

PRINCIPIOS DE APLICACIÓN DEL ESTANDAR  
OPC UA EN UN CASO DE ESTUDIO.



Carlos Alberto Castaño G.  
Edgar Fernando Villamarin M.

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
GRUPO I+D EN AUTOMÁTICA INDUSTRIAL  
INGENIERÍA EN AUTOMÁTICA INDUSTRIAL  
POPAYÁN 2008

PRINCIPIOS DE APLICACIÓN DEL ESTANDAR  
OPC UA EN UN CASO DE ESTUDIO.



Carlos Alberto Castaño G.  
Edgar Fernando Villamarin M.

Monografía presentada como requisito parcial para optar por el título de Ingeniero en  
Automática Industrial

Director  
MSc. OSCAR AMAURY ROJAS

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
GRUPO I+D EN AUTOMÁTICA INDUSTRIAL  
INGENIERÍA EN AUTOMÁTICA INDUSTRIAL  
POPAYÁN 2008

Nota de aceptación:

---

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Popayán, Noviembre de 2008

## AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

MSc. OSCAR AMAURY ROJAS

Ing. Libardo Steven Muñoz

Ing. Wolfgang Mahnke

Ing. Tom Burke

Ing. Carlos Ardila

Sección Estudiantil ISA Unicauca

Grupo de Investigación en Automática Industrial

OPC Foundation

.....

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

*Gracias a mi familia por su gran apoyo y comprensión, especialmente a mis padres por brindarme la oportunidad de realizar esta carrera y por ser siempre un ejemplo a seguir. A mis amigos y a la universidad de Cauca*  
*CARLOS*

*Gracias a Dios, a mi familia, a mis amigos y a la Universidad del Cauca.  
Pero sobretodo gracias a mi Mamá Gloria, mi Papá Jaime y a mis hermanos Andrés y  
Cristina por la fe en mi y su constante apoyo*  
*FERNANDO*

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	1
1 MARCO CONCEPTUAL .....	3
1.1 OPC ARQUITECTURA UNIFICADA (OPC UA) .....	3
2 ARQUITECTURA DE LOS SERVIDORES OPC UA .....	13
2.1 METAMODELO DE INFORMACIÓN DE OPC UA .....	13
2.2 ADDRESSSPACE DE OPC UA .....	14
2.2.1 NodeClass Base. ....	15
2.2.2 Reglas de diseño en el estándar OPC UA.....	17
2.3 SERVICIOS ABSTRACTOS DE OPC UA.....	19
2.3.1 Conjunto de servicios <i>Discovery</i> [7].....	19
2.3.2 Conjunto de servicios <i>SecureChannel</i> [7].....	19
2.3.3 Conjunto de servicios <i>Session</i> [7].....	20
2.3.4 Conjunto de servicios <i>NodeManagement</i> [7]. ....	20
2.3.5 Conjunto de servicios <i>View</i> [7].....	20
2.3.6 Conjunto de servicios <i>Query</i> [7]. ....	20
2.3.7 Conjunto de servicios <i>Attribute</i> [7]. ....	20
2.3.8 Conjunto de servicios <i>Method</i> [7]. ....	20
2.3.9 Conjunto de servicios <i>MonitoredItem</i> [7]. ....	21
2.3.10 Conjunto de servicios <i>Subscription</i> [7]. ....	21
2.4 TECNOLOGÍAS BASE PARA LA COMUNICACIÓN EN OPC UA .....	21
2.4.1 Perfil de los servidores OPC UA. ....	22
2.4.2 Stack de Servicios web XML [10].....	23
2.4.3 Stack Binario nativo con SOAP [10]. ....	24
2.4.4 Stack Binario Nativo [10]. ....	24
3 ANÁLISIS Y DISEÑO PARA LA IMPLEMENTACIÓN DE UN SERVIDOR OPC UA BASADO EN LA METODOLOGÍA ORIENTADA A OBJETOS.....	25
3.1 DESCRIPCIÓN DEL SISTEMA.....	25
3.1.1 Usuario. ....	27
3.1.2 ClienteUA.....	27
3.1.3 RegistroUsuario. ....	27
3.1.4 ActualizarSeguridad. ....	28
3.1.6 Sesión.....	30
3.1.7 ActualizarSesion. ....	31
3.1.8 ValidarUsuario.....	32
3.1.9 Subscripciones. ....	33
3.1.10 ItemsMonitorear. ....	33
3.1.11 AdministracionNodos. ....	34
3.1.12 Referencias. ....	35
3.1.13 Atributos.....	36
3.1.14 Vistas.....	37
3.1.15 Consultas.....	38
3.2 MODELO DE CLASES DEL SISTEMA OPC UA. ....	38
4. PRINCIPIOS DE APLICACIÓN DEL ESTANDAR OPC UA, LLEVADOS A UN CASO DE ESTUDIO.....	42
4.1.1 Consideraciones del modelo de información de la empresa caso de estudio .....	44

4.2 APLICACIÓN DE LA ARQUITECTURA OPC UA AL CASO DE ESTUDIO .....	45
4.3 DESARROLLO DE LOS PASOS PARA LA APLICACIÓN DEL ESTANDAR OPC UA AL CASO DE ESTUDIO .....	46
4.3.1 Establecer los criterios para la definición de los elementos del <i>AddressSpace</i> .....	46
4.3.2 Estructurar las capas del <i>AddressSpace</i> para la aplicación [13]. .....	49
4.3.3 Aplicar los modelos de información dentro de cada una de las capas del <i>AddressSpace</i> .....	51
4.3.4 Codificación de los modelos de Información del caso de estudio en el esquema XML UA. ....	74
4.3.5 Validación de los esquemas XML de OPC UA y de los documentos instanciados en el caso de estudio. ....	77
4.3.6 Pautas técnicas para la comunicación entre los servidores OPC UA del nivel de gestión y control de procesos. ....	79
4.3.7 Migración de la tecnología OPC COM a OPC UA en las empresas de manufactura.....	90
5. CONCLUSIONES .....	95
6. BIBLIOGRAFÍA.....	97



## LISTA DE FIGURAS

Figura 1. Interoperabilidad y Conectividad entre diferentes aplicaciones .....	3
Figura 2: Arquitectura OPC UA.....	4
Figura 3. Modelo de referencias .....	7
Figura 4. Ejemplo de una Variable definida por un VariableType .....	10
Figura 5. Ejemplo de un TypeDefinitionNode Complejo.....	11
Figura 6. Objetos y sus componentes definidos por ObjectTypes.....	12
Figura 7 Utilización de la ModelingRule New.....	18
Figura 8. Conjunto de servicios <i>MonitoredItem</i> y <i>Subscription</i> . .....	21
Figura 9. Stacks OPC UA .....	23
Figura 10. Modelo de casos de uso del sistema OPC UA. ....	26
Figura 11. Pantalla actualizar seguridad .....	28
Figura 12. Pantalla de conexión al servidor. ....	29
Figura 13. Pantalla para la creación de una sesión del servidor. ....	31
Figura 14. Ventana para la actualización de .....	32
una sesión en el servidor. ....	32
Figura 15. Modelo de clases del sistema OPC UA para la implementación. ....	39
Figura 16. Diagrama de secuencia ValidarUsuario. ....	40
Figura 17. Diagrama de colaboración ValidarUsuario. ....	41
Figura 18. Validación del modelo orientado a objetos del sistema OPC UA.....	41
Figura 19. Representación de un nodo en el AddressSpace de OPC UA. ....	48
Figura 20. Especialización de la representación de Nodos .....	48
Figura 21. Estructura del AddressSpace .....	50
Figura 22. Ejemplo de la capa de tipos del modelo de materiales. ....	54
Figura 23. Interacción de la capa de instancias y la capa de tipos del AddressSpace .....	58
Figura 24. Ejemplo del modelo de equipos del caso de estudio. ....	61
Figura 25. <i>TypeDefinitionNodes</i> del modelo de definición de producto.....	66
Figura 26. Representación del AddressSpace del modelo programa de producción del caso de estudio. ....	71
Figura 27. Interfaz de validación del esquema XML del modelo de materiales en OPC UA.....	78
Figura 28. Stack de comunicaciones de OPC UA .....	79
Figura 29. Vista del diseño del esquema XML para la NodeClass <i>ObjectType</i> .....	80
Figura 30. Esquema para la codificación de objetos del AddressSpace. ....	84
Figura 31. Esquema para la codificación de referencias tipos del AddressSpace.....	86
Figura 32. Flujo de datos a través del stack de comunicaciones del caso de estudio.....	88
Figura 34 Cliente COM accediendo a un servidor UA.....	91
Figura 35. Cliente UA accediendo a un servidor DA COM.....	91
Figura 36. Proxy Cliente DA.....	93
Figura 37. Wrapper servidor .....	94

## LISTA DE TABLAS

Tabla 1	Nodo clase Base .....	15
Tabla 2	Valores del tipo de datos NodeClass .....	16
Tabla 3	Resumen preliminar del sistema OPC UA.....	25
Tabla 4	Definición del actor Usuario.....	27
Tabla 5	Definición del actor ClienteUA.....	27
Tabla 6	Definición del actor RegistroUsuario .....	27
Tabla 7	Definición del caso de uso ActualizarSeguridad. ....	28
Tabla 8	Definición del caso de uso Conectarse.....	29
Tabla 9	Definición del caso de uso Sesión.....	30
Tabla 10	Definición del caso de uso ActualizarSesion.....	31
Tabla 11	Definición del caso de uso ValidarUsuario .....	32
Tabla 12	Definición del caso de uso Subscripciones.....	33
Tabla 13	Definición del caso de uso ItemsMonitorear .....	34
Tabla 14	Definición del caso de uso AdministracionNodos .....	35
Tabla 15	Definición del caso de uso Referencias. ....	36
Tabla 16	Definición del caso de uso Atributos.....	36
Tabla 17	Definición del caso de uso Vistas. ....	37
Tabla 18	Definición del caso de uso Consultas.....	38
Tabla 19	Criterios para el nombramiento de los elementos del <i>AddressSpace</i> .....	47
Tabla 20	Formato para la definición de nodos. ....	47
Tabla 21	Definición del <i>ObjectType</i> LacteoType .....	55
Tabla 22	Definición del <i>VariableType</i> RangoType .....	57
Tabla 23	Identificación del <i>DataType</i> RangoDataType.....	57
Tabla 24	Definición del objeto UnidadTermizacionType. ....	62
Tabla 25	Definición del <i>ObjectType</i> SensoresHistoricalConfigurationType. ....	63
Tabla 26	Definición del <i>SensoresNivelHistoricalEventConfigurationType</i> .....	63
Tabla 27	Definición de la <i>ReferenceType</i> Contiene .....	64
Tabla 28	Definición del <i>VariableType</i> ParametroProductoType.....	67
Tabla 29	Definición del objeto SegmentoProductoCelulaType.....	67
Tabla 30	Definición del <i>ObjectType</i> SegmentoProductoType .....	68
Tabla 31	Definición del objeto ProgramaProduccionType.....	71
Tabla 32	Definición del objeto SolicitudProduccionType. ....	72
Tabla 33	Definición de la variable RequerimientoSegmentoType .....	73
Tabla 34	Definición de la <i>ReferenceType</i> ComponeDe.....	74
Tabla 35	Definición de la <i>ReferenceType</i> PuedeContener.....	74
Tabla 36	Esquema XML de la clase Materiales a partir del estándar B2MML. ....	75
Tabla 37	Esquema XML de Materiales mapeado a OPC UA.....	76
Tabla 38	Codificación del objeto tipo LacteoType del caso de estudio.....	81
Tabla 39	Codificación del objeto pasteurizador del caso de estudio.....	84
Tabla 40	Codificación de la referencia tipo ConformadoPor del caso de estudio .....	86
Tabla 41	Mensaje SOAP para el intercambio de información entre el cliente y el servidor. ....	89

## INTRODUCCIÓN

Desde 1996 la OPC Foundation como propuesta estándar de comunicación para acceso en tiempo real a los dispositivos de campo ha promovido la interoperabilidad en el campo de la automatización industrial [1]. Inicialmente la propuesta de la organización consistía en los estándares OPC basados en las tecnologías COM/DCOM para ser utilizadas en la comunicación de las aplicaciones de supervisión y los sistemas de control, sin embargo cada día las empresas requieren de mayor eficiencia en la integración de los procesos productivos y de negocios, por lo tanto la OPC Foundation extendió su misión creando un nuevo estándar para la integración de los sistemas de negocio con los procesos de manufactura en una arquitectura unificada a la que denomino OPC UA, con el propósito de desacoplar toda implementación de un tipo de tecnología específica lo cual cambia respecto a su propuesta inicial.

Esta nueva propuesta inicio en enero de 2004, donde la OPC Foundation conformó un grupo de trabajo para desarrollar una nueva arquitectura que llevara la propuesta de interoperabilidad a la vanguardia de la tecnología proporcionando un marco de interoperabilidad multiplataforma viable durante los próximos años. En junio de 2006 este grupo entrego las primeras especificaciones del nuevo estándar denominado OPC UA. [2]

Actualmente el estándar OPC UA es objeto de investigación y desarrollo dentro de la OPC Foundation y sus grupos de trabajo, en Latinoamérica universidades y empresas de Colombia y Brasil lideran las investigaciones del estándar ya que disponen de las especificaciones UA. Razón por la cual dentro del grupo de investigación de la universidad del Cauca este es el primer proyecto de investigación del estándar OPC UA a través del cual se busca establecer los principios para la aplicación en un caso de estudio.

Es así que con el propósito de sentar las bases de la investigación del estándar OPC UA en un caso de estudio, el presente proyecto toma como referencia las especificaciones [UA parte 1], [UA parte 3], [UA parte 5], [UA parte 6], [UA parte 7] y [UA parte 8] del estándar para la aplicación del metamodelo y los esquemas de codificación propuestos por OPC UA, de igual manera con éstas especificaciones se definen las pautas para la comunicación entre los niveles de gestión y control del caso de estudio, estableciendo los métodos, los protocolos de seguridad y de transporte que el estándar propone y que hacen parte de la organización para la interoperabilidad de servicios web y de la World Wide Web.

El caso de estudio base para la investigación está enfocado en los niveles de gestión y control de procesos, para ello se utiliza la información representativa de la empresa en cada nivel para ser aplicada con el estándar OPC UA, con el fin de brindar una base para realizar un análisis deductivo de la aplicación en cada nivel.

Debido a la importancia del estándar en la automatización industrial, a través de este proyecto se proponen los principios para la aplicación del estándar OPC UA para brindar las bases que posteriormente permitan llegar a una implementación en el grupo de investigación en Automática industrial de la universidad del Cauca.

A través del proyecto se evidenciará que OPC UA es un estándar que permite estructurar modelos de información estándares y no estándares, los cuales son accedidos a través de servicios web. OPC UA no establece flujos de información y funcionalidades en ninguno de los niveles de aplicación, por lo tanto las aplicaciones cliente obtienen los datos a través de servicios web de los sistemas OPC UA y establecen sus propias funcionalidades.

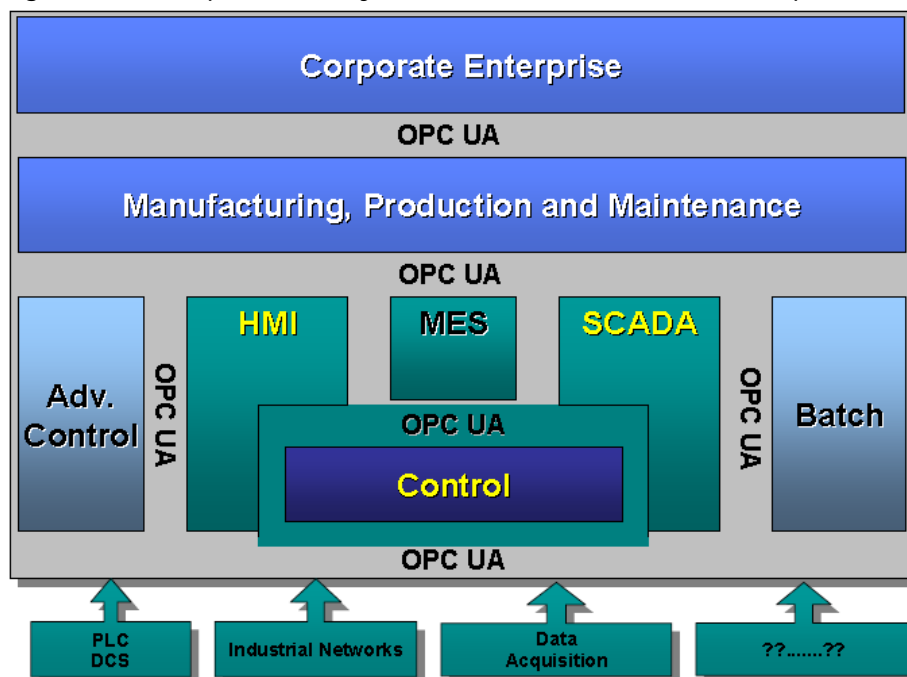
El documento consta de cuatro capítulos en los cuales se presentan los resultados de este trabajo. En el primero de ellos se introduce al lector al conocimiento del estándar a través de un marco conceptual de OPC UA necesario para comprender la arquitectura de OPC UA. En el segundo capítulo se explica con un nivel mayor de detalle la arquitectura del estándar OPC UA y todos sus componentes. Posteriormente se explican los fundamentos de la arquitectura orientada a servicios a través de la metodología propuesta por IBM en el diseño de arquitecturas SOAD (Diseño y Análisis Orientado a servicios) la cual se diseña y se valida a través de Rational Rose. Seguidamente se presentan los principios y la aplicación al caso de estudio del estándar OPC UA definiendo los elementos del AddressSpace y aplicando los esquemas de codificación propuestos por el proyecto que cumplen con la recomendación del estándar ISA 95 y OPC UA, estableciendo las pautas para una comunicación segura entre los niveles de gestión y control de procesos. Como en la mayoría de las evoluciones dadas por el avance en las tecnologías es fundamental establecer las pautas para la migración de tecnologías, en el capítulo IV el proyecto propone los pasos para llegar desde los actuales servidores OPC COM a los sistemas OPC UA, definiendo las funciones de las herramientas wrappers y proxies propuestas por la OPC Foundation.

## 1 MARCO CONCEPTUAL

### 1.1 OPC ARQUITECTURA UNIFICADA (OPC UA)

OPC UA es un estándar de plataforma independiente a través del cual varios tipos de sistemas y dispositivos pueden comunicarse enviando mensajes entre clientes y servidores sobre varios tipos de redes definidas por OPC UA [4]. La figura 1 expone su alcance en las empresas de manufactura, donde la interoperabilidad y la conectividad entre las distintas aplicaciones son la base de los requerimientos. El objetivo del estándar es obtener una arquitectura en la cual los datos de los dispositivos del nivel inferior de la planta se transfieran a través de las distintas aplicaciones hasta los niveles de gestión sobre tecnologías de comunicación y modelos de información estándar. Para ello OPC UA proporciona a través de las especificaciones el metamodelo de información UA y las tecnologías para realizar el mapeo de la información.

Figura 1. Interoperabilidad y Conectividad entre diferentes aplicaciones



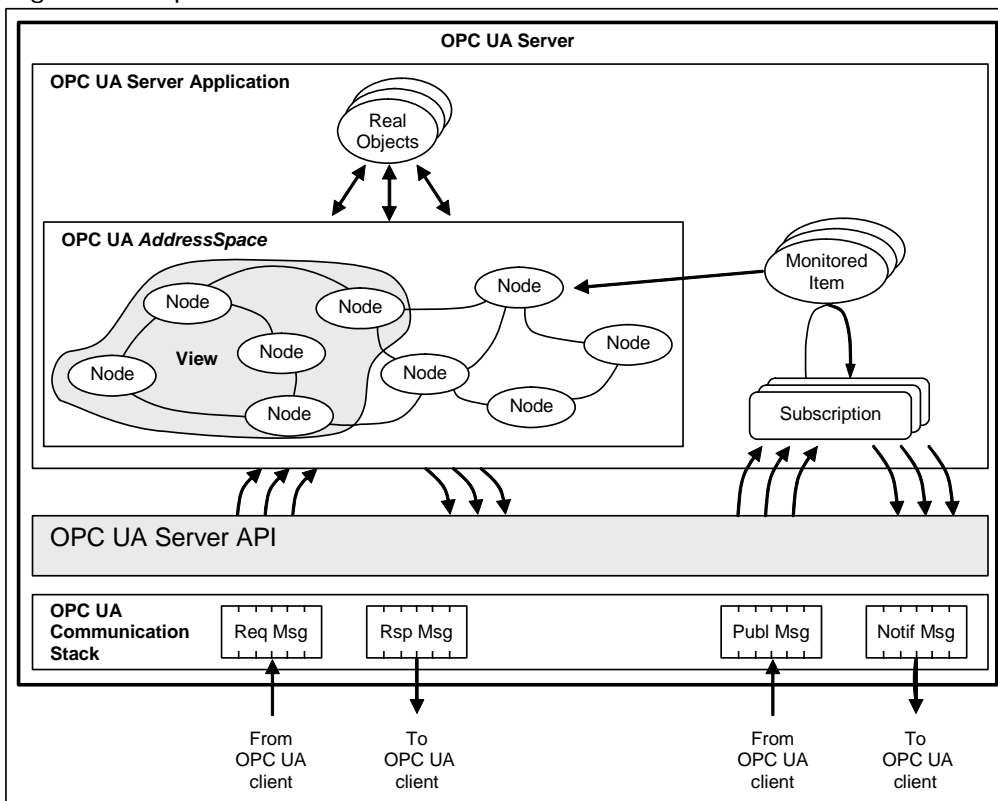
Fuente: OPC Foundation, OPC UA Specification Part 1: Concepts. Versión 1.01.01, noviembre 2007.

El metamodelo de OPC UA es la base para el diseño y la aplicación de los modelos de información de las empresas, el cual se presenta en los servidores a través del espacio de direcciones denominado AddressSpace de UA. El AddressSpace está conformado por nodos de diferentes clases y tipos, los cuales se definen a partir del metamodelo de OPC UA. Para definir los nodos, el metamodelo define un nodo básico el cual tiene un conjunto de atributos que los definen y a través de la especialización de este nodo define los demás conceptos que lo componen.

Las tecnologías para la comunicación sobre las cuales el estándar se implementa hacen parte de los estándares de internet aplicados en los servicios web. El estándar OPC UA define las propiedades y funciones de los servicios a partir de los cuales establece el conjunto de servicios abstractos de UA.

Estos servicios acceden al espacio de información del servidor a través de una API proporcionada por la OPC Foundation. La figura 2 expone la relación entre los servicios web, la API y el metamodelo de OPC UA.

Figura 2: Arquitectura OPC UA



Fuente: OPC Foundation, OPC UA Specification Part 1: Concepts. Versión 1.01.01, noviembre 2007.

Para la comunicación OPC UA puede ser mapeado en diferentes protocolos de intercambio de datos. Por el momento la OPC Foundation ha definido 2 protocolos, el protocolo TCP y el protocolo HTTP, los cuales son ampliamente reconocidos en el intercambio de datos a través de redes Ethernet (TCP) e Internet (HTTP). Para la codificación de los datos OPC UA establece dos métodos de codificación, el UA Binario y los esquemas XML. Combinando los métodos de codificación con los protocolos de transporte se puede lograr una compensación entre portabilidad y eficiencia en la comunicación entre los diferentes niveles de una red corporativa. Utilizando codificación binaria de datos para ser enviados sobre TCP se cumple con los requerimientos de tiempo real de los sistemas de control y combinando codificación XML para ser enviada a través de HTTP se solucionan los problemas de interoperabilidad en los sistemas de gestión.

Basado en una Arquitectura Orientada a Servicios para las aplicaciones industriales, OPC UA unifica los conceptos anteriores de OPC basado en COM, a partir del metamodelo y del conjunto de servicios UA permitiendo mover datos e información desde los niveles inferiores del proceso de producción hasta los sistemas de planificación de recursos empresariales.

A continuación se describen los conceptos básicos para la comprensión del estándar y la aplicación en el proyecto.

*AddressSpace* [4]. Es el conjunto de objetos e información relacionada que el servidor OPC UA coloca a disposición de los clientes. Este representa el contenido del servidor como un conjunto de nodos interconectados por referencias.

Modelo de información [4]. Es una estructura de soporte que define, caracteriza y relaciona recursos de información de un sistema dado o de un conjunto de sistemas. El núcleo del modelo de espacio de direcciones soporta la representación de modelos de información en el *AddressSpace*.

*NodeClass* [4]. Es la clase a la cual pertenece un nodo en el *AddressSpace*, el conjunto de *NodeClass* se define como el metamodelo de OPC UA.

Nodo [4]. Es el componente fundamental en la arquitectura debido a que un nodo se utiliza para representar todos los elementos del *AddressSpace*.

Atributos [4]. Los atributos son una característica básica de los nodos y son los únicos elementos de un servidor que tienen valores. Los atributos de los nodos son definidos por el estándar y están presentes según la *NodeClass* de OPC UA.

Variables [4]. Estas son utilizadas para representar valores. Dos tipos de variables son definidas, las *Properties* y las *DataVariables*, estas difieren en el tipo de dato que representan y en la posibilidad de contener otras variables.

*Properties* [4]. Las propiedades representan las características de los objetos, las *DataVariables* y otros nodos definidos por el servidor. Se diferencian de los atributos ya que las propiedades caracterizan lo que el nodo representa. Para prevenir la recursión, no está permitido que las propiedades definan propiedades para ellas mismas.

Para identificar fácilmente las propiedades, el *BrowseName* de ellas debe ser único en el contexto del nodo que las contiene.

*DataVariables* [4]. Se utilizan especialmente para representar el contenido de los objetos. Por ejemplo, un objeto está definido para que contenga una secuencia de bytes, la cual puede definirse como una *DataVariable* que es un arreglo de bytes. Las propiedades de una *DataVariable* pueden ser utilizadas para presentar el tiempo de creación y el propietario del objeto. Aquellas *DataVariables* que contengan *DataVariables* adicionales se les denomina complejas y pueden ser utilizadas para presentar más de un valor de la variable.

*VariableType* [4]. Una *VariableType* es un nodo que se utiliza para definir el tipo de una variable.

Datos complejos [4]. Son datos que están compuestos de elementos, o que se componen de más de un tipo de dato básico, su composición se asemeja a la de una estructura.

Objeto [4]. Es un nodo que representa un elemento físico o abstracto de un sistema, el cual puede ser definido como una instancia de un *ObjectType*. Los objetos son modelados utilizando el modelo de objetos de OPC UA, por lo tanto los sistemas, subsistemas y dispositivos son ejemplos de objetos.

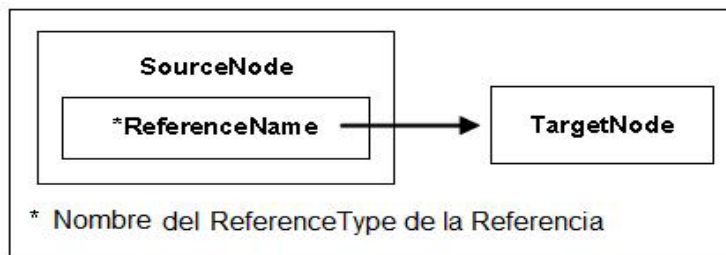


*ObjectType* [4]. Es un nodo que se utiliza para definir el tipo de un objeto.

Referencia [4]. Las referencias son una relación explícita entre los nodos, donde todas son definidas por los tipos de referencias o *ReferenceType*. Al igual que los atributos, las referencias están definidas como componentes fundamentales de los nodos, pero diferente de ellos, ya que estas son definidas como instancias de los nodos *ReferenceType*.

El nodo que contiene la referencia es denominado *SourceNode* y el nodo que es referenciado se denomina *TargetNode*. La combinación del *SourceNode*, la *ReferenceType* y el *TargetNode* son utilizados en los servicios de OPC UA para identificar las referencias, así cada nodo puede referenciar a otro nodo con la misma *ReferenceType* solo una vez. La figura 3. Ilustra el modelo de las referencias.

Figura 3. Modelo de referencias



Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model 1.01.04, noviembre 2007.

