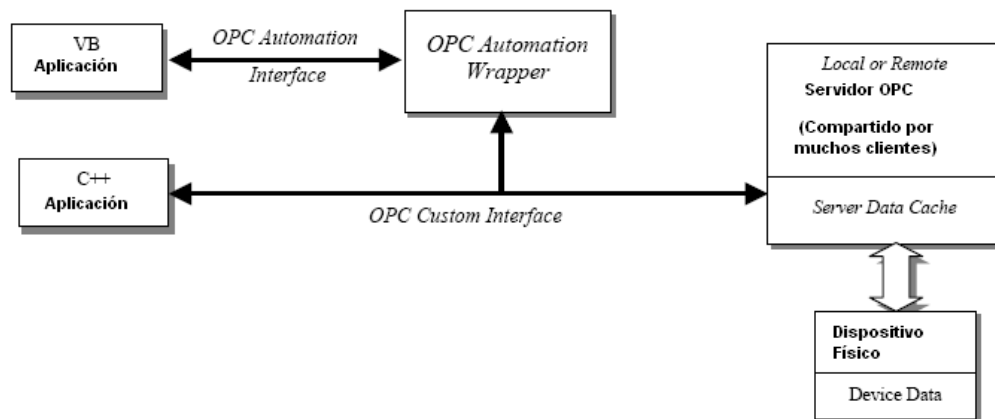


Figura 6. Conjuntos de interfaces de OPC



Figura 7. Arquitectura Cliente Servidor OPC

Como toda implementación COM, la arquitectura OPC tiene un modelo Cliente – Servidor donde el objeto Servidor OPC (el Servidor es tomado como un objeto OPC Server), provee una interfaz al objeto OPC como se muestra en la Figura 7. Por tanto el servidor de datos OPC está compuesto por el Servidor, el Grupo y el ítem. La especificación

define un modelo de objetos, los cuales tienen propiedades y métodos específicos. Estos objetos son mostrados y explicados en la Tabla 1 y Figura 8.

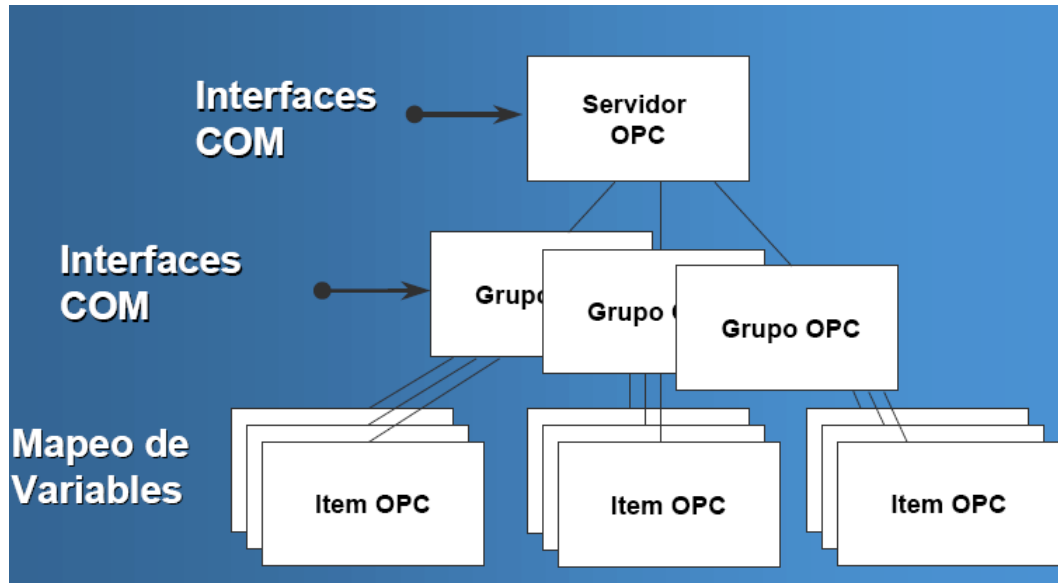


Figura 8. Modelo de objetos de la interfaz de automatización

Esencialmente las variables son definidas en los controladores y reflejadas al Servidor OPC DA como se muestra en Figura 9

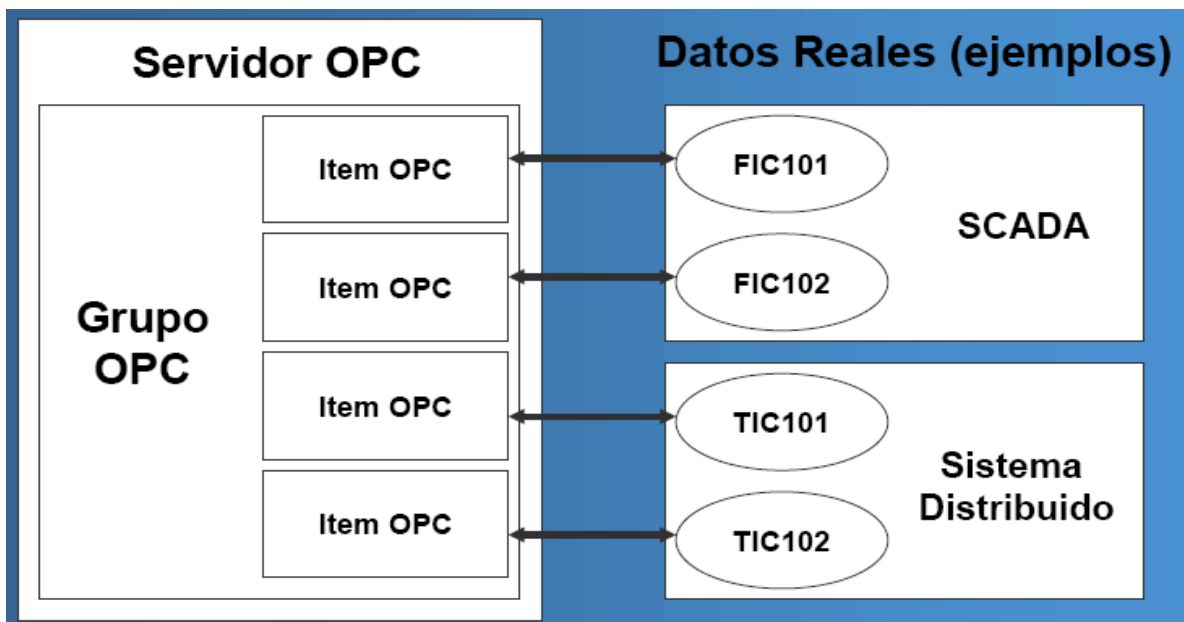


Figura 9. Ejemplo de lectura de Servidor OPC

A continuación se muestra en la Tabla 1 la explicación de las características de cada objeto, lo que dará una visión general de la especificación [7].

| OBJETO | DESCRIPCION |
|---------------------|---|
| Servidor OPC | Los objetos correspondientes a la categoría servidor OPC (OPCServer) los genera el Cliente. Las propiedades de un servidor OPC contienen información general sobre el servidor. Cuando se crea un objeto OPCServer, también se crea además un objeto de colección de OPCGroup como propiedad del objeto OPCServer. |
| OPCBrowser | El objeto OPCBrowser es una colección que permite averiguar el área de direccionamiento del servidor OPC. Debe crearse un objeto de la categoría OPCBrowser con el método CreateBrowser para el objeto OPCServer. Pueden crearse diferentes objetos OPCBrowser para un servidor. |
| Grupos OPC | El objeto OPCGrupos (OPCGroups) es un objeto de colección para crear y administrar los grupos OPC. Las propiedades predeterminadas de los grupos OPC especifican los valores por defecto para la creación de todos los grupos OPC. |
| Grupo OPC | La categoría Grupo OPC (OPC Group) gestiona las variables individuales del proceso, los elementos OPC. Con estos objetos de grupo, un cliente puede formar unidades con sentido semántico de elementos OPC y ejecutar operaciones con ellas. |
| Ítems OPC | El objeto ítems OPC (OPCItems) es un objeto de colección para la creación y la administración de elementos OPC. Las propiedades predeterminadas de los elementos OPC especifican los valores por defecto para la creación de todos los elementos OPC. |
| Ítem OPC | Un objeto de la categoría ítem OPC representa un vínculo con una variable del proceso, por ejemplo para el módulo de entrada de un controlador programable. Una variable de proceso es un dato de las E/S del proceso que puede escribirse/leerse, como por ejemplo la temperatura de un tanque. Cada variable de proceso está asociada a un valor (tipo de datos variante), a una cualidad y a un sello horario. |

Tabla 1. Descripción del modelo de objetos OPC DA.

Desde los anteriores conceptos teóricos fundamentales sobre OPC y en especial de la especificación Interfaz de automatización, se inicia toda la construcción de la solución y aporte del presente proyecto, basados en el cumplimiento de los objetivos.

4. DISEÑO E IMPLEMENTACION DEL SISTEMA UTILIZANDO EL MODELO MCS

El modelo para la construcción de soluciones MCS tiene como paradigma principal, el desarrollo de una técnica incremental mediante la cual el desarrollador sea provisto en una etapa temprana del proyecto, (en este caso una aplicación software), de una capacidad operacional inicial, donde se pueda ejercitar por medio de incrementos de funcionalidad, enfocados a una actividad permanente de revisión de los subproductos desarrollados, en cada una de las fases tratando de corregir hasta llegar al Sistema/Solución completo [8]. Como muestra la Figura 10 rápidamente se está tocando y evaluando todas las etapas hasta llegar a un desarrollo consistente obteniendo versiones del proyecto o de los subproductos y evaluando hasta llegar a la mejor solución. Se explicará entonces siguiendo el modelo MCS y apoyándonos técnicamente del diseño UML, cada una de las etapas del proyecto desarrollado.

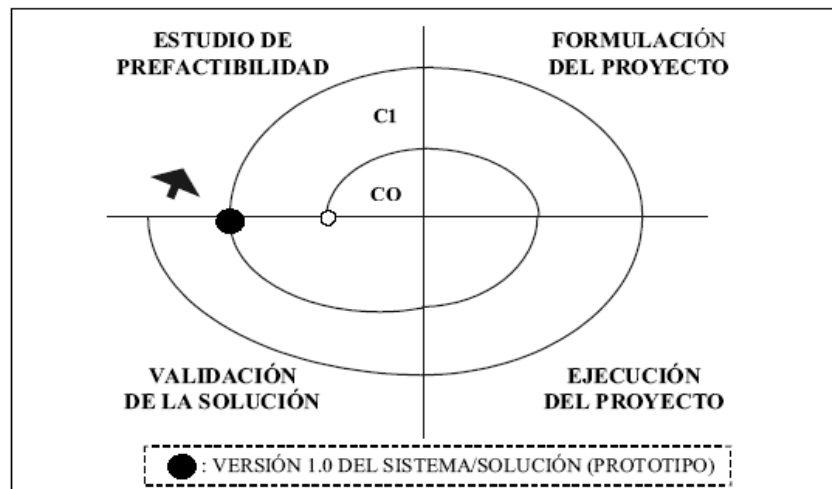


Figura 10. Visión general del proceso de desarrollo MCS

4.1 ESTUDIO DE PREFACTIBILIDAD

4.1.1 Declaración Inicial del Negocio

Declaración del Propósito: el sistema permite interconectar a través del estándar de comunicación abierta (OPC), uno o dos PLCs de diferentes marcas, permitiendo en primera instancia acceder a los datos y estableciendo a partir de esto la capacidad de almacenamiento de datos y funcionalidades de un sistema SCADA.

El sistema es desarrollado en Visual Studio ya que esta herramienta permite acceder a servicios de interoperabilidad (interoperability services) e interfaces gráficas de fácil uso y amenas al usuario.

4.1.2 Características Esenciales

a) Esenciales

- Permitir al usuario, en el caso de un operario observar el mímico (HMI que muestra el proceso en forma de objetos gráficos dinámicos) de la planta cuando el proceso este en marcha o no lo este.
- Permitir al operario ver las alarmas, tendencias y reportes que se obtengan del proceso y han sido previamente configuradas según su tipo.
- Conectar dos PLCs de diferentes marcas (por ejemplo Siemens y Allen Bradley).
- La interacción del operario con los PLCs a través del sistema.
- Permitir al usuario, en este caso un administrador gestionar las diferentes funciones que debe realizar el sistema como los son: Reportes, Alarmas y Tendencias.
- Permitir al administrador configurar el sistema.

b) Opcionales

- Conectar más de dos PLCs al sistema supervisorio.

4.1.3 Personas y otros que Interactúan con el Sistema.

- Operario: Persona que manipula el software con ciertas limitaciones para ejecutar sus funciones.
- Servidores OPC: Son controladores OPC que permiten acceder a los datos de los PLCs.
- PLC: Controlador Lógico Programable donde se conectan los sensores y actuadores del proceso.

4.1.4 Beneficios que se Logran

- En primera instancia los beneficios son de orden económico, puesto que permite conectar varios PLCs de diferentes propietarios en un solo sistema supervisorio, sin necesidad de adquirir el software de una sola marca.
- Beneficios desde el punto de vista del manejo y acceso de la información ya que se logra obtener datos del proceso en tiempo real entre dispositivos de diferentes fabricantes de una forma simple y rápida lo que consecuentemente ofrece todas las ventajas inherentes de la integración de sistemas y de la interoperabilidad de dispositivos (observar apartado 2.4).

4.1.5 Criterios que Garantizan la Calidad del Sistema

- Interacción de los datos del PLC bajo el protocolo de comunicación OPC con Visual Studio.NET que es conjunto de herramientas con una plataforma enfocada en la integración de servicios y la interoperabilidad.
- Interfaz de fácil uso por parte del usuario.
- Implementación de sistema de seguridad mediante un control de acceso con su respectiva base de datos de usuarios.
- Interacción con una base de datos donde se almacenen las variables del sistema y demás datos importantes.

4.1.6 Lista de Características del Sistema

| Identificador | Definición | Conjunto de variables | |
|-------------------------------|--|------------------------------|---------------|
| Control de Acceso | Se encarga de establecer el acceso al sistema, según el tipo de usuario comprobando su contraseña. | Estado: | Aprobado |
| | | Prioridad: | Crítica |
| | | Nivel riesgo: | Ordinario |
| Visualización de los procesos | Interfaz gráfica donde se exponen las funciones del sistema. | Estado: | Aprobado |
| | | Prioridad: | Crítica |
| | | Nivel riesgo: | Ordinario |
| Configuración del Sistema | Manejo del protocolo OPC, bajo la plataforma de desarrollo del sistema | Estado: | Aprobado |
| | | Prioridad: | Importante |
| | | Nivel riesgo: | Significativo |
| Gestión de datos | Se encarga de la interconexión de los servidores con la plataforma donde se desarrollará el sistema. | Estado: | Aprobado |
| | | Prioridad: | Importante |
| | | Nivel riesgo: | Significativo |
| Configurar alarmas | Crea, guarda y edita alarmas. | Estado: | Aprobado |
| | | Prioridad: | Importante |
| | | Nivel riesgo: | Ordinario |
| Gestión de tendencias | Se observa el comportamiento de una de las variables del proceso. | Estado: | Aprobado |
| | | Prioridad: | Crítica |
| | | Nivel riesgo: | Ordinario |
| Gestión reportes | Permite buscar y imprimir reportes. | Estado: | Aprobado |
| | | Prioridad: | Crítica |
| | | Nivel riesgo: | Ordinario |

Tabla 2. Características generales del sistema

4.1.7 Modelo actual del negocio

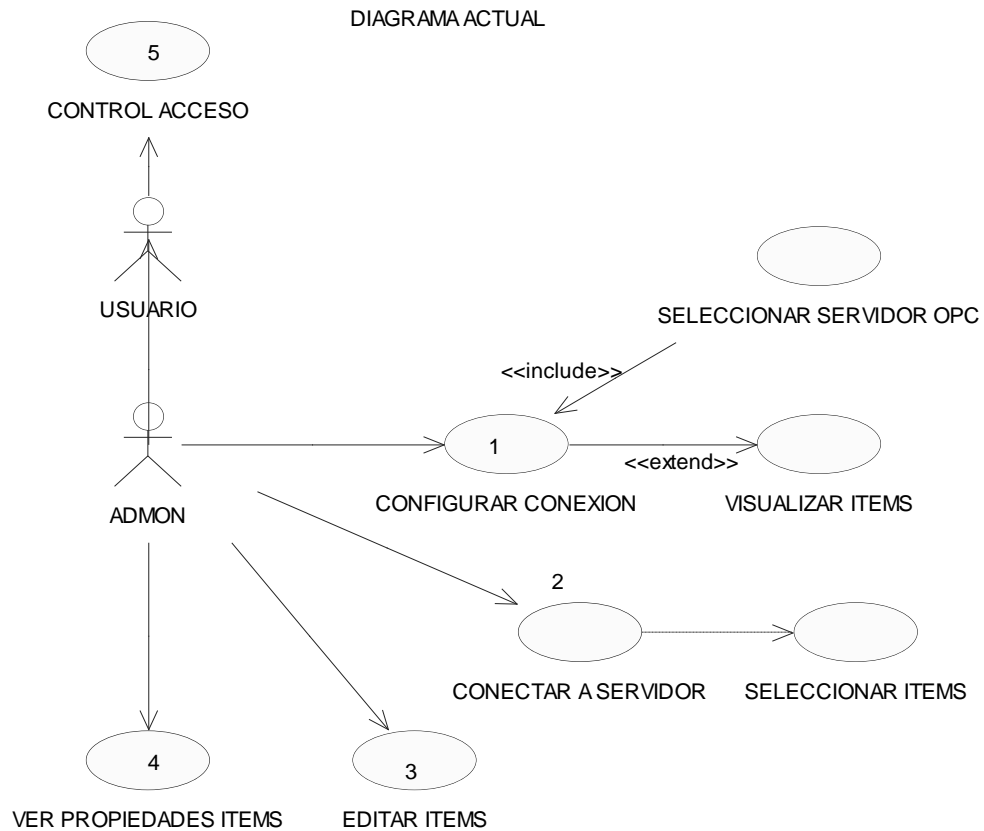


Figura 11. Casos de uso generales del modelo actual.

La Figura 11 nos muestra las actuales aplicaciones OPC Cliente encontradas en el mercado; se observan los requerimientos desde la perspectiva del usuario. Aunque hay diferencias de aplicación a aplicación, en general pretenden resolver el mismo problema de acceder a los datos de un PLC determinado, comunicándose con su servidor OPC respectivo y lograr visualizar el valor y las propiedades de los ítems accedidos. Generalmente estas aplicaciones se enfocan en ser una especie de visualizador o explorador de las variables en el PLC, algunas con diferentes funciones de configuración para establecer grupos de ítems.

La visualización del proceso en las aplicaciones clientes OPC actuales, corresponde solamente a observar el cambio de las variables del proceso en tiempo real pero no de funciones de integración de datos de proceso, que permitan visualizar las variables obtenidas en un sistema supervisorio con interfaces de usuario amigables, interoperando

distintos PLCs, en los cuales se implementan funciones donde se observe la interacción real de dispositivos de distintas marcas [9].

La siguiente tabla explicará uno a uno cada caso de uso que realiza un usuario de este tipo de clientes OPC en general.

| | |
|---------------------------|---|
| Caso de uso No. 1: | Configurar la conexión |
| Iniciador: | Administrador |
| Propósito: | Establecer la conexión con el PLC para obtener los respectivos datos |
| Resumen: | El administrador configura la conexión al PLC, seleccionado de una lista donde aparecen diferentes servidores OPC, el servidor donde se encuentran las variables a visualizar, luego se conecta al servidor y recibe los ítems de este. |
| | |
| Caso de uso No. 2: | Conectar a Servidor |
| Iniciador: | Operario |
| Propósito: | Adicionar los ítems que quiere visualizar |
| Resumen: | El operario selecciona los ítems mostrados del servidor OPC y se conecta observando sus propiedades |
| | |
| Caso de uso No. 3: | Editar Ítems |
| Iniciador: | Administrador |
| Propósito: | Edita los ítems seleccionados |
| Resumen: | El usuario elimina, de la visualización los ítems que no desee observar |
| | |
| Caso de uso No. 4: | Ver propiedades Ítems |
| Iniciador: | Administrador |
| Propósito: | Permitir al usuario ver las propiedades de los ítems en especial su valor actualizado. |
| Resumen: | Al adicionar los ítems en el caso de uso 2, se inicia una lectura sincrónica o asincrónica de estos, el usuario observa el valor, calidad, fecha y hora de la lectura, además de otras propiedades |
| | |
| Caso de uso No. 5: | Control de acceso |
| Iniciador: | Usuario |
| Propósito: | Validar el acceso al sistema de los diferentes usuarios |
| Resumen: | El usuario puede entrar como administrador , a diferencia de un usuario corriente tendrá más opciones en la configuración |

Tabla 3. Descripción de los casos de uso del modelo actual

4.1.8 Modelo inicial de casos de uso del sistema.

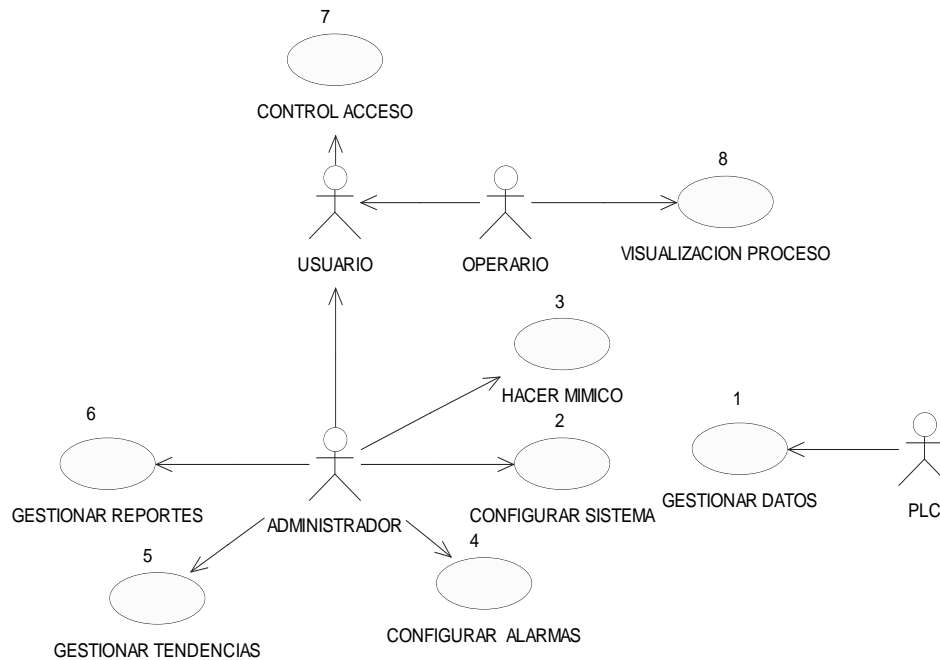


Figura 12. Casos de uso generales planteados inicialmente

La Figura 12 plantea los casos de uso iniciales que incluyen tácitamente los requerimientos desde la perspectiva del usuario (es decir siguiendo los parámetros de los objetivos del proyecto) y teniendo en cuenta las características deseadas de la Tabla 2. En primera instancia se explican cada caso de uso en la Tabla 4 definiendo su propósito, resumen y el actor iniciador. Desde este punto empieza el análisis y desarrollo del sistema o aplicación OPC buscada, teniendo como base los requerimientos y a partir de los casos de uso se obtienen los demás diagramas que describen la aplicación desde la perspectiva de diseño y el modelo final planteado para codificación. Estos diagramas irán cambiando y siendo modificados a medida que se van estableciendo nuevos requerimientos o detectando que los planteados inicialmente no satisfacen el propósito final, de acuerdo a la metodología MCS.

| | |
|---------------------------|--|
| Caso de uso No. 1: | Gestionar datos |
| Iniciador: | PLC |
| Propósito: | Acceder a los datos de los PLCs conectados para entregarlos al sistema |
| Resumen: | Al PLC, se conecta todas las variables del proceso y permite por medio de controladores OPC acceder a los datos y llevarlos al cliente (si no hay PLC conectado no hay gestión de datos). Además darles prioridad. |

| | |
|---------------------------|--|
| Caso de uso No. 2: | Configurar sistema |
| Iniciador: | Administrador |
| Propósito: | Configurar las variables del sistema dependiendo del tipo de variables (direccionamiento). |
| Resumen: | Seleccionar servidor OPC, realizar una organización jerárquica, eventos, condiciones, errores, búsqueda. |
| | |
| Caso de uso No. 3: | Hacer mímico |
| Iniciador: | Administrador |
| Propósito: | Realizar el mímico para el proceso |
| Resumen: | Se importan, configuran y sincronizan con el proceso real las figuras del sistema supervisorio para ser visualizadas posteriormente por el operario. |
| | |
| Caso de uso No. 5: | Gestionar tendencias |
| Iniciador: | Administrador |
| Propósito: | Permitir ver la tendencia del proceso y guardar o modificar sus características o propiedades a graficar de algún PLC. |
| Resumen: | El administrador hace cambios en las propiedades de las graficas y las guarda |
| | |
| Caso de uso No. 6: | Generar reportes |
| Iniciador: | Administrador |
| Propósito: | Permitir al admón. Imprimir, buscar, eliminar reportes |
| Resumen: | El admón. Elimina, busca, imprime reporte según el tipo de PLC (1 o 2) al que acceda. |
| | |
| Caso de uso No. 7: | Control De acceso |
| Iniciador: | Usuario |
| Propósito: | Conocer la identidad de los usuarios |
| Resumen: | Verificar que el usuario esta registrado en la base de datos del sistema y corresponde según tipo de usuario. |
| | |
| Caso de uso No. 8: | Visualización mímico |
| Iniciador: | Operario |
| Propósito: | Observar las características de los procesos que ejecutan los PLCs |
| Resumen: | El operario observa los procesos que se conectan a los PLCs. |

Tabla 4. Descripción de los casos de uso iniciales

4.1.9 Arquitectura Inicial del Sistema

La arquitectura utilizada es cliente servidor, donde gracias al protocolo de comunicación OPC, se permite que interactúe un cliente con dos servidores.

4.1.10 Diagrama de Arquitectura



Figura 13. Arquitectura inicial del sistema

4.1.11 Listas de Riesgos

| Identificador | Descripción | Impacto | Alternativas |
|---|--|--|--|
| Baja disponibilidad de PLCs de diferentes marcas | Solo se tiene PLCs de una misma marca en el departamento | Atraso en la ejecución del proyecto y validación de la ejecución | Trabajar con software de simulación de PLCs Siemens, el cual es más fácil de conseguir. |
| Especificaciones del computador e interfaces incorrectas o mal configuradas | Las especificaciones en los puertos para tomar los datos y llevarlos al sistema y especificaciones técnicas de los computadores que se tiene actualmente no sean la apropiadas | Atraso debido a la imposibilidad de adquirir los datos por insuficiencia de puertos o características no configurables de los equipos de computo | Obtener los permisos para realizar cambios al hardware o pedir en tal caso un cambio de equipos. |
| Limitación del servidor de Siemens | Servidor de prueba gratuito, con un límite de tiempo de 2 horas aproximadamente. | Retraza la elaboración del proyecto debido al tiempo limitado de ejecución, para realizar pruebas | Aprovechar al máximo el tiempo de ejecución, programando las pruebas antes de ejecutarlas. |
| Lectura en tiempo real de las variables del proceso | El tiempo de respuesta depende donde se desarrolle el sistema supervisorio | Algún tipo de retraso al cambiar una variable en el proceso. | Buscar la lógica o subrutina más eficiente, en cuanto al tiempo de lectura. |

Tabla 5. Descripción de riesgos del proyecto

4.1.12 Lista De Prioridad De Casos De Uso Del Sistema

Caso de uso N° 1: Gestión de Datos

Valor de prioridad: 1

Debe implementarse en el primer ciclo.

Caso de uso N° 2: Configuración del sistema

Valor de prioridad: 2

Debe implementarse en el primer ciclo.

Caso de uso N° 3: Hacer mímico

Valor de prioridad: 1

Debe implementarse en el segundo ciclo

Caso de uso N° 4: Configuración Alarmas

Valor de prioridad: 2

Debe implementarse en el segundo ciclo.

Caso de uso N° 5: Gestión de Tendencias

Valor de prioridad: 3

Debe implementarse en el segundo ciclo.

