

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Título:

Configurador para el Servidor OPC de Acceso a Datos e
integración con el Recolector del SCADA “Guardián del
ALBA”

Autor:

Yolier Galán Tassé.

Tutor:

Ing. Yunier Velázquez Batista.

La Habana 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yolier Galán Tassé

Ing. Yunier Velázquez Batista

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Nombre y apellidos: Yunier Velázquez Batista.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero Informática

e-mail: yunierve@uci.cu

Cargo: Líder de la línea de Integración y Despliegue del Centro de Desarrollo de Informática Industrial.

AGRADECIMIENTOS

Agradezco a la revolución cubana por darme la posibilidad de estudiar en esta extraordinaria universidad y a su creador Comandante en Jefe Fidel Castro.

A mis padres Marcelina Tassé Rosabal y José E. Galán Vázquez por darme todas sus confianzas

A mi novia Yaiset Moreno Aleaga por apoyarme y depositar su confianza en mí.

A todos mis amigos por estar a mi lado cuando más lo necesitaba, los cuales nunca olvidaré y espero verlos en el transcurso de la vida.

A mi tutor Ing. Yunier Velázquez Batista porque supo guiarme en todo el desarrollo del trabajo y por su ayuda incondicional.

A todas aquellas personas que de una u otra forma contribuyeron a la realización del trabajo.

A TODOS MIS MÁS SINCEROS AGRADECIMIENTOS.

DEDICATORIA

A mis padres, a toda mi familia y a mi novia.

RESUMEN

Este trabajo se realiza con el objetivo de proveer al Centro de Desarrollo de Informática Industrial (CEDIN) de la facultad 5 en la Universidad de las Ciencias Informáticas un servidor OPC (OLE for Control Process) de Acceso a Datos, que pueda recolectar variables mediante los manejadores desarrollados para el Centro.

Para ello se realiza un análisis sobre el estándar OPC describiendo brevemente cada una de las especificaciones que brinda y se enfatiza en la especificación de Acceso a Datos. Además, se da una breve reseña de los principales proveedores de soluciones OPC en el mundo, cuyos productos que distribuyen son bajo licencia propietaria. También se describen las tecnologías, herramientas, lenguajes y la metodología para el desarrollo de software.

Se diseña e implementa un configurador de fácil uso e intuitivo que permite crear la configuración del servidor OPC y estructurar el espacio de direcciones del mismo. Además, se integra el componente de adquisición de datos desarrollado en el CEDIN y el toolkit OPCWorkshop utilizado para el desarrollo de servidores OPC DA.

Como resultado se puede decir que el CEDIN cuenta con un Servidor OPC DA de fácil configuración y capaz de recolectar variables mediante los manejadores ya implementados, brindándole a los clientes OPC sin que estos tengan la necesidad de conocer los protocolos de comunicación de las fuentes de orígenes.

Palabras claves: Estándar OPC, OPC DA, Configurador, Componente.

ÍNDICE

RESUMEN	6
ÍNDICE	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA. ESTADO DEL ARTE	14
1.1 Introducción.....	14
1.2 OPC.....	14
1.2.1 Surgimiento de OPC.....	14
1.2.2 Definición	15
1.2.3 Arquitectura.	16
1.2.4 Especificaciones.....	16
1.2.4 Desarrollo de OPC en el mundo	19
1.2.5 Próxima generación de OPC.....	20
1.3 Tecnologías, metodología, lenguajes y herramientas de desarrollo.	21
1.3.1 Metodología RUP (Rational Unified Process).....	21
1.3.2 Herramientas de desarrollo	22
1.3.2.1 Visual Paradigm para UML	22
1.3.2.2 Microsoft Visual C++ 2005	22
1.3.2.3 Framework Qt.....	23
1.3.3 Lenguajes	23
1.3.3.1 Lenguaje de Modelado UML	23
1.3.3.2 Lenguaje de programación: C++.....	23
1.4 Conclusiones parciales.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1 Introducción.....	25
2.2 Soluciones Técnicas.....	25

2.3 Modelo del dominio	26
2.3.1 Clases conceptuales o Glosario de términos del dominio	27
2.3.2 Requisitos funcionales	27
2.3.3 Requisitos No Funcionales.....	29
2.3.4 Descripción del sistema propuesto.....	30
2.4 Conclusiones parciales	48
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN.....	49
3.1 Introducción.....	49
3.2 Diseño del sistema.....	49
3.2.1 Diseño del subsistema “Configurador OPC”	49
3.2.2 Diseño del subsistema “Servidor OPC”	53
3.3 Implementación del sistema.	54
3.3.1 Estilo de código.	54
3.3.2 Diagramas de componentes.	57
3.4 Conclusiones parciales.....	60
CONCLUSIONES GENERALES	61
RECOMENDACIONES.....	62
REFERENCIAS BIBLIOGRÁFICAS	63
BIBLIOGRFÍA.....	64
ANEXOS	66
Anexo 1: Descripción del CU Cargar configuración en el configurador.....	66
Anexo 2: Pruebas realizadas al sistema.	68
GLOSARIO DE TÉRMINOS Y ABREBIATURAS (GTA)	77

ÍNDICE DE TABLAS

Tabla 1: Descripción de actores	31
Tabla 2: Clasificación de los Casos de uso del sistema.....	31
Tabla 3: Descripción del CU Gestionar Canal	34
Tabla 4: Descripción del CU Gestionar Subcanal.....	37
Tabla 5: Descripción del CU Gestionar Dispositivo	40
Tabla 6: Descripción del CU Gestionar Variable.....	43
Tabla 7: Descripción del CU Gestionar Grupo OPC	45
Tabla 8 : Descripción CU Guardar Configuración.....	47
Tabla 9: Descripción del CU Iniciar publicación OPC.....	48
Tabla 10: Descripción de las clases del paquete View.....	51
Tabla 11: Descripción de las clases del paquete Objeto.	53
Tabla 12 Descripción de las clases del subsistema Servidor OPC.....	54
Tabla 13: Descripción del CU Cargan configuración en el Configurador.....	67
Tabla 14 Casos de pruebas realizados	73

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Red donde no se cumple con el estándar OPC.....	14
Ilustración 2: Solución con OPC.....	15
Ilustración 3: Relación Cliente/Servidor de OPC	16
Ilustración 4: Interfaces de OPC.....	17
Ilustración 5: Modelo de Dominio.	27
Ilustración 6: Diagrama de casos de uso del sistema.....	30
Ilustración 7: Arquitectura en capa del subsistema Configurator OPC	50
Ilustración 8: Diagrama de clases del paquete View.	51
Ilustración 9: Diagrama de clases del paquete Objetos.....	52
Ilustración 10: Arquitectura en capa del Subsistema Servidor OPC.....	53
Ilustración 11: Diagrama de clases del subsistema Servidor OPC.	54
Ilustración 12: Diagrama de componente del paquete View.	58
Ilustración 13: Diagrama de componentes del paquete Object.	58
Ilustración 14: Diagrama de componentes del subsistema Servidor OPC.....	59
Ilustración 15: Diagrama de componentes del sistema.	59
Ilustración 16: Caso de prueba 1.....	73
Ilustración 17: Caso de prueba 2.....	74
Ilustración 18: Caso de prueba 3.....	75
Ilustración 19: Caso de prueba 4.....	76
Ilustración 20: Caso de prueba 5.....	76

INTRODUCCIÓN

Con el vertiginoso avance de la automatización en la industria, se han creado aplicaciones de software especialmente diseñadas para el control de la producción, proporcionando comunicación entre los dispositivos de campos (controladores autónomos y autómatas programables) y los ordenadores. Estas aplicaciones destinadas para el control supervisorio y adquisición de datos, por sus siglas en inglés SCADA (**S**upervisory **C**ontrol **a**nd **D**ata **A**cquisition), cumplen con un conjunto de características como es la recolección, procesamiento y visualización de los datos mediante mímicos o informes; almacenamiento de datos históricos, manejo de alarmas y comandos, establecimiento de políticas de seguridad, etc. Para potenciar el desarrollo de este tipo de sistemas, y estandarizar el manejo de la información, surge el estándar OPC como resultado de un grupo de trabajo integrado por proveedores mundiales de instrumentación y sistemas de control de procesos, inicialmente formado por Fisher-Rosemount, Rockwell Software, Opto 22, Intellution e Intuitive Technology. (1)

OPC, del inglés OLE for Process Control, es una serie de especificaciones estándares, resultado de la colaboración de varios de los principales proveedores de automatización de todo el mundo que trabajan en cooperación con Microsoft. Basado originalmente en las tecnologías OLE de Microsoft COM (Component Object Model) y DCOM (Distributed Component Object Model), las especificaciones definen un conjunto estándar de objetos, interfaces y métodos para su uso en el control de procesos y aplicaciones industriales de automatización para facilitar la interoperabilidad (2). Dentro de sus especificaciones la principal y más importante es la de acceso a datos “OPC Data Access” (OPC DA).

En el Centro de Desarrollo de Informática Industrial (CEDIN), perteneciente a la Universidad de las Ciencias Informáticas (UCI) se ha desarrollado en colaboración con empresas venezolanas el SCADA “Guardián del ALBA” (GALBA), para el que se implementó un servidor OPC DA con la finalidad de publicar variables a terceros habilitados con el estándar OPC. Otras de las características importantes del SCADA GALBA es que posee un manejador para la recolección de variables desde servidores OPC DA, por lo que habilita al SCADA como un cliente de OPC DA.

La adquisición de datos del Servidor OPC DA depende de los servicios de adquisición, procesamiento, comunicaciones y configuración del SCADA, por lo que un mal funcionamiento de alguno de estos servicios afecta la publicación de variables en el Servidor OPC DA, ya que este no

maneja la comunicación con los dispositivos de campo. Además, se necesitan recursos de cómputo donde se ejecuten tanto el SCADA como el servidor OPC en sistemas operacionales diferentes, debido a que OPC es dependiente de la tecnología COMDCOM de Microsoft.

En algunos escenarios se han presentado situaciones donde por razones de accesibilidad a dispositivos de campo se dificulta el control de variables por el SCADA, teniendo que recurrir a la recolección de variables desde servidores OPC desarrollados por terceros, y en algunos casos los SCADA existentes en las instalaciones no implementan servidores OPC DA.

Ante la problemática planteada se define el siguiente **problema científico**:

¿Cómo contar con un Servidor OPC de Acceso a Datos independiente de los servicios del SCADA “Guardián del ALBA”?

Se tiene como **objeto de estudio**: El estándar OPC para el manejo de información.

Teniendo como **campo de acción**: Los servidores OPC de Acceso a Datos.

Para dar solución al problema científico se propone como **objetivo general**: Integrar el módulo de adquisición al Servidor OPC de Acceso a Datos y desarrollar la aplicación de configuración.

Tareas de investigación:

1. Estudio sobre las metodologías y herramientas de desarrollo a utilizar para buenas prácticas de desarrollo.
2. Estudio sobre el framework QT para el desarrollo de las interfaces gráficas.
3. Diseño e implementación de interfaces que permitan la integración del módulo de adquisición y el Servidor OPC DA.
4. Diseño e implementación de un Configurador que permita la edición de variables para el Servidor OPC DA.

Durante la realización de este trabajo se utilizarán diferentes métodos científicos para estudiar las características de objeto de investigación:

Métodos Teóricos:

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

- **Analítico-sintético:** Para la extracción de las características, rasgos y elementos más importantes para un servidor OPC DA dentro de los sistemas SCADA.
- **Análisis histórico lógico:** Para inspeccionar sobre la evolución de los servidores OPC DA para sistemas SCADA tanto en Cuba como en el resto del mundo y además de conocer las tendencias actuales sobre el protocolo OPC.

Método Empírico:

- **Revisión de la documentación:** Para seleccionar la información necesaria en la investigación a partir del estudio de documentos y diferentes bibliografías.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ESTADO DEL ARTE

1.1 Introducción

En el presente capítulo se realiza un análisis del estado del arte de los servidores OPC, haciendo énfasis en especificaciones OPC desarrolladas para el acceso a datos. Así como el estudio de la metodología, herramientas y lenguaje de programación a usar para el desarrollo de la solución.

1.2 OPC

1.2.1 Surgimiento de OPC

En los sistemas de automatización se encuentran múltiples elementos de control y monitorización, cada uno con su protocolo de comunicaciones específico (Modbus, BISAP-Serial, Ethernet, etc.) y con características propias. Para el acceso a cada elemento de control se necesita utilizar un programa distinto por cada protocolo, acompañado de otro programa para la visualización de los datos recolectados del dispositivo, los distintos programas lo provee el propio fabricante. (3)

Esto trae consigo que los desarrolladores de sistemas de automatización, se vean obligado a utilizar el software y hardware de un determinado proveedor, en la mayoría de los casos. Las aplicaciones de software desarrolladas por un fabricante no son compatibles con el hardware de otro, aunque tengan el mismo protocolo, ya que no implementan interfaces comunes para poder lograr la compatibilidad entre ambos.

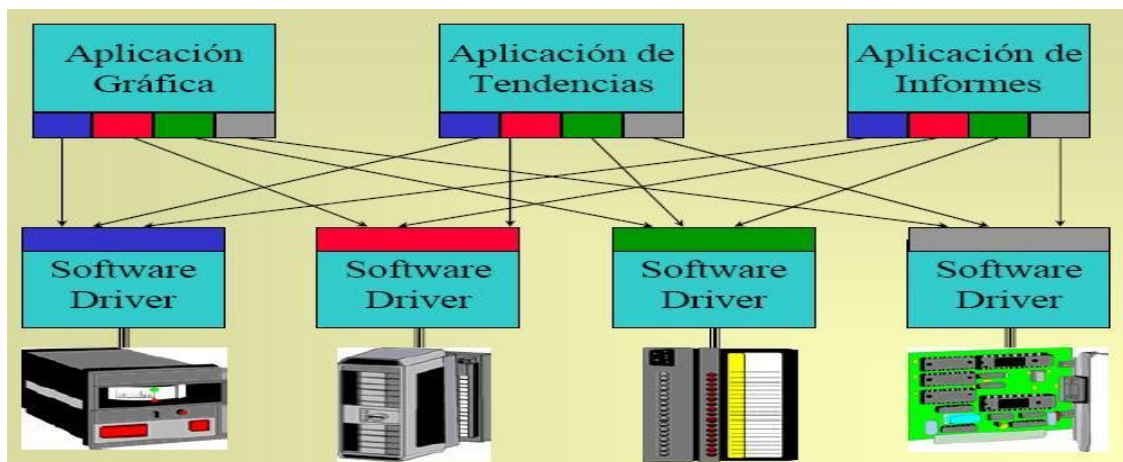


Ilustración 1: Red donde no se cumple con el estándar OPC.

OPC surgió con la idea de suprimir el problema anteriormente descrito creando un estándar orientado al modo de intercambio de datos, independientemente de la tecnología utilizada para hacerlo. Cualquiera que sea la fuente de datos (un PLC, un regulador de temperatura, etc.) el formato de presentación y acceso a los datos será fijo. De esta manera, permitirá intercambiar datos con cualquier equipo que cumpla el estándar OPC y además una reducción de costes considerable, pues cada manejador se deberá escribir una sola vez. (3)

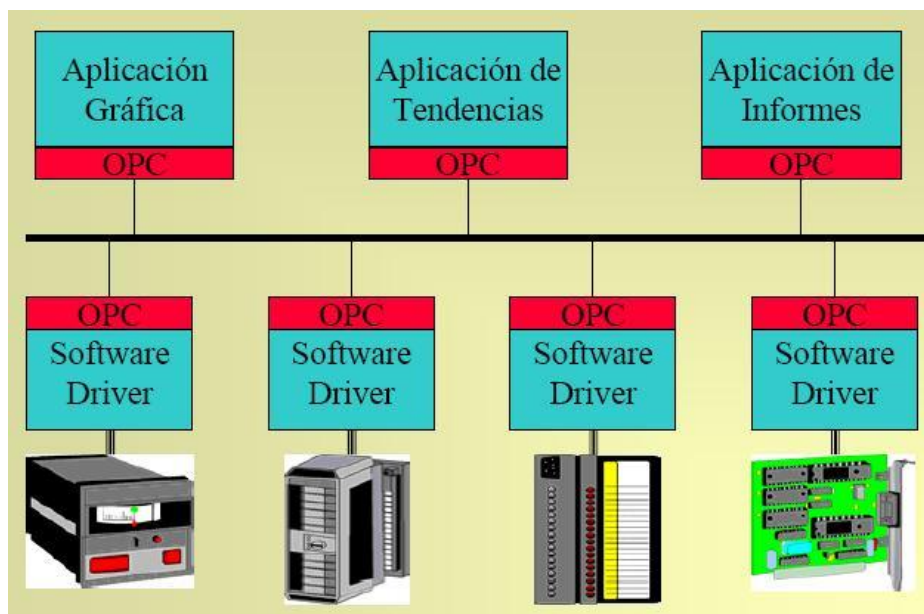


Ilustración 2: Solución con OPC

La Asociación Internacional de Estándares, siglas en inglés ISA, hace su primer anuncio de OPC en 1995. Las especificaciones OPC se mantienen a través de la Fundación OPC. Actualmente la fundación cuenta con más de 400 miembros de todo el mundo, incluidos casi todos los principales proveedores mundiales de sistemas de control de procesos e instrumentación.

