

www.biblioteca.ve
UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
ESCUELA DE SISTEMAS
DEPARTAMENTO DE INVESTIGACIÓN DE OPERACIONES

“DESCRIPCIÓN DE PROTOCOLOS PARA HOLONES DE PRODUCCIÓN EN XML”

Autor: Héctor Alberto Zambrano Vethencourt

PROYECTO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD
DE LOS ANDES COMO REQUISITO FINAL PARA OPTAR AL
TÍTULO DE INGENIERO DE SISTEMAS

Tutor: Profesor Edgar Chacón

NOVIEMBRE DE 2005

AGRADECIMIENTOS

A Dios, por concederme la constancia y fortaleza en estos últimos meses.

A mis padres, por su infinita paciencia y sabiduría.

A mis hermanos, por sus eternas presencias dentro mi.

Al profesor Edgar, por ayudarme a afilar mis sentidos y agudizar mi percepción.

A Carla Elizabeth, por rescatarme el aliento con su cariño y paciencia, cuando parecía desvanecerse.

A Carlos Alberto y Joan Manuel, por ser mis compañeros y ejemplos de lucha hasta el final.

A Héctor Alfonso por ser mi compañero sincero, aún en la distancia.

“...no te des por vencido, aún vencido;

no te sientas esclavo, aún esclavo...”

Anónimo

TABLA DE CONTENIDOS

AGRADECIMIENTOS	II
TABLA DE CONTENIDOS	III
ÍNDICE DE FIGURAS	VI
RESUMEN	VIII
INTRODUCCIÓN	1

www.bdigital.ula.ve

CAPITULO

I MARCO TEÓRICO

1.1. Definición del Problema	5
1.2. Justificación	5
1.3. Objetivos Generales	6
1.4. Metodología	6
1.5. Producción. Conceptos Básicos	7
1.6. La Cadena del Valor y la Ventaja Competitiva	24
1.7. Fundamentos Teóricos sobre Holones	27
1.8. Sistemas Distribuidos	32

II EL MODELO PROPUESTO

2.1. Los Sistemas De Información Para La Planificación Y Programación Avanzada De La Producción (PPAP)	41
2.2. El Modelo Propuesto	45
2.3. Cliente	49
2.4. Subsistema Ventas	49
2.5. Subsistema Diseño	50
2.6. Subsistema PPAP	52
2.7. Subsistema Compras	64
2.8. Subsistema Producción	65

III DESCRIPCIÓN DEL MODELO Y SUS PROTOCOLOS EN UML

3.1. Justificación del Uso de Subsistemas e Interfaces en UML Versus el Uso de Paquetes	69
3.2. UML y las Interfaces del Modelo	70
3.3. Intercambio de Información	71
3.4. Protocolos para el Envío de Objetos de Información entre PPAP y Demás Subsistemas	81
3.5. Definición de Interfaces	100

VI DOCUMENTACIÓN DE PROTOCOLOS EN XML

4.1. XML Generalidades	104
4.2. Reglas Generales para la Generación de Documentos en XML	108

4.3. Generación de Documentos en XML	120
CONCLUSIONES	137
RECOMENDACIONES	140
BIBLIOGRAFÍA	141

ANEXOS

A. Esquema Original PSLX	146
B. Esquema Final PSLX-Tesis	153
C. Guía de Instancias	174
D. Definición de Interfaces	191
E. Algunos Esquemas Desplegados	210

ÍNDICE DE FIGURAS

1.1. Transformación de Recursos	10
1.2. Esquema de un Sistema de Gestión de Producción	17
1.3. La Cadena de Valor de Porter	25
1.4. Representación Distribuida de Cliente/Servidor de Tres Capas	34
1.5. Un Sistema de Base de Datos Distribuida	36
1.6. Representación de una Base de Datos Distribuida de Cliente/Servidor de Tres Capas	37
2.1. Arquitectura de Sistemas de Información	42
2.2. Capas para la Interoperabilidad de Aplicaciones	43
2.3. Ontología para Componentes del Core	44
2.4. Ontología de Ítem y Operación	44
2.5. Modelo Propuesto	48
3.1. Patrones Básicos de Mensajes de Respuesta	83
3.2. Patrones para Reservar una Respuesta	84
3.3. Diagrama Secuencial VENTAS-DISEÑO	86
3.4. Diagrama Secuencial VENTAS-PPAP	88
3.5. Diagrama Secuencial PPAP-VENTAS	90
3.6. Diagrama Secuencial, Posibilidades de Petición de VENTAS	91
3.7. Diagrama Secuencial, Posibilidades de Peticiones de PPAP	92

3.8. Diagrama Estructural VENTAS	94
3.9. Diagrama Estructural DISEÑO	95
3.10. Diagrama Estructural Compras	96
3.11. Diagrama Estructural Producción	97
3.12. Diagrama Estructural PPAP	98
3.13. Diagrama Estructural “Macro”	99
4.1. Capas para la Comunicación de Mensajes	122
4.2. Esquema y Meta Esquema	122
4.3. Caso 1	127
4.4. Caso 2	128
4.5. Caso 3	129
4.6. Caso 4	130
4.7. Caso 5	131
4.8. Caso 6	132
4.9. Caso 7	133
4.10. Caso 8	134
4.11. Caso 9	135
4.12. Caso 10	136

RESUMEN

Actualmente, los sistemas de manufactura están sujetos a muchos cambios, tanto en su estructura, como en las tecnologías de información que utilizan para comunicarse. Por lo tanto, para que las empresas se mantengan productivas, es imperante que la gestión en línea de la producción sea eficiente, manteniendo información acerca del estado de los procesos, y los mecanismos de actuación sobre los mismos. Debido a que los sistemas de producción actual son sistemas distribuidos, es necesario poder transmitir la información entre las máquinas distribuidas que ejecutan los procesos de producción mediante el paso de “mensajes”; y que esta información sea entendida por todos los componentes del sistema.

Para resolver dichos problemas, en este estudio se propone el uso de un modelo basado en un sistema holónico de producción, que es adaptable a una o varias ontologías. El modelo se prueba a través de diagramas realizados en UML, mostrando su estructura, y funcionamiento. A través de este modelo, se describen protocolos para el envío de los mensajes entre subsistemas; éstos se consideran objetos que salen desde una interfaz como “objetos-petición”, en un subsistema correspondiente a un centro de decisiones, y llegan a otra interfaz en otro subsistema, en donde este último contesta con un “objeto-respuesta”. Para que los subsistemas que se comunican entiendan los mensajes, se propone un esquema de datos común en XML, a través de cuyo uso, todas las aplicaciones que corran dentro del sistema, no necesiten migrar de formato de datos al transmitirse mensajes.

INTRODUCCIÓN

La manufactura en su sentido más amplio, es el proceso de convertir la materia prima en productos. Incluye: el diseño del producto, la selección de la materia prima, y la secuencia de procesos a través de los cuales será fabricado el producto [Kalpakjian y Schmid, 2002].

La definición de fabricación/manufactura como la acción de hacer/producir productos/artículos revela poco sobre la complejidad del problema. Una definición más específica la da CAM-I (Consortium for Advanced Manufacturing International): “Una serie de actividades y operaciones interrelacionadas que involucran diseño del producto, maquinarias y herramientas, planificación de procesos, materiales, compras, manufactura (la producción o fabricación propiamente dicha), servicios de apoyo, marketing, ventas, envío y servicio al cliente”.

La competitividad global y las necesidades cambiantes de los usuarios, están forzando un cambio primordial en los estilos de fabricación y configuración de organizaciones industriales. La planificación de fabricación centralizada y secuencial, la asignación de recursos, y los mecanismos de control tradicionales están siendo considerados poco flexibles para responder a cambios en los estilos de fabricación y a los requisitos de fabricación altamente variables. Los enfoques tradicionales limitan las capacidades de expansión y re-configuración de los sistemas de fabricación. La organización centralizada de tipo jerárquico tradicional puede provocar caídas del sistema por fallos en un sólo punto, o por fragilidad de los planes [Boggino, 2005].

Los sistemas de fabricación futuros deberán ser más orientados al tiempo, aunque todavía deberán centrarse en el coste y la calidad de fabricación. Estas consecuencias implican: productos más complejos, productos que cambian con mayor rapidez, introducción más rápida de los productos al mercado e inversiones reducidas. El futuro del sector de la fabricación estará determinado por la manera en que éste satisfaga los desafíos de la “nueva fabricación”. Tales sistemas de fabricación necesitarán satisfacer los siguientes requisitos fundamentales [HMS, 1994] [Shen y Norrie, 1999]: *integración de la empresa, organización distribuida, entornos heterogéneos, inter-operabilidad, cooperación, integración de humanos con el software y hardware, agilidad, escalable y tolerancia a fallos.*

En respuesta a estas tendencias, en los últimos 10 años, los investigadores han estado aplicando tecnología de agentes a la integración de empresas de fabricación y administración de cadenas de suministro, planificación de fabricación, asignación de recursos y ejecución de control, manipulación de materiales y desarrollo de nuevos tipos de sistemas de fabricación tales como *sistemas holónicos de fabricación.*

Los sistemas holónicos se originaron a partir de teorías filosóficas sobre la creación y evolución de sistemas adaptativos complejos (sistemas sociales, teoría de la evolución). Dado que los filósofos no sólo observan los fenómenos sino que también tratan de explicarlos, los sistemas holónicos pueden tener fundamentos más sólidos que muchos de sus competidores [Boggino, 2005], como por ejemplo: biónicos [Okino, 1993], genéticos [Ueda, 1993], fractales [Warnecke, 1992], aleatorios [Iwata y Onosato, 1994] o virtuales [Kimura, 1993].

Este trabajo está organizado en cuatro capítulos conformados por: generalidades, el modelo propuesto, descripción del modelo en UML y generación de documentos en XML.

En el capítulo 1 se describen conceptos sobre los sistemas de producción tradicionales, fundamentos sobre holones, sistemas distribuidos y bases de datos distribuidas.

En el capítulo 2 se explica el modelo propuesto, a través de una estructuración de las que se consideran las clases principales que se utilizan en la PPAP, agrupándolas dentro de subsistemas (que se consideran, interactúan bajo un esquema holónico), y definiendo las relaciones principales que existen entre dichas clases o módulos.

En el capítulo 3 se explican los protocolos de envío de objetos de información, que usa el modelo propuesto, para lograr el intercambio de mensajes y cooperación entre los subsistemas del mismo. También se describe el modelo desarrollado en el capítulo 2, pero detallando aspectos estructurales y evaluando los aspectos funcionales a través de diagramas en UML.

El capítulo 4 consiste en el desarrollo del estándar de mensajería en XML, para el entendimiento de los subsistemas del modelo desarrollado en el capítulo 2, usando como base de transmisión de los mensajes, los protocolos de envío de objetos desarrollados en el capítulo 3 y un tipo de sincronía específica en la comunicación.

www.bdigital.ula.ve

CAPÍTULO I

MARCO TEÓRICO

Los sistemas de producción son sistemas distribuidos, donde cada uno es responsable de asegurar la producción de un determinado producto en un tiempo debido, bajo unas restricciones dadas. La producción es asegurada si se tiene:

- a) Recursos necesarios: Equipos, insumos, materia prima, servicios.
- b) Elementos de ingeniería que permitan definir un método para un producto.
- c) Información necesaria para conocer el objetivo.

Asumiendo que cada unidad de producción posee la capacidad de procesamiento para tomar decisiones, la información del objetivo de producción, el estado del proceso, debe poder viajar a través de un soporte de comunicaciones, para establecer sistemas de coordinación y planificación que permitan definir los objetivos de producción coherentes para cada unidad.

Existe la tecnología para realizar la transferencia de información, pero lo que falta como propuesta es la estandarización de la mensajería para transferir la información.



Dentro del grupo de investigadores del Laboratorio de Sistemas Discretos, Automatización e Integración (LASDAI), se está trabajando en la definición de mecanismos de negociación entre los sistemas y formas de representación de la información, asumiendo que los sistemas son considerados holónicos.

1.1. Definición del Problema

La gestión en línea de la producción sigue siendo un problema por resolver, ya que la información del estado de los procesos y los mecanismos de actuación sobre los mismos no están completamente definidos. Como continuación de los trabajos que se han ido desarrollando en el Laboratorio de Sistemas Discretos, Automatización e Integración (LASDAI), existe una propuesta para la definición de esquemas de representación de procesos para su supervisión, y la forma de comunicación entre los procesos y los centros de coordinación bajo un esquema holónico.

El problema planteado es el de la construcción de un esquema de mensajería para interconectar los centros de decisión con los procesos de planta.

1.2. Justificación

Las empresas de fabricación del siglo 21 se encuentran en un entorno en el cual los mercados son muy cambiantes, nuevas tecnologías emergen constantemente, y los competidores se multiplican globalmente. Las estrategias de fabricación deben cambiar para soportar la competitividad global, la innovación e introducción de nuevos productos, y la respuesta rápida al mercado; con el fin de maximizar sus beneficios minimizando los costes de producción.



1.3. Objetivos Generales

Construir un conjunto de protocolos de intercambio de información entre procesos de producción y los centros de gestión, que permitan el manejo coordinado del proceso productivo. Los mensajes se considerarán documentos y se describirán mediante XML.

1.3.1. *Objetivos Específicos*

- Proponer un modelo (bajo un esquema holónico), para la estructuración de las clases principales y sus relaciones, que esté capacitado para obedecer a un modelo abstracto de datos, basado en una ontología (por ejemplo, la ontología PSLX).
- Diseñar protocolos para definir el “camino” que recorre un objeto de negocios, cuando se envía desde un sitio a otro dentro del sistema holónico.
- Un esquema que permita la representación de procesos, desarrollado en XML, como propuesta para la estandarización de la mensajería para transferir la información.

1.4. Metodología

1. Estudiar los conceptos de sistemas holónicos de producción.
2. Hacer una revisión bibliográfica de los trabajos precedentes.
3. Desarrollar un modelo estructural para la planificación y programación avanzada de la producción.
4. Validar el funcionamiento del modelo, realizando pruebas de cooperación en UML.
5. Estudiar XML, esquemas y mecanismos de despliegue.



6. Validar un protocolo de intercambio de datos representados en XML, entre aplicaciones distribuidas, usando el estándar descrito por PSLX para este proceso.
7. Realizar unas pruebas de cooperación.
8. Presentar los resultados.

1.5. Producción. Conceptos Básicos

El término “producción” se aplica, de manera general, a la obtención de los bienes materiales que se precisan por una sociedad, esto es, bienes de consumo y bienes de inversión; también se conoce como una de las funciones necesarias, en toda empresa u organización que realice una actividad económico-social, sin importar si se trata de una empresa manufacturera o de servicios.

En toda empresa que actúe en el mercado se puede distinguir, al menos, tres funciones principales: la función comercial, la función administrativa y financiera, y la función de producción.

La función comercial se orienta a conseguir clientes para los productos o servicios, por tanto, no existe cuando se trata de una institución no mercantil, como una alcaldía o un hospital del seguro social. En cambio, la función de producción, cuyo objetivo apunta a las operaciones físicas que se precisan realizar para la transformación de los materiales en productos o para la realización de un servicio, existe siempre, tanto si se trata de una fábrica, como de un supermercado o de una alcaldía [FEMZ, 1999].