Caso de uso N° 6: Gestión de Reportes

Valor de prioridad: 4

Debe implementarse en el segundo ciclo.

Caso de uso N° 7: Control de Acceso

Valor de prioridad: 5

Debe implementarse en el segundo ciclo

Caso de uso N° 8: Visualización Procesos

Valor de prioridad: 6

Debe implementarse en el segundo ciclo.

4.1.13 Revisión y Evaluación de la fase de Estudio de Prefactibilidad

Se propone realizar cambios en varios casos de uso, ya que estos no reflejaban de forma clara y consistente la utilización por parte de un agente externo de un sistema que tiene como objetivos principales de interoperabilidad, la integración de un proceso y la supervisión de los datos. Sin embargo la realización de estos cambios se evidenciará mejor si se realiza primero una evaluación basada en el análisis de los escenarios de los casos de uso, de esta forma confirmar y realizar las correcciones, lo cual quedará consignado en la segunda fase de formulación del proyecto. Lo que hasta ahora se evidencia simplemente es una propuesta de arquitectura inicial, sin embargo es al formular el proyecto donde se notará de forma más explícita los cambios en los modelos iniciales.

4.2 FORMULACION DEL PROYECTO

4.2.1 Modelo Esencial De Casos De Uso Del Sistema

Diagrama de Casos de Uso del Sistema: inicialmente se plantean los casos de uso que a medida que avance el proyecto irán mejorando, mediante el estudio de los escenarios en una primera instancia y teniendo en cuenta los requerimientos y la satisfacción de estos.

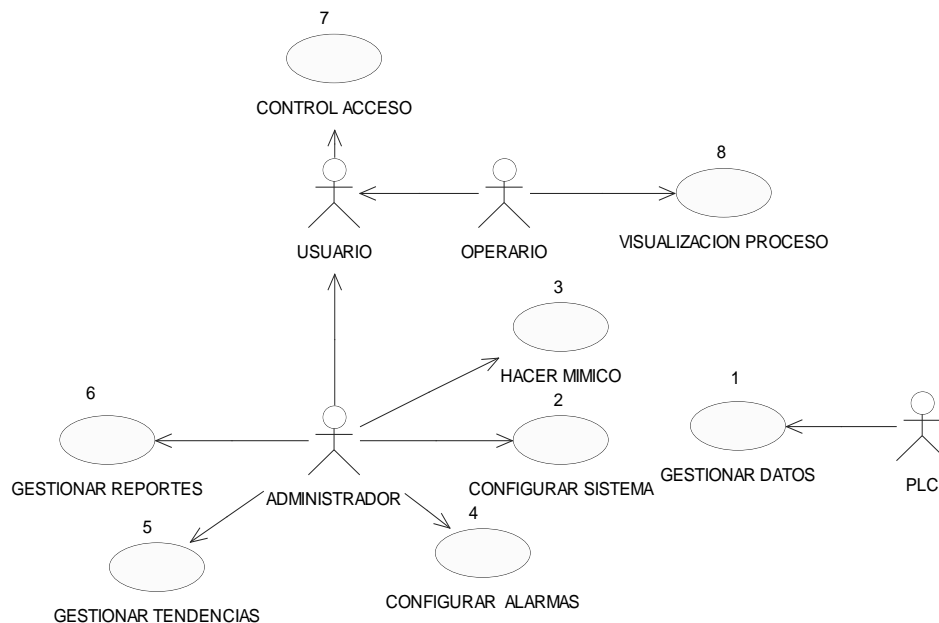


Figura 14. Casos de uso planteados inicialmente

4.2.2 Descripción de los Escenarios de los casos de Uso esenciales

Los escenarios muestran paso a paso como interviene el actor en el sistema y como responde este ante la solicitud del usuario, se determina de manera más extensa en cada caso de uso, su propósito, sus requerimientos y precondiciones para que continúe el flujo del sistema, por lo tanto se establece un análisis mayor que debe arrojar cambios que determinen si los requerimiento del sistema a construir están bien enfocados, resuelven los objetivo planteado y son una descripción valida y eficiente de la aplicación software a construir.

| | |
|---|---|
| Caso de uso No.7 : | Control de acceso |
| Propósito: | Validar el ingreso de los actores involucrados |
| Resumen: | Uno de los actores ingresará el nombre y la clave de acceso al Sistema. |
| Requerimientos: | Iniciar sistema |
| Pre-Condiciones: | El administrador configura la base de datos con información para acceso de usuarios. |
| ESCENARIO | |
| Actores | Sistema |
| 1. Administrador inicia el sistema y lo configura | 1.a. Configura el sistema, (con los demás casos de uso) |
| 2. El Administrador cierra la configuración (salir) | 2.a. Aparece una ventana de ingreso de Nombre, rol, login y password para el acceso como administrador u operario. |
| 3. El operario accede al sistema. | 3.a. El sistema valida el ingreso de operario, si antes no se ha configurado el sistema, el operario no podrá acceder |
| 4. El operario ingresa y escoge la visualización | 4.a Hay una pestaña para acceder a las diferentes visualizaciones. |
| Poscondición: | El administrador coloca en marcha el proceso de gestión, y luego el operario visualiza el proceso |
| Flujos Alternativos: Ingreso de login y/o password incorrecto. | |
| Actores | Sistema |
| 1. El operario puede cancelar la visualización. | 1a. El sistema vuelve a desplegar la pantalla de acceso. |
| 2. El administrador puede cancelar la configuración. | 2a. El sistema no se configura. |
| Excepciones: Ninguna | |
| Caso de uso No. 8: Visualizar Proceso | |
| Iniciador: | Operario |
| Propósito: | Mostrar mímicos con procesos en marcha |
| Resumen: | El operario abre los dos procesos los cuales muestran el comportamiento de los sistemas. |
| Requerimientos: | Generar los mímicos de los sistemas actuales |
| Pre-Condiciones: | <ul style="list-style-type: none"> - Control de acceso operario - Haber sido previamente realizada la configuración |
| ESCENARIO | |
| Actores | Sistema |
| 1. El operario después de haber accedido abre los procesos | 1.a. El sistema despliega los mímicos correspondientes a cada proceso. |
| 2. El operario puede utilizar los botones proceso 1 o proceso 2 | 2.a. Se despliega el proceso seleccionado por el operario |
| 3. El operario sale del mímico | 3.a. El sistema permite que el operario |

| | |
|---|--|
| | salga al presionar la tecla Salir, muestra nuevamente el control de acceso. 3.b. control de parada de emergencia, reanudar aun por definir |
| Poscondición: | El sistema despliega los mímicos de los procesos |
| Flujos Alternativos | |
| Actores | Sistema |
| El operario puede abrir un solo proceso | 1. Muestra el proceso elegido |
| Excepciones | |
| Actores | Sistema |
| El operario realiza Una parada de emergencia y reanuda el proceso | 1. Se manda una solicitud de parada, reanudar, el proceso. Salir del sistema mímico no implica parar los procesos. |
| Requerimientos funcionales: | no Ninguno hasta el momento |
| Caso de uso No. 5: | Configurar Sistema |
| Propósito: | Gestionar la configuración necesaria para adquirir los datos de los PLCs |
| Resumen: | El administrador establece la conexión escogiendo primero los servidores, luego estableciendo grupo, nodo, ítems. |
| Requerimientos: | - Configurar el sistema de Gestión. |
| Pre-Condiciones: | - Configurado por primera vez los parámetros de control de acceso o haber validado password como administrador - Debe existir una conexión física o emulada de los dispositivos y sus respectivos controladores (iniciar antes los programas remotos) |
| ESCENARIO | |
| Actores | Sistema |
| 1. El administrador busca el servidor (con Buscar) | 1.a. El sistema despliega el formulario para configuración |
| 2. El administrador elige servidor (Seleccionar) | 2.a. El sistema despliega los servidores disponibles |
| 3. El administrador elige servidor para el primer proceso, indicando Grupo, y otras (por definir) | 3. El sistema busca el primer servidor, (selecciona) |
| 4. El administrador conecta al primer servidor | 4. El sistema recibe los datos del primer proceso. |
| 5. El administrador elige servidor para el segundo proceso indicando su respectivo Grupo | 5. El sistema busca el segundo servidor |
| 6. El Administrador conecta (conectar) el segundo servidor indicando Grupo y nodo | 6. El sistema recibe datos del segundo proceso. |

| | |
|---|---|
| | |
| Poscondición: | Queda listo para realizar configuración de Tags y gestión de alarmas, eventos, y tendencia con los datos que se empiezan a recibir. |
| Flujos Alternativos | |
| Actores | Sistema |
| El Administrador puede cancelar la configuración | 1. No busca servidores |
| Excepciones | |
| Actores | Sistema |
| El Administrador no configura el nodo. | 1. No envía datos al sistema (Enviar Variable) |
| El administrador no encuentra servidores | 2. El sistema genera mensaje de no ubicación de servidores |
| El administrador encuentra servidores pero no recibe los datos | 3. El sistema genera mensaje de error 4. El sistema aborta la conexión |
| Caso de uso No. 4: Gestionar alarmas | |
| Iniciador: | Administrador |
| Propósito: | Suministrar las información a través de las tags de los rangos anormales y mostrar en el proceso cuando esto ocurra |
| Resumen: | Se configuran las alarmas o se colocan en una Lista para luego ser mostradas en una eventualidad por el proceso. |
| Requerimientos: | - Mostrar las alarmas |
| Pre-Condiciones: | - Control de acceso del Administrador - Haber realizado la conexión con éxito - Selección de las variables tomadas como alarmas |
| ESCENARIO | |
| Actores | Sistema |
| 1. el Administrador abre la ventana de Gestion en el menú Herramientas | 1.a. El sistema muestra una lista donde se pueden colocar las alarmas correspondientes |
| 2. El administrador visualiza los ítems y coloca los valores que ocasionarían eventos de alarmas según el proceso (adicionar). Luego las adiciona para ser mostradas aparte. Elige de cada PLC las que serán tomadas como Alarmas | 2.a. El sistema guarda el ítem seleccionado como alarma para el primer PLC. 2.b. El sistema guarda el segundo grupo seleccionado (ventana del PLC 2) como alarmas y las lleva a la ventana EstadoAlarmas correspondiente según sea el PLC. |
| 3. El administrador muestra las alarmas (Mostrar) para cada proceso, en esta ventana el administrador al haber previamente escogido los ítems puede configurar el valor para el cual la alarma | 3.a. El sistema muestra los ítems seleccionados previamente por el Administrador y responde a la configuración del valor máximo, mínimo o eventual para el cual se debe atender la alarma. |

| | | |
|---|--|--|
| se activara y la prioridad que esta tiene | | 3.b. cada alarma seleccionada y su valor (máx. o min.) será mostrado en los procesos |
| 4. El administrador al hacer clic y coloca los valores, La seleccionara (seleccionar). | | 4.a. El sistema realiza una comparación con el valor colocado en las cajas de texto con el valor que tiene el item seleccionado previamente y cuando sea el valor debe suceder un evento de aviso. |
| 4.El administrador aplica salir | | 4.El sistema va a la pantalla anterior en cada caso |
| Poscondición: | El sistema debe avisar en el proceso (1 o 2) y en la lista respectiva su estado. | |
| Flujos Alternativos | | |
| Actores | | Sistema |
| El Administrador cierra la ventana de Alarmas | | El sistema no recibe datos de configuración de alarmas. |
| El administrador elimina en la pantalla de Alarmas las que no corresponden al suceso. | | El sistema no recibe los datos de esta alarma |
| Excepciones | | |
| Actores | | Sistema |
| El Administrador nota una alarma y la establece como alarma atendida.(con el fin de establecer si fue atendida o no) | | 1 .a. El sistema prioriza las alarmas según rangos dados en la configuración. 1.b. El sistema cambia la prioridad de la alarma atendida. |
| Requerimientos funcionales: | no | Ninguno hasta el momento |

Tabla 6. Escenarios de los casos de uso esenciales

4.2.3 Revisión y Evaluación de los Casos de Uso Iniciales

En este punto se puede evidenciar, a través del estudio de los escenarios de los casos de uso iniciales presentados en la Figura 14, varios cambios, entre los cuales se tienen:

- Para el **caso de uso No 8 Visualizar Proceso**, se determinó que si se quiere cumplir con el objetivo de interoperar dos dispositivos de diferente fabricante, se mostraría de mejor manera configurando un solo proceso donde se evidencie la interoperación de los dos dispositivos y no dos procesos por aparte como se describe en los escenarios en la Tabla 6. Partiendo de este requerimiento el caso de uso y su propósito cambian radicalmente.
- En el **caso de uso No 5 Configurar Sistema** se observa que no capta de forma clara los requerimientos del sistema OPC, ya que, si es cierto que se necesita en primer

lugar la configuración del sistema, sin embargo este caso de uso no describía que clase de configuración o que elementos en la configuración son necesarios, podrían entonces dejarse de captar casos de uso que describieran mejor la utilidad del sistema desde la perspectiva del usuario [10].

- El **caso de uso No 1 Gestionar datos** (Figura 14) no es un caso de uso representativo de los requerimientos buscados (los casos de uso modelan implícitamente requerimientos [10]), ya que verdaderamente la norma establece las interfaces para el acceso a datos tanto de una aplicación cliente, como de una aplicación servidor; y es la aplicación particular la que establece los elementos de implementación final.
- El actor PLC o PLCs muestra la frontera del sistema pero a pesar de que recibe o entrega eventos al sistema, este actor no desempeña en los casos de uso mencionados un papel de iniciador, es decir no interviene de forma directa utilizando el sistema para su funcionamiento, si no que el sistema utiliza al PLC para su funcionamiento. Esto determina que para la explicación de los casos de uso este actor será considerado implícitamente [10].
- Se debe aclarar que se muestran en general en los diferentes diagramas, los que se consideran más importantes y relevantes para explicar el sistema, puesto que de lo contrario se perdería el objetivo de dar a entender las diferentes vistas del sistema y se saturaría la explicación [10].
- No se ha tenido en cuenta hasta ahora en los escenarios la interacción con una base de datos, lo cual es determinante en la dinámica del sistema.

4.2.4 Diagrama General de casos de Uso Mejorados.

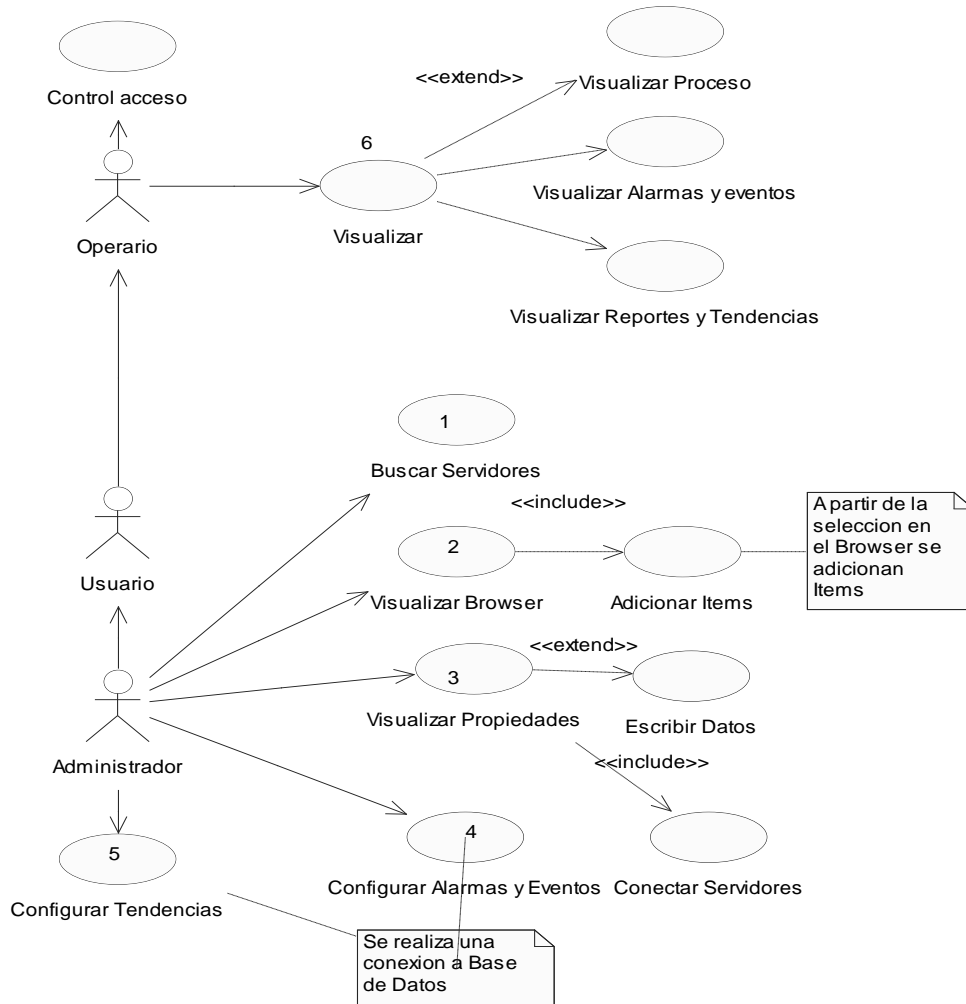


Figura 15. Casos de uso mejorados

La Figura 15 muestra un cambio en los diagramas planteados en la Figura 15 del estudio de prefactibilidad, producto del estudio de los escenarios (Tabla 6) los cuales mostraban de manera muy superficial el sistema. En este diagrama se trata de reunir los requerimientos del sistema de una forma mas clara y captando efectivamente los objetivos del proyecto propuesto. Se realiza a continuación una descripción detallada de los escenarios de los casos de uso mejorados

Descripción de los Escenarios de los casos de Uso mejorados.

| | | |
|---|--|--|
| Caso de uso No.1 | Buscar Servidores | |
| iniciador | Administrador. | |
| Propósito: | Buscar los servidores que se encuentran instalados en los nodos. | |
| Resumen: | Se solicita al sistema que busque los servidores disponibles y sean desplegados en una lista para seleccionar dos que serán conectados y se obtendrán sus datos por parte del sistema para posterior manejo. | |
| Requerimientos: | Mostrar la lista de servidores actual. | |
| Pre-Condiciones: | Se debe haber instalado los servidores OPC de cada PLC, además haber accedido al sistema como administrador. | |
| ESCENARIO | | |
| Actores | Sistema | |
| 1. Administrador entra y busca Servidores | 2. El sistema despliega una lista de servidores que se están ejecutando actualmente. | |
| 3. el administrador selecciona de la lista de servidores OPC, dos (2) de cada respectivo PLC. | 4. El sistema guarda los servidores seleccionados. | |
| Poscondición: | El sistema se conecta mostrando su estado. | |
| Flujos Alternativos | | |
| Actores | Sistema | |
| 1. El administrador sale de la selección de servidores | 2. El sistema despliega nuevamente el control de acceso. | |
| Excepciones | | |
| Actores | Sistema | |
| 1. El administrador no seleccionada ningún servidor. | 2. El sistema no es configurado. (posible mensaje "Sistema no configurado") | |
| 3. Si el administrador se equivoca en la selección de servidores OPC, debe salir y volver a elegir. | 3a. El sistema sale a la vista donde se puede volver a buscar servidores para que el administrador establezca la escogencia correcta. | |
| Caso de uso No. 2 | Visualizar Browser. | |
| Iniciador | Administrador | |
| Propósito: | Observar las direcciones e ítems de cada servidor OPC según las Tags de cada PLC (observar e forma de árbol los datos obtenidos de cada PLC) | |
| Resumen: | Se muestran la diferentes ramas y hojas de cada raíz de los servidores seleccionados | |
| | | |

| | |
|--|---|
| Requerimientos: | - Visualizar las variables de los servidores OPC según cada PLC. |
| Pre-Condiciones: | - Debe existir una conexión física o emulada de los dispositivos y sus respectivos controladores (iniciar antes los programas remotos) y establecer la conexión con los servidores y procesos ya sea emulados o reales. |
| ESCENARIO | |
| ACTORES | Sistema |
| 1. El administrador abre cada raíz de cada servidor y cada rama donde se encontraran los ítems o datos obtenidos del servidor, esto se realizara para cada servidor. | 1. El sistema despliega todos los datos obtenidos de los servidores en forma de árbol. (se coloca 1a. por que el sistema responde a la vez que el usuario abre cada rama |
| 2. El administrador busca y selecciona las variables que va a utilizar de cada servidor para el proceso determinado | 3. El sistema guarda las variables seleccionadas según la categoría (alarmas o eventos) |
| Poscondición: | Queda listo para realizar configuración de Tags y gestión de alarmas, eventos, y tendencia con los datos que se empiezan a recibir. |
| Flujos Alternativos | |
| Actores | Sistema |
| 1. El administrador cancela el Browser | 2. El sistema vuelve a menú principal. |
| Excepciones | |
| Actores | Sistema |
| 1. Si el administrador se equivoca al seleccionar las variables correspondientes puede eliminar y volver a adicionar las correctas | 1. El sistema elimina las variables que no han sido correctamente seleccionadas y guarda la nueva selección realizada por el administrador. |
| Caso de uso No 3 Visualizar Propiedades | |
| Iniciador | Administrador |
| Propósito | Observara las propiedades de los Ítems de cada servidor y realizar la escritura. |
| Resumen | Los ítems se establecen en una tabla donde se observan sus propiedades (valor, time, calidad etc.) |
| Requerimientos | Mostrar Ítems y escribir valores |
| Precondiciones | Haber seleccionado y adicionado los Ítems según sea el proceso a supervisar |
| ESCENARIO | |
| ACTORES | Sistema |
| 1. El administrador visualiza las propiedades y elimina el ítem que no corresponda al proceso que se va supervisar | 2. El sistema muestra las propiedades de los ítems. Elimina ítems. |
| 3. El administrador escribe a una dirección específica hacia un PLC determinado. | 4. El sistema escribe en una dirección determinada del servidor o servidores. |

| | |
|--|---|
| Poscondiciones: | La variable escrita cambia de valor. |
| Flujo Alternativo | |
| Actores | Sistema |
| 1. El administrador cancela la visualización de propiedades | 2. El sistema vuelve al formulario principal. |
| Excepciones | |
| 1. El administrador debe digitar bien la dirección específica hacia los servidores | 2. El sistema avisa si no se ha digitado bien la dirección o en caso alternativo simplemente no se escribe el valor y por lo tanto no se observa un cambio de valor en la variable. |
| Interfaz requerida | |
| Caso de uso No. 4: | Configurar alarmas y eventos |
| Iniciador: | Administrador |
| Propósito: | Suministrar las información a través de las tags de los rangos anormales y mostrar en el proceso cuando esto ocurra |
| Resumen: | Los ítems seleccionados como alarmas son configurados según su estado o nivel crítico. Se configuran las alarmas y se colocan en una Lista para luego ser mostradas en una eventualidad por el proceso. |
| Requerimientos: | - Mostrar las alarmas |
| Pre-Condiciones: | <ul style="list-style-type: none"> - Control de acceso del Administrador - Haber realizado la conexión con éxito - Selección de las variables tomadas como alarmas |
| ESCENARIO | |
| Actores | Sistema |
| 5. el Administrador abre la ventana de Alarmas y eventos en el menú Herramientas | 2. El sistema despliega los ítems que han sido seleccionados como alarmas o eventos. |
| 2. El administrador selecciona los ítems tomados como alarmas y los relaciona con un color según el nivel crítico (alto-alto, leve, moderado etc.). El administrador visualiza los ítems que ocasionarían alarmas según el proceso. Luego las adiciona para ser mostradas en el instante que ocurran en el supervisorio. | 4. El sistema relaciona cada ítem con un color o atributo. Son guardados los ítems (direcciones) en una base de datos según sea su función o el requerimiento posterior por parte del sistema o del usuario (supervisorio). |
| Poscondición: | El sistema debe avisar en el proceso y en la lista respectiva su estado. |
| Flujos Alternativos | |
| Actores | Sistema |
| 1.El Administrador cierra la ventana de Alarmas | 2. El sistema no recibe datos de configuración de alarmas. |
| 3. El administrador elimina en la pantalla de Alarmas las que no corresponden al | 4. El sistema no recibe los datos de esta alarma |

| | | |
|--|---|---|
| proceso. | | |
| Excepciones | | |
| Actores | | Sistema |
| 1. El Administrador nota una alarma y la establece como alarma atendida.(con el fin de establecer si fue atendida o no) | | 2. El sistema prioriza las alarmas según rangos dados en la configuración. 2.b El sistema cambia la prioridad de la alarma atendida. |
| Requerimientos funcionales: no | | Ninguno hasta el momento |
| Interfaz requerida : falta | | |
| Caso de uso no 7 | | Control de acceso |
| Propósito: | Validar el acceso de los usuarios del sistema | |
| Resumen: | Uno de los actores entrara el nombre y la clave de entrada | |
| Requerimientos: | Iniciar sistema | |
| Pre-Condicion: | El administrador antes debe haber configurado el sistema. | |
| ESCENARIO | | |
| Actores | | Sistema |
| 1. Administrador abre el sistema y configura | | 1.a. Configura la conexión, luego las demás configuraciones (ver caso uso 1,2, 3, 4) |
| 2. El Administrador acepta la configuración realizada y sale del menú de configuración | | 2.a. aparece una ventana pidiendo usuario y contraseña para volver a entrar como administrador u operario. |
| 3. El operario accede al sistema. | | 3.a. El sistema valida el ingreso de operario, si antes no se ha configurado el sistema, el operario no podrá acceder. |
| 4. El operario entra y elige la visualización | | 4.a. Hay una pestaña para acceder a las diferentes visualizaciones. |
| Poscondición: | El administrador coloca en marcha el proceso de gestión, y luego el operario visualiza el proceso | |
| Flujos Alternativos | | |
| Actores | | Sistema |
| 1. El operario puede cancelar la visualización. | | 1a. El sistema vuelve a desplegar la pantalla de acceso. |
| 2. El administrador puede cancelar la configuración. | | 2a. El sistema no se configura. |
| Excepciones | | |
| Actores | | Sistema |
| 1. El operario entra con nombre y | | 1a. El sistema despliega nuevamente la |

| | | | |
|---|---|--|--|
| contraseña pero el sistema no ha sido configurado | | pantalla de acceso y da el aviso de que no hay una configuración predeterminada | |
| 2. El administrador cancela la configuración | | 2a. El sistema no es iniciado. | |
| Caso de uso No. 6 | Visualizar | | |
| Iniciador: | Operario | | |
| Propósito: | Mostrar mímicos con procesos en marcha y demás funciones como reportes, tendencias. | | |
| Resumen: | El operario abre el proceso y visualiza el comportamiento del proceso, además puede imprimir reportes, tendencias | | |
| Requerimientos: | - Generar los mímicos de los sistemas actuales | | |
| Pre-Condiciones: | <ul style="list-style-type: none"> - Control de acceso operario - Haber sido previamente realizada la configuración | | |
| ESCENARIO | | | |
| Actores | | Sistema | |
| 1. El operario después de haber accedido abre el proceso | | 1. El sistema despliega los mímicos correspondientes al proceso y muestra el menú para las demás funciones de supervisión. | |
| 2. El operario puede visualizar reportes, tendencias, alarmas, realizar paradas de emergencia y reanudar. | | 2. a Se despliega el proceso seleccionado por el operario. 2. b Según la selección el sistema puede desplegar las alarmas, visualizar reportes, imprimir reportes, visualizar tendencias. 2. c El sistema realiza parada de emergencia o reanudar según haya sido llamado este evento por el operario. | |
| Poscondición: | El sistema despliega el menú para visualización según la configuración del administrador. | | |
| Flujos Alternativos | | | |
| Actores | | Sistema | |
| 1. El usuario sale de la visualización | | 2. El sistema vulva a control de acceso. | |
| Excepciones | | | |
| Actores | | Sistema | |
| 1. El operario para (emergencia), reanuda el proceso | | 2. Se manda una solicitud de parada, reanudar, el proceso. Salir del sistema mímico no implica parar los procesos. | |
| Requerimientos funcionales: | no | Ninguno hasta el momento | |

| | | |
|--|---|--|
| Caso de uso No. 5 | Tendencias y reportes | |
| Iniciador | Administrador | |
| Propósito: | Configurar las tendencias y reportes del proceso. | |
| Resumen: | El administrador ha seleccionado los ítems correspondientes al proceso y desea graficar tendencias e imprimir reportes. | |
| Requerimientos: | Graficar variables en el tiempo y generar reportes. | |
| Pre-Condiciones: | Se debe haber realizado una selección previa de las variables para estos requerimientos. | |
| ESCENARIO | | |
| ACTORES | | Sistema |
| 1. El administrador selecciona graficar una variable vs. El tiempo u otra variable que pueda determinar un comportamiento evidenciable o una tendencia a tener en cuenta. (se ha comunicado con una base de datos donde están guardadas las variables para esta función) | | 2. El sistema grafica desplegando en un plano x-y la tendencia configurada de las variables respectivas. |
| 3. El administrador guarda las gráficas | | 4. El sistema guarda estas gráficas para mostrarlas en el sistema de supervisión. |
| 5. El administrador configura la generación de reportes según las funciones de alarmas o tendencia o eventos específicos de variables guardadas en una base de datos (según la clase son guardadas en base de datos específicas) | | 6. El sistema genera reportes de las variables seleccionadas. |
| Poscondición: | | Reportes y tendencias guardadas y listas para ser vistas por el operario. |
| Flujos Alternativos | | |
| Actores | | Sistema |
| 1. El Administrador puede cancelar la configuración de reportes y tendencias | | 2. El sistema vuelve al menú principal. |
| Excepciones | | |
| Actores | | Sistema |
| 1. El administrador cambia la variable a graficar, elimina, edita etc. | | 2. El sistema responde ante el cambio de variables y elimina la anterior variable a graficar. |

Tabla 7. Descripción de los escenarios de los casos de uso mejorados.