1.2.2 Definición

OPC es un conjunto de especificaciones para la interconexión sobre el sistema operativo Microsoft Windows y hardware de control de procesos. Las especificaciones están basadas en COM/DCOM, tecnologías desarrolladas por Microsoft para su sistema operativo, las mismas definen un conjunto estándar de objetos, interfaces y métodos para su uso en el control de procesos y aplicaciones industriales de automatización, para facilitar la interoperabilidad.

1.2.3 Arquitectura.

OPC emplea la arquitectura cliente-servidor, lo que hace posible la comunicación entre elementos que cumplan con el estándar. Permite además el acceso a los datos de forma local o remota en tiempo real.

Aunque OPC está diseñado principalmente para acceder a datos desde un servidor en red, las interfaces OPC pueden utilizarse en muchos lugares dentro de una aplicación. En el nivel más bajo pueden obtener los datos en bruto de los dispositivos físicos para sistemas SCADA o Sistemas de Control Distribuido (DCS), o desde sistemas SCADA o DCS para la aplicación. La arquitectura y diseño hace que sea posible construir un servidor OPC que permite a una aplicación cliente acceder a los datos de muchos servidores OPC, proporcionados por diferentes proveedores de OPC que se ejecutan en diferentes nodos, a través de un único objeto. (4)

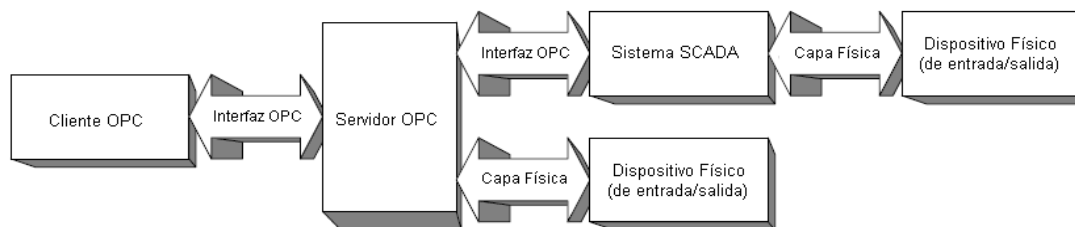


Ilustración 3: Relación Cliente/Servidor de OPC

Cliente OPC: Aplicación que solo utiliza los datos brindados por el servidor OPC. Cualquier cliente OPC puede conectarse con cualquier servidor OPC sin importar el tipo de dispositivo que recoge los datos.

Servidor OPC: Es una aplicación que recolecta los datos de los dispositivos que se encuentran configurados en campo, permite el acceso libre a los datos recolectados desde otras aplicaciones que los soliciten (Clientes OPC).

1.2.4 Especificaciones

OPC es un estándar dividido en varias especificaciones, las más comunes son:

- OPC DA (Data Access, Acceso a Datos).
- OPC HDA (Historical Data Access, Acceso a Datos Históricos).

- OPC A&E (Alarm and Events, Alarmas y Eventos).
- OPC DX (Data Exchange, Intercambio de datos).
- OPC XML-DA (Extensible Markup Language-Data Access, Lenguaje de Mercado Extensible para Acceso a Datos).
- OPC Security (Seguridad).

1. OPC DA

OPC DA es una especificación para dos conjuntos de interfaces, las interfaces OPC Personalizadas y las interfaces OPC de Automatización. Una interfaz de Automatización revisada es proporcionada con la versión 2.0 de la especificación OPC.

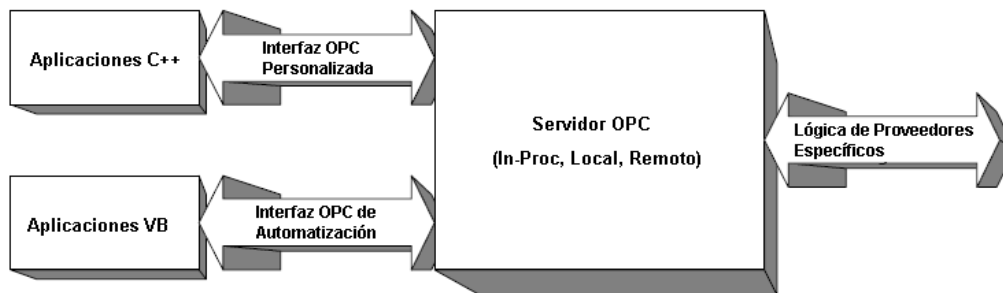


Ilustración 4: Interfaces de OPC

La especificación define las interfaces COM, no su implementación. Se especifica el comportamiento que se espera que las interfaces proporcionen a las aplicaciones cliente que las utilizan.

Hay varias consideraciones únicas en la implementación de un servidor OPC. El principal problema es la frecuencia de transferencia de datos por vías de comunicación no compatible de dispositivos físicos. Por lo tanto, esperamos que el Servidor OPC DA sea una aplicación local o remota que incluye código que se encarga de la recopilación de datos eficiente desde un dispositivo físico.

Una aplicación cliente OPC se comunica con un servidor OPC a través de las interfaces OPC personalizadas y de automatización. Los servidores OPC deben implementar la interfaz personalizada y, opcionalmente, puede implementar la interfaz de automatización. (4)

Los servidores OPC DA permiten a los clientes recuperar la información a través los siguientes objetos: el Servidor, Grupo y Elemento.

- El objeto Servidor OPC mantiene información sobre el servidor y actúa como un contenedor de objetos grupos OPC.
- El objeto Grupo OPC mantiene información sobre sí mismo y provee el mecanismo para contener y organizar lógicamente elementos OPC.
- El objeto Elemento OPC representa la conexión de la fuente de dato con el servidor.

Esta especificación define dos interfaces para el manejo de la escritura y lectura, las cuales son:

- Interfaz síncrona: El cliente puede realizar una lectura de forma síncrona. Esto es apropiado para cuando el cliente es bastante simple y está leyendo cantidades relativamente pequeñas.
- Interfaz asíncrona: Es cuando el cliente se suscribe a los datos provenientes de la cache usando las interfaces AdviseSink o IOPCDataCallback, es más complejo, pero más eficiente. El acceso asíncrono es recomendado porque este minimiza el uso de la memoria del cómputo y los recursos de red. (4)

OPC DA brinda el último dato recolectado a muchos clientes OPC DA según la frecuencia de muestreo que este impone sobre los grupos configurados.

2. OPC HDA

Dedicado al acceso a datos históricos exclusivamente. Los datos históricos son utilizados para el análisis de los datos a través de gráficas de tendencias, análisis estadísticos y mantenimiento predictivo.

3. OPC A&E

Esta especificación describe las interfaces y los objetos que se implementan en los servidores OPC A&E. Estos servidores proporcionan mecanismos que permiten a los clientes visualizar las notificaciones de alarma y eventos (en contraste con el flujo continuo de datos) que fueron

activadas por el servidor OPC A&E. Estos incluyen alarmas de proceso, las acciones del operador, mensajes informativos y de seguimiento.

4. OPC DX

Esta especificación es utilizada para el intercambio de datos entre servidores OPC. Regula las comunicaciones servidor a servidor a través de las redes de comunicación Ethernet. Permite servicios de gestión y configuración remotos.

5. OPC XML DA

El objetivo es desarrollar flexibles y consistentes regulaciones para hacer disponible datos tecnológicos vía OPC usando XML en la Internet/Intranet.

XML, el lenguaje extensible de marcado, y lenguaje de esquema basado en XML ofrece un medio más para describir e intercambiar información estructurada entre aplicaciones. XML es una tecnología que está disponible más fácilmente en una amplia gama de plataformas. OPC XML-Data Access (OPC XML-DA) es la adopción del conjunto de tecnologías XML para facilitar el intercambio de datos a través de Internet, y hacia el dominio empresarial.(5)

6. OPC de Seguridad

Todos los servidores OPC proporcionan información que es valiosa para la empresa y si no se actualiza, podría repercutir de manera importante a los procesos de la planta. La especificación Seguridad de OPC define cómo controlar el acceso del cliente a estos servidores con el fin de proteger esta información sensible y para protegerla contra la modificación no autorizada de los parámetros del proceso. (1)

1.2.4 Desarrollo de OPC en el mundo

En el mundo existen varias soluciones OPC entre las que se encuentran:

- **KepServerEx:** Es uno de los servidores OPC desarrollado por la compañía Kepware, empresa que tiene buen prestigio a nivel mundial en software de comunicaciones para la automatización y ofrece una experiencia única, tanto en OPC como en dispositivo de comunicaciones embebidos. (6)

- **OPC DataHub:** Software que permite recolectar datos de diferentes fuentes y los pone a disposición a cliente para su uso. (7)
- **MatrikonOPC:** Es el mayor proveedor mundial de productos de conectividad OPC, con una colección de más de 500 interfaces como Servidores OPC, Clientes OPC y OPC HDA. Con más de 30.000 usuarios y más de 100.000 instalaciones en todo el mundo, MatrikonOPC ofrece conectividad a todos los sistemas de control más importantes y de aplicación en el mercado. (8)

Las empresas encontradas que desarrollan soluciones OPC orientan sus productos en dos vertientes, una primera al desarrollo de aplicaciones específicas orientadas al usuario final, y otra al desarrollo de herramientas o SDK (del inglés System Development Kit) con el objetivo de proporcionar una base de desarrollo a quien desee habilitar aplicaciones con el estándar OPC.

Los proveedores de servidores OPC como aplicaciones finales generalmente implementan solo un protocolo, por lo que para cubrir una amplia gama de protocolos de deben adquirir esta misma cantidad de servidores por parte de los clientes. Otros proveedores de instrumentación de campo solo desarrollan servidores OPC para su hardware específico y en muchas ocasiones no se conocen los protocolos que implementan. Los que desarrollan SDKs solo dan versiones beta, o de prueba, y en muchas veces no dan acceso al código fuente, sino solamente bibliotecas de vinculación estática o dinámica con la implementación contenida. El principal factor de todos estos sistemas y soluciones es su condición de propietario por lo que tiene un propósito comercial, lo que induce la disposición de recursos financieros para su adquisición.

1.2.5 Próxima generación de OPC

La nueva generación de OPC conocida como OPC-UA (**OP**en **C**onnectivity **U**nified **A**rchitecture). Esta generación ofrece soluciones a los problemas de la pasada generación y añade nuevas características significativas. OPC UA se basa en una tecnología conocida como "Arquitectura Orientada a Servicios" (SOA). SOA implica la creación de programas (servicios) que realizan una función muy específica. Estos servicios están disponibles para cualquier aplicación de software remoto que tiene tanto la autoridad como la necesidad para el servicio. Dado que los sistemas de automatización requieren alta confiabilidad y rendimiento, la interfaz debe funcionar de una manera muy fiable. La seguridad es una de las principales preocupaciones, por lo que la interfaz debe proporcionar apoyo para el acceso seguro e

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA "Guardián del ALBA"

inmunidad a los ataques maliciosos que pueden comprometer un sistema de automatización. Además la gestión del acceso seguro es a través de certificados de seguridad. (9)

OPC UA utiliza los conceptos de las especificaciones de OPC anteriores y las amplía con nuevos datos complejos y la capacidad de mantener el contexto. Está diseñado para que pueda ser desarrollado en cualquier sistema operativo.

1.3 Tecnologías, metodología, lenguajes y herramientas de desarrollo.

Se exige analizar las herramientas y tecnologías que se correspondan con los requisitos establecidos que se describen en capítulo siguiente.

1.3.1 Metodología RUP (Rational Unified Process)

RUP es un proceso de desarrollo de Software que define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo. El proceso unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. (10)

El Proceso Unificado utiliza el UML (Lenguaje Unificado de Modelado) para reparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado – sus desarrollos fueron paralelos--. (10)RUP Posee tres características fundamentales que influyeron en la selección de esta robusta metodología:

- ✓ **Dirigido por casos de uso:** Los Casos de Uso (CU) reflejan lo que los usuarios futuros necesitan y desean, constituyen la guía fundamental establecida para las actividades a realizar durante el proceso de desarrollo del sistema.
- ✓ **Centrado en la Arquitectura :** La arquitectura muestra la visión común del sistema completo en el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- ✓ **Iterativo e incremental:** RUP divide el proyecto en fases de desarrollo, propone además que cada una de ellas se desarrolle en iteraciones.

1.3.2 Herramientas de desarrollo

Herramientas de desarrollo a usar:

- Visual Paradigm.
- Microsoft Visual C++.
- Framework Qt.

1.3.2.1 Visual Paradigm para UML

Es una potente herramienta CASE (Computer-Aided Software Engineering, Ingeniería de Software Asistida por Computadora) que permite la generación de código e ingeniería tanto directa como inversa, utiliza el lenguaje de modelado UML para visualizar y diseñar los elementos de software. Posibilita el modelado de caso de usos incluyendo todas las funciones del Diagrama de Caso de Uso, editor de flujo de eventos y la generación de un diagrama de actividad. Los desarrolladores pueden diseñar la documentación del sistema con una plantilla de diseño. El analista de sistemas pueden estimar las consecuencias de los cambios con los diagramas de análisis de impacto, tales como la matriz y el diagrama de análisis.

Es una herramienta colaborativa lo que facilita la edición de un proyecto por varios usuarios, al mismo tiempo puede generar la documentación de proyecto automáticamente en varios formatos como PDF, HTML y formatos de Microsoft Word y puede integrarse con el control de versiones. Esta herramienta soporta el ciclo de vida completo de desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegues.

1.3.2.2 Microsoft Visual C++ 2005

Microsoft Visual C++ 2005 proporciona un entorno de desarrollo eficaz y flexible para crear aplicaciones basadas en Microsoft Windows y en Microsoft .NET. Se puede utilizar como un sistema de desarrollo integrado o como un conjunto de herramientas individuales.

C++ es el lenguaje de alto nivel más popular del mundo, y Visual C++ ofrece a los desarrolladores una herramienta universal con la que generar software. (11)

Se utiliza Microsoft Visual C++ como IDE por las dependencias del servidor OPC con la tecnología .NET.

1.3.2.3 Framework Qt.

Qt es un conjunto de bibliotecas multipropósito que permiten el desarrollo de interfaces gráficas. Está desarrollado en C++ y se puede integrar con distintos IDE existentes, como es el caso de Eclipse, Microsoft Visual Studio, Code::Blocks entre otros. Está disponible para sistemas operativos como Microsoft Windows, Linux, Mac OS X, bajo licencias GPL v1, GPL v2, GPL v3 y LGPL.

1.3.3 Lenguajes

1.3.3.1 Lenguaje de Modelado UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de sistemas informáticos, así como para el modelado de negocios y otros sistemas no software. El UML representa una colección de las mejores prácticas de ingeniería que han demostrado su eficacia en el modelado de sistemas grandes y complejos. (12)

Características de UML:

- Proporciona un mecanismo de extensibilidad, los cuales permiten a sus usuarios refinar su sintaxis y su semántica. Puede además, ajustarse a un sistema, proyecto o proceso de desarrollo específico si es necesario.
- UML no es un lenguaje de programación.
- UML captura información sobre una estructura estática y comportamiento dinámico de un sistema

1.3.3.2 Lenguaje de programación: C++

En la actualidad, C++ es un lenguaje versátil, potente y general. Mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet (12).

Las características más notables de C++ son:

- ✓ Programación orientada a objetos, permitiendo mayor reutilización de código en un modo más lógico y productivo.
- ✓ Portabilidad, prácticamente se puede compilar el mismo código en casi cualquier tipo de ordenador y sistema operativo sin realizar ningún cambio.
- ✓ Programación modular. El cuerpo de una aplicación en C + + puede estar compuesto por varios archivos de código fuente que se compilan por separado y luego unidas entre sí.
- ✓ C Compatibilidad. C++ es compatible con el lenguaje C. Cualquier código escrito en C puede ser incluido en un programa de C ++ sin realizar ningún cambio.
- ✓ Velocidad. El código resultante de una compilación de C + + es muy eficiente, debido a su efecto dualidad tanto a alto nivel como a bajo nivel de idiomas y al reducido tamaño del propio lenguaje.

1.4 Conclusiones parciales.

En este capítulo se expusieron las características más importantes del estándar OPC, se detallaron además las principales especificaciones que brinda la Fundación OPC para la automatización industrial, haciendo énfasis en la especificación de Acceso a Datos (OPC DA) como referencia de desarrollo. Se presentaron características de los principales proveedores de servidores OPC en el mundo, de los que no se pudo obtener aplicación alguna debido a que todos están bajo licencias comerciales, manteniendo el *Toolkit* de desarrollo OPCWorkshop, utilizado en la versión actual del Servidor OPC de Acceso a Datos del SCADA “Guardián del ALBA”. Se justificaron las herramientas, tecnologías y metodologías utilizadas para el diseño e implementación de la solución.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se exponen las principales características de los componentes de software existentes que servirán para conformar la solución. Se definen los principales conceptos que conforman el modelo de dominio, así como los requisitos funcionales y no funcionales que regirán el desarrollo de la solución propuesta. Partiendo de esto se determinarán los casos de uso a realizar. Además se describen los procesos de las principales funcionalidades del sistema.