*ReferenceType* [4]. Es un nodo que se utiliza para definir el tipo y especificar la semántica de una referencia. Es así como el nombre de una *ReferenceType* identifica como el *SourceNode* está relacionado con el *TargetNode* y generalmente refleja una operación entre ellos.

*HierarchicalReferences*[4]. Las referencias jerárquicas son referencias que se utilizan para construir jerarquías en el AddressSpace. Todos los tipos de referencias jerárquicos se derivan de las *HierarchicalReferences*.

*View* [4]. Es un subconjunto específico del AddressSpace que es de interés al cliente. Las Views esconden algunos de los nodos o referencias del AddressSpace por lo tanto son utilizadas para restringir los nodos que el servidor hace visible al cliente, de esta forma las Views limitan el tamaño del AddressSpace a los servicios de solicitudes enviadas por algún cliente.

*RootNode* [4]. Es el nodo de la parte superior a través del cual se acceden a los componentes de una jerarquía.

*MonitoredItem* [4]. Es una entidad creada por el cliente en el servidor que monitorea un nodo en el *AddressSpace* y su valor real durante el tiempo de ejecución. Cuando la entidad detecta un cambio en los datos o la presencia de una alarma o un evento, genera una notificación que es transferida al cliente a través de un servicio de suscripción.

Subscripción [4]. Es un término general que describe un conjunto de nodos seleccionados por el cliente, los cuales monitorea el servidor periódicamente bajo la existencia de alguna condición, y para la cual el servidor envía notificaciones al cliente cuando la condición es detectada.

Mensaje [4]. Es la unidad de dato transferida entre un cliente y un servidor el cual representa un servicio de solicitud o de respuesta.

Servicio [4]. Es una operación que el cliente invoca para obtener información de un servidor OPC UA. Un servicio es similar al llamado de un método en un lenguaje de programación o una operación en un contrato WSDL de los servicios web.

Notificación [4]. Es un término general para datos que anuncia la detección de un evento o un cambio en el valor del atributo. Las notificaciones son enviadas en *NotificationMessages*.

*NotificationMessages* [4]. Es un mensaje publicado desde una suscripción la cual contiene una o más notificaciones.

Cliente [4]. Es una aplicación software que envía mensajes a servidores OPC UA a través de un conjunto de servicios estándar.

Stack de comunicaciones [4]. Es un conjunto de módulos software formado en capas entre la aplicación y el hardware, los cuales proporcionan las funciones para codificar, encriptar y establecer el formato de un mensaje para enviarlo, y decodificar, desencriptar y desempaquetar un mensaje recibido.

Servidor [4]. Es una aplicación software que implementa y publica información a los clientes a través de servicios.

Sesión [4]. Es una conexión lógica de larga duración entre un cliente y un servidor. Una sesión mantiene información de las llamadas de servicios entre cliente y servidor.

*SourceNode* [5]. Un *SourceNode* es un nodo que tiene una referencia a otro nodo. Por ejemplo, en la referencia "A contiene a B", "A" es el *SourceNode*.

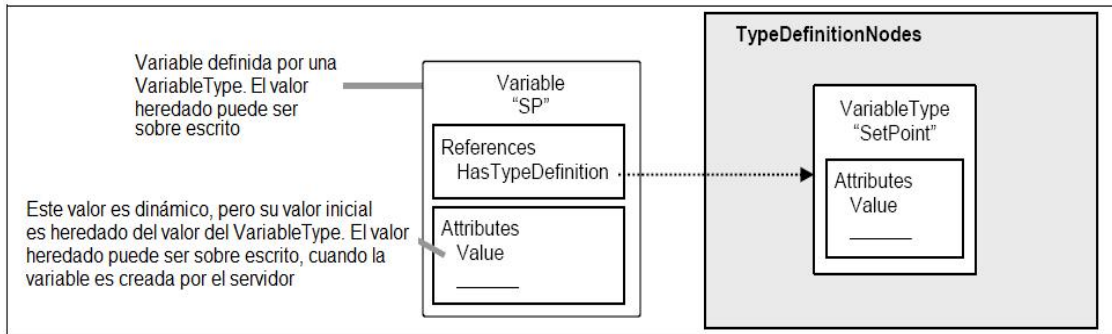
*TargetNode* [5]. Un *TargetNode* es un nodo que es referenciado por otro nodo. Por ejemplo, en la referencia "A contiene a B", "B" es el *TargetNode*.

*ModellingRule* [5]. La *ModellingRule* o regla de modelamiento de un nodo define como este será utilizado para instanciación, o cual fue la regla utilizada para crear el nodo.

*TypeDefinitionNode* [5]. Es requerido que los servidores OPC UA proporcionen los tipos de definiciones para los objetos y variables de un sistema, estas definiciones son representadas en un *TypeDefinitionNode* o capa de tipos, la cual se relaciona mediante referencias *HasTypeDefinition* a los nodos de la capa de instancias. Los Objetos y Variables heredan los atributos especificados por sus *TypeDefinitionNode*.

La figura 4 ilustra un ejemplo en el cual se describe como se utiliza la referencia *HasTypeDefinition*. En el, la *VariableType* "SetPoint" proporciona una definición común de SetPoint la cual puede ser utilizada por otros objetos, de esta manera todas aquellas *DataVariables* SetPoint que tengan esta misma definición tienen una referencia *HasTypeDefinition* a la *VariableType* SetPoint de la capa de tipos.

Figura 4. Ejemplo de una Variable definida por un VariableType



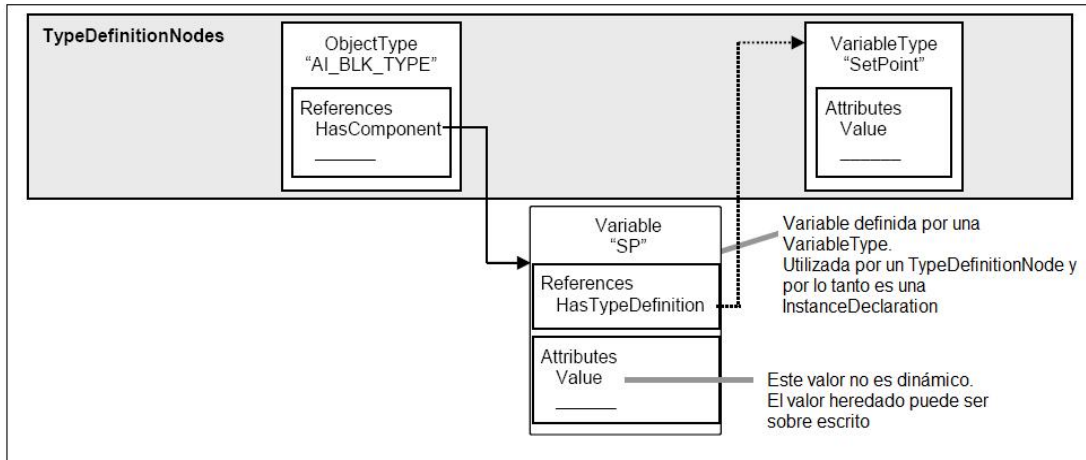
Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model. Version 1.01.04, noviembre 2007.

*TypeDefinitionNode* Complejas y sus *InstanceDeclaration* [5]. Los *TypeDefinitionNode* o nodos de la capa de tipos que son complejos definen referencias a otros nodos como parte de su definición. Las reglas de modelamiento especifican si estos nodos son instanciados cuando se crea una instancia del nodo de la capa de tipos.

Un *TypeDefinitionNode* complejo normalmente referencia instancias en lugar de otros *TypeDefinitionNode*, porque así se pueden tener diferentes nombres para varias instancias de un mismo nodo de la capa de tipos. Además esto permite al diseñador definir valores por defecto y adicionar referencias para estas instancias que son específicas del nodo de la capa de tipos.

Por ejemplo, en la figura 5 el *ObjectType* "AI\_BLK\_TYPE", representa un bloque funcional, que tiene una referencia *HasComponent* a la variable "SP" del *VariableType* "SetPoint". La variable "SP" podría adicionar propiedades que no fueron definidas por su *TypeDefinitionNode* "SetPoint", así mismo se podría definir un valor por defecto para el "SP", donde cada instancia de "AI\_BLK\_TYPE" tendría una variable "SP" cuyo valor esta fijado inicialmente.

Figura 5. Ejemplo de un TypeDefinitionNode Complejo.

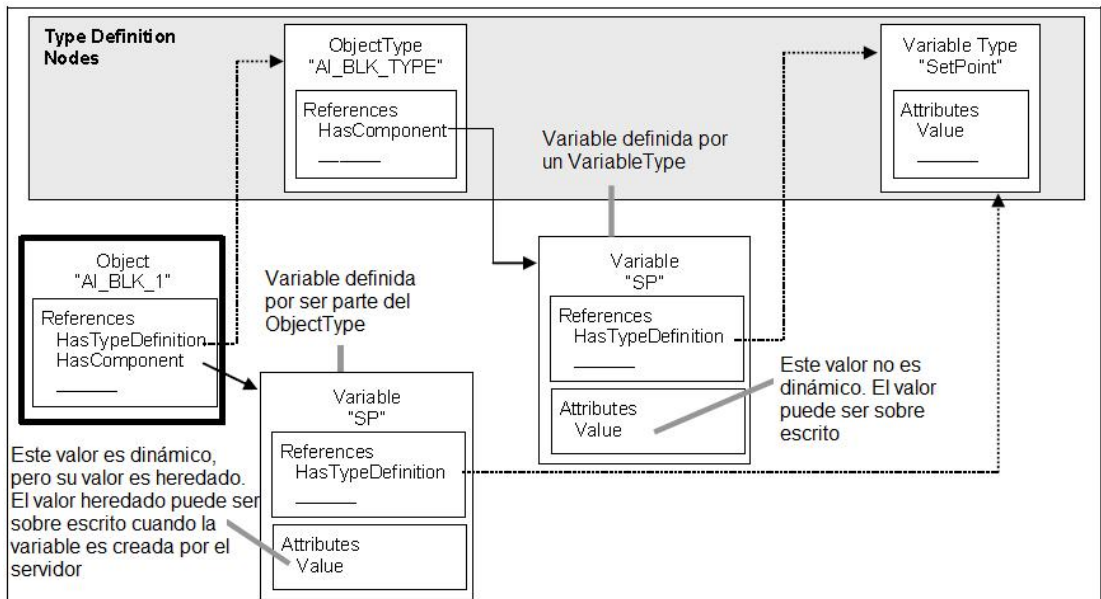


Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model 1.01.04, noviembre 2007.

Para distinguir las instancias utilizadas por los nodos de la capa de tipos de aquellas instancias que representan datos reales, las primeras instancias son llamadas *InstanceDeclaration*, las *InstanceDeclaration* y las instancias pueden ser compartidas y por lo tanto referenciadas por nodos de la capa de tipos.

Instanciación de TypeDefinitionNode complejas [5]. La instanciación de los *TypeDefinitionNode* complejos depende de las *ModellingRules* de los nodos instanciados, la figura 6 presenta una instancia del *TypeDefinitionNode* "AI\_BLK\_TYPE", donde la *ModellingRule New*, es aplicada a las variables que esta contiene, de manera que una instancia de "AI\_BLK\_TYPE", llamada AI\_BLK\_1 tiene una referencia *HasTypeDefinition* a "AI\_BLK\_TYPE". AI\_BLK\_1 también contiene una variable "SP" que tiene el mismo *BrowseName* como el de la variable "SP" utilizada por el *TypeDefinitionNode* y de este modo se refleja la estructura definida por el *TypeDefinitionNode*.

Figura 6. Objetos y sus componentes definidos por ObjectTypes



Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model 1.01.04, noviembre 2007.

Las instancias basadas en las *InstanceDeclaration* siempre deben mantener el mismo *BrowseName* de la *InstanceDeclaration* del cual estas son derivadas. Las Instancias derivadas de una *InstanceDeclaration* deben ser del mismo nodo de la capa de tipos o un subtipo de este nodo.

**DataItem** [6]. Un *DataItem* representa un enlace arbitrario a datos de automatización en línea, por ejemplo los datos que representan información válida actual de un equipo.

**AnalogItem** [6]. Los *AnalogItems* son *DataItems* que representan cantidades físicas que varían continuamente. OPC UA define un *VariableType* específico para identificar un *AnalogItem*.

**DiscreteItem** [6]. Los *DiscreteItems* son *DataItems* que representan datos que pueden tomar solo un cierto número de valores posibles. Se utilizan *VariableTypes* específicos para identificar *DiscreteItems* con dos estados o con múltiples estados.

**EngineeringUnits** [6]. Son *AnalogItems* que representan cantidades físicas de variables continuas (por ejemplo longitud, masa, tiempo, temperatura) como un entero o un valor de punto flotante. Las *EngineeringUnits* especifican las unidades de medición para estas cantidades.

## 2 ARQUITECTURA DE LOS SERVIDORES OPC UA

La arquitectura software estándar de los servidores OPC UA, consiste en un conjunto de patrones y abstracciones que proporcionan un marco de referencia para guiar el diseño y la implementación de servidores UA. En este apartado se presentan los principios del marco teórico de la arquitectura, para ello se expone el metamodelo de información de OPC UA, a partir del cual la OPC Foundation proporciona la base para el diseño de los estándares de información. Posteriormente se presenta el conjunto de reglas definidas por el estándar, las cuales deben ser consideradas para el diseño del modelo de información y luego los servicios UA que acceden a la información del servidor describiendo las tecnologías base sobre las que OPC UA propone la implementación.

### 2.1 METAMODELO DE INFORMACIÓN DE OPC UA

El metamodelo de OPC UA consiste de la definición de nodos que tienen una semántica propia dentro del AddressSpace, a continuación se presentan estos tipos de definiciones las cuales solo pueden ser actualizadas por la OPC Foundation.

La NodeClass *Base* [5] es el nodo estándar y básico a partir del cual el conjunto de nodos del metamodelo de información se derivan. Este define un conjunto de atributos que son comunes para todos los demás nodos del AddressSpace e identifican y describen cada nodo.

La NodeClass *ReferenceType* [5] define la semántica básica de las relaciones que existen entre los nodos del AddressSpace, este nodo se clasifica en relaciones jerárquicas y no jerárquicas las cuales contienen un conjunto adicionales de referencias. Los tipos de referencias establecidas por OPC UA se exponen en el anexo A.

La NodeClass *View* [5] define la semántica de un nodo que sirve de vista hacia el AddressSpace, este agrupa un conjunto de nodos con el propósito de estructurar y organizar la información del AddressSpace.

La NodeClass *Objects* [5] define la semántica de los objetos de un AddressSpace, la definición de objeto manejada por OPC UA es idéntica a la de los lenguajes de programación orientada a objetos.

La NodeClass ObjectTypes [5] define la semántica de un conjunto de nodos que representan los objetos tipos que existen en el AddressSpace. Estos objetos hacen parte de la capa de tipos del AddressSpace, la cual se diseña con los objetos tipos y las variables tipos, extraídos a partir de la semántica del modelo de información que se define y desde la cual se instancia el funcionamiento del proceso.

La NodeClass Variable [5] define la semántica de los nodos que representan variables, las cuales pueden ser representadas como propiedades o como variables complejas. Para cada uno de estos nodos define un conjunto de restricciones que especializa su función en el AddressSpace. Las propiedades se consideran las ramas de una estructura jerárquica, y definen las características de los nodos, mientras que las variables complejas representan el contenido de un objeto relacionando un conjunto de propiedades.

La NodeClass VariableType [5] define la semántica de nodos que representan las variables tipos de la capa de tipos del AddressSpace.

La NodeClass DataType [5] define la semántica de los tipos de datos del AddressSpace. Los tipos de datos definidos por OPC UA se exponen en el anexo A

La NodeClass Method [5] define la semántica de los nodos que representan los métodos o las funciones que otros tipos de nodos podrán instanciar para la ejecución.

Como es posible observar en OPC UA todos los elementos del AddressSpace son nodos, las vistas, las variables, los objetos, los tipos de datos y las referencias, que tienen una semántica y unas propiedades definidas por el estándar UA. La OPC Foundation define que solo esta organización es la responsable de definir y/o extender el metamodelo de OPC UA y por lo tanto otras organizaciones no podrán definir nuevas NodeClass.

## 2.2 ADDRESSSPACE DE OPC UA

A partir de las NodeClass presentadas anteriormente, se diseña y se implementa el AddressSpace de un servidor OPC UA. El AddressSpace es una combinación de nodos en red estructurados y organizados en la manera que el diseñador considere apropiada. Para ello el AddressSpace define la capa de tipos la cual esta conformada por nodos del tipo VariableType o nodos del tipo ObjectType. Así mismo, el AddressSpace define otra capa en la cual se encuentran las instancias de la capa de



tipos y está conformada por objetos, vistas y variables. Los tipos de datos no hacen parte del AddressSpace de manera directa por lo que estos son instanciados a través del namespace.

La función de instanciación en el AddressSpace parte de las siguientes reglas para definir las variables y los objetos desde los nodos de la capa de tipos:

- La instancia hereda completamente todo del objeto o la variable.
- Las instancias heredan los valores iniciales de los atributos, con excepción para los atributos NodeClass, NodeId y BrowseName.
- Los nodos que son referenciados desde la capa de tipos son instanciados dependiendo de la regla de modelamiento del TargetNode.

Los atributos y las reglas de modelamiento son explicados posteriormente en el literal 2.2.3.

#### 2.2.1 NodeClass Base.

Este es el nodo principal del estándar a partir del cual se define el conjunto de nodos del metamodelo de OPC UA. Sus atributos son especificados en la tabla 1 y no existe ningún tipo de referencia especificada para este nodo.

Tabla 1 Nodo clase Base

Nombre	Uso	Tipo de Dato	Descripción
Atributos			
NodeId	M (Obligatorio)	NodeId	--
NodeClass	M	NodeClass	--
BrowseName	M	QualifiedName	--
DisplayName	M	LocalizedText	--
Description	O (Opcional)	LocalizedText	--
<b>Referencias</b>		<b>Esta nodo clase no especifica referencias</b>	

Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model. Versión 1.01.04, noviembre 2007.

A continuación se explica cada uno los atributos que según el campo "uso" deben estar definidos en todos los nodos clase del AddressSpace. Los tipos de datos de los

atributos hacen parte de los datos definidos por OPC UA, que son presentados en el anexo A.

*NodeId* [5]. Este atributo es un identificador que distingue los diferentes nodos del *AddressSpace*. La estructura de los *NodeIds* depende del tipo de dato *NodeId*.

*NodeClass* [5]. Este atributo define el tipo de nodo en el *AddressSpace*. El tipo de dato del atributo es *NodeClass*, cuyos valores se presentan en la tabla 2.

Tabla 2 Valores del tipo de datos *NodeClass*

Valor
<i>DataType</i>
<i>Method</i>
<i>Object</i>
<i>ObjectType</i>
<i>ReferenceType</i>
<i>Variable</i>
<i>VariableType</i>
<i>View</i>

Fuente: OPC Foundation, OPC UA Specification Part 3: *AddressSpace Model*. Versión 1.01.04, noviembre 2007.

*BrowseName* [5]. Esta propiedad proporciona una descripción clara del nodo que facilita la exploración de nodos a través *AddressSpace*. Aunque el *NodeId* proporciona una identificación clara del nodo, esta no se puede confundir con el *BrowseName* debido a que conjunto de nodos pueden tener un mismo *BrowseName*. Este atributo tampoco puede ser utilizado como el nombre de presentación del nodo ya que para eso existe el atributo *DisplayName*. La estructura del *BrowseName* contiene el identificador del espacio de nombres "*namespace*" y una cadena de texto; el *namespace* es proporcionado con el fin de hacer el *BrowseName* único en el contexto de un conjunto de nodos.

El tipo de dato que define un *BrowseName* es la estructura *QualifiedName* expuesta en el anexo A.

*DisplayName* [5]. Este atributo contiene el nombre del nodo que se presenta a los clientes. La estructura del *DisplayName* esta definida por el tipo de dato *LocalizedText* el cual contiene una cadena de texto traducida al lugar especificado en el identificador para la localidad de este tipo de dato.

*Description* [5]. Este atributo es opcional para la NodeClass Base, en OPC UA se permite explicar el significado del nodo en un texto del tipo LocalizedText.

## 2.2.2 Reglas de diseño en el estándar OPC UA.

2.2.2.1 Creación de subtipos [5]. La creación de subtipos de los nodos *ReferenceType*, *DataType*, *ObjectTypes* y *VariableTypes*, utiliza la referencia estándar *HasSubtype* la cual define la relación entre nodos del mismo tipo. Las reglas para realizar subtipos son:

- Los subtipos no heredan el valor del *NodeId* o el *BrowseName* del tipo padre.
- La semántica de una referencia es heredada a sus subtipos y puede ser refinada.
- Si una referencia especifica algún tipo de restricciones, por ejemplo, “no se permiten ciclos”, esta restricción debe ser heredada y solo puede ser refinada, por ejemplo “no se permiten ciclos” puede ser refinada como “Debe ser una jerarquía árbol– con solo un padre”, pero no debe ser reducida, por ejemplo “se permiten ciclos”.
- Las restricciones respecto a los nodos clase que pueden ser referenciados son heredadas y solo se puede restringir más. Por ejemplo, si la referencia “A” no tiene permitido relacionar un Objeto con un ObjectType, esto igualmente es cierto para sus subtipos.
- Una referencia puede tener solo un súper tipo o no tenerlo, pero no puede tener más de uno, ya que las ReferenceType no soportan herencia múltiple.

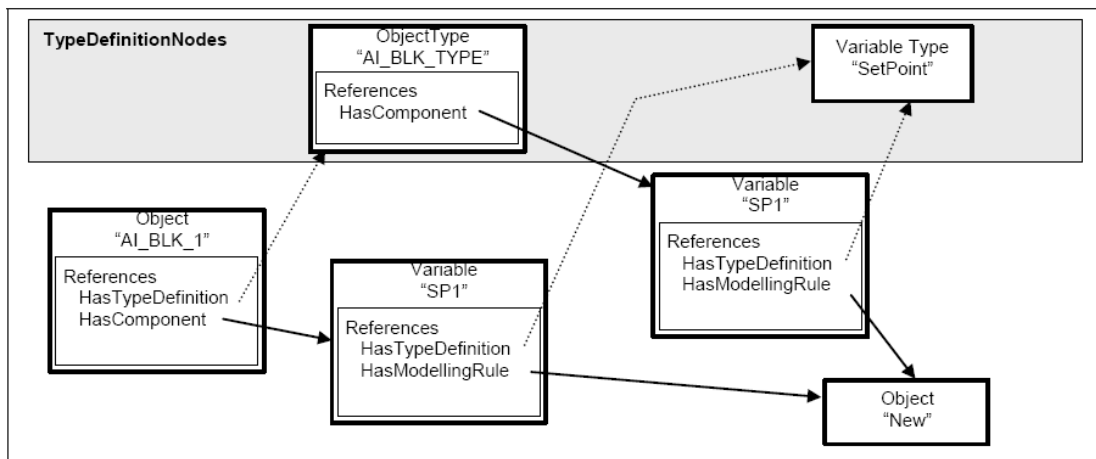
2.2.2.2 Uso de las ModelingRule o Reglas de modelamiento [5]. Las reglas de modelamiento corresponden al concepto de pertenencia de los nodos dentro del cual ellos están. Las reglas se aplican de forma específica para cada NodeClass, es así que para las variables y los métodos es necesario definir la regla de modelamiento, para los ObjectTypes y los VariableTypes este concepto no aplica, y para los objetos el concepto de pertenencia es opcional, es decir que no todos deben especificar el tipo de pertenencia a otro nodo.

Para cada método, variable y objeto se define exactamente una regla de modelamiento ya sea utilizando la referencia HasModelingRule o utilizando la regla por defecto del nodo.

A continuación se presentan las reglas de modelamiento las cuales se representan en el AddressSpace como objetos del tipo ModellingRuleType:

- *None*. Si un nodo referenciado por un tipo de nodo esta marcado con la regla None, este pertenece al tipo de nodo y no a las instancias. Por ejemplo, un nodo ObjectType que contiene una propiedad que describe los escenarios donde el objeto es utilizado y cuya regla es None, no se considera en las instancias de este tipo. Debido a que None es la regla por defecto de las variables y los métodos, tiene el mismo significado omitir la regla.
- *New*. Si un nodo referenciado por un tipo de nodo esta marcado con la regla New, este nodo será creado por cada instancia del tipo. Por ejemplo, el tipo de objeto "AI\_BLK\_TYPE" tiene una variable "SP1" cuya regla es *New*, entonces las instancias de este tipo "AI\_BLK\_1" tienen creada nuevamente "SP1" como una copia de la "SP1" del tipo del cual fue instanciado. La figura 7 ilustra este ejemplo.

Figura 7 Utilización de la ModelingRule New



Fuente: OPC Foundation, OPC UA Specification Part 3: AddressSpace Model. Versión 1.01.04, noviembre 2007.

Los nuevos nodos son creados teniendo la misma NodeClass y un NodeId diferente. Todos los otros atributos tendrán inicialmente los mismos valores; aunque el BrowseName siempre será el mismo.

- *Shared*. La regla Shared especifica que este nodo puede ser compartido por muchos otros nodos, es decir, que el nodo tiene muchas referencias Aggregates apuntando a él. Por lo tanto para estas reglas no existe una semántica definida

respecto a la propiedad de ellas, y el servidor puede o no eliminarlas aun cuando el nodo tiene referencias *Aggregates* apuntando a ella.

En algunos, por ejemplo, cuando el nodo sea una variable o método, el servidor debe suprimir el nodo si el último nodo que lo relaciona con las referencias *HasComponent* o *HasProperty* se ha eliminado, ya que las variables y los métodos nunca están solos.

- *OptionalNew*. Esta regla de modelado se utiliza cuando se quiere dejar como opcional la creación de objetos cuando se hace una instancia. Es decir cuando se haga una instancia de un tipo los objetos y variables que componen el tipo y contienen esta regla serán creados en la instancia opcionalmente.

## 2.3 SERVICIOS ABSTRACTOS DE OPC UA

Los servicios abstractos de OPC UA consisten de un conjunto de servicios web para el acceso al *AddressSpace*, a diferencia de las especificaciones OPC basadas en COM, los servicios web UA están unificados para que los clientes puedan acceder a los datos, alarmas, históricos y eventos con la seguridad requerida por el servidor. Los servicios de OPC UA están agrupados en conjuntos dependiendo de su funcionalidad.

A continuación se realiza una descripción de los servicios web de OPC UA:

### 2.3.1 Conjunto de servicios *Discovery* [7].

Este conjunto de servicios define los servicios necesarios para descubrir los puntos de acceso a los servidores y leer su configuración de seguridad. Los servicios de descubrimiento son implementados por servidores individuales dedicados a esto.

### 2.3.2 Conjunto de servicios *SecureChannel* [7].

Este conjunto define los servicios web utilizados para establecer un canal de comunicación que asegure la confidencialidad de todos los mensajes intercambiados con el servidor. Estos servicios son diferentes a los otros servicios porque no se implementan directamente en la aplicación UA, si no que son implementados por el stack de comunicación implementado para el servidor.

### 2.3.3 Conjunto de servicios *Session* [7].

Este conjunto define los servicios necesarios para establecer una conexión con la capa de aplicación de los servidores OPC UA, en el contexto de una sesión.

### 2.3.4 Conjunto de servicios *NodeManagement* [7].

Este conjunto define los servicios para agregar o borrar nodos del *AddressSpace* y las referencias entre ellos. Todos los nodos agregados continúan existiendo en el *AddressSpace* incluso si el cliente que los creó se desconecta del servidor.

### 2.3.5 Conjunto de servicios *View* [7].

Este conjunto de servicios permite que los clientes descubran nodos dentro de una vista a través de la exploración ya sea desde la parte superior a la inferior de una jerarquía, o seguir las referencias entre los nodos contenidos en ella. De esta manera es que la exploración permite a los clientes conocer la estructura de una vista.

### 2.3.6 Conjunto de servicios *Query* [7].

El conjunto de servicios *Query* o de consulta permite a los clientes seleccionar un subconjunto de nodos en el espacio de direcciones o en una vista basado en los criterios proporcionados por el cliente para establecer la selección. Los nodos y sus atributos seleccionados a través de estos servicios son llamados el conjunto de resultados, donde puede ser un conjunto "plano" de nodos, o se preserva la estructura retornando las referencias que conectan a los nodos en el conjunto de resultados.