La “gestión de la producción” se orienta a la utilización más económica de unos medios (máquinas, espacios, instalaciones o recursos de cualquier tipo) por unos empleados u operarios, con la finalidad de transformar unos materiales en productos o de realizar unos servicios. Por ejemplo: Supongamos que se tiene una fábrica de ladrillos. Este es un caso típico de producción de bienes materiales, es decir, la transformación de una materia prima



(tierras) en productos para construcción (ladrillos). Aquí la gestión se orientará a conseguir la máxima productividad con un consumo mínimo de energía y una buena calidad del producto final. En este caso se trata de una producción continua, por tanto, los resultados se miden por periodo (número de unidades/día) y por unidad de producto¹. Los conceptos que se encuentran definidos bajo esta sección, hasta la sección de Toma de Decisiones (sec. 1.5.5.), fueron extraídos en su totalidad de [FEMZ, 1999].

1.5.1. Conceptos Relacionados con la Gestión

- **Capacidad:**

Con este concepto se refiere a la cantidad de unidades de producto que se puede obtener por unidad de tiempo en un proceso, utilizando al máximo los recursos disponibles.

- **Capacidad demostrada:**

Si se desea un mayor acercamiento a la realidad del proceso se deberá calcular *la capacidad demostrada*. Este valor se refiere al valor medio de las capacidades desarrolladas por el proceso durante un cierto periodo de tiempo. Por ejemplo:

$$\text{Capacidad demostrada} = \frac{1200 + 1150 + 1080 + 1230}{4} = 1165$$

- **Capacidad productiva:**

Es la cantidad de recursos, principalmente fuerza laboral y maquinaria, que están disponibles en el proceso de producción.

- **Carga:**

Es la cantidad de producto por unidad de tiempo que se le exige a un proceso en un momento determinado.

¹ Consumo/ladrillo, cantidad de materia prima que se utilizó para lograr una unidad de producto



- **La sobrecarga. Los “cuellos de botella”:**

Ocurre si la carga es superior a la capacidad, ya que el proceso no puede operar todo lo deseado. Los recursos que limitan la capacidad y por tanto originan la sobrecarga se denominan “cuellos de botella”. La solución a la sobrecarga puede tomarse modificando (aumentando o disminuyendo) la cantidad de producto que puede ser procesada por esos recursos, o aumentar la cantidad de recursos, para a su vez, aumentar la capacidad.

- **El plazo de ejecución (lead time):** También se le refiere como tiempo de producción. Es el tiempo necesario para realizar una operación, o varias operaciones.
- **Tiempo de espera:** Es el tiempo que está el producto en espera hasta que comienza la operación.
- **Tiempo de preparación:** Es el tiempo que se necesita para disponer adecuadamente los recursos que van a efectuar la operación. Por lo general este tiempo se requiere la primera vez que se procesa el producto o cuando se cambia el tipo de producto.
- **Tiempo de operación:** Es el tiempo consumido por los recursos cuando se efectúa la operación.
- **Tiempo de transferencia:** Es el tiempo necesario para transportar una cantidad de producto que ya ha sido sometido a una operación, al lugar en donde será sometido a otra nueva operación.

1.5.2. Procesos Productivos

A continuación se tratarán conceptos concernientes a los procesos productivos, entre ellos: La transformación, logística de producción, tipos de procesos productivos y producción según el tipo de proceso productivo.



1.5.2.1. La Transformación

Todos los sistemas que generan bienes y servicios, lo que en realidad hacen, es transformar unos bienes, para obtener otros diferentes, que se llaman “Productos Finales” (ver figura 1.1).



Figura 1.1. Transformación de Recursos. Tomado de “Curso de gestión de producción” por la Federación de Empresarios del Metal de Zaragoza (FEMZ), España, Zaragoza, 1999, p. 8.

www.bdigital.ula.ve

Lo importante desde el punto de vista de la gestión no es sólo la transformación física, si es que ésta se da; lo importante también es la transformación económica que siempre sucede, esta transformación económica se refiere a la transformación de la *utilidad*.

- **Utilidad marginal:**

Consiste en la cantidad monetaria dispuesta a pagar por los consumidores para conseguir unos productos que aumenten su satisfacción.

- **Valor agregado o añadido:**

La cantidad que el consumidor paga por los productos finales es mayor que la que pagaría por los recursos. A este cambio económico se le conoce como Valor Agregado, y permite al *Sistema de Producción* obtener medios económicos para conseguir nuevos recursos con los que se puede generar más cantidad de productos finales.



- **Costo de la producción (incorporado):**

La producción requiere sacrificar los recursos para conseguir los productos finales. Una medida monetaria de la cantidad de recursos empleados es lo que se conoce como *costo incorporado*.

- **Rendimiento de la transformación:**

La diferencia entre ambas magnitudes económicas (valor añadido y coste incorporado), es lo que mide el rendimiento económico de la producción:

$$\text{Valor agregado} - \text{Costo de la producción} = \text{Rendimiento de la transformación:}$$

Sabiendo esto, se puede definir la producción como: el estudio de las técnicas de gestión empleadas para conseguir la mayor diferencia entre el valor agregado y el costo incorporado, es decir, mayor ganancia.

1.5.2.2. Logística de la Producción

Según esta perspectiva, la producción cumple una doble misión; de un lado actúa como elemento de comunicación acerca de las necesidades de bienes que tiene el mercado, y por otro actúa como distribuidor de los productos finales. “La Producción” tiene como misión hacer rentable la transformación y la distribución de productos.

1.5.2.3. Tipos de Procesos Productivos

- En cuanto al tipo de producto, hay dos procesos básicos:

Fabricaciones. Los que producen bienes tangibles.

Servicios. Los que producen bienes intangibles. Son productos con dos cualidades: muy individualizados y muy perecederos.

- En cuanto a las diferencias en la gestión:



Inventarios. Los procesos de bienes tangibles, si así se desea, pueden contar con inventarios de productos. En los servicios no hay esta posibilidad.

Las necesidades de capacidad. Este punto es consecuencia del anterior; todo proceso de producción ha de estar dispuesto a suministrar las cantidades de productos finales que se demanden en todo momento para ello se tienen dos posibilidades: una, emplear los inventarios para mantener una capacidad de producción estable, de manera que cuando la demanda cae por debajo de la capacidad, se produce para el almacén; y dos, al contrario, cuando la demanda de productos finales supera la capacidad, la parte no producida se toma del almacén.

El proceso se diseña para una capacidad que cubra una supuesta demanda media. Esta estrategia puede hacerse en las “fabricaciones”, pero es obvio que no se contemplará en los “servicios”, ya que carecen de “stocks” (almacenes). En estos hay que diseñar el proceso para abastecer la demanda punta y modificar ésta según evolucione la demanda.

Coste de inventario (productos en el almacén): Los inventarios tienen un coste consecuencia del dinero atado al producto, es decir, todo producto ha costado a la empresa una cierta cantidad, si la empresa no vende ese producto, no recupera el dinero invertido en él y por tanto no puede producir más unidades.

1.5.2.4. Los Procesos Productivos o Fabriles

Fabricación “Justo A Tiempo”. Un sistema de fabricación basado en una gestión sin “stocks” -en realidad con pocos stocks- para ahorrar esos intereses, es la base de los llamados “Justo A Tiempo”.

La producción “contra-stock”. Si un cliente necesita un producto con más rapidez que lo que se tarda la fábrica en producirlo, se tienen que tener producidas ciertas cantidades de producto, para que el cliente no espere. Producir de esta manera exige manejar grandes cantidades de productos finales baratos, pues de otra manera los costes de los “stocks” lastran la rentabilidad del proceso.



La producción por diseño. En las producciones exclusivas, el cliente quiere un producto único donde el coste unitario es importante, pero no siempre fundamental. Lo importante en estos casos suele ser el plazo en el que el cliente podrá disfrutar del producto, por lo que la gestión se encamina a que todos los recursos se hallen disponibles en el momento oportuno en que se necesiten. Por supuesto, estos procesos no suelen tener stocks, a no ser de algunas materias primas, pero no de productos finales.

La producción por ensamblado. En las producciones por ensamblado se realizan múltiples productos basados en opciones. En tales casos sería inviable mantener “stocks” de todos los productos posibles de producir.

1.5.2.5. La Producción según el Tipo de Proceso

Procesos de flujo continuo. Se refiere a aquellos procesos donde el flujo de producto sigue siempre una secuencia de operaciones que viene establecida por las características de producto. Como es lógico pensar, estos procesos se adaptan bien a aquellos casos en que se requiere producir contra-stock.

- **Procesos continuos:** Son aquellos que producen sin pausa alguna y sin transición entre operación y operación.
- **Procesos en serie:** En estos procesos hay una transición entre las operaciones, y están diferenciadas por requerir la aplicación de maquinaria o mano de obra distinta, para cada operación. Entre ellos se pueden encontrar:
 - **Procesos de flujo en lotes:** En estos procesos cualquier cambio entre productos de la misma familia requiere una preparación previa de la maquinaria, como ocurre en las imprentas, al cambiar las planchas. La preparación supone un tiempo en el que la línea de producción estará parada.
 - **Procesos de flujo alternado (o flujo mezclado):** Que son una particularidad de los anteriores, ya que producen lotes, pero de cantidades mínimas e incluso de unidades.



Procesos de flujo discontinuo o flujo intermitente. Corresponde esta denominación a aquellos productos que no tienen definida una secuencia fija de operaciones. El flujo de operaciones queda determinado por el producto procesado. Este tipo de proceso es el adecuado para fabricar productos diferentes, es decir para productos obtenidos por ensamblado (tipo taller).

Procesos sin flujo. Se refieren a aquellos procesos donde se disponen las operaciones alrededor del producto. No existe de antemano ningún flujo definido, por consiguiente son los adecuados para los productos por diseño (o por proyecto).

Lo que se ha mencionado, corresponde a situaciones idóneas, no obstante cuanto más se acerque la relación producto-servicio---proceso a estas condiciones, su gestión será más fácil.

1.5.3. La Gestión de la Producción: Necesidades según los Tipos

La gestión de la producción o, lo que es lo mismo, el conjunto de decisiones de dirección o manejo, se orienta siempre a conseguir la mayor eficacia y/o eficiencia del sistema. Las medidas de eficacia sólo miden la salida del sistema, la producción en sí misma, pero no su coste. Las medidas de eficiencia son medidas de rendimiento, es decir, de resultados comparados con costes.

La gestión de la producción se puede expresar esquemáticamente, donde se destacan los siguientes subsistemas de gestión.

1.5.3.1. Subsistema de Planificación

Planificación de la capacidad. Es una previsión de las necesidades de capacidad de la planta a largo plazo. A partir de esta previsión se determinan las inversiones en instalaciones y maquinaria.

Previsión de ventas. Consiste en una suposición de ventas de los distintos productos sobre la base de unas determinadas acciones comerciales.



Plan de producción. Se trata de determinar el plan más conveniente en relación con los costes totales implicados. Determina las necesidades de plantilla fija y eventual y las necesidades de materiales a medio plazo.

Gestión de materiales. Consiste en determinar las necesidades de materiales a medio plazo y la relación con la gestión de “stocks” o gestión de inventarios. Determina, por tanto, las inversiones de circulante.

Ordenación de la producción. Convierte las necesidades anteriores en órdenes concretas de compra y/o producción a corto plazo.

Programación de la producción. Trata de optimizar los recursos productivos a corto plazo, programando órdenes concretas y definiendo prioridades.

1.5.3.2. Subsistema Operativo

Es el conjunto de operaciones de ejecución de la producción, desde la entrada de materias primas de los proveedores, hasta la salida de los productos finales a los clientes. El seguimiento de estas operaciones suministra la información para el “control de producción”.

1.5.3.3. Subsistema de Control

Control de producción. Está constituido por la comparación de las medidas de ejecución de las operaciones, con las previsiones (fechas de terminación, tiempos, costes de materiales). Hay un control de producción a corto plazo que regula la programación de la producción y otro a medio plazo que modifica o regula el plan anual de producción.

Control de Inventarios (stocks). Las entradas y salidas de materiales y productos terminados se controlan en esta función, que está directamente ligada con la gestión de materiales.



1.5.3.4. Subsistema Financiero

La ejecución de la producción da lugar a imputaciones de costes y compromisos financieros que son necesarios conocer con precisión.

Todos los subsistemas y funciones de gestión no tienen la misma relevancia en los diferentes tipos de producción (ver figura 1.2).

www.bdigital.ula.ve

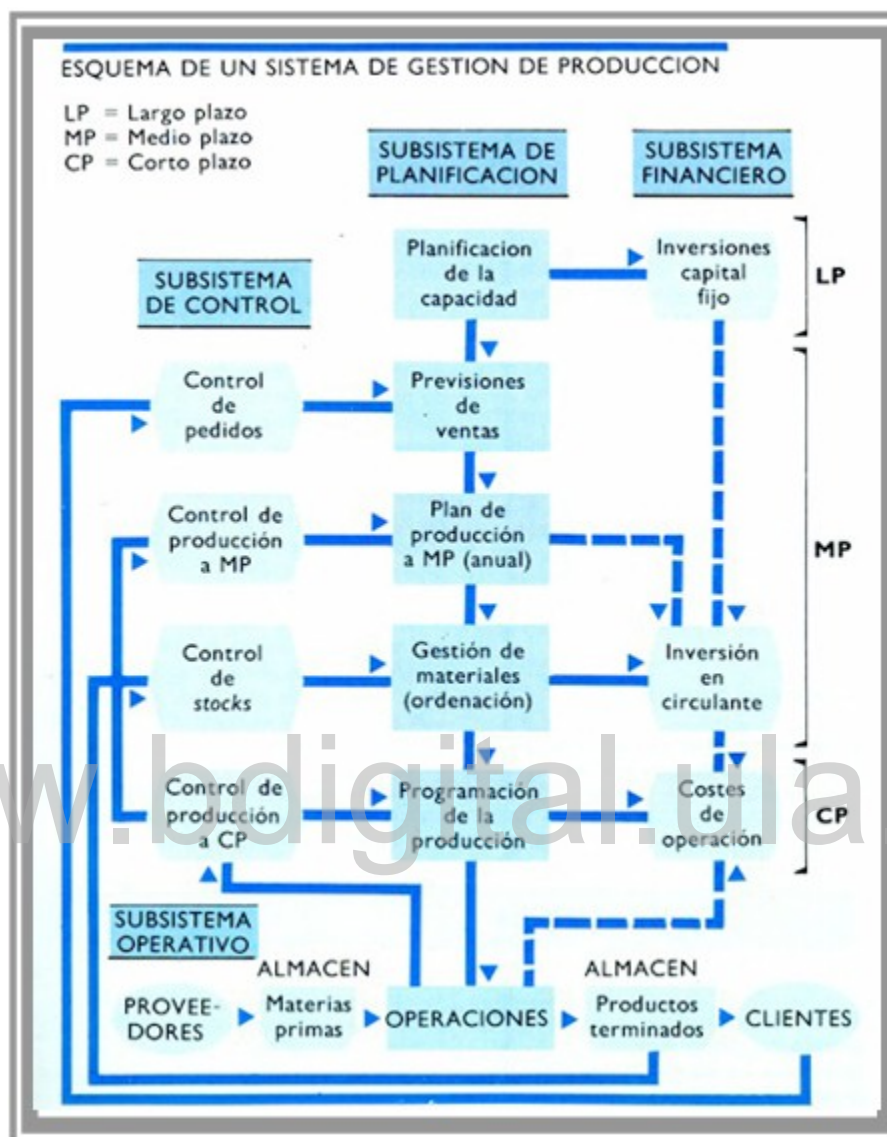


Figura 1.2. Esquema de un Sistema de Gestión de Producción. Tomado de “Curso de Gestión de Producción” por Federación de empresarios del metal de Zaragoza, Zaragoza, España, 1999, p. 16.