4.2.5 Diagramas generales de Secuencia.

Los diagramas de secuencia generales salen de los casos de uso; en particular del estudio de los **escenarios** donde al describir cada escenario para el actor, se indican operaciones que este solicita y la respuesta del sistema. Cada operación causa en el sistema un estado posterior o poscondición. El objetivo es determinar que hace el sistema y no como lo hace [10]. Luego se debe determinar de forma concreta lo que debe realizar cada **operación**, esto se realiza a través de los **contratos**, definiendo responsabilidades, y analizando de manera especial las poscondiciones. Los contratos se realizan teniendo en cuenta la relación cruzada con las funciones definidas en la Tabla 8 y Tabla 9.

En general la metodología para determinar las operaciones sale de la sintaxis de la descripción de los escenarios, al captar los verbos que indican ciertas operaciones de entrada al sistema, tomando este como una caja negra, por ejemplo en el caso de uso No 1, Tabla 7 "El actor busca los servidores", esto se convierte en la operación "buscarServidores".

Los contratos a su vez determinarán las posibles relaciones, asociaciones, creación y en general la dinámica de los objetos del sistema como se verá más adelante.

También es importante no saturar la explicación de las diferentes vistas del modelado, puesto que muchas veces se pierde el objetivo de dar a entender claramente el sistema a realizar, por lo tanto se describirán los casos de uso que se consideran más importantes [10]. Los siguientes diagramas mostrados en la figuras, muestran al Actor efectuando una operación que entra al sistema, tomado como una caja negra, luego en los diagramas de colaboración se mostrará la respuesta del sistema ante estas operaciones.

Diagrama de Secuencia Caso de uso No 1 Buscar Servidores

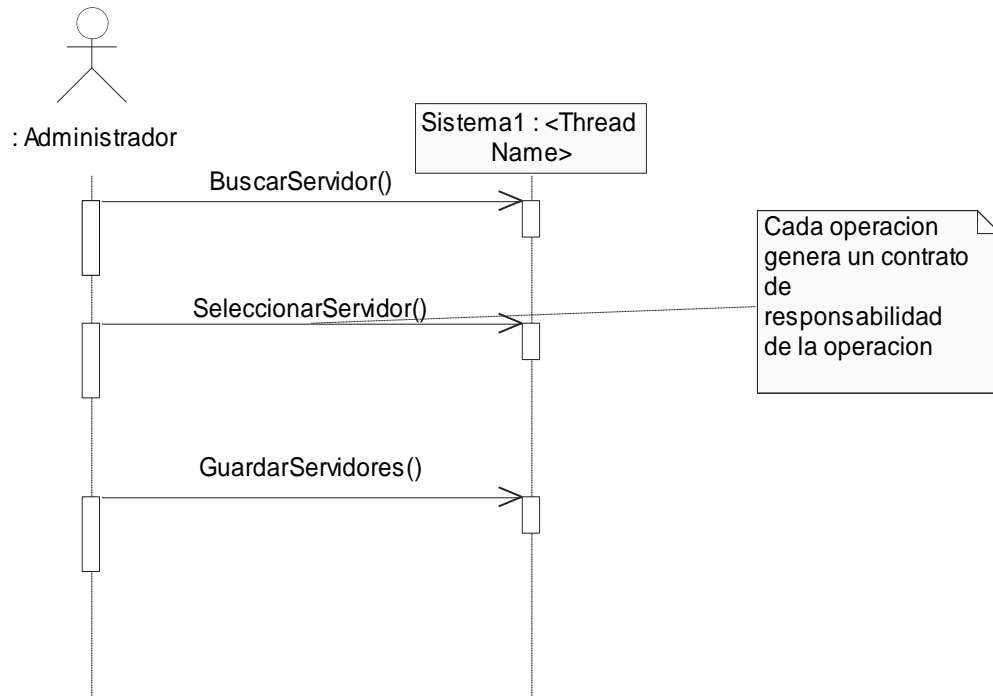


Figura 16. Secuencia Buscar Servidores

La figura anterior muestra como el administrador entra a configurar el sistema, y necesita en primera instancia buscar los servidores disponibles en el equipo, por lo tanto requiere de una lista donde se pueda elegir dos servidores (siguiendo el requerimiento de interoperabilidad), paso seguido necesita guardar estos servidores o nombres de los servidores en una base de datos para poder posteriormente realizar otros requerimientos funcionales.

Diagrama de Secuencia Caso de uso No 2 Visualizar Browser.

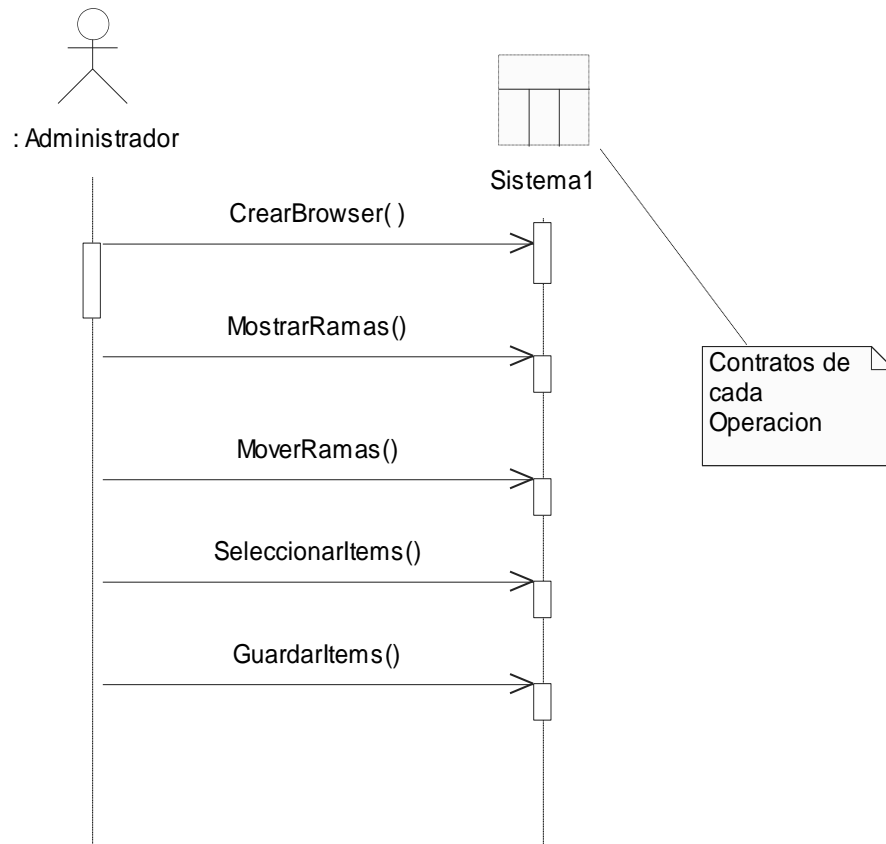


Figura 17. Secuencia Visualizar Browser

La figura anterior describe una secuencia especial, pues la función que se quiere determinar es una especie de buscador en forma de árbol descrito en la especificación OPC de interfaz de automatización 2.02, que facilita la exploración de los datos de los servidores. El administrador realizará estas operaciones a medida que selecciona con el Mouse, las diferentes raíces, ramas u hoja del árbol. Luego al visualizar los ítems podrá seleccionar los que necesite para posteriores usos o configuraciones especiales.

Diagrama de Secuencia Caso de uso No 3 Visualizar Propiedades

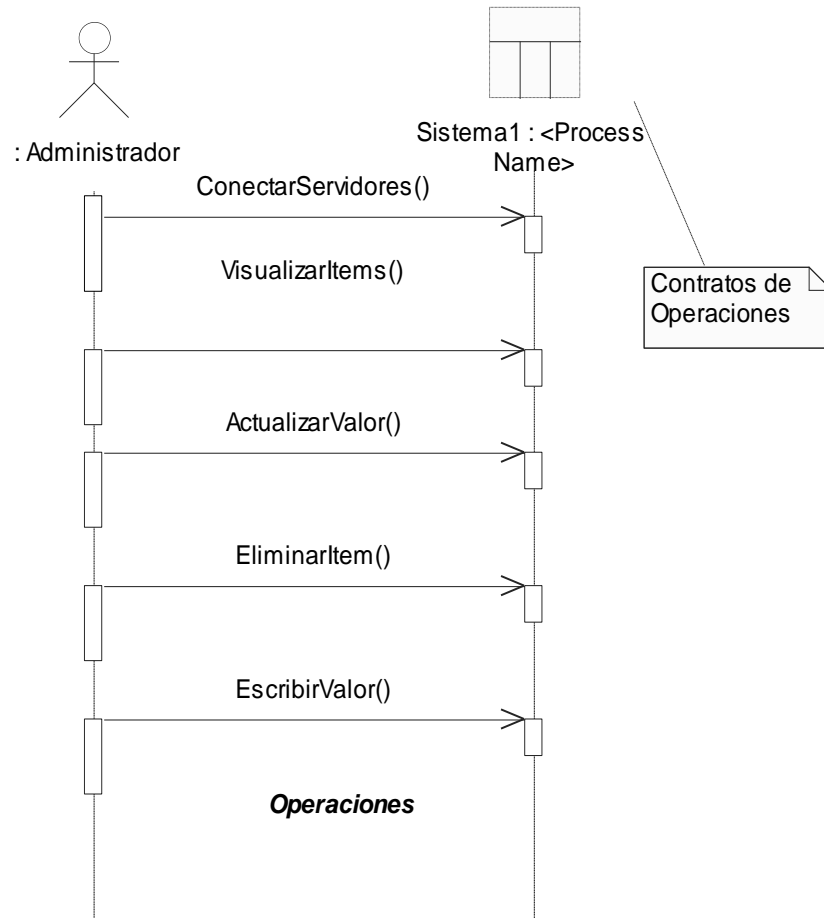


Figura 18. Secuencia Visualizar Propiedades.

La anterior figura en esencia quiere visualizar las propiedades de los ítems seleccionados (su valor, calidad, dirección y tiempo) en el caso de uso No 3, para esto tendrá que realizar primero una lectura, además podrá efectuar edición de estos ítems.

Diagrama de Secuencia Caso de uso No 4 Configurar Alarmas y Eventos

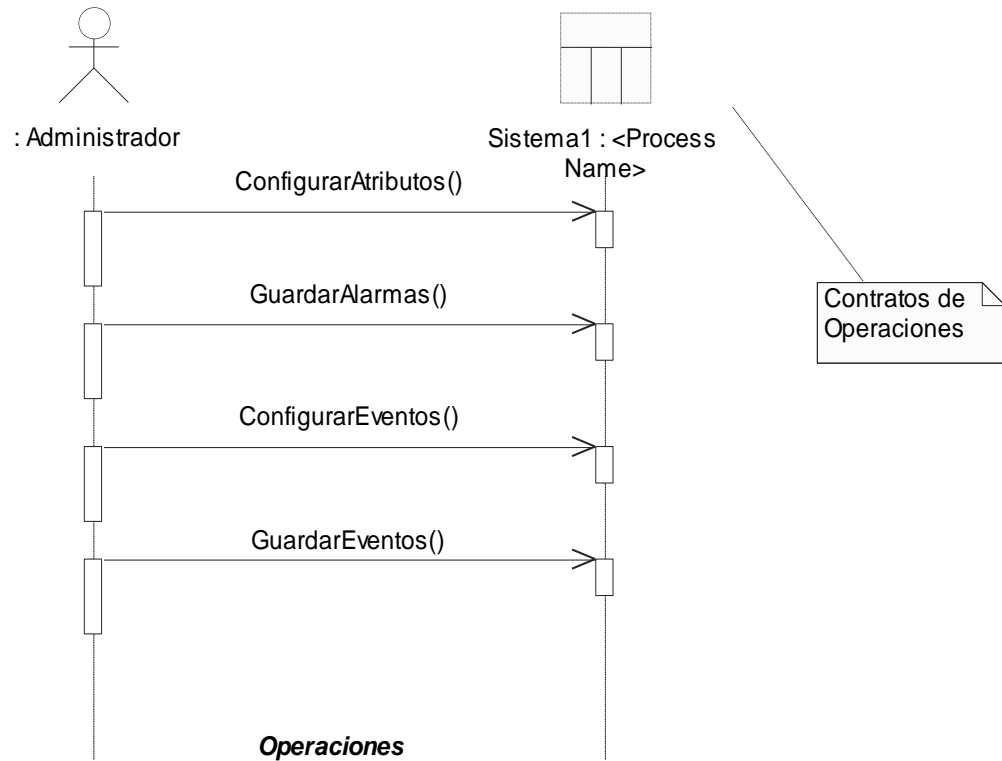


Figura 19. Secuencia Configurar Alarmas y Eventos

La anterior figura muestra en general lo que se pretende realizar para configurar Alarmas y eventos. Al obtener los datos de los servidores se pretende seleccionar los que son considerados alarmas de acuerdo al proceso que controla cada PLC y guardar en una tabla diferente asignando propiedades especiales como nombre, estado, nivel de prioridad, logrando así configurar determinado ítems como alarmas del proceso que validará la aplicación.

Diagrama de Secuencia Caso de uso No 5 Configurar Tendencias

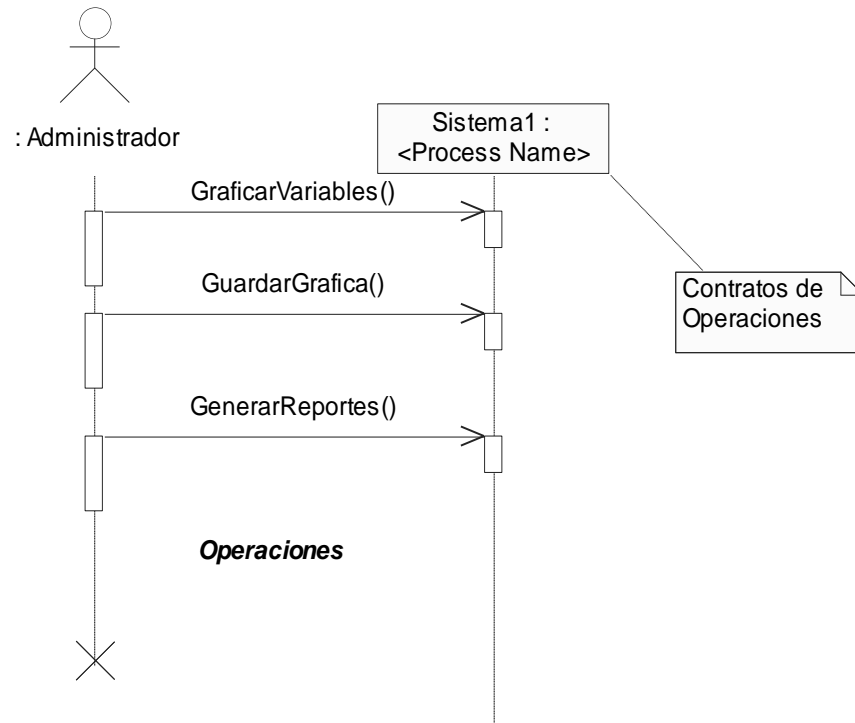


Figura 20. Secuencia Configurar Tendencias.

La anterior figura muestra la secuencia que pretende tomar un ítem específico del proceso (por Ej.; temperatura, llenado de un tanque, presión etc.) y configurarlo para graficar el comportamiento de esta variable en el tiempo.

Cada operación tiene un contrato específico (con un formato especial) que mostrará las responsabilidades de cada operación, las poscondiciones y excepciones; lo cual conducirá a la formación de diferentes objetos y su correspondiente interacción.

4.2.6 Interfaces Requeridas

Basados en el estudio de los escenarios Tabla 6 y Tabla 7, se definen las interfaces requeridas por el usuario para interactuar con el sistema y acceder a sus funciones. Los diagramas de secuencia generales (punto anterior), ayudan a concretar que tipo de servicios deben tener estas interfaces. Las siguientes figuras muestran las interfaces requeridas.

- Interfaces Caso de Uso No 1 Buscar Servidor

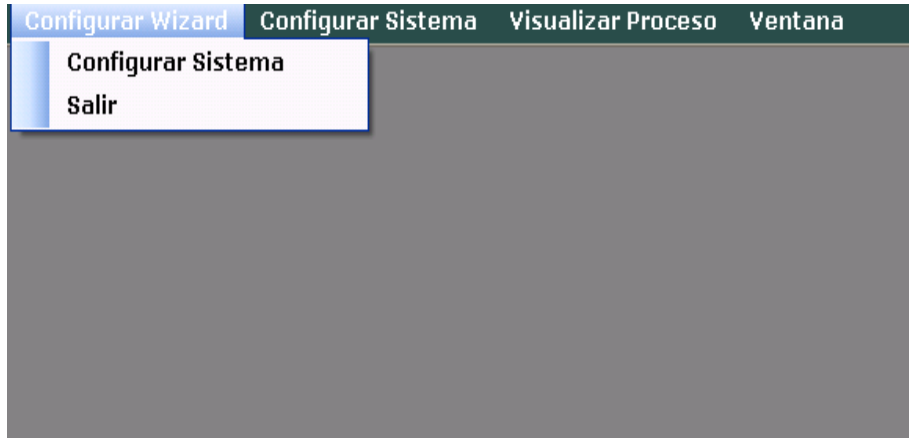


Figura 21. Interfaz frmPrincipal.



Figura 22. Interfaz Buscar Servidores

- Interfaces Caso de Uso No 2 Visualizar Browser

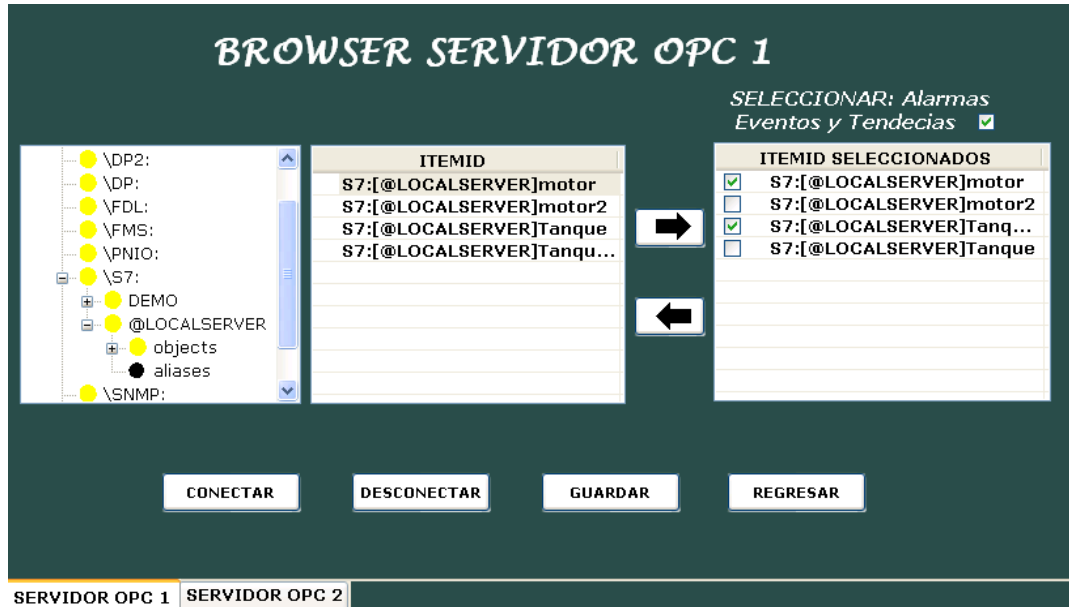


Figura 23. Interfaz Visualizar Browser

- Interfaces Caso de Uso No 3 Visualizar Propiedades

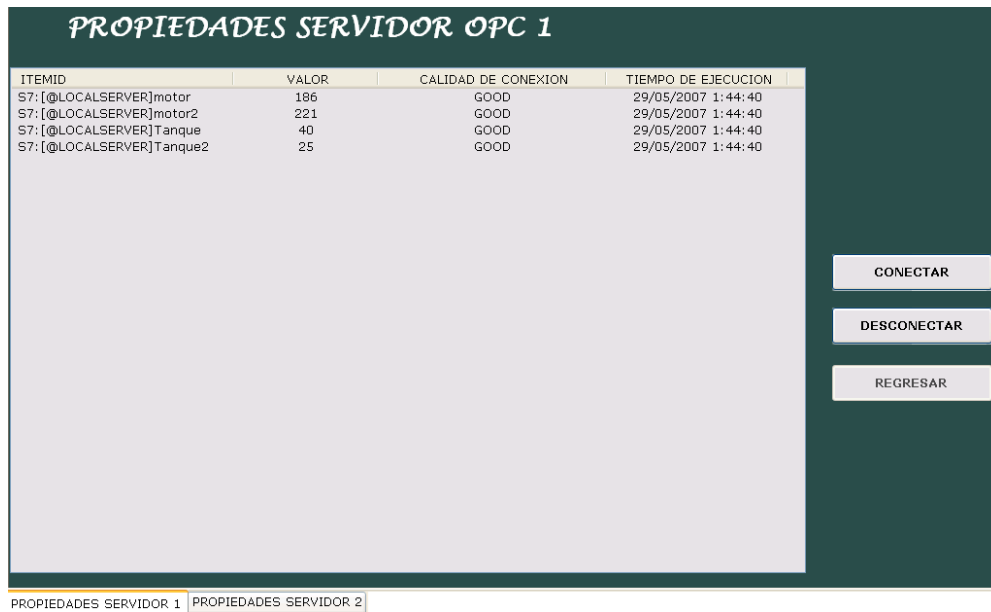


Figura 24. Interfaz Visualizar Propiedades

- Interfaces Caso de Uso No 4 Configurar Alarma y Eventos

CONFIGURAR ALARMAS

ITEMID

ALARMA

ADICIONAR

NOMBRE

NIVEL

RANGOS

BAJO NORMAL

ALTO NORMAL

GUARDAR

REGRESAR

ALARMAS | EVENTOS | TENDENCIAS

Figura 25. Interfaz Configurar Alarmas y Eventos

CONFIGURAR TENDENCIAS

ITEMID

TENDENCIA

GRAFICA

NOMBRE

GUARDAR

SALIR

ALARMAS | EVENTOS | TENDENCIAS

Figura 26. Interfaz Configurar Eventos

- Interfaces Caso de Uso No 5 Configurar Tendencias

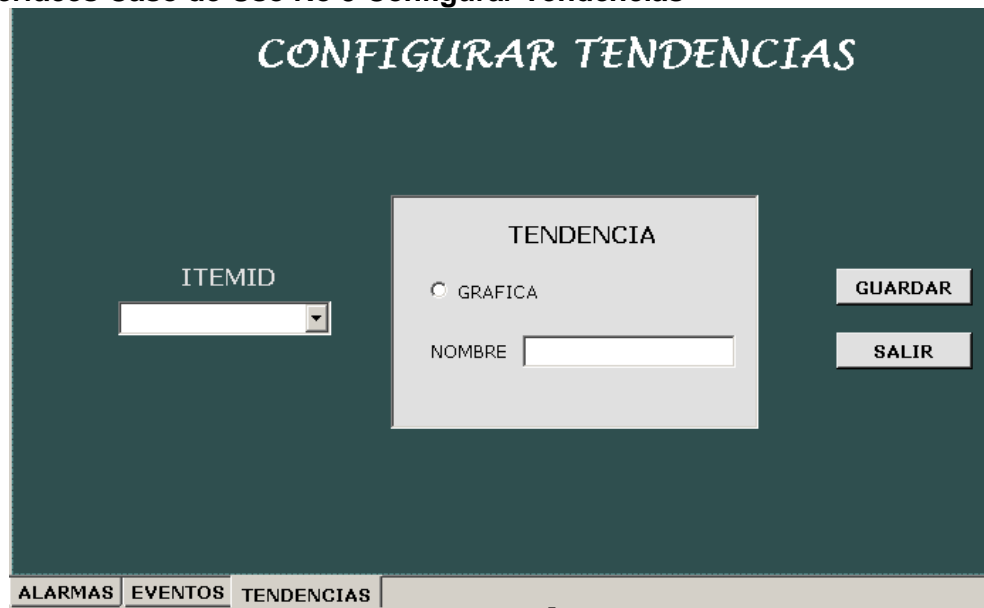


Figura 27. Interfaz Configurar Tendencia

- Interfaces Caso de Uso No 6 Visualizar

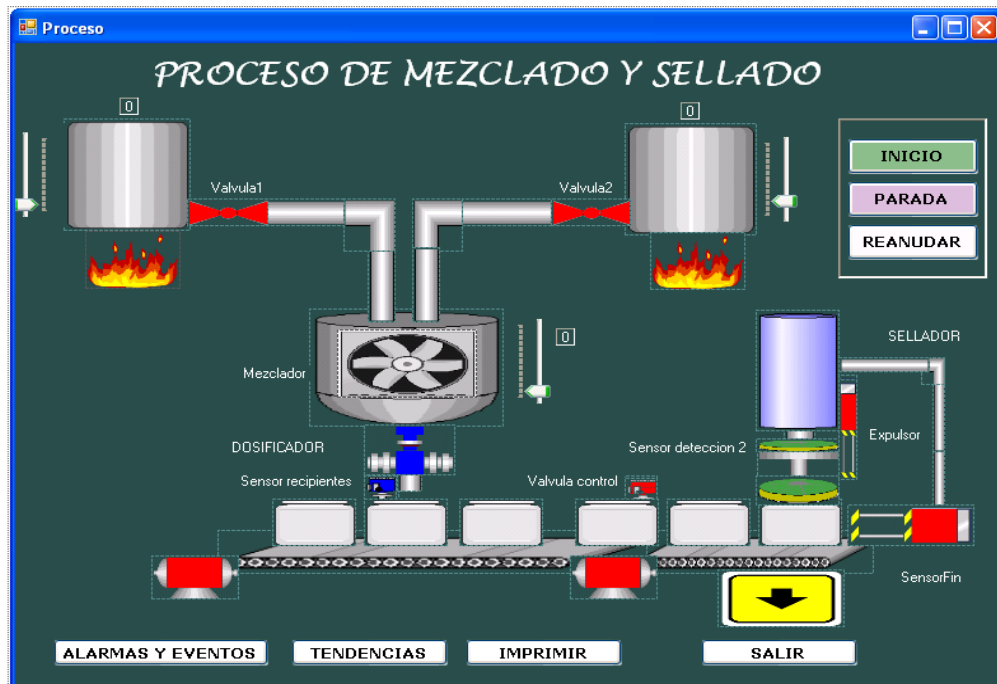


Figura 28. Interfaz Visualizar Proceso

- Interfaces Caso de Uso No 7 Control Acceso



Figura 29. Interfaz Control Acceso

4.2.7 Contratos de las Operaciones Según cada Caso de Uso.

Debido a que cada operación establece un cambio en el sistema y la respuesta de este establece una poscondición, es importante que estas estén enmarcadas dentro de unas funciones que realiza el sistema específicamente (ver Tabla 8 y Tabla 9), y que referenciadas en cada contrato explican concretamente lo que se requiere de cada operación. Las tablas siguientes establecen estas funciones requeridas de la aplicación OPC, divididas entre funciones básicas y funciones de supervisión (SCADA), tomando también como referencia el estudio de Pre-factibilidad (Características Esenciales) en la Tabla 2.