### 2.3.7 Conjunto de servicios *Attribute* [7].

Este conjunto de servicios es utilizado para leer y escribir los atributos que hacen parte de los nodos.

### 2.3.8 Conjunto de servicios *Method* [7].

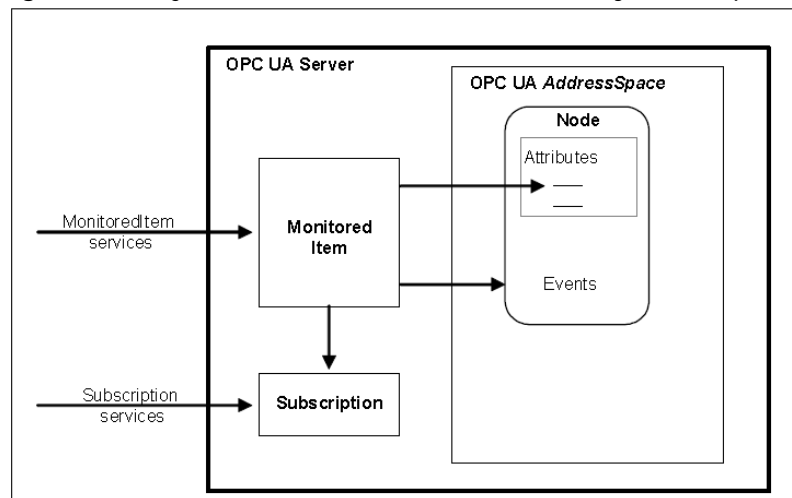
Un método es un componente de un objeto, los cuales son descubiertos a través de los servicios de búsqueda y consulta. Los métodos son invocados y entregan una respuesta después de completar el proceso satisfactoriamente o no.

### 2.3.9 Conjunto de servicios *MonitoredItem* [7].

Este conjunto de servicios es utilizado por el cliente para crear y mantener ítems monitoreados o *MonitoredItems*. Los *MonitoredItems* se encargan de monitorear variables, atributos y notificar eventos.

La forma como los servicios de monitoreo interactúan con el servidor se presenta en la figura 8, los cuales están asociados con el conjunto de servicios Subscription a través del cual se realiza la notificación de un evento al cliente.

Figura 8. Conjunto de servicios *MonitoredItem* y *Subscription*.



Fuente: OPC Foundation, *OPC UA Specification Part 4: Services. Version 1.01.17*, noviembre 2007.

### 2.3.10 Conjunto de servicios *Subscription* [7].

El conjunto de servicios *Subscription* es utilizado por el cliente para crear y mantener suscripciones en el servidor. Las suscripciones publican periódicamente mensajes de notificación de los ítems monitoreados, los cuales han sido asignados a la suscripción. El formato del mensaje es específico del tipo ítem monitoreado; variables, atributos o eventos.

## 2.4 TECNOLOGÍAS BASE PARA LA COMUNICACIÓN EN OPC UA

La historia de la automatización demuestra que no existen protocolos o estándares únicos en el proceso de integración de empresas ya que son sistemas conformados por

subsistemas que interactúan para proveer un producto o un servicio [8]. Para desarrollar sus actividades un sistema se apoya en las tecnologías para la comunicación y presentación de la información las cuales varían considerablemente de un sistema o subsistema a otro.

Con el propósito de posicionarse en los proyectos de integración empresarial, desde el nivel de procesos hasta el nivel de negocios, a través de una arquitectura software interoperable y escalable al desarrollo de las nuevas tecnologías. OPC UA propone que cada servidor establecido en los niveles jerárquicos de una empresa adopte un perfil que favorezca las características técnicas requeridas por las empresas [9]. Por lo tanto a continuación se describen los perfiles y las características técnicas de las tecnologías base para la implementación.

#### 2.4.1 Perfil de los servidores OPC UA.

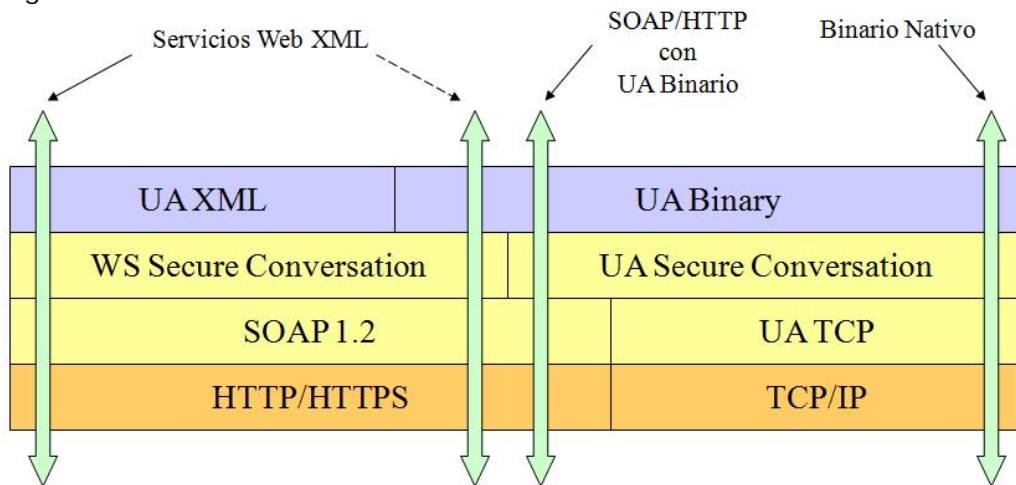
El estándar OPC UA describe un gran número de servicios, tecnologías base y un modelo de información unificado que está definido dentro de las características técnicas de un servidor o un cliente UA. Debido a que cada una de estas características corresponde a las tecnologías y los requerimientos del momento, OPC UA unifica características específicas que grupalmente sean posibles de verificar y renovar las cuales ha denominado Unidades de conformidad.

El perfil de un servidor se define a partir del agrupamiento de las unidades de conformidad definidas para él, es así que definido un modelo de información específico, un método de codificación, un protocolo de seguridad y uno de transporte, se podría especificar el perfil del servidor [9]. A continuación se describe el Stack de comunicaciones de OPC UA el cual está definido por las tres últimas unidades del perfil nombradas anteriormente, y que brindan el soporte para las interacciones cliente servidor. Los perfiles del stack son relacionados a continuación y presentados en la figura 9:

- SOAP/HTTP + XML + WS-Security
- SOAP/HTTP + UA Binary + WS-Security
- SOAP/HTTP + UA Binary + UA Security
- UA TCP + UA Binary + UA Security



Figura 9. Stacks OPC UA



Fuente: UA Devcon 2007: Information Model and Services 2007.ppt, noviembre 2007.

El stack cumple tres funciones para proporcionar una comunicación acorde a la especificación.

Codificar los datos [9]. Actualmente los métodos de codificación propuestos se realizan a través de esquemas binarios o esquemas XML proporcionados por el estándar.

Modelos de seguridad de la implementación del stack [9]. Actualmente el estándar propone dos modelos, el WS-Security el cual hace parte de la especificación para servicios web WS-\* y el UA Binary el cual es una versión binaria de WS-Secure Conversation los cuales especifican como implementar los algoritmos de seguridad en los mensajes UA.

Protocolo de transporte [9]. Los protocolos pueden ser HTTP/SOAP o TCP UA.

#### 2.4.2 Stack de Servicios web XML [10].

Este stack se utiliza principalmente para dar soporte a herramientas ERP, MES y otras aplicaciones de negocio a través de la codificación y decodificación de mensajes de Solicitud-Respuesta de los esquemas XML o binarios definidos por el estándar OPC UA. La especificación de seguridad WS-Secure Conversation definida para el stack establece la forma de compartir información a través del estándar WS Trust y la manera de asegurar el intercambio de mensajes a través de la especificación WS Security los cuales hacen parte de las especificaciones WS-\* definidos por la organización para la interoperabilidad de servicios Web WS-I (Web Services Interoperability Organization).

#### 2.4.3 Stack Binario nativo con SOAP [10].

Este stack combina la codificación de datos en esquemas binarios con el protocolo de mensajería SOAP sobre HTTP/HTTPS, y presenta una característica diferente al stack de Servicios XML respecto al protocolo de seguridad, ya que este implementa el UA Secure Conversation el cual es una versión binaria de la especificación WS-Secure Conversation. El UA Secure Conversation tiene la función de asegurar una comunicación sobre protocolos de transporte que no soporten SOAP o XML.

#### 2.4.4 Stack Binario Nativo [10].

Este stack combina la codificación de datos UA binario con el protocolo de transporte UA TCP, y el protocolo de seguridad UA Secure Conversation, el tipo de codificación hace que el tamaño de los mensajes sea más pequeño, por lo tanto es el stack que presenta un mejor desempeño respecto a la velocidad de transmisión de los datos.

Debido a que los puertos a utilizar por las aplicaciones UA sean estándar, la OPC Foundation en conjunto con la IANA (Internet Assigned Numbers Authority) designaron el puerto 4840 para que las aplicaciones UA se comuniquen sobre TCP y el puerto 4843 para comunicaciones sobre TCP con mecanismos SSL/TLS de seguridad

### 3 ANÁLISIS Y DISEÑO PARA LA IMPLEMENTACIÓN DE UN SERVIDOR OPC UA BASADO EN LA METODOLOGÍA ORIENTADA A OBJETOS.

A través del análisis y el diseño del sistema OPC UA se pretende obtener los objetos y los flujos de información que describen la dinámica del sistema, con el objetivo de proponer las bases para la implementación del estándar en futuros proyectos. Para el desarrollo de este capítulo se parte del análisis a los requerimientos funcionales del sistema el cual describe su propósito a través de los diferentes escenarios descritos en los casos de uso. Posteriormente se abstraen los conceptos más significativos de la definición de los casos de uso con los cuales se realiza el análisis que permita concluir las unidades fundamentales para la persistencia del sistema. A partir de la investigación realizada al estándar OPC UA se realizan los diagramas de secuencia con los cuales se modela la secuencia de información con las instancias de las clases identificadas en el sistema. Finalmente la herramienta Sybase PowerDesigner utilizada para la implementación de los modelos y su validación crea automáticamente los diagramas de colaboración a partir de los diagramas propuestos, con los cuales se modela los flujos y los tipos de información que se intercambian entre los objetos.

#### 3.1 DESCRIPCIÓN DEL SISTEMA.

La descripción que se presenta en la tabla 3 del servidor OPC UA es un resumen preliminar de las necesidades que sirven de partida para comprender los requerimientos que describen las funcionalidades que un servidor OPC UA debe brindar a los clientes UA.

Tabla 3. Resumen preliminar del sistema OPC UA

<p>Un servidor OPC UA es un sistema que permite que los clientes accedan a datos y eventos de un proceso o una aplicación a través de un espacio de información, el cual está constituido por nodos que representan la información del sistema.</p> <p>Cada cliente se debe conectar con el servidor en el contexto de una sesión en la cual el usuario registre sus datos. Cuando el usuario haya creado la sesión y el sistema haya validado los datos, el sistema debe presentar el espacio de información como la estructura de un explorador que a través de carpetas describe los diferentes tipos de nodos.</p> <p>Las diferentes actividades relacionadas con el flujo de trabajo son:</p> <p>Es importante que el cliente pueda agregar y borrar nodos del AddressSpace y las referencias entre ellos. Todos los nodos que el cliente agregue deben permanecer en el AddressSpace después que el cliente cierre la sesión.</p> <p>El cliente puede realizar consultas o generar reportes, para ello solo debe definir las instancias de la capa de tipos que proporcionan los atributos incluidos en la consulta, el resultado debe ser un arreglo cuyos valores se entregan en el mismo orden que la consulta fue realizada.</p> <p>El cliente necesita acceder en forma de lectura y escritura a los atributos de uno o más nodos del</p>
--

AddressSpace. Si el valor del atributo es un arreglo este debe poderse leer completamente o en partes. Dado que existe un conjunto de nodos que representan métodos de los objetos del AddressSpace, es necesario que estos se puedan consultar e invocar.

Es importante que el servidor permita subscribir datos que serán monitoreados, estos datos o ítems pueden ser atributos de un nodo e identifican la subscripción a utilizar para enviar las notificaciones al cliente en forma de estructuras de datos que describen el cambio de los datos o de un evento al cliente. En el momento de definir los ítems monitoreados, el cliente necesita definir el intervalo de muestreo y los criterios para que el servidor genere la notificación, también debe ser posible modificar estos parámetros sin necesidad de borrarlos de la subscripción y si una subscripción es borrada todos los ítems asociados son borrados.

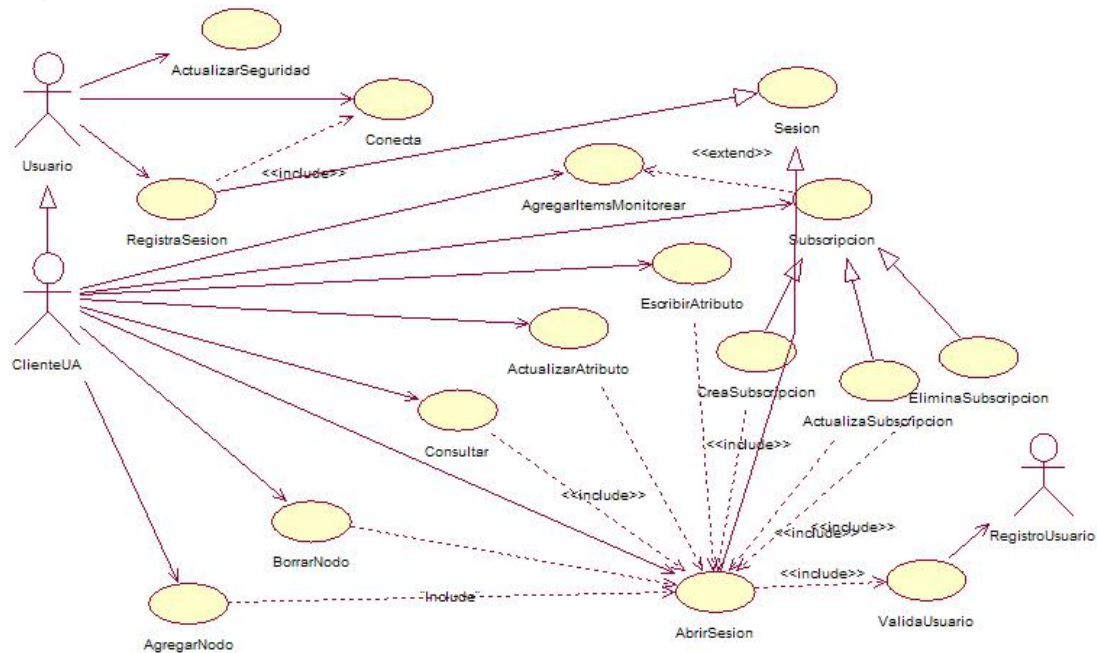
Es necesario que el cliente tenga la capacidad de crear subscripciones para que el servidor reporte las notificaciones al cliente. Estas se pueden crear, modificar, borrar de una sesión a otra, asociando el conjunto de ítems monitoreados. Las subscripciones tienen un intervalo de publicación de notificaciones.

La sesión debe terminar cuando el tiempo de expiración establecido en el momento de crear la conexión se haya cumplido o cuando el cliente explícitamente lo solicite.

Fuente: Elaboración propia, abril 2008.

El modelo de casos de uso identificado a partir de los requerimientos del sistema OPC UA se presenta en la figura 10.

Figura 10. Modelo de casos de uso del sistema OPC UA.



Fuente: Elaboración propia, abril 2008.

A continuación se describe un grupo de casos de uso identificados en el modelo de usos del sistema OPC UA. Los actores identificados corresponden al usuario, el cliente del sistema UA y las bases de datos que son componentes externos al sistema.

### 3.1.1 Usuario.

Este es un actor secundario que representa aquellos visitantes del sistema que acceden a la aplicación y que aún no han creado una sesión en el servidor. Cuando el usuario se registra este se convierte en un cliente UA. La definición del actor se presenta en la tabla 4.

Tabla 4. Definición del actor Usuario.

Actor	Usuario
Caso de uso	ActualizarSeguridad, Conectarse, RegistrarSesion,
Tipo	Secundario
Descripción	Este es el genérico del sistema que se conecta al servidor pero que aún no ha creado una sesión en el sistema.

Fuente: Elaboración propia, abril 2008.

### 3.1.2 ClienteUA.

Es el actor principal del sistema el cual representa cualquier cliente genérico UA registrado y con la capacidad de trabajar con el servidor UA. La definición del actor se presenta en la tabla 5.

Tabla 5. Definición del actor ClienteUA

Actor	ClienteUA
Caso de uso	AbrirSesion, BorrarNodos, Suscripciones, ItemsMonitorear, Consultas, AdministracionNodos, Referencias, Atributos, Vistas
Tipo	Primario
Descripción	Este es el actor principal del sistema.

Fuente: Elaboración propia, abril 2008.

### 3.1.3 RegistroUsuario.

Es un actor secundario que representa la base de datos en la cual se almacena la información de los usuarios del sistema. La definición del actor se presenta en la tabla 6.

Tabla 6. Definición del actor RegistroUsuario

Actor	RegistroUsuario
Caso de uso	ValidarUsuario
Tipo	Secundario
Descripción	Este es un actor secundario y representa la base de datos donde se guarda la información relacionada con la sesión de los clientes UA.

Fuente: Elaboración propia, abril 2008.

Como apoyo a los casos de uso, se presentan las diferentes pantallas diseñadas, y que están asociadas con los casos de uso del sistema OPC UA.

### 3.1.4 ActualizarSeguridad.

Este caso de uso está relacionado con la interfaz V-1 representada en la figura 11, para solicitar al cliente la actualización de los parámetros de seguridad. La definición del caso de uso se presenta en la tabla 7.

Tabla 7. Definición del caso de uso ActualizarSeguridad.

Id.	CU-B001
Caso de uso	ActualizarSeguridad
Actores	UsuarioUA
Tipo	Básico
Propósito	Definir el modo y las políticas de seguridad cuando el cliente accede al servidor.
Resumen	Los usuarios inician este caso de uso para definir los parámetros de seguridad de la conexión al servidor.
Precondiciones	Ninguna
Flujo principal	Se presenta al usuario la ventana (V-1). El usuario selecciona el modo y las políticas de seguridad. Solo si el modo de seguridad es "Signed and Encrypted". El usuario deberá definir la política de seguridad. Si el usuario selecciona "None". No es posible establecer una conexión con el servidor. Si el usuario selecciona "Signed". No es necesario establecer la política de seguridad ya que solo se envía en la conexión el certificado de seguridad del cliente.
Subflujos	Presenta la ventana de conexión al servidor (V-2).
Excepciones	Si el usuario no actualiza las políticas de seguridad después de cinco minutos, la aplicación cancela la operación de actualización.

Fuente: Elaboración propia, abril 2008.

Figura 11. Pantalla actualizar seguridad

Fuente: Elaboración propia, abril 2008.

### 3.1.5 Conecta.

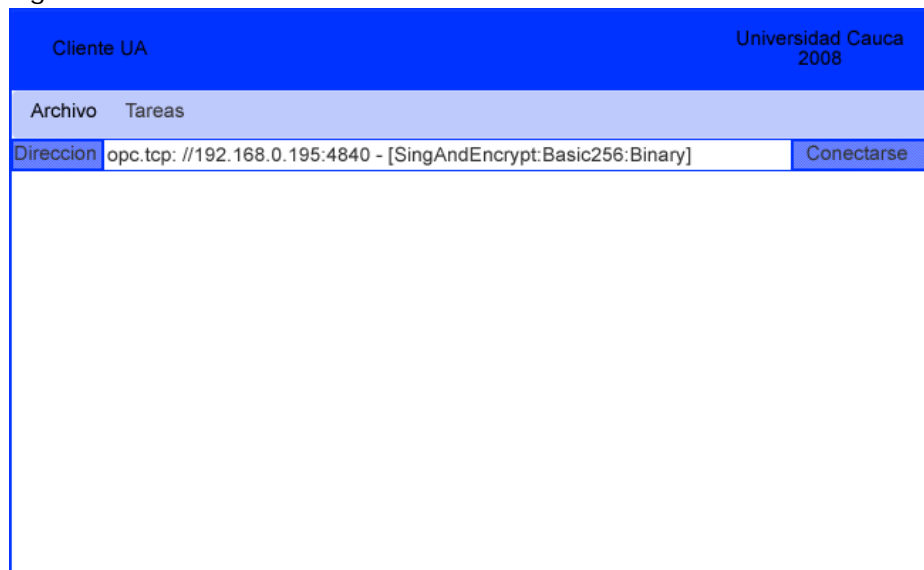
Este caso de uso está relacionado con la interfaz V-2 de la figura 12, y representa la operación de un usuario para establecer una conexión con el servidor. La definición del caso de uso se presenta en la tabla 8.

Tabla 8. Definición del caso de uso Conectarse.

Id.	CU-B002
Caso de uso	Conecta
Actores	Usuario
Tipo	Básico.
Propósito	Conectarse con la aplicación principal del servidor UA.
Resumen	Indicar el endpoint de conexión al servidor que se desea conectar.
Precondiciones	Haber establecido los parámetros de seguridad.
Flujo principal	Se presenta al usuario la ventana V-2. El usuario ingresa la dirección o endpoint de acceso al servidor. Se envía la solicitud al servidor con los parámetros de seguridad definidos.
Subflujos	Con la conexión al servidor, la aplicación presenta la ventana V-3 para: Registrar el nombre de usuario y el password para acceder a la sesión del servidor. O crear una sesión en el servidor, para ello se debe indicar los parámetros del registro de una sesión.
Excepciones	Si la conexión no se realiza, el servidor retorna un mensaje Bad_ConnectionRejected al usuario.

Fuente: Elaboración propia, abril 2008.

Figura 12. Pantalla de conexión al servidor.



Fuente: Elaboración propia, abril 2008.

### 3.1.6 Sesión.

Este caso de uso está vinculado con la ventana V-3 de la figura 13, a través de ella los clientes tienen la posibilidad de abrir o modificar una sesión o aquellos usuarios que no han creado una sesión, pueden ingresar los parámetros para crearla. La definición del caso de uso se presenta en la tabla 9. Sesión

Tabla 9. Definición del caso de uso Sesión.

Id.	CU-B003
Caso de uso	Sesion
Actores	ClienteUA, Usuario. RegistroUsuario
Tipo	Básico.
Propósito	Permitir al cliente abrir o modificar una sesión en el servidor. A los usuarios que no tienen una sesión creada en el servidor les permite crear una.
Resumen	Permitir abrir, crear o modificar una sesión de un cliente o un usuario en el sistema OPC UA.
Precondiciones	Que exista una conexión con el servidor.
Flujo principal	Se presenta la ventana V-3. El usuario tiene tres opciones: Ingresar los parámetros para crear una sesión en el servidor. Actualizar los parámetros de la sesión en el servidor. Ingresar el nombre de usuario y la contraseña para acceder a la sesión del servidor.
Subflujos	Si el usuario selecciona Actualizar sesión, se presenta la ventana V-4 solicitando nuevamente los parámetros de la sesión. Si el usuario ingresa nombre de usuario y contraseña, el sistema abre la sesión cuando la validación del usuario con la base de datos del registro de usuario es positiva. Si el usuario ingresa los parámetros para crear una sesión, el sistema crea el registro de usuario en la base de datos.
Excepciones	Si el usuario cierra la ventana para el ingreso o la creación de una sesión, el sistema automáticamente cancela la conexión con el servidor.

Fuente: Elaboración propia, abril 2008.



Figura 13. Pantalla para la creación de una sesión del servidor.

Fuente: Elaboración propia, abril 2008.

### 3.1.7 ActualizarSesion.

Este caso de uso está vinculado con la ventana V-4 de la figura 14, a través de ella los clientes actualizan sus datos para acceder a su sesión en el servidor. La definición del caso de uso se presenta en la tabla 10.

Tabla 10. Definición del caso de uso ActualizarSesion.

Id.	CU-B004
Caso de uso	ActualizarSesion
Actores	ClienteUA, RegistroUsuario
Tipo	Básico.
Propósito	Actualizar una sesión en el servidor.
Resumen	Actualiza una sesión en el servidor, para ello se indica el nombre de la sesión, el usuario y el password.
Precondiciones	Existir una conexión entre el cliente y el servidor. Haber seleccionado la opción "Actualizar sesión"
Flujo principal	Se presenta la ventana V-4. El cliente ingresa los anteriores y los nuevos parámetros de una sesión.
Subflujos	El sistema automáticamente registra los cambios en el actor registro de usuario.
Excepciones	Si el usuario no digita los datos de la sesión inicial de manera correcta, el sistema informa que los datos están mal, y permite que el usuario lo intente hasta por cinco veces. Posteriormente se cierra la conexión automáticamente.

Fuente: Elaboración propia, abril 2008.

Figura 14. Ventana para la actualización de una sesión en el servidor.

The image shows a graphical user interface window titled "Actualizar Sesión". The window has a blue header bar with the title in white text. Below the header, there are four text input fields, each with a label to its left: "Nombre Sesión", "Nombre Usuario", "Password", and "Repetir Password". At the bottom of the window, there are two buttons: "OK" on the left and "Cancelar" on the right. The entire window is enclosed in a blue border.

Fuente: Elaboración propia, abril 2008.

### 3.1.8 ValidarUsuario.

Este caso de uso está vinculado con la pantalla V-3, este caso de uso es una inclusión al caso de uso AbrirSesion y su resultado establece si la sesión se presenta al cliente. La definición del caso de uso se presenta en la tabla 11

Tabla 11. Definición del caso de uso ValidarUsuario

Id.	CU-B005
Caso de uso	ValidarUsuario
Actores	ClienteUA
Tipo	Inclusión.
Propósito	Validar un usuario ya registrado para el uso del sistema OPC UA.
Resumen	Validar la sesión del usuario mediante un login y un password a verificarse en el sistema.
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso AbrirSesión.
Flujo principal	Se validan los datos de usuario ingresados en el caso de uso AbrirSesion con el actor RegistroUsuario.
Subflujos	Si el resultado de la validación es correcto, el sistema presenta la ventana principal del espacio de información del servidor.
Excepciones	Si el login y el password no se valida correctamente. Se solicita al usuario volver a registrarse, después de cinco intentos se cierra la conexión.

Fuente: Elaboración propia, abril 2008.

### 3.1.9 Suscripciones.

Este caso de uso consiste en la posibilidad de crear suscripciones que notifican al cliente el suceso de un evento o una condición específica. La definición del caso de estudio se presenta en la tabla 12.

Tabla 12. Definición del caso de uso Suscripciones

Id.	CU-B006
Caso de uso	Suscripciones
Actores	ClienteUA
Tipo	Básico
Propósito	Crear, modificar o borrar suscripciones del servidor.
Resumen	Este caso de uso se inicia por el usuario, para que este reciba notificaciones generadas por los ítems monitoreados.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	<p>El cliente selecciona la opción "crear suscripción"</p> <p>El sistema presenta una ventana en la cual el usuario debe ingresar los parámetros:</p> <ul style="list-style-type: none"> <li>Nombre de la suscripción</li> <li>Intervalo de publicación</li> <li>Prioridad</li> <li>Habilitar publicación</li> </ul> <p>El cliente guarda los parámetros de una suscripción.</p> <p>Si el usuario desea, puede modificar los parámetros de una suscripción para ello el cliente selecciona la opción "Modificar Suscripción".</p> <p>El sistema presenta una ventana con los parámetros actuales de una suscripción.</p> <p>El cliente modifica los nuevos parámetros de la suscripción.</p> <p>El cliente acepta los cambios de la suscripción.</p> <p>El cliente selecciona la opción "Borrar Suscripción"</p> <p>El sistema solicita se confirme la acción.</p> <p>Si el usuario confirma la acción, el sistema automáticamente elimina los ítems monitoreados a través de la suscripción.</p>
Subflujos	<p>Si la creación de la suscripción es exitosa, el sistema presenta una ventana a través de la cual se agregan los ítems a una suscripción.</p> <p>Si la opción "Modificar suscripción" es exitosa el sistema informa la actualización correcta de los parámetros.</p> <p>Si la opción "Borrar Suscripción" es exitosa, el sistema informa la lista de ítems monitoreados que han sido eliminados de la suscripción.</p>
Excepciones	Si alguna de las opciones no se realiza satisfactoriamente, el sistema informa el error y regresa al menú inicial.