1.5.4. El Proceso de Planificación

La planificación es el fundamento de la gestión administrativa. Sin un plan no hay bases para establecer cuáles deben ser las acciones que la empresa ha de tomar en el futuro,



ni existen referencias que permitan comparar lo conseguido con lo que se hubiera deseado conseguir.

Por tanto, todo plan debe constar de los siguientes elementos:

- Los objetivos que la empresa se propone alcanzar en el futuro.
- Los medios con los que empresa va a contar para alcanzar esos objetivos.
- El tiempo durante el cual la empresa va a disponer de dichos medios. Se conoce también como “horizonte temporal de la planificación”.

Comúnmente los “horizontes temporales” en una empresa se establecen como se expresa en los incisos siguientes.

- **Planificación estratégica:**

Es corriente definir un periodo de 2 años como el mínimo que debe incluir el plan estratégico. No puede decirse lo mismo para el máximo valor de dicho periodo, algunas empresas toman 3, 5 o más años como límite temporal. Un buen criterio para establecer cuál debe ser el máximo horizonte temporal es: *“Aquel durante el cual se prevé una demanda creciente o estable de los productos”* [FEMZ, 1999]. Cuando la demanda comienza a declinar deberá como mínimo, iniciarse una nueva planificación estratégica.

- **Planificación táctica:**

Mínimo horizonte temporal: No es posible establecer planes sobre materiales o sobre recursos en un horizonte menor que el *“plazo de fabricación”*.

Máximo horizonte temporal: Es regla práctica para una gran parte de los procesos productivos, emplear para esta planificación un horizonte de un año como máximo (criterio basado en la coincidencia con el ciclo fiscal de la empresa).



- **Programación:**

El mínimo horizonte temporal de la planificación a corto plazo vendrá determinado, por el tiempo de producción más largo de cualquiera de los productos que intervienen en el plan.

Para determinar el máximo, se tendrá en cuenta la importancia en la inexactitud de los pronósticos hechos sobre la demanda.

1.5.4.1. Largo Plazo o Planificación Estratégica

Los objetivos de la planificación estratégica de la producción se fijan en una concepción global, es decir, son de carácter genérico que con posterioridad darán paso a otros más concretos. Los puntos a desarrollar por esta etapa de la planificación son:

El “plan de negocios”. El Plan de Negocios es el documento en el que la empresa establece sus líneas de actuación a largo plazo sobre el mercado, sus productos y los medios de producción necesarios que ha de disponer para conseguir los objetivos marcados.

La determinación del “plan de producción agregado”. Permite establecer una tasa de producción uniforme, de manera que sea el almacén el que absorba las diferencias entre producción cuando ésta supera a la demanda y al contrario.

Ajuste previo de la “capacidad de producción”. Se debe verificar si se disponen de los recursos suficientes para llevar a cabo la tasa de producción. Las pautas que habrá que tomar en esta etapa de la planificación, serán las formas de evitar que surjan las sobrecargas en la planificación de los recursos.

Financiación de los recursos y de la producción. Consiste en establecer los costes de todos los recursos que van a intervenir:

- Costes de la materia prima.

Coste de la mano de obra directa.



- Costes indirectos. Se refiere a aquellos costes como los de supervisión, control de calidad, aprovisionamientos, administración, etc.

1.5.4.2. Medio Plazo o Planificación Táctica. La Integración con el Plan de Negocios

La planificación táctica debe ser el vínculo de unión entre los objetivos fijados por la dirección de la empresa, y las disponibilidades para conseguirlos; da a la planificación un sentido práctico a medida que se hace presente el futuro. Los objetivos de la planificación táctica de la producción, ahora se aproximan más a realidad del proceso. Estos objetivos son:

- Conocer cuánto hay que producir de cada uno de los productos comercializados.
- En qué fecha hay que producir esas cantidades.

La previsión de ventas. Una gestión eficaz de la producción exige tener un marco de referencia, para un horizonte suficiente -generalmente anual- que es la previsión de las ventas o, mas exactamente, de la demanda previsible por cada línea de producto y por periodos de tiempo adecuados, generalmente mensuales.

Plan maestro de producción. Cuando las previsiones de ventas dan cifras muy regulares, mes tras mes, es posible convertir estas previsiones en plan de producción, sin ningún cambio. En muchos casos las ventas siguen pautas propias del mercado que no tienen por qué coincidir con la evolución de la capacidad de la fábrica. El plan sirve para establecer una política de producción a medio plazo (un año) y, por tanto, dimensionar la plantilla fija, los turnos a realizar y las necesidades de recursos financieros para soportar “stocks” o invertir en maquinaria adicional.

El plan de producción es, por tanto, una hipótesis de trabajo y no necesariamente un programa cerrado de producción. Las órdenes de producción específicas y los programas a corto plazo sirven para corregir y ajustar los planes de producción, mes a mes. Los costes del plan que es necesario considerar son:



CAPÍTULO I MARCO TEÓRICO

- Los costes de mantenimiento de los “stocks” por exceso de producción (costes financieros de almacenaje).
- Los costes de falta de servicio (o ruptura del “stock”), es decir, cuando no sea posible servir pedidos por falta de stock.
- Los costes por horas extraordinarias de trabajo (diferencias con el coste de la hora normal), cuando la producción excede la capacidad normal.
- El coste de desocupación, coste de oportunidad cuando la capacidad es superior a las necesidades de producción.

La técnica MRP-I. Planificación de la necesidad de materiales. Permite acortar y racionalizar el procedimiento de planificación. Está técnica determina las necesidades de materiales a partir de las cantidades requeridas de los productos finales, por lo tanto, hay que fijar en primer lugar las necesidades de éstos y las fechas de entrega, para pasar a continuación a la determinación de los productos intermedios y así ir descendiendo en la estructura de elaboración del producto hasta llegar a la materia prima.

Para ello se debe partir de ciertos documentos elaborados por la propia empresa, estos son:

- Programa maestro de producción
- Lista de materiales
- Registros de inventarios.

Hay otro tipo de planificación que no se hace sobre la base de los productos finales sino sobre módulos o subconjuntos, que ensamblados de distinta manera dan como resultado el producto final. El Programa Maestro de Producción parte ahora de los módulos y determina las necesidades de materia prima para producir estos. Los productos finales llegan a través de las opciones, que sólo se montan en el instante que se conoce la



confirmación del pedido. A partir de este instante entra en juego otro documento denominado *Programa del Ensamblado Final (PEF)*

Las diferencias entre el Programa Maestro y el Programa de Ensamblado Final:

- El Programa Maestro de Producción trabaja con previsiones de demanda. El Programa de Ensamblado Final trabaja con pedidos confirmados.
- El Programa Maestro prevé las necesidades de productos finales o de subconjuntos a partir de las materias primas. El Programa de Ensamblado Final determina las necesidades de subconjuntos a partir de los productos finales.

1.5.4.3. Corto Plazo

Aunque no tiene un nombre específico, se observa que coincide con lo que en producción se conoce como “Programación”.

Programación y control de operaciones. El objetivo de la programación y control de operaciones es cumplimentar en cantidad y plazo las órdenes de fabricación, optimizando, en lo posible, los costes de producción a corto plazo. Esta programación implica:

- La determinación de la *relación carga/capacidad* a corto plazo, para cada uno de los puestos de trabajo de la planta de producción.
- La determinación de las necesidades de horas extras o turnos, a corto plazo, para dichos puestos y, en su caso, las necesidades de subcontratación para conseguir el cumplimiento de los plazos.
- El establecimiento de la “secuencia de lanzamiento” de las distintas órdenes y las fechas previstas de comienzo y terminación para cada operación y, consiguientemente, la fecha de terminación prevista para cada orden.



CAPÍTULO I MARCO TEÓRICO

El control de las operaciones implica un seguimiento de la realización, para comprobar “desviaciones” y tomar medidas para su corrección.

Los objetivos de la programación se pueden concretar en:

- Cumplir plazos.
- Minimizar los ciclos de fabricación.

Minimizar la infrautilización de la maquinaria.

Para minimizar los ciclos de fabricación -lo que significa menos inversión en stocks en curso- se tendería a lanzar una orden lo más *tarde* posible, partiendo de la fecha de terminación *-programación hacia atrás-*.

Para minimizar la infrautilización de la maquinaria, interesa adelantar en lo posible el comienzo de las órdenes *-programación hacia delante-* para saturar cuanto antes la maquinaria, aún cuando no sea necesario desde el punto de vista de los plazos.

El conflicto entre ambas políticas es evidente. La *programación*, por tanto, persigue lograr un equilibrio satisfactorio entre ambas posiciones extremas.

Programación maestra de la producción. El Programa Maestro determina la producción que hay que realizar en el corto plazo y deriva del Plan Maestro, del que se obtienen los datos. El alcance temporal del Programa Maestro es relativo y como en el caso anterior (Plan Maestro de Producción) depende del tipo de proceso y su duración, aunque es práctica común tomar dos o tres meses. En lo que se refiere al horizonte temporal mínimo viene determinado de manera obligatoria por el tiempo de procesado del producto.

Se debe confeccionar en las mismas unidades que se realiza la producción. También se deben desagrupar los datos de la producción procedentes del Plan Maestro, es decir, tomar individualmente los productos de las familias en que fueron agrupados. Debe plasmarse el programa en unidades de producción: litros, kilogramos, unidades físicas, etc. y no en términos monetarios, horas, etc.



1.5.5. Toma de Decisiones

Como principio general, la “toma de decisiones” es considerada como un proceso mental humano del más alto nivel. Es, según PSLX [PSLX Consortium, 2005], el proceso de determinar información relacionada a planificación y programación, para actividades dentro de una empresa. Incluye no solo planificación y programación, sino los procesos asociados que crean y manejan información para estos propósitos. La toma de decisiones debería ser realizada tomando en cuenta lo siguiente:

- Ser ejecutada con base en información correcta y precisa.
- Incluir personal de los pisos de planta.
- Monitorear actividades en línea para proveer una respuesta.
- Compartir información respecto a causas y resultados, para colaborar con otros.
- Ser simple y visible en términos de mecanismos.

1.6. La Cadena del Valor y la Ventaja Competitiva

La cadena de valor fue descrita y popularizada por Michael Porter en su best-seller de 1985: *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY The Free Press.

Aquí Michael Porter indica que: La cadena de valor clasifica las actividades que producen valor añadido en una organización, siendo las principales: La Logística de Insumos, Operaciones, Logística de Distribución, Mercadeo y Ventas, Servicios y las funciones de apoyo: Compras, Desarrollo Tecnológico, Recursos Humanos e Infraestructura Corporativa [Chacón, Altamiranda y Colina].



CAPÍTULO I MARCO TEÓRICO

El marco de la cadena de valor enseguida se puso en el frente del pensamiento de gestión de empresa como una poderosa herramienta de análisis para planificación estratégica. Su objetivo último es maximizar la creación de valor mientras se minimizan los costes. El concepto ha sido extendido más allá de las organizaciones individuales, ya que puede ser aplicado también a cadenas de suministro completas y a redes de distribución.

La puesta a disposición de un conjunto de productos y servicios, al consumidor final, moviliza diferentes actores económicos, cada uno de los cuales gestiona su cadena de valor; las interacciones sincronizadas de esas cadenas de valor locales crean una cadena de valor ampliada que puede llegar a ser global. Capturar el valor generado a lo largo de la cadena, es la nueva aproximación que han adoptado muchos estrategas de la gestión. A base de explotar la información que se dirige hacia arriba y hacia abajo dentro de la cadena, las compañías pueden intentar superar los intermediarios creando nuevos modelos de negocio [Wikipedia, Cadena del valor].

En la figura 1.3 se puede observar la clasificación de las principales actividades o elementos fundamentales, que contempla el modelo de Porter.

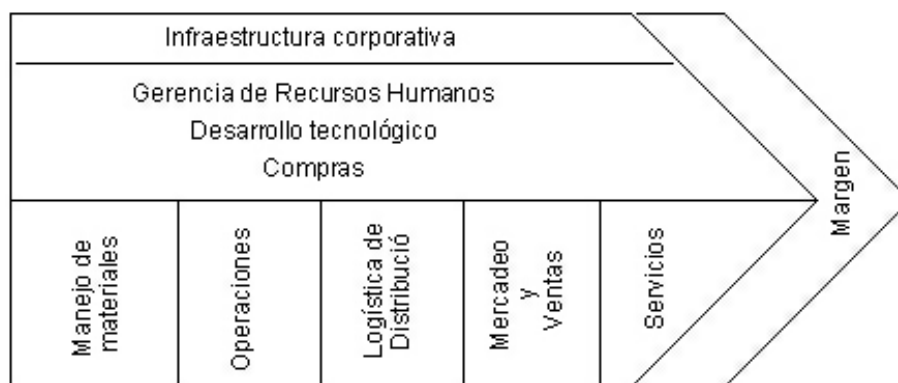


Figura 1.3. La Cadena de Valor de Porter. Tomado de “Estructuras Teleinformáticas para la Implantación de Sistemas de Monitoreo y Supervisión de Sistemas de Producción” por Edgar Chacón, Edmary Altamiranda, Eliecer Colina.



CAPÍTULO I MARCO TEÓRICO

Para Michael Porter -reconocido académico en teorías administrativas-, el "valor" es la suma de los beneficios percibidos, menos los costos percibidos por el cliente al adquirir y usar un producto o servicio.

La cadena de valor es una forma de análisis de la actividad empresarial mediante la cual descomponemos una empresa en cada una de sus partes, buscando identificar fuentes de ventaja competitiva en aquellas actividades que generan valor. La ventaja competitiva se logra cuando la empresa desarrolla e integra cada una de las actividades que conforman su cadena de valor a un menor costo y con una mayor diferenciación a la de la competencia. En otras palabras la cadena de valor de una empresa la forman todas aquellas actividades que aportan un valor agregado y los márgenes de utilidad que estas representan. Al momento de analizar una empresa en la cual deseamos invertir, resulta útil evaluar la cadena de valor que ésta genera, ya que con dicho análisis podemos proyectar si generará o no utilidades, y determinar si la actividad económica que realiza es rentable [Synthesis C.I., 2004].

En general existen dos tipos de actividades en la cadena de valor, las primarias y las de soporte. Las actividades primarias son aquellas que están relacionadas directamente con la realización del producto y las de soporte son las relacionadas al recurso humano, infraestructura y desarrollo tecnológico. Una vez identificadas, se deben asociar los costos e ingresos por actividad y estimar el margen que aportan al negocio. De esta manera, la cadena de valor busca la fuente de ventaja competitiva en aquellas actividades que generan valor y los márgenes que éstas aportan. La ventaja competitiva puede ser costos bajos o diferenciación [Synthesis C.I., 2004].



1.7. Fundamentos Teóricos sobre Holones

El desafío actual radica en el hecho de que los sistemas industriales de la “vida real” necesitan, tanto alto rendimiento como reactividad [Boggino, 2005].