FUNCIONES BASICAS OPC.

| REFERENCIA | FUNCION | CATEGORIA |
|------------|---|-----------|
| R1.1 | Selección de servidores OPC locales | Evidente |
| R1.2 | Conectar al servidor seleccionado | Evidente |
| R1.3 | Captura información sobre los servidores seleccionados | Evidente |
| R1.4 | Introducción de contraseña e identificación del usuario | Evidente |
| R1.5 | Ofrece mecanismos de comunicación entre el sistema y los procesos | Ocultas |
| R1.6 | Ofrece mecanismos de almacenamiento persistente | Ocultas |
| R1.7 | Mostrar los datos obtenidos para posterior manipulación. | |

Tabla 8. Funciones Básicas OPC

FUNCIONES DE SUPERVISION.

| REFERENCIA | FUNCION | CATEGORIA |
|------------|--|-----------|
| R2.1 | Mostrar mímico de el proceso | Evidente |
| R2.2 | Gestionar configuración de los datos según categoría específica de alarmas o eventos. | Evidente |
| R2.3 | Establecer funciones para graficar las tendencias del dato seleccionado en esta categoría. | Evidente |
| R2.4 | Registrar los datos para establecer reportes. | Evidente |

Tabla 9. Funciones de supervisión OPC.

Se necesita especificar concretamente las responsabilidades de cada operación, acercándose el sistema a **como hace y como responde ante cada operación**. En la siguiente tabla (que es un formato de contrato) se explican estas operaciones según cada caso de uso. Del análisis de esta tabla saldrá el comportamiento dinámico del sistema, el contrato constituirá una gran herramienta de análisis e investigación, pues permite describir los cambios necesarios para que el sistema funcione sin necesidad de describir como se logran. Se centrará el análisis en lo que debe suceder pero aun no en la manera de conseguirlo, por lo tanto las poscondiciones descritas en cada contrato jugarán un papel determinante [10].

| | |
|--|---|
| Caso de uso No 1. Operaciones | a) BuscarServidores() b) SeleccionarServidores() |
| Responsabilidades | Buscar los servidores disponibles remotos o locales. |
| Tipo | Sistema |
| Referencias cruzadas | R1.1 |
| Excepciones | <ul style="list-style-type: none"> • Es la primera opción que debe realizar el administrador para que el sistema empiece a ser configurado. • Si el administrador se equivoca en la selección de los servidores |
| Precondiciones | Deben estar instalados los respectivos servidores a ínter operar. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Si se buscaron los servidores, se desplegó lista de servidores OPC encontrados y se creó una instancia ServidoresOPC que devuelve el nombre de los servidores. • Si se seleccionaron dos servidores, se establece una opción guardar en base de datos (enlace) para guardar los nombres de los servidores a conectar posteriormente. | |
| *Se determinaron las dos operaciones , pues están muy relacionadas | |

| | |
|---|---|
| Caso de uso No 1. Operación | GuardarServidores() |
| Responsabilidades | Realizar el enlace con una base de datos para guardar los nombre de los servidores seleccionados (2). |
| Tipo | Sistema |
| Referencias cruzadas | R1.1, R1.6 |
| Excepciones | La conexión a la base de datos debe ser establecida correctamente |
| Precondiciones | Deben estar instalados los respectivos servidores a ínter operar. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Si se guardaron servidores se realizó un enlace con una base de datos que guardó el nombre de los servidores | |

| | |
|--|---|
| Caso de uso No 2. Operación | Browser() |
| Responsabilidades | Visualizar en forma de árbol los diferentes elementos del sistema como son Servidores OPC, direcciones de memoria de cada Servidor e Ítems de cada PLC. |
| Tipo | Sistema interno (externo será llamada en general Browser) |
| Referencias cruzadas | R1.1, R1.3, R1.6. |
| Excepciones | |
| Precondiciones | Selección de dos servidores OPC |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se estableció un enlace con la base de datos para que devuelva el nombre de los servidores seleccionados y guardados en el caso de uso anterior. • Si se muestran los elementos en forma de árbol en un caso parcial los servidores se ha establecido una instancia ServidoresOPC, que envía la información correspondiente a cada servidor. (hay una conexión implícita). • Se establecieron los diferentes Grupos e ítems para cada uno de los servidores (según especificación).tanto para Siemens como Allen Bradley(por validación y correspondencia con los objetivos se trabajara con estos dos servidores OPC) • Si hay equivocación por parte del administrador al seleccionar los ítems se establece una conexión con una operación eliminar ítems que no elimina los ítems que no corresponden al proceso; se podría realizar en la tabla que aparece en el Browser o en este caso de uso. | |

| | |
|--------------------------------------|--|
| Caso de uso No 2. Operaciones | a) MostrarRamas() b) MoverRamas() c) SeleccionarItems() |
|--------------------------------------|--|

| | |
|--|---|
| | d) GuardarItems() *** |
| Responsabilidades | Visualizar en forma de árbol los diferentes elementos del sistema, permitir desplazar hacia arriba o hacia abajo las ramas de cada Servidor y mostrar sus ítems correspondientes. Son operaciones que permiten la visualización en forma de Árbol (Root, Branches, Leaves según especificación OPC acceso a datos.) |
| Tipo | Sistema interno |
| Referencias cruzadas | R1.1, R1.3, R1.6. |
| Excepciones | |
| Precondiciones | Selección de dos servidores OPC |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se mostraron las ramas de los servidores (operación a)). • Si se obtuvo la raíz de cada servidor se puede mover hacia arriba o hacia abajo las respectivas ramas. (operación b)). • Si se mostraron las ramas del árbol, dentro de cada rama se pudieron observar las hojas, que es donde queda ubicado cada ítem o ítems de los servidores el cual interesa manipular. • Según el proceso a supervisar el administrador estableció un enlace con un método que le permite seleccionar los ítems que visualiza en el árbol y llevarlos a una tabla de ítems seleccionados. c) • Si se seleccionaron los ítems para supervisar el proceso según su categoría se estableció un enlace con una base de datos que permitió guardar los ítems seleccionados según categoría o tipo. d) | |

| | |
|--|--|
| Caso de uso No 3. Operación | ConectarServidores() |
| Responsabilidades | Conectar a cada servidor OPC de cada PLC |
| Tipo | Sistema |
| Referencias cruzadas | R1.2, R1.3, |
| Excepciones | Debe haber una conexión a la base de datos que guardó el nombre de los servidores seleccionados. |
| Precondiciones | |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se realizó un enlace con una base de datos que guardó el nombre de los servidores. • Si se conectó a cada servidor se realiza un enlace a cada Servidor, Grupo y nodo respectivamente por servidor, Siemens y Allen Bradley (según validación y objetivos propuestos) • Si se estableció conexión se crea instancia para realizar la lectura asincrónica o sincrónica. | |

| | |
|--|--|
| Caso de uso No 3. Operaciones. | a) VisualizarItems() b) ActualizarValor() |
| Responsabilidades | Mostrar las propiedades de los ítems previamente guardados en la base de datos y adicionados en una tabla por categorías . Realizar conexión para visualizar el valor de los ítems (posiblemente eliminar estos ítems uno a uno). |
| Tipo | Sistema |
| Referencias cruzadas | R1.1, R1.5 |
| Excepciones | La conexión a la base de datos debe ser establecida correctamente |
| Precondiciones | El objeto Browser ha obtenido eficazmente las direcciones de los servidores y ha habido una selección que ha sido guardada. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se realizó un enlace con una base de datos que guardó el nombre de los servidores. a) • Si se muestran los ítems ha habido una comunicación con la base de datos específica para los ítems guardados anteriormente en el caso de uso No 2; • Si se pueden ver las propiedades de los ítems ha habido una creación de una instancia conexión para cada servidor respectivamente (valor, calidad, tiempo indicados como propiedades en la especificación OPC acceso a datos. a) y b) • Si se puede visualizar el valor de cada ítem, ha habido un enlace con una lectura sincrónica o asincrónica para ver actualizadamente los datos de cada servidor. b) • Si se realizó una lectura de los datos se estableció un método que termina esta lectura cuando el administrador así lo considere | |
| Caso de uso No 3 extendido. Operación | EscribirValor() |
| Responsabilidades | Permite al usuario escribir y mandar una valor a una dirección en el servidor (o PLC) |
| Tipo | Sistema |
| Referencias cruzadas | R1.1, R1.5 |
| Excepciones | Si el valor no es escrito correctamente no se puede enviar el valor. |
| Precondiciones | Debe haber una conexión a los servidores OPC y haber sido seleccionados los ítems para visualizar sus propiedades. |
| POSCONDICIONES | |

| |
|---|
| <ul style="list-style-type: none"> • Si se logró la escritura y el envío del valor, se enlazó al Servidor OPC al cual se escogió según la dirección específica |
| <ul style="list-style-type: none"> • Si el valor cambió al ser escrito se crea instancia con Ítems OPC, con servidores y con el grupo al que pertenece la dirección que fue escrita. |

| | |
|-----------------------------------|--|
| Caso de uso No 4 Operación | ConfigurarAtributos() |
| Responsabilidades: | Configurar atributos de alarmas según algún estado determinado. |
| Tipo | Sistema |
| Referencias cruzadas | R2.2, R1.6 |
| Excepciones | |
| Precondiciones | Haber sido seleccionados los ítems apropiados que describen estados de alarma (según la programación del lenguaje de los PLC, se ha determinado las alarmas del proceso) |

POSCONDICIONES

| |
|---|
| <ul style="list-style-type: none"> • Se estableció una asociación entre una propiedad (Alto-Alto, Medio-alto, Leve, leve-Leve; o a través de un color) y un ítem determinado del proceso seleccionado previamente como alarma. |
| <ul style="list-style-type: none"> • Se estableció un enlace con la base de datos que guardó previamente los ítems seleccionados como alarmas |
| <ul style="list-style-type: none"> • Si la alarma es activada o se alcanzó el rango del valor del ítem para el cual se consideró como alarma, entonces se creó una instancia alarma activa para que dé aviso de este evento. |

| | |
|-----------------------------------|---|
| Caso de uso No 4 Operación | GuardarAlarmas() |
| Responsabilidades: | Guardar las alarmas activas. |
| Tipo | Sistema |
| Referencias cruzadas | R2.2, R1.6 |
| Excepciones | |
| Precondiciones | Haber configurado los atributos y las condiciones de alarma |

POSCONDICIONES

| |
|---|
| <ul style="list-style-type: none"> • Si se guardó la alarma configurada entonces se estableció un enlace con base de datos que almacena las condiciones de alarma y la configuración establecida. |
|---|

| | |
|--|---|
| Caso de uso No 4 Operación | a) ConfigurarEventos() b) GuardarEvento() |
| Responsabilidades: | Determinar que debe hacer el sistema cuando ocurre un evento del proceso (Controlado por PLC) y guardar esta información. |
| Tipo | Sistema |
| Referencias cruzadas | R2.2, R1.6 |
| Excepciones | |
| Precondiciones | Haber seleccionado el respectivo ítem del proceso encargado de un evento en particular. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Si ocurrió un evento (o el ítem seleccionado cumplió la condición en la cual es tomado como un evento del proceso) entonces se realizó una conexión con una base de datos que almacena el evento. b) • Si ocurrió el evento también se creó una instancia que controla los eventos (manejador de eventos) para evitar que haya conflictos entre operaciones simultaneas. a) • Si el ítem seleccionado en la categoría de evento cumple una condición determinada (o eventual) se crea una instancia que debe avisar al sistema y almacenarlo. | |

| | |
|---|--|
| Caso de uso No 5 Operación | GraficarVariables() |
| Responsabilidades: | Generar una gráfica de las variables seleccionadas en la categoría de reportes |
| Tipo | Sistema |
| Referencias cruzadas | R2.3, R2.4 |
| Excepciones | |
| Precondiciones | Haber seleccionado el ítem o ítems que cumplirán la función de graficar. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Si se graficó la variable en el tiempo entonces se estableció un enlace con una Conexión y con una lectura asincrónica o sincrónica que permitirá obtener los cambios de la variable o ítem seleccionado para ser graficados. • Si se graficó la variable se creó una instancia que permite la configuración | |

| |
|--|
| de un método de graficación (función para graficar) |
|--|

| | |
|--|---|
| Caso de uso No 5 Operación | GuardarGráficas () |
| Responsabilidades: | Guardar la gráfica de las variables configuradas en la categoría de reportes para ser accedidas por el usuario. |
| Tipo | Sistema |
| Referencias cruzadas | R2.3, R2.4, R1.6 |
| Excepciones | |
| Precondiciones | Haber seleccionado el ítem o ítems que cumplirán la función de graficar. |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se estableció una conexión con una base de datos que guardó la configuración de la gráfica. | |
| <ul style="list-style-type: none"> • Se observó la tendencia de la gráfica por lo tanto se creó instancia que fue accedida por el operario para ser visualizada en el caso de uso Visualizar (caso de uso No 6) | |

| | |
|--|---|
| Caso de uso No 5 Operación | GenerarReportes () |
| Responsabilidades : | Generar los reportes del proceso. |
| Tipo | Sistema |
| Referencias cruzadas | R2.3, R2.4, R1.6 |
| Excepciones | Fallo en la conexión a base de datos. |
| Precondiciones | Haber seleccionado el ítem o ítems que cumplirán la función de reportes |
| POSCONDICIONES | |
| <ul style="list-style-type: none"> • Se creó una asociación con una base de datos que guardó previamente la selección del ítem o ítems para reportes. | |
| <ul style="list-style-type: none"> • Si se visualizó el reporte se creó una asociación con un método que permite imprimir los reportes. | |

Tabla 10. Contratos de las operaciones de los casos de uso.

*** Las operaciones anteriores trabajan unidas para lograr el efecto de visualizar los ítems en una interfaz en forma de árbol dinámico y además poder seleccionar los ítems visualizados de cada servidor para posterior manipulación.

4.2.8 Diagramas de Colaboración Específicos.

Los diagramas de colaboración salen del análisis de los contratos de las operaciones, del análisis de los escenarios. Particularmente se tiene en cuenta las poscondiciones del sistema, la sintaxis de las oraciones que describen el comportamiento del sistema (algunas frases importantes están resaltadas en negrilla para observar esta metodología en la Tabla 10). Se observa en la poscondiciones que los verbos se colocaron en tiempo pasado para indicar que si ocurrió la operación, ocurrió un evento o se formó una relación o se creó un objeto [10].

El objetivo de estos diagramas es mostrar el intercambio dinámico de mensajes entre los objetos del sistema. Se muestra la dinámica interna de la aplicación es decir como se llevó a la implementación según los requerimientos de los casos de uso pero basados en la especificación OPC que definen un conjunto de interfaces con métodos y propiedades pero no su implementación; pues esta surge primero de la aplicación específica a la cual se quiere llegar y segundo del lenguaje de programación que se va a utilizar (los manejos de datos por referencia y datos por valor pueden cambiar según el lenguaje, lo cual se debe tener en cuenta para establecer la lógica de programación, para dar un ejemplo) Se trató en lo posible de no enviar mensajes a las interfaces de presentación, solo si es absolutamente necesario, buscando una arquitectura por capas [10] .

En los siguientes diagramas de colaboración se muestra lo extraído en los contratos de cada operación de los casos de uso del sistema, por tanto cada contrato es una explicación de los diagramas, sin embargo cuando sea necesario, se aclararán ciertos diagramas de colaboración. Se debe tener en cuenta que algunos diagramas presentan comentarios que extienden la explicación o que determinan un comportamiento requerido del objeto.

Diagrama de Colaboración caso de uso No 1: algunos diagrama son divididos en dos o mas para lograr una mejor vista y explicación del comportamiento de los objetos ante los mensajes que reciben.

a) Operación BuscarServidores()

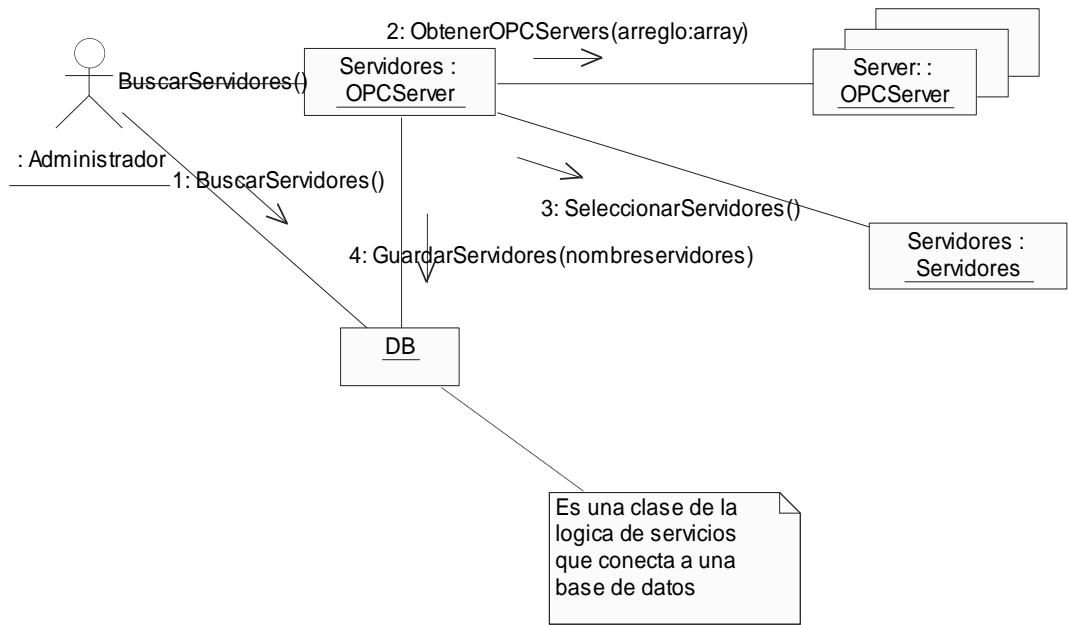


Figura 30. Diagrama de colaboración BuscarServidores()

b). Operación SeleccionarServidores().

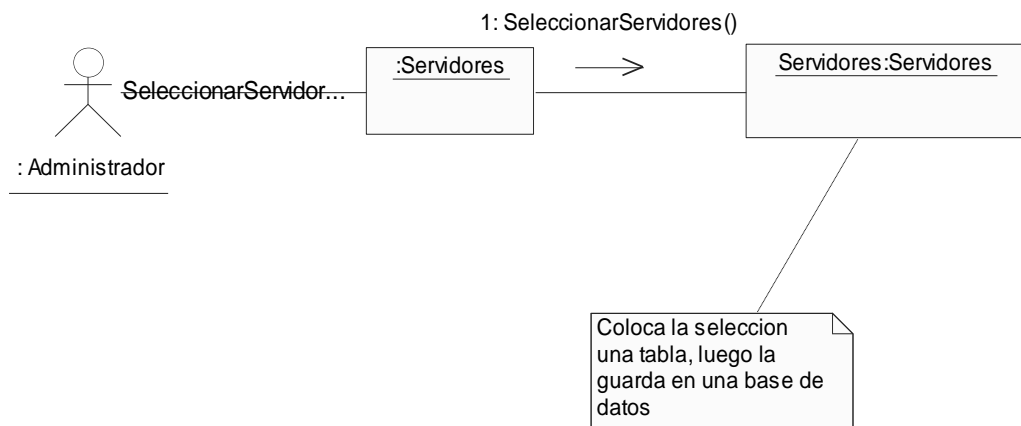


Figura 31. Diagrama de colaboración SeleccionarServidores().

Diagrama de Colaboración Caso de Uso No 2.

a) Operación Browser().

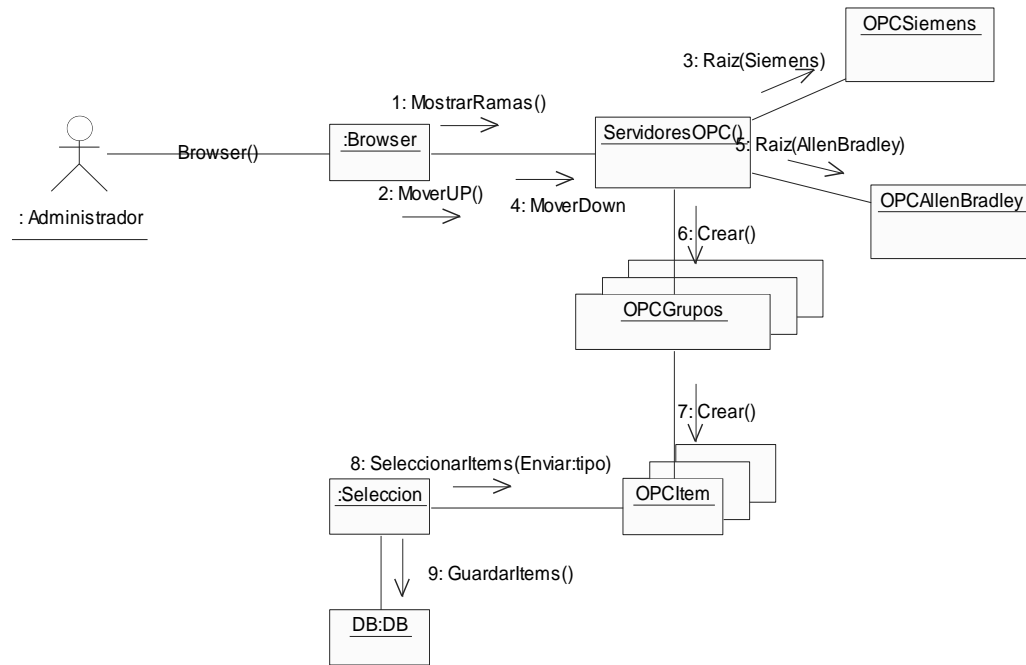


Figura 32. Diagrama de colaboración Browser().

Se debe aclarar que el Browser toma los servidores (2) que han sido seleccionados y guardados en una base de datos previamente; el objeto DB:DB es el encargado de realizar esta operación junto con el objeto Servidores. (Observar contrato Browser Tabla 10).

b) Operación MostrarRamas().

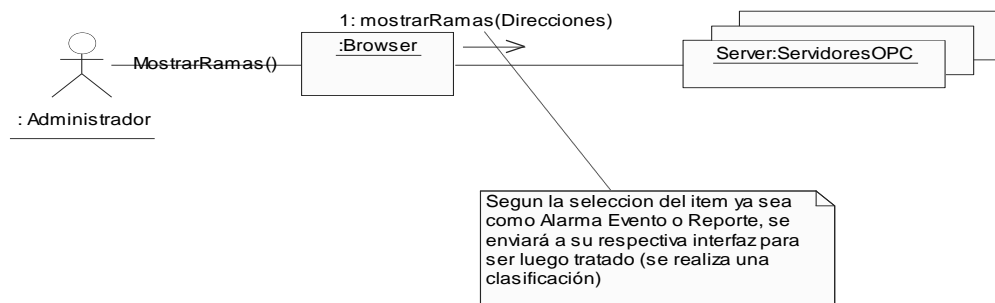


Figura 33. Diagrama de colaboración Mostrarramas()

Diagrama de colaboración caso de uso No 3

a) Operación VisualizarItems().

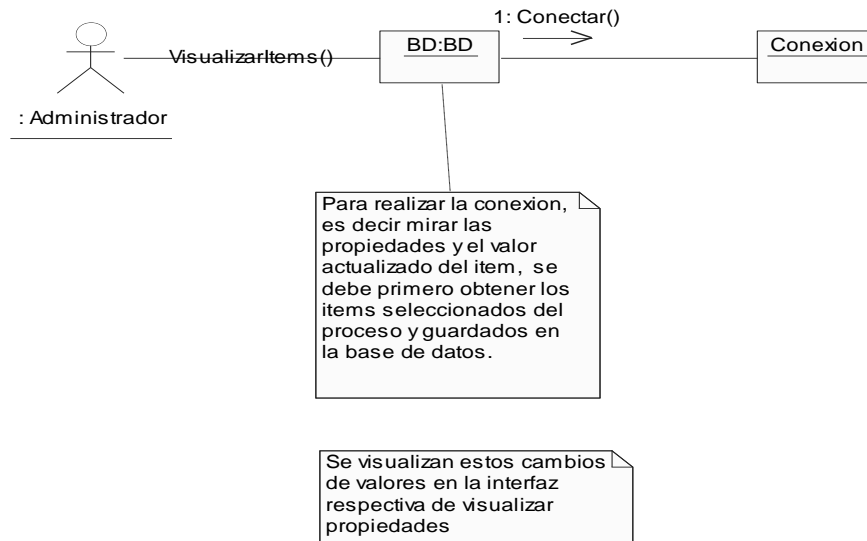


Figura 34. Diagrama de colaboración VisualizarItems().

b) Operación ActualizarValor().

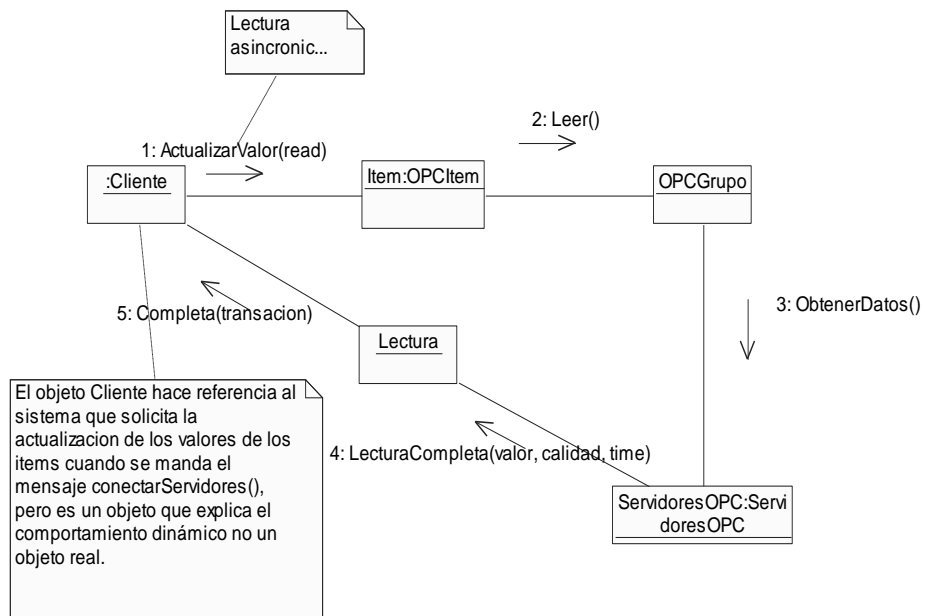


Figura 35. Diagrama de colaboración ActualizarValor().

c) Operación EscribirValor()

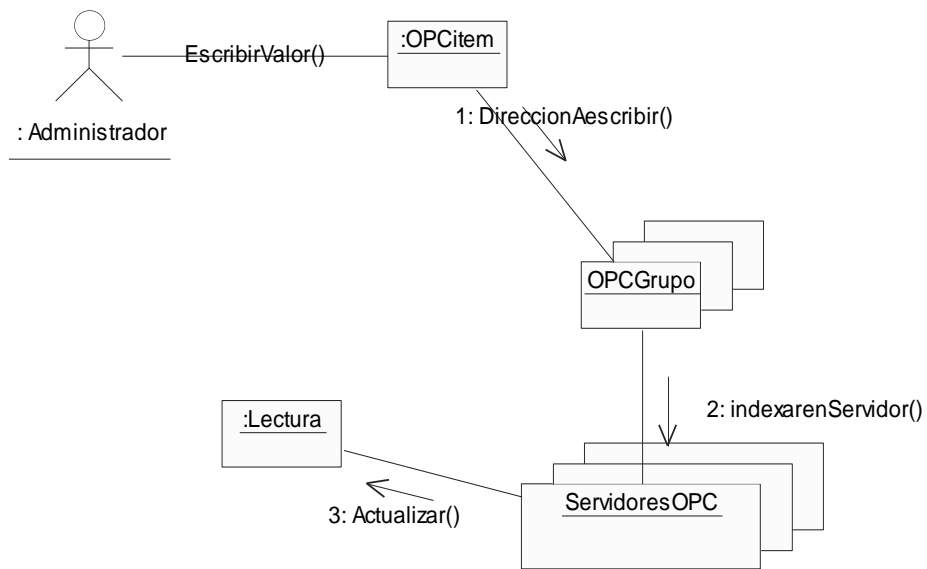


Figura 36. Diagrama de colaboración EscribirValor().

d) Operación. Inclusión ConectarServidores()

Según la Figura 15, el caso de uso Conectar servidores está incluido en Visualizar Propiedades. También se debe observar que algunas veces casos de uso y operaciones pueden tener los mismos nombres, se debe recordar que el caso de uso alude a un requerimiento implícito o tácito del sistema, mientras que una operación alude a un proceso o a un evento que produce reacciones significativas en el sistema o un conjunto de sucesos que tienen un objetivo en común para permitir que ciertas responsabilidades determinadas en los contratos, se lleven a cabo.