2.2 Soluciones Técnicas.

La solución del presente trabajo consta de dos partes fundamentales, una es la integración de dos componentes desarrollados en el CEDIN para conformar un servidor OPC DA, y la segunda representa la interfaz gráfica para la configuración de dicho servidor.

Componente OPCWorkshop:

Es un toolkit de desarrollo de servidores OPC para Acceso a Datos basado en la versión 2.05a. Ha servido como implementación base de servidores OPC sobre la cual se han desarrollado trabajos de gran impacto. El componente OPCWorkshop cuenta con las siguientes características:

1. Implementa las interfaces COM requeridas por la especificación DA 2.05a, así como algunas de las interfaces opcionales como es el caso de IOPCBrowseServerAddressSpace, muy útil por los clientes para la navegación por el espacio de direcciones del servidor.
2. Posibilita a los clientes las operaciones de escritura y lectura de variables tanto síncronas como asíncronas.
3. . Representa el acceso al espacio de direcciones de la siguiente forma:
 - **Jerárquico:** Se representa en forma de árbol, donde los elementos o variables son las hojas y los grupos, nodos o ramas. Además es capaz de posibilitar la navegación de forma plana.
 - **Plano:** Consiste en devolver el identificador de todas las variables que se encuentran registradas en el servidor.

4. No cuenta con un mecanismo de manejadores y adquisición de datos desde los dispositivos de campo o fuente de datos.

Componente Recolector:

Componente para la adquisición de datos desde los dispositivos de campo a través de la red. Utiliza los manejadores de dispositivos desarrollados por la Línea de Desarrollo de Drivers, lo que le permite incorporar nuevos protocolos de comunicación paulatinamente, ya que estos funcionan como extensiones al implementar la Interfaz Genérica para los manejadores.

Características:

1. Permite la configuración de los canales de comunicación, manejadores, dispositivos y variables, tanto al iniciar el servicio como durante la ejecución.
2. Realiza operaciones de lectura y escritura de variables en los dispositivos de campo.
3. Implementa un *pool* de hilos para el manejo concurrente de las operaciones.
4. Permite organizar las operaciones por prioridad.

La integración del componente servidor OPC DA y de adquisición será mediante una aplicación principal sin interfaz de usuario, la que contiene las rutinas necesarias para la configuración de los componentes y comunicación entre ellos.

Una vez integrado los dos componentes antes descritos, el módulo de recolección se encarga de recolectar desde los dispositivos de campo y actualizar la cache del componente OPC DA, este último brinda el espacio de direcciones y los valores actualizados a los clientes. Para lograr todo este mecanismo cada componente debe conocer que variables se van a recolectar, a que dispositivo pertenecen, y que espacio de direcciones se mostrará; todo esto se conoce a través de un archivo de configuración en formato XML que debe ser generado por el Configurador que formará parte de este trabajo.

2.3 Modelo del dominio

Un modelo del dominio captura los tipos más importantes de los objetos en contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema (10).

En la presente sesión se describe los conceptos más importantes y sus relaciones, para lograr un mejor entendimiento de la solución.

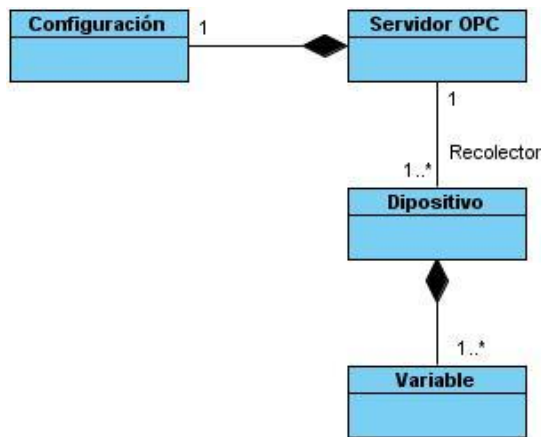


Ilustración 5: Modelo de Dominio.

2.3.1 Clases conceptuales o Glosario de términos del dominio.

Configuración: Contiene la estructura y organización necesaria para el funcionamiento del Servidor OPC DA.

Variable: Información que representa un dato obtenido desde un dispositivo de campo u otra fuente de datos.

Dispositivo: Aparato encargado de realizar una acción antes prevista.

Servidor OPC: Aplicación de software que implementa el estándar OPC de Acceso a Datos del lado del servidor.

2.3.2 Requisitos funcionales

Los requisitos funcionales son capacidades o funciones que el sistema debe cumplir, son las funcionalidades que el sistema debe cumplir. Logrando el cumplimiento de dichos requisitos se logra la satisfacción del cliente.

El sistema debe ser capaz de:

RF 1 Gestionar Canal.

Este requisito expresa la necesidad de gestionar canales de comunicación, los cuales pueden ser exclusivos o compartidos.

RF 1.1 Crear Canal.

RF 1.2 Modificar Canal.

RF 1.3 Eliminar Canal.

RF 2 Gestionar Subcanal.

Este requisito expresa la necesidad de gestionar los subcanales, los cuales contienen información del manejador.

RF 2.1 Crear Subcanal.

RF 2.2 Modificar Subcanal.

RF 2.3 Eliminar Subcanal.

RF 3 Gestionar Dispositivos.

Este requisito expresa la necesidad de gestionar dispositivos lo cuales contienen parámetros de conexión con cada dispositivo de campo.

RF 3.1 Crear Dispositivos.

RF 3.2 Modificar Dispositivos.

RF 3.3 Eliminar Dispositivos.

RF 4 Gestionar Variables.

En este requisito se expone la necesidad de crear, modificar y eliminar variables asociadas a un dispositivo.

RF 4.1 Crear Variables del dispositivo.

RF 4.2 Modificar Variables del dispositivo.

RF 4.3 Eliminar Variables del dispositivo.

RF 5 Gestionar Grupo.

Este requisito expone la necesidad de gestionar grupo para organizar las variables dentro de cada dispositivo, en el mismo puede contener varios grupos y a su vez variables

RF 5.1 Crear Grupo.

RF 5.2 Modificar Grupo.

RF 5.3 Eliminar Grupo.

RF 6 Guardar en formato XML la configuración editada.

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA "Guardián del ALBA"

El Configurador debe ser capaz de guardar en un archivo de formato XML la configuración previamente editada.

RF 7 Cargar la configuración del lado del configurador.

El configurador debe ser capaz de cargar una configuración guardada.

RF 8 Cargar la configuración del lado del servidor.

El servidor debe ser capaz a la hora de iniciar cargar la configuración desde un fichero XML.

2.3.3 Requisitos No Funcionales.

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad del mantenimiento, extensibilidad y fiabilidad (10).

A continuación se hace mención de cada uno de los requisitos no funcionales del sistema, se encuentran separados por categoría.

RNF de Software.

- Sistema Operativo Windows 2000 (Service Pack 4 requerido) o una versión superior.

RNF de Hardware.

- Mínima memoria RAM 512 MB (1G Recomendado).
- Microprocesador a 1.6 GHZ o superior.

RNF Restricciones en el diseño e implementación.

- Uso del framework Qt para el desarrollo de la interfaz visual.
- Emplear el paradigma programación orientado a objetos.
- C++ como lenguaje de programación.

RNF de Seguridad.

- Disponibilidad: Los datos recolectados por el módulo de recolección deberán estar disponibles a los clientes OPC con una estampa de tiempo válida.

RNF de Usabilidad.

- El configurador debe permitir la fácil configuración de canales, subcanales, dispositivos y variables.
- Las operaciones deben ser lo más intuitivas posibles a la vista del cliente.

RNF de Disponibilidad.

- El servidor debe mantenerse en funcionamiento las 24 horas del día y los 7 días de la semana.

RNF Soporte.

- Una vez terminada la aplicación debe ser de fácil instalación y puesta en ejecución.

2.3.4 Descripción del sistema propuesto.

2.3.4.1 Diagrama de casos de uso.

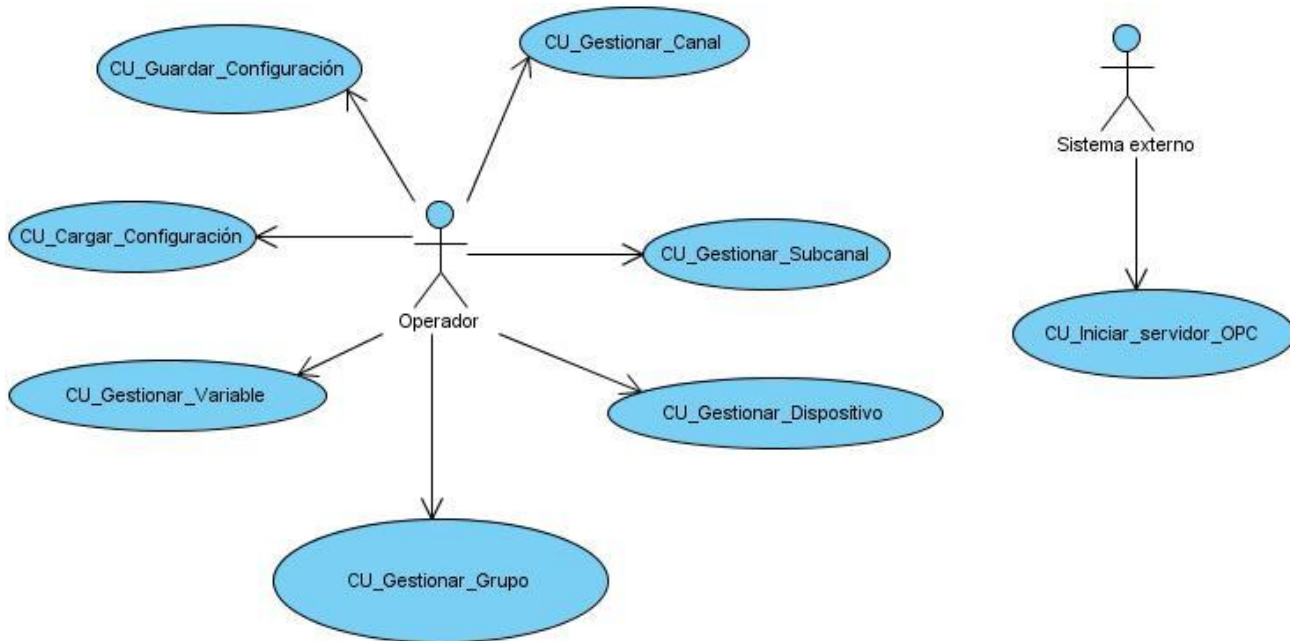


Ilustración 6: Diagrama de casos de uso del sistema

2.3.4.2 Descripción de los Actores.

Actores	Descripción
Operador	Es la persona encargada de crear la configuración.
Sistema Externo	Programa que inicia el servidor. Ej. Cliente OPC

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

Tabla 1: Descripción de actores

2.3.4.3 Casos de Usos (CU) del Sistema.

Los casos de usos del sistema se clasifican como críticos, secundarios, auxiliares u opcionales en dependencia de la importancia que tienen dentro del sistema. A continuación se presenta una tabla con esta clasificación.

Críticos	Secundarios	Auxiliares	Opcionales
CU Gestionar Canal CU Gestionar Subcanal CU Gestionar Dispositivo CU Gestionar Variable CU Gestionar Grupo Guardar Configuración Iniciar servidor OPC	Cargar configuración en el configurador	No se aplica	No se aplica

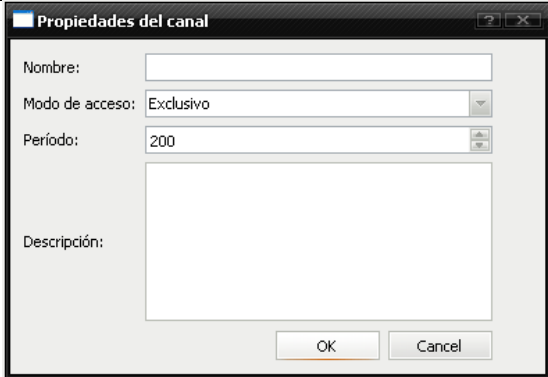
Tabla 2: Clasificación de los Casos de uso del sistema


2.3.4.4 Descripción textual de los casos de uso arquitectónicamente significativos.

A continuación se describirán los casos de usos significativos, los otros se describen en los anexos.

Nombre del caso de uso:	CU Gestionar Canal
Actor	Operador
Propósito	Permite crear, modificar y eliminar los datos de un canal de comunicación.
Resumen	Se inicia cuando el operador decide agregar, modificar, o eliminar un canal.
Referencia	RF 1, 1.1, 1.2, 1.3
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	Se debe especificar el camino hacia los manejadores del SCADA.
Post-condiciones	Quedan insertados, modificados o eliminados los datos del canal.
Curso Normal de los eventos	

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA "Guardián del ALBA"

Acción del actor	Respuesta del sistema
1. El actor necesita insertar, modificar o eliminar un canal.	1.1 El sistema muestra las siguientes opciones: <ul style="list-style-type: none"> a) Si decide insertar un canal, ir a la sección “Insertar Canal”. b) Si decide modificar un canal, ir a la sección “Modificar Canal”. c) Si decide eliminar un canal, ir a la sección “Eliminar canal”.
Sección “Insertar Canal”	
2. El operador introduce los datos del canal (Nombre, Modo de Acceso, Período y Descripción) y presiona el botón aceptar.	2.1 El sistema verifica que todos los campos estén llenos. 2.2 El sistema verifica que los datos de entrada correspondan con los tipos de datos que representan. 2.3 El sistema inserta los datos del canal.
Curso alternativo de los eventos:	
	2.1 El sistema muestra el mensaje de error: “No se pueden dejar campos vacíos”. 2.2 El sistema muestra el mensaje de alerta: “Dato inválido”.
Prototipo de interfaz de usuario	
	
Sección “Modificar Canal”.	
	1.2 El sistema verifica si el canal contiene subcanales.

	1.3 El sistema muestra una interfaz de usuario con los datos insertados del canal anteriormente.
2. El operador introduce los nuevos datos a modificar del canal y presiona el botón aceptar.	2.1 El sistema verifica que los campos estén llenos. 2.2 El sistema valida que los datos entrados correspondan con el tipo de datos que representa. 2.3 El sistema almacena los cambios realizados correctamente.
Curso Alternativo de los eventos:	
	1.2 En caso de contener subcanales el canal a modificar el sistema invalida el campo "Modo de acceso".
	2.1 El sistema muestra el mensaje de error: "No pueden existir campos vacíos". 2.2 El sistema muestra el mensaje de alerta: "Datos inválidos".
Prototipo de interfaz de usuario	
	
Sección "Eliminar Canal"	
	1.2 El sistema verifica si el canal seleccionado contiene subcanal. 1.3 El sistema elimina el canal seleccionado por el operador.
Curso Alternativo de los eventos:	

1.2 El sistema despliega un mensaje de error “El canal contiene subcanales asociados”.

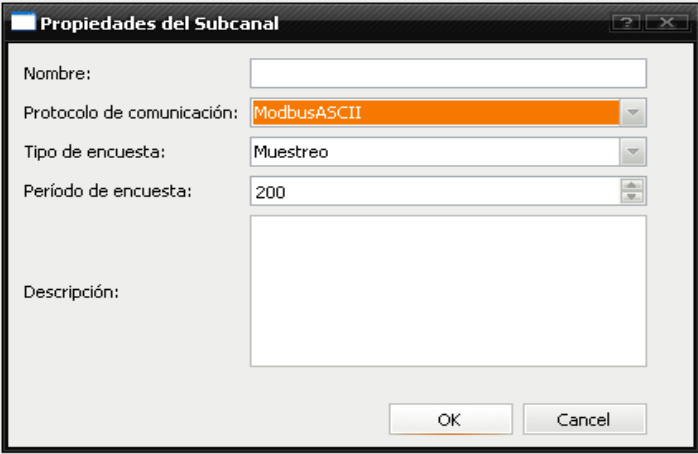
Prototipo de interfaz de usuario



Tabla 3: Descripción del CU Gestionar Canal

Nombre del caso de uso:	CU Gestionar Subcanal	
Actor	Operador	
Propósito	Permite crear, modificar y eliminar los datos de un Subcanal de comunicación.	
Resumen	Se inicia cuando el operador decide agregar, modificar, o eliminar un Subcanal.	
Referencia	RF 2, 2.1, 2.2, 2.3	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir un canal.	
Post-condiciones	Quedan insertados, modificados o eliminados los datos del Subcanal.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El actor necesita insertar, modificar o eliminar un Subcanal.	1.1 El sistema muestra las siguientes opciones: a) Si decide insertar un subcanal, ir a la sección “Insertar Subcanal”. b) Si decide modificar un subcanal, ir a la sección “Modificar Subcanal”.	