La respuesta a este desafío se busca en las teorías sobre sistemas adaptables complejos. Koestler [Koestler, 1971] hizo la observación que los sistemas complejos sólo pueden surgir si se componen de subsistemas estables, cada uno de ellos capaz de sobrevivir ante disturbios, pero además, capaz de cooperar para formar un sistema estable mas complejo. Estos conceptos han dado lugar al término “fabricación holónica”. La fabricación holónica (Holonc Manufacturing - HM) es una organización altamente distribuida, donde la inteligencia se distribuye sobre las entidades individuales. El elemento nuevo en la fabricación holónica es el hecho que las entidades individuales trabajan juntas en jerarquías temporales, para obtener un objetivo global [Boggino, 2005].

La Fabricación Holónica es un paradigma desarrollado en el marco del programa IMS (Sistemas de Fabricación Inteligente). El trabajo realizado en el programa IMS, ha trasladado estos conceptos al mundo de la fabricación, considerando al sistema de fabricación como un compuesto de módulos autónomos con control distribuido [Boggino, 2005].

El HMS (Holonc Manufacturing Systems-Sistemas Holónicos de Fabricación o Manufactura) es una iniciativa de investigación para sistemas de fabricación avanzados, inspirada en los conceptos propuestos por Koestler. El objetivo es lograr en la fabricación, los beneficios que las organizaciones holónicas proporcionan a los organismos vivos o a las sociedades, esto es, estabilidad ante alteraciones, adaptación y flexibilidad ante cambios, y uso eficiente de los recursos disponibles. El nuevo paradigma combina los conceptos naturales de los sistemas jerárquicos y la integración de los elementos autónomos dentro de un sistema distribuido. Cuando por el contrario las arquitecturas de los sistemas de fabricación convencionales están modeladas a lo largo de líneas jerárquicas de relaciones de orden-obediencia, la arquitectura HMS está modelada utilizando relaciones todo-parte.



El consorcio HMS, compuso un glosario con definiciones para un amplio rango de términos que ayudan a entender y a guiar la traducción de los conceptos holónicos a escenarios de fabricación [Seidel, 1994]. Las definiciones principales son las siguientes:

- **Holón (holón de manufactura o producción):**

Una unidad de construcción autónoma y cooperativa de un sistema de fabricación para transformar, transportar, almacenar y/o validar objetos de información y físicos. El holón consiste de una parte de procesamiento de información y muchas veces de una parte de procesamiento físico. Un holón puede ser parte de otro holón. Ejemplos de holones de fabricación son herramientas, maquinas, equipos de transporte, trabajadores humanos, información del producto (diseño, plan de desarrollo), máquinas de operaciones, órdenes del cliente, etc. Nuevamente, cada uno de estos puede ser tratado como un todo en sí mismo, o como una parte de una organización mayor [Wyns, 1999].

- **Autonomía:**

La capacidad de una entidad de crear y controlar la ejecución de sus propios planes y/o estrategias.

- **Cooperación:**

Un proceso por el cual un conjunto de entidades desarrollan planes aceptados mutuamente y ejecutan dichos planes.

- **Holarquía (manufacturing holarchy):**

Un sistema de holones que pueden cooperar para lograr un objetivo. La holarquía define las reglas básicas para la cooperación de los holones y por tanto limita su autonomía [Wyns, 1999].



- **Sistema holónico manufacturero (HMS):**

Un sistema holónico de fabricación integra el rango completo de actividades de producción, desde la recepción de una orden, pasando por el diseño, la fabricación, y el mercadeo para lograr la empresa de fabricación ágil. Está organizado como una holarquía, que define las reglas básicas para la cooperación de los holones. Un sistema holónico de fabricación no está, en cambio, organizado de manera fija, sino se organiza a sí mismo de manera dinámica para lograr sus objetivos, y auto-adaptarse ante cambios en su entornos o en sí mismo [Wyns, 1999].

1.7.1. Tipos de Holones

1.7.1.1. Holón Básico

Los holones de orden, holones de producto y holones de recurso son holones básicos. Los holones básicos son necesarios en todo sistema de control manufacturero basado en PROSA. Esto en contraste con los *staff holon* que son opcionales [Wyns, 1999].

1.7.1.2. Holón de Orden

Un holón de orden representa una tarea en el sistema manufacturero. Es responsable de realizar el trabajo asignado correctamente y a tiempo. Administra el producto físico que está siendo producido, el modelo de estado del producto y todo el procesamiento de información logística relacionada con el trabajo. [Wyns, 1999].

1.7.1.3. Holón de Producto

Un holón de producto mantiene el conocimiento del proceso y el producto para asegurar la correcta realización del producto con suficiente calidad. Un holón de producto contiene información consistente y actualizada del ciclo de vida del producto, requerimientos del usuario, diseño, curso de acción de los planes, lista de materiales, procedimientos de control de calidad, etc. [Wyns, 1999].



1.7.1.4. Holón de Recursos

Un holón de recurso contiene una parte física, particularmente un recurso de producción del sistema manufacturero, y una parte de procesamiento de información que controla el recurso. Ofrece capacidad de producción y funcionalidad a los holones que lo rodean. Contiene los métodos para asignar los recursos de producción, y el conocimiento y procedimientos para organizar, usar y controlar estos recursos de producción para manejar la producción. [Wyns, 1999].

1.7.1.5. Holón consultor o experto (staff holon)

Estos holones asisten a los holones básicos a realizar su trabajo. Ellos consideran algunas facetas de los problemas de control manufacturero de los holones básicos y los proveen con suficiente información de tal manera que estos puedan tomar la decisión correcta para resolver el problema. El holón empleado es considerado como un experto externo que asesora [Wyns, 1999].

1.7.2. Sistemas de Fabricación Holónicos Versus Control Jerárquico y Heterárquico

1.7.2.1. Control Heterárquico

Una forma de control altamente distribuida implementada por un sistema de agentes independientes sin un control directo explícito o centralizado [Wyns, 1999].

El enfoque heterárquico prohíbe todo tipo de jerarquía con el objeto de dar todo el poder a los módulos básicos, generalmente llamados “agentes”. Este enfoque consiste, por ejemplo, sólo de estaciones y órdenes de trabajo. Cada orden negocia con las estaciones de trabajo para obtener el procesamiento de los recursos, para ello utiliza todas las posibles alternativas de procesamiento disponibles para poder afrontar situaciones imprevistas [Boggino, 2005].



1.7.2.2. Control Jerárquico

El diseño está basado en un enfoque top-down (arriba-abajo) y define estrictamente los módulos del sistema y su funcionalidad. Más aún, la comunicación entre los módulos también se define estrictamente, y se limita, ya que los módulos sólo pueden comunicarse con sus módulos padres y módulos hijos. En una arquitectura jerárquica, los módulos no pueden tomar la iniciativa. De esta manera el sistema es vulnerable ante perturbaciones y su autonomía y reactividad ante disturbios son débiles. La arquitectura resultante es muy rígida, y por tanto cara de desarrollar y difícil de mantener [Boggino, 2005].

1.7.2.3 Control Holónico

La fabricación (manufactura) holónica combina las ventajas de los sistemas jerárquicos y de los heterárquicos al mismo tiempo que evita sus desventajas. Para evitar las arquitecturas rígidas de los sistemas jerárquicos, los sistemas holónicos otorgan autonomía (libertad de decisión) a los módulos individuales (holones). Esto otorga al sistema respuesta rápida ante disturbios, y la habilidad para reconfigurarse a sí mismo ante nuevos requerimientos.

Comparado con los sistemas de control holónicos, los sistemas de control heterárquico pueden ser imprevisibles y además incontrolables. Esto se debe a la inexistencia de jerarquías en los sistemas heterárquicos. Es por ello que los sistemas de fabricación holónicos poseen jerarquías, pero estas jerarquías son flexibles, o “flojas”. Esta jerarquía difiere del control jerárquico tradicional en lo siguiente:

- los holones pueden pertenecer a múltiples jerarquías,
- los holones pueden formar jerarquías temporales,
- los holones no dependen de operaciones propias de (ligadas a) cada holón en una jerarquía para lograr sus objetivos.



Para distinguir entre las jerarquías estrictas de los sistemas de control jerárquicos, y las jerarquías “flojas” de los sistemas de control holónicos, se ha introducido el término holarquía, para identificar a las jerarquías flexibles y flojas. Sin embargo, al otorgar reglas y consejos, la holarquía limita la autonomía de los holones individuales, para asegurar desempeños controlables y predecibles a diferencia de los sistemas heterárquicos [Boggino, 2005].

1.8. Sistemas Distribuidos

En un sistema distribuido los procesos están repartidos en distintas máquinas, e intercambian información mediante paso de mensajes sobre algún sistema de comunicación. Cuando se les compara con los sistemas tradicionales, “centralizados”, surgen nuevos problemas, debidos a la propia distribución: retardos, particiones de la red, fallo independiente de máquinas, etc. Pero también hay nuevas posibilidades, como por ejemplo la replicación de procesos en diferentes máquinas para incrementar la tolerancia a fallos. Todo esto hace que el diseño y la implementación de sistemas distribuidos sea una tarea compleja y difícil. Así, es particularmente útil encontrar formas de simplificar los problemas, y proporcionar módulos o paradigmas probados y listos para usar, que resuelvan algún aspecto difícil de la construcción de sistemas distribuidos. Esto se aplica, en particular, a los protocolos de comunicación que se usan, esto significa que: dependiendo de las características que proporcione un protocolo, la construcción de algún tipo de sistema distribuido sobre él puede ser mucho más fácil [González, 1998].

Otras definiciones de sistemas distribuidos que se pueden referenciar son:

"Sistemas cuyos componentes hardware y software, que están en ordenadores conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo prefijado por un esquema cliente-servidor" [Sistemas Distribuidos, 2003].



“Un sistema distribuido es aquel en el que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes” [Sevilla, 2005].

“Colección de elementos de cómputo autónomo que se encuentran físicamente separados y no comparten una memoria común, se comunican entre sí a través del intercambio de mensajes utilizando un medio de comunicación” [Cárdenas, 2004].

Una de las primeras caracterizaciones de un sistema distribuido fue realizada por Enslow ya en 1978 [Enslow, 1978], que le atribuye las siguientes propiedades:

- Está compuesto por *varios recursos informáticos* de propósito general, tanto físicos como lógicos, que pueden asignarse dinámicamente a tareas concretas.
- Estos recursos están *distribuidos físicamente*, y funcionan gracias a una red de comunicaciones.
- Hay un *sistema operativo* de alto nivel, que unifica e integra el control de los componentes.
- El hecho de la distribución, es *transparente*, permitiendo que los servicios puedan ser solicitados especificando simplemente su nombre (no su localización).
- El funcionamiento de los recursos físicos y lógicos está caracterizado por una *autonomía coordinada*.

A continuación se muestra la “representación distribuida de cliente/servidor de tres capas”. La interacción con el usuario se realiza en el servidor, el cliente hace de pasarela entre el usuario y el servidor

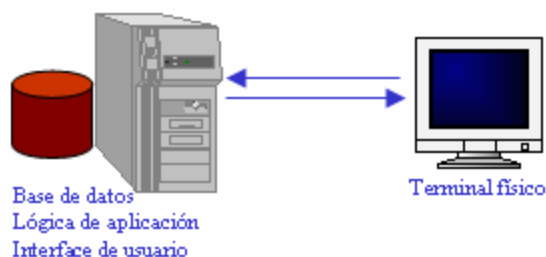


Figura 1.4. Representación Distribuida de Cliente/Servidor de Tres Capas.

Tomado de “Sistemas Distribuidos”. [Documento en línea]. 2003. Disponible en: <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>.

1.8.1. ¿Qué es un Protocolo?

Es un conjunto bien conocido de reglas y formatos, que se utilizan para la comunicación entre procesos que realizan una determinada tarea. Se requieren dos partes:

- Especificación de la secuencia de mensajes que se han de intercambiar.
- Especificación del formato de los datos en los mensajes.

Un protocolo permite que componentes heterogéneos de sistemas distribuidos puedan desarrollarse independientemente, y por medio de módulos de software que componen el protocolo, haya una comunicación transparente entre ambos componentes. Es conveniente mencionar que estos componentes del protocolo deben estar tanto en el receptor como en el emisor [Sistemas Distribuidos, 2003].

Ejemplos de protocolos usados en los sistemas distribuidos [Sistemas Distribuidos, 2003]:

- **IP: Protocolo de Internet.** Protocolo de la capa de Red, que permite definir la unidad básica de transferencia de datos y se encarga del direccionamiento de la información, para que llegue a su destino en la red.



- **TCP: Protocolo de Control de Transmisión.** Protocolo de la capa de Transporte, que permite dividir y ordenar la información a transportar en paquetes de menor tamaño para su transporte y recepción.
- **HTTP: Protocolo de Transferencia de Hipertexto.** Protocolo de la capa de aplicación, que permite el servicio de transferencia de páginas de hipertexto entre el cliente WEB y los servidores.
- **SMTP: Protocolo de Transferencia de Correo Simple.** Protocolo de la capa de aplicación, que permite el envío de correo electrónico por la red.

1.8.2. ¿Qué son Objetos Distribuidos?

En los sistemas cliente/servidor, un objeto distribuido es aquel que está gestionado por un servidor y sus clientes invocan sus métodos utilizando un "método de invocación remota". El cliente invoca el método mediante un mensaje al servidor que gestiona el objeto, se ejecuta el método del objeto en el servidor y el resultado se devuelve al cliente en otro mensaje. Una de las tecnologías orientadas a los objetos distribuidos es: CORBA (Common Object Request Broker Architecture). Tecnología introducida por el Grupo de Administración de Objetos OMG, creada para establecer una plataforma para la gestión de objetos remotos independiente del lenguaje de programación [Sistemas Distribuidos, 2003].

1.8.3. ¿Qué son Bases de Datos Distribuidas?

Un sistema de bases de datos distribuidas, como el ilustrado en la figura 1.5, es un ambiente en el que data encontrada en dos o más instancias, es accesible como si esta data estuviera en una sola instancia [Dye, 1999]. La data referenciada puede ser a "tiempo real", o puede tener segundos, horas e incluso días de estar almacenada. Generalmente, las diferentes instancias son alojadas en diferentes nodos servidores, y la comunicación entre ellos es vía SQL*Net (para Oracle7) o Net8 (para Oracle8) [Dye, 1999]. Además de los servidores de bases de datos, un sistema de bases de datos distribuidas usualmente incluye también, servidores de aplicaciones y clientes.



Según [Sistemas Distribuidos, 2003], una base de datos distribuida es una colección de datos (base de datos) construida sobre una red y que pertenece, lógicamente, a un solo sistema distribuido, la cual cumple las siguientes condiciones:

- La información de la base de datos esta almacenada físicamente en diferentes sitios de la red.
- En cada sitio de la red, la parte de la información, se constituye como una base de datos en sí misma.
- Las bases de datos locales tienen sus propios usuarios locales, sus propios DBMS (Sistemas Administradores de Bases de Datos) y programas para la administración de transacciones, y su propio administrador local de comunicación de datos.