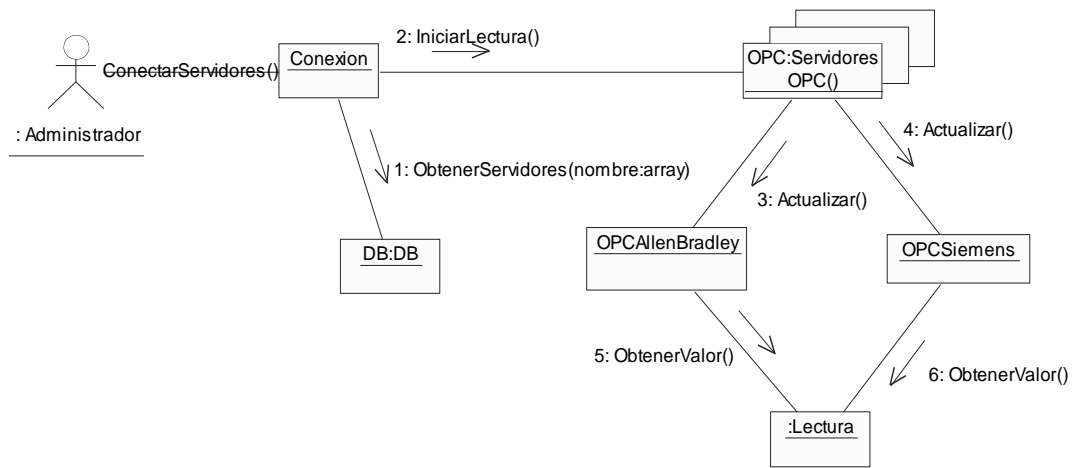


Figura 37. Diagrama de colaboración ConectarServidores().

Diagrama de Colaboración Caso de uso No 4. Configurar Alarmas y eventos

a) Operación ConfigurarAtributos().

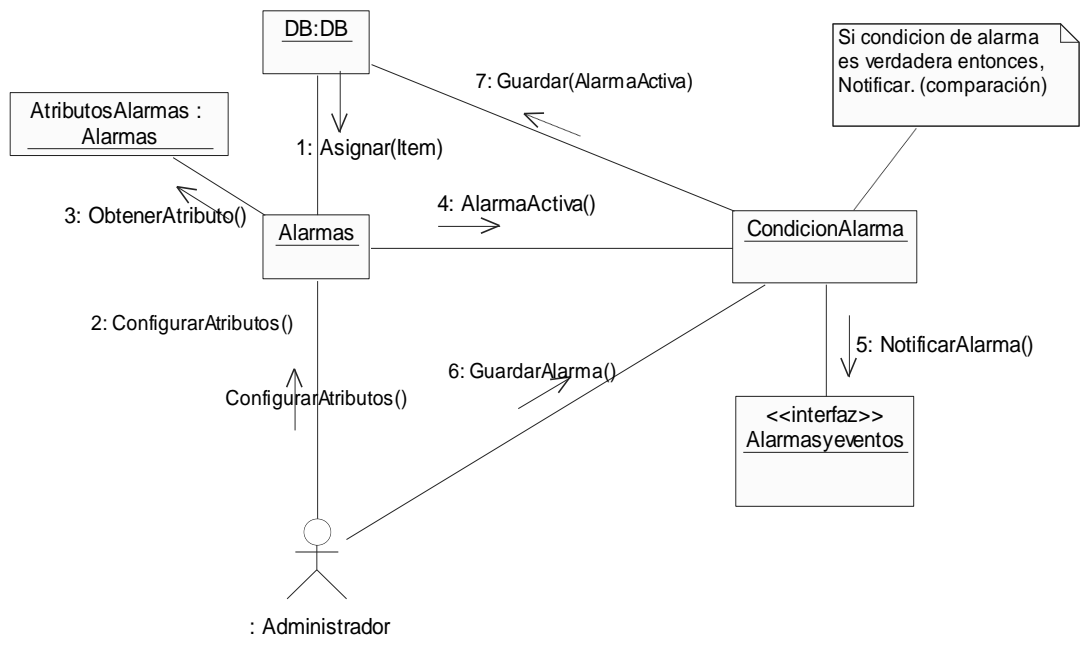


Figura 38. Diagrama de colaboración ConfigurarAtributos().

b) Operación GuardarAlarma().

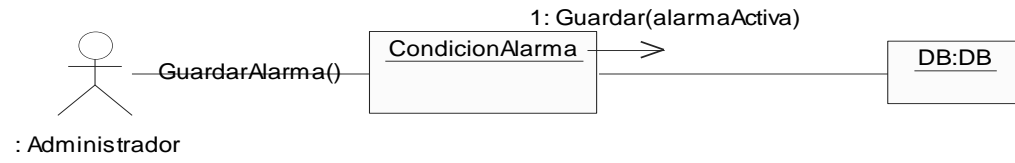


Figura 39. Diagrama de colaboración GuardarAlarmas()

c) Operación configurarEvento()

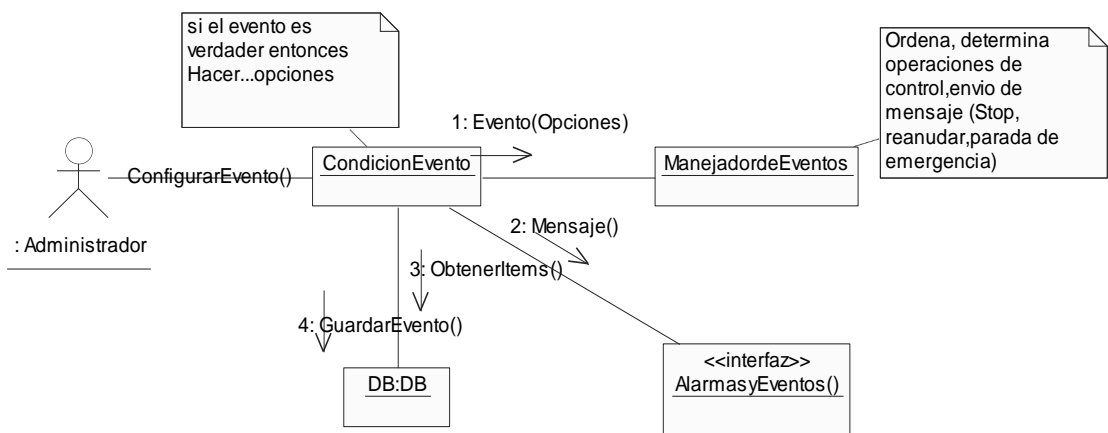


Figura 40. Diagrama de Colaboración ConfigurarEvento()

Diagrama de Colaboración Caso de Uso No 5 Configurar Tendencias.

a) Operación GraficarTendencias().

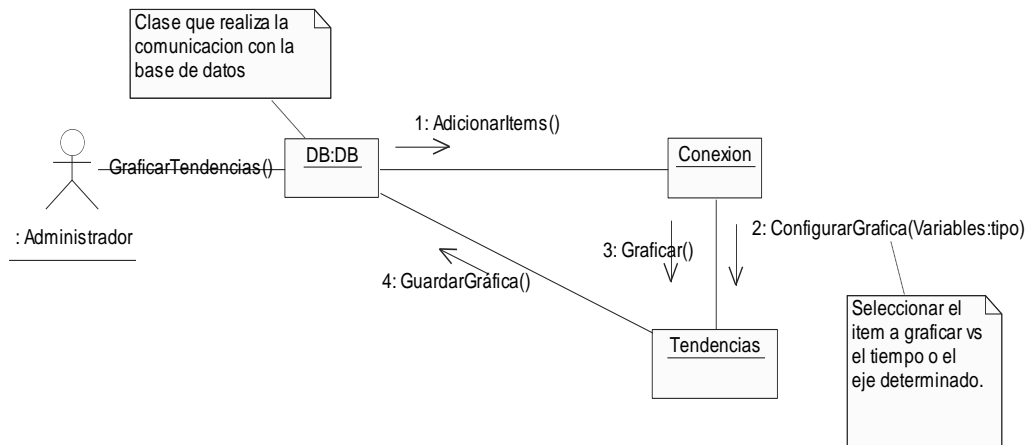


Figura 41. Diagrama de colaboración GraficarTendencias().

b) Operación GuardarGraficas().

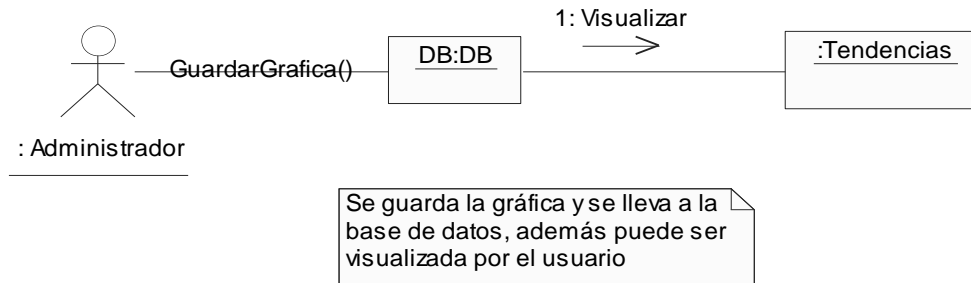


Figura 42. Diagrama de colaboración GuardarGrafica()

c) Operación GenerarReportes().

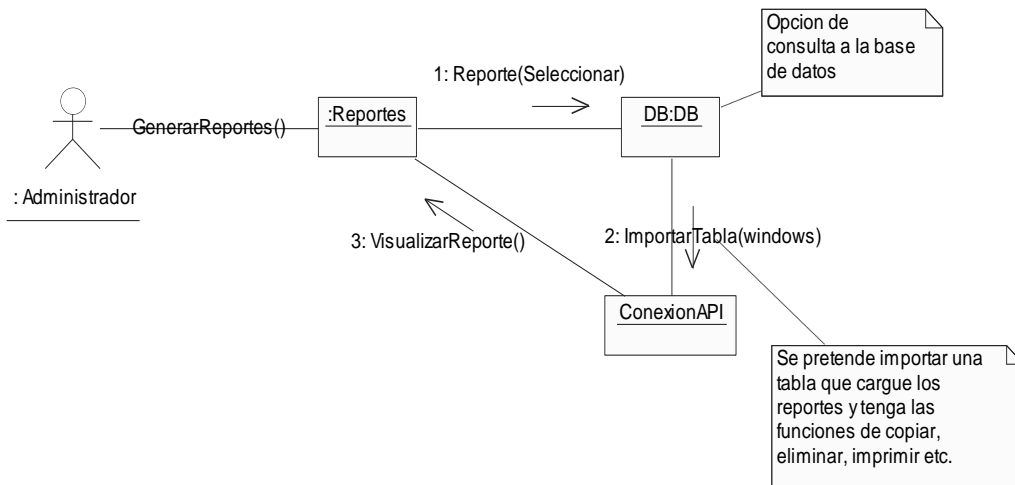


Figura 43. Diagrama de colaboración GenerarReportes()

4.2.9 Descripción de Paquetes y Clases de análisis Esenciales del sistema

Los Diagramas presentados en este aparte, surgen de varios análisis, entre ellos los diagramas de colaboración principalmente, que a pesar de mostrar un conjunto de objetos, se construyen las diferentes clases tomando los mensajes como operaciones de cada clase (mas adelante se formarán métodos). Se agrupan los objetos que tengan las mismas características, o características comunes (apoyados también de la especificación dada por OPC) para formar clases con operaciones y atributos no repetidos (ver Figura 30 a Figura 43). Se debe tener en cuenta que no necesariamente todos los objetos descritos y sus mensajes pasarán directamente a formar parte de las clases, puesto que los diagramas de componentes son modelos interactivos (no estáticos), mientras que las clases son diagramas estáticos que poseen una visibilidad, es decir sus atributos son vistos por otra clase para acceder a ellos [10][11]. También en este análisis esencial de clases se muestran las interfaces necesarias para que el sistema interactúe con el usuario y se muestran las interfaces de la norma OPC acceso a datos interfaz de

automatización 2.02, con la cual interactúa la aplicación realizada; es decir que sobre las interfaces dadas por la norma (dadas pero no implementadas) se construye toda la lógica y dinámica del sistema. Haciendo una analogía para entender este punto la similitud es parecida a tener un manual de usuario, que le enseña a este como comunicarse con un sistema, así mismo la especificación OPC acceso a datos interfaz de automatización (ver Cap 3) especifica como debe comunicarse el cliente con el servidor.

Descripción de las clases de análisis: las clases están empaquetadas en tres partes siguiendo una arquitectura software por capas, la cual establece un orden que otorga capacidad de reutilizar el código y hace la aplicación independiente de la interfaz de usuario.

La primera capa es la capa de **Lógica de presentación**, la segunda es la **Lógica de Negocios** y la tercera es la **Lógica de Servicios**.

Se observa en la Figura 44. Como las clases de presentación se comunican con la lógica de negocios, pero no directamente con la lógica de Servicios

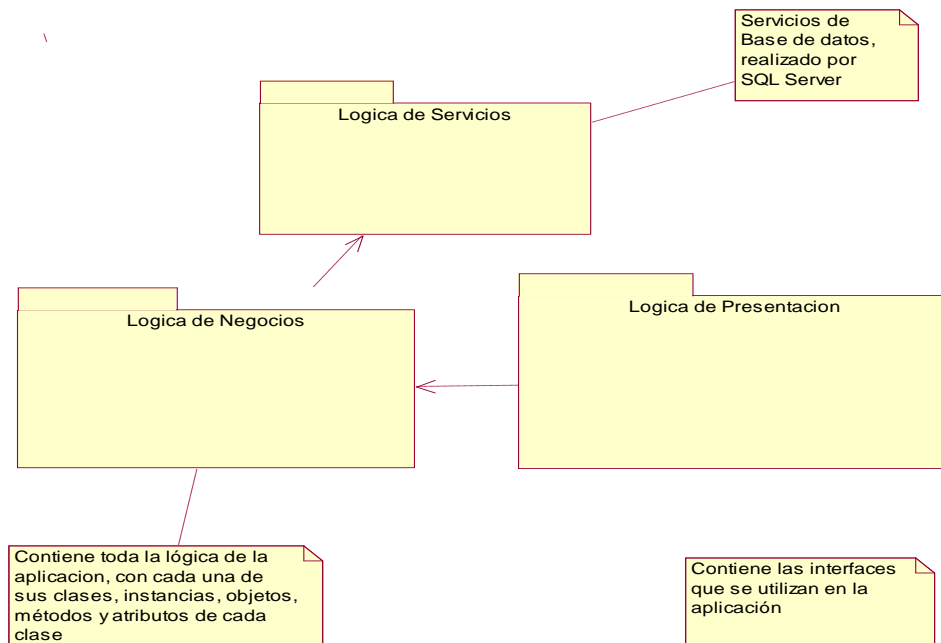


Figura 44. Paquetes del sistema

a) Clases de análisis tipo interfaz: estas clases están empaquetadas en **Lógica de presentación** su objetivo es permitir al usuario interactuar con el sistema, como se observa, muchos de los métodos que tienen son iguales a los métodos de los diagramas de colaboración y secuencia, debido a que la arquitectura software divide al

sistema en las tres capas ya mencionadas, y esto requiere que no se envíen mensajes directamente de algún objeto a una interfaz, si no que el objeto indicado en la Lógica de negocios realice la operación que se solicita (los mensajes se convirtieron en operaciones, mas concretamente en métodos). Siguiendo los requerimientos del diseño UML no se muestran todos los métodos y propiedades, solo los más importantes que determinan el comportamiento y la visibilidad de cada clase.

- **Descripción de las clases Tipo interfaz**

Clase ControlAcceso: Como se muestra en la figura primero hay una clase encargada de controlar el acceso, esta clase control de acceso valida si es un operario o administrador.

Clase Principal: si el usuario ha sido validado como administrador esta clase contiene todas la interfaces para que se pueda configurar el sistema.

Clase BuscarServidores: permite al administrador Busque los servidores y seleccione dos y los guarde en la base de datos.

Clase Browser: permite al administrador buscar las variables en cada servidor y seleccionar las que necesita ínter operar en el proceso, luego guardarlas en la base de datos.

Clase VisualizaPropiedades: Como siguiente paso de configuración Visualiza las propiedades de las variables que se guardaron haciendo una lectura o actualización de valores de cada ítem (habrá por lo tanto una adición de ítems especificado en la norma OPC para esta operación.).

Clase AlarmasyEventos: permite al administrador tomar de las base de datos los ítems y configurar cuales de los ítems son alarmas o eventos y asignar las respectivas propiedades.

Clase Tendencias: permite al administrador seleccionar un ítem específico a graficar en el tiempo

Clase Visualizar: esta clase permite al usuario (que ha sido validado como tal), observar el proceso (a través de un mímico de proceso) y las alarmas, reportes y eventos configurados previamente por el administrador, por lo tanto las clases **Visualizar_Alarmas, Visualizar_Tendencias y Reportes** son clases de visualización o menús que están dentro de **Visualizar** que al ser accedidas muestran la información correspondiente.

Las anteriores clases se muestran en la Figura 45.

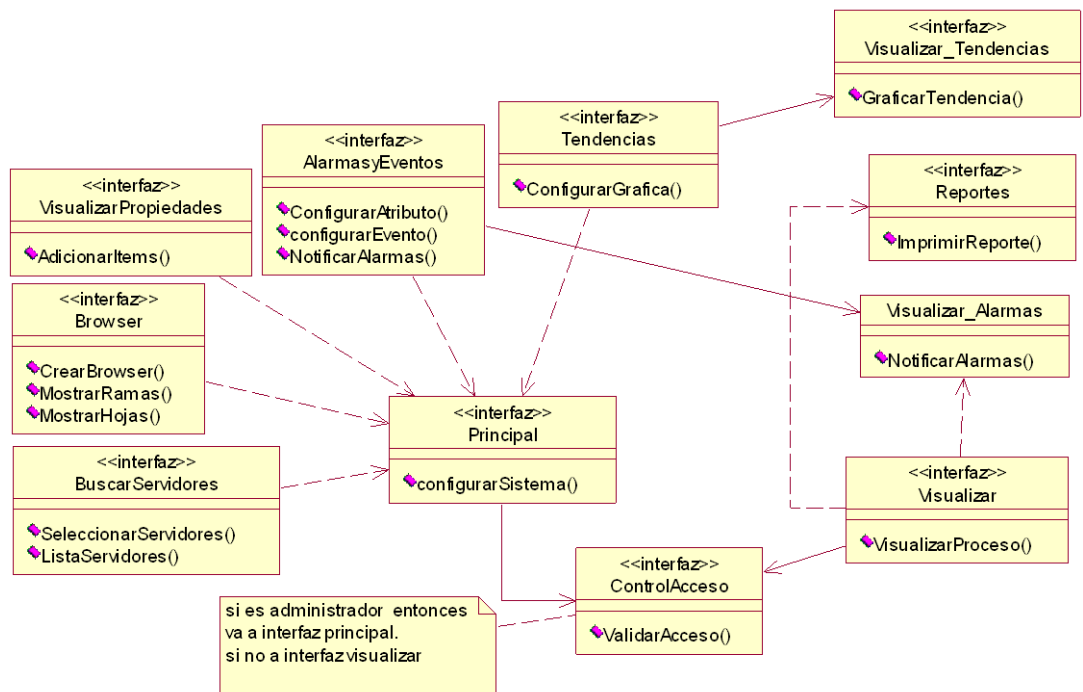


Figura 45. Clases tipo interfaz.

e) **Clases de análisis tipo Control (lógica interna de la aplicación):** estas son las clases reunidas dentro de **Lógica de Negocios**, su objetivo es realizar las funciones internas lógicas de la aplicación y la dinámica del sistema descritas en los diagramas de objetos. Se tiene en cuenta que a partir del análisis de los mensajes y de la agrupación de objetos surgen las clases que se encuentran en este punto [10] [11].

- **Descripción de las clases de la Lógica Negocio**

Clase OPCSiemens: Es una de las clase más importante del sistema, puesto que aquí se configuran los objetos Grupo, Grupos, Ítem e ítems descritos en la especificación OPC acceso a datos interfaz de automatización. Una de sus funciones más importantes es realizar la lectura o comunicación con cada servidor y obtener las propiedades necesarias para acceder a los datos. En este caso accede a los datos del servidor Siemens.

Clase OPCAllenBradley: tiene las mismas características explicadas de la clase OPCSiemens. Se comunica con el servidor AllenBradley. Aunque es importante aclarar puntualmente que en la selección de los servidores se pudieron seleccionar dos de cualquier fabricante, se colocaron estos nombres de servidores a estas clases para mostrar y explicar de forma clara el proceso de construcción de toda la aplicación y la interoperabilidad entre dos marcas distinta y en especial para el proceso de validación del proyecto; sin embargo en el desarrollo del proyecto se observa que puede ser dimensionado a cualquier tipo de PLC que contenga servicios OPC.

Clase Servidores: contiene métodos para guardar los servidores seleccionados y los ítems de cada servidor. Se relaciona particularmente con la clase DB, para manejar la lógica de la base de datos.

Clase Alarmas_y_Eventos: es la clase encargada de configurar y establecer propiedades de los ítems seleccionados como alarmas según el proceso.

Clase Reporte: es la clase encargada de la configuración de reportes del sistema según sea requerido.

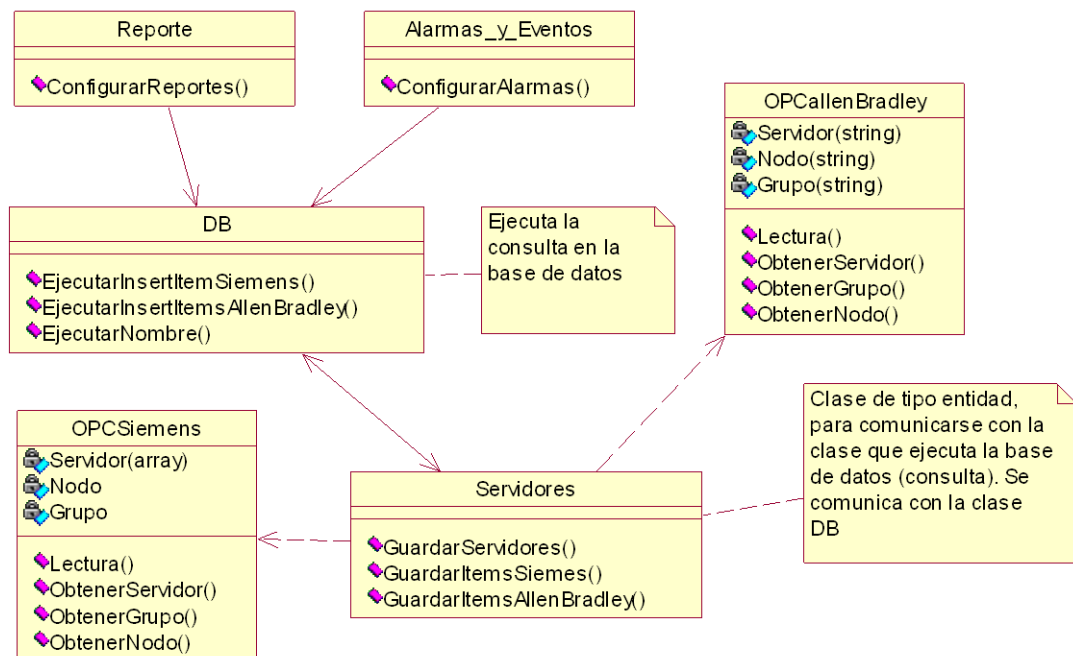


Figura 46. Clases de la Lógica de negocios

f) **Clases de la Lógica de Servicios:** son accedidas a través del espacio de nombres (name Space) y no directamente, es decir siempre se comunican con la clase **Servidores** y no directamente con **Presentación**, son las clases encargadas de realizar la consulta y conexión con la base de datos

- **Descripción de las clases de análisis de la Lógica de servicios:** estas son las clases que manejan la conexión, consulta y manejo de la base de datos requerida para guardar y extraer los ítems, nombre de Servidores y otras configuraciones del sistema.

Clase DB: es la clase encargada de realizar la conexión y consulta con base de datos. Están dentro de la **lógica de Servicios**

4.2.10 Revisión y Evaluación de la fase de Formulación del Proyecto.

Se realiza una revisión a la Figura 15, pues esta presenta una inconsistencia, ya que el usuario es quien debe hacer el uso del caso Control de acceso. Ya sea administrador u operario, deben pasar primero por el control de acceso.

Se determinó al analizar esta fase, que se requiere encerrar los casos de uso, en el caso general **Configurar Sistema**, en el siguiente orden: primero buscar los servidores y seleccionar uno o máximo dos, luego guardar en la base de datos, acto seguido aparecerá el Browser para buscar las variables de cada servidor seleccionado. Al ser elegidas las variables o ítems según un proceso específico, se guardan en la base de datos, y al realizar este procedimiento se presenta la interfaz de visualización de los ítems para adicionar los que se han escogido de la base de datos y observar sus propiedades. Luego se configuran las alarmas y los eventos, y por último las tendencias, reportes y el proceso a visualizar gráficamente (Mímico). Es decir que el procedimiento anterior esta incluido dentro de **Configurar Sistema**.

Implícitamente se deben establecer varias excepciones para lograr más estabilidad en el sistema, y no permitir un uso desordenado del sistema mediante el uso de mensajes que guíen al usuario a realizar una buena configuración.

Se determinó que la interfaz de configuración del sistema, también es una herramienta para visualizar el comportamiento del proceso, debido a que se puede acceder a las variables o ítems de los servidores (Browser), y realizar una conexión para observar el valor de estas (Visualizar Propiedades). Se muestra en la Figura 47 el cambio en este caso de uso por parte del usuario.