Fuente: Elaboración propia, abril 2008.

### 3.1.10 ItemsMonitorear.

Este caso de uso asocia los ítems a una suscripción, estos ítems se pueden agregar o borrar de una suscripción. La definición del caso de uso se presenta en la tabla 13.

Tabla 13. Definición del caso de uso ItemsMonitorear

Id.	CU-B007
Caso de uso	ItemsMonitorear
Actores	ClienteUA
Tipo	Básico
Propósito	Agregar o borrar los ítems a las suscripciones
Resumen	Subscribir los objetos del AddressSpace que notifiquen al cliente la ocurrencia de un evento o un suceso.
Precondiciones	Debe existir una suscripción a la cual se puedan agregar los ítems monitoreados.
Flujo principal	<p>Para agregar ítems a una suscripción se puede hacer de dos maneras:                      Se selecciona el objeto.                      El cliente indica "Agregar a Suscripción"                      El sistema presenta la lista de suscripciones creadas por el cliente. La lista contiene los siguientes parámetros:                      Nodo                      NodeId                      Intervalo de muestreo.                      El usuario selecciona la suscripción a la cual desea agregar el ítem.                      El sistema informa si el ítem fue agregado exitosamente a la suscripción.                      Otra opción para agregar ítems a una suscripción es:                      Se selecciona la suscripción.                      Se indica "agregar ítems a monitorear"                      El sistema presenta la lista de objetos del espacio de información estructurada en forma de árbol.                      El cliente selecciona los ítems a monitorear.                      El sistema automáticamente agrega los ítems a la suscripción.                      El sistema presenta una ventana con los parámetros que describen el estado de los ítems monitoreados.                      Nodo                      NodeId                      Intervalo de muestreo.</p>
Subflujos	Cada vez que se agreguen ítems a una suscripción, el sistema presenta la lista de ítems que hacen parte de ella.
Excepciones	Si el cliente no ha creado alguna suscripción, el sistema no permite que el cliente realice esta acción.

Fuente: Elaboración propia, abril 2008.

### 3.1.11 AdministracionNodos.

Este caso de uso representa un conjunto de funciones que se realizan sobre la unidad fundamental del espacio de información del sistema UA, esto permite que el cliente agregue o borre nodos del sistema los cuales permanecen incluso después de cerrar la sesión con el servidor. La definición formal del caso de uso se presenta en la tabla 14

Tabla 14. Definición del caso de uso AdministracionNodos

Id.	CU-B008
Caso de uso	AdministracionNodos
Actores	ClienteUA
Tipo	Básico.
Propósito	Este conjunto de funciones le permite al cliente agregar o borrar uno o más nodos del AddressSpace.
Resumen	El usuario inicia este caso de uso. Estas funciones permiten al cliente agregar o borrar uno o más nodos del AddressSpace. Asegurando que cada uno esté relacionado por una referencia jerárquica, de tal manera que se asegure la conectividad de los nodos.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	<p>El cliente Selecciona la función "Agregar Nodos"</p> <p>El sistema solicita se definan los parámetros:  NodeId  NodeClass  BrowseName  DisplayName  Description</p> <p>El usuario define los parámetros.</p> <p>Si el nodo hace parte de la definición de la capa de tipos, el usuario debe definir la relación apropiada que debe existir entre los nodos.</p> <p>Si el nodo hace parte de la capa de instancias, el sistema automáticamente solicita el tipo de definición del nodo.</p> <p>El cliente selecciona la función "Eliminar nodos"</p> <p>El sistema solicita se defina el parámetro:  DeleteTargetReference</p> <p>El sistema elimina las referencias del nodo según la definición del parámetro.</p> <p>El sistema informa al cliente el resultado de la operación.</p>
Subflujos	<p>Para los nodos agregados, el sistema automáticamente brinda la posibilidad de agregar referencias dependiendo del tipo del nodo.</p> <p>Si el nodo eliminado es propietario de otro nodo, este también es eliminado.</p> <p>Cuando un nodo es borrado y este hace parte de una vista, la propiedad ViewVersion de la vista cambia.</p> <p>Cuando un nodo monitoreado es borrado, una notificación al cliente es enviada.</p>
Excepciones	Si los atributos básicos de los nodos no se definen, el sistema informa al cliente que el nodo se pudo crear.

Fuente: Elaboración propia, abril 2008.

### 3.1.12 Referencias.

Este caso de uso representa un conjunto de funciones especializadas en la administración del tipo de nodos que interconectan las variables y los objetos del sistema, a través de ellas el cliente UA puede agregar o borrar una o mas referencias. La definición del caso de uso se presenta en la tabla 15.

Tabla 15. Definición del caso de uso Referencias.

Id.	CU-B009
Caso de uso	Referencias
Actores	ClienteUA
Tipo	Básico
Propósito	Este conjunto de funciones es utilizado para agregar o borrar una o más referencias a uno o más nodos.
Resumen	El usuario inicia este caso de uso. La funcionalidad le permite agregar o borrar una o más referencias.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	Para agregar una referencia el sistema realiza las siguientes actividades: El usuario selecciona el nodo. El usuario selecciona "agregar referencia" al nodo. El usuario selecciona el tipo de referencia. Solo si la referencia es soportada por el nodo, el sistema solicita el TargetNode de la referencia. El cliente define el NodeId de la referencia. Para eliminar una referencia el sistema solicita: El identificador del nodo que contiene la referencia a borrar El identificador de la referencia que se desea borrar El identificador del nodo al que apunta la referencia.
Subflujos	Las referencias que hagan parte de una vista y sean borradas del espacio de información afectan la propiedad ViewVersion de la vista.
Excepciones	Si no se indican los parámetros para borrar o agregar una referencia, se debe informar al cliente los parámetros que hacen falta. Si la referencia no es soportada por el nodo, el sistema informa al cliente que no es posible agregar la referencia al nodo.

Fuente: Elaboración propia, abril 2008.

### 3.1.13 Atributos.

Este caso de uso representa un conjunto de funciones encargadas de acceder y modificar los atributos de los nodos del espacio de información. La definición del caso de uso se presenta en la tabla 16.

Tabla 16. Definición del caso de uso Atributos.

Id.	CU-B010
Caso de uso	Atributos
Actores	ClienteUA
Tipo	Básico.
Propósito	Esta es una funcionalidad que le permite al cliente acceder a los atributos que son parte de los nodos.
Resumen	El usuario inicia este caso de uso. Este caso de uso le permite al cliente leer datos actuales o datos históricos, escribir en los valores de los atributos y actualizar valores históricos.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	El cliente selecciona el nodo. El cliente indica "Atributos"

	<p>El sistema presenta los atributos del nodo.  El cliente tiene la posibilidad de actualizar los valores de los atributos del nodo.  El cliente actualiza los parámetros del nodo.  El sistema solicita antes de cerrar la ventana, se confirme si se desean guardar los cambios  El cliente confirma si desea actualizar el valor de los atributos.</p>
Subflujos	<p>Si el cliente actualiza los atributos, el sistema indica el resultado del proceso.  Si el atributo es un ítem monitoreado, el sistema notifica al cliente el cambio de valor.</p>
Excepciones	No se pueden cambiar valores si el atributo es solo lectura.

Fuente: Elaboración propia, abril 2008.

### 3.1.14 Vistas.

Este caso de uso representa el conjunto de funciones encargadas de la creación y la exploración de los nodos de una vista del espacio de información. El espacio de información por completo es una vista por defecto del sistema OPC UA. La definición del caso de uso se presenta en la tabla 17.

Tabla 17. Definición del caso de uso Vistas.

Id.	CU-B011
Caso de uso	Vistas
Actores	ClienteUA
Tipo	Básico.
Propósito	Crear y Explorar las vistas que organizan y estructuran el AddressSpace.
Resumen	Agrupar un conjunto de nodos en una estructura que sirve para que los clientes accedan a un grupo de nodos de interés.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	<p>El cliente solicita crear una vista.  El sistema crea un nodo del tipo View.  El sistema solicita los nodos que hacen parte de la vista.  El cliente selecciona los nodos que hacen parte de la vista  El cliente relaciona los nodos con el nodo tipo View a través de referencias Organizes.  El cliente guarda la versión de la vista.</p>
Subflujos	Ninguno.
Excepciones	Todos los nodos de la vista deben estar referenciados entre ellos. Si no es así, el sistema informa al cliente que el nodo no se incluir en la vista.

Fuente: Elaboración propia, abril 2008.

### 3.1.15 Consultas.

Este caso de uso representa las consultas que el cliente UA puede realizar al servidor UA, para ello el sistema debe implementar una estructura genérica que permita explorar el espacio de información. La definición formal del caso de uso se presenta en la tabla 18.

Tabla 18. Definición del caso de uso Consultas.

Id.	CU-B012
Caso de uso	Consultas
Actores	ClienteUA
Tipo	Básico.
Propósito	Diseñar y crear las consultas que se requieren para explorar el AddressSpace.
Resumen	La función realiza las consultas de manera genérica para que el cliente pueda acceder a una gran variedad de información, independientemente del sistema subyacente.
Precondiciones	Abrir una sesión en el servidor.
Flujo principal	El cliente solicita crear una consulta. El cliente define los nodos a consultar El cliente define los criterios de búsqueda de la consulta El sistema retorna en forma de arreglo el valor de los atributos consultados.
Subflujos	Por cada consulta realizada se emite una ventana con el reporte de la información de los nodos.
Excepciones	Si los atributos seleccionados no tienen ningún valor, el sistema informa que estos se encuentran vacíos.

Fuente: Elaboración propia, abril 2008.

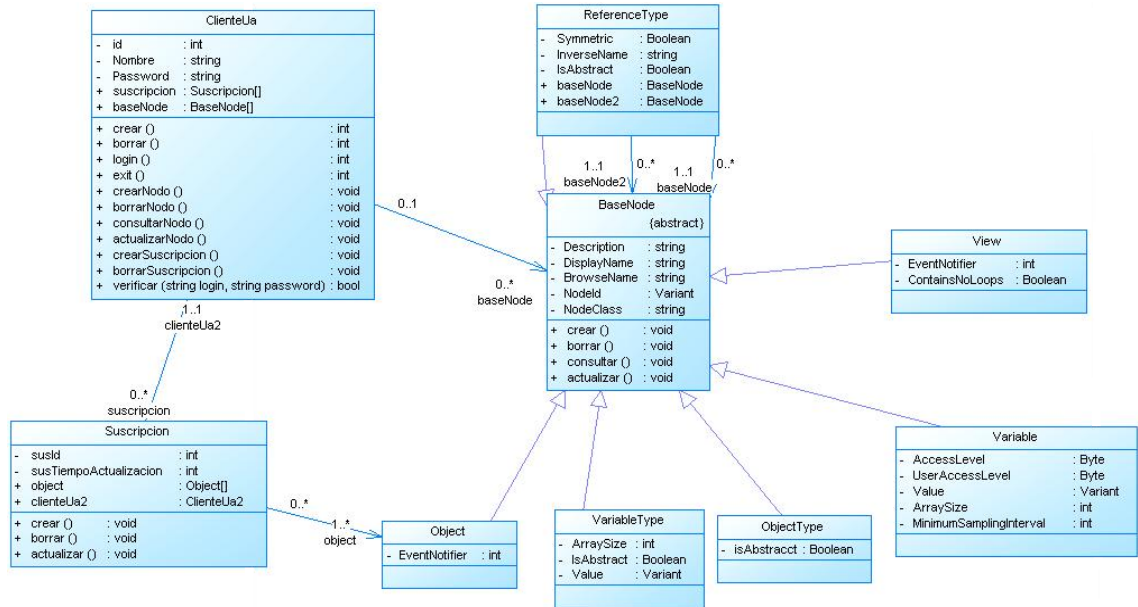
### 3.2 MODELO DE CLASES DEL SISTEMA OPC UA.

El modelo de clases del sistema OPC UA obtenido a partir del análisis realizado a los diferentes escenarios del modelo de caso de usos, permitió obtener las unidades básicas que modelan la información del sistema, con la cual se describe la estructura del sistema. El modelo se presenta en la figura 15.

Para obtener el modelo se utilizó como referencia el metamodelo de OPC UA el cual define los nodos del AddressSpace con los cuales se modela la información, sin embargo en un análisis orientado a objetos es necesario que las clases se auto describan con sus atributos y sus métodos, por lo tanto para el metamodelo de OPC UA fue necesario definir las funciones básicas que se desarrollan en el AddressSpace.



Figura 15. Modelo de clases del sistema OPC UA para la implementación.



Fuente: Elaboración propia, noviembre 2008.

Además de las clases identificadas a partir del metamodelo se creó la clase cliente UA con la cual se instancia los objetos que representan los usuarios que acceden al sistema. El objeto Suscripción representa la unidad básica a partir de la cual se instancian las suscripciones que el cliente puede realizar con el propósito de monitorear el funcionamiento de los nodos objeto en el sistema.

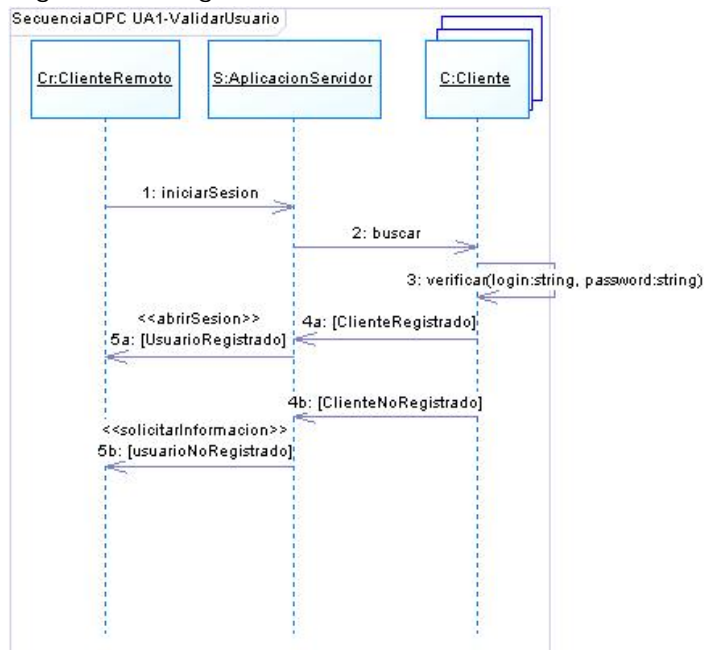
El modelo de clases permite la definición de los objetos básicos que almacenan la información y ejecutan los procesos o funcionalidades del sistema, para representar estas interacciones entre los objetos se utilizan los diagramas de secuencia con los cuales se modela la interacción de los objetos para brindar una representación de la funcionalidad del sistema en el tiempo.

### 3.3 DIAGRAMA DE SECUENCIA DEL SISTEMA OPC UA.

Los diagramas de secuencia del modelo son presentados en el disco compacto anexo a la monografía, en este se observan los diagramas que describen los flujos de información que se presentan con las funcionalidades del sistema. Por ejemplo la validación de usuario, agregar, borrar y consultar un nodo, etc.

La figura 16 expone parte de los diagramas de secuencia para el modelo de caso de usos presentados en la figura 14, a través del cual se valida el ingreso de usuario al sistema OPC UA.

Figura 16. Diagrama de secuencia ValidarUsuario.



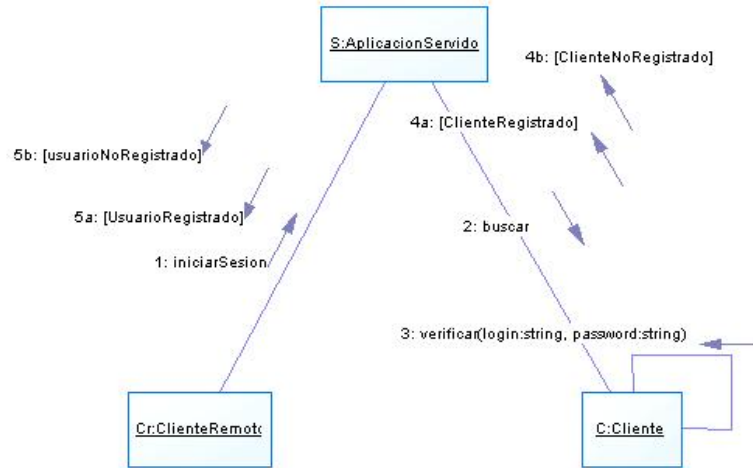
Fuente: Elaboración propia, noviembre 2008.

### 3.4 DIAGRAMA DE COLABORACION DEL SISTEMA OPC UA.

Los diagramas de colaboración del sistema OPC UA presentan las interacciones organizadas alrededor de los roles del sistema. A diferencia de los diagramas de secuencia, los diagramas de comunicación muestran explícitamente las relaciones de los roles del modelo.

Por otra parte, el diagrama de colaboración no presenta el tiempo como una dimensión aparte y por ello es necesario etiquetar con números de secuencia, tanto la secuencia de mensajes como los hilos concurrentes del sistema. En el disco compacto anexo a la monografía se presentan todos los diagramas de colaboración realizados para el proyecto con los cuales se describe el modelo de casos de uso de la figura 14. En la figura 17 se presenta el diagrama de colaboración para los casos de uso cuyo propósito es describir la validación del usuario en el sistema OPC UA.

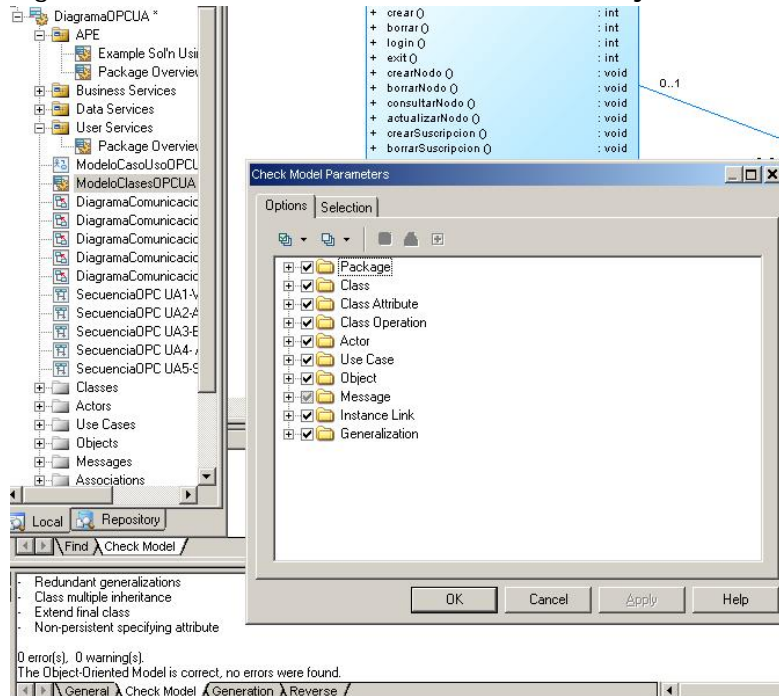
Figura 17. Diagrama de colaboración ValidarUsuario.



Fuente: Elaboración propia, noviembre 2008.

Como se presenta en la figura 18, la validación de los modelos obtenidos a partir del análisis y el diseño orientado a objetos del sistema tienen la integridad que le permite relacionar correctamente los flujos de información con los objetos y las unidades básicas del sistema.

Figura 18. Validación del modelo orientado a objetos del sistema OPC UA



Fuente: Elaboración propia, noviembre 2008.

#### 4. PRINCIPIOS DE APLICACIÓN DEL ESTANDAR OPC UA, LLEVADOS A UN CASO DE ESTUDIO

Para la aplicación del estándar OPC UA se han identificado 5 principios que las empresas de manufactura pueden utilizar con el propósito de establecer los sistemas OPC UA en su organización. Con el objetivo de brindarle a las empresas de manufactura una forma ágil de implementar el estándar, se establecen de forma general los principios y luego se direccionan al caso de estudio.

- 1 **Información sistematizada dentro de las empresas de manufactura:** para la aplicación del estándar OPC UA se requiere que las empresas cuenten con algún sistema informático que sistematice los flujos de información, ya que el estándar OPC UA tiene como objetivo almacenar la información de las diferentes fuentes de datos dentro de un espacio de información unificado el cual es accedido a través de servicios web por los sistemas informáticos de la empresa. Debido a que la implementación del estándar no hace parte del alcance del proyecto, para el trabajo desarrollado contar con las aplicaciones web de la empresa caso de estudio que acceden al servidor no es de relevancia para el proyecto.
- 2 **Análisis de los requerimientos de la empresa:** las empresas de manufactura manejan grandes volúmenes de datos y dependiendo de los requerimientos hay información que es más relevante para determinadas aplicaciones o procesos de la empresa. Por esta razón, este análisis tiene como propósito organizar la información que se almacenará en los sistemas OPC UA, la cual será accedida por las aplicaciones UA a través de servicios web.

Para el caso de estudio, se parte de los modelos de información propuestos en la investigación realizada en el proyecto "APLICACIÓN DE LA NORMA ISA S95 EN UN CASO DE ESTUDIO". En ese proyecto se recopiló la información de la empresa a través de la investigación al personal, los sistemas y los flujos de información del caso de estudio. La descripción de los requerimientos de la empresa se presentan en el literal 4.1.

- 3 **Aplicación de la arquitectura OPC UA:** para realizar la aplicación de los sistemas OPC UA en las empresas, el personal encargado de la implementación debe realizar la aplicación del metamodelo en la empresa y establecer las características técnicas

de los servidores para la comunicación y la seguridad en el intercambio de mensajes.

Considerando este principio, aquellas personas encargadas de llevar a cabo la implementación del estándar OPC UA en las empresas deben determinar los requerimientos técnicos de los servidores OPC UA o lo que el estándar determina "*Perfil del servidor*".

Este principio es desarrollado a partir del literal 4.2. En este se definen las pautas técnicas para la comunicación entre servidores, los perfiles de los servidores UA y se especifican los protocolos y tipos de codificación que se deben utilizar y la combinación de estos para lograr compensación entre portabilidad y eficiencia en la comunicación.

- 4 **Implementar los servidores OPC UA en la empresa:** este principio define que el personal responsable de la implementación de los sistemas OPC UA debe definir los equipos, las tecnologías y las interfaces software en la implementación de los servidores OPC UA y la comunicación con las aplicaciones cliente a través de las cuales los servidores OPC UA interactúan. Para el caso de estudio y el alcance del proyecto la implementación del estándar no hace parte del desarrollo del proyecto
- 5 **Documentar el trabajo desarrollado:** en los proyectos de aplicación del estándar OPC UA es indispensable realizar una completa documentación del diseño y de la implementación de los sistemas OPC UA con el propósito de dejar constancia de los métodos y procedimientos utilizados para facilitar las posteriores actualizaciones del sistema.

A partir del literal 3 de esta monografía, se documenta el trabajo realizado para exponer el desarrollo de los principios propuestos para la aplicación del estándar. En los anexos B, C, D y E de esta monografía se presenta en forma detallada la aplicación del metamodelo de OPC UA a través del diseño del AddressSpace en los servidores a nivel de gestión y control de procesos del caso de estudio.

#### 4.1 PRESENTACIÓN DE LA EMPRESA CASO DE ESTUDIO

La empresa caso de estudio tiene como objetivo la producción y comercialización de productos lácteos y sus derivados, para su funcionamiento interno genera gran cantidad de información la cual es relevante tanto para el negocio como para los procesos de producción. Toda esta información es recolectada y analizada con el fin de garantizar el cumplimiento de estándares de calidad, aumentar los niveles de

productividad y mejorar el desempeño en los procesos traducidos en beneficios para la empresa. Para la aplicación del estándar OPC UA a la empresa caso de estudio se parte de los modelos de información realizados en el proyecto “APLICACIÓN DE LA NORMA ISA S95 EN UN CASO DE ESTUDIO” dentro del grupo de investigación I+D en Automática Industrial de la universidad del Cauca, el propósito es establecer los principios de aplicación de este nuevo estándar sobre otros reconocidos y aceptados mundialmente de los cuales el grupo de investigación tiene referencias, de tal manera que se relacione esta nueva investigación en los proyectos precedidos.

Es importante resaltar que hasta ahora no se conocen trabajos que integren estos estándares, sin embargo la OPC Foundation define a OPC UA como el estándar de estándares porque brinda la plataforma para el desarrollo de los demás estándares, los cual está en proceso de validación dentro de los grupos de investigación en automatización industrial. Por ejemplo el grupo IICI (Industrial Interoperability Compliance Institute) formulo un proyecto de negocios para establecer la interoperabilidad industrial, donde la iniciativa se centra en los estándares propuestos por la organización ISA y la OPC Foundation. Por lo tanto es importante destacar que esta aplicación teórica en el grupo de investigación desarrolla un valioso aporte en aras de lograr la integración de estándares como ISA 95 y OPC UA.

El proyecto define la aplicación del estándar UA en los niveles de gestión y control de procesos, para ello se hace referencia a los modelos de información de mayor dominio en cada uno de estos niveles de tal manera que el conocimiento de la información sea transversal en toda la investigación. La información a nivel de gestión formula los requerimientos de producción para el nivel de control y administración de manufactura, el cual dispone de información referente al desempeño de las operaciones para la transformación de materia prima y la utilización de equipos en la producción.

#### 4.1.1 Consideraciones del modelo de información de la empresa caso de estudio

Estas consideraciones se hacen teniendo en cuenta los diferentes elementos que hacen parte del modelo de información y se presentan a continuación:

- Acorde a la estructura jerárquica de la pirámide para la automatización industrial, por cada nivel existe al menos un servidor OPC UA en el cual se encuentra disponible la información que corresponde a las funciones identificadas en el estándar ISA S95.

- Los servidores OPC UA ubicados en los niveles de la estructura jerárquica, tienen la capacidad de acceder a los servidores inferiores o superiores de acuerdo a la propiedad de agregación de servidores del estándar OPC UA.
- El protocolo para la comunicación de los servidores en los diferentes niveles hace parte de la definición del perfil del servidor.

#### 4.2 APLICACIÓN DE LA ARQUITECTURA OPC UA AL CASO DE ESTUDIO

Para el desarrollo de este principio se proponen las siguientes pautas para la aplicación del estándar OPC UA al caso de estudio dentro de los cuales se estructura el *AddressSpace*. Posteriormente en las pautas se indica la definición de las características técnicas que dan soporte a la comunicación y almacenamiento de datos de cada uno de los servidores del caso de estudio.

Pautas para la aplicación del estándar OPC UA:

1. Establecer los criterios para la definición de los elementos del *AddressSpace*. Esta pauta para la aplicación se lleva a cabo de manera transversal en todas las capas del *AddressSpace*, debido a que es necesario compartir y reutilizar el conocimiento. Esta propuesta se hace mediante la definición de criterios para el nombramiento de los elementos del *AddressSpace* del caso de estudio que permita normalizar y ahorrar tiempo en la actualización y mantenimiento de los diseños desarrollados.
2. Estructurar las capas del *AddressSpace* para la aplicación. La estructuración del *AddressSpace* tiene como objetivo presentar cada una de las capas y la dinámica que existe entre ellas.
3. Aplicar los modelos de información dentro de cada una de las capas del *AddressSpace*. Con base a las capas definidas del *AddressSpace* de la pauta anterior, se realiza la definición de los elementos según la aplicación al caso de estudio.
4. Codificar el modelo de información que se obtuvo en la pauta tres. A través de la herramienta ALTOVA XMLSpy Enterprise Edition 2008 se realiza la codificación de los modelos de información del caso de estudio y se proponen los esquemas XML UA del caso estudio.

5. Validar los esquemas XML UA del caso de estudio. A través de la herramienta ALTOVA XMLSpy Enterprise edition 2008 se validan los esquemas generados en la aplicación del modelado UA del caso de estudio.
6. Pautas técnicas para la comunicación de los servidores a nivel de gestión y de control de procesos. Definición de las unidades de conformidad para la estructuración del stack de comunicación de los servidores OPC UA del nivel de control y de gestión.
7. Pautas para la migración de servidores OPC DA COM a los servidores OPC UA. Se establecen las pautas para la migración de las tecnologías COM a las tecnologías UA a través de servicios web.