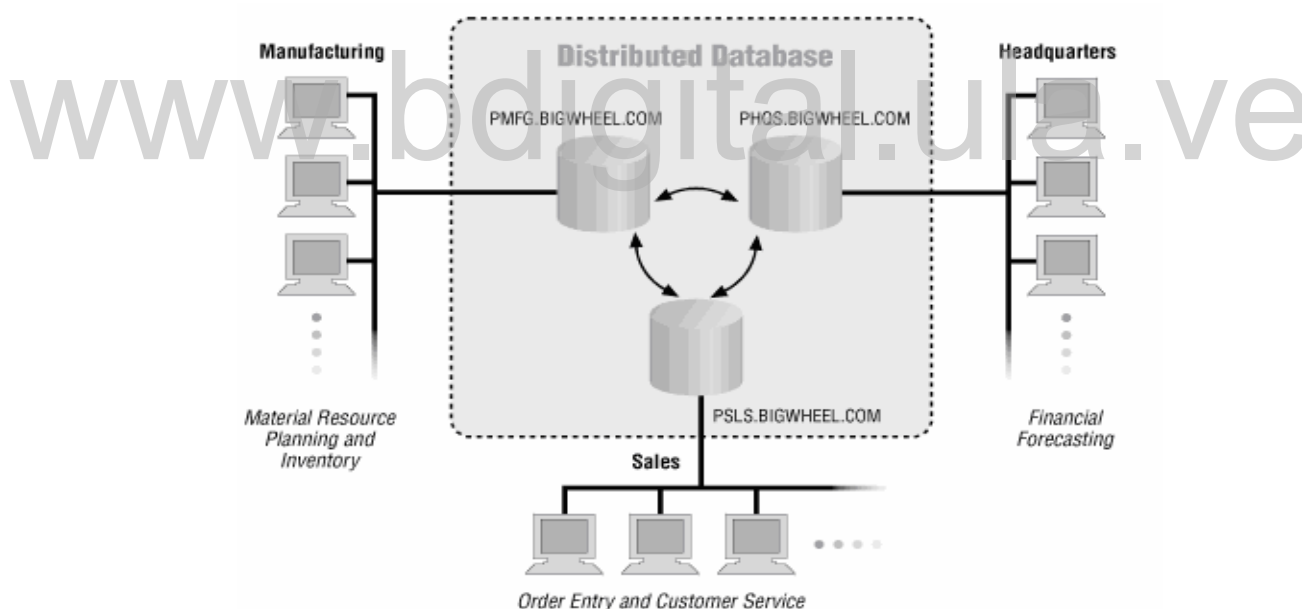


Figura 1.5. Un Sistema de Base de Datos Distribuida. Tomado de “Oracle Distributed Systems” por Charles Dye. 1999.

- Estas bases de datos locales deben de tener una extensión, que gestione las funciones de sociedad necesarias; la combinación de estos componentes con los sistemas de administración de base de datos locales, es lo que se conoce como Sistema Administrador de Base de Datos Distribuidas.



- Este gestor global permite que usuarios puedan acceder a los datos desde cualquier punto de la red, como si lo hicieran con los datos de su base de datos local, es decir, para el usuario, no debe existir diferencia en trabajar con datos locales o datos de otros sitios de la red.

En consecuencia, la base de datos distribuida es como una unidad virtual, cuyas partes se almacenan físicamente en varias bases de datos "reales" distintas, ubicadas en diferentes sitios.

A continuación se muestra la “representación de bases de datos distribuidas de cliente/servidor de tres capas”. El cliente realiza la interacción con el usuario, ejecuta la aplicación, debe conocer la topología de la red, así como la disposición y ubicación de los datos. Se delega parte de la gestión de la base de datos al cliente.

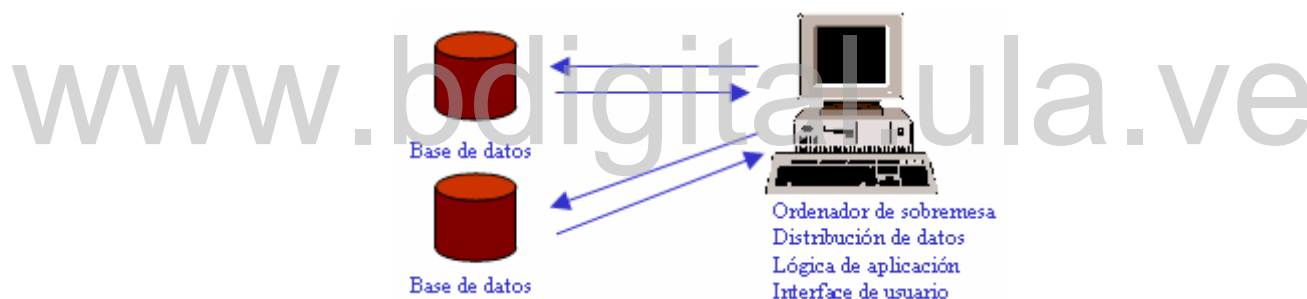


Figura 1.6. Representación de una Base de Datos Distribuida de Cliente/Servidor de Tres Capas. Tomado de “Sistemas Distribuidos”. [Documento en línea]. 2003. Disponible en: <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>.

1.8.3.1 Ventajas de las Bases de Datos Distribuidas

Descentralización. En un sistema centralizado, existe un administrador que controla toda la base de datos, por el contrario en un sistema distribuido existe un administrador global que lleva una política general y delega algunas funciones a administradores de cada localidad para que establezcan políticas locales y así un trabajo eficiente.



CAPÍTULO I MARCO TEÓRICO

Economía. Existen dos aspectos a tener en cuenta. El primero son los costes de comunicación; si las bases de datos están muy dispersas y las aplicaciones hacen amplio uso de los datos puede resultar más económico dividir la aplicación y realizarla localmente. El segundo aspecto es que cuesta menos crear un sistema de pequeños ordenadores con la misma potencia que un único ordenador.

Mejora de rendimiento. Pues los datos serán almacenados y usados donde son generados, lo cual permitirá distribuir la complejidad del sistema en los diferentes sitios de la red, optimizando la labor.

Mejora de fiabilidad y disponibilidad. La falla de un lugar, de varios lugares o de un enlace de comunicación, no implica la inoperatividad total del sistema, incluso si se tienen datos duplicados, puede que exista una disponibilidad total de los servicios.

Crecimiento. Es más fácil acomodar el incremento del tamaño en un sistema distribuido, porque la expansión se lleva a cabo añadiendo poder de procesamiento y almacenamiento en la red, al añadir un nuevo nodo.

Flexibilidad. Permite acceso local y remoto de forma transparente.

Disponibilidad. Como se indicó anteriormente, pueden estar los datos duplicados, con lo que varias personas pueden acceder simultáneamente de forma eficiente.

Control de Concurrencia. El sistema administrador de base de datos local, se encarga de manejar la concurrencia (usar un recurso compartido sin interferencia) de manera eficiente.

CAPÍTULO II

EL MODELO PROPUESTO

Los sistemas de tecnologías de información manufacturera vigente, no están necesariamente equipados para proveer soluciones al ambiente que se maneja en el mercado actual y venidero, que básicamente se describe de la siguiente manera [PSLX White Paper, 2005]:

- Comportamiento inesperado de la demanda e incertidumbre en el ambiente manufacturero.
- Acortamiento del ciclo de vida de los productos y manufacturación ágil.
- Cambios en los procesos de negocios para la gestión de la cadena de suministro (supply chain) y requerimientos del intercambio electrónico de datos (EDI).

De hecho, según PSLX [PSLX White Paper, 2005], se cree que los sistemas de información (SI) vigentes tienen las siguientes tres debilidades:



1.) Falta de capacidad funcional y aplicabilidad de los sistemas de planificación de recursos de la empresa (ERP) convencionales.

La mayoría de los fabricantes desarrollaron sistemas de información para empresas MRP o MRP II al final de los setentas (1970s) u ochentas (1980s). Algunos de ellos fueron reemplazados por ERP a lo largo de los últimos diez años. Pero aún así, tuvieran MRP, MRP II o ERP, el algoritmo básico empleado en estos sistemas es del año 1970, por lo que se le dificulta seguir el ambiente dinámico actual, sin poder proveer información precisa a los gerentes de planta.

2.) Limitación de la arquitectura cliente-servidor para los sistemas de información.

Las bases de datos convencionales en los ordenadores centrales (mainframe computers) y los sistemas cliente-servidor, necesitan ser diseñados de manera precisa con un alto nivel de atención destinado a asegurar la consistencia entre los esquemas relacionados. Estas bases de datos niegan información que no concuerda con su esquema. De cualquier manera esta información es la fuente de conocimiento en los pisos de planta, y necesita ser compartida entre diferentes individuos en diferentes divisiones. Hasta ahora esto sólo puede ser alcanzado vía comunicación cara-a-cara.

3.) Poca flexibilidad y robustez para el mejoramiento continuo.

Ambos, los procesos de negocios y los de ingeniería, necesitan ser continuamente mejorados en las industrias manufactureras. Por otro lado, los sistemas de información implementados por fabricantes no pueden ser cambiados cada vez que hay un mejoramiento de un proceso. Algunas veces, éstos pueden no modificar sus procesos de negocios porque la revisión de los sistemas de TI incluiría una gran inversión, así como riesgos potenciales. Además, la migración de datos de un sistema previo a uno nuevo, siempre es un ejercicio trabajoso y complicado, y alguna información crítica no puede ser heredada.



2.1. Los Sistemas de Información para la Planificación y Programación Avanzada de la Producción (PPAP)

La arquitectura popular actual para sistemas de información, está diseñada sobre la base de sistemas cliente/servidor. Sin embargo, existen desventajas en correr sistemas inmensos sobre esta tecnología, ya que toda la data detallada que no es referida en el esquema, permanece fuera del sistema; asimismo la revisión del esquema requiere un esfuerzo extraordinario porque puede afectar considerablemente muchas aplicaciones de software existentes. Para ajustar los sistemas de información a casos particulares de pisos de planta, la data que representa cada proceso de producción es muy importante. Además un esquema que representa una planta necesita ser flexible, de manera tal que pueda ser adaptado en respuesta a cambios actuales [PSLX White Paper, 2005].

La arquitectura del sistema de información que el consorcio PSLX (PSLX Consortium) propone, consta de las ventajas de la tecnología actual del ambiente cliente/servidor, y las ventajas que provienen de los sistemas autónomos distribuidos, que permiten a los sistemas locales cambiar a niveles de detalle, sin acordarlo con otros. La figura 2.1 muestra la diferencia entre las dos arquitecturas de sistemas de información. Comparada con la arquitectura actual, que tiene toda la data almacenada en una base de datos central, la arquitectura futura tendrá varias bases de datos pequeñas (Mini Bases de Datos o MBD). Información específica de dominio y local, es creada y almacenada independientemente en las bases de datos locales.

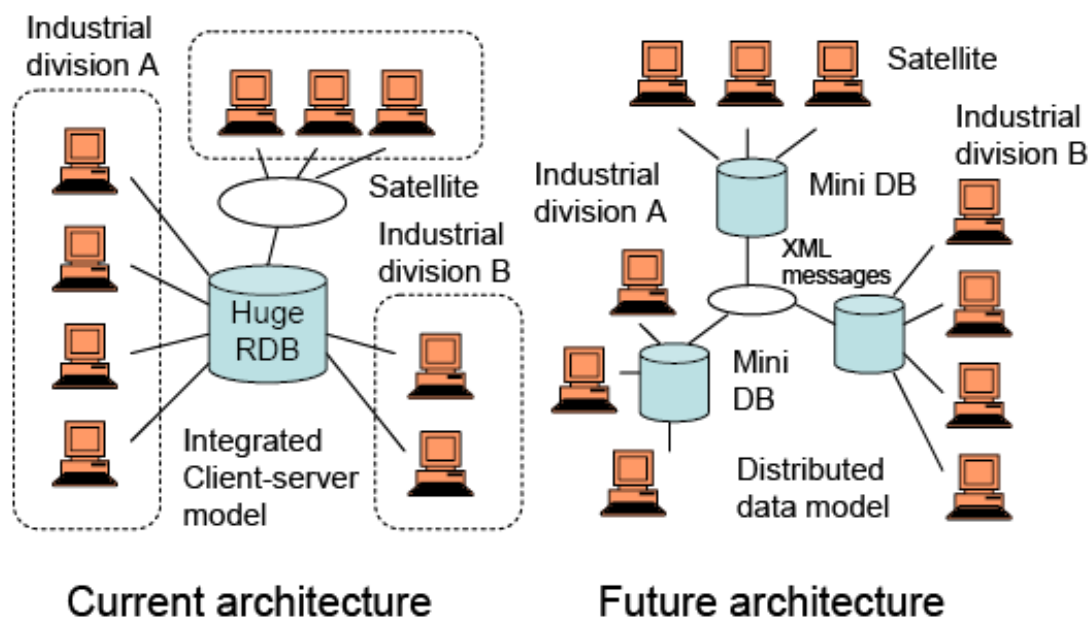


Figura 2.1. Arquitectura de Sistemas de Información. Tomado de “Advanced Planning and Scheduling (APS) Conceptual definition and Implementation” por PSLX White Paper, PSLX Consortium. 2005

En la arquitectura futura, la consideración debe estar dada a la interoperabilidad entre bases de datos autónomas, cada una de las cuales tiene su subsistema relacionado. La colaboración entre diferentes bases de datos y la consistencia de datos serán los asuntos a tratar por los diseñadores. Para establecer la comunicación entre diferentes esquemas de bases de datos, se necesita que haya una discusión en las semánticas detrás de los esquemas de datos; para este fin, se utiliza un “modelo abstracto de datos” [PSLX White Paper, 2005]. Los modelos abstractos de datos, necesitan tener una ontología detrás de sus definiciones. Ontología (en este contexto) es la base de todas las definiciones de semántica, representando los elementos primitivos y sus relaciones.

Para poder establecer la integración de aplicaciones entre diferentes actividades de negocio, se necesita que haya un esquema de datos común que esté definido de antemano, junto con el uso acordado de semánticas para los elementos de datos definidos en el esquema. La figura 2.2 muestra la jerarquía de los modelos de datos según PSLX.



CAPÍTULO II EL MODELO PROPUESTO

Un “esquema común”¹ está definido como un modelo de datos, que actúa para integrar “diferentes esquemas de implementación” en aplicaciones diferentes. Un esquema común debería ser definido para cada actividad de negocios antes de que los sistemas de aplicación sean implementados. Posteriormente los programadores de sistemas de aplicaciones pueden escoger el esquema más apropiado e interpretar las semánticas, utilizando la ontología definida por PSLX², para aplicarlo a su propio esquema de implementación.

Un modelo abstracto de datos, posee un conjunto completo de objetos y relaciones para uso general en Planificación y Programación Avanzada de la Producción (PPAP). Este modelo es completamente independiente de la forma de implementación de los sistemas de información. En la figura 2.3 se muestra un ejemplo de una ontología definida por PSLX para los componentes del “core” (núcleo), y en la figura 2.4 una ontología definida para los elementos básicos “item” y “operación” [Yasuyuki, OASIS y PSLX Consortium, 2005] necesaria para representar problemas de PPAP.

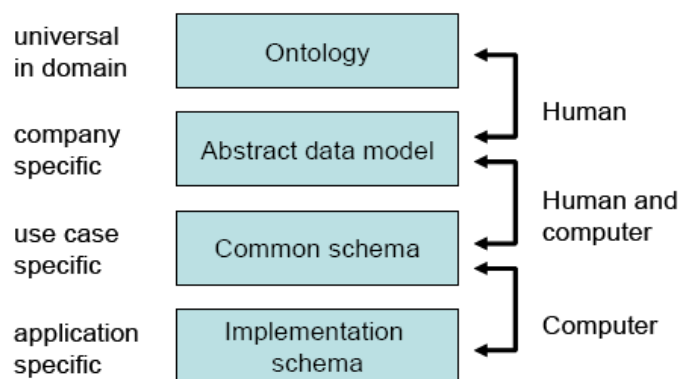


Figura 2.2. Capas para la Interoperabilidad de Aplicaciones. Tomado de “Advanced Planning and Scheduling (APS) Conceptual definition and Implementation” por PSLX White Paper, PSLX Consortium. 2005

¹ Véase sección 4.3.1 del capítulo IV para más información acerca del esquema común utilizado en este modelo.