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

	c) Si decide eliminar un subcanal, ir a la sección “Eliminar Subcanal”.
Sección “Insertar Subcanal”	
2. El operador introduce los datos del Subcanal (Nombre, Protocolo de comunicación, Tipo de encuesta, Período de encuesta y Descripción).	<p>2.1 El sistema verifica que todos los campos estén llenos.</p> <p>2.2 El sistema verifica que los datos de entrada correspondan con los tipos de datos que representan.</p> <p>2.3 El sistema inserta los datos del Subcanal.</p>
Curso alternativo de los eventos:	
	<p>2.1 El sistema muestra el mensaje de error: “No se pueden dejar campos vacíos”.</p> <p>2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
Prototipo de interfaz de usuario	
	
Sección “Modificar Subcanal”.	
	<p>1.2 El sistema verifica si el Subcanal contiene dispositivos.</p> <p>1.3 El sistema muestra una interfaz de usuario con los datos insertados del subcanal anteriormente.</p>

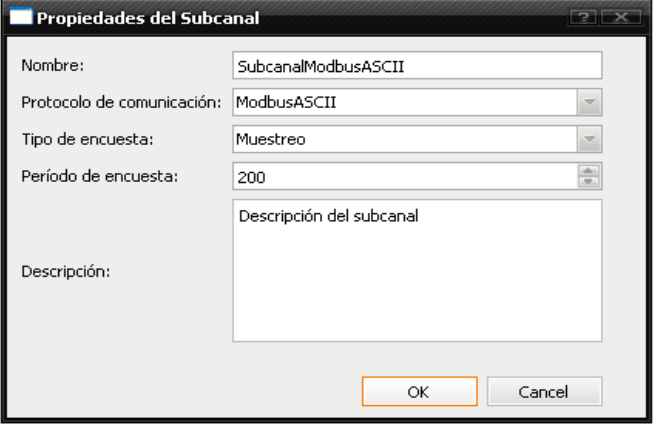
<p>2. El operador introduce los nuevos datos a modificar del Subcanal.</p>	<p>2.1 El sistema verifica que los campos estén llenos.</p> <p>2.2 El sistema valida que los datos entrados correspondan con el tipo de datos que representa.</p> <p>2.3 El sistema almacena los cambios realizados correctamente.</p>
--	--

Curso Alternativo de los eventos:

	<p>1.2 En caso de contener dispositivos el subcanal a modificar, el sistema invalida el campo “Protocolo de comunicación”.</p>
--	--

	<p>2.1 El sistema muestra el mensaje de error: “No pueden existir campos vacíos”.</p> <p>2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
--	--

Prototipo de interfaz de usuario



Sección “Eliminar Subcanal”

	<p>1.2 El sistema verifica si el subcanal seleccionado contiene dispositivos.</p> <p>1.3 El sistema elimina el subcanal seleccionado por el operador.</p>
--	---

Curso Alternativo de los eventos:


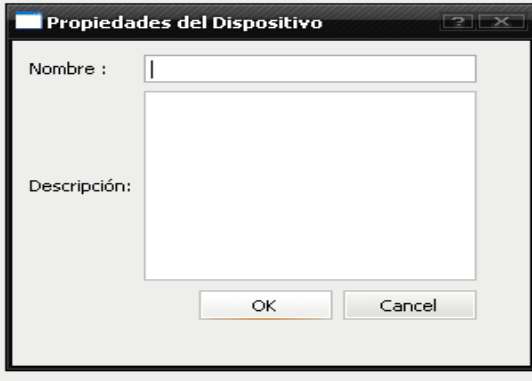
	1.2 El sistema despliega un mensaje de error “El subcanal contiene dispositivos asociados”.	
Prototipo de interfaz de usuario		
		
Relaciones	CU Incluidos	No tiene
	CU Extendidos	No tiene

Tabla 4: Descripción del CU Gestionar Subcanal

Nombre del caso de uso:	CU Gestionar Dispositivo	
Actor	Operador	
Propósito	Permite crear, modificar y eliminar los datos de un Dispositivo.	
Resumen	Se inicia cuando el operador decide agregar, modificar, o eliminar un Dispositivo.	
Referencia	RF 3, 3.1, 3.2, 3.3	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir un subcanal.	
Post-condiciones	Quedan insertados, modificados o eliminados los datos del Dispositivo.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El actor necesita insertar, modificar	1.1 El sistema muestra las siguientes opciones:	

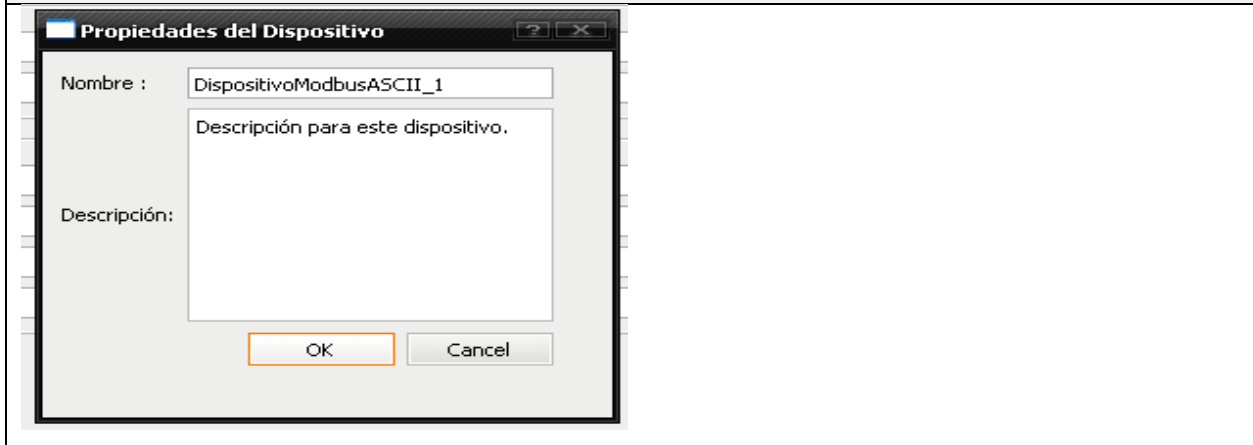
o eliminar un Dispositivo.	<p>a) Si decide insertar un dispositivo, ir a la sección “Insertar Dispositivo”.</p> <p>b) Si decide modificar un dispositivo, ir a la sección “Modificar Dispositivo”.</p> <p>c) Si decide eliminar un dispositivo, ir a la sección “Eliminar Dispositivo”.</p>
Sección “Insertar Dispositivo”	
2. El operador introduce los datos del Dispositivo (Nombre y Descripción).	<p>2.1 El sistema verifica que todos los campos estén llenos.</p> <p>2.2 El sistema verifica que los datos de entrada correspondan con los tipos de datos que representan.</p> <p>2.3 El sistema inserta los datos del Dispositivo.</p>
Curso alternativo de los eventos:	
	<p>2.1 El sistema muestra un mensaje de error “No se pueden dejar campos vacíos”.</p> <p>2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
Prototipo de interfaz de usuario	
	
Sección “Modificar Dispositivo”.	
	1.2 El sistema muestra una interfaz de usuario con los datos insertados del dispositivo anteriormente.

<p>2. El operador introduce los nuevos datos a modificar del Dispositivo.</p>	<p>2.1 El sistema verifica que los campos estén llenos.</p> <p>2.2 El sistema valida que los datos entrados correspondan con el tipo de datos que representa.</p> <p>2.3 El sistema almacena los cambios realizados correctamente.</p>
---	--

.Curso Alternativo de los eventos:

	<p>2.1 El sistema muestra un mensaje de error “No pueden existir campos vacíos”.</p> <p>2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
--	---

Prototipo de interfaz de usuario



Sección “Eliminar Dispositivo”

	<p>1.2 El sistema verifica si el dispositivo seleccionado contiene variables asociadas.</p> <p>1.3 El sistema elimina el dispositivo seleccionado por el operador.</p>
--	--

Curso Alternativo de los eventos:

	<p>1.1 El sistema despliega un mensaje de error “El dispositivo contiene dispositivos asociados”.</p>
--	---

Prototipo de interfaz de usuario

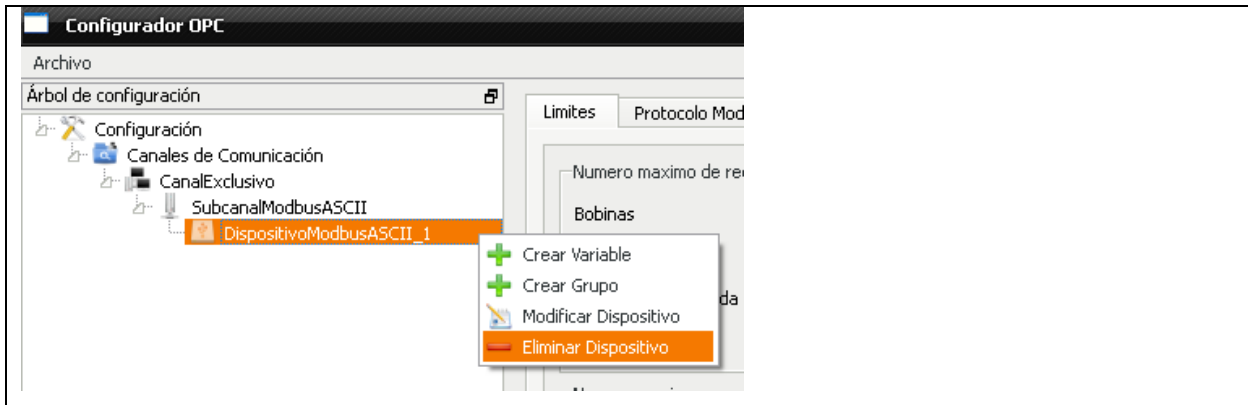


Tabla 5: Descripción del CU Gestionar Dispositivo

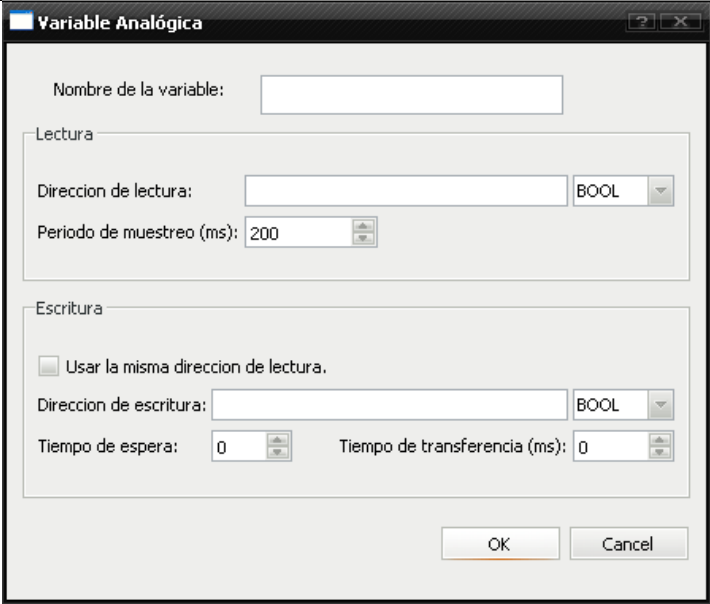
Nombre del caso de uso:	CU Gestionar Variable	
Actor	Operador	
Propósito	Permite crear, modificar y eliminar los datos de una Variable.	
Resumen	Se inicia cuando el operador decide agregar, modificar, o eliminar una Variable.	
Referencia	RF 4, 4.1, 4.2, 4.3	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir un dispositivo o un grupo.	
Post-condiciones	Quedan insertados, modificados o eliminados los datos del Variable.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El actor necesita insertar, modificar o eliminar una Variable.	1.1 El sistema muestra las siguientes opciones: <ul style="list-style-type: none"> a) Si decide insertar una variable, ir a la sección "Insertar Variable". b) Si decide modificar una variable, ir a la sección "Modificar Variable". c) Si decide eliminar una variable, ir a la sección "Eliminar Variable". 	
Sección "Insertar Variable"		

<p>2. El operador introduce los datos de la variable (Nombre, Dirección de lectura, Tipo de dato de lectura, Período de muestreo, Dirección de escritura, Tipo de dato de escritura, Tiempo de espera y Tiempo de transferencia).</p>	<p>2.1 El sistema verifica que todos los campos estén llenos. 2.2 El sistema verifica que los datos de entrada correspondan con los tipos de datos que representan. 3.1 El sistema inserta los datos de la Variable.</p>
---	--

Curso alternativo de los eventos:

	<p>2.1 El sistema muestra un mensaje de error “No se pueden dejar campos vacíos”. 2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
--	--

Prototipo de interfaz de usuario



Sección “Modificar Variable”.

	<p>1.2 El sistema muestra una interfaz de usuario con los datos insertados de la variable anteriormente.</p>
--	--

<p>2. El operador introduce los nuevos datos a modificar de la variable.</p>	<p>2.1 El sistema verifica que los campos estén llenos.</p> <p>2.2 El sistema valida que los datos entrados correspondan con el tipo de datos que representa.</p> <p>2.3 El sistema almacena los cambios realizados correctamente.</p>
--	--

Curso Alternativo de los eventos:

	<p>2.1 El sistema muestra un mensaje de error “No pueden existir campos vacíos”.</p> <p>2.2 El sistema muestra el mensaje de alerta: “Datos inválidos”.</p>
--	---

Prototipo de interfaz de usuario

Sección “Eliminar Variable”

	<p>1.2 El sistema elimina la variable seleccionada por el operador.</p>
--	---

Curso Alternativo de los eventos:

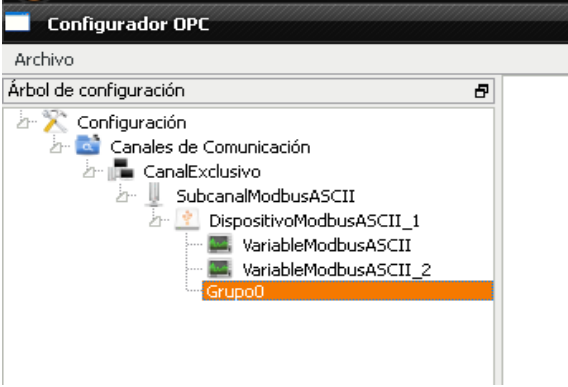
--	--

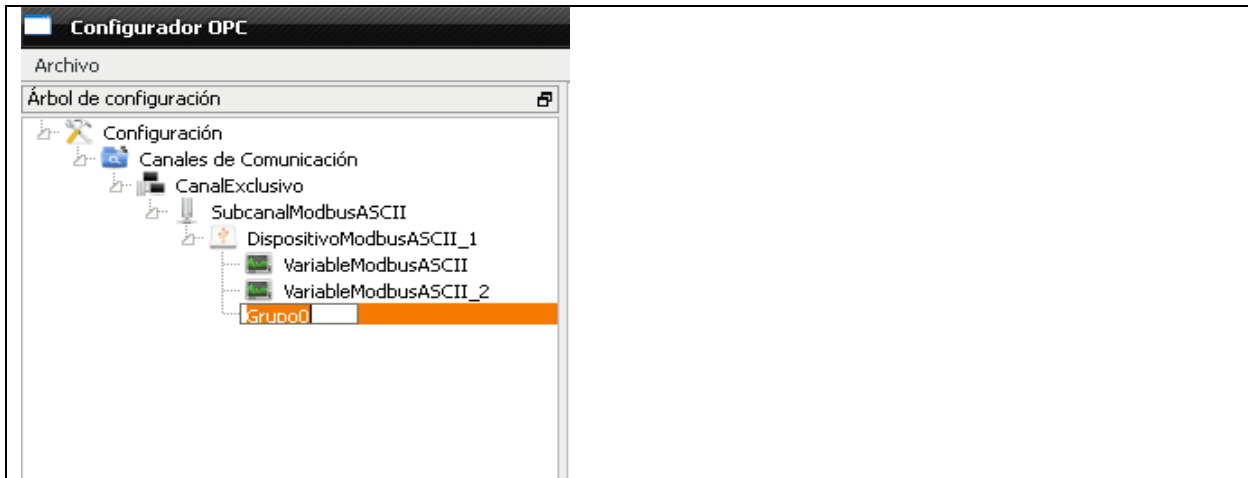
Prototipo de interfaz de usuario



Tabla 6: Descripción del CU Gestionar Variable

Nombre del caso de uso:	CU Gestionar Grupo	
Actor	Operador	
Propósito	Permite crear, modificar y eliminar los datos de un Grupo.	
Resumen	Se inicia cuando el operador decide agregar, modificar, o eliminar un Grupo.	
Referencia	RF 5, 5.1, 5.2, 5.3	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir un dispositivo.	
Post-condiciones	Quedan insertados, modificados o eliminados los datos del Grupo.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El actor necesita insertar, modificar o eliminar una Grupo.	1.1 El sistema muestra las siguientes opciones: <ul style="list-style-type: none"> d) Si decide insertar un grupo, ir a la sección "Insertar Grupo". e) Si decide modificar un grupo, ir a la sección "Modificar Grupo". f) Si decide eliminar un grupo, ir a la sección "Eliminar Grupo". 	
Sección "Insertar Grupo"		

	2.1 El sistema inserta un grupo de nombre autogenerado (Ej. Grupo0).
Curso alternativo de los eventos:	
Prototipo de interfaz de usuario	
	
Sección “Modificar Grupo”.	
1. El operador selecciona el grupo a modificar y presiona dos veces el clic izquierdo.	1.2 El sistema selecciona el nombre del grupo.
2. El operador introduce el nuevo nombre del grupo.	2.1 El sistema verifica que no deje el grupo sin nombre. 2.2 El sistema valida que no exista un grupo con el mismo nombre introducido. 2.3 El sistema almacena el cambio realizado correctamente.
Curso Alternativo de los eventos:	
	2.1 El sistema muestra un mensaje de error “No puede dejar el nombre del grupo vacío”. 2.2 El sistema muestra un mensaje de error “El nombre introducido ya está asignado a otro grupo”.
Prototipo de interfaz de usuario	



Sección “Eliminar Grupo”

- 1.2 El sistema verifica que el grupo no contenga ninguna variable asociada.
- 1.3 El sistema elimina el grupo seleccionado por el operador.