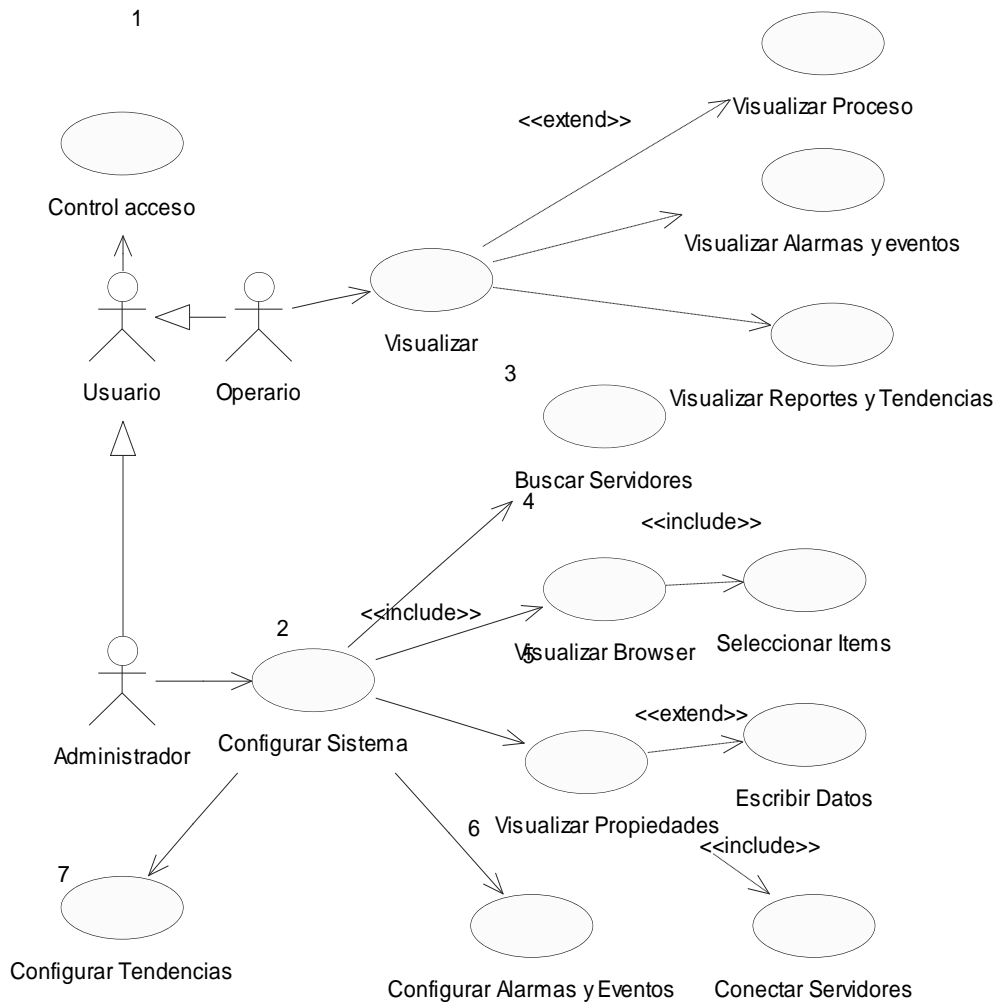


Figura 47. Casos de uso mejorados finales

4.3 EJECUCION DEL PROYECTO

En este aparte debido a los anteriores análisis, se muestran los modelos de diseño esencial de la aplicación requerida.

4.3.1 Diagramas de Secuencia Específicos de Casos de uso de Diseño

Los diagramas de secuencia específicos son generados a partir de los diagrama de colaboración y hacen parte de los diagramas interactivos, pues consideran como su nombre lo indica, la interacción de los diferentes objetos trabajando en la consecución de

una responsabilidad definida previamente en los contratos y en los escenarios de los casos de uso. Algunos diagramas de secuencias se muestran en un solo diagrama según el caso de uso, pero otros debido a su complejidad se muestran según sus operaciones (divididos). Las explicaciones de estos diagramas están contenidas en los escenarios y en los contratos Tabla 7 y Tabla 10, la respuesta del sistema en muchos casos, produce la creación o relación con otro objeto, sin embargo según la notación UML, cada mensaje entre paréntesis lleva la respuesta del sistema y el tipo de datos que espera.

Diagrama de Secuencia caso de uso No 1 Buscar servidores

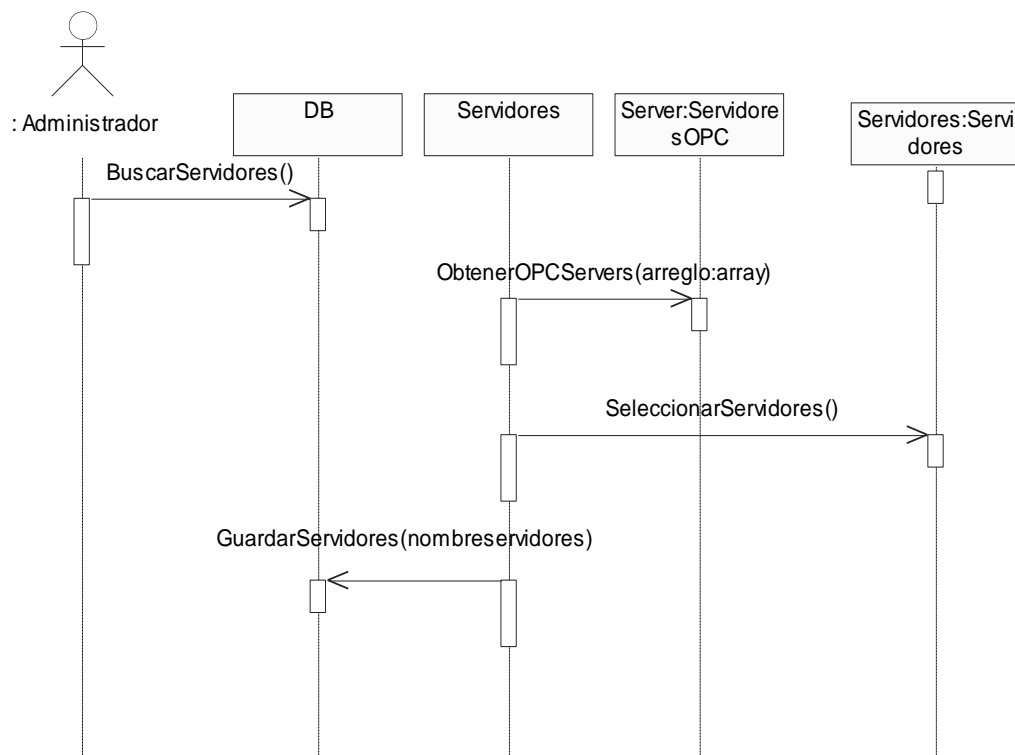


Figura 48. Diagrama de secuencia Caso de uso No 1 Buscar Servidores

Diagrama de Secuencia caso de uso No 2 Visualizar Browser

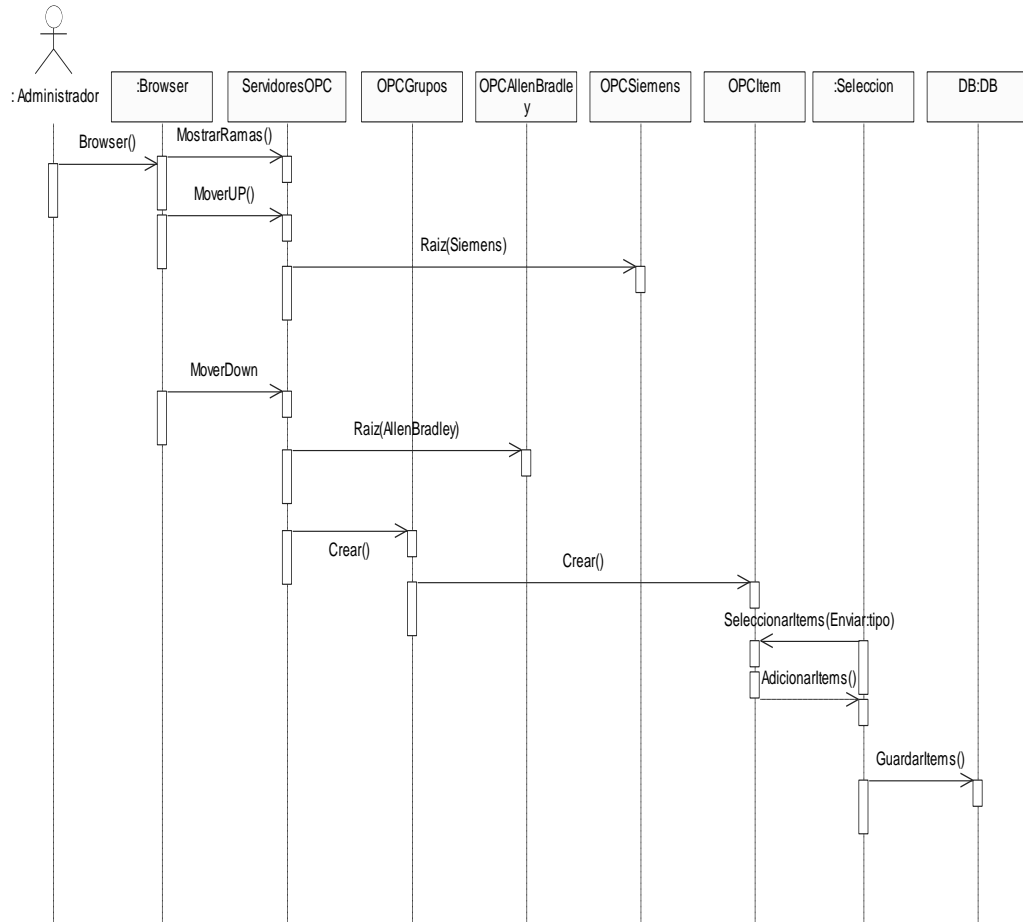


Figura 49. Diagrama de secuencia caso de uso No 2, Visualizar Browser

Diagrama de Secuencia caso de uso No 3 Visualizar Propiedades

a) Operación VisualizarItems()

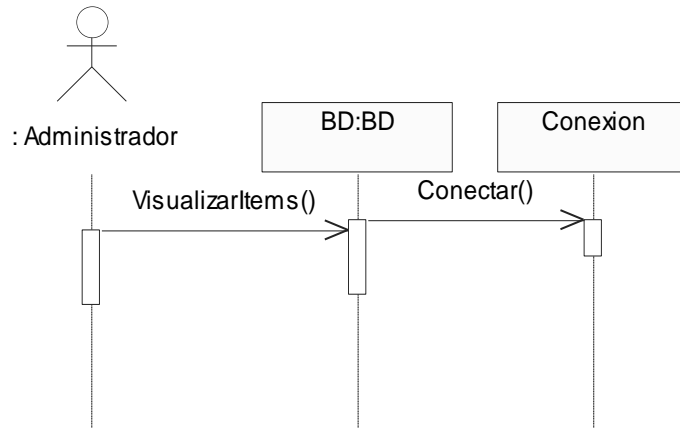


Figura 50. Diagrama de secuencia operación VisualizarItems().

b) Operación ConectarServidores().

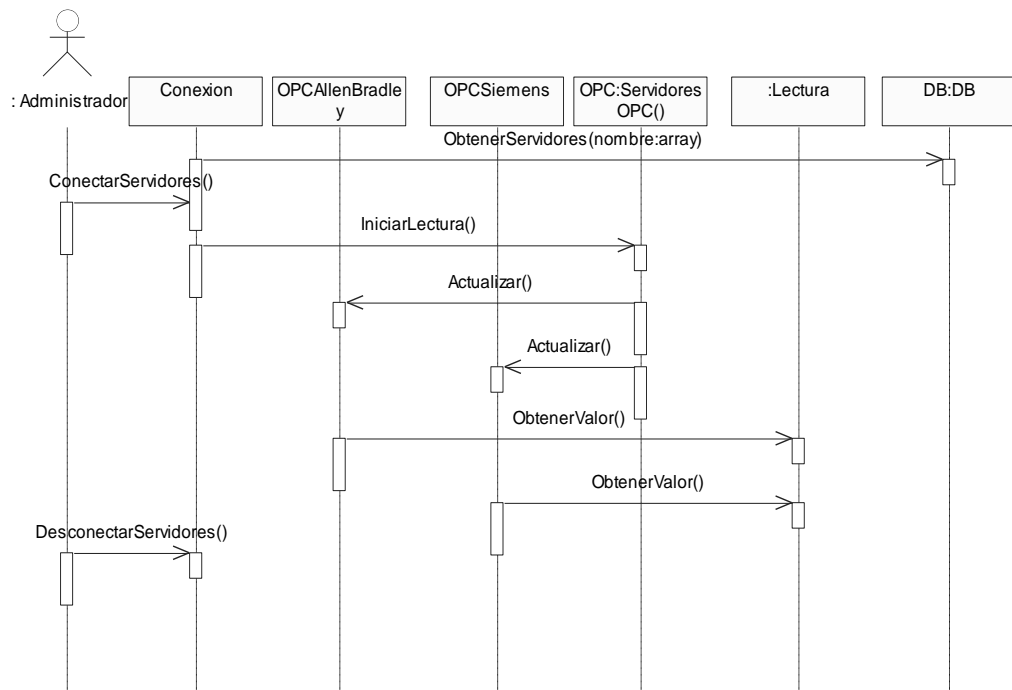


Figura 51. Diagrama de secuencia operación conectarServidores().

c) Operación ActualizarValor()

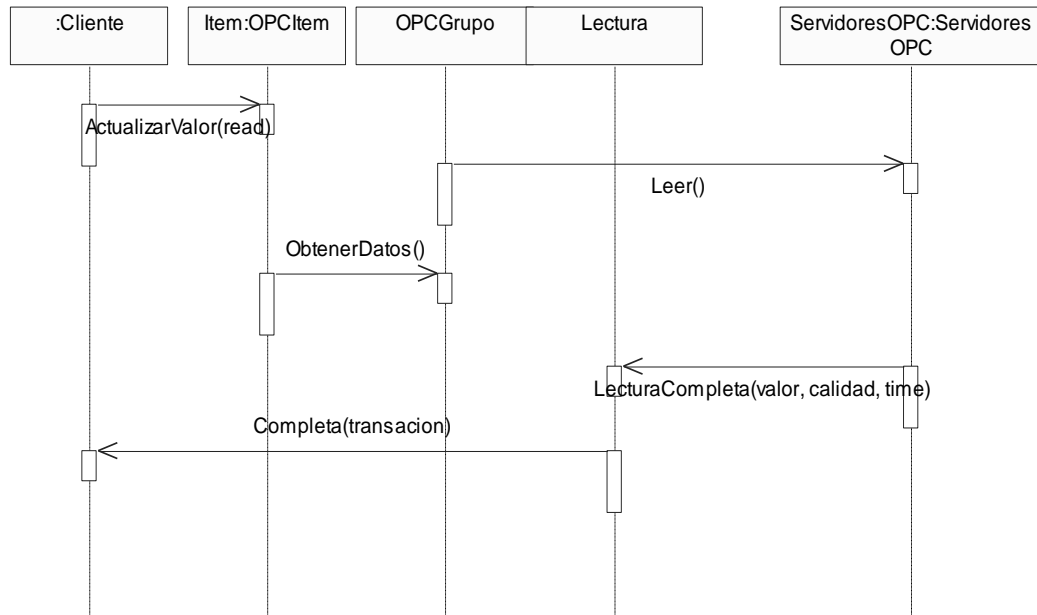


Figura 52. Diagrama de secuencia operación ActualizarValor()

d) Operación EscribirValor().

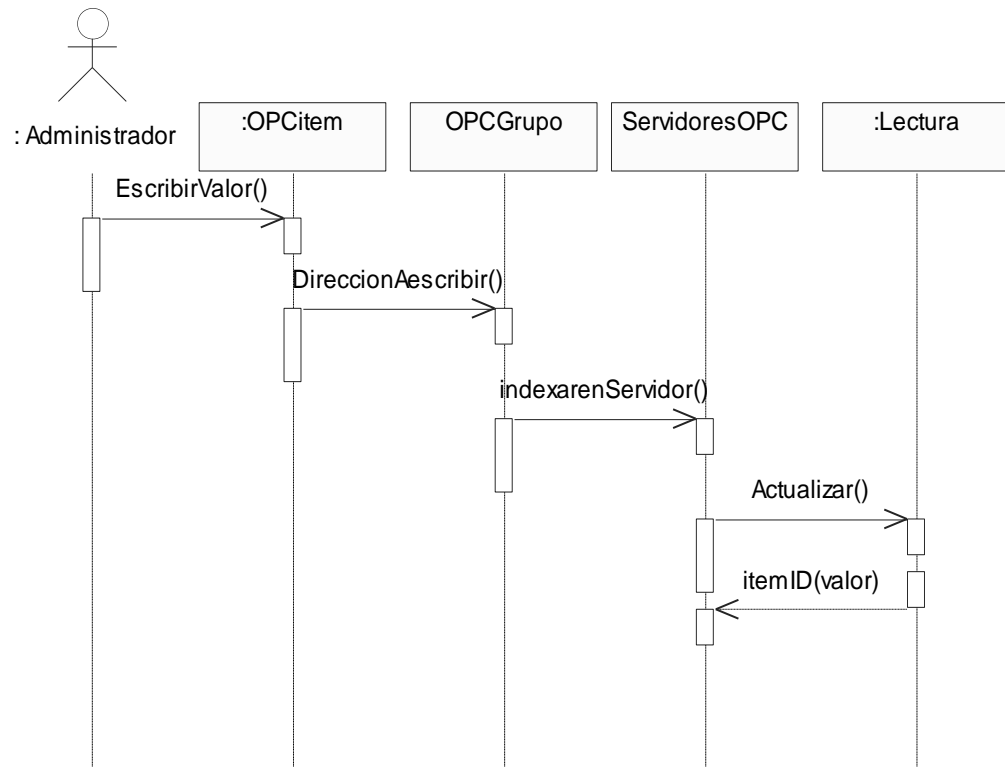


Figura 53. Diagrama de secuencia operación EscribirValor()

Diagrama de Secuencia caso de uso No 4 Configurar Alarmas y Eventos

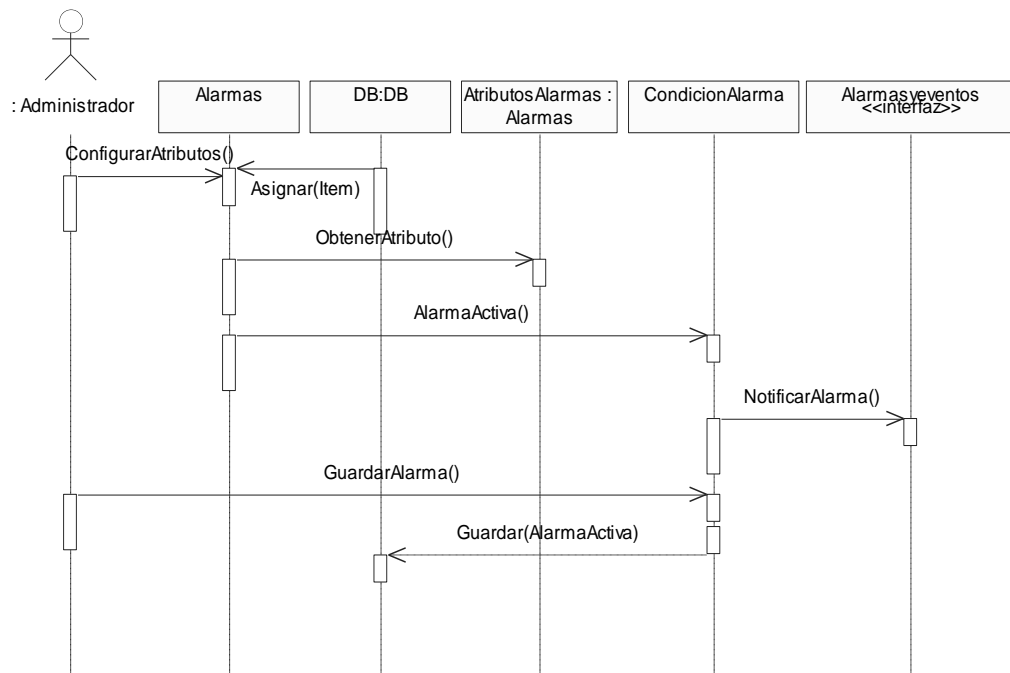


Figura 54 Diagrama de secuencia operación ConfigurarAtributos()

a) Operación ConfigurarEvento().

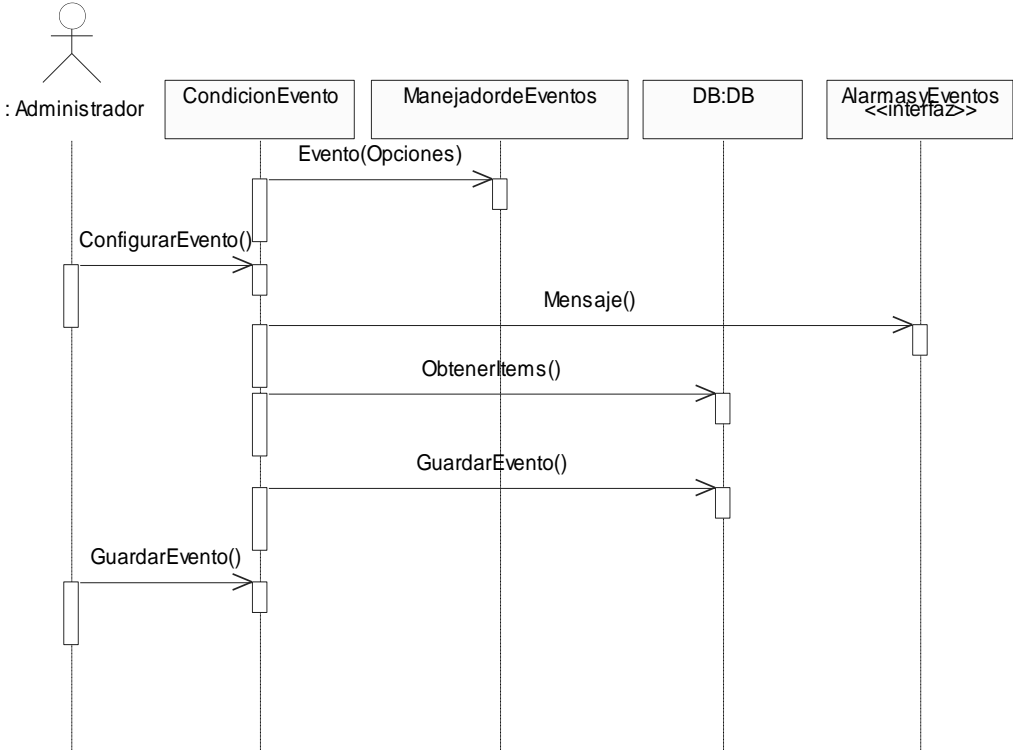


Figura 55. Diagrama de secuencia operación ConfigurarEvento().

Diagrama de Secuencia caso de uso No 5 Configurar Tendencias.

a) Operación GraficarTendencias().

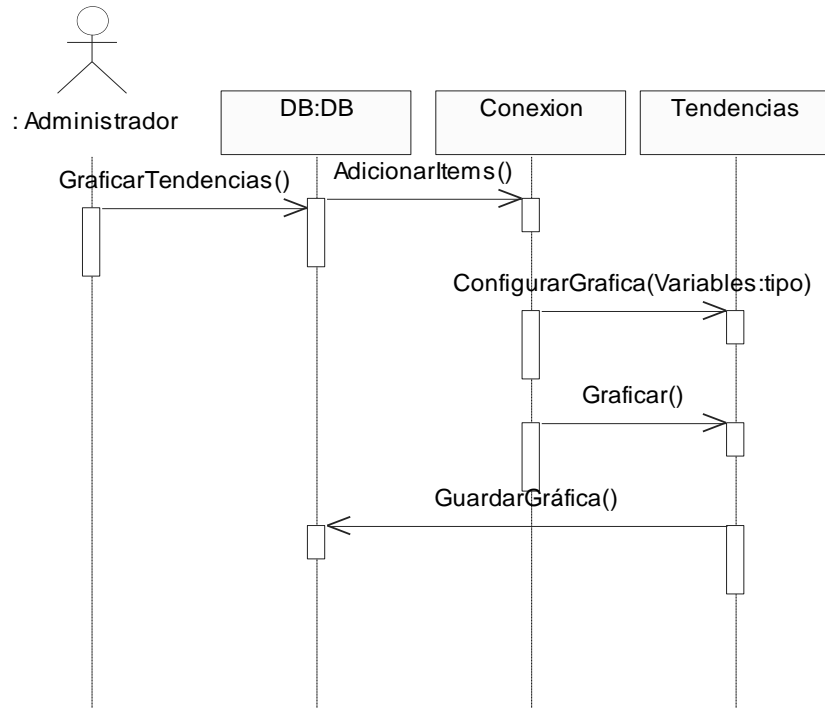


Figura 56. Diagrama de secuencia operación GraficarTendencias().

b) Operación GenerarReportes().

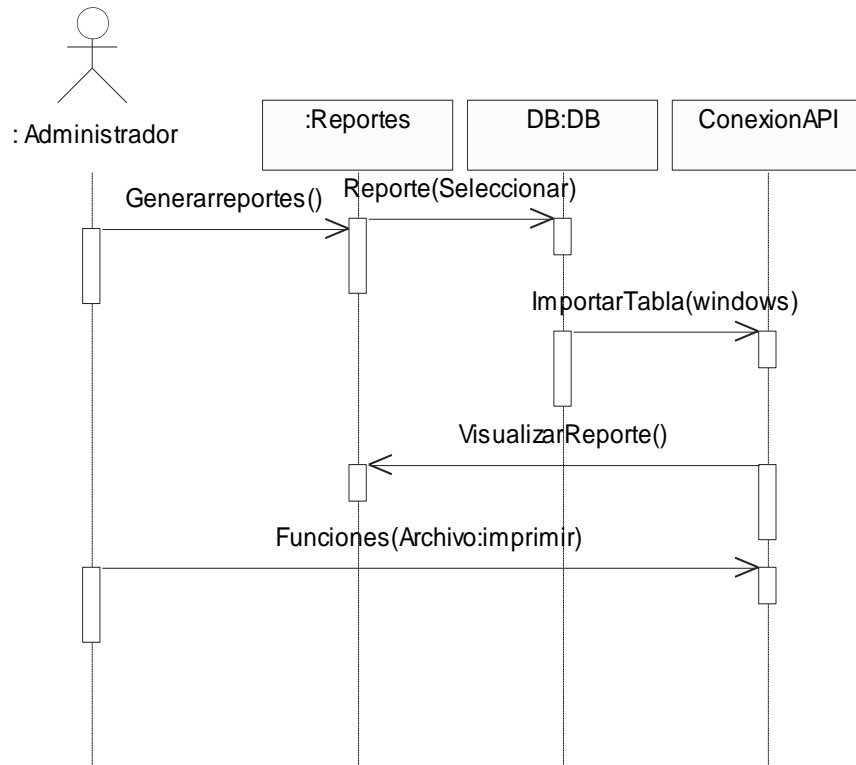


Figura 57. Diagrama de secuencia operación GenerarReportes().

4.3.2 Descripción de paquetes y Clases de Diseño de implementación del Sistema

- **Diagrama de componentes:** la Figura 58 muestra las clases y librerías que utiliza el proyecto para comunicarse y realizar las diferentes funciones de la aplicación.

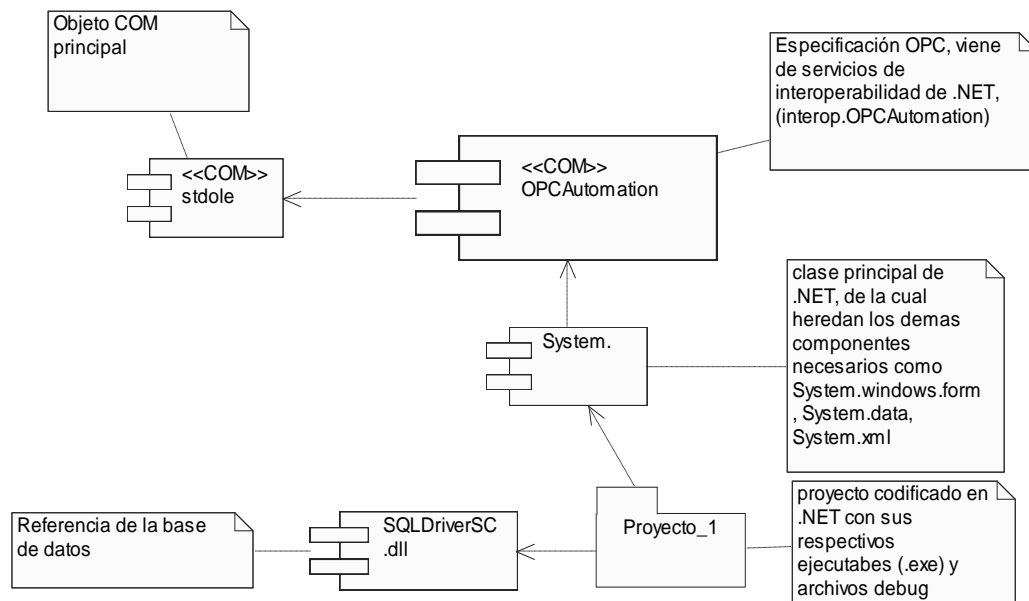


Figura 58. Diagrama de Componentes del sistema.