² En PSLX se definen 27 ontologías. Cada una tiene relación con otra, la representación de sus semánticas depende de sus posiciones. Véase PSLX-03 Domain Object Model para información detallada.



Core components

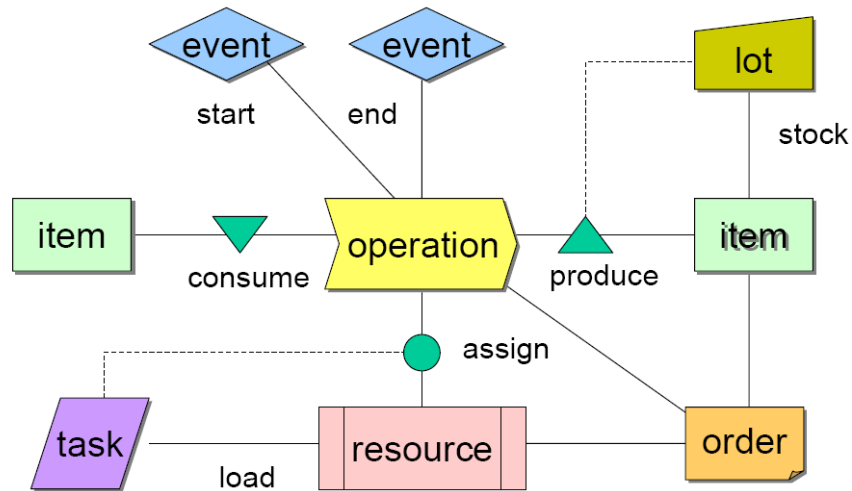


Figura 2.3. Ontología para Componentes del Core. Tomado de “Advanced Planning and Scheduling (APS) Conceptual definition and Implementation” por PSLX White Paper, PSLX Consortium, 2005

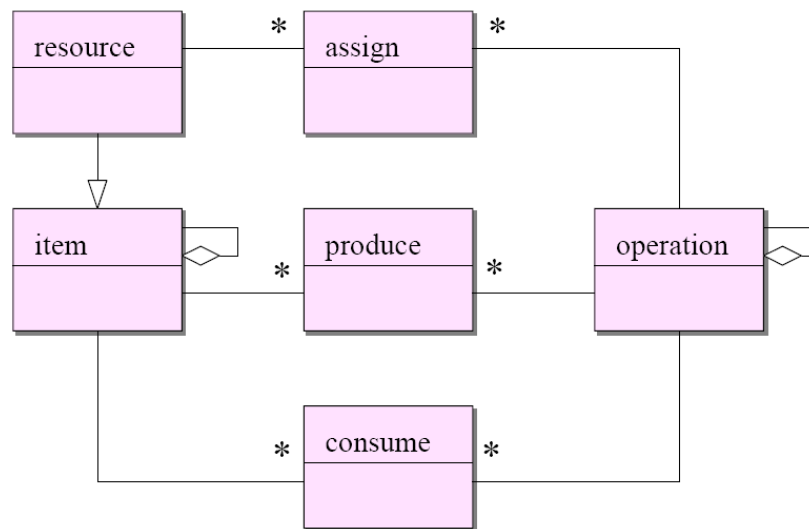


Figura 2.4. Ontología de Ítem y Operación. Tomado de “PSLX-03, PSLX Domain Objects”. PSLX Consortium, 2003



2.2. El Modelo Propuesto

El modelo que se diseña en este capítulo, toma en consideración las capas para interoperabilidad de aplicaciones descritas anteriormente y mostradas en la figura 2.2. Esto es:

- Funciona bajo la arquitectura del sistema de información propuesto por el Consorcio PSLX.
- Puede utilizar el modelo abstracto de datos “general” propuesto por PSLX³, que posee de base la ontología de esa misma organización; pero también puede utilizar un modelo abstracto de datos diferente al definido por PSLX, aún usando la misma ontología de base.
- Utiliza el esquema común propuesto por Yasuyuki, OASIS y PSLX, pero ampliado⁴ según las especificaciones de PSLX para XML [PSLX-04, 2003].

Pero además, el modelo servirá como soporte para el desarrollo y/o evaluación de los siguientes ítems a lo largo del estudio:

- Una estructuración de las clases principales que se utilizan en la PPAP, agrupándolas dentro de subsistemas (que se consideran, interactúan bajo un esquema holónico), y definiendo las relaciones principales que existen entre dichas clases o módulos.
- Protocolos que definen el camino que recorre un objeto, cuando se envía desde un subsistema a otro, a través de diagramas funcionales de UML.
- Organización de las clases principales y de las interfaces encargadas del envío y recepción de objetos en cada subsistema, a través de diagramas estructurales en UML.

³ Referido a PSLX-03 Domain Objects, que lidia con la representación correcta del conocimiento en varios tipos de actividades de negocios prácticas.

⁴ Véase sección 4.3.1 capítulo IV



CAPÍTULO II EL MODELO PROPUESTO

- Especificación de las funciones de las interfaces dependiendo de su ubicación en cada subsistema.
- Ampliación del esquema de XML común “*pps-schema-core-draft-1[1].1.*” especificado por [Yasuyuki, OASIS y PSLX Consortium, 2005], que puede ser utilizado para el modelo abstracto de datos que se asigne al modelo propuesto (el de PSLX u otro desarrollado).
- Formateo de los objetos que se envían entre subsistemas, según el tipo de mensaje (petición, respuesta, recibido, excepción), tomando en cuenta el método de comunicación (síncrona o asíncrona).

De carácter general, la organización del modelo está dada de la siguiente manera:

Cinco (5) “Subsistemas”, **Compras, Ventas, Diseño, Producción y PPAP**, que contienen a cada módulo o clase con sus operaciones correspondientes⁵.

Cada uno de ellos consta de:

- Un “Centro de gestión” que corresponde al núcleo de procesamiento de los datos de cada centro de gestión. Contiene también una “Mini Base de Datos” (MBD), que almacena las operaciones modulares individuales (base de datos distribuida) y las decisiones tomadas por la integración operativa de los módulos correspondientes a cada centro de gestión, de esta manera se producen los objetos que se envían por las interfaces correspondientes.
- Al menos una interfaz “Petición y Recepción-Objetos” que se encarga, como su nombre lo indica, de realizar las peticiones de objetos que el centro de gestión que la contiene (local), necesita realizar. Las peticiones son objetos que se envían a la interfaz “Recepción y Respuesta de Peticiones-*Nombre_CentroDeGestión_quePide*” del centro de gestión externo que es objetivo

⁵ Las operaciones de cada módulo o clase, se escapan del objetivo de este estudio, se asume que éstas se implementarán en otro estudio.



de la petición. Además, “Petición y Recepción-Objetos” se encarga de recibir los objetos que son enviados a ella como respuesta a una petición realizada, y los envía para ser almacenados en la MBD correspondiente al subsistema que generó el objeto-petición.

- Al menos una interfaz “Recepción y Respuesta de Peticiones-*Nombre_CentroDeGestión_quePide*” que se encarga de enviar la petición proveniente de un centro de gestión externo, al centro de gestión local a través de la interfaz adecuada⁶ (*getInterfaz* o *setInterfaz*), quien contesta con un objeto a dicha interfaz, que a su vez envía ese objeto a la interfaz que realizó la petición (Petición y Recepción-Objetos del subsistema cliente).
- Todas las interfaces atadas a un centro de decisiones particular (pertenecientes a un subsistema determinado).
- Todas las interfaces que utilicen los módulos o clases integrantes del sub-sistema correspondiente. Cada módulo se comunica con otro y con el centro de decisiones correspondiente, a través de una interfaz⁷.

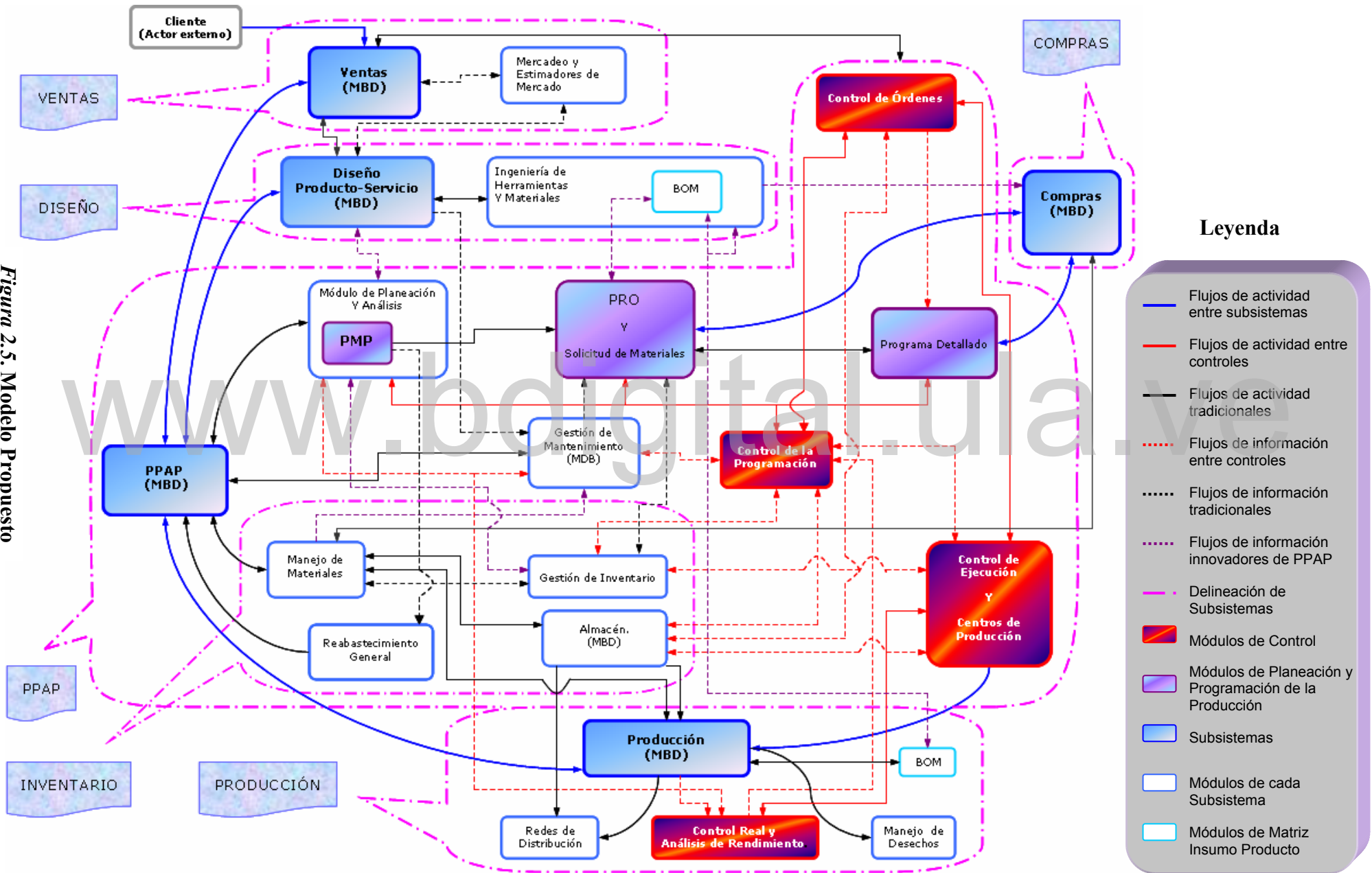
Teniendo en cuenta lo explicado, el “modelo propuesto” trata de resolver las debilidades expuestas a principio del capítulo en cuestión, a través de la “holarquía”⁸ que se define en este capítulo. Así pues, la explicación de la funcionalidad y conectividad de cada módulo del modelo en cuestión, siguen a continuación; véase primero la figura 2.5, en donde se plasma la representación gráfica de la estructuración, externa e interna, de cada uno de los subsistemas y sus asociaciones.

⁶ Véase capítulo IV.

⁷ Se asume que los módulos de un subsistema que asisten al centro de decisiones, están programados para funcionar bajo un sistema holónico.

⁸ Un sistema de holones que pueden cooperar para lograr un objetivo. La holarquía define las reglas básicas para la cooperación de los holones y por tanto limita su autonomía [Wyns, 1999]

Figura 2.5. Modelo Propuesto





2.3. Cliente (Actor Externo)

El módulo **Cliente** es una persona o empresa que solicita al menos un servicio y/o producto de la empresa en cuestión. Una empresa (con estructura semejante o no a la descrita en el modelo propuesto) puede ser a su vez cliente de otra compañía cuya relación esta dada directamente con el módulo **Ventas**.

2.4. Subsistema Ventas

Contiene un conjunto de módulos que colaboran entre sí para proveer –en conjuntos con las interfaces correspondientes- servicios a otros subsistemas; éstos se muestran a continuación:

2.4.1. Ventas (*Centro de Gestión*)

El módulo **Ventas** funciona como un centro de gestión (toma de decisiones) y una mini base de datos que proporciona y pide información del producto solicitado por **Cliente** al mismo y a sus módulos hijos, también informa al cliente acerca del estado de su orden: si fue aprobada o no, debe especificar si fue puesta en la cola regular o como orden urgente, si no fue aprobada debe detallar la razón(es) por la cual no se pudo agregar la orden; además debe manejar todas las características de índole promocional, monetaria, publicitaria, etcétera, de los productos o servicios que ésta provea. Debe proveer al módulo **Planeación y Programación Avanzada de la Producción (MBD)** con una “previsión o plan de ventas” [PSLX-02, 2003] y “plan de capacidad” [FEMZ, 1999] previo.

2.4.2. *Mercadeo y Estimadores de Mercado*

Debe lograr que ocurra el intercambio comercial entre **Ventas** y **Cliente**, es decir, convencer a éste de que el producto o servicio ofrecido es mejor que el de la competencia, siendo esto posible a través de un conjunto de elementos como por ejemplo la “mezcla de



mercadotecnia” llamada también “*Las Cuatro P*” definidas como: precio, promoción, producto y plaza.

Suministra información respecto de las necesidades de productos y/o servicios del mercado al módulo de **Diseño y Especificación del producto y/o servicio**. También debe suministrar una previsión de ventas según las tendencias del mercado al módulo **Ventas (MBD)**.

2.5. Subsistema Diseño

Contiene un conjunto de módulos que colaboran entre sí para proveer –en conjuntos con las interfaces correspondientes- servicios a otros subsistemas; éstos se muestran a continuación:

2.5.1. *Diseño y Especificación del Producto y/o Servicio (Centro de Gestión)*

Para poder abastecer el nuevo mercado de productos con ciclos de vida más cortos, poder sacar nuevos productos constantemente o para realizar cambios en los diseños de productos en existencia en la empresa, se necesita preparar nuevos documentos técnicos, nuevos equipos y nuevas instrucciones para los trabajadores, usualmente sin mucho preaviso; esto significa que este módulo debe lidiar con las complicaciones de crear o revisar los datos necesarios para los nuevos productos, debe lograrlo con la colaboración de **Mercadeo y Estimadores de Mercado** (Subsistema Ventas) y de **Ingeniería de Herramientas y Materiales**.