Curso Alternativo de los eventos:

- 1.2 El sistema muestra un mensaje de error “El grupo contiene variables asociadas”.

Prototipo de interfaz de usuario

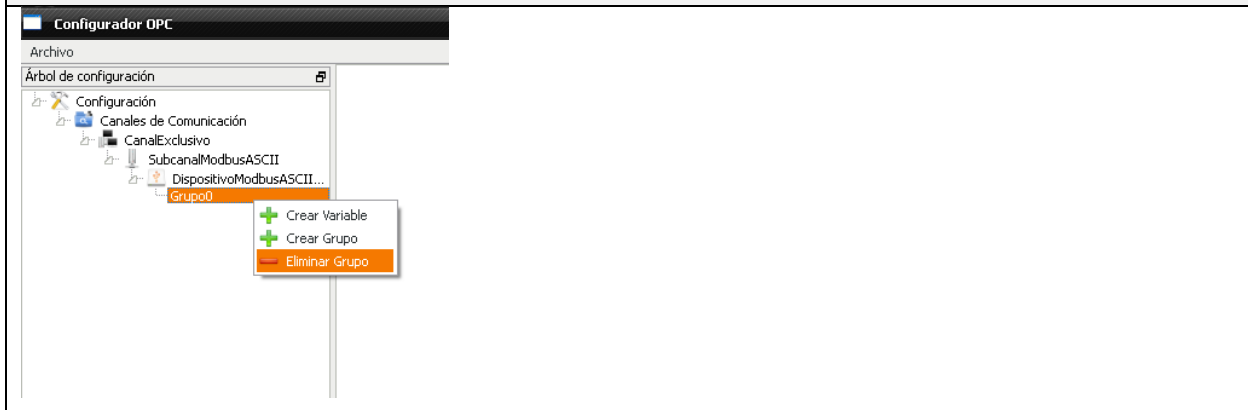


Tabla 7: Descripción del CU Gestionar Grupo OPC

Nombre del caso de uso:	CU Guardar Configuración
Actor	Operador
Propósito	Permite guardar la configuración.

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

Resumen	Se inicia cuando el operador decide guardar la configuración	
Referencia	RF 6	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir previamente una configuración.	
Post-condiciones	Queda guardado en un XML la configuración editada.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1. El operador necesita guardar la configuración.	1.1 El sistema muestra una interfaz gráfica para especificar donde guardar la configuración.	
2. El operador selecciona el directorio donde guardar la configuración.	2.1 El sistema guarda la configuración en el directorio especificado	
Curso alternativo de los eventos:		
Prototipo de interfaz de usuario		

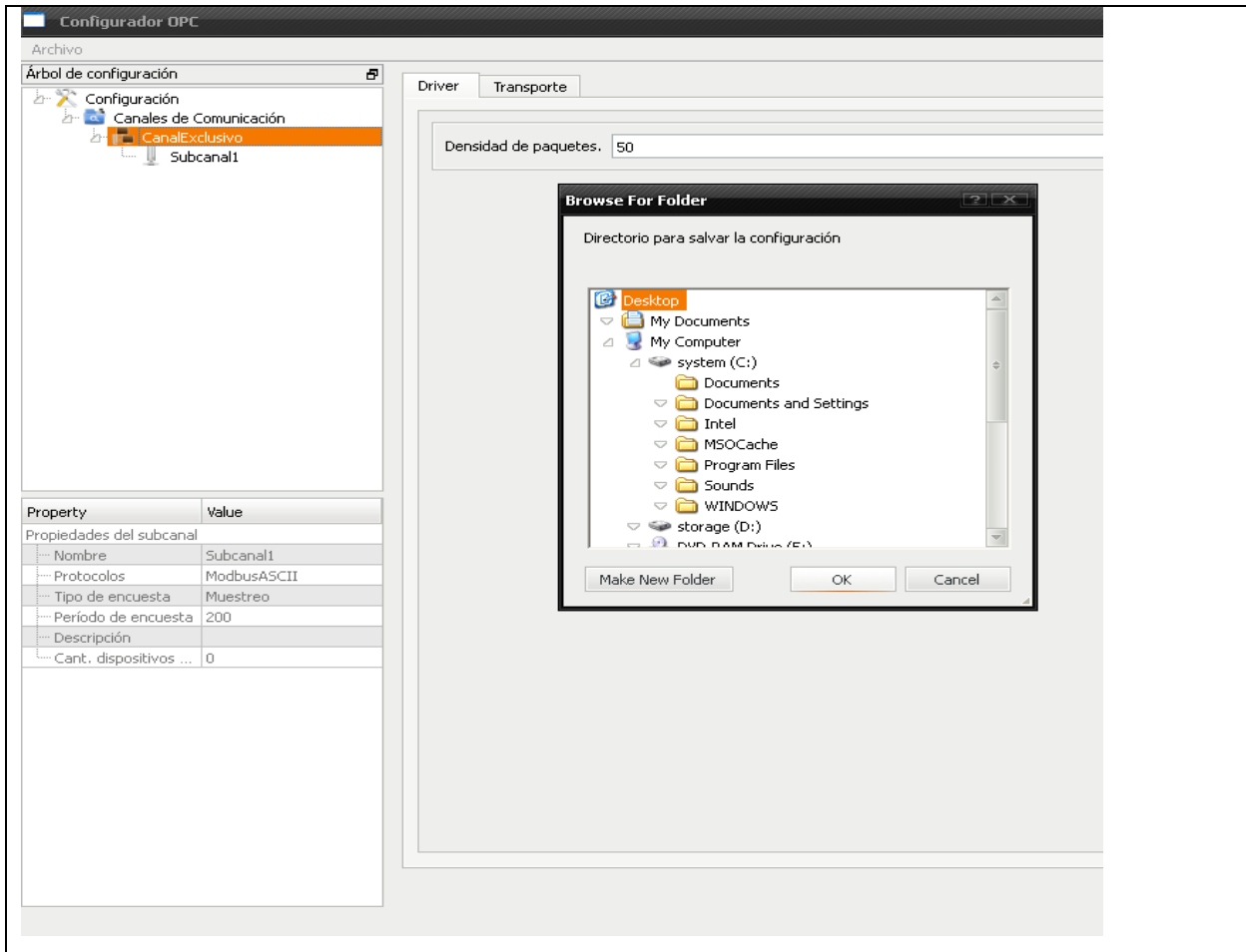


Tabla 8 : Descripción CU Guardar Configuración

Nombre del caso de uso:	CU Iniciar Servidor OPC DA
Actor	Sistema Externo
Propósito	Iniciar servidor OPC DA y publicar las variables configuradas por el editor
Resumen	Se inicia cuando el operador decide guardar la configuración
Referencia	RF 8
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	Existencia de un fichero de configuración.
Post-condiciones	Queda iniciado el servidor OPC DA.

Curso Normal de los eventos	
Acción del actor	Respuesta del sistema
1. El Sistema Externo invoca las interfaces COMDCOM del servidor OPC	1.1 El sistema activa el módulo de recolección. 1.2 El sistema carga la configuración. 1.3 El sistema crea el espacio de direcciones OPC a partir de la configuración cargada.
	2.1 El sistema se inicia. 2.2 El sistema empieza el proceso de recolección y publicación de variable mediante el estándar OPC.
Curso alternativo de los eventos:	
Prototipo de interfaz de usuario	

Tabla 9: Descripción del CU Iniciar publicación OPC

2.4 Conclusiones parciales

En este capítulo se expusieron las bases del desarrollo del sistema propuesto. En el siguiente capítulo se diseñará e implementará una estructura de clases en correspondencia con lo definido previamente en este capítulo, para de esta forma dar cumplimiento a los objetivos propuestos.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN

3.1 Introducción

En el presente capítulo se realiza el diseño del sistema, donde se modelan los subsistemas y sus clases separadas por paquetes, para lograr así un mejor entendimiento de los mismos. Se exponen además los diagramas de componentes de cada uno de los paquetes. Después serán implementadas las clases del diseño para arrojar como resultado un sistema cumpla con los requisitos funcionales expuestos en el capítulo anterior.

3.2 Diseño del sistema.

La función principal del sistema a diseñar es la recolección y publicación de variables mediante el estándar OPC el cual permita a los clientes OPC leer y escribir las variables publicadas. Para llevar a cabo este objetivo se decidió separar por subsistemas la solución, uno crear una interfaz visual que permita crear la configuración del servidor OPC, nombrado “Configurador OPC”, y otro encargado de la recolección desde distintos dispositivos de campo y la publicación sobre el estándar OPC, nombrado “Servidor OPC”. Se define de esta forma porque el Servidor OPC debe correr en un segundo plano, además que, cuando inicie cargue desde un fichero XML la configuración editada por el Configurador OPC.

3.2.1 Diseño del subsistema “Configurador OPC”.

El subsistema “Configurador OPC” agrupa las funcionalidades de los CUS Gestionar Canal, Gestionar Subcanal, Gestionar Dispositivo, Gestionar Variable, Gestionar Grupo y Guardar Configuración de los cuales las relaciones están representadas en el diagrama de CUS. El subsistema está formado por tres paquetes en dos niveles o capas distintas. Para el diseño de este subsistema arquitectónico se utilizó el patrón n-capa como se muestra en la siguiente imagen

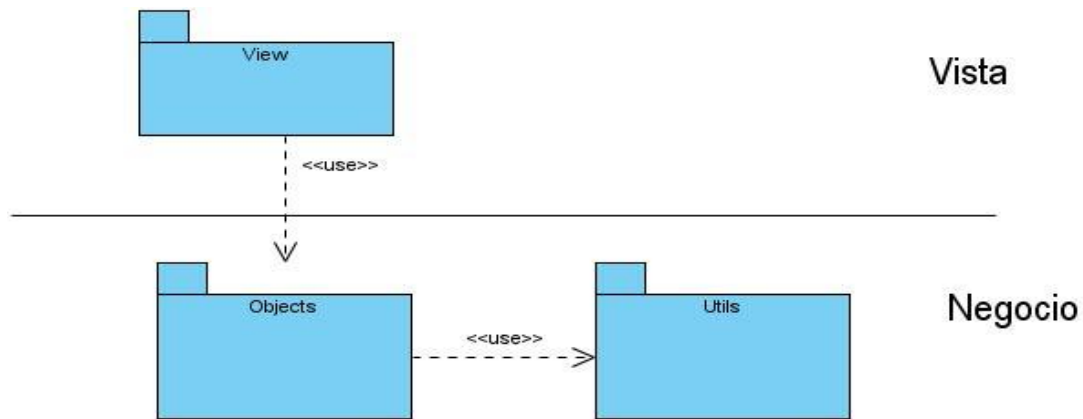


Ilustración 7: Arquitectura en capa del subsistema Configurator OPC

En la capa de negocio se encapsulan los paquetes que contienen la funcionalidad del subsistema, como son:

- “Objects”: Este paquete contiene la lógica del dominio, que está representado por clases abstractas o del mundo real y sus relaciones.
- “Utils”: Este paquete contiene clases y métodos que poseen funcionalidades reutilizables por el paquete Objeto.

En la capa vista se encuentra el paquete View que contiene todas las interfaces gráficas con que el usuario interactúa. Permitiendo abstraer al usuario de la lógica del negocio.

3.2.1.1 Paquete View.

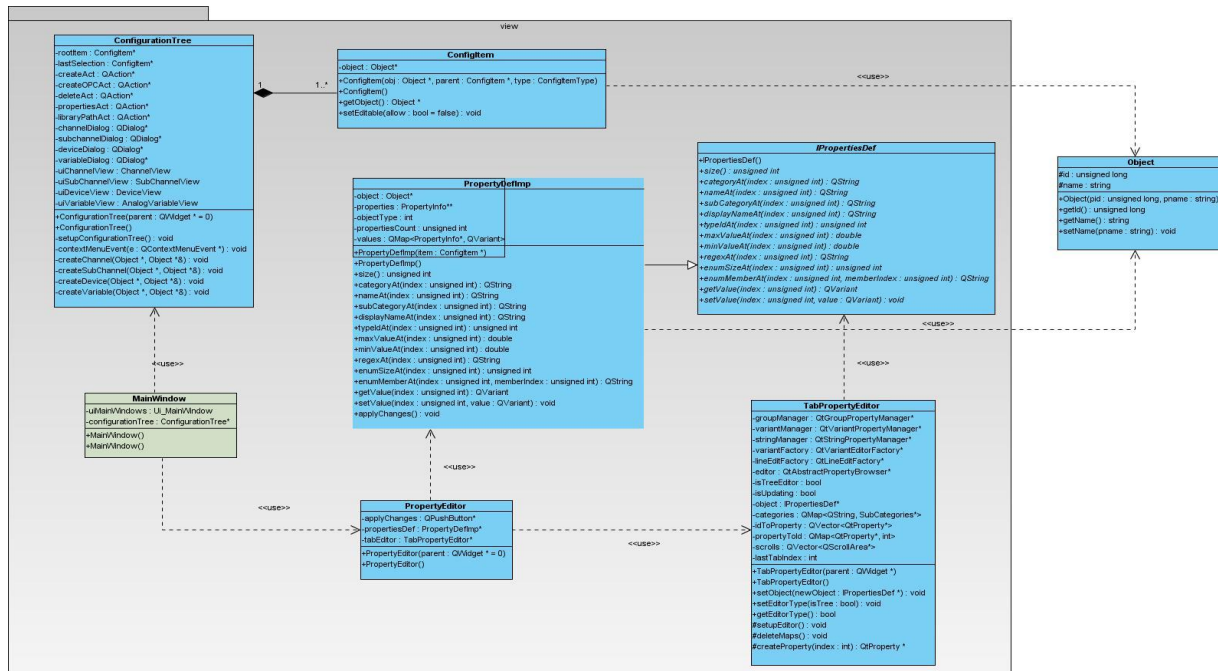


Ilustración 8: Diagrama de clases del paquete View.

- Descripción de las clases del paquete View.

Nombre de la clase	Descripción
MainWindow	Clase principal del paquete. Esta clase implementa la interfaz de usuario principal, con la que el usuario va a interactuar.
ConfigurationTree	Representa el árbol de configuración de la aplicación
ConfigItem	Clase que encapsula las características comunes a los elementos de configuración.
PropertyEditor	Clase que se encarga de edición de las propiedades
PropertyDefImp	Representa una instancia configurable. Solo puede contener un dispositivo o un manejador.
TabPropertyEditor	Clase que se encarga de editar las propiedades
IPropertiesDef	Representa un elemento con propiedades

Tabla 10: Descripción de las clases del paquete View.