- a) **Paquete <<COM>> OPCAutomation:** es el paquete más importante del sistema, contiene las librerías del objeto COM., está contenido dentro de servicios de interoperabilidad de Visual Estudio.NET. A través de este paquete es que podemos acceder a los diferentes métodos y atributos que nos permiten realizar el acceso a datos y las diferentes funciones que se logran implementar en la aplicación OPC. La Figura 59 muestra la comunicación y las diferentes clases de este paquete, las cuales fueron explicadas en la Tabla 1, en el capítulo 3.
- b) **Proyecto_1:** contiene la aplicación implementada en Visual Basic.NET, la arquitectura es realizada en **tres capas** como se observa en la Figura 60.
- c) **SQLDriverSC.dll:** es una librería dinámica que permite la conexión con la base de datos y la utilización de los métodos de consulta y demás que se necesitan para el manejo de la base de datos.
- d) **System:** paquete principal o superclase de visual basic.NET del que heredan las demás librerías de aplicación y Windows Forms. Contiene todas las clases, métodos y atributos para el manejo de la lógica, interfaces y funciones en este lenguaje.

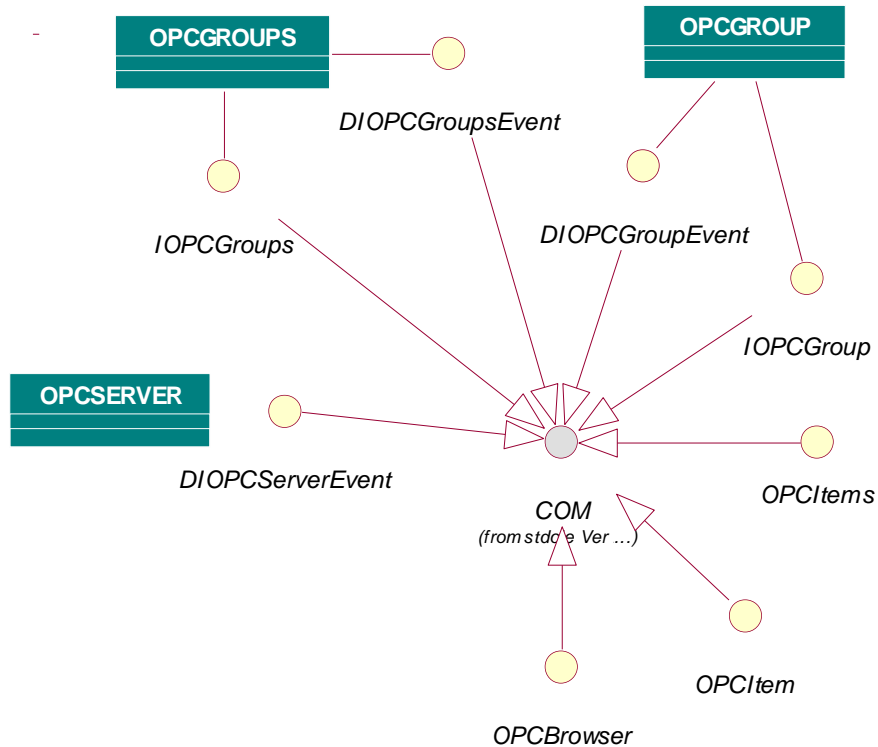


Figura 59. Interfaces y clases de OPC Automation

- **Clases de Diseño de Implementación del Sistema:** la **lógica de presentación** contiene todas las interfaces con sus respectivos métodos y excepciones, que permiten al usuario comunicarse con la lógica del sistema para realizar las diferentes configuraciones y servicios que ofrece la aplicación.

La **lógica de negocios** contiene todas las clases que implementan la lógica de la aplicación descrita en los diferentes diagramas de colaboración y secuencia, en estas se encuentran los diferentes métodos, objetos, instancias de clases, atributos de clases que en su interacción determinan las funcionalidades del sistema y llevan a cabo todos los requerimientos solicitados en los casos de uso. (Ver diagramas de colaboración y secuencia Figura 30 a Figura 43 y Figura 48 a Figura 57, capítulos 4.2.8 Diagramas de Colaboración Específicos. y 4.3).

En la **lógica de servicios** se encuentran las clases que establecen la comunicación y consulta de la base de datos utilizada para guardar y ordenar los datos de acuerdo a su tipo o utilización, como se puede ver en los diferentes diagramas de componentes

siempre se guardan los ítems adquiridos en una base de datos que se va a llamar luego para iniciar la comunicación y realizar los diferentes servicios de la aplicación.

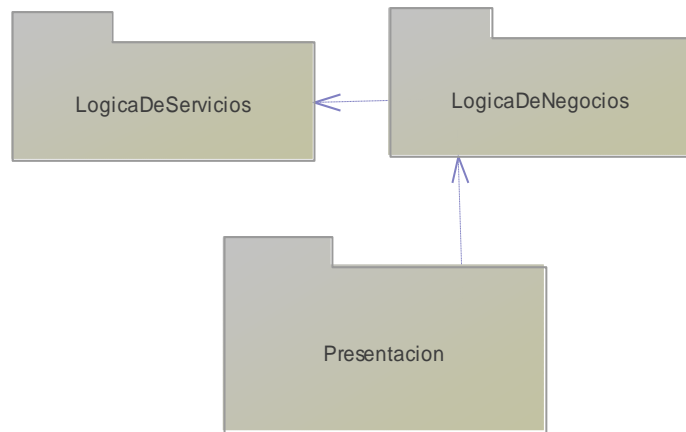


Figura 60. Componentes generales de la arquitectura del sistema

Se muestran las clases de implementación las cuales cambiaron con respecto a las clases de diseño de la Figura 45 y Figura 46, (sección 4.2.9) en que se dejó una sola clase que maneja la configuración de alarmas, eventos y reportes de la aplicación.

a) Lógica de Negocios

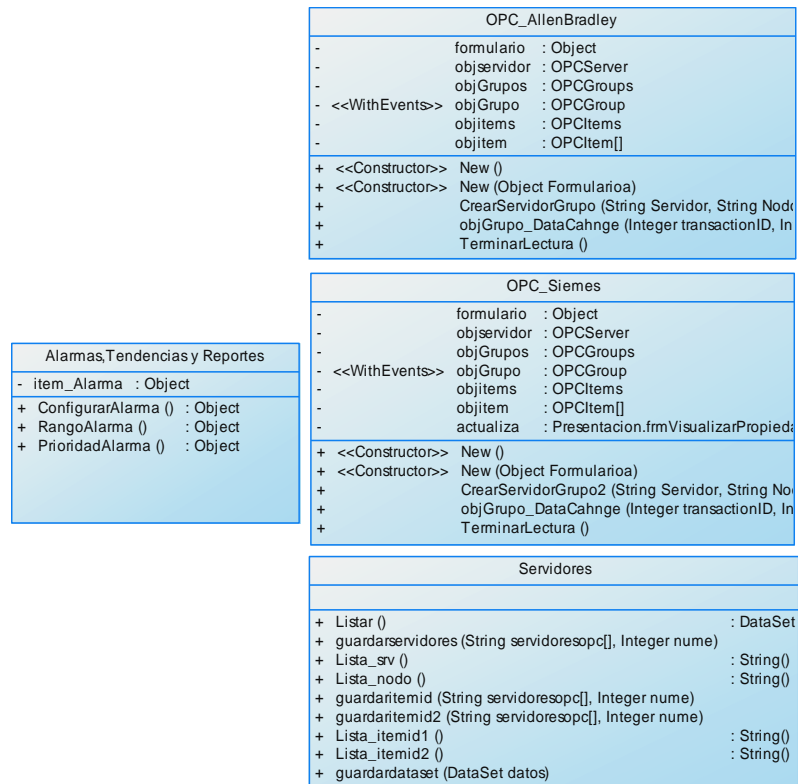


Figura 61. Clases de implementación lógica de negocios

b) Clases de Presentación

El siguiente diagrama se presenta en dos partes para visualizarlo de una mejor manera.

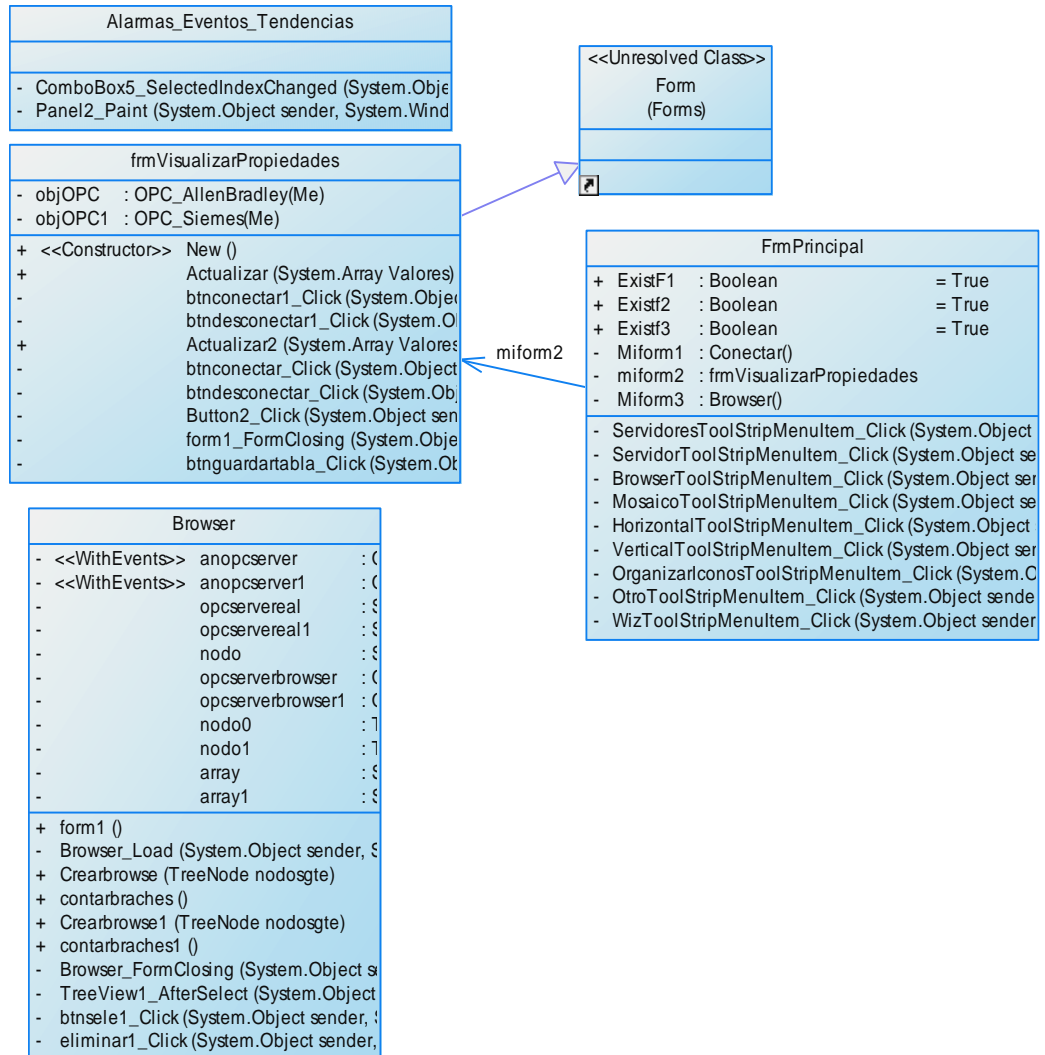


Figura 62. Clases tipo interfaz implementadas principales

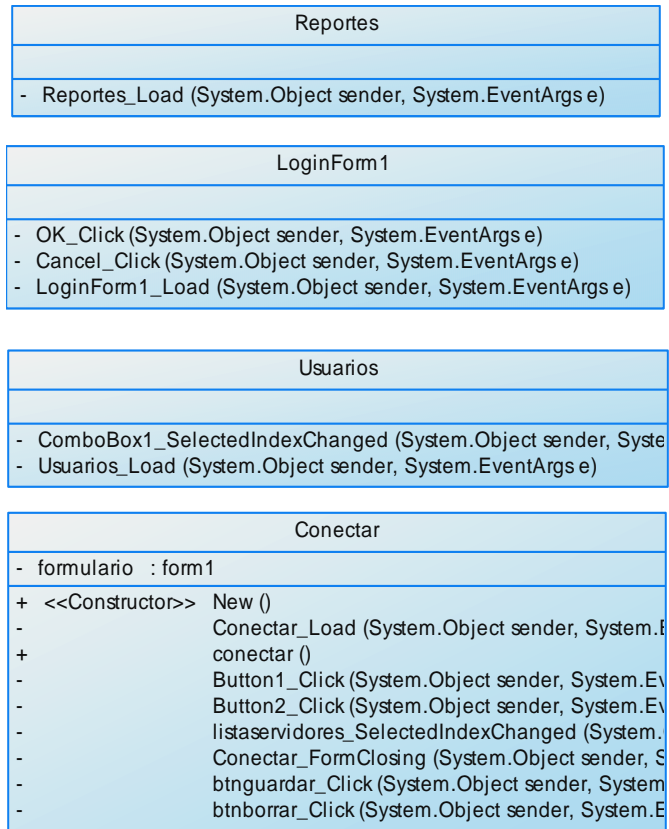


Figura 63. Clases tipo interfaz implementadas

c) Lógica de Servicios

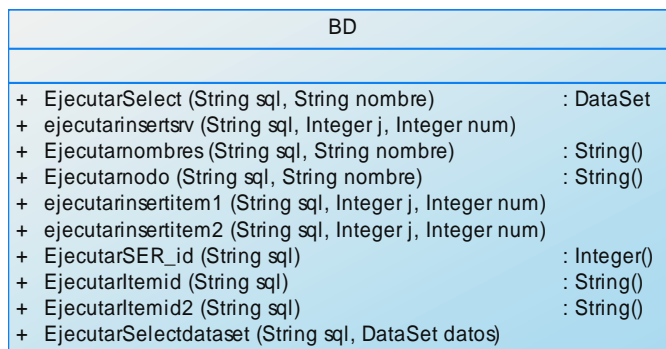


Figura 64. Clase tipo Servicios implementada

Diagrama Entidad – Relación: los diagramas Entidad-Relación, nos describen el modelado y diseño de la bases de datos. Primero se observa el modelo conceptual,

donde observamos dos entidades, **Servidores** e **Ítems**, con sus respectivas propiedades. También se observa la entidad **Usuarios** que se utiliza para el control de acceso de los usuarios. Las anteriores entidades se describen en la Tabla 11 y

Tabla 14. La entidades Ítems y Servidores comparten la llave primaria para establece relación entre el servidor guardado y sus ítems. Es decir que cada servidor debe guardar sus propios ítems.

a) Descripción de las entidades y sus Propiedades

| Propiedad | Tipo | Descripción |
|--------------------|-------------------|---|
| Ser_Id:. | Entero largo | Es la llave primaria o identificador |
| Ser_Nodo: | Caracter Variable | Guarda el nodo donde está los servidores instalados |
| Ser_Nombre: | Caracter Variable | Guarda los nombres de los servidores |

Tabla 11. Entidad Servidores.

| Propiedad | Tipo | Descripción |
|---------------------------------|-------------------|--|
| Ite_Descripcion: | Variable Caracter | Es la llave primaria de esta entidad |
| Ite_Es_Alarma | Carácter (1) | Guarda si el ítem seleccionado previamente es alarma o no |
| Ite_Nombre_Alarma | Variable caracter | Guarda el nombre de la alarma |
| Ite_Prioridad | Caracter | Guarda la prioridad de la alarma (Alta, media, Baja) |
| Ite_Rango_Normal_Inicial | Entero largo | Guarda el rango normal inicial que un ítem seleccionado como alarma puede tener. |
| Ite_Rango_Normal_Final | Entero largo | Guarda el rango normal final que un ítems seleccionado como alarma puede tener |
| Ite_Graficar | Caracter (1) | Guarda si el ítem se va a graficar (S/N) |
| Ite_Nombre_Evento | Variable Caracter | Guarda el nombre del evento |
| Ite_Es_Evento | Carácter(1) | Guarda si el ítem es evento |

| | | |
|-------------------------|--------------|--|
| | | (S/N) |
| Ite_Valor_Evento | Entero largo | Guarda el valor del evento para comparar |
| Ite_Fecha_Alarma | Date y Time | |

Tabla 12. Entidad Ítems

| Propiedad | Tipo | Descripción |
|--------------------------|-------------------|---|
| Ite_Descripcion | Variable caracter | Dirección de la variable |
| Ite_Nombre | Variable caracter | Guarda el nombre de la variable |
| Ite_Prioridad | Caracter | Guarda la prioridad de la alarma |
| Ite_Valor | Entero Largo | Guarda el valor de la alarma |
| Ite_Rango_Inicial | Entero Largo | Guarda el rango inicial de la Alarma |
| Ite_Rango_Final | | Guarda al rango final de la alarma |
| Ite_Fecha | | Guarda la fecha en que ocurre la alarma |
| Ite_Hora | | Guarda la hora en que ocurre la alarma |

Tabla 13 Entidad Alarmas.

| Propiedad | Tipo | Descripción |
|--------------------|-------------------|--|
| Usu_Login | Variable carácter | Guarda el login de usuario |
| Usu_Clave | Variable carácter | Guarda la clave de usuario |
| Usu_Nombre: | Variable carácter | Guarda los nombres de los usuarios |
| Usu_Rol | Caracter | Guarda si es administrador u operario (A /O) |

Tabla 14. Entidad Usuarios

Modelo Conceptual Servidores-Ítems: las entidades Servidores e Ítems tienen una relación padre-hija respectivamente, es decir una dependencia 1 a 1 de Ítems hacia Servidores (ítems depende de Servidores), con una cardinalidad 1 a muchos, comparten la llave primaria de la entidad Servidores, a su vez que **Ite_Descripcion** es la llave primaria de la entidad Ítems. Observamos las características mencionadas en las tablas anteriores y el modelo conceptual en la Figura 65, a su vez que Figura 66 nos muestra su modelo físico, mostrando la entidad (en este caso tabla) **Usuarios**. Servidores Guarda los dos nombres de servidores que ha elegido el administrador para

interoperar, por lo tanto cada servidor se relaciona con su tipo de Ítems determinados en cada proceso. La entidad Alarmas me ayuda a guardar y controlar las alarmas que se presentan en el proceso; y la entidad Usuarios se utiliza para guardar y controlar los diferentes usuarios que se establecen en la configuración por parte del administrador.

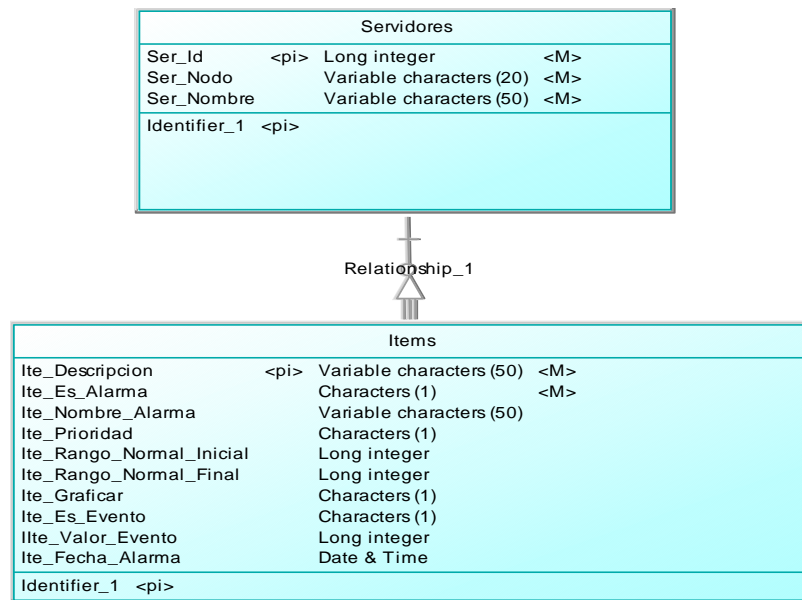


Figura 65. Modelo conceptual Servidores-Ítems.

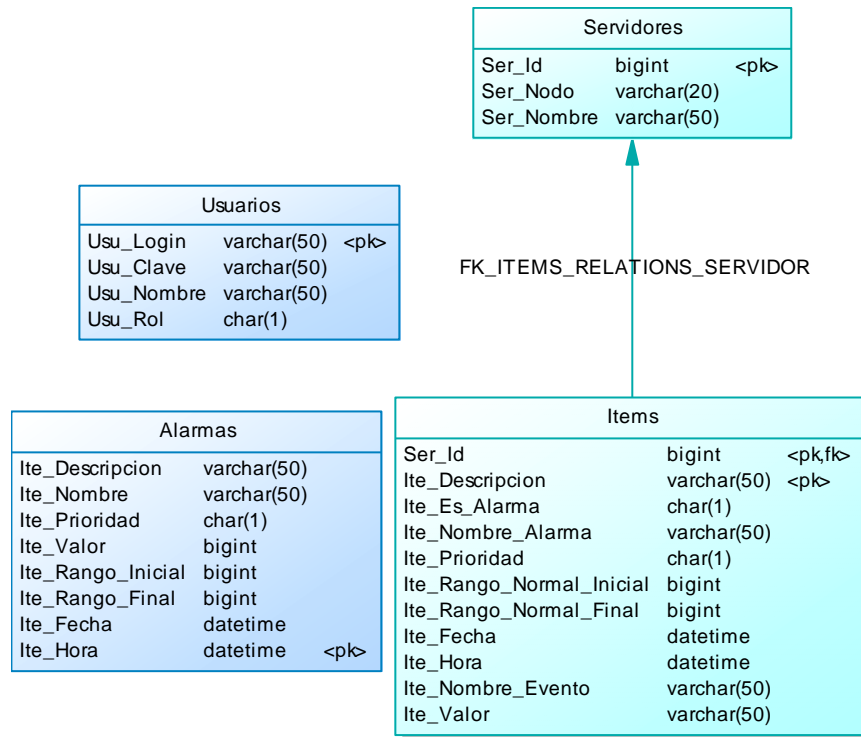


Figura 66. Modelo Físico Servidor Ítems y Usuarios

Los anteriores modelos son llevados a SQL Server 2005, para implementar la base de datos del sistema, y poder guardar, consultar y realizar diferentes operaciones con los datos guardados.

- **Modelo de Despliegue**

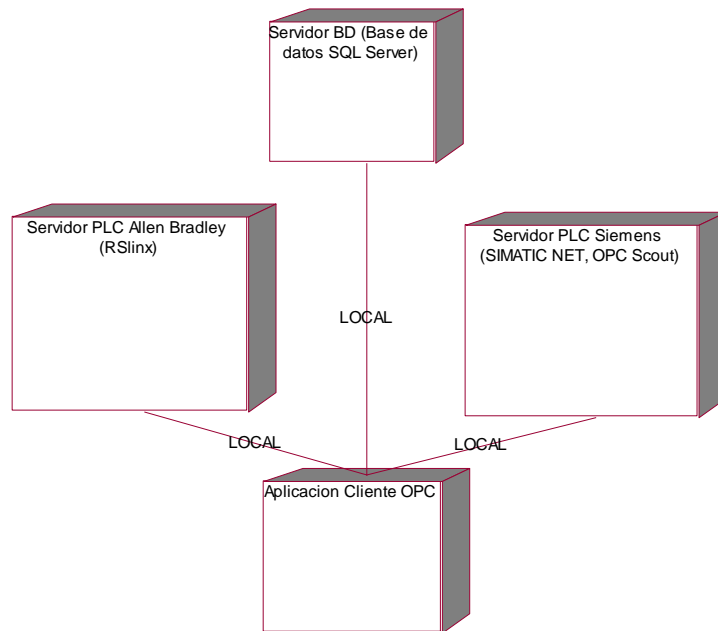


Figura 67. Modelo de despliegue.

El modelo de despliegue describe los servidores para utilizar la aplicación, se utilizaron los servidores SQL Server 2005 para el manejo de la base de datos, el Servidor de Allen Bradley RSlinx y el SIMATIC NET, en su aplicación específica OPCScout que permite simular variables para el servidor OPCSimaticNET

4.4 VALIDACION DE LA SOLUCION

4.4.1 Desarrollo de Los programas en los PLCs a interoperar

Para validar el sistema solución realizado, se utilizaron dos emuladores de PLCs de distinta marca, con sus respectivos componentes, Se describen a continuación los pasos efectuados para realizar la validación con estos sistemas:

- Para el PLC Allen Bradley, se utilizó el emulador RSLogix Emulate 500, (disponible en el departamento de Electrónica y Control de la Facultad de electrónica y Telecomunicaciones) y el servidor RsLinx que posee el respectivo Servidor OPC, se realizó el programa en escalera (Ladder) a través del Rs Logix500, en la Figura 68 se muestra esta herramienta.

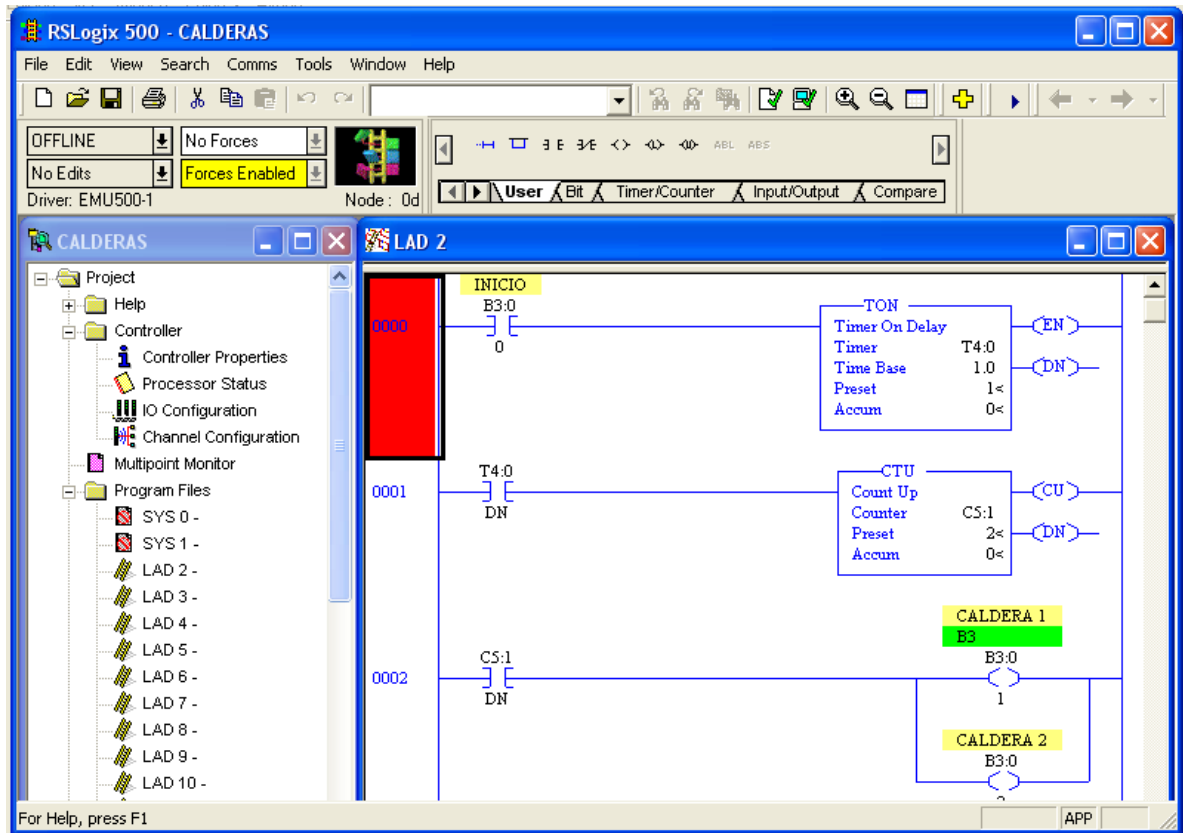


Figura 68. Programación del PLC con RSLogix 500

- Luego este programa se carga en el RSLogix Emulate, para emular el comportamientos de PLC respectivo, la Figura 69 muestra el RSLogix Emulate

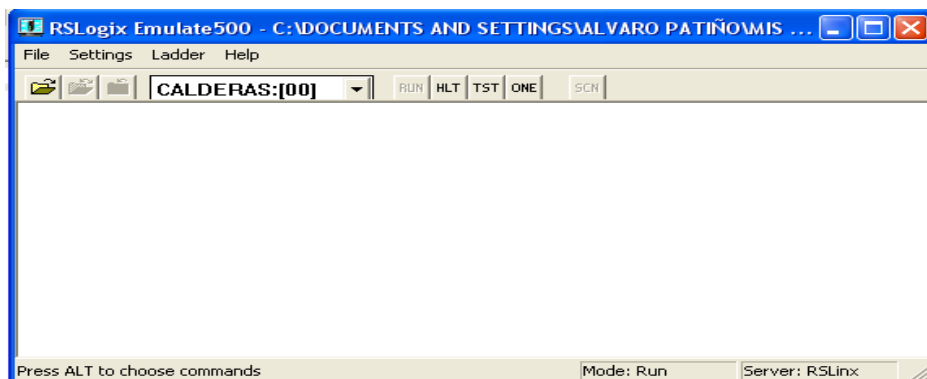


Figura 69. RSLogix Emulate

- Anteriormente se ha instalado y se encuentra corriendo en el PC (Computador personal) el Servidor RSLinx el cual tiene incluido el Servidor OPC que se va a comunicar con la aplicación realizada, la figura muestra el servidor RSLinx. En este

servidor se realiza la comunicación con el emulador del PLC respectivo, en este caso es el SLC 500.