Debe publicar los diseños, nuevos o no, para los módulos que requieran de dicha información⁹ (en el caso de este modelo, según las relaciones que aquí se proponen) como lo son **Ventas** y **Módulo de Programación y Análisis (PPAP)**. También debe recibir

⁹ Véase capítulo IV



información sobre los materiales a utilizar en los diseños, a través de **Ingeniería de Herramientas y Materiales**.

2.5.2. Ingeniería de Herramientas y Materiales

En éste módulo se deben analizar de manera constante y exhaustiva los materiales y herramientas innovadoras, además de las de uso regular dentro de la empresa.

Los agentes de compras, como contactos de la empresa con el personal de ventas de los posibles proveedores, saben a menudo que hay materiales y herramientas nuevas y más baratas antes de que los departamentos de ingeniería se percaten de ello. Comúnmente, los agentes no tienen conocimientos técnicos necesarios para evaluar los nuevos materiales, pero deben hacer que sean examinados [Riggs, 2001], y es aquí justamente donde se observa la comunicación necesaria entre **Ingeniería de Herramientas y Materiales y Compras**.

Además, este módulo debe ser capaz de proporcionar información acerca de los materiales que serán utilizados en una producción, tanto innovadora como regular, a su submódulo **BOM** y además recibir de él, las órdenes de clave o fórmulas requeridas para construir los subproductos de todos los diseños a fabricar por la empresa.

Por otra parte, debe estar preparado para la selección y diseño de herramientas de corte, plantillas, accesorios y matrices necesarios para realizar cada operación específica en los centros de producción

BOM: Matriz Insumo Producto. El módulo **BOM** maneja datos técnicos, como las listas de materiales y datos de enrutamiento de producción, a través de la habilitación de la colaboración entre **Producción e Ingeniería**. La información incluye no sólo datos formales y aprobados, sino también datos informales que pueden no ser consistentes con los datos formales.

El BOM convencional representa los componentes del producto, conectando directamente un padre a un hijo. Esto es útil para calcular la cantidad de material requerido



para una determinada cantidad de producto final. Por otro lado, la información de enrutamiento para producir el producto es gestionada usando datos manejados independientemente, mientras los sistemas de programación utilizan los datos para calcular la carga de cada recurso.

En este modelo se integran el BOM convencional y los datos de enrutamiento a través de una nueva estructura de datos llamada BOM de operación-céntrica [PSLX White Paper, 2005], haciendo énfasis en operaciones que puedan conectar ambos, “ítems” en el BOM común y “recursos” en la tabla de enrutamiento, con mira en esto, debe establecerse un canal de comunicaciones entre **PPAP** e **Ingeniería** (a través de **Diseño**).

2.6. Subsistema PPAP

Contiene un conjunto de módulos que colaboran entre sí para proveer –en conjunto con las interfaces correspondientes- servicios a otros subsistemas; éstos se muestran a continuación:

2.6.1. Planeación y Programación Avanzada de la Producción -PPAP- (centro de gestión)

La planeación o planificación es una acción para especificar un objetivo o medios para componer la brecha entre la demanda y la realidad, y para decidir la estructura y un parámetro entre ellas. La planeación o planificación muestra el problema mismo y las maneras concretas de resolver el problema. Sin embargo, el propósito principal de la planeación es especificar la estructura y un valor “paramétrico” respecto de la relación causa y efecto. Por lo tanto el resultado obtenido no siempre garantiza la posibilidad de ejecución en el tiempo actual y así, es requerido “agendar” o programar para generar la información del plan, que es, planificar los resultados más positivos [PSLX White Paper, 2005].



La programación o agendamiento es una acción para reducir al máximo la competencia por los recursos, decidir el tiempo de comienzo y término de la operación concreta, y la asignación de tiempo para cada recurso, para garantizar que los objetivos y los medios especificados en la planificación, puedan ser ejecutados.

La información de entrada para un problema de programación o agendamiento es la información del plan obtenido por “planeación”, una y solo una de las posibilidades obtenidas por ésta, es considerada por la programación. Aún si la mejor solución-convenida es obtenida para un problema de programación, no es garantía de que satisfaga el requerimiento original [PSLX White Paper, 2005].

Este módulo contiene data de la iteración de **Planeación y Programación Avanzada de la Producción** en cuestión, es decir, de la información necesaria para generar una decisión en el módulo de **Planeación y Análisis**; y la información de “previsión de ventas” deducida de **Ventas**. De esta manera sus sub-módulos pueden realizar sus funciones respectivas, con o sin la mediación del módulo **Control de la Programación**, según los parámetros almacenados en la base de datos, obtenidos de **Ventas y Reabastecimiento**¹⁰, para dar comienzo así, a la siguiente iteración. Para ello contiene una MBD (Mini Base de Datos).

Se encarga también de coordinar todos los controles (los que se encuentran dentro de PPAP), y por último, funciona como un centro de decisiones.

2.6.2. Módulo de Planeación y Análisis

La “planeación” o “planificación” está generalmente definida como una actividad para clarificar las acciones u operaciones, para alcanzar una meta determinada y reservar suficiente capacidad de recursos para alcanzar al menos los objetivos mínimos [PSLX White Paper, 2005].

¹⁰ Por ejemplo: el caso en que comience la fabricación de productos en una empresa por primera vez; o el caso ideal en que no se necesite un control de la programación, porque todo lo previsto fue precisado de manera inequívoca (sin devoluciones de materiales, cambios de órdenes, máquinas averiadas, etc.)



En la planificación, los parámetros principales tendrán valor para cierto período de tiempo, por ejemplo: “cantidad de producción este mes”, “ventas de barriles el próximo mes”, “resumen de trabajo a sobre-tiempo la próxima semana”, etc. Son variables asociadas a períodos de tiempo.

Para optimizar el uso de los recursos en la satisfacción de los requerimientos de la empresa se emplean algoritmos de optimización como: algoritmos genéticos (AG), técnicas Tabu-search; también algoritmos más clásicos como Simplex, redes CPM, cálculos PERT y simulación, que se usan de manera individual o en combinación para alcanzar una “optimalidad progresiva de la planeación” [PSLX White Paper, 2005].

A través de estos algoritmos y con el fin de optimizar los recursos este módulo debe:

- Planear la capacidad.
- Asignar recursos a la producción.
- Conocer el estado de máquinas, instalaciones, operadores e inventarios.

Este módulo contiene información de la iteración de **Planeación y Programación de la Producción** en cuestión, la “previsión de producción” deducida de **Ventas** y maneja la “capacidad agregada” a nivel empresarial o a niveles de áreas particulares dentro de la fábrica. El ciclo de toma de decisiones corresponde a un horizonte de planeación de mediano a largo plazo. La máxima capacidad de recursos para producción puede ser cambiada, si es necesario. Los aspectos financieros de la empresa están incluidos en este aspecto de toma de decisiones, de tal manera que los beneficios de la empresa puedan ser optimizados. Recopilando y asociando estos datos a los conceptos desarrollados en el capítulo I, la definición apropiada sería la que corresponde a “Planificación Estratégica”.

Este módulo recibe información sobre las órdenes de compra y previsión de ventas (planificación estratégica) contenidas en la MBD para ejecutar sus cálculos de optimización. Recibe información del módulo **Control Real y Análisis de Rendimiento**



que actualiza su conocimiento sobre el estado (máquinas, operadores, etc) y rendimiento de la producción en proceso, y así, con ayuda del módulo **Control de la Programación**, poder rectificar en la medida que sea necesario la planeación. Para lograr esto también es necesario conocer el estado de los inventarios provistos por el módulo **Gestión de Inventarios**.

Entre sus funciones más importantes está la de generar el Plan Maestro de la Producción (PMP), que se explicará a continuación.

2.6.2.1. Plan Maestro de Producción (PMP)

Contiene la asignación preliminar de operaciones a estaciones de trabajo, según el orden sucesivo de la red, es decir, es el resultado arrojado por los métodos utilizados para alcanzar una solución a la planeación, pero organizado de manera tal, que pueda hacerse público a cualquier módulo que necesite de su información.

El PMP decide volúmenes de producción y los tiempos de comienzo y término para algunos productos finales particulares, de acuerdo a las órdenes del cliente. Esta es una toma de decisiones de corto a mediano plazo. La cantidad de cada producto está determinada por una combinación entre órdenes del cliente recibidas y órdenes proyectadas, calculadas a través de los “pronósticos de demanda o previsión de ventas”. Los recursos objetivos a este nivel son similares a los de planificación estratégica, no obstante, la limitación de la capacidad para toda la fábrica o área en particular está basada en parámetros de restricción más que en parámetros de decisión. El “plan” o “agenda” generado en este nivel es usado para forjar una especie de “contrato”¹¹ entre los módulos de **Ventas** y **Producción** o manufactura. Al mismo tiempo, todas las actividades de negocios son sincronizadas a este, confirmando la viabilidad del plan o agenda de acuerdo con la información de su capacidad local [PSLX White Paper, 2005].

¹¹ Advanced Planning and Scheduling (APS) Conceptual definition and Implementation, January 2005, PSLX Consortium, p 2-12.



2.6.3. PRO (Planeación de Requerimientos para las Operaciones) y Solicitud de Materiales.

Este módulo reúne: Órdenes de venta, previsión de ventas, plan maestro de la producción, lista de materiales, tiempos de entrega y tamaño en lotes. Aquí la cantidad y la fecha de producción de productos finales, que provee el PMP, son extendidas a las operaciones necesarias para producir los productos; luego esas operaciones son asignadas a recursos particulares a ciertos tiempos del horizonte de planificación.

Este proceso de toma de decisiones lidia principalmente con operaciones; subsiguientemente, la capacidad de recursos e inventario de partes intermedias y componentes, son discutidas en relación a la operación asociada. Los conceptos de: MRP-I y II (Material Requirement Planning o Planificación de la Necesidad de Materiales), CRP (Capacity Requirement Planning o Planeación de los Requerimientos de Capacidad), ERP (Enterprise Resource Planning o Planificación de los Recursos de la Empresa) son introducidos en este nivel.

El módulo en materia, recibe información de **Gestión de Mantenimiento** acerca de los materiales y máquinas que hay que tomar en cuenta en la reposición debida al mantenimiento, aparte de la tomada en cuenta en la planificación estratégica (que por su naturaleza menos precisa, hay que reincluir en este nivel de planificación) y cuyo control está relacionado directamente con el módulo **Control de la Programación**. De igual manera, necesita manejar la información que le provee **Gestión de Inventario** (que también es renovada por **Control de la Programación**). Procura –naturalmente- sus datos a **Programa Detallado**.

Utiliza los datos provistos por el **BOM** para calcular la carga de cada recurso y solicita los materiales a **Compras**, una vez que ha finiquitado el plan de operaciones.



2.6.4. Programa Detallado (Solicitud Dinámica de Materiales)

Es el programa último, definitivo, detallado para operaciones de piso de planta. Igualmente, respecto a la planeación y programación de los materiales y la capacidad, se enfoca en las operaciones. Una característica de la programación de planta detallada, es que lidia con restricciones y requerimientos detallados en cada piso de planta distribuido.

Además, la granularidad (nivel de detalle) de los programas que resultan de este proceso de toma de decisiones, es más precisa que aquella que es producida por el módulo **PRO**; siendo así, gracias a que utiliza el plan elaborado por este último. Generalmente el detalle de los elementos del **Programa Detallado**, corresponde a una unidad de actividades apropiada, ordenada por los gerentes de los pisos de planta como parte de los procedimientos diarios [PSLX White Paper, 2005].

La salida o el resultado del **Programa Detallado**, es utilizado como una fuente para despachar información, cuando el tiempo para una actividad en un programa se está acercando y entrando al período de ejecución, las órdenes de trabajo son enviadas a los operadores de planta correspondientes [PSLX White Paper, 2005].

El **Control de la Programación** se encarga de administrar la información manejada por **Programa Detallado** respecto a prioridad de órdenes (con la ayuda de **Control de Ordenes**), cambio de materiales, materiales utilizados, etc. En este nivel se puede solicitar la compra de los materiales necesarios, cuando los cambios de agenda y eventos inoportunos (resueltos aquí, usando la ayuda de los módulos de **Control**) así lo requieran; es por ello la referencia a dinamismo en su título. Aún así, el **Programa Detallado** debe mantener los costos dentro de los límites razonables, al mismo tiempo en que se satisfacen los otros objetivos [Riggs, 2001].

2.6.5. Gestión de Mantenimiento

Los programas de mantenimiento están estrechamente vinculados con las políticas de reposición de la empresa o fábrica. Este módulo se encarga de administrar las políticas



CAPÍTULO II EL MODELO PROPUESTO

de mantenimiento o servicio, que son: reemplazo cíclico de máquinas individuales, sustitución en grupo de partes gastadas, reparaciones a solicitud, máquinas de reserva y mantenimiento preventivo (mantenimiento productivo)¹² [Riggs, 2001].

Este módulo ha de mantener registros de funcionalidad, completos, exactos y detallados, es decir, registros de las reparaciones, servicios y estado actual de las máquinas en todos los pisos de planta; esto se logra mediante la constante petición de datos a **Control Real y Análisis del Rendimiento**. Para almacenar estos datos, este módulo contiene una MBD.

Gestión de Mantenimiento también debe transferir sus datos a **PRO** para mantenerlo al tanto de la política de mantenimiento que va a utilizar, al realizar la planificación en la iteración de planeación y programación en cuestión.

Recibe de **Manejo de Materiales** parámetros, como listas de materiales, etcétera, destinados a ser administrados por **Gestión de Mantenimiento**, siendo **Control de la Programación** el responsable de suministrar o agregar estos materiales y/o máquinas de reposición a **PRO** según lo considere pertinente.

En resumen, con el fin de optimizar la reposición de materiales y/o máquinas de la empresa, el módulo en análisis debe - en conjunto datos y controles suministrados por **Control Real y Análisis del Rendimiento** y **Control de la Programación** - replantear las políticas de reposición tantas veces como sea necesario y exigido por los módulos de control; publicará sus resultados (que es hacerlos accesibles a cualesquiera de los módulos que lo soliciten) y **Control de la Programación** informará a cualquiera de las tres etapas de planeación y programación acerca de la nueva política, cuando lo considere oportuno.

¹² Los operadores hacen todo el mantenimiento de rutina, con el fin de dejar libre a los trabajadores de mantenimiento para el verdadero trabajo preventivo.



2.6.6. *Inventario*

Contiene una cantidad de sub-módulos que se encargan de proveer las decisiones respecto del manejo de materiales, gestión de inventario, almacenamiento y reabastecimiento general. Estos módulos son:

2.6.6.1. *Manejo de Materiales*

Este módulo coordina y mantiene el flujo de materiales en la fábrica: listas hechas a partir de la recepción de materiales (destinados o no a mantenimiento) provenientes de **Compras**, control de calidad, devoluciones, suministro y transporte de la materia prima.

Manejo de Materiales toma de **Planeación y Programación de la Producción** la iteración en cuestión y la “previsión de ventas”; recibe de **Compras** los materiales pertenecientes a la iteración en proceso y los que corresponden a compras extras a la misma (debido a órdenes urgentes y/o mantenimiento); de los cuales elabora una lista que pasa a **Gestión de Mantenimiento** y a **Gestión de Inventario**. Además, se encarga realizar las pruebas de calidad a los materiales, y así reportar los materiales que aprobaron las pruebas, los que no las aprobaron y serán devueltos a **Compras** para ser repuestos, y por último, los que no las aprobaron, pero no serán devueltos sino desechados.