3.2.1.2 Paquete Objects

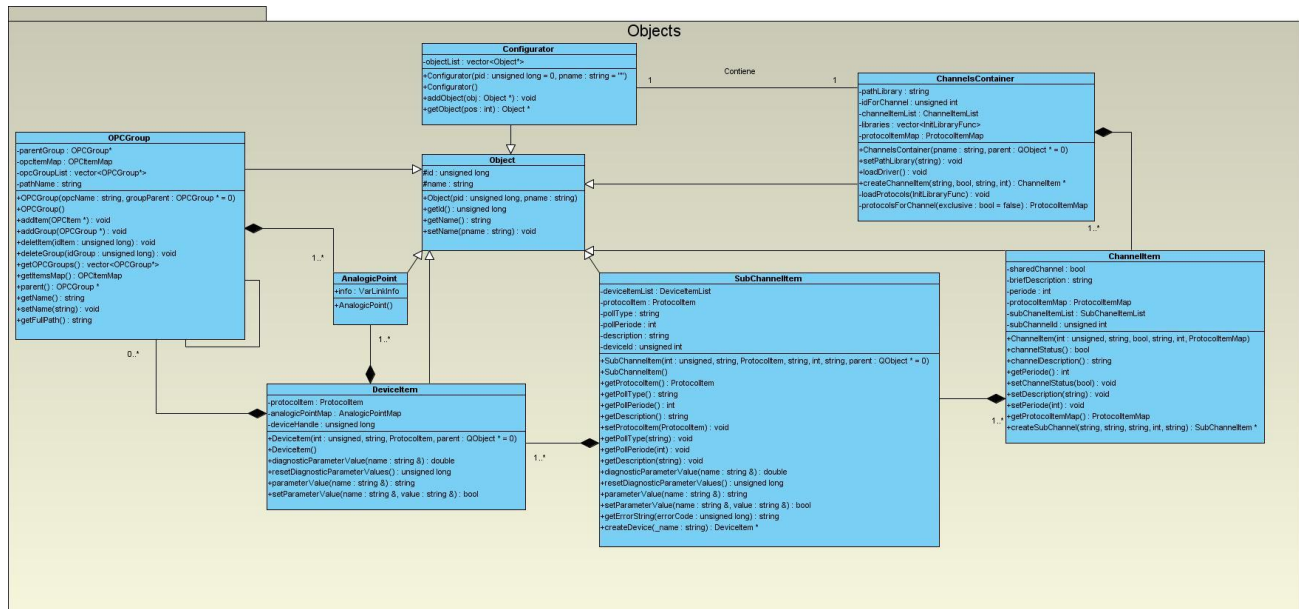


Ilustración 9: Diagrama de clases del paquete Objects.

- Descripción de las clases del paquete Objects

Nombre de la clase	Descripción
Object	Clase base de los objetos del visual.
Configurator	Clase donde empieza el árbol de configuración.
ChannelsContainer	Clase que contiene todos los canales.
ChannelItem	Clase que encapsula el concepto de canal de comunicación.
SubChannelItem	Clase que encapsula el concepto de subcanal de comunicación, y contiene las propiedades de configuración del manejador.
DeviceItem	Clase que encapsula el concepto de dispositivo y contiene las propiedades para el acceso al dispositivo. Esta clase almacena las variables y los grupos que se le creen.
AnalogicPoint	Clase que contiene la información de escritura y lectura de una variable.
OPCGroup	Clase utilizada de forma organizativa el cual almacena grupos y

variabes.

Tabla 11: Descripción de las clases del paquete Objeto.

En este subsistema se aplican varios patrones de diseño:

- **Experto:** Este patrón consiste en asignar la responsabilidad al experto en la información o sea la clase que posea la información necesaria para cumplir la responsabilidad. Las clases que aplican este patrón se encuentran en todo este subsistema.

3.2.2 Diseño del subsistema “Servidor OPC”

El subsistema “Servidor OPC” está compuesto por dos módulos ya implementados que son: el módulo OPCWorkshop y el Recolector. En este subsistema se agrupa el CUS Iniciar Servidor OPC y CUS Cargar configuración. Al igual que el subsistema anterior este utiliza el patrón arquitectónico N-capas.

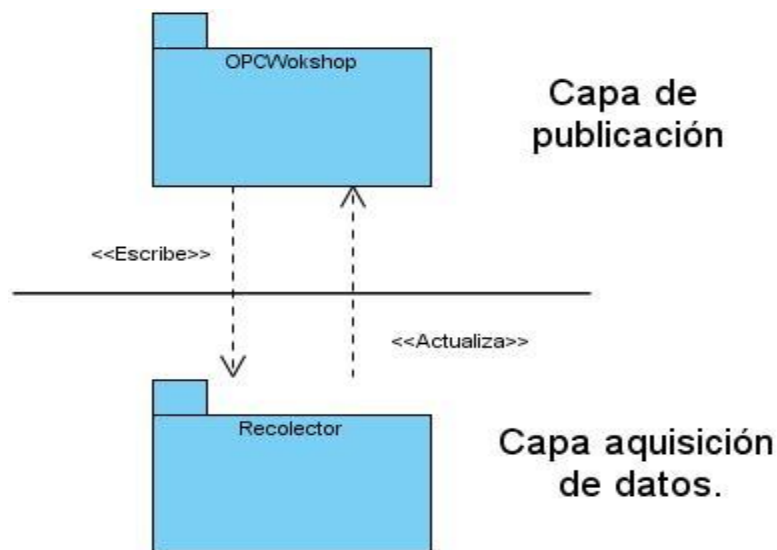


Ilustración 10: Arquitectura en capa del Subsistema Servidor OPC.

Cada capa está relacionada con un módulo diferente. Para lograr la comunicación de dichos módulos se diseñaron varias interfaces, las cuales se exponen a continuación.

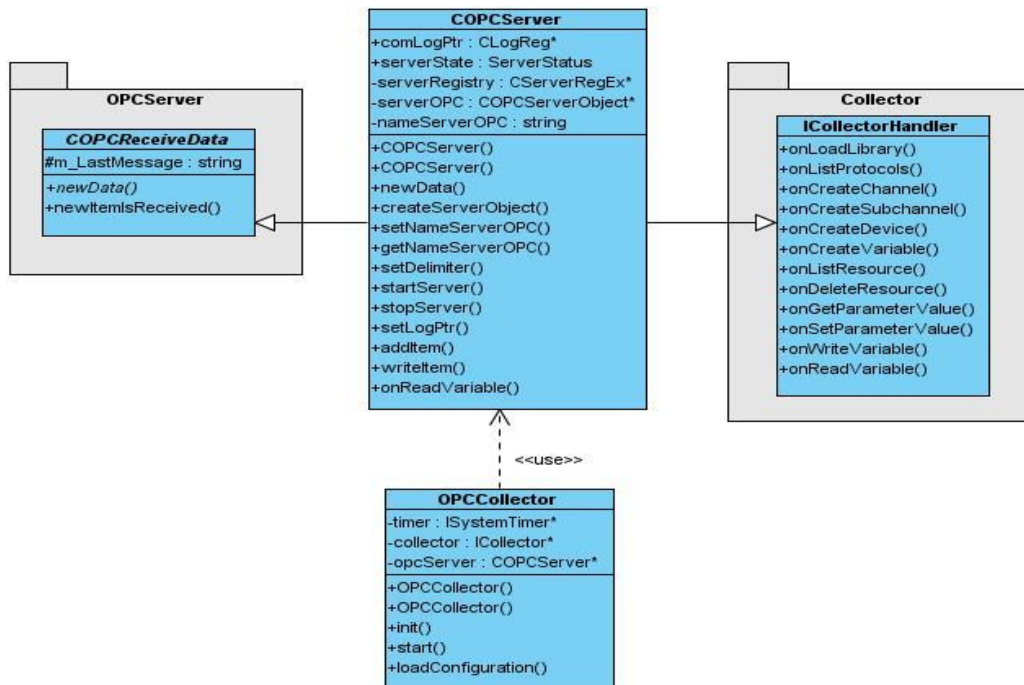


Ilustración 11: Diagrama de clases del subsistema Servidor OPC.

Nombre de la clase	Descripción
COPCReceiveData	Clase abstracta, receptor de datos desde conexiones OPC.
IColecторHandler	Clase que permite la re-implementación de sus métodos en las clases hijas para uso del recolector.
COPCServer	Clase OPCServer. Desciende de COPCReceiveData y de IColecторHandler.
OPCCollector	Clase OPCCollector. Se encarga de manejar las funcionalidades del recolector que se usa en el Servidor OPC. Es el encargado de leer y escribir variables de diversos protocolos y debe ser configurado para que realice sus operaciones.

Tabla 12: Descripción de las clases del subsistema Servidor OPC.

3.3 Implementación del sistema.

3.3.1 Estilo de código.

Nombres.

- Los nombres de las clases son sustantivos singulares.
- Los nombres deben reflejar el qué y no el cómo.
- Los nombres no deben revelar detalles de implantación.
- Escoger nombres lo suficientemente largos para ser expresivos, pero evitando manejar nombres que dificulten la labor de implantación.
- Evitar nombres que permitan una interpretación subjetiva (evitar ambigüedad y asegurar abstracción).
- Evitar redundancia no repitiendo nombres de clases en sus elementos.
- Concatenar calificadores de cómputo a las variables que almacenen el producto de tal cómputo (avg, sum, min, max, index).
- Dado que los nombres generalmente son el producto de concatenar varias palabras, se debe emplear mayúscula para el inicio de cada palabra y minúscula para el resto, con excepción de la primera letra del nombre, la cual debe ser en minúscula.
- En el caso de las clases se utiliza la estructura anterior pero con la excepción de que la primera letra del nombre debe ser en Mayúscula.
- Variables booleanas deben contener “is” en su nombre.
- Los nombres de constantes deben contener solo letras mayúsculas.
- Minimizar el uso de abreviaciones. En caso de ser requeridas, se debe ser consistente en su uso y cada abreviación debe significar solo una cosa. En general agregar a la documentación las abreviaturas.
- Los nombres de los métodos son frases que incluyen verbos.
- Los nombres de los atributos y parámetros son frases con sustantivos.
- Evitar el rehuso de nombres para distintos propósitos.

Manejo de Errores.

- Se pueden manejar los errores mediante mecanismos de excepciones o mediante valores de retorno, aunque esto debe ser uniforme dentro de un mismo objeto.
- Es buena práctica emplear herramientas para identificar errores en la codificación en caliente.

Documentación y Comentarios.

- En el código debe documentarse en forma explicativa los pasos que se van ejecutando.
- Emplear oraciones completas al documentar código.
- Documentar mientras se programa.
- Documentar cualquiera cosa que no sea obvia en el código.
- Documentar eliminación de errores y cambios sobre el código.
- Al modificar el código se deben actualizar todos los comentarios y documentación asociada.
- Documentar cada rutina agregando: nombre del desarrollador, fecha, parámetros de entrada, valores de retorno, precondiciones, post-condiciones, dependencia con otros métodos o funciones y descripción general del algoritmo. Además, de realizarse cambios al código, debe indicarse el nombre de la persona que realizó el cambio, la fecha y la descripción del cambio, comenzando desde el o los cambios más recientes.
- Evitar agregar comentarios al final de líneas de código, salvo en el caso de declaraciones. En este caso tales comentarios deben estar alineados.
- Antes de la entrega de la aplicación, eliminar todos los comentarios superfluos y/o temporales con la finalidad de evitar confusiones en su mantenimiento.

Codificación.

- Se establece un tamaño de indentación estándar de cuatro espacios, sin tabulaciones. Alinear secciones del código.
- Alinear verticalmente llaves de apertura y cierre.
- Usar espacios antes y después de los operadores que el lenguaje de programación permita.
- Emplear líneas en blanco para organizar el código, permitiendo crear párrafos de código para una mejor lectura.
- Evitar colocar más de una sentencia por línea.
- Emplear constantes en sustitución de números o cadenas de caracteres literales.
- Minimizar el alcance de las variables para evitar confusión y facilitar el mantenimiento.
- Emplear cada variable y rutina solo para un propósito.
- Evitar el uso de variables públicas, sustituirlas por variables privadas y métodos que provean el valor de tal variable, para mantener el encapsulamiento.

- Minimizar el uso de conversiones de tipo forzadas (castings), cuando se requiera su uso, debe ser comentada la justificación.
- Emplear select-case o switch en sustitución de if anidados sobre las mismas variables.
- Liberar apuntadores de manera explícita.
- Emplear i, j, k, l, p, q, r para contadores en ciclos.
- Comentar siempre las llaves que cierran.
- Emplear al máximo operadores del tipo: +=, *=, /=, -=, ++, --, etc.
- Mantener la modularidad del código bajo el criterio de la lógica que encierra, no exagerar la modularidad.
- Emplear correctamente los tipos de ciclos: si es al menos una vez usar *do-while*, si es ninguna o más veces usar *while*, y si se conoce el número exacto de ciclos usar for.
- Inicializar todas las variables.
- Emplear líneas en blanco para separar pasos lógicos (declaraciones, lazos, etc.).
- Siempre asignar NULL a los apuntadores luego de ser destruidos (solo aplica para C).
- Evitar prácticas que incrementan explosivamente la complejidad, como lo son: objetos y variables globales y saltos tipo *goto*.

3.3.2 Diagramas de componentes.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándar de componentes son los siguientes: (7)

- <<executable>>: Es un programa que puede ser ejecutado en un nodo.
- <<file>>: Es un fichero que contiene código fuente a datos.
- <<library>>: Es una biblioteca estática o dinámica.
- <<table>>: Es una tabla de la base de datos..
- <<document>>: Es un documento.

A continuación se muestra los diagramas de componentes por paquetes, en los mismos se muestran las distribuciones de las clases en fichero de código fuente .h y .cpp.

3.3.2.1 Diagrama de componentes del paquete View.

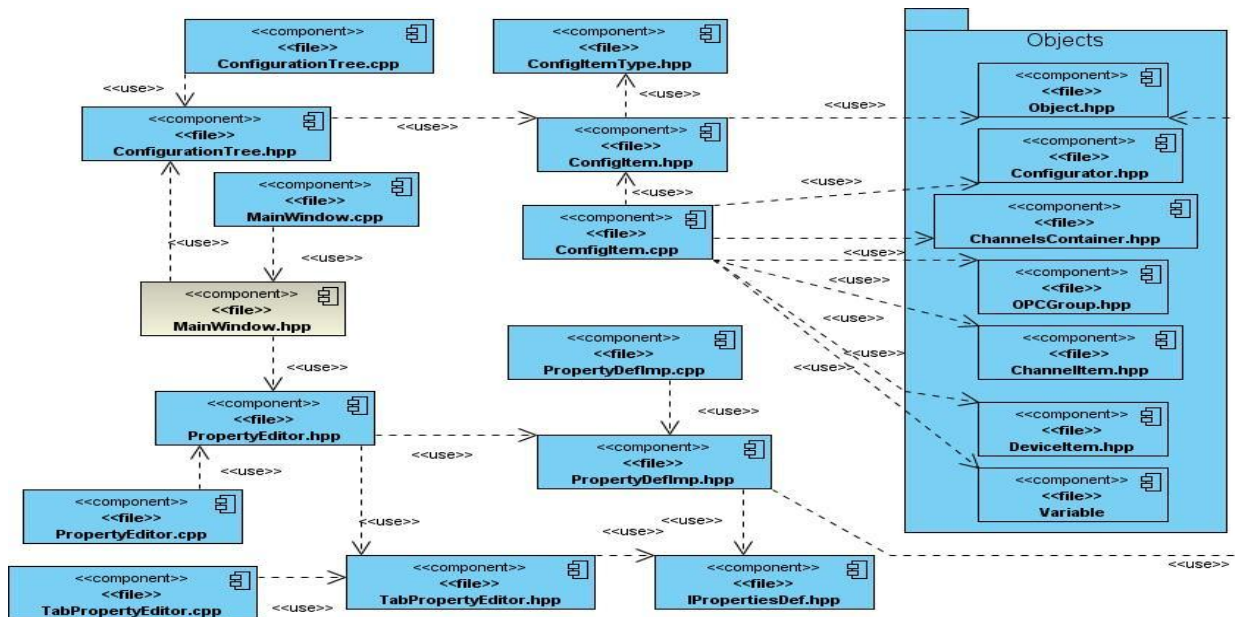


Ilustración 12: Diagrama de componente del paquete View.