#### 4.3 DESARROLLO DE LOS PASOS PARA LA APLICACIÓN DEL ESTANDAR OPC UA AL CASO DE ESTUDIO

A continuación se desarrolla cada uno de los pasos propuestos para la aplicación del metamodelo al caso de estudio, estos pasos se desarrollan paralelamente para cada uno de los niveles. Sin embargo, el tercer paso se complementa con los anexos B, C, D y E en los cuales se estructura y se presenta la aplicación del modelado de información del caso de estudio.

##### 4.3.1 Establecer los criterios para la definición de los elementos del *AddressSpace*.

Las convenciones presentadas a continuación representan la recopilación y la propuesta de un conjunto de referencias para la construcción del *AddressSpace*, este paso presenta los criterios para definir los atributos básicos de los nodos, así como la guía para la definición de los nombres de los nodos y las convenciones para graficar los nodos en el *AddressSpace*.

- A. Criterios para el nombramiento de los elementos del *AddressSpace*. El proyecto propone un conjunto de reglas para el nombramiento de los elementos o nodos que conforman el *AddressSpace*, la estructura para su definición se presenta en la tabla 19.



Tabla 19. Criterios para el nombramiento de los elementos del *AddressSpace*

Elementos	Descripción
	Todos los nombres de los elementos se escriben completamente, así el nombre completo este conformado de varios nombres. No es necesario definir las preposiciones o conjunciones que relacionan las palabras. Todos los elementos del AddressSpace se describen con nombres en singular. Todos los nombres de los elementos de la capa de tipos identificados deben tener el sufijo "Type" a excepción de la referencias.
ObjectType	Siempre comienzan con Mayúscula. Si el nombre se compone de varias palabras entonces cada inicial debe ser en Mayúscula.
VariableType	Siempre comienzan con Mayúscula. Si el nombre se compone de varias palabras entonces cada inicial debe ser en Mayúscula.
DataType	Siempre comienzan con Mayúscula. Si el nombre se compone de varias palabras entonces cada inicial debe ser en Mayúscula.
Objetos	Siempre comienzan con minúscula. Si el nombre se compone de varias palabras entonces cada inicial subsiguiente debe ser en Mayúscula
Variables	Siempre comienzan con minúscula. Si el nombre se compone de varias palabras entonces cada inicial subsiguiente debe ser en Mayúscula
Atributos	Siempre comienzan con mayúscula. Si el nombre se compone de varias palabras entonces cada inicial subsiguiente debe ser en Mayúscula.

Fuente: Elaboración propia, Agosto 2008

- B. Convenciones para la descripción de los nodos [5]. Para la definición de los nodos se utiliza la definición propuesta por el estándar OPC UA, el cual propone se defina cada nodo a través de tablas que especifican sus atributos y referencias. La tabla 20 presenta el esquema de la tabla para la definición de nodos.

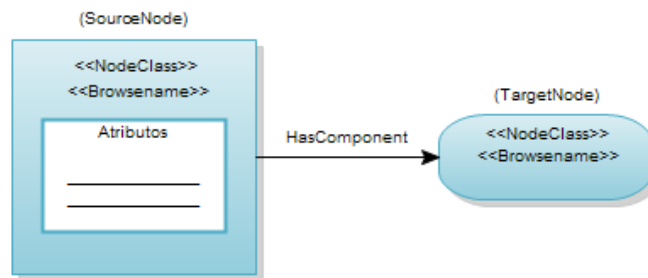
Tabla 20. Formato para la definición de nodos.

Atributo	Valor				
Nombre del atributo	Valor del atributo. Si este es un atributo opcional que no esta establecido. El símbolo "--" será utilizado.				
Referencias	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
Nombre del tipo de referencia	Nombre del Nodo clase del TargetNode.	El BrowseName del TargetNode. Si las referencias son instanciadas por el servidor, entonces el valor del BrowseName TargetNode es "--".	Atributos del nodo referenciado, solo aplica para variables y para objetos.	Regla de modelamiento del objeto referenciado	
Nota –					
1) Notas					

Fuente: OPC Foundation, *OPC UA Specification Part 3: Information Model*. Version 1.01.08

C. Convenciones para la diagramación del AddressSpace. La representación grafica aporta claridad para el entendimiento de los modelos, por lo tanto es importante definir convenciones para su representación, los nodos se representan como se expone en la figura 19. la cual es una variación a la representación del estándar con el propósito de hacer está un poco mas explicita.

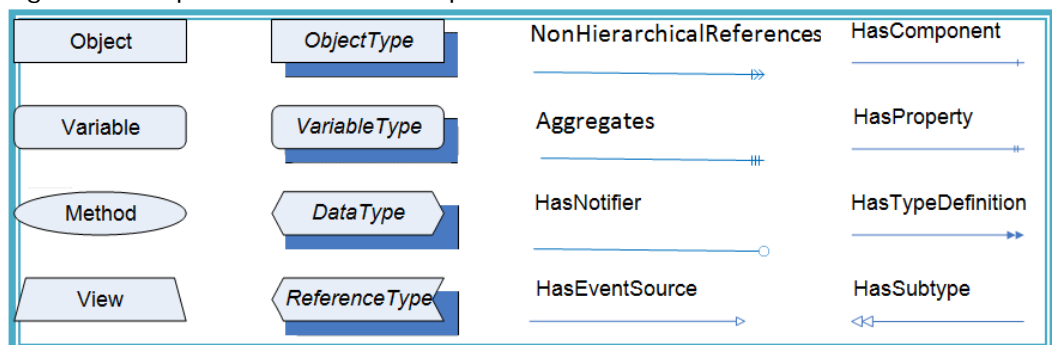
Figura 19. Representación de un nodo en el AddressSpace de OPC UA.



Fuente: Elaboración propia, Agosto 2008

Debido a que absolutamente todos los elementos del AddressSpace son nodos, es necesario especificar una representación especializada de cada uno. La figura 20 es un aporte del proyecto a la propuesta realizada por la OPC Foundation en el seminario UA DevCon 2007 la cual no hace parte del estándar.

Figura 20. Especialización de la representación de Nodos



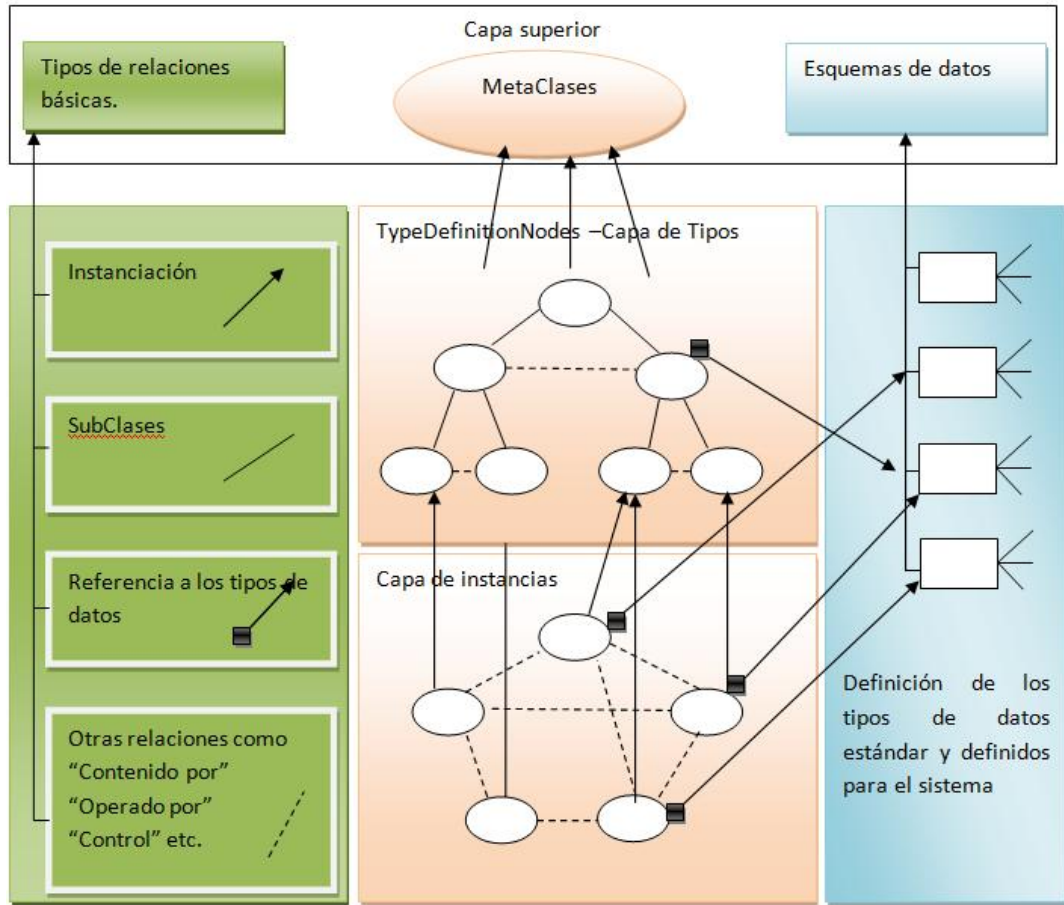
Fuente: Elaboración propia, Agosto 2008

#### 4.3.2 Estructurar las capas del AddressSpace para la aplicación [13].

La estructuración en capas del AddressSpace propuesta por OPC UA, tiene como propósito organizar funcionalmente los diferentes nodos del AddressSpace, la siguiente lista de capas hace parte del AddressSpace, el cual se presenta de forma estructurada en la figura 21.

- Capa del metamodelo de información
- Capa de tipos
- Capa de instancias
- Capa de referencias
- Namespace de los tipos de datos.

Figura 21. Estructura del AddressSpace



Fuente: OPC Today and in the Future, OPC ARC Work Group June 2006.ppt

La capa superior representa el conjunto de elementos básicos definidos por OPC UA. Esta capa solo se modifica por la OPC Foundation y no hay otra organización autorizada para extender la definición de esta capa, por lo tanto en ella no se realiza una aplicación. Los elementos definidos en esta capa son referenciados en el capítulo II.

La columna izquierda presenta el conjunto de relaciones definidas por OPC UA, las cuales son establecidas a partir de la NodeClass *References* de la capa superior. Este conjunto de relaciones define su semántica. Por ejemplo, instanciación, subtipos, relaciones de pertenencia, operaciones, entre otras. Para la aplicación del modelado en el caso de estudio se identifican las relaciones con su semántica para establecer el modelo y se estructuran dentro de la jerarquía de OPC UA.

La columna derecha, indica la definición de los tipos de datos aplicados al caso de estudio, algunos de ellos son definiciones por el estándar OPC UA y otros corresponden a la creación de tipos de datos complejos en el proyecto necesarios para la aplicación en el caso de estudio.

La capa de tipos o *TypeDefinitionNodes* consiste de los tipos de nodos a partir de los cuales se instancian los objetos, variables o vistas en la capa de instancias. Los nodos de esta capa pueden ser abstractos o no, si es el caso estos tienen una relación indirecta con la capa de instancias.

La capa de instancias representa la información del servidor OPC UA la cual puede estar en función de la lógica o física del caso de estudio. Cada vez que se inicia sesión en el servidor los objetos presentan la información de los nodos.

El conjunto de relaciones establecidas por OPC UA y aquellas creadas por el sistema permiten que la representación de los nodos en el *AddressSpace* tenga una estructura poco clara, sin embargo esta es una de las propiedades más importantes de los servidores OPC UA ya que la modularidad e independencia de los objetos permiten que estos sean reutilizables y que el diseño sea escalable.

4.3.3 Aplicar los modelos de información dentro de cada una de las capas del *AddressSpace*.

Para llevar a cabo la aplicación de estos modelos en el proyecto se han tenido en cuenta los siguientes servidores:

- Servidor del nivel de control de procesos
- Servidor del nivel de gestión

Para la definición de cada una de las capas de estos servidores el proyecto propone llevar a cabo las siguientes fases:

- Definición de la capa de tipos del *AddressSpace*.
- Definición de los tipos de datos
- Definición de los tipos de relaciones.
- Definición de las instancias declaradas del *AddressSpace*

4.3.3.1 AddressSpace del servidor de control de procesos. La información que corresponde al nivel de control de procesos relaciona la información de materiales y equipos para el caso de estudio.

La definición y la codificación completa de los nodos está estructurada y organizada en los respectivos anexos de los modelos con el propósito de no extender el capítulo con las definiciones de la aplicación. Sin embargo para cada uno de los modelos se presenta una abstracción de los nodos que representan la semántica del modelo.

A Modelo de materiales [14]. El modelo de objetos de material establece la información referente a los recursos de material para planificación, programación y reportes del desempeño de producción. Para ello el modelo estructura la información a través de objetos que representan la agrupación de materiales con características comunes o materiales específicos, así mismo el modelo define variables simples y complejas que representan propiedades o parámetros de los tipos de objetos o del modelo del *AddressSpace*.

A.1 Nodos de la capa de tipos del modelo de materiales. Los nodos identificados en el modelo de materiales se presentan a continuación. La definición de los atributos y de las referencias entre los nodos se presenta en el Anexo B.

- LacteoType
- EdulcoranteType
- EdulcoranteType
- EstabilizanteType
- SaborizanteType
- MilkingType
- MaterialEmpaqueType

Estos nodos de la capa de tipos son agrupaciones de materiales que definen un conjunto de características comunes para los materiales del proceso UHT quienes son parte de la capa de tipos e igualmente son descritos en el anexo B. A continuación se presentan los materiales específicos identificados dentro del caso de estudio.

- LecheCrudaEnteraType
- LecheEstandarizadaHigienizadaType
- LecheTermizadaType
- LecheBaseUHTType
- LecheSaborizadaUHTType
- CremaLecheType
- AzucarType
- CarrageninaType
- TripolifosfatioSodioType
- TripolifosfatioSodioType
- TetrabrickType

Cada objeto tiene las propiedades que describen las características que lo describen y para representarlas en el caso de estudio, se ha creado un conjunto de variables simples y complejas que se utilizan para instanciarlas en la definición del objeto. Las variables para el modelo se presentan a continuación y se definen en el anexo B.

- CapacidadType
- RangoType
- VitaminaType

A.2 Nodos de la capa de referencias del modelo de materiales. La relación más apropiada e identificada entre los grupos de materiales y el material en la capa de tipos expresa una especialización entre el grupo y el material. Por ejemplo, en la figura 22 el objeto LacteoType representa la agrupación de objetos con características similares a los lácteos, este objeto tiene relacionado el *ObjectType* LecheCrudaEnteraType el cual es una manera de especializar su definición indicando un tipo de leche que es cruda y entera. Para representar esta función se ha utilizado la referencia *HasSubtype* que define la especialización entre los nodos y que es utilizada directamente en el *AddressSpace*.

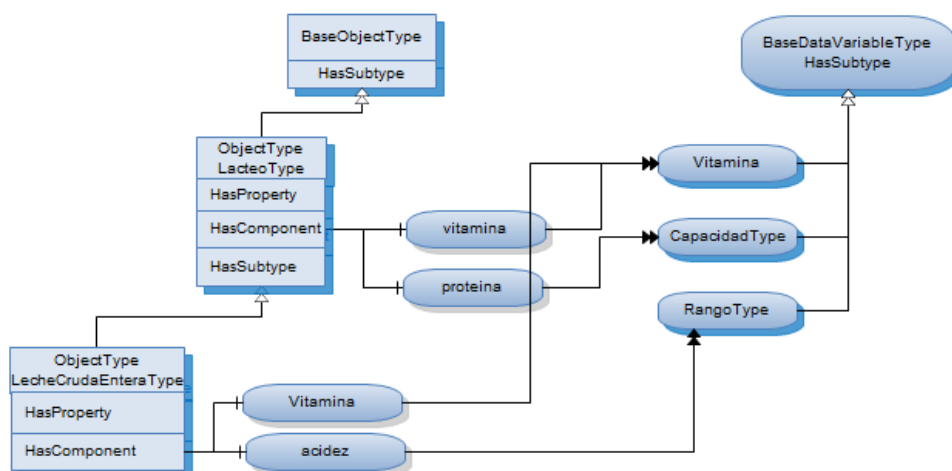
Respecto a las variables identificadas y que hacen parte de los objetos de la capa de tipos se utiliza la relación de agregación *HasComponent* la cual define a través de un arreglo las propiedades del producto lácteo. A pesar que la relación *HasProperty* tiene una semántica idéntica a la relación *HasComponent*, esta no se utiliza ya que las variables complejas no pueden ser el *TargetNode* de este tipo de relación.

A.3 Tipos de datos del modelo de materiales. Los tipos de datos definidos corresponden a los utilizados para definir el tipo de dato de las variables del caso de estudio, estos son construidos con base a la definición de los tipos de datos de OPC UA. Los tipos de datos se presentan a continuación y su definición se presenta en el anexo B.

- RangoDataType.
- CapacidadDataType
- VitaminaDataType

En la figura 22 se presenta la aplicación del metamodelo UA a un grupo de nodos del modelo de materiales del caso de estudio, en el se presentan los nodos base del estándar a partir del cual se crean los objetos y las variables del modelo. En este ejemplo el objeto *LacteoType* tiene relacionado el objeto *LecheCrudaEnteraType* a través de la referencia *HasSubtype* y cada nodo tiene referencias *HasComponent* a las instancias de las variables definidas del caso de estudio.

Figura 22. Ejemplo de la capa de tipos del modelo de materiales.



Fuente: Elaboración propia, Agosto 2008



El nodo LacteoType de la figura 22 es una especialización del objeto base, ya que no existe una definición más concreta para este tipo de objeto. A través de este nodo se representa la agrupación de materiales con características similares a la leche y sus derivados y se asocia al objeto LecheCrudaEnteraType para indicar que este nodo es una especialización del nodo lacteoType.

En el ejemplo se presenta la variable vitamina la cual es una instancia declarada del nodo LecheCrudaEnteraType, esta variable es heredada del objeto LacteoType, por lo tanto todas las instancias del objeto LecheCrudaEnteraType tienen idénticamente definida esta propiedad.

Los tipos de datos no son representados directamente en el *AddressSpace*, esto es algo implícito en las variables las cuales instancian a través de un namespace los tipos de datos del sistema.

Las relaciones que hacen parte del *AddressSpace* son instanciadas directamente y se representan en el AddressSpace, la definición de las relaciones definidas para el caso de estudio se presentan en el anexo B. Las referencias definidas por el estándar se presentan en el anexo A.

A continuación se presenta la realización de la definición de los nodos de cada una de las capas del ejemplo del modelo de materiales.

El objeto LacteoType agrupa los alimentos lácteos y sus derivados, definiendo un conjunto de propiedades básicas para todos ellos, la definición formal del objeto se presenta en la tabla 21.

Tabla 21. Definición del *ObjectType* LacteoType

Atributo	Valor				
NodeId	5001				
NodeClass	ObjectType				
BrowseName	LacteoType				
DisplayName	Lacteos				
Description	Grupo de alimentos que abarca la leche y sus derivados. Se caracteriza por aportar gran cantidad de calcio, proteínas y vitaminas.				
IsAbstract	False				
Referencias	NodeClass	BrowseName	Data Type	Type Definition	Modelling Rule
Subtipo del Base <i>ObjectType</i> , hereda sus propiedades.					

HasProperty	Variable	especie	String	PropertyType	None
HasProperty	Variable	origen	String	PropertyType	None
HasComponent	Variable	proteína	Capacidad DataType	CapacidadType	None
HasComponent	Variable	lactosa	Capacidad DataType	CapacidadType	None
HasComponent	Variable	vitamina	VitaminasD ataType	VitaminasType	None
HasComponent	Variable	aflatoxina	Capacidad DataType	CapacidadType	None
HasSubtype	ObjectType	LecheCrudaEnteraType	Definido en el anexo B		
HasSubtype	ObjectType	LecheEstandarizadaHigienizadaType	Definido en el anexo B		
HasSubtype	ObjectType	LecheTerminizadaType	Definido en el anexo B		
HasSubtype	ObjectType	LecheBaseUHTType	Definido en el anexo B		
HasSubtype	ObjectType	LecheSabORIZADAUHTType	Definido en el anexo B		
HasSubtype	ObjectType	CremaLecheType	Definido en el anexo B		

Fuente: Elaboración propia, Agosto 2008

A través de las relaciones HasProperty se definen las propiedades cuyos valores son simples tipos de datos, como por ejemplo la propiedad "especie" la cual describe a través de una cadena de texto del tipo string la especie del animal que produce la leche. La relación HasComponent por su parte relaciona variables complejas que permiten agrupar conjunto de propiedades, como por ejemplo la propiedad "proteína" la cual describe su valor a través de una cantidad y una unidad de medida.

La variable tipo RangoType es utilizada para especificar variables que definen sus valores mediante un valor máximo, mínimo, estándar, y las unidades de medida, como se presenta en la tabla 22.

Tabla 22. Definición del VariableType RangoType

Atributo	Valor				
BrowseName	5019				
IsAbstract	False				
ArraySize	-1				
DataType	RangoDataType				
Referencias	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtipo del BaseDataVariableType.					
HasProperty	Variable	minimo	Number	PropertyType	OptionalNew
HasProperty	Variable	maximo	Number	PropertyType	OptionalNew
HasProperty	Variable	estandar	Number	PropertyType	New
HasProperty	Variable	unidadMedida	EUInformation	PropertyType	New

Fuente: Elaboración propia, Agosto 2008

Esta variable es compleja y se estructura como un conjunto de variables simples instanciadas a partir de las variables básicas de OPC UA. Los tipos de datos hacen parte de los datos definidos por el estándar y definen el tipo de dato RangoDataType.

El tipo de dato RangoDataType es creado como un subtipo del tipo BaseDataType definido por OPC UA. La estructura para su definición se presenta en la tabla 23.

Tabla 23. Identificación del DataType RangoDataType

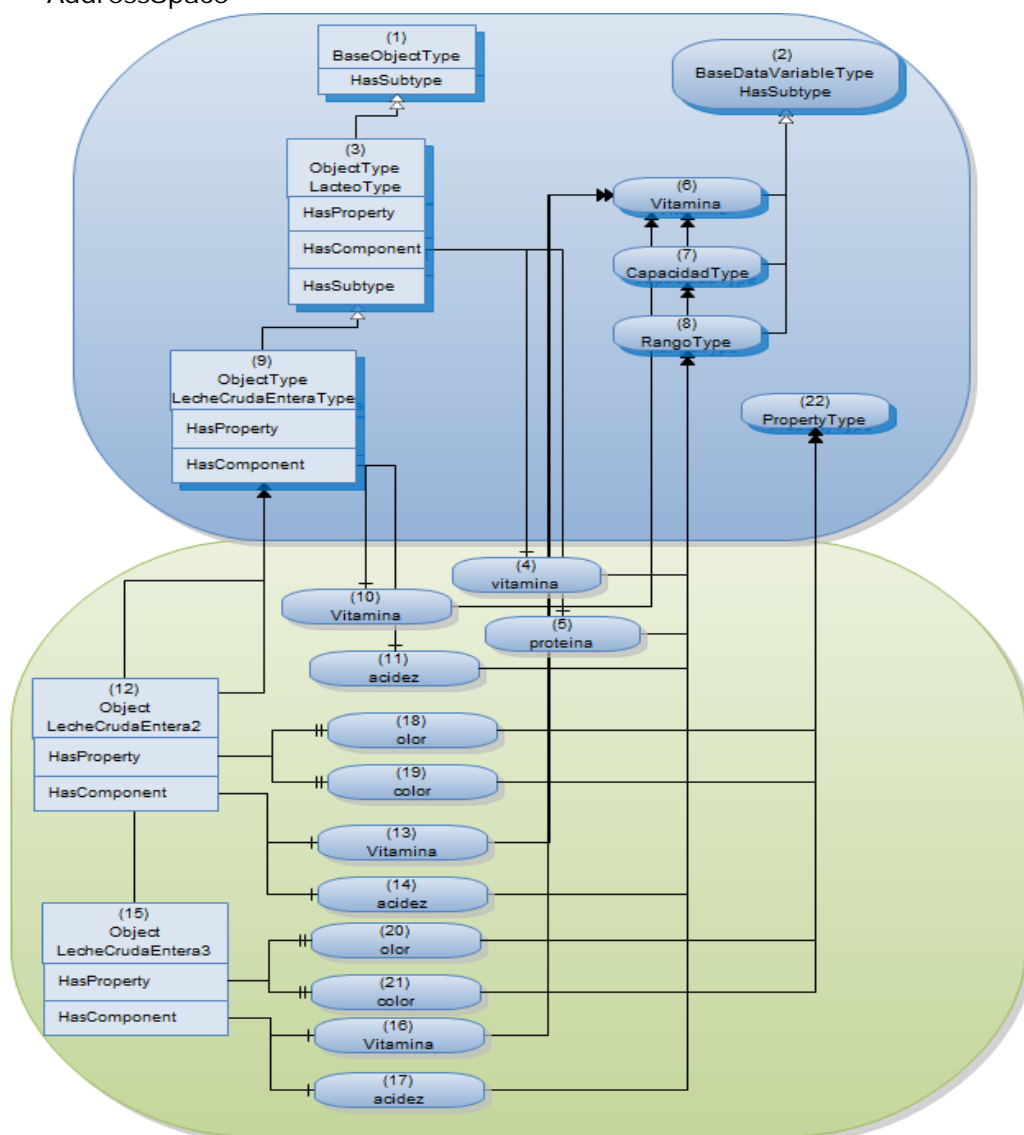
Nombre	Tipo	Descripción
RangoDataType	Estructura	
minimo	Number	Valor mínimo de la variable que puede tener
Máximo	Number	Valor máximo de la variable que puede tener
Estándar	BaseDataType	Valor estándar o promedio de la de la variable
UnidadMedida	EUInformation	Unidad de medida de la variable.

Fuente: Elaboración propia, Agosto 2008

A.4 Ejemplo de la instanciación de la capa de tipos del modelo de materiales. Un ejemplo de la relación entre la capa de tipos y la capa de instancias se modela en la figura 23, en ella la capa de tipos es instanciada para crear los objetos y las variables que el cliente consulta a través de los servicios web UA. La capa de color azul representa los nodos de la capa de tipos que relacionan las instancias declaradas. Como se presenta en esta figura, las instancias declaradas son copias idénticas en todas las instancias de los objetos de leche cruda, esta una propiedad adicional en el concepto de instanciación de OPC UA. Así mismo en la grafica se observa la aplicación del concepto tradicional de instanciación en el cual las variables del tipo PropertyType

son instanciadas para crear distintas variables, como por ejemplo olor y color en el ejemplo.

Figura 23. Interacción de la capa de instancias y la capa de tipos del AddressSpace



Fuente: Elaboración propia, Agosto 2008

Los *NodeIds* presentados en el ejemplo corresponden a números simbólicos en el *AddressSpace*, la estructura real está definida en el documento *NodeIdsCasoEstudio.csv* anexo al proyecto y definido por el tipo de dato *NodeId* presentado en el anexo A, el cual incluye el *namespace* y el identificador del nodo.

De esta forma la información del servidor está disponible a los clientes donde cada nodo tiene definido los atributos que lo describen e identifican y un conjunto de referencias que son establecidas para relacionar las propiedades del nodo.

B. Modelo de equipos [14]. El modelo de equipos describe la forma de representar los equipos de un proceso de producción de forma estándar, el modelo de referencia para el desarrollo de la investigación corresponde al resultado del proyecto "APLICACIÓN DEL ESTÁNDAR ISA 95 EN CASO DE ESTUDIO".

A través de la especificación de acceso a datos basada en COM, la OPC Foundation define un conjunto de interfaces que acceden a los grupos e ítems provenientes de los PLC's del proceso, sin embargo estas interfaces solo permiten el acceso a datos en tiempo de ejecución. Para la configuración y el acceso a alarmas y eventos generados por el PLC, el estándar basado en COM define un conjunto de interfaces diferentes y por lo tanto se tienen servidores diferentes que hay que configurar por separado. Por lo tanto, OPC UA se propone integrar en un único modelo, los datos, las alarmas y los eventos, razón por la cual este proyecto se permite extender su alcance hasta la configuración de alarmas.

Las etapas para la aplicación del modelo de equipos al metamodelo de OPC UA se exponen a continuación:

- Identificación de los nodos de cada una de las capas del *AddressSpace*.
- Identificación de las instancias declaradas.
- Definición de los objetos y las propiedades que soportan la generación de eventos e históricos.