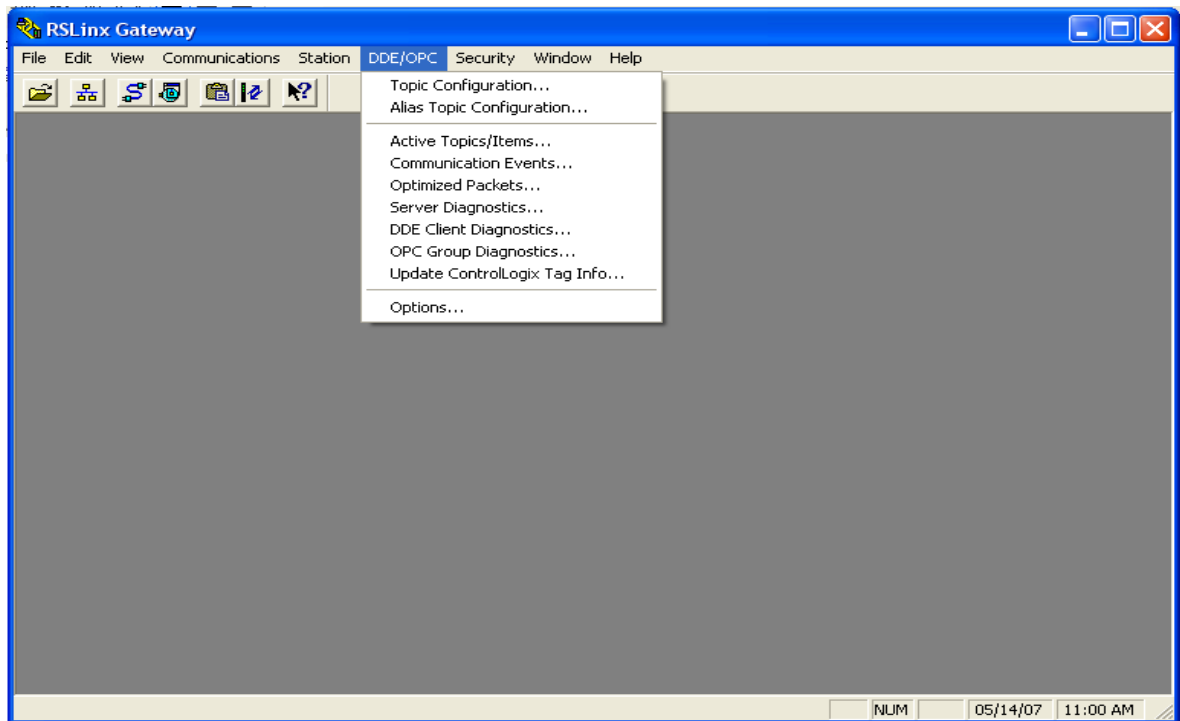


Figura 70. Servidor RSLinx.

- Para el PLC Siemens se utilizó una herramienta de SIMATIC NET, el OPC Scout, que nos permite simular variables de diferente tipo para un servidor OPC. en esta herramienta se configura y seleccionan los grupos y los ítems que se quieren visualizar y manipular en un cliente OPC. Esta herramienta nos permite simular variables de proceso para validar la herramienta (debido a que el Departamento de Electrónica y Control no posee PLCs Siemens, se propuso desde el principio un simulador de prueba disponible gratuitamente en Internet [12]). Como se muestra en la Figura 71 , este simulador OPC permite generar valores, seleccionando el tipo (entero, bit, arreglo) y la forma de incremento del valor (aleatoria, creciente, decreciente etc.)

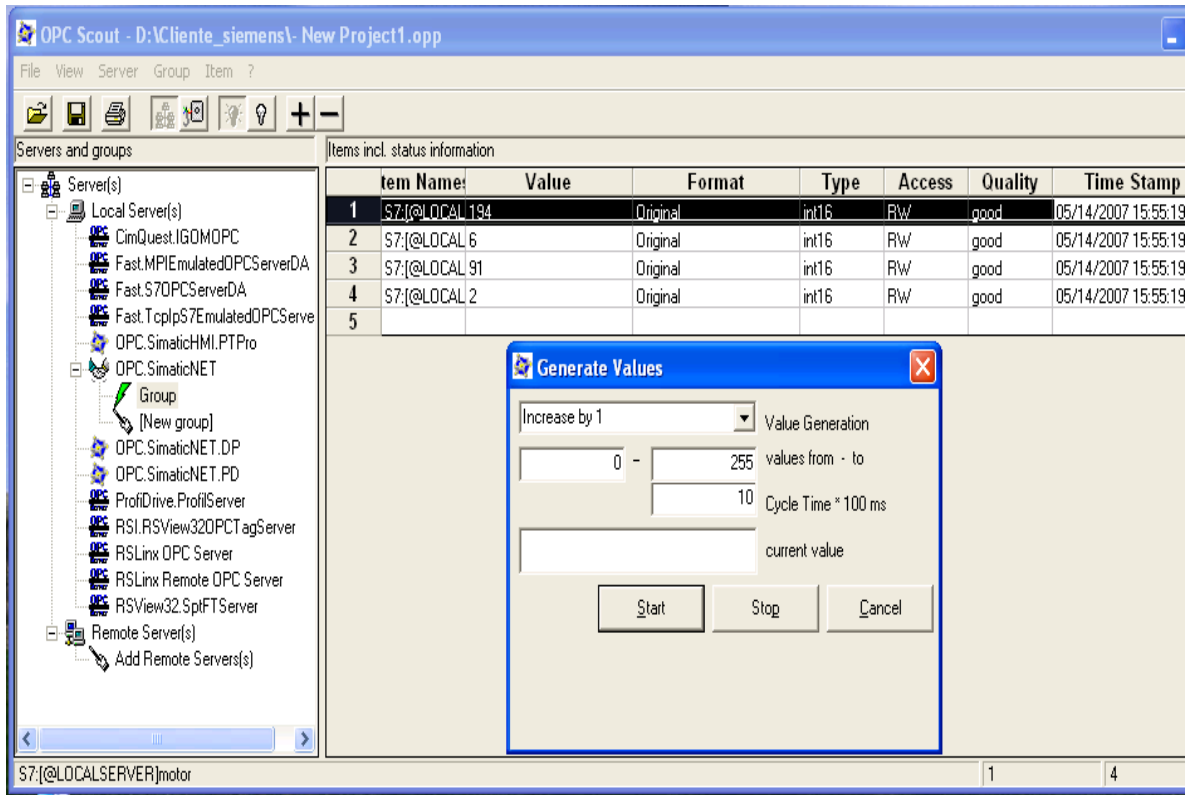


Figura 71. Simulador para Servidor OPC Siemens (OPC Scout)

- Por último se integran los dos módulos de Programas en un solo proceso (se integran las variables configuradas en el OPC Scout y el proceso programado en RSLogix 500) en la aplicación OPC realizada en Visual Basic.NET.

4.4.2 Descripción del Proceso Utilizado en la Validación

El proceso comienza calentando dos calderas hasta una determinada temperatura, en la cual se abrirán las válvulas y las sustancias calentadas empezaran a fluir hacia el tanque de mezclado. En el tanque de mezclado hay un sensor de nivel que me indica un valor para empezar a mezclar las sustancias. Cuando se halla realizado el mezclado durante un cierto tiempo la válvula dosificador se abrirá e iniciará el proceso de llenado de recipientes. En este subproceso, un sensor detecta si llegó un recipiente y arranca el motor de una segunda banda transportadora. Un sensor de fin de carrera indica que un recipiente lleno, llegó al punto de sellado y por lo tanto es sellado (máquina selladora), al ser sellado la máquina selladora indica con otro sensor que ha acabado de sellar (on/off) y es expulsado hacia otro lugar del proceso.

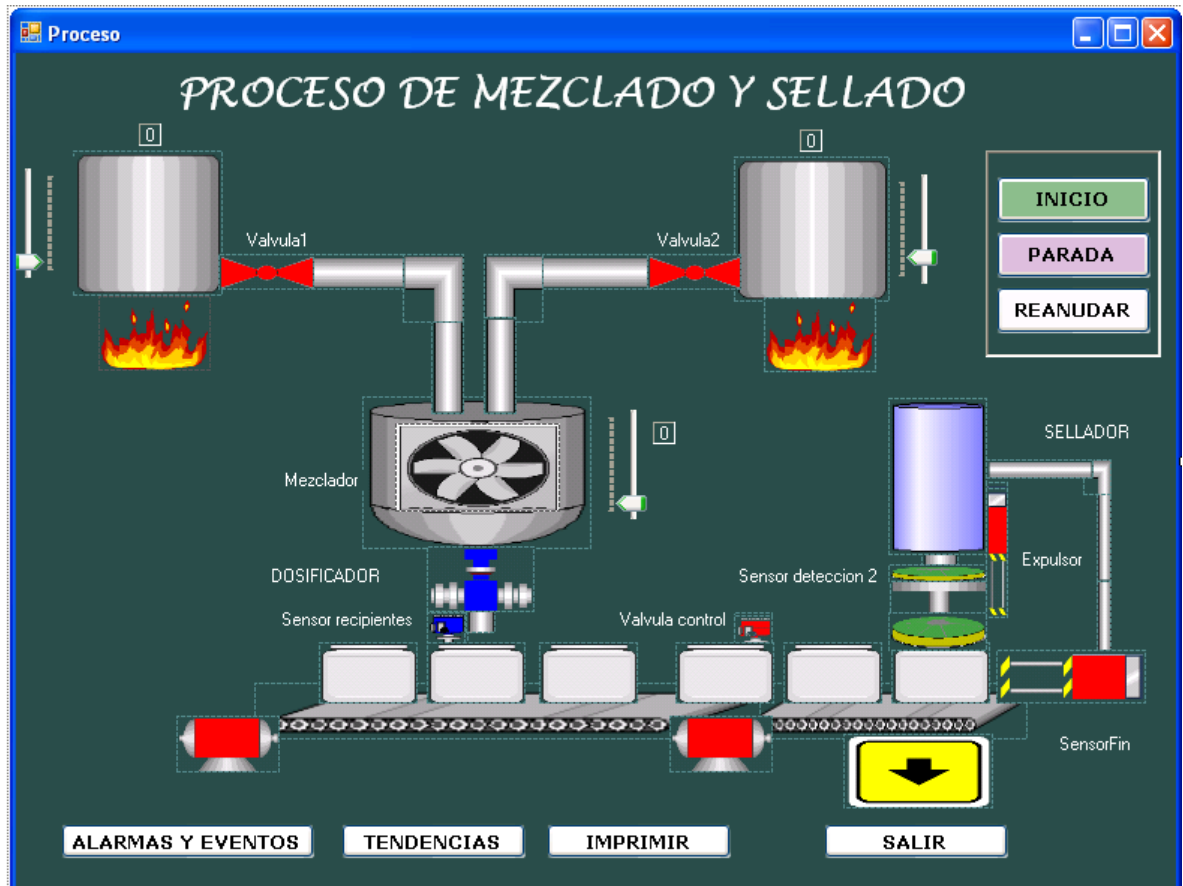


Figura 72. Interfaz HMI de operario para el Proceso

Lo que se pretende es que el proceso sea controlado por dos PLCs de distinto fabricante, en este caso Siemens y Allen Bradley, en distintas etapas, e interoperen para lograr un proceso eficiente, además de que haya una supervisión del trabajo que realiza cada uno.

4.4.3 Ejecución de Pruebas

La siguiente tabla muestra las pruebas que se realizaron en el sistema para evaluar su estabilidad, funcionalidad y correcto funcionamiento según los casos de uso.

- **Ejecución de pruebas de interfaz**

| | |
|--------------------------------|-------------------------------|
| Nombre e identificador: | Prueba 1 |
| Prueba: | Caso de uso Control de Acceso |
| Fecha de realización | Mayo 2007 |

| | |
|-------------------------------|--|
| Resultado de la prueba | Si se puede escribir Nombre, login, Clave y Rol en la interfaz de control de acceso. |
| Observaciones | Ninguna |

| | |
|--------------------------------|--|
| Nombre e identificador: | Prueba 2 |
| Prueba: | Caso de uso Buscar Servidores |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | La interfaz permite buscar el servidor, despliega la lista, y permite con el botón indicado con flecha seleccionar los servidores y luego guardarlos, presenta el mensaje "Registro satisfactorio", con el cual indica que se guardaron los nombres en la base de datos. |
| Observaciones | Ninguna |

| | |
|--------------------------------|---|
| Nombre e identificador: | Prueba 3 |
| Prueba: | Caso de uso Visualizar Browser |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | La interfaz permite buscar las direcciones de las variables programadas en los Plc's, de una manera muy práctica para el usuario, como también la selección de las direcciones que son alarmas, eventos y tendencias. Cuando son guardadas en la base de datos se presenta el mensaje "Registro satisfactorio". |
| Observaciones | Ninguna |

| | |
|--------------------------------|--|
| Nombre e identificador: | Prueba 4 |
| Prueba: | Caso de uso Visualizar Propiedades |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Con el botón conectar permite visualizar las propiedades de las direcciones almacenadas en la interfaz anterior; además permite tener conectado los dos servidores OPC al mismo tiempo, observándose un buen comportamiento del sistema. |
| Observaciones | Ninguna. |

| | |
|--------------------------------|---|
| Nombre e identificador: | Prueba 5 |
| Prueba: | Caso de uso Configurar Alarmas y Eventos |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Esta interfaz permite al usuario ir filtrado la información de la direcciones seleccionadas en el |

| | |
|----------------------|---|
| | browser, lo que facilita al usuario la manipulación del sistema |
| Observaciones | Ingresar el caso de uso Configurar Tendencias. |

| | |
|--------------------------------|--|
| Nombre e identificador: | prueba 6 |
| Prueba: | Caso de uso Configurar Tendencias |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Las pruebas muy similar al anterior por lo que las interfaces se agruparon, teniendo una mejor presentación para el usuario. |
| Observaciones | Ninguna. |

| | |
|--------------------------------|--|
| Nombre e identificador: | Prueba 7 |
| Prueba: | Caso de uso Visualizar |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | En este caso de uso se observó el buen comportamiento del mímico; como también se manipuló los diferentes botones a los que se tiene acceso en esta interfaz, observando las alarmas, eventos, reportes y la grafica de tendencia con un buen resultado. Se utilizó el botón inicio para que empezara a funcionar el mímico, el botón parada y reanudar obteniendo un buen resultado el sistema. |
| Observaciones | Ninguna. |

Tabla 15. Pruebas de Interfaz

- **Ejecución Pruebas de Funcionalidad**

| | |
|--------------------------------|---|
| Nombre e identificador: | Prueba 8 |
| Prueba: | Caso de uso Control Acceso |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Al ingresar nombre "operador", el login "directo" y contraseña "prueba", el sistema no aceptó al usuario y no lo dejó ingresar a ninguna interfaz por que no estaba registrado en la base de datos. |
| Observaciones | Ninguna. |

| | |
|--------------------------------|-------------------------------|
| Nombre e identificador: | prueba 9 |
| Prueba: | Caso de uso Buscar Servidores |
| Fecha de realización | Mayo 2007 |

| | |
|-------------------------------|---|
| Resultado de la prueba | Se selección un servidor y se guardó en la base de datos, luego se seleccionó el otro servidor y se guardó. |
| Observaciones | Ninguna. |

| | |
|--------------------------------|---|
| Nombre e identificador: | prueba 10 |
| Prueba: | Caso de uso Visualizar Browser |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Se seleccionaron diferentes direcciones de los servidores OPC, luego se chequearon, se devolvieron al cuadro de Itemid. |
| Observaciones | Ninguna. |

| | |
|--------------------------------|---|
| Nombre e identificador: | prueba 11 |
| Prueba: | Caso de uso Visualizar propiedades |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Se conectó y desconectó varias veces las diferentes conexiones a los servidores OPC, no se provocó ningún error |
| Observaciones | Ninguna |

| | |
|--------------------------------|---|
| Nombre e identificador: | prueba 12 |
| Prueba: | Caso de uso Configurar Alarmas y Eventos |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Se guardaron una a una las direcciones que son alarmas, si no se los coloca los valores sale error y no permite ser guardada en la base de datos. |
| Observaciones | Ninguna |

| | |
|--------------------------------|--|
| Nombre e identificador: | prueba 13 |
| Prueba: | Caso de uso Configurar Tendencias |
| Fecha de realización | Mayo 2007 |
| Resultado de la prueba | Se buscó la dirección existente se seleccionó gráfica y el nombre de la gráfica, no se causó ninguna clase de error. |
| Observaciones | Ninguna. |

Tabla 15. Pruebas de Funcionalidad

4.4.4 Manuales de usuario

El manual de usuario se describe en el ANEXO A

5. APORTES DEL PROYECTO REALIZADO

Aunque en las conclusiones respectivas se mostrará en detalle los avances y logros obtenidos con la aplicación OPC, se mencionan en este aparte los aspectos técnicos y de lógica de implementación que se destacan y constituyen un aporte significativo en las aplicaciones OPC actuales y en el uso de la especificación OPC interfaz de automatización, además de la integración e interoperabilidad buscada y que surgen de enfrentar ciertos problemas en el transcurso de desarrollo del sistema. Los siguientes aspectos en particular mencionan estos aportes:

- En cuanto a la lectura en tiempo real de las variables (ver Figura 34 y Figura 35) seleccionadas en el Browser (variables de los PLCs), la especificación implementa tres formas de leer las variables, Lectura asincrónica (asynconRead), lectura sincrónica (synchronRead) que son métodos del objeto Grupo OPC (OPCGroup) y Datachange que es un evento del objeto Grupo OPC. En una primera aproximación en la lectura de datos se utilizó el asynconRead, pero este método de la especificación solo es posible utilizarlo, cuando se encuentra el dispositivo físico conectado para establecer la lectura, es decir si la fuente de datos es física, lo cual es mencionado en la especificación OPC con el método OPCDataSource (OPCDataSource.OPCDevice). Se estudió la aplicación del segundo método de lectura, llamado Synchronread, pero el establecer una lectura sincrónica como su nombre lo indica, implicaba que el cliente o el sistema contara con un timer para realizar la comunicación (frecuencia de lectura), lo que causaba demora en la lectura de datos. Debido a estos inconvenientes se exploró el método Datachange, el cual es activado por el servidor cuando hay cambios en los valores de los ítems del grupo; aprovechado esta utilidad y otro método de la especificación que está a nivel de objeto **ítem** (asynconRead y SynchronRead esta a nivel de Grupo), llamado "Read", se realizó una combinación de estos dos métodos para poder realizar una lectura asincrona en la memoria cache (OPCDataSource.OPCCache) con excelentes resultados ya que se obtuvo una lectura en tiempo real, además de poder adicionar gran cantidad de ítems de cada PLC para ser leídos al mismo tiempo. (la cantidad dependería de la memoria del computador).
- Al principio del desarrollo de la aplicación se tenía que escribir las direcciones de cada PLC para poder leer o escribir los ítems relacionados. En la implementación del caso de uso Visualizar Browser (ver Figura 32 y Figura 33), se logró implementar todos los métodos descritos en la especificación para obtener un Browser (buscado o explorador en forma de árbol) que permite buscar las variables de cualquier tipo de servidor OPC y de manera automática se pueden encontrar las direcciones completas (sin necesidad de conocer la forma de direccionamiento de un determinado PLC) de cada PLC para luego ser utilizadas según los requerimientos. Otra importancia relevante del Browser, es que se utiliza para dos PLCs, permitiendo claramente realizar una configuración que muestra el proceso de interoperabilidad de los dos sistemas y es escalable a más de dos PLCs. Generalmente las aplicaciones existentes que implementan este objeto se limitan a conectarse a un servidor OPC y a mostrar sus ítems, pero generalmente no permiten visualizar dos conexiones y mostrar un proceso de integración horizontal.

- En cuanto al manejo de las alarmas y eventos, se logró realizar una interfaz que interactúa con la base de datos, en donde se guardan las condiciones y atributos para que se cumplan estos dos aspectos, lo cual permite que sea fácilmente configurado el sistema.
- Se incluyeron funciones adicionales como “Configurar Usuarios” que permite adicionar y guardar diferentes usuarios que manejan el sistema, diferenciados en dos tipos Operario u Administrador. También se adicionó “Eliminar Configuración” que me permite eliminar usuarios establecidos anteriormente y borrar la configuración guardada en la base de datos, para así poder generar una nueva configuración.

6. CONCLUSIONES

- En el presente trabajo se ha mostrado que la especificación OPC interfaz de automatización acceso a datos permite gran versatilidad a la hora de implementar una aplicación OPC, en este caso un cliente que se conecta a dos servidores. Esto se evidencia en la etapa de diseño, pues a pesar de partir de unos objetivos definidos y unos requerimientos basados en los casos de uso (desde la perspectiva del usuario) la norma permitió implementar libremente la aplicación sin cohibir o limitar el desarrollo de la aplicación. Simplemente como se observa en los diferentes diagramas, se estableció la dinámica entre objetos a partir de los casos de uso y el desarrollo de las operaciones y se tuvo en cuenta la estructura de objetos OPC, Servidor, Grupo e ítems (Tabla 1) implementando cada método y atributo, hasta lograr que el sistema realizará las funciones previstas.
- Toda la etapa de diseño UML, plantea una metodología de desarrollo para implementar clientes OPC, la cual hasta el momento no se encuentra en la literatura sobre OPC. Se observa en la etapa de diseño claramente que a partir de los requerimientos, de los casos de uso y de unos objetivos claros, se obtienen los demás diagramas de diseño, hasta lograr la implementación en Visual Basic.NET. Se inició con la formulación clara de los casos de uso, estudiando los escenarios (Tabla 7), mas concretamente estudiando la descripción de las acciones de los actores. Luego obteniendo de estas, operaciones explícitas, a las cuales se les asignan unas responsabilidades mediante el uso de los contratos. Los contratos fueron una herramienta poderosa, para que a partir del análisis de la sintaxis de las poscondiciones y basados en la especificación, se obtuvieran los objetos de diseño dinámico, que describen la comunicación a través de mensajes que luego determinarán métodos dentro de las clases. Como se observa cada paso en el diseño esta ligado al anterior pero todos se enmarcan en los casos de uso, no perdiendo así de vista los objetivos iniciales y logrando obtener una aplicación que cumple satisfactoriamente los requerimientos de integración e interoperabilidad.
- El utilizar la plataforma de Visual Studio.NET en la implementación y programación, conlleva implícitamente a que la aplicación pueda ser utilizada para integración vertical, ya que esta plataforma esta basada en servicios Web, servicios de interoperabilidad y establece un sistema común para aplicaciones realizadas bajo Windows; por lo tanto se pueden realizar conexiones con otro tipo de aplicaciones cliente o servidor en otros niveles de planta, tomando como referencia la especificación OPC y la especificación OPC UA (arquitectura unificada) que se encuentra en desarrollo actualmente.
- El sistema resulta altamente escalable, esto se puede evidenciar claramente, por la arquitectura en tres capas que permite que el código sea reutilizado y se puede seguir trabajando extensivamente sobre la aplicación, en otras funciones o mejorando las ya existentes. Y en segundo lugar por que no solo se pueden inter operar los dos PLCs mencionados en la validación, si no cualquier tipo de PLC que implemente la

especificación OPC, además de que la estructura interna del programa realizado y su orden permite fácilmente implementar una integración para más de dos PLCs.

- El diseño visual permite un fácil manejo de la configuración del sistema, por su facilidad y capacidad de mostrar claramente la interoperabilidad explícita entre dos dispositivos, garantizando así una gran funcionalidad y servicio para el usuario a través de la interfaz MDI (interfaz dinámica)
- El diseño de las bases de datos permite realizar cambios si se presentan nuevos requerimientos, y su implementación facilita el manejo adecuado de los datos de los PLCs, para posterior manipulación y configuración en un sistema SCADA. Permitiendo así, configurar, seleccionar y dar atributos a los ítems, según las necesidades del proceso y de la supervisión.
- Se implementó en cuanto a la comunicación de los PLCs una lectura de variables en tiempo real, lo cual es un requerimiento esencial en los sistemas de automatización industrial. Además de que esta lectura se realiza en ambos PLCs, al mismo tiempo, permitiendo obtener los datos de manera óptima, e integrarlos en un sistema supervisorio, con capacidad de mostrar alarmas, eventos, tendencias y reportes, además de un sistema que muestra gráficamente el proceso para efectos de validación.
- La aplicación realizada amplía los horizontes de sistemas bajo plataformas Windows en las redes industriales y específicamente en el desarrollo de aplicaciones cliente bajo OPC.

7. REFERENCIAS

- [1] VIDAL Pablo y FIGUEROA Sergio “OPC: Un estándar en la redes industriales y redes de campo”. Artículo. Universidad de la Frontera. Chile. 2000.
- [2] Industrial Ethernet Book Article. OPC Foundation 2006. URL <http://www.opcfoundation.org>.
- [3] VIDAL Fabián Yesid y MUÑOZ Libardo Steven. “Aplicación de la norma ISA 95 a un caso de estudio”. Proyecto de Grado. Universidad del Cauca 2007.
- [4] OPC Foundation. What is OPC?. URL http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC
- [5] Escuela Técnica superior de Ingeniería, OPC Conceptos Fundamentales. OPC%20Conceptos%20Fundamentales.pdf. 2001.
- [6] OPC Technology, COM/OLE based technology for Process Control. Artículo. URL <http://www.TheCodeProject.com> , <http://www.suchit-tiwari.org/>
- [7] Data access Automation interface Standar 2.02. OPC Foundation. 2002.
- [8] Serrano, C. E. Construcción de Soluciones. Modelo de Referencia para el Desarrollo de Proyectos v.1.1.1. Universidad del Cauca. 2002.
- [9] Marlon Martínez. Cliente OPC realizado en Visual Basic.NET. URL <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art123.asp>
- [10] LARMAN Craig. UML y Patrones, Introducción al análisis y Diseño orientado a objetos. 1 Edición. ISBN 970-17-0261-1. Editorial PRENTICE HALL, México, 1999.
- [11] BOOCH Grady, RUMBAUGH James y JACOBSON Ivar. El Lenguaje Unificado de Modelado Guía del Usuario. 2 Edición. ISBN 10: 84-7829-076-5. Editorial PEARSON EDUCACIÓN, S.A. Madrid 2006
- [12] Siemens España. URL <http://www.Siemens.com.es>

8. BIBLIOGRAFIA

Control-Magazine An article from 2004, 'Unified architecture. The future of OPC', written by OPC Technical Director, Jim Luth.

Estrada Mauricio, URIGO Inteligencia de planta Mayo 3 del 2006, inteligenciadeplantamestrada@urigo.com.

Barrantes Muñoz Alexander. Interfaz Humano-Máquina para la familia de PLC's Unitronics. pb0438t.pdf

Lozano de Castro Carlos. Interfaz Hombre Maquina, Introducción a SCADA. Introducci%F3n%20a%20los%20SCADAS.pdf

Bases de Datos. Lafacu. URL http://www.lafacu.com/apuntes/informatica/base_datos/