3.3.2.2 Diagrama de componentes del paquete Object.

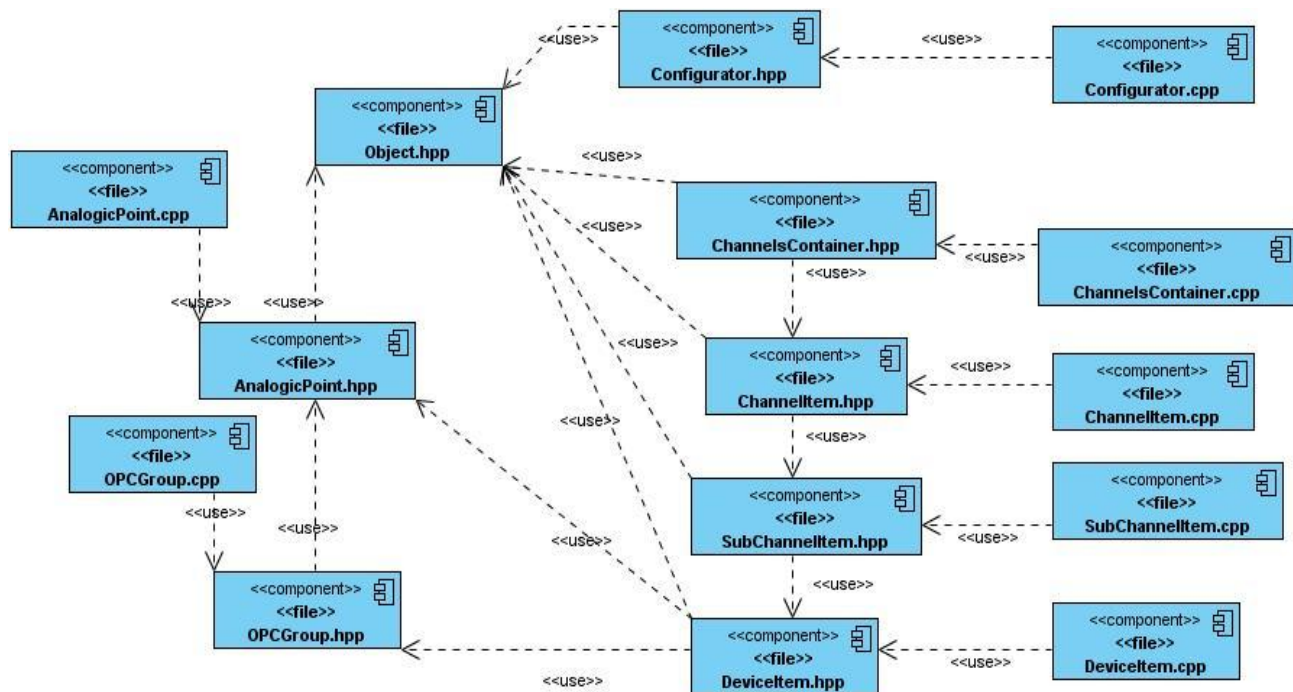


Ilustración 13: Diagrama de componentes del paquete Object.

3.3.2.3 Diagrama de componentes del subsistema Servidor OPC.

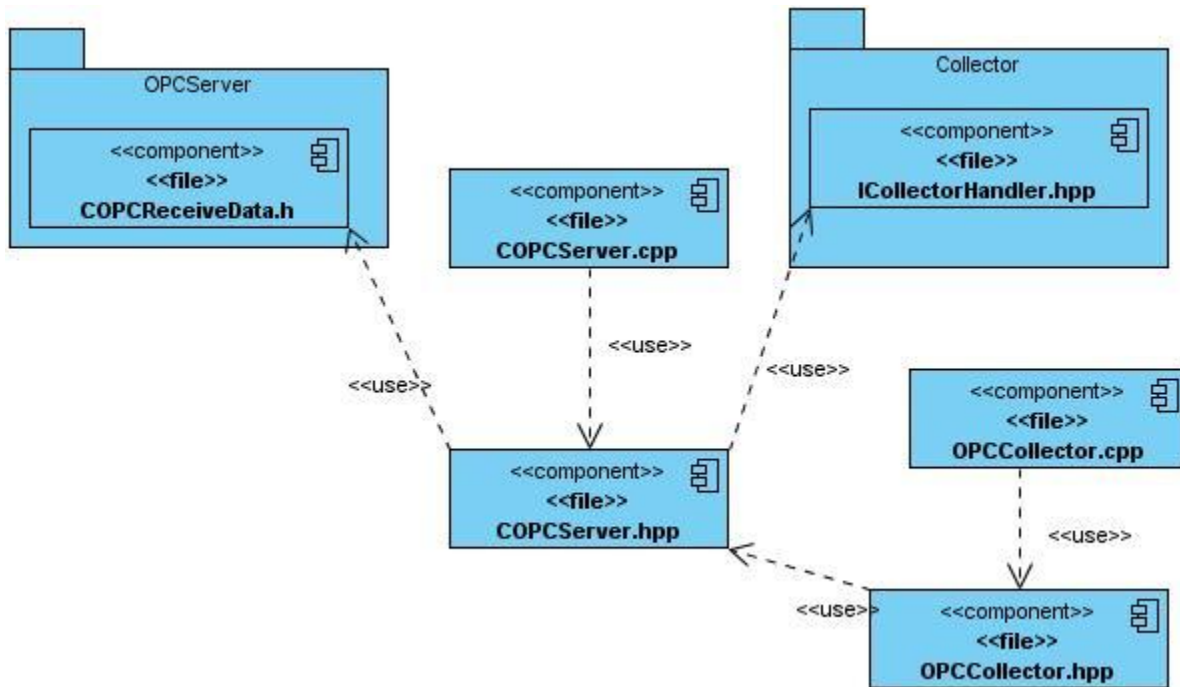


Ilustración 14: Diagrama de componentes del subsistema Servidor OPC.

3.3.2.4 Diagrama de componentes del sistema.

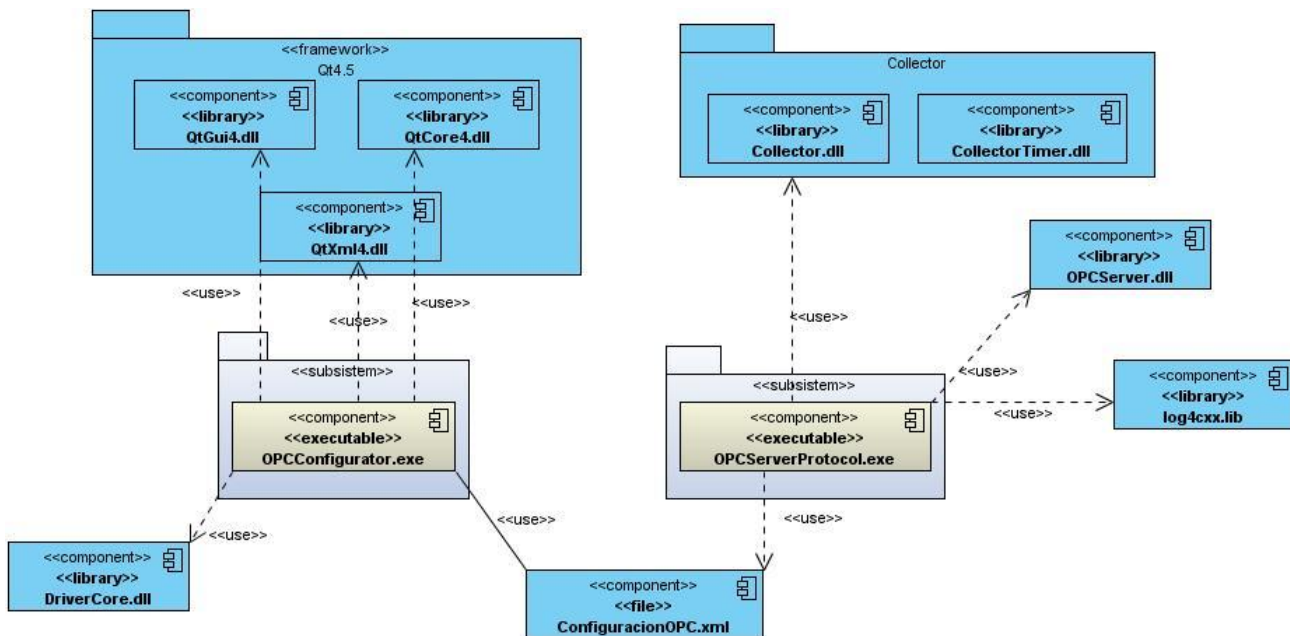


Ilustración 15: Diagrama de componentes del sistema.

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

En el diagrama anterior existen dos componentes principales que son: el componente OPCConfigurator.exe y el OPCServerProtocol.exe, ambos tiene estereotipos <<executable>> lo cual es el resultado del proceso de compilar el código fuente de los subsistemas antes descritos. Los demás componentes que se encuentra representados ayudan al buen funcionamiento y/o rendimiento del sistema.

3.4 Conclusiones parciales.

A través de este capítulo quedaron expuestas las clases de diseño para cada subsistema y sus diagramas de componentes, lo que dio a lugar a la implementación de los subsistemas donde se lograron los requisitos funcionales descritos en el capítulo anterior. Después de la implementación, se le realizaron pruebas que pueden encontrarse en los anexos del presente trabajo.

CONCLUSIONES GENERALES

Una vez alcanzado los objetivos trazados para este trabajo, el Centro de Desarrollo Informática Industrial (CEDIN), cuenta con un servidor OPC DA independiente a los servicios del SCADA GALBA, ya que posee su propio mecanismo de configuración y adquisición de datos.

La integración con el módulo de adquisición posibilita la utilización de los manejadores desarrollados para el SCADA Guardián del Alba, así como cualquier otro que cumpla con la interfaz genérica de los manejadores.

El configurador cuenta con una interfaz de usuario amigable, permite la configuración de canales, manejadores y dispositivos necesario para el proceso de comunicación con la fuente de dato a través de los distintos protocolos. Además posibilita crear el espacio de direcciones del Servidor OPC DA.

RECOMENDACIONES

Luego de haber concluido este trabajo se recomienda:

- Integrarle al servidor OPC un mecanismo que le permita utilizarse como un servicio de Windows, con vista de poder detener al servidor sin necesidad de terminar el proceso bruscamente.
- Mejorar la edición del espacio de direcciones del Servidor OPC DA.

REFERENCIAS BIBLIOGRÁFICAS

1. **Rodríguez Penin, Aquilino.** *Sistemas SCADA - 2da Edición.* Barcelona España : MARCOMBO, 2007.
2. **Foundation, Opc.** *OPC Data Access Custom Interface Specification.* s.l. : Internet, 2002.
3. **Foundation, OPC.** The OPC Foundation. *The OPC Foundation.* [En línea] OPC Foundation, 2010. [Citado el: 4 de Junio de 2010.] http://opcfoundation.org/Default.aspx/01_about/01_history.asp?MID=AboutOPC.
4. **Kepware Technologies, Inc.** Kepware Technologies, Inc. *Kepware Technologies, Inc.* [En línea] 2010. [Citado el: 5 de Mayo de 2010.] http://www.kepware.com/products/products_opcservers.asp.
5. OPC DataHub. *OPC DataHub.* [En línea] Cogent Real-Time Systems Inc., 1995-2010. [Citado el: 5 de Mayo de 2010.] <http://www.opcdatahub.com/Home.html>.
6. **Inc, Matrikon.** MatrikonOPC. *MatrikonOPC.* [En línea] Matrikon Inc, 2010. [Citado el: 5 de Mayo de 2010.] <http://www.matrikonopc.com/>.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 1999.
8. **Microsoft.** Visual C++. *MSDN.* [En línea] Microsoft, 2010. [Citado el: 1 de Mayo de 2010.] [http://msdn.microsoft.com/es-es/library/60k1461a\(v=VS.80\).aspx](http://msdn.microsoft.com/es-es/library/60k1461a(v=VS.80).aspx).
9. **Object Management Group, Inc.** *OMG Unified Modeling Language.* s.l. : Internet, 1999.
10. **Soulie, Juan.** cplusplus.com. *cplusplus.com.* [En línea] 29 de Septiembre de 2009. [Citado el: 1 de Mayo de 2010.] <http://cplusplus.com/info/description/>.
11. **Foundation, OPC.** OPC Unified Architecture. *OPC Foundation.* [En línea] OPC Foundation, 2010. [Citado el: 2 de Mayo de 2010.] http://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC#Presentations.

BIBLIOGRFÍA

- AG, T. (2010). *Technosoftware*. Recuperado el 5 de Mayo de 2010, de Technosoftware:
<http://www.technosoftware.com/opc>
- Corporation, N. (2008-2010). *Qt*. (Nokia Corporation) Recuperado el 1 de Mayo de 2010, de
<http://qt.nokia.com/>
- DataHub, O. (1995-2010). *OPC DataHub*. (Cogent Real-Time Systems Inc.) Recuperado el 5 de Mayo de 2010, de OPC DataHub: <http://www.opcdatahub.com/Home.html>
- Foundation, O. (2002). *OPC Data Access Custom Interface Specification*. Internet.
- Foundation, O. (2010). *OPC Unified Architecture*. (OPC Foundation) Recuperado el 2 de Mayo de 2010, de OPC Foundation.
- Foundation, O. (2010). *The OPC Foundation*. (O. Foundation, Editor, & OPC Foundation)
Recuperado el 4 de Junio de 2010, de The OPC Foundation:
http://opcfoundation.org/Default.aspx/01_about/01_history.asp?MID=AboutOPC
- Inc, M. (2010). *MatrikonOPC*. (Matrikon Inc) Recuperado el 5 de Mayo de 2010, de MatrikonOPC:
<http://www.matrikonopc.com/>
- International, V. P. (2004-2010). *Visual Paradigm UML 7.2 Model-Code-Deploy Platform*.
Recuperado el 6 de Mayo de 2010, de Boost Productivity with Innovative and Intuitive Technologies: <http://www.visual-paradigm.com/product/vpum/>
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Modeling Language Reference Manual*. California: Addison Wesley, Longman, Inc.
- Kepware Technologies, I. (2010). *Kepware Technologies, Inc.* Recuperado el 5 de Mayo de 2010, de Kepware Technologies, Inc.: http://www.kepware.com/products/products_opcservers.asp
- Microsoft. (2010). *Visual C++*. (Microsoft) Recuperado el 1 de Mayo de 2010, de MSDN:
[http://msdn.microsoft.com/es-es/library/60k1461a\(v=VS.80\).aspx](http://msdn.microsoft.com/es-es/library/60k1461a(v=VS.80).aspx)

Object Management Group, I. (1999). *OMG Unified Modeling Language*. Internet.

Paine, T. (1 de July de 2008). *Sensors*. Recuperado el 2 de Mayo de 2010, de Meet the Next Generation of OPC: <http://www.sensorsmag.com/networking-communications/industrial-networking/meet-next-generation-opc-1494>

Rodríguez Penin, A. (2007). *Sistemas SCADA - 2da Edición*. Barcelona España: MARCOMBO.

Soulie, J. (29 de Septiembre de 2009). *cplusplus.com*. Recuperado el 1 de Mayo de 2010, de [cplusplus.com: http://cplusplus.com/info/description/](http://cplusplus.com/info/description/)

ANEXOS

Anexo 1: Descripción del CU Cargar configuración en el configurador

Nombre del caso de uso:	CU Guardar Configuración	
Actor	Operador	
Propósito	Cargar la configuración antes editada	
Resumen	Se inicia cuando el Operador decide cargar configuración	
Referencia	RF 7	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Existencia de un fichero de configuración.	
Post-condiciones	Queda iniciado el servidor OPC DA.	
Curso Normal de los eventos		
Acción del actor	Respuesta del sistema	
1 El operador se decide cargar la configuración.	1.1 El sistema despliega una interfaz visual que le posibilita al usuario buscar el archivo XML que contiene la configuración. 1.2 El sistema carga la configuración.	
Curso alternativo de los eventos:		
Prototipo de interfaz de usuario		

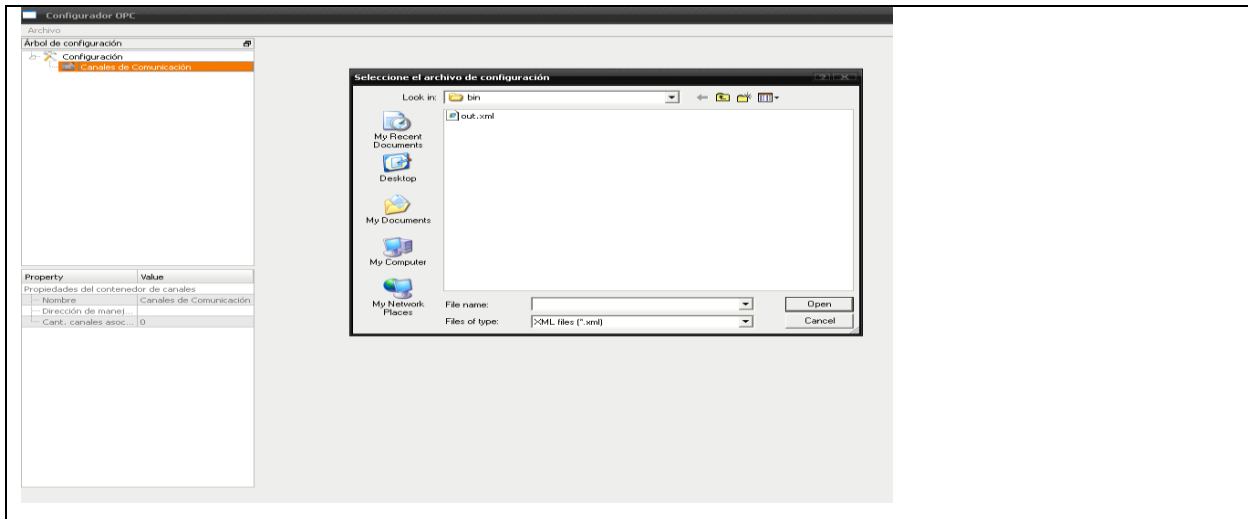


Tabla 13: Descripción del CU Cargan configuración en el Configurator.

Anexo 2: Pruebas realizadas al sistema.