B.1 Nodos de la capa de tipos del modelo de equipos. Para la identificación de los nodos de la capa de tipos se toma como referencia el modelo jerárquico de equipos del estándar ISA 95 el cual estructura la organización del proceso en áreas, células de proceso y unidades de proceso, las cuales contienen los equipos del proceso UHT. El resultado de este análisis propone el siguiente conjunto de nodos los cuales corresponden con el nivel de unidad de proceso.

- UnidadRepcionType
- UnidadEstandarizacionHigienizacionType
- UnidadTermizacionType

- UnidadDosificacionType
- UnidadUltrapasterizacionType
- UnidadEmpacadoType

A continuación se presentan los nodos identificados del caso de estudio que hacen parte de la capa de tipos y que conforman las unidades de proceso identificadas anteriormente. La definición completa de los nodos se presenta en el anexo C.

- TanqueType
- BombaType
- ClarificadoraType
- MezcladorType
- HomogenizadorType
- IntercambiadorType
- PasteurizadorType
- UltrapasteurizadorType
- EmpacadoraType

B.2 Nodos de la capa de referencias del modelo de equipos. Para relacionar los nodos de la capa de tipos, el modelo del caso de estudio expone una relación de pertenencia de las unidades de proceso con los equipos, con la cual se estructura la semántica del modelo. Esta relación se define como una especialización de las referencias Aggregates de OPC UA la cual se presenta a continuación y se define en el anexo C.

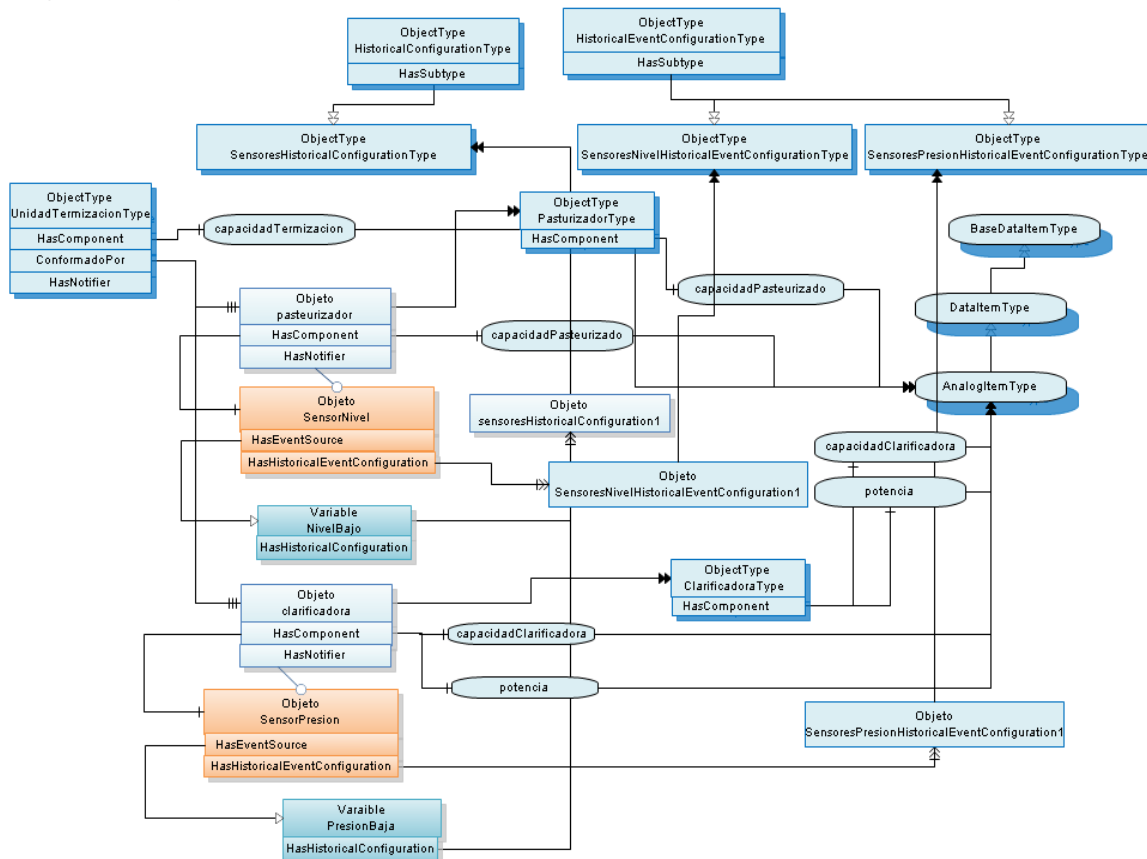
- ConformadaPor

En la figura 24 se expone parte del modelado de equipos de OPC UA, en la cual se modela la unidad de proceso de termización de la empresa caso de estudio como un ObjectType conformado por los objetos pasteurizador y una maquina clarificadora, que notifican la ocurrencia de eventos a través de las relaciones que soportan dicha generación.

En el modelado el objeto sensorNivel es un notificador de eventos con una referencia HasEventSource a la variable nivelBajo, la cual es un nodo con la capacidad de generar eventos cuando una condición anormal se presenta. La definición de los nodos se presenta a continuación y hace parte de la definición del modelo presentada en el anexo C.

En el modelado se expone la variable *AnalogItem* de la especificación de acceso a datos de OPC UA [UA Parte 8], mediante la cual se establecen los parámetros de una variable analógica del proceso, esta variable es instanciada para referenciar las variables de los equipos del proceso.

Figura 24. Ejemplo del modelo de equipos del caso de estudio.



Fuente: Elaboración propia, Agosto 2008

A continuación se expone la definición de los nodos más relevantes en la aplicación del ejemplo y de la semántica del modelo de equipos. Sin embargo la definición completa del modelo de equipos se presenta en el anexo C.

El objeto UnidadTermizacionType define las propiedades básicas de la unidad de termización del proceso UHT y los equipos con los cuales la unidad se conforma, un pasteurizador y una clarificadora quienes son instancias declaradas de la unidad de termización, y por lo tanto deberán existir con esa denominación en todas las instancias del objeto UnidadTermizacionType.

Este objeto tipo no tiene la propiedad de soportar eventos o históricos, por lo tanto en su definición no se incluyen referencias para la creación de estos datos. La definición del objeto se presenta en la tabla 24.

Tabla 24. Definición del objeto UnidadTermizacionType.

Atributo	Valor				
NodeClass	5030				
BrowseName	UnidadTermizacionType				
DisplayName	Unidad de Termización				
Description	Agrupa a los elementos con características similares a las unidades de Termización				
IsAbstract	False				
Referencias	NodeClass	BrowseName	Data Type	Type Definition	Modelling Rule
Subtipo del ObjectType BaseObjectType.					
HasComponent	Variable	capacidadTermizacion	Number	AnalogItemType	New
ConformadoPor	Object	pasteurizador	--	PasteurizadorType	New
ConformadoPor	Object	clarificadora	--	ClarificadoraType	New
ConformadoPor	Object	intercambiadorPlacas	--	IntercambiadorType	New
Nota – 1) Notas					

Fuente: Elaboración propia, Agosto 2008

El objeto pasteurizador del modelado representa el equipo de proceso de la unidad de termización que tiene la propiedad de soportar la creación de eventos. A continuación se presenta el método de configuración de la propiedad EventNotifier del objeto.

- Campo 0: **1**. Indica que puede ser utilizado para suscribirse a eventos.
- Campo 1: **0**. Reservado para futuros requerimientos y siempre debe ser cero.
- Campo 2: **1**. Indica que el historial de los eventos se puede leer
- Campo 3: **1**. Indica que el historial de los eventos se puede escribir
- Campo 4: **0**. Reservado para futuros requerimientos y siempre es cero.



- Campo 5: **0**. Reservado para futuros requerimientos y siempre es cero.
- Campo 6: **0**. Reservado para futuros requerimientos y siempre es cero.
- Campo 7: **0**. Reservado para futuros requerimientos y siempre es cero.

El *ObjectType HistoricalConfigurationType* hace parte de la definición del estándar UA para acceso a históricos, este objeto establece las características generales de un nodo con la configuración histórica para una variable. El diseño para la aplicación propuesto define el objeto tipo *SensoresHistoricalConfigurationType* específico para el registro de datos históricos de los sensores del proceso, su definición se presenta en la tabla 25.

Tabla 25. Definición del *ObjectType* SensoresHistoricalConfigurationType.

Atributo	Valor				
BrowseName	SensoresHistoricalConfigurationType				
IsAbstract	False				
Referencias	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Subtipo del objeto HistoricalConfigurationType.					
Nota – 1) Notas					

Fuente: Elaboración propia, Agosto 2008

El *ObjectType HistoricalEventConfigurationType* establece las características para que un objeto o una vista almacene eventos históricos. Con el propósito de especificar un objeto que colecciona eventos históricos específicos a la información proveniente de los sensores se define el *ObjectType SensoresNivelHistoricalEventConfigurationType* el cual es un subtipo del objeto base definido por el estándar UA. La definición del objeto se presenta en la tabla 26.

Tabla 26. Definición del *SensoresNivelHistoricalEventConfigurationType*

Atributo	Valor		
BrowseName	SensoresNivelHistoricalEventConfigurationType		
IsAbstract	False		
Referencias	NodeClass	BrowseName	ModellingRule
Subtipo del objeto HistoricalEventConfigurationType.			

Fuente: Elaboración propia, Agosto 2008

El *ObjectType SensoresHistoricalConfigurationType* es un subtipo del *HistoricalConfigurationType* el cual es definido por OPC UA para el registro y almacenamiento de datos en el tiempo. El propósito de realizar los subtipos del objeto

base es definir un tipo específico o concreto para el modelo de equipos que estructure la información. La definición del objeto se presenta en el anexo C.

El objeto *SensorNivel* es una instancia del objeto tipo *SensorNivelType* el cual no se presenta en la figura, este objeto representa el sensor para la medición de nivel, el cual es un componente de la unidad de termización y por lo tanto relacionado a través de la referencia *HasComponent*. Su definición se presenta en el anexo C.

La referencia “*ConformadaPor*” expresa la relación de agregación que tienen las unidades de proceso con los equipos. Este tipo de referencia es concreta y por lo tanto puede ser utilizada directamente en el *AddressSpace*. Esta es un subtipo de la referencia *Aggregates* de OPC UA, que relaciona objetos u *ObjectTypes* con los objetos y variables, o para relacionar las variables o variables complejas con sus *DataVariables*. La definición se presenta a continuación.

Si el *TargetNode* es una *Variable*, el *SourceNode* puede ser una variable, un Objeto, un *ObjectType*, una *DataVariable* o una *VariableType*.

La definición formal de esta referencia se presenta en la tabla 27.

Tabla 27. Definición de la *ReferenceType* Contiene

Atributos	Valor		
BrowseName	ConformadaPor		
InverseName	HaceParteDe		
Symmetric	False		
IsAbstract	False		
Referencias	NodeClass	BrowseName	Comentario
Subtipo de la referencia <i>Aggregates</i> definida por OPC UA.			

Fuente: Elaboración propia, Agosto 2008

4.3.3.2 *AddressSpace* del servidor a nivel de gestión. El *AddressSpace* del servidor a nivel de gestión propuesto para el caso de estudio estructura la información de definición de producto y del programa de producción, el propósito es modelar la relación que existe entre la información de materiales, los recursos de producción y la definición de un producto en el *AddressSpace* del servidor.

A. Modelo de definición de producto [14]. Este modelo establece las recetas para elaborar los productos en sus diferentes presentaciones, definiendo los procedimientos y los recursos necesarios. La información del modelo hace parte del servidor a nivel de gestión que está encargado de administrar la oferta de productos al mercado. Para la

aplicación del modelo se han identificado los nodos de la capa de tipos del caso de estudio y los tipos de relaciones con el propósito de determinar la semántica del modelo a partir de la cual se instancia el funcionamiento del proceso, posteriormente se identifican y se definen el conjunto de instancias declaradas.

La aplicación del modelo se presenta en la figura 25, en ella se expone los nodos identificados en el segmento de producto milking chocolate del caso de estudio para la capa de tipos del AddressSpace.

A.1 Nodos de la capa de tipos del modelo de definición de producto. Los nodos de la capa de tipos del modelo de definición de producto establecen la semántica para la definición de los segmentos de producto del caso de estudio, cada segmento de producto define los parámetros de los productos, así como los recursos necesarios para la producción. En el caso de estudio, los nodos de la capa de tipos identificados se listan a continuación y se definen en el anexo D.

- SegmentoProductoCelulaType
- SegmentoProductoType
- ParametroProductoType
- MaterialType
- EquipoType
- DuracionType

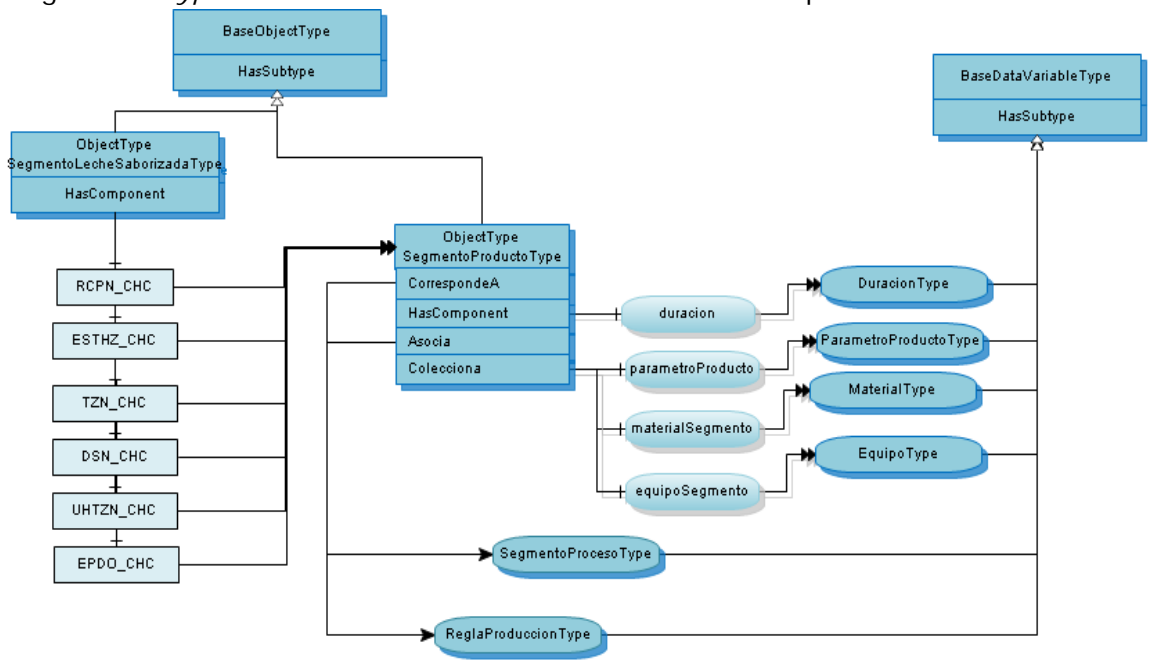
A.2 Nodos de la capa de referencias del modelo de definición de producto. La semántica del modelo define un conjunto de relaciones adicionales a las definidas por el estándar OPC UA, las cuales describen la semántica entre los objetos y las variables del modelo. A continuación se listan el conjunto de referencias identificadas y se definen en el anexo D.

- CorrespondeA
- Asocia
- Colecciona

A.3 Instancias declaradas del modelo de definición de producto. El modelo de definición de producto a nivel de célula de proceso se compone de un conjunto de instancias de segmento de producto, las cuales en el caso de estudio corresponden uno a uno con los segmentos de proceso y con el producto milking chocolate. A continuación se presenta el conjunto de instancias definidas para el caso de estudio.

- RCPN\_CHC
- ESTHZ\_CHC
- TZN\_CHC
- DSN\_CHC
- UHTZN\_CHC
- EPDO\_CHC

Figura 25. *TypeDefinitionNodes* del modelo de definición de producto.



Fuente: Elaboración propia, Agosto 2008

La variable tipo ParametroProductoType define los nombres y los tipos de la información que es enviada al nivel de control de procesos, para establecer los parámetros de un producto. Esta variable tipo compleja describe dichos parámetros, a

través de propiedades simples de OPC UA, para cada una de las reglas de producción y por cada segmento de proceso. La definición de la variable se presenta en la tabla 28.

Tabla 28. Definición del VariableType ParametroProductoType

Atributo	Valor				
BrowseName	ParametroProductoType				
IsAbstract	False				
ArraySize	-1				
DataType	ParametroProductoDataType				
Referencias	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	nombre	String	HasProperty	New
HasProperty	Variable	valor	BaseDataType	HasProperty	New
HasComponent	Variable	unidadMedida	EUInformation	HasProperty	New
Nota –					
1) Notas					

Fuente: Elaboración propia, Agosto 2008

El objeto tipo SegmentoProductoCelulaType define las propiedades de un segmento de producto visto desde la perspectiva de un segmento a nivel de célula de proceso. Para su definición el objeto está conformado por las instancias declaradas de los segmentos de producto analizados por cada segmento de proceso. La definición del objeto tipo se presenta en la tabla 29.

Tabla 29. Definición del objeto SegmentoProductoCelulaType.

Atributo	Valor				
NodeId					
NodeClass	ObjectType				
BrowseName	SegmentoProductoCelulaType				
DisplayName	Segmento de Producto.				
Description	Este segmento mapea los segmentos de proceso a nivel de célula de proceso. Para nuestro caso asocia el segmento de proceso de leche saborizada. Para el segmento de producto de leche saborizada existe los segmentos de producto definidos a partir del tipo SegmentoProductoType.				
IsAbstract	False				
Referencias	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
Subtipo del BaseObjectType.					
Asocia	variable	reglaProduccion	NodeId	PropertyType	New
HasComponent	Object	segmentoProductoRecepcion	--	SegmentoProductoType	New
HasComponent	Object	segmentoProductoE	--	SegmentoProd	New

		standarizacion		uctoType	
HasComponent	Object	segmentoProductoT ermizacion	--	SegmentoProd uctoType	New
HasComponent	Object	segmentoProductoP asterizacion	--	SegmentoProd uctoType	New
HasComponent	Object	segmentoProductoU ltrapasterizacion	--	SegmentoProd uctoType	New
HasComponent	Object	segmentoProductoE mpacado	--	SegmentoProd uctoType	New
Nota – 1) Notas					

Fuente: Elaboración propia, Agosto 2008

A través de las referencias a las variables que tiene definidas, el objeto tipo *SegmentoProductoType* especifica los valores y las cantidades necesarias de materiales y equipos de un producto específico. Este es un esquema general para la instanciación de cada uno de los segmentos del proceso para la producción de milking chocolate del caso de estudio. La definición del objeto se presenta en la tabla 30.

Tabla 30. Definición del *ObjectType* *SegmentoProductoType*

Atributo	Valor				
NodeId					
NodeClass	ObjectType				
BrowseName	SegmentoProductoType				
DisplayName	SegmentoProductoAsociadoProceso.				
Description	Este segmento de producto de leche saborizada mapea los segmentos de proceso. Sin embargo este detalla los recursos utilizados y el valor de las propiedades y los parámetros que se requieren por cada segmento para la producción.				
IsAbstract	False				
Referencias	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
Subtipo del Base <i>ObjectType</i> .					
HasComponent	Variable	duracion	DuracionDataty pe	DuracionType	New
CorrespondeA	Variable	segmentoProceso	NodeId	HasProperty	New
Asocia	Variable	idReglaProduccion	NodeId	HasProperty	New
Colecciona	Variable	parametroProducto	ParametroData Type	ParametroType	New
Colecciona	Variable	materialSegmento	MaterialDataTy pe	MaterialType	New
Colecciona	Variable	equipoSegmento	EquipoDataTyp e	EquipoType	New
Nota – 1) Notas					

Fuente: Elaboración propia, Agosto 2008

B. Modelo del programa de producción [14]. Este modelo registra la información que un programa de producción genera en sus operaciones, por ejemplo las cantidades a producir, la disponibilidad de materiales y de equipos, así como los tiempos para producir. Para la aplicación del modelo se identifican los nodos de la capa de tipos y los tipos de referencias del caso de estudio, con el propósito de determinar la semántica del modelo a partir de la cual se instancia el funcionamiento del proceso. Posteriormente se identifican el conjunto de instancias declaradas del caso de estudio.

La figura 26 expone parte de la semántica del modelado aplicado a partir de los nodos identificados para el caso de estudio, la definición de los nodos del modelo se presentan en el anexo E.

B.1 Nodos de la capa de tipos del modelo del programa de producción. A continuación se presentan los nodos identificados para la capa de tipos del modelo del caso de estudio las cuales se definen en el anexo E.

- ProgramaProducciónType
- SolicitudProduccionType
- RequerimientoSegmentoType
- ParametroProduccionType
- RequerimientoEquipoType
- RequerimientoMaterialProducidoType
- RequerimientoMaterialConsumidoType
- PropiedadType
- DuracionType

B.2 Nodos de la capa de referencias del modelo del programa de producción. Para realizar la estructura de los nodos y de la semántica del modelo se hace uso de las relaciones definidas por OPC UA y de un conjunto de referencias diseñadas en el caso de estudio las cuales se presentan a continuación y se definen en el anexo E.

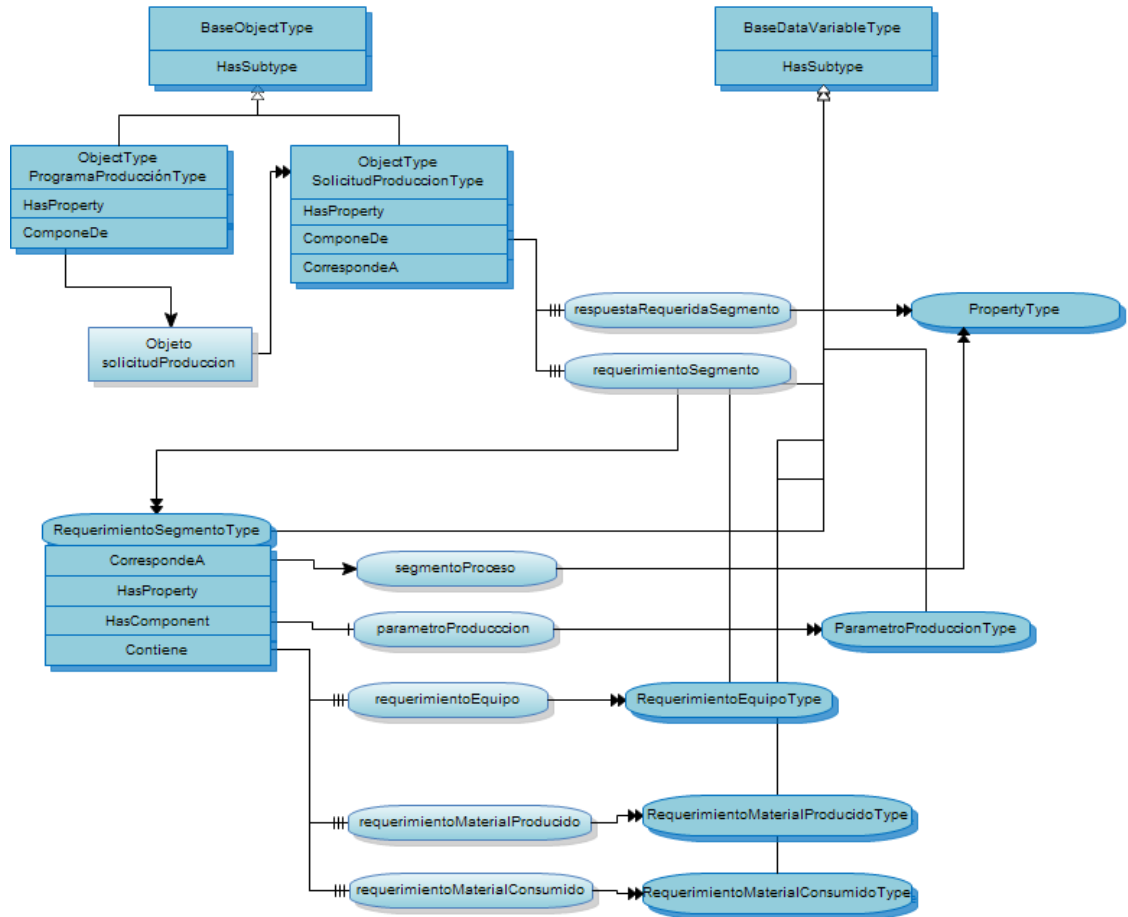
- ComponeDe
- CorrespondeA
- Contiene

B.3 DataTypes del modelo del programa de producción. Los tipos de datos aunque no hacen parte directa del AddressSpace, se definen por que cada variable del sistema asocia un tipo dato específico. Las variables tipos del caso de estudio asocian el siguiente conjunto de datos.

- RequerimientoSegmentoDataType
- ParametroProduccionDataType
- RequerimientoEquipoDataType
- RequerimientoMaterialProducidoDataType
- RequerimientoMaterialConsumidoDataType
- PropiedadDataType
- DuracionDataType



Figura 26. Representación del AddressSpace del modelo programa de producción del caso de estudio.



Fuente: Elaboración propia, Agosto 2008

El **ObjectType ProgramaProduccionType** estructura la información de programación de operaciones de manufactura, la cual consiste de un conjunto de parámetros asociados a una o más solicitudes de producción. La definición de sus propiedades y de sus tipos se presenta en la tabla 31.

Tabla 31. Definición del objeto **ProgramaProduccionType**

Atributo	Valor
NodeId	
NodeClass	ObjectType
BrowseName	ProgramaProducciónType
DisplayName	Programa de producción
Description	Contiene la información del programa de producción. Las propiedades del documento se especifican en la tabla. Este objeto contiene las solicitudes de producción.

IsAbstract	False				
Referencias	NodeClass	BrowseName	Data Type	Type Definition	Modelling Rule
Subtipo del BaseObject Type.					
HasProperty	Variable	tiempoInicio	Time	PropertyType	New
HasProperty	Variable	tiempoFinal	Time	PropertyType	New
HasProperty	Variable	fechaCreado	UtcTime	PropertyType	New
HasProperty	Variable	localizacion	string	PropertyType	New
HasProperty	Variable	tipoElemento	String	PropertyType	New
ComponeDe	Objeto	solicitudProduccion	--	SolicitudProduccionType	New
Nota –					
1) Notas					

Fuente: Elaboración propia, Agosto 2008

El Object Type SolicitudProduccionType identifica las solicitudes de producción que conforman un programa de producción, cada solicitud asocia una regla de producción, una respuesta y unos requerimientos específicos por cada segmento de proceso identificado. La definición del objeto y de sus tipos se presenta en la tabla 32.

Tabla 32. Definición del objeto SolicitudProduccionType.

Atributo	Valor				
NodeId					
NodeClass	ObjectType				
BrowseName	SolicitudProduccionType				
DisplayName	Solicitud de producción				
Description	Es objeto es una solicitud de ejecución de un batch de Milking.				
IsAbstract	False				
Referencias	NodeClass	BrowseName	Data Type	Type Definition	Modelling Rule
Subtipo del BaseObject Type.					
PuedeCorresponderA	Variable	idReglaProduccion	NodeId	PropertyType	New
HasProperty	Variable	tiempoInicio	UtcTime	PropertyType	New
HasProperty	Variable	tiempoFinal	UtcTime	PropertyType	New
HasProperty	Variable	prioridad	String	PropertyType	New
ComponeDe	Variable	requerimientoSegmento	RequerimientoSegmentoDataType	RequerimientoSegmentoType	New
ComponeDe	Variable	respuestaRequeridaSegmento	RespuestaRequeridaSegmentoType	RespuestaRequeridaSegmentoType	OptionalNew
Nota –					
1) Notas					

Fuente: Elaboración propia, Agosto 2008

La variable tipo `RequerimientoSegmentoType` describe los parámetros y los requerimientos por cada segmento de proceso, como complemento a cada una de las solicitudes de producción. La definición de los parámetros y de los requerimientos se presenta en la tabla 33.

Tabla 33. Definición de la variable `RequerimientoSegmentoType`

Atributo	Valor				
BrowserName	RequerimientoSegmentoType				
IsAbstract	False				
ArraySize	-1				
DataType	RequerimientoSegmentoDataType				
Referencias	NodeClasses	BrowserName	DataType	TypeDefinition	Modelling Rule
Subtipo del <code>BaseDataVariableType</code> .					
CorrespondeA	Variable	segmentoProceso	NodeId	PropertyType	New
HasProperty	Variable	tiempoInicioMinimo	UtcTime	PropertyType	New
HasProperty	Variable	tiempoFinalMaximo	UtcTime	PropertyType	New
HasProperty	Variable	duracion	DuracionDataType	DuracionType	New
HasComponent	Variable	parametroProduccion	ParametroDataType	ParametroType	New
PuedeContener	Variable	requerimientoEquipo	RequerimientoDataEquipoType	RequerimientoEquipoType	New
PuedeContener	Variable	requerimientoMaterialProducido	RequerimientoDataMaterialProducidoType	RequerimientoMaterialProducidoType	New
PuedeContener	Variable	requerimientoMaterialConsumido	RequerimientoDataMaterialConsumidoType	RequerimientoMaterialConsumidoType	New
Nota – 1) Notas					

Fuente: Elaboración propia, Agosto 2008

Las referencias creadas para estructurar la capa de tipos, parten de la definición de OPC UA y se extienden para definir nuevas referencias en este modelo.

La referencia `ComponeDe` es un tipo de referencia concreta, subtipo de la referencia `Aggregates`. Su semántica indica una relación de composición "Se compone de". Por lo tanto el `TargetNode` de esta referencia es una parte del `SourceNode`. La definición de la referencia se presenta en la tabla 34.

Tabla 34. Definición de la ReferenceType ComponeDe

Atributos	Valor		
BrowseName	ComponeDe		
InverseName	HaceParteDe		
Symmetric	False		
IsAbstract	False		
Referencias	NodeClass	BrowseName	Comentario
Subtipo de la referencia Aggregates.			

Fuente: Elaboración propia, Agosto 2008

La referencia Contiene es un tipo de referencia concreta, subtipo de la referencia Aggregates. Su semántica indica una relación de composición "Contiene un". Por lo tanto el TargetNode hace parte del SourceNode. Esta referencia es utilizada para relacionar objetos u ObjectTypes con los objetos y variables. Así como variables o variables complejas con sus DataVariables. La definición de esta relación se presenta en la tabla 35.

Tabla 35. Definición de la ReferenceType PuedeContener

Atributos	Valor		
BrowseName	Contener		
InverseName	ContenidoPor		
Symmetric	False		
IsAbstract	False		
Referencias	NodeClass	BrowseName	Comentario
Subtipo de la referencia Aggregates.			

Fuente: Elaboración propia, Agosto 2008

#### 4.3.4 Codificación de los modelos de Información del caso de estudio en el esquema XML UA.

Posterior al modelado del caso de estudio aplicando los principios del estándar OPC UA se realiza la codificación de los modelos siguiendo la especificación [UA parte 6] del estándar OPC UA. A continuación se presentan los pasos realizados para la codificación de la información del caso de estudio.

En el literal 4.3.6.1 se presentan los ejemplos de la codificación a diferentes tipos de nodos del caso de estudio y en los documentos anexados se incluye los esquemas y la codificación completa de cada uno de los modelos del caso de estudio.

4.3.4.1 Realizar los esquemas para la codificación del caso de estudio. Para realizar la codificación del modelo caso de estudio, el proyecto propone la construcción de esquemas XML UA que cumplan con la recomendación del estándar OPC UA y se deriven de los esquemas B2MML del WBF el cual implementa los modelos ISA-95 utilizados en el caso de estudio. El propósito es generar los modelos estándar B2MML2UA-V01 que describan la información que cumpla con los estándares relacionados a partir del cual se instancia los modelos caso de estudio.

La tabla 36 presenta el esquema y la codificación XML del modelo de materiales B2MML.

Tabla 36. Esquema XML de la clase Materiales a partir del estándar B2MML.

diagram	
elementos	<a href="#">ID</a> <a href="#">Description</a> <a href="#">MaterialClassProperty</a> <a href="#">MaterialDefinitionID</a> <a href="#">Any</a>
Utilizado por	elements <a href="#">MaterialClass</a> <a href="#">MaterialInformationType/MaterialClass</a>
Codigo XMI	<pre> &lt;xsd:complexType name="MaterialClassType"&gt;   &lt;xsd:sequence&gt;     &lt;xsd:element name="ID" type="IDType"/&gt;     &lt;xsd:element name="Description" type="DescriptionType" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;xsd:element name="MaterialClassProperty" type="MaterialClassPropertyType" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;xsd:element name="MaterialDefinitionID" type="MaterialDefinitionIDType" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;!-- added for v0300 --&gt;     &lt;xsd:group ref="Extended:MaterialClass" minOccurs="0"/&gt;     &lt;xsd:element name="Any" type="AnyType" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xsd:sequence&gt; &lt;/xsd:complexType&gt; </pre>

Fuente: Elaboración propia, Agosto 2008

La tabla 37 presenta el esquema y la codificación XML del modelo de materiales mapeado al estándar OPC UA desde el modelo B2MML a partir del cual se realizan las instancias del caso de estudio.

La definición de la clase materiales se realiza a partir de la extensión del objeto ObjectTypeNode el cual hace parte de la definición del estándar OPC UA. Para que cumpla con la definición de los esquemas B2MML a este objeto instanciado se le agregan los elementos que complementan su definición a través de referencias a los elementos que lo especializan. Los tipos datos para los objetos y las variables de los esquemas propuestos son definidos a partir de la definición en el estándar OPC UA.

Tabla 37. Esquema XML de Materiales mapeado a OPC UA

Diagrama	<p>The diagram illustrates the class structure for MaterialClassType. It is an extension of the base class ua:ObjectTypeNode. The extension includes several elements: ua:Nodeid, ua:NodeClass, ua:BrowseName, ua:DisplayName, ua:Description, and ua:References. Additionally, it includes an element ua:IsAbstract. The diagram also shows references to other classes: MaterialClassProperty (0..∞), MaterialDefinitionID (0..∞), Extended:MaterialClass (0..∞), and Any (0..∞).</p>
tipo	extension of <a href="#">ua:ObjectTypeNode</a>
propiedades	base <a href="#">ua:ObjectTypeNode</a>
elementos	<a href="#">ua:Nodeid</a> <a href="#">ua:NodeClass</a> <a href="#">ua:BrowseName</a> <a href="#">ua:DisplayName</a> <a href="#">ua:Description</a> <a href="#">ua:References</a> <a href="#">ua:IsAbstract</a> <a href="#">MaterialClassProperty</a> <a href="#">MaterialDefinitionID</a> <a href="#">Any</a>
Utilizado por	elements <a href="#">MaterialClass</a> <a href="#">MaterialInformationType/MaterialClass</a>
Código XML	<pre>&lt;xsd:complexType name="MaterialClassType"&gt;   &lt;xsd:complexContent&gt;     &lt;xsd:extension base="ua:ObjectTypeNode"&gt;       &lt;xsd:sequence&gt;         &lt;xsd:element name="MaterialClassProperty" type="MaterialClassPropertyType"           minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xsd:element name="MaterialDefinitionID" type="MaterialDefinitionIDType"           minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xsd:group ref="Extended:MaterialClass" minOccurs="0"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:extension&gt;   &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt;</pre>

	<pre> &lt;xsd:element name="Any" type="AnyType" minOccurs="0" maxOccurs="unbounded"/&gt; &lt;/xsd:sequence&gt; &lt;/xsd:extension&gt; &lt;/xsd:complexContent&gt; &lt;/xsd:complexType&gt; </pre>
--	---

Fuente: Elaboración propia, Agosto 2008

#### 4.3.4.2 Instanciación de los esquemas propuestos para la codificación.

La instanciación de los esquemas propuestos se realiza para aplicar los modelos de información del caso de estudio. La codificación de los documentos XML instanciados se presentan completamente para cada uno de los modelos en los anexos del proyecto.

#### 4.3.5 Validación de los esquemas XML de OPC UA y de los documentos instanciados en el caso de estudio.

La validación se realiza sobre los esquemas XML de B2MML mapeados al estándar OPC UA los cuales son instanciados para realizar la aplicación sobre los modelos de información del caso de estudio. Para realizar el mapeo se han tenido en cuenta las recomendaciones de las especificaciones [UA parte 3], [UA parte 5] y [UA parte 8] en el modelado de los esquemas y para realizar la codificación de los datos se siguió la recomendación de la especificación [UA parte 6]. Los resultados se presentan en formato digital, y se presentan a continuación.

Esquemas propuestos de los modelos de información a OPC UA

- B2MML2UA-V01-Material.xsd
- B2MML2UA-V01-Equipment.xsd
- B2MML2UA-V01-ProductDefinition.xsd
- B2MML2UA-V01-ProductionSchedule.xsd

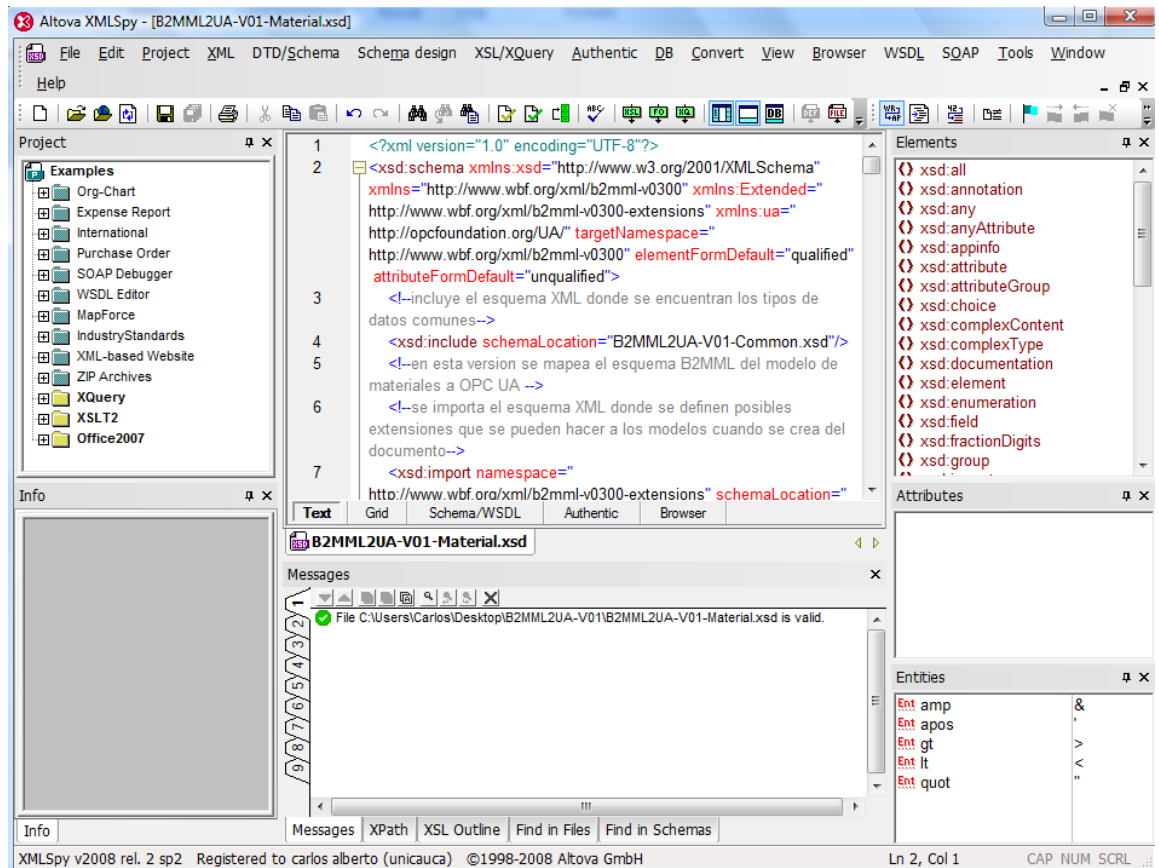
Instancias de los esquemas propuestos al caso de estudio.

- Instance-B2MML2UA-V01-Material.xml
- Instance-B2MML2UA-V01- Equipment.xml
- Instance-B2MML2UA-V01- ProductDefinition.xml
- Instance-B2MML2UA-V01- ProductionSchedule.xml

Para validar los esquemas y las instancias de los modelos se utilizo la herramienta ALTOVA XMLSpy Enterprise Edition 2008 a través de la cual se comprobó que los esquemas y los documentos cumplieran con las recomendaciones de la W3C y las

especificaciones OPC UA. La figura 27 presenta la validación del esquema propuesto para el modelo de materiales UA.

Figura 27. Interfaz de validación del esquema XML del modelo de materiales en OPC UA.



Fuente: Elaboración propia, Agosto 2008

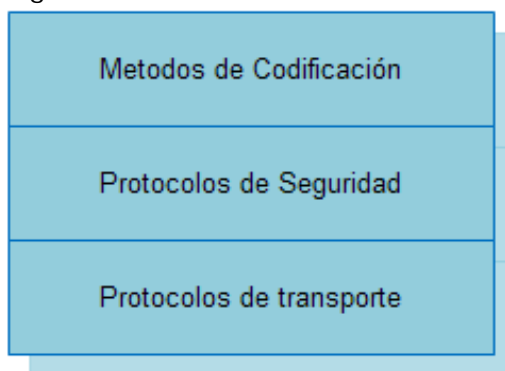
Este método de validación fue realizado para todos los esquemas y los documentos del caso de estudio en los cuales se comprobó la calidad de los esquemas y los documentos instanciados del caso de estudio. De esta manera se espera que esta propuesta aporte a la integración y la implementación de los estándares de manufactura, beneficiando los proyectos de automatización industrial con esquemas estándares de información.



#### 4.3.6 Pautas técnicas para la comunicación entre los servidores OPC UA del nivel de gestión y control de procesos.

Para establecer las pautas que definen las unidades de conformidad del perfil del servidor OPC UA se consideran y se establecen las características técnicas sobre el cual cada sistema desempeñara su función. Para el caso de estudio los requerimientos de interoperabilidad tienen mayor prioridad que la velocidad de respuesta que se podría dar en la comunicación de los equipos en un proceso, ya que la información en los niveles del caso de estudio maneja grandes volúmenes de datos que son utilizados para ser administrados dentro del área de gestión. Para establecer las características técnicas del servidor se definen las unidades de conformidad del stack de comunicaciones del estándar OPC UA presentado en la figura 28.

Figura 28. Stack de comunicaciones de OPC UA



Fuente: Elaboración propia, Agosto 2008

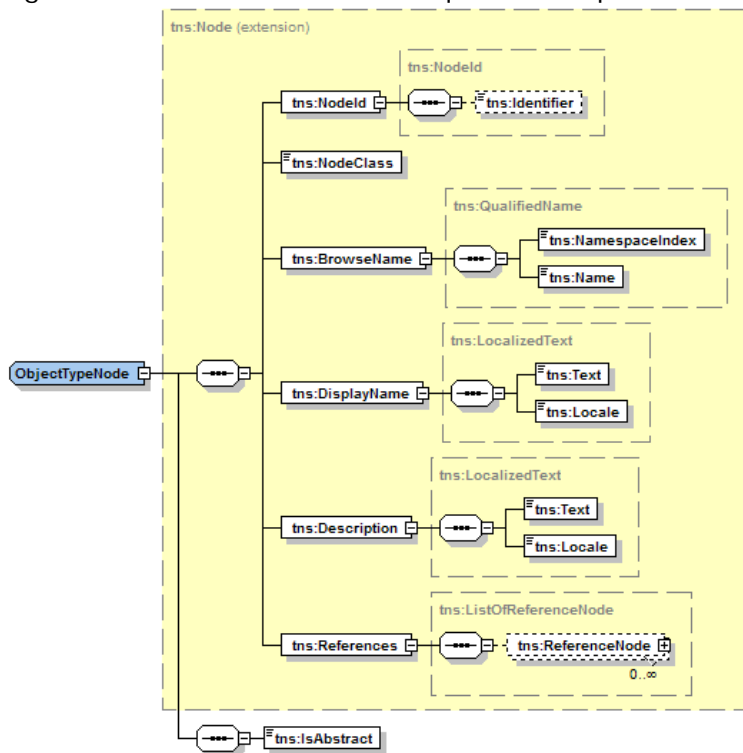
Como se ilustra en la figura 28, el stack está constituido por tres capas básicas que definen los métodos de codificación, los protocolos de seguridad, y los protocolos de transporte. Los métodos y los protocolos son definidos como unidades de conformidad las cuales en conjunto con el modelo de información definen el perfil del servidor.

4.3.6.1 Unidad de conformidad para el método de codificación del modelo de datos para el servidor OPC UA. Los métodos de codificación permiten convertir el lenguaje de modelamiento del AddressSpace en los lenguajes manejados por el servidor OPC UA para la comunicación en red. Para que esta se realice de forma estándar la OPC Foundation brinda los esquemas de codificación de datos y de mensajes los cuales son la base para desarrollar la codificación de los nodos del modelo.

El esquema de codificación XML es una sintaxis basada en XML para la descripción de datos codificados como texto XML. Las reglas para la codificación de mensajes UA XML es un esquema del estándar que define la estructura de todos los tipos de datos y mensajes. Este esquema captura todos los nombres de la normativa para los tipos y sus campos así como el orden en que ellos deben aparecer []. Por lo tanto la unidad de conformidad de la capa de codificación se define con el esquema XML de OPC UA.

La figura 29 presenta el esquema XML para la codificación de *ObjectTypes* propuesto en los documentos para descripción de servicios WSDL de OPC UA. Este esquema es estándar por lo tanto se aplica a todos los objetos tipos identificados para el caso de estudio.

Figura 29. Vista del diseño del esquema XML para la NodeClass *ObjectType*



Fuente: Elaboración propia, Agosto 2008

Con el propósito de exponer la aplicación del esquema, en la tabla 38 se presenta la aplicación de la codificación del objeto tipo LacteoType del modelo de materiales.

Tabla 38. Codificación del objeto tipo LacteoType del caso de estudio

<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; - &lt;!-- edited with XMLSpy v2008 sp1 (http://www.altova.com) by Tesis (OPC UA) --&gt; - &lt;!-- Sample XML file generated by XMLSpy v2008 sp1 (http://www.altova.com) --&gt;   &lt;tns:ObjectTypeNode xsi:schemaLocation="http://opcfoundation.org/UA/ Types.xsd"   xmlns:tns="http://opcfoundation.org/UA/"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;   tns:NodeId&gt;     &lt;tns:Identifier&gt;LacteoType&lt;/tns:Identifier&gt;     &lt;/tns:NodeId&gt;     &lt;tns:NodeClass&gt;ObjectType_2&lt;/tns:NodeClass&gt;   &lt;tns:BrowseName&gt;     &lt;tns:NamespaceIndex&gt;0&lt;/tns:NamespaceIndex&gt;     &lt;tns:Name&gt;LacteoType&lt;/tns:Name&gt;     &lt;/tns:BrowseName&gt;   &lt;tns:DisplayName&gt;     &lt;tns:Text&gt;LacteoType&lt;/tns:Text&gt;     &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt;     &lt;/tns:DisplayName&gt;   &lt;tns:Description&gt;     &lt;tns:Text&gt;Grupo de alimentos que abarca la leche y sus derivados. Se caracteriza por aportar     gran cantidad de calcio, proteínas y vitaminas.&lt;/tns:Text&gt;     &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt;     &lt;/tns:Description&gt;   &lt;tns:References&gt;   &lt;tns:ReferenceNode&gt;   &lt;tns:ReferenceTypeId&gt;     &lt;tns:Identifier&gt;HasProperty_10&lt;/tns:Identifier&gt;     &lt;/tns:ReferenceTypeId&gt;     &lt;tns:IsInverse&gt;true&lt;/tns:IsInverse&gt;     &lt;tns:ServerIndex&gt;0&lt;/tns:ServerIndex&gt;   &lt;tns:TargetId&gt;     &lt;tns:Identifier&gt;especie&lt;/tns:Identifier&gt;     &lt;tns:NamespaceURI&gt;EmpresaCasoEstudio&lt;/tns:NamespaceURI&gt;     &lt;/tns:TargetId&gt;     &lt;/tns:ReferenceNode&gt;   &lt;tns:ReferenceNode&gt;   &lt;tns:ReferenceTypeId&gt;     &lt;tns:Identifier&gt;HasProperty_10&lt;/tns:Identifier&gt;     &lt;/tns:ReferenceTypeId&gt;     &lt;tns:IsInverse&gt;true&lt;/tns:IsInverse&gt;     &lt;tns:ServerIndex&gt;0&lt;/tns:ServerIndex&gt;   &lt;tns:TargetId&gt;     &lt;tns:Identifier&gt;origen&lt;/tns:Identifier&gt;     &lt;tns:NamespaceURI&gt;EmpresaCasoEstudio&lt;/tns:NamespaceURI&gt;     &lt;/tns:TargetId&gt;     &lt;/tns:ReferenceNode&gt;   &lt;tns:ReferenceNode&gt;   &lt;tns:ReferenceTypeId&gt; </pre>
---

```

<tns:Identifier>HasProperty_10</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
  <tns:Identifier>proteina</tns:Identifier>
  <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
  </tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
  <tns:Identifier>HasProperty_10</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>true</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
  <tns:Identifier>lactosa</tns:Identifier>
  <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
  </tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
  <tns:Identifier>HasProperty_10</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>true</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
  <tns:Identifier>vitamina</tns:Identifier>
  <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
  </tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
  <tns:Identifier>HasProperty_10</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>true</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
  <tns:Identifier>aflotaxina</tns:Identifier>
  <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
  </tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
  <tns:Identifier>HasSubtype_10</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>true</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
  <tns:Identifier>LecheCrudaEnteraType</tns:Identifier>
  <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
  </tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>

```

```

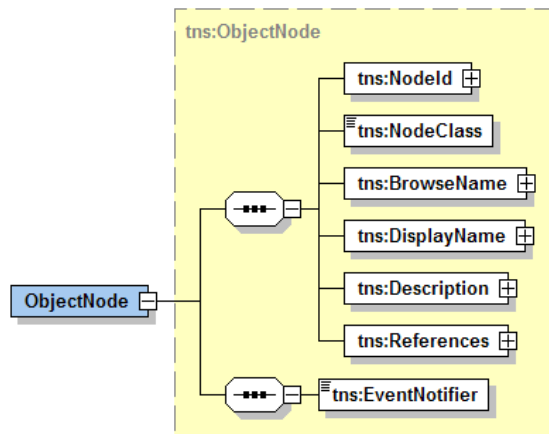
<tns:Identifier>HasSubtype_10</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
<tns:Identifier>LecheEstandarizadaHigienizadaType</tns:Identifier>
<tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
</tns:TargetId>
</tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
<tns:Identifier>HasSubtype_10</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
<tns:Identifier>LecheTermizadaType</tns:Identifier>
<tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
</tns:TargetId>
</tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
<tns:Identifier>HasSubtype_10</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
<tns:Identifier>LecheBaseUHTType</tns:Identifier>
<tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
</tns:TargetId>
</tns:ReferenceNode>
= <tns:ReferenceNode>
= <tns:ReferenceTypeId>
<tns:Identifier>HasSubtype_10</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
= <tns:TargetId>
<tns:Identifier>CremaLecheType</tns:Identifier>
<tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
</tns:TargetId>
</tns:ReferenceNode>
</tns:References>
<tns:IsAbstract>true</tns:IsAbstract>
</tns:ObjectTypeNode>

```

Fuente: Elaboración propia, Agosto 2008

La figura 30 representa el esquema de datos propuesto por el estándar OPC UA para la codificación de objetos, el cual fue tomado de los archivos para la descripción de servicios WSDL del estándar OPC UA. Por lo tanto aplica para todos los objetos del caso de estudio.

Figura 30. Esquema para la codificación de objetos del AddressSpace.



Fuente: Elaboración propia, Agosto 2008

La tabla 39 corresponde a la codificación de un objeto de la capa de instancias, el cual es un pasteurizador que soporta la generación de eventos.

Tabla 39. Codificación del objeto pasteurizador del caso de estudio

<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; - &lt;!-- Sample XML file generated by XMLSpy v2008 sp1 (http://www.altova.com) --&gt; = &lt;tns:ObjectNode xsi:schemaLocation="http://opcfoundation.org/UA/ Types.xsd" xmlns:tns="http://opcfoundation.org/UA/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt; </pre>
<pre> = &lt;tns:NodeId&gt; &lt;tns:Identifier&gt;parteurizador&lt;/tns:Identifier&gt; &lt;/tns:NodeId&gt; &lt;tns:NodeClass&gt;Object_1&lt;/tns:NodeClass&gt; = &lt;tns:BrowseName&gt; &lt;tns:NamespaceIndex&gt;1&lt;/tns:NamespaceIndex&gt; &lt;tns:Name&gt;pasterizador&lt;/tns:Name&gt; &lt;/tns:BrowseName&gt; = &lt;tns:DisplayName&gt; &lt;tns:Text&gt;pasterizador&lt;/tns:Text&gt; &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt; &lt;/tns:DisplayName&gt; = &lt;tns:Description&gt; &lt;tns:Text&gt;Instancia del Objectype PasterizadorType el cual identifica un pasteurizador del proceso que pertenece al Objectype UnidadTermizacionType&lt;/tns:Text&gt; &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt; &lt;/tns:Description&gt; = &lt;tns:References&gt; = &lt;tns:ReferenceNode&gt; = &lt;tns:ReferenceTypeId&gt; &lt;tns:Identifier&gt;HasComponent_11&lt;/tns:Identifier&gt; &lt;/tns:ReferenceTypeId&gt; </pre>

```

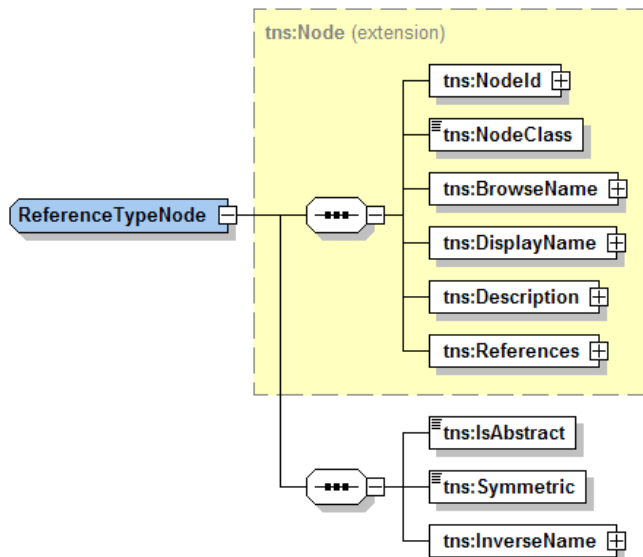
    <tns:IsInverse>true</tns:IsInverse>
    <tns:ServerIndex>0</tns:ServerIndex>
  <tns:TargetId>
    <tns:Identifier>capacidadTermizacion</tns:Identifier>
    <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
</tns:ReferenceNode>
<tns:ReferenceNode>
  <tns:ReferenceTypeId>
    <tns:Identifier>HasComponent_11</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>true</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
  <tns:TargetId>
    <tns:Identifier>SensorNivel</tns:Identifier>
    <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
</tns:ReferenceNode>
<tns:ReferenceNode>
  <tns:ReferenceTypeId>
    <tns:Identifier>HasNotifier_12</tns:Identifier>
  </tns:ReferenceTypeId>
  <tns:IsInverse>false</tns:IsInverse>
  <tns:ServerIndex>0</tns:ServerIndex>
  <tns:TargetId>
    <tns:Identifier>SensorNivel</tns:Identifier>
    <tns:NamespaceURI>EmpresaCasoEstudio</tns:NamespaceURI>
  </tns:TargetId>
</tns:ReferenceNode>
</tns:References>
<tns:EventNotifier>13</tns:EventNotifier>
</tns:ObjectNode>

```

Fuente: Elaboración propia, Agosto 2008

La figura 31 representa el esquema para la descripción de referencias tipo del AddressSpace, este hace parte del archivo para la descripción de servicios web WSDL del estándar OPC UA y por lo tanto se aplica para la codificación de todas los tipos de referencias.

Figura 31. Esquema para la codificación de referencias tipos del AddressSpace.