Caso de prueba 1		
Acción del operador.	Respuesta esperada	Respuesta obtenida
<ul style="list-style-type: none"> - El operador se posiciona sobre el nodo “Canales de Comunicación” del árbol de configuración y presiona clic inverso sobre dicho nodo. - El operador selecciona la opción “Crear canal” del menú contextual que el sistema despliega. - El operador llena los campos de la interfaz gráfica con el mismo nombre del canal adicionado anteriormente y presiona el botón aceptar. <p>Datos de pruebas:</p> <ol style="list-style-type: none"> 1. Nombre : Canal1 2. Modo de Acceso: Compartido 3. Período: 300 4. Descripción: No obligatoria. 	<p>El sistema debe notificar al operador que los nombres de canales no se pueden repetir.</p>	<p>El sistema emite un mensaje de error especificando al operador que no puede insertar un canal de igual nombre que los ya insertados.</p> <p>Ilustración 16.</p>
Caso de uso 2		
Acción del operador.	Respuesta esperada	Respuesta obtenida

<p>Precondición: Se debió haber creado con anterioridad un canal y un subcanal de nombre “Subcanal1”.</p> <ul style="list-style-type: none"> - El operador se posiciona sobre el nodo “Canal1” creado anteriormente - El operador selecciona la opción “Crear Subcanal” del menú contextual que el sistema despliega. - El operador llena los campos de la interfaz gráfica con el mismo nombre del subcanal adicionado anteriormente y presiona el botón aceptar. <p>Datos de pruebas:</p> <ol style="list-style-type: none"> 1. Nombre : Subcanal1 2. Protocolo de comunicación: Se selecciona de los que en el combobox aparecen. 3. Tipo de encuesta: Muestreo 4. Período de encuesta: 200 5. Descripción: Campo no 	<p>El sistema debe notificar al operador que los nombres de los subcanales no se pueden repetir.</p>	<p>El sistema emite un mensaje de error especificando al operador que no puede insertar un subcanal de igual nombre que los ya insertados.</p> <p>Ilustración 17</p>
--	--	--

obligatorio.		
Caso de uso 3		
Acción del operador.	Respuesta esperada	Respuesta obtenida
<p>Precondición: Se debió haber creado con anterioridad un subcanal y un dispositivo de nombre "Dispositivo1".</p> <ul style="list-style-type: none"> - El operador se posiciona sobre el nodo "Subcanal1" creado anteriormente - El operador selecciona la opción "Crear Dispositivo" del menú contextual que el sistema despliega. - El operador llena los campos de la interfaz gráfica con el mismo nombre del dispositivo adicionado anteriormente y presiona el botón aceptar. <p>Datos de pruebas:</p> <ol style="list-style-type: none"> 1. Nombre : Dispositivo1 2. Descripción: Campo no 	<p>El sistema debe notificar al operador que los nombres de los Dispositivos no se pueden repetir.</p>	<p>El sistema emite un mensaje de error especificando al operador que no puede insertar un dispositivo de igual nombre que los ya insertados.</p> <p>Ilustración 18</p>

obligatorio.		
--------------	--	--

Caso de uso 4		
Acción del operador.	Respuesta esperada	Respuesta obtenida
<p>Precondición: Se debió haber creado con anterioridad un dispositivo y una variable de nombre "Variable1".</p> <p>-El operador se posiciona sobre el nodo "Dispositivo1" creado</p>	<p>El sistema debe notificar al operador que los nombres las Variables no se pueden repetir.</p>	<p>El sistema emite un mensaje de error especificando al operador que no puede insertar una variable de igual nombre que los ya</p>

<p>anteriormente.</p> <p>-El operador selecciona la opción “Crear Variable” del menú contextual que el sistema despliega.</p> <p>-El operador llena los campos de la interfaz gráfica con el mismo nombre de la variable adicionada anteriormente y presiona el botón aceptar.</p> <p>Datos de pruebas:</p> <ol style="list-style-type: none"> 1. Nombre : Variable1 2. Dirección de lectura: Dirección válida para el dispositivo antes configurado. 3. Dirección de escritura: Dirección válida para el dispositivo antes configurado. 4. Período de muestreo: 200 5. Tipo de dato de lectura: INT 6. Tipo de dato de escritura: INT 		<p>insertados.</p> <p>Ilustración 19.</p>
---	--	---

Caso de uso 5		
Acción del operador.	Respuesta esperada	Respuesta obtenida
Precondición: Se debió haber	El sistema debe guardar	El sistema

Configurador para el Servidor OPC de Acceso a Datos e integración con el Recolector del SCADA “Guardián del ALBA”

<p>creado con anterioridad canales, subcanales, dispositivos y variables.</p> <ul style="list-style-type: none"> - El operador selecciona en el menú archivo la opción “Guardar”. - El operador seguidamente especifica la dirección donde desea guardar la configuración. 	<p>correctamente la configuración.</p>	<p>guarda la configuración creada.</p> <p>Ilustración 20.</p>
--	--	---

Tabla 14: Casos de pruebas realizados

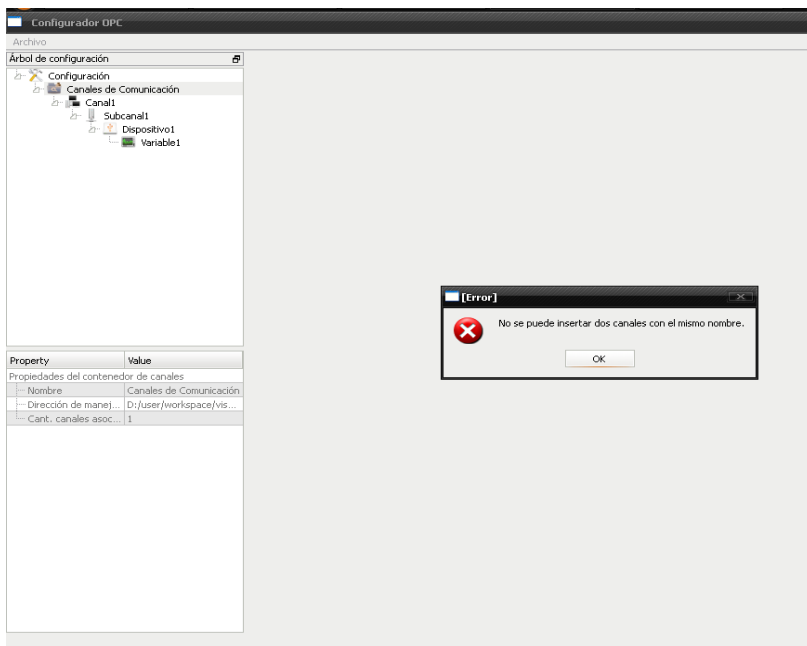


Ilustración 16: Caso de prueba 1.

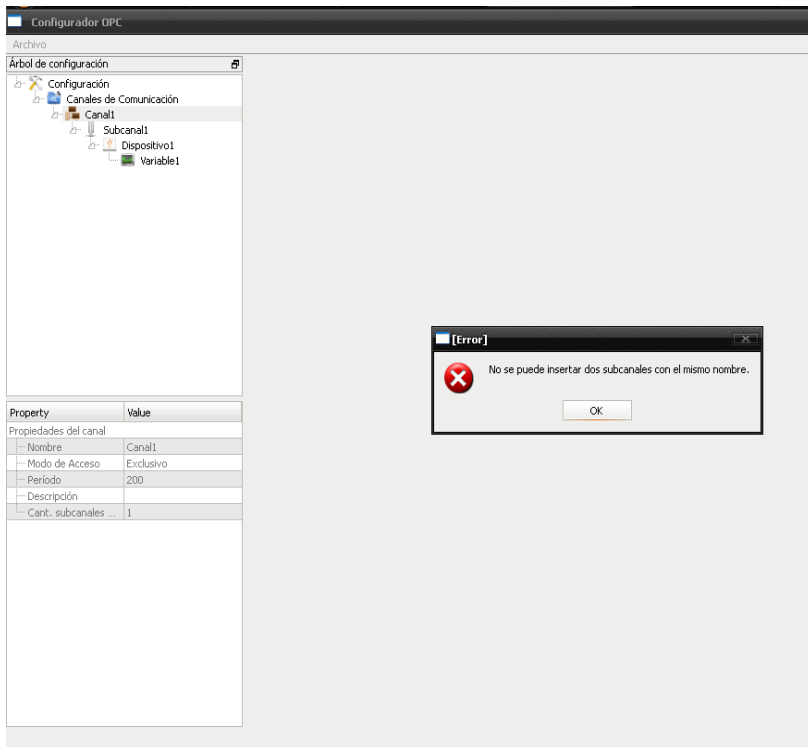


Ilustración 17: Caso de prueba 2.

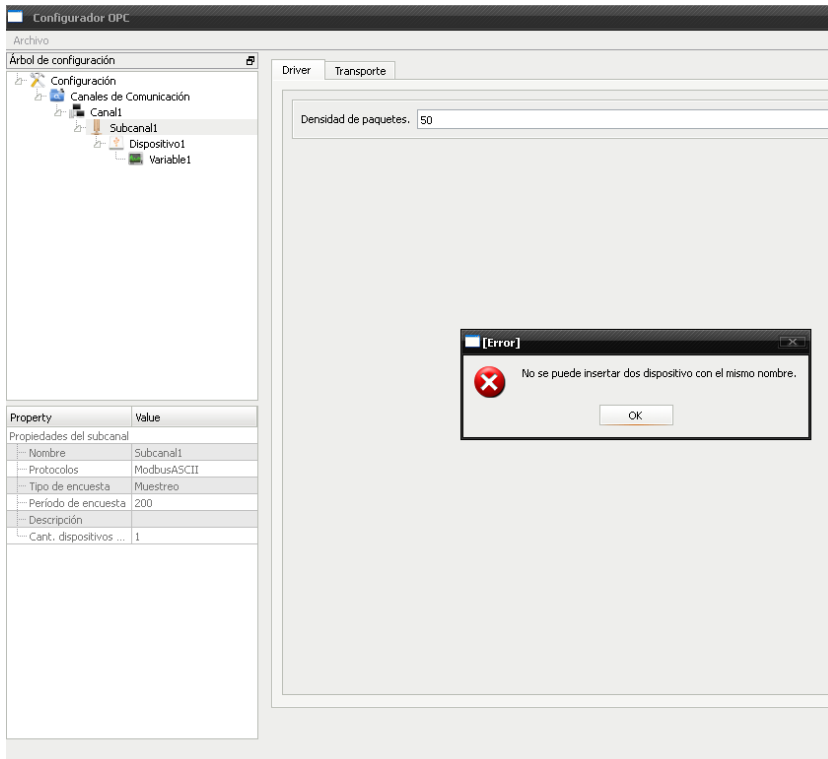


Ilustración 18: Caso de prueba 3.

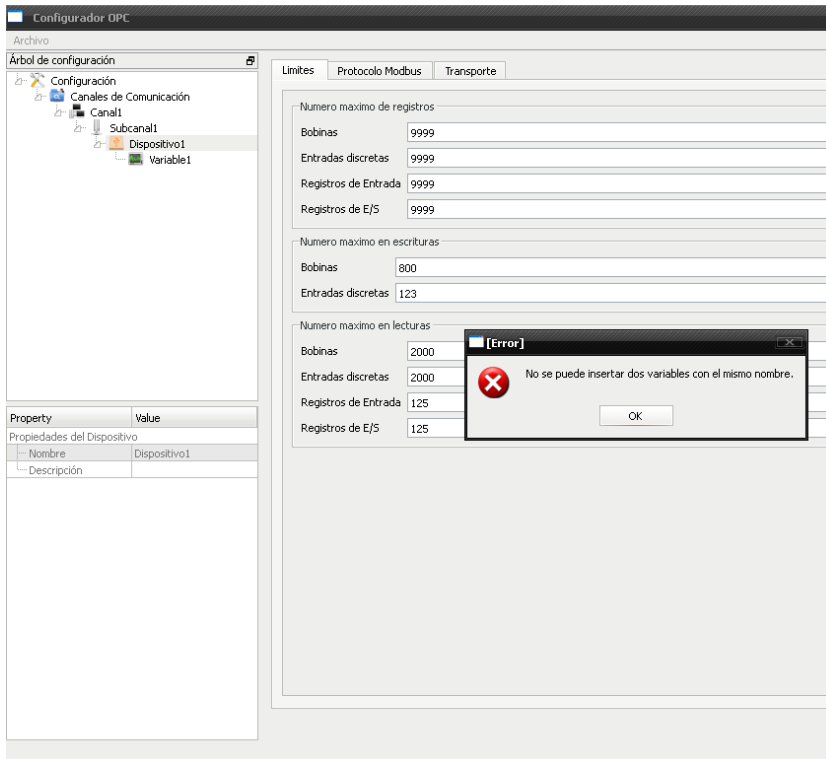


Ilustración 19: Caso de prueba 4.

```

<opcconfig>
  <channelCollection>
    <channel exclusive="1" id="1027" name="Canal1"/>
  </channelCollection>
  <deviceCollection>
    <device name="Dispositivo1" idSubchannel="603" idDevice="633"/>
  </deviceCollection>
  <libraryCollection libraryPath="D:/user/workspace/visual-studio/RecolectorGraficoSolution/bin/release/">
    <library>ModbusDriver.dll</library>
  </libraryCollection>
  <pointCollection>
    <point deviceId="633" period="200" writeAddress="1:100" pointId="1001" writeType="3"
      opcItemId="Canal1.Subcanal1.Dispositivo1.Variable1" readAddress="1:100" writeArraySize="5"
      name="Variable1" readArraySize="5" readType="3"/>
  </pointCollection>
  <propertyCollection>
    <property useTerminalServer="0" packageDensity="50" stopBits="StopBits1"
      hostName="/dev/ttyS0" ipPort="9999" id="603" dataBits="DataBit8" flowControl="FlowNone"
      baudRate="BaudRate9600" parity="ParityNone" ipAddress="192.168.10.59"/>
    <property maxInputs="9999" numberOfRetries="2" useSingleWrite="0" maxCoilsPerReadTransaction="2000"
      id="633" maxInputRegisters="9999" maxHoldingRegistersPerReadTransaction="125" maxHoldingRegisters="9999"
      maxInputRegistersPerReadTransaction="125" endianness="deviceLittleEndian" connectionTimeout="5000"
      responseTimeout="5000" inputsBaseAddress="0" slaveID="1" maxCoilsPerWriteTransaction="800"
      maxHoldingRegistersPerWriteTransaction="123" inputRegistersBaseAddress="0" maxInputsPerReadTransaction="2000"
      closeAfterTransaction="1" retriesInterval="1000" coilsBaseAddress="0" maxCoils="9999" holdingRegistersBaseAddress="0"/>
  </propertyCollection>
  <subChannelCollection>
    <subchannel protocolo="ModbusASCII" idChannel="1027" name="Subcanal1" idSubchannel="603"/>
  </subChannelCollection>
</opcconfig>

```

Ilustración 20: Caso de prueba 5

GLOSARIO DE TÉRMINOS Y ABREVIATURAS (GTA)

SCADA: Supervisión, Control y Adquisición de Datos, (por sus siglas en inglés **S**upervisory **C**ontrol and **D**ata **A**cquisition).

OLE: Acrónimo del término inglés “Object Linking and Embedding”. Es la tecnología para la vinculación e inserción de objetos en Windows.

OPC: Acrónimo del término inglés “OLE for Process Control”.

OPC DA: Especificación OPC de Acceso a Datos.

COM: **C**omponent **O**bject **M**odel es una plataforma de Microsoft para componentes de software.

DCOM: Acrónimo del término inglés (**D**istributed **C**omponent **O**bject **M**odel). Es una tecnología propietaria de Microsoft para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí.

Ciente OPC: Aplicación que solo utiliza los datos brindados por el servidor OPC.

Servidor OPC: Es una aplicación que recolecta de los dispositivos de campos y publica las variables recolectadas.

OPC HDA: (Historical Data Access, Acceso a Datos Históricos). Especificación OPC dedicado al acceso a datos históricos exclusivamente.

OPC A&E: (Alarm and Events, Alarmas y Eventos). Especificación OPC para el procesamiento de alarmas y eventos.

OPC DX: (Data Exchange, Intercambio de datos). Especificación OPC utilizada para el intercambio de datos entre servidores OPC

OPC XML: (Extensible Markup Language, Lenguaje de marcado extensible). Especificación OPC para el uso de XML.

OPC Security: (Seguridad), Especificación OPC enfocada a la seguridad en los Servidores OPC.

OPC-UA: (**OP**en **C**onnectivity-**U**nified **A**rchitecture, Conectividad abierta a la arquitectura unificada). Especificación que integra las funcionalidades de las anteriores especificaciones OPC.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema software.

RUP: Acrónimo (Rational Unified Process en inglés) Es una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable.

CASE: (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

QT: Es un framework multiplataforma utilizado principalmente para desarrollar interfaces gráficas de usuario.

IDE: Entorno de desarrollo integrado (acrónimo en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación.

PLC: Es un hardware industrial, que se utiliza para la obtención de datos.