Fuente: Elaboración propia, Agosto 2008

La aplicación del esquema al caso de estudio, se presenta en el ejemplo de la tabla 40 el cual corresponde a la definición del tipo de referencia ConformadaPor del caso de estudio.

Tabla 40. Codificación de la referencia tipo ConformadoPor del caso de estudio

<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; - &lt;!-- Sample XML file generated by XMLSpy v2008 sp1 (http://www.altova.com) --&gt; = &lt;tns:ReferenceTypeNode xsi:schemaLocation="http://opcfoundation.org/UA/ Types.xsd" xmlns:tns="http://opcfoundation.org/UA/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt; = &lt;tns:NodeId&gt; &lt;tns:Identifier&gt;ConformadaPor&lt;/tns:Identifier&gt; &lt;/tns:NodeId&gt; &lt;tns:NodeClass&gt;ReferenceType_5&lt;/tns:NodeClass&gt; = &lt;tns:BrowseName&gt; &lt;tns:NamespaceIndex&gt;1&lt;/tns:NamespaceIndex&gt; &lt;tns:Name&gt;ConformadaPor&lt;/tns:Name&gt; &lt;/tns:BrowseName&gt; = &lt;tns:DisplayName&gt; &lt;tns:Text&gt;Se conforma por&lt;/tns:Text&gt; &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt; &lt;/tns:DisplayName&gt; = &lt;tns:Description&gt; &lt;tns:Text&gt;Se conforma por elementos del TargetNode&lt;/tns:Text&gt; &lt;tns:Locale&gt;en-US&lt;/tns:Locale&gt; &lt;/tns:Description&gt; = &lt;tns:References&gt; = &lt;tns:ReferenceNode&gt; </pre>
--



```

<tns:ReferenceTypeId>
  <tns:Identifier>String</tns:Identifier>
</tns:ReferenceTypeId>
<tns:IsInverse>true</tns:IsInverse>
<tns:ServerIndex>0</tns:ServerIndex>
<tns:TargetId>
  <tns:Identifier>String</tns:Identifier>
  <tns:NamespaceURI>String</tns:NamespaceURI>
</tns:TargetId>
</tns:ReferenceNode>
</tns:References>
<tns:IsAbstract>true</tns:IsAbstract>
<tns:Symmetric>false</tns:Symmetric>
<tns:InverseName>
  <tns:Text>HaceParteDe</tns:Text>
  <tns:Locale>en-US</tns:Locale>
</tns:InverseName>
</tns:ReferenceTypeNode>

```

Fuente: Elaboración propia, Agosto 2008

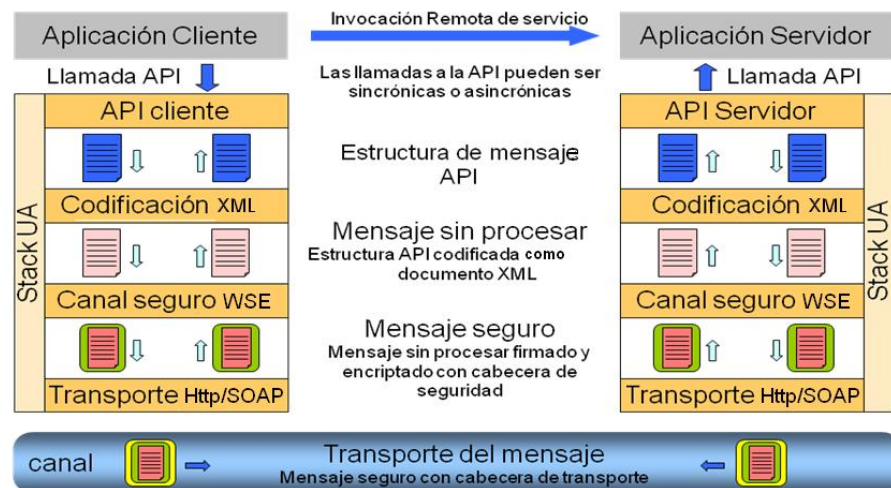
4.3.6.2 Unidad de conformidad para la capa de seguridad. Los métodos de seguridad del estándar OPC UA están diseñados para asegurar la integridad y fiabilidad de los mensajes UA.

El protocolo WS Secure Conversation hace parte de las especificaciones de la organización para la interoperabilidad de servicios WS-I. En ella los mensajes intercambiados a través del estándar para mensajería SOAP, se aseguran a través de la especificación WS-Secure Conversation, para ello el protocolo define la manera de negociar la información a través de la especificación WS-Trust, asegurando el intercambio de mensajes con los mecanismos definidos en la especificación WS-Security. Los mecanismos para la firmar y encriptar elementos XML son descritos en la especificación XML Signature y XML Encryption. Esta es la recomendación de la W3C bajo la cual OPC UA establece la seguridad de los datos y los mensajes. Por lo tanto ya que los mensajes XML se transmiten a través de SOAP, la unidad de conformidad para la seguridad está definida por la especificación WS Secure Conversation.

4.3.6.3 Unidad de conformidad de la capa de transporte. Con base en las unidades de conformidad definidas anteriormente, la capa de transporte del perfil del servidor se basa en el protocolo de aplicación Http, debido a que el estándar SOAP utilizado para la transmisión de mensajes XML, permite el intercambio únicamente sobre este protocolo. Por lo tanto, OPC UA sigue la recomendación SOAP versión 1.2 de la W3C.

4.3.6.4 Interacción de las unidades de conformidad de los servidores caso de estudio. El flujo de datos a través del stack de comunicaciones de los servidores se modela en la figura 32. En ella los servidores pueden asumir el rol de clientes o servidores en el momento de la comunicación.

Figura 32. Flujo de datos a través del stack de comunicaciones del caso de estudio.



Fuente: Elaboración propia, Agosto 2008

La interacción inicia cuando el cliente realiza una invocación de un servicio web al servidor UA. Para realizar la invocación, el cliente utiliza una API como interfaz de llamada al servidor, la cual es proporcionada por la OPC Foundation y permite realizar la llamada a los servicios.

Por ejemplo si el cliente quiere solicitar un servicio para hacer una lectura de un dato en el servidor, la solicitud se lee en la API la cual entrega el mensaje en el formato XML "sin procesar" a la capa llamada canal seguro implementada en W-SE donde se realiza la firma al mensaje, se encripta y se le agrega la cabecera de seguridad, este mensaje seguro pasa a la capa de transporte donde se le agrega la cabecera de transporte acorde al protocolo SOAP/HTTP, el cual finalmente se encarga de enviar el mensaje al servidor UA.

En el lado del servidor se realiza el proceso contrario, la capa de transporte lleva el mensaje con la cabecera de seguridad al canal seguro, la cual des-encripta y revisa la firma digital del mensaje y lo concerniente a su seguridad, obteniendo el mensaje "sin procesar", codificado en esquemas XML. Posteriormente el mensaje es entregado a la

siguiente capa donde es decodificado, obteniendo la estructura de mensaje API que contiene los parámetros de entrada del servicio.

4.3.6.5 Envío de mensajes SOAP entre el cliente y un servidor OPC UA. En la tabla 41 se presenta la aplicación del mensaje de solicitud SOAP realizada por el cliente para agregar nodos al AddressSpace, la cual sigue la estructura descrita en el literal anterior para llegar desde el cliente al servidor, este mensaje SOAP desarrollado cumple con las recomendaciones de la W3C y con el estándar OPC UA, y esta validado en la herramienta ALTOVA XmlSpy Enterprise Edition 2008.

Tabla 41. Mensaje SOAP para el intercambio de información entre el cliente y el servidor.

```

- <!-- edited with XMLSpy v2008 sp1 (http://www.altova.com) by Tesis (OPC UA) -->
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:AddNodeRequest xmlns:m="http://opcfoundation.org/UA/">
      <m:RequestHeader>
        <m:SessionId>0</m:SessionId>
        <m:Timestamp>2001-12-17T09:30:47.0Z</m:Timestamp>
        <m:SequenceNumber>0</m:SequenceNumber>
        <m:ReturnDiagnostics>0</m:ReturnDiagnostics>
        <m:AuditLogEntryId>String</m:AuditLogEntryId>
      </m:RequestHeader>
      <m:NodesToAdd>
        <m:AddNodeItem>
          <m:ParentNodeId>
            <m:Identifier>LacteoType</m:Identifier>
            <m:NamespaceURI>EmpresaCasoEstudio</m:NamespaceURI>
          </m:ParentNodeId>
          <m:ReferenceTypeId>
            <m:Identifier>HasSubtype_15</m:Identifier>
            <m:NamespaceURI>http://OPCFoundation.org/UA</m:NamespaceURI>
          </m:ReferenceTypeId>
          <m:RequestedNodeId>
            <m:Identifier>LecheCrudaType</m:Identifier>
            <m:NamespaceURI>EmpresaCasoEstudio</m:NamespaceURI>
          </m:RequestedNodeId>
          <m:BrowseName>
            <m:NamespaceIndex>1</m:NamespaceIndex>
            <m>Name>LecheCrudaType</m>Name>
          </m:BrowseName>
          <m:NodeClass>ObjectType</m:NodeClass>
          <m:NodeAttributes>
            <m:TypeId>
              <m:Identifier>ObjectTypeAttributes</m:Identifier>
              <m:NamespaceURI>http://OPCFoundation.org/UA</m:NamespaceURI>
            </m:TypeId>
          </m:NodeAttributes>
        </m:AddNodeItem>
      </m:NodesToAdd>
    </m:AddNodeRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

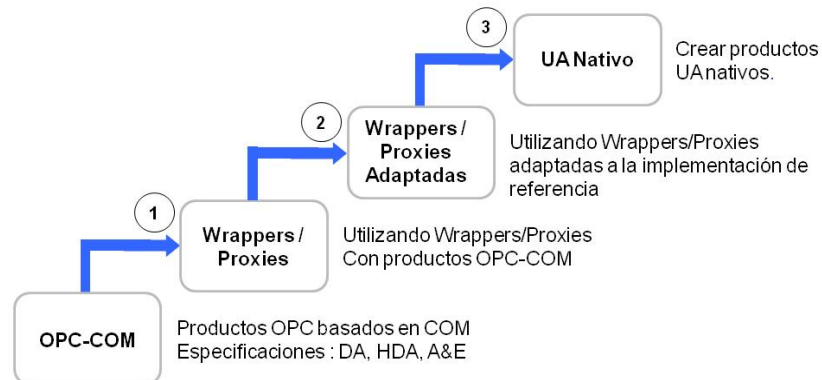
</m:TypeId>
= <m:Body>
<m:displayName>Leche cruda</m:displayName>
<m:description>Leche cruda que proviene de los hatos.</m:description>
<m:isAbstract>false</m:isAbstract>
</m:Body>
</m:NodeAttributes>
= <m:TypeDefinition>
<m:Identifier>BaseObjectType</m:Identifier>
<m:NamespaceURI>http://OPCFoundation.org/UA</m:NamespaceURI>
</m:TypeDefinition>
</m:AddNodeItem>
</m:NodesToAdd>
</m:AddNodeRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Fuente: Elaboración propia, Agosto 2008

4.3.7 Migración de la tecnología OPC COM a OPC UA en las empresas de manufactura. Como en la mayoría de las evoluciones dadas por el avance en las tecnologías, es necesario definir los pasos para llegar desde los actuales servidores OPC COM a los sistemas OPC UA, por lo tanto es el propósito presentar la ruta de migración propuesta por la OPC Foundation para realizar la migración hasta llegar completamente a los sistemas UA nativos. Dichas etapas se ilustran en la figura 33.

Figura 33. Etapas necesarias para la migración de COM-UA



Fuente: OPC Foundation UA, DevCon 2007 Migration to UA.ppt, agosto de 2008

4.3.7.1 Utilización de Wrapper/Proxies con productos basados en OPC COM [16]. En esta etapa se identifican los productos COM que se desean mapear a UA, estos pueden ser clientes COM que van a acceder datos de un servidor UA como se presenta en la figura 34, o servidores COM que van a ser accedidos por clientes UA como se presenta en la figura 35.

Las herramientas Proxies son utilizadas para que clientes OPC COM accedan a los servidores UA de manera estándar.

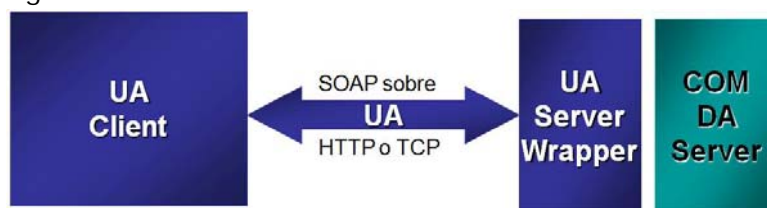
Figura 34 Cliente COM accediendo a un servidor UA.



Fuente: OPC and OPC Unified Architecture Expo, noviembre de 2007

Las Wrappers son las herramientas utilizadas para que clientes UA accedan a servidores OPC COM de forma estándar.

Figura 35. Cliente UA accediendo a un servidor DA COM.



Fuente: OPC and OPC Unified Architecture Expo, noviembre de 2007

Actualmente la OPC Foundation trabaja en el desarrollo de estas herramientas de tal manera que estas sean estándar en todas las aplicaciones UA y los datos publicados por los servidores OPC DA, HDA y A&E pueden ser mapeados fácilmente y publicados a través de UA. Es importante resaltar que estas herramientas wrapper/proxies no implementan todas las funcionalidades de los servicios UA, sin embargo se podrá acceder a gran parte de la funcionalidad [2].

Esto se hace ya que OPC UA busca ser una tecnología flexible y escalable, que puede ser adaptada a sistemas anteriores sin que los clientes tengan que hacer grandes inversiones en actualizaciones y en plataformas específicas.

A continuación se explica el funcionamiento de cada una de las herramientas.

- Proxy [16]. El proxy se utiliza cuando un cliente OPC COM se quiere comunicar con un servidor OPC UA y acceder a este. El proxy se encarga de mapear de la tecnología COM a UA, para ello la OPC Foundation ha estimado la creación de los proxies para clientes DA, A&E, HDA y XML-DA.

Un proxy DA es un servidor OPC DA COM que conecta y expone un subconjunto de datos de un servidor UA, ya que los clientes DA solo conocen de interfaces COM y los servidores UA se conectan a través de los endpoints. Por lo tanto es el encargado de gestionar las llamadas UA, conectar servidores UA con clientes COM, hacer el manejo de sesión y realizar todas las conversiones de llamadas COM a llamadas UA.

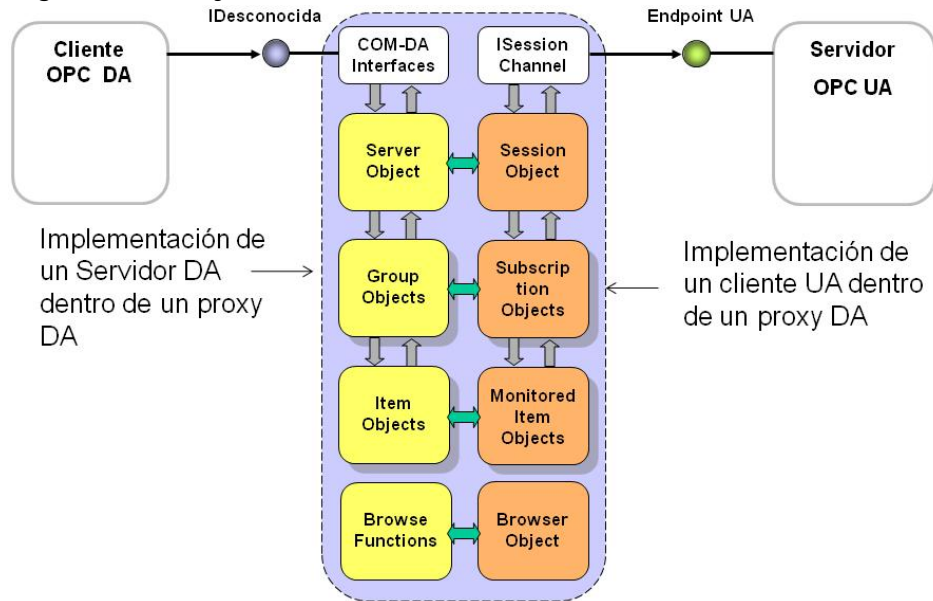
Para acceder a los objetos UA el proxy mapea la información del servidor UA de tal manera que si el cliente DA está solicitando grupos, el proxy entrega objetos del tipo FolderType, o cuando el cliente solicite ítems el proxy entrega variables del tipo DataItemType y cuando solicite sus propiedades entregue las variables del tipo PropertyType.

Para realizar el proceso de mapeo los proxies implementan en su parte interna un servidor DA y un cliente UA. El servidor DA toma las llamadas DA y crea los objetos, grupos e ítems necesarios para realizar el mapeo con los elementos del cliente UA, el cual se encarga de conectarse al endpoint del servidor y negociar la seguridad, crear la sesión y suscribir las notificaciones que el cliente considere.

Aunque el proxy brinda muchas funcionalidades para el acceso a servidores UA existen muchas limitaciones debido a que DA no maneja conceptos que se manejan en UA tales como referencias y variables semánticas generando algunas limitaciones en el acceso al AddressSpace UA. Un Proxy DA específico se presenta en la figura 36, en él un cliente DA a un lado de la comunicación y en el otro lado un servidor UA se comunican utilizando este proxy.

Cuando el proxy crea la instancia del servidor DA, en el lado del cliente UA de la misma proxy se crea automáticamente la sesión en el servidor UA.

Figura 36. Proxy Cliente DA



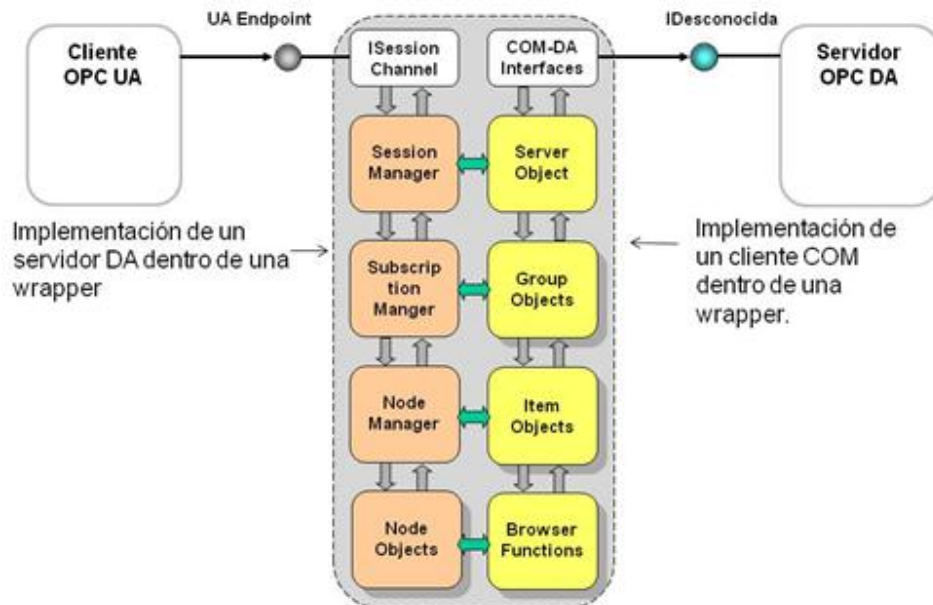
Fuente: OPC Foundation UA, DevCon 2007 Migration to UA.ppt, agosto de 2008

- Wrapper [16]. La Wrapper se utiliza cuando un cliente UA se quiere comunicar con un servidor OPC COM y acceder a este. La Wrapper al igual que el proxy se encarga de mapear de una tecnología a la otra. Existen la propuesta para desarrollar varios tipos de Wrappers para los servidores COM DA, A&E, HDA y XML-DA las cuales realicen el proceso contrario al proxy quien mapea de UA a OPC COM.

Un ejemplo de una Wrapper DA se muestra en la figura 37, la cual se encuentra como interfaz en la comunicación de un cliente UA y un servidor COM DA, la Wrapper implementa por un lado el servidor UA que se encarga de tomar las solicitudes UA, para crear la sesión y los objetos UA necesarios. Estos objetos dirigen las llamadas al servidor DA a través de un cliente DA que es implementado en la wrapper. Esta parte se conecta al servidor DA para autenticar usuarios, hacer llamadas, acceder a la información DA y realizar las consultas.

Cuando se quiere acceder a los objetos DA, la Wrapper mapea de tal manera que si el cliente UA está solicitando objetos del tipo FolderType a través de la wrapper se entrega grupos de información en DA, cuando se soliciten Variables del tipo DataItemType le entregue ítems y cuando le solicite variables del tipo PropertyType entregue las propiedades del DataItemType.

Figura 37. Wrapper servidor



Fuente: OPC Foundation UA, DevCon 2007 Migration to UA.ppt, agosto de 2008

4.3.7.2 Wrapper y proxys adaptadas a implementaciones de referencia [16]. Utilizando las herramientas que la OPC Foundation brinda a sus miembros, como son las librerías, código fuente, kits de desarrollo software, los desarrolladores de software podrán utilizar las implementaciones de referencia para personalizar las herramientas de la manera deseada.

Para este propósito la OPC Foundation y un grupo de miembros voluntarios de la fundación (The "UA Early Adopter Working Group") están creando varias herramientas para ayudar en el desarrollo de aplicaciones UA y acelerar la adopción de esta tecnología. Estas herramientas incluyen desde ejemplos de servidores UA y clientes UA hasta wrappers y proxys [2].

4.3.7.3 Creación de productos nativos [16]. Los productos nativos son implementaciones específicas del modelo de arquitectura de OPC UA, para su creación la OPC Foundation en conjunto con el grupo de adaptadores iniciales de la norma desarrollan los códigos utilizados para la producción de sistemas. Las librerías, APIs y demás archivos que están en versión beta no se recomiendan para la creación de dichos sistemas, sin embargo la OPC Foundation impulsa su desarrollo para realimentar otros sistemas de tal manera que posteriormente estos sean evaluados rigurosamente con un potencial de usuarios beta.



## 5. CONCLUSIONES

1. El estándar OPC UA se perfila como el estándar de estándares porque a través de su definición propone las pautas para la implementación de servidores de información con la capacidad de estructurar estándares de información como ISA-95 y porque este se basa en estándares de internet ampliamente reconocidos como SOAP, WSDL, XML, UDDI y Servicios web, con un metamodelo de información estándar establecido por la OPC Foundation.
2. El estándar OPC UA se encuentra en fase de desarrollo y de validación ya que las especificaciones aun no se dan por terminadas y porque los elementos del estándar como los SDK's y las APIs para los diferentes ambientes de desarrollo como .Net, JAVA y ANSI C, aun se encuentran en etapa de desarrollo y de prueba.
3. Las clases abstractas de OPC UA definen el metamodelo del estándar UA y la OPC Foundation es la única organización con la capacidad de realizar sus actualizaciones. El metamodelo de OPC UA permite diseñar cualquier modelo de información dentro del paradigma orientado a objetos.
4. El estándar OPC UA es una especificación multi-partes con el propósito de brindar una solución modular que sea escalable a los nuevos requerimientos y a los avances tecnológicos en el tiempo.
5. Es importante resaltar que el conjunto de reglas propuestas para el nombramiento y la diagramación de los nodos en el *AddressSpace* servirá para que el conocimiento se comparta y se reutilice en posteriores investigaciones.
6. El resultado de la aplicación del estándar OPC UA a los modelos de información desarrollados en el caso de estudio es el primer acercamiento que se ha hecho en la integración de los estándares OPC UA e ISA S95 dentro del entorno investigativo de la automatización industrial. Actualmente se conformó el grupo de investigación IICI (The Industrial Interoperability Compliance Institute). El objetivo de este grupo es trabajar con los estándares OPC UA, ISA 95 e ISA 88 con el propósito de crear un *framework* de aplicación estandarizado y un punto de referencia para la interoperabilidad que permita adoptar los procesos de negocios y de manufactura.
7. El método de codificación del estándar OPC UA basado en los esquemas XML UA permite que otros modelos de información descritos a través de XML sean mapeados a través de la extensión de los elementos del estándar OPC UA y de la creación de nuevos elementos que complementan la definición del modelo de información. Sin embargo la extensión de los modelos a OPC UA produce esquemas XML que en el momento de ser instanciados requieren más líneas de código.

8. Los esquemas B2MML2UA propuestos en el proyecto son la representación de los modelos B2MML a OPC UA a partir de los cuales es posible instanciar modelos de información basados en ISA-95
9. Para los productos OPC basados en las tecnologías COM el estándar OPC UA formula la estrategia de migración a UA. Sin embargo estos productos estarán limitados en las arquitecturas orientadas a servicios porque los endpoints de las wrapper o las proxies mapean respectivamente cada interface COM.
10. La arquitectura SOA que propone el estándar OPC UA es la revolución en arquitecturas software en la industria, por esta razón debería ser explorado en el entorno investigativo de la universidad del Cauca.

## 6. BIBLIOGRAFÍA

- [1] OPC FOUNDATION. What is OPC? [En línea]. 2008. (Consulta: 11 Febrero de 2008).  
[http://www.opcfoundation.org/Default.aspx/01\\_about/01\\_what\\_is.asp?MID=AboutOPC](http://www.opcfoundation.org/Default.aspx/01_about/01_what_is.asp?MID=AboutOPC)
- [2] OPC FOUNDATION. OPC Unified Architecture. [En línea]. 2008. (Consulta: 20 marzo de 2008).  
[http://www.opcfoundation.org/Default.aspx/01\\_about/UA.asp?MID=AboutOPC](http://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC)
- [3] Managing Automation. ICONICS, Kepware Deliver Complete OPC-UA Solution. [En línea]. 2008. (Consulta: 25 abril de 2008).  
[http://www.managingautomation.com/maonline/exclusive/read/ICONICS\\_Kepware\\_Deliver\\_Complete OPCUA\\_Solution\\_27754612](http://www.managingautomation.com/maonline/exclusive/read/ICONICS_Kepware_Deliver_Complete OPCUA_Solution_27754612)
- [4] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 1: Overview and Concepts, Version 1.01.01- October 29, 2007.
- [5] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 3: Address Space Model, Version 1.01.04 - August 31, 2007.
- [6] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 8: Data Access, Version 1.01.06 - June 11, 2007.
- [7] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 4: Services, Version 1.01.17 - December 18, 2007.
- [8] INSTRUMENTOS Y CONTROLES, Conferencia sobre automatización, Carlos Fernando Guiot, Enero 21, 2008.
- [9] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 7: Profiles, Version 0.23- July 28th, 2006.
- [10]. OPC FOUNDATION, UA DevCon 2007 - 210 Implementing UA Stack Video. [En línea]. 2008. (Consulta: 20 Julio de 2008).  
<http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=460&CN=KEY&CI=287&CU=14>
- [11] IBM, Elements of Service-Oriented Analysis and Design. [En línea]. 2008. (Consulta: 30 Julio de 2008).  
<https://www.ibm.com/developerworks/webservices/library/ws-soad1/>

- [12] Larman, Graig. UML y PATRONES, Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda edición, 2003.
- [13] IBM, IBM WebSphere Enterprise Service Bus (ESB). [En línea]. 2008. (Consulta: 24 Julio de 2008).  
[http://www-01.ibm.com/software/pe/info/middleware/applications/?ca=esb&me=W&met=inba&re=WebSpherehome\\_443x160](http://www-01.ibm.com/software/pe/info/middleware/applications/?ca=esb&me=W&met=inba&re=WebSpherehome_443x160)
- [14] OPC FOUNDATION, UA DevCon 2007 - 107 Architecture PPT. [En línea]. 2008. (Consulta: 10 Junio de 2008).  
<http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=434&CN=KEY&CI=287&CU=9>
- [15] The Instrumentation, Systems and Automation Society (ISA), ANSI/ISA 95.00.01-2000, Enterprise-Control System Integration Part 1: Models and Terminology, North Carolina, 2000, ISA.
- [16] OPC FOUNDATION, OPC Unified Architecture, Draft Specification Part 6: Data Access, Version 1.00 - February 20, 2008.
- [17] FOUNDATION UA, DevCon 2007 - 301 Migration to UA Video. [En línea]. 2008. (Consulta: 20 Agosto de 2008).  
<http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=464&CN=KEY&CI=287&CU=14>