Por otra parte, **Control de la Programación** maneja información de última hora, tiene conocimiento de los materiales que han sido devueltos, desechados y de los materiales que ya han sido aprobados por control de calidad y se encuentran disponibles, para almacén o para ir directo como materia prima **Producción**. Además, el módulo en tema administrará los materiales que han de ser transportados con destino a almacenes y/o a los centros de producción, según sea requerido por **Gestión de Inventario** y/o **Producción** respectivamente.

2.6.6.2. *Gestión De Inventario*

Este módulo realiza todo los cálculos necesarios para la determinación del tamaño del lote de producción y el punto de reorden [Riggs, 2001]. Para esto, el módulo esgrima



CAPÍTULO II EL MODELO PROPUESTO

conceptos fundamentales como: costos de inventario, naturaleza de la demanda de inventario, modelos de inventario, MEL (Magnitud Económica del Lote), descuentos por volumen de compra, LEP (Lote Económico de Producción) y políticas de inventario. Este módulo recibe de **Manejo de Materiales** la lista de los materiales recibidos y aprobados por control de calidad.

Gestión de Inventario dictará las pautas temporales a **Manejo de Materiales** en lo que respecta a los materiales que se enviarán al almacén, según la política de inventario desarrollada, que será a su vez publicada para que **Control de la Programación** consiga realizar las acomodaciones necesarias en cuanto a la reprogramación de cualesquiera de los tres niveles de planeación y programación. De igual manera, pasará su información al **Módulo de Planeación y Análisis** y a **PRO**, que de manera recíproca envían la suya para notificar al módulo en cuestión acerca de: Número de unidades demandadas por año (estimadas para el plan) y número de unidades por pedido (utilizados para calcular el MEL), cantidad ordenada y producción por día (utilizados para calcular el LEP), entre otros; de forma que **Gestión de Inventario** puede realizar los cálculos condicionantes de la política final o temporal de inventario. Por último, se transmiten los datos a **Control de Ejecución y Centros Producción** quién dictará las acciones a tomar con respecto a la ejecución de la transportación de los materiales, a su debido momento, sobre la base de lo decidido en este módulo como política de inventario.

2.6.6.3. Almacén (MBD)

El módulo **Almacén** funciona a manera de una mini base de datos, que mantiene información actualizada en relación al estado del inventario, llevando además de registros, una cuenta física de las cantidades de disponibles.

Control de Ejecución y Centros de Producción, siguiendo la política de gestión de inventario establecida, le indica cuándo debe hacer la petición (a **Manejo de Materiales**) del transporte de los materiales del almacén a los respectivos centros de producción en **Producción** y/o a las **Redes de Distribución**, y también de los materiales recibidos por **Manejo de Materiales** hasta el almacén según sea el caso. Mantiene una



comunicación constante con **Control de Ordenes** y **Control de la Programación** para proveerlos del estado cuantitativo último de los materiales pertenecientes a materia prima, sub-productos, WIP (Work In Process)¹³, partes, material y productos finales.

2.6.6.4. Reabastecimiento General

Se encarga de preparar, según la rutina impuesta por el Plan Maestro de Producción (**PMP**) y como parte normal del proceso de producción a largo plazo, una lista de materiales, productos y sub-productos necesarios para continuar con la siguiente iteración de producción en el modelo propuesto. Por lo que pasa estos datos a la base de datos de **Planeación y Programación de la Producción (MBD)** en donde se almacenan para ser utilizados como parte del estudio próximo, en los módulos de planeación y análisis.

2.6.7. Los Controles de PPAP

2.6.7.1. Control de Órdenes

Se encarga de manejar la información de estado y las actividades de priorizar órdenes. Recibe de **Ventas** la información del producto y busca en la mini base de datos **Almacén** para observar si existe el producto. Debe ser capaz de verificar el estado de una orden (una vez que haya sido aprobada), en qué lugar está colocada en categorización hecha por **Programación Detallada**, y si se encuentra en proceso en los **Centros de Producción**. Debe examinar si puede ser asignada al lote en producción, al siguiente lote (lote en cola) o si es imposible ser producida por diversas razones (agotamiento de inventario, cierre de la producción del ítem debido a la temporada, etc). Además debe gestionar las órdenes de alta prioridad notificando directamente a **Control de Ejecución** y **Control de la Programación** sin la aprobación de ningún otro módulo.

¹³ Ítem existente en medio de un proceso, que no sea convertido todavía, ni en un sub-producto, ni en un producto final, según el diccionario de PSLX.



Se encarga de ajustar la cantidad de oferta y demanda desde el punto de vista de toda la empresa. Puede también cambiar el “plan de ventas” y el “plan de capacidad” si así lo requiere.

2.6.7.2. Control de la Programación

Se encarga de la rectificación dinámica de los planes diseñados por cualesquiera de los tres niveles de **Planeación y Programación de la Producción**. Es dinámica porque puede realizarse en cualquier momento y a cualquier nivel de planificación; dicho dinamismo da lugar a la “optimalidad progresiva” de la planeación.

El **Control de la Programación** coordina, mantiene, actualiza y corrige cualquier brecha de información entre los módulos copartícipes, quienes son: Módulo de Planeación y Análisis, PRO, Programa detallado, Gestión de Mantenimiento, Gestión de Inventario y Control de Ejecución y Centros de Producción [PSLX White Paper, 2005].

Como un ejemplo muy simple se puede exponer lo siguiente: En una fábrica de zapatos se han ordenado - a través de **Compras** - suelas para zapatos deportivos y casuales, cuando el personal de recepción – en **Manejo de Materiales** – recibe el material, hace el conteo y pruebas de calidad e informa que sólo llegó ochenta por ciento de la cantidad de suelas de zapatos requeridas. Como **Gestión de Inventario** tiene conocimiento de esto por su relación de datos directa con **Manejo de Materiales**, el control en cuestión debe informar del imprevisto a los diferentes niveles de planificación -aunque previamente se haya revisado el “stock de seguridad” en **Almacén** y haya sido posible o cubrir o no, la falta de material- pero sobre todo debe informarse a **Programa Detallado**, que es la programación última antes de la ejecución del plan y la más cercana a los centro de producción, para reajustarlo momentáneamente y formular el reclamo y/o petición inaplazable a los proveedores a través de **Compras**.



2.6.7.3. *Control de Ejecución y Centros de Producción*

Debe administrar la utilización de los equipos en el orden de sucesión del proceso productivo, que no es más que: El orden en que los trabajos pasan por las máquinas o estaciones de trabajo [PSLX-02, 2003].

Indica a **Almacén** cuándo debe hacer la petición a **Manejo de Materiales**, respecto al transporte de los materiales recibidos por dicha entidad hasta el **Almacén**, y del **Almacén** a los respectivos centros de producción en **Producción** y/o a las **Redes de Distribución**. Para esto debe tener conocimiento del plan o política de inventario que demanda de **Gestión de Inventario**. Según sea el caso y con la ayuda de **Control de la Programación**, los materiales pueden enviarse directamente desde **Manejo de Materiales** hasta los centros de producción en **Producción**, sin obligatoriamente pasar por el **Almacén**.

El **Control de Ejecución** se encarga de coordinar: Las órdenes urgentes designadas por **Control de Ordenes**, el plan de ejecución actualizado que maneja **Control de la Programación** y las estadísticas de la ejecución física de los procesos que manipula **Control Real y Análisis de Rendimiento**, con la mira puesta en optimizar las rutinas de los centros de operaciones en **Producción**.

En cada centro se realiza cierto número de elementos de trabajo¹⁴, los tiempos para realizar los elementos, se derivan de los estudios de medición del trabajo conseguidos a través de **Control Real y Análisis de Rendimiento**. El período (intervalo) permitido para terminar las operaciones en cada estación se determina por la velocidad de la línea de ensamble; suponiendo que todas las estaciones de trabajo comparten el mismo tiempo de ciclo permitido, se crea entonces tiempo “ocioso” o flotante para una estación, cuando el trabajo que tiene asignado requiere menos tiempo que el establecido por el ciclo; a este proceso se le denomina: “Balanceo de Línea” [PSLX-02, 2003]. El objetivo del balanceo de línea es reducir el tiempo ocioso, al mismo tiempo que se asignan las operaciones a las estaciones de trabajo, de acuerdo con un orden sucesivo tecnológico determinado

¹⁴ En pro de la coordinación, integración y optimización de las actividades que conforman los procesos de negocio



previamente, que es calculado a través de **Control de la Programación**. Se logra un equilibrio perfecto cuando las asignaciones no dejan tiempo ocioso, ahora bien, el interés no debe centrarse exclusivamente en lograr un equilibrio perfecto (que no es sino ideal), sino en obtener distribución y flujo óptimos en relación con el resto de las operaciones de producción.

2.7. Subsistema Compras

Contiene un módulo que, en conjunto con las interfaces correspondientes, provee servicios a otros subsistemas; éste se muestra a continuación:

2.7.1. Compras (*Centro de Gestión*)

Este módulo cumple una función de servicio, que como otros, apoya las actividades de distintas operaciones y al mismo tiempo recibe la asistencia de otras unidades de operación.

Recibe de **PRO y Solicitud de Materiales** información que indica la cantidad de materiales y/o subproductos que necesitan ser surtidos por los proveedores, para satisfacer la producción prevista en el plan desarrollado por dicho módulo. En **Compras** se manejan los “convenios de reciprocidad”¹⁵, registros de elaboración, revisión de facturas y otros detalles financieros de la transacción con materiales; maneja tiempos de entrega, listas de proveedores, entre otros afines [PSLX White Paper, 2005]. Acepta órdenes o cambios urgentes designados por **Control de Ordenes** a través de **Programa Detallado**.

Una vez que recepción -en el módulo **Manejo de Materiales**- acepta un embarque, es desembalado y el personal informa la cantidad y calidad de los suministros recibidos, de manera que **Compras** usa esta información para evaluar el servicio de los

¹⁵ Según los cuales se le compran materiales a un proveedor, que corresponde comprando los productos elaborados por el comprador



proveedores. El ciclo de compras comienza con la decisión de comprar materiales, y termina cuando el material es aceptado por la unidad de recepción que notifica al módulo que dio lugar a la orden de compra.

Normalmente, **Compras** no tiene la autoridad para sustituir o modificar los materiales designados por las unidades de operación; pero sí está obligado a poner en duda las requisiciones y a proponer alternativas hábiles de lograr mejores precios [PSLX White Paper, 2005].

2.8. Subsistema Producción

Contiene un conjunto de módulos que colaboran entre sí para proveer –en conjuntos con las interfaces correspondientes- servicios a otros subsistemas; éstos se muestran a continuación:

2.8.1. Producción (Centro de Gestión)

En este módulo se indican y ejecutan las operaciones necesarias para producir los componentes, así como el orden sucesivo en que normalmente se realizan, con la ayuda de **Ingeniería de Herramientas y Materiales** y a través del **BOM**. Aquí también se especifica el centro de trabajo en el que se efectúa cada operación, el tiempo requerido de preparación y procesamiento, el equipo y los materiales.

Producción recibe de **Manejo de Materiales** y de **Almacén** -en cada uno de sus centros de trabajo, en los pisos de planta- las cantidades especificadas en el último programa de producción, de materiales y sub-productos requeridos. También permite a **Control Real y Análisis de Rendimiento** mantener una constante verificación de los datos que arrojan los centros de ejecución, a manera de registros del itinerario de producción.

El módulo **Producción** está comprometido a pedir (a **Manejo de Materiales**) el transporte de sus productos finales hacia las **Redes de Distribución** y **Manejo de**



Desechos, según lo amerite el programa y con la ayuda de **Control de Ejecución y Centros de Producción**.

Producción necesita unos módulos que asisten y cooperan con éste. Están clasificados de la siguiente manera:

2.8.1.1. Control Real Y Análisis de Rendimiento

Esta unidad monitorea el fenómeno principal de agregación de valor al producto en proceso, a través de cada centro de operaciones. Para ello, mantiene registros de itinerario que proporcionan datos útiles para calcular las necesidades de capacidad, para controlar la producción en proceso.

Los registros del centro de trabajo definen los recursos de fabricación cuyas funciones son iguales o similares. La información incluye el número de máquinas y operadores, la capacidad estándar, el número de turnos y la eficiencia de cada centro de trabajo; también incluye los detalles de la producción en proceso, como por ejemplo: tiempos de corrida y de preparación, tiempos de espera, tiempos de transferencia, tiempos de movimiento y haciendo cola, inventario restante de protección, cantidad terminada, etc. Toda esta información se envía al **Módulo de Planeación y Análisis** para coordinar las estadísticas a largo plazo, a **Control de la Programación** para coordinarlas a mediano y corto plazo, y a **Gestión de Mantenimiento** para mantener los registros exactos de la funcionalidad de los recursos.

2.8.1.2. BOM: Matriz Insumo Producto

Al igual que el BOM de Diseño, el módulo **BOM** de manufactura maneja datos técnicos, como las listas de materiales y datos de enrutamiento de producción. La información incluye no sólo datos formales y aprobados, sino también datos informales que pueden no ser consistentes con los datos formales.

La diferencia entre este **BOM** y el de **Diseño**, es que éste contiene la información original en el lado de manufactura en función a las posibilidades de manufactura en cada



piso de planta, la información se envía a **Ingeniería** para la reestructuración del **BOM** de **Diseño** según la “realidad” de las líneas de producción y los pisos de planta.

2.8.1.3. Redes de Distribución

Las **Redes de Distribución** reciben los sub-productos o productos terminados por parte del **Almacén de Producción**, para ser distribuidos según la planificación realizada por la misma entidad.

2.8.1.4. Manejo de Desechos

Como su nombre lo indica, este módulo administra los siguientes desechos:

Ítems defectuosos, recursos consumibles que generen desechos, recursos no consumibles que hayan sido sustituidos en la ejecución de un programa de mantenimiento de cualquier índole. También maneja desechos generados en los departamentos de trabajo en cada centro de gestión, así como en cualquier otro módulo distribuido en cada subsistema; entre otros.

CAPÍTULO III

DESCRIPCIÓN DEL MODELO Y SUS PROTOCOLOS EN UML

www.bdigital.ula.ve

El aspecto más crucial de una clase es que tiene “semántica” (algo acordado sobre significado entre los que están comunicándose). Por ejemplo, cuando se discute “previsión de ventas”, qué significa previsión de ventas para la audiencia (las demás clases u subclases que conocen el término), ¿es un esfuerzo que dura una semana o un año? ¿Requiere del uso de un controlador o de otros recursos humanos? etc. Este significado es muy específico para la audiencia y el dominio en el que la clase es usada.

Para que estas clases o módulos se entiendan al comunicarse, primero deben lograr enviar sus objetos-mensajes desde un subsistema, a alguno diferente. Para lograr este fin, a lo largo de este capítulo se desarrollan los “protocolos de envío de objetos” entre subsistemas, en donde serán usados diagramas funcionales en UML (específicamente *Diagramas Secuenciales*), como las pruebas de cooperación pertinentes a realizar, para comprobar el envío exitoso de objetos a través de interfaces que, aunque iguales en denominación, son ligeramente diferentes en su estructura, según el subsistema donde se las encuentre.



Para conocer la información que las interfaces manejan, se definen “componentes de información”. Estos sirven para organizar la información que manipulan las interfaces, en conjuntos o componentes de data, que simultáneamente con los cuadros del Anexo D, definen las funciones principales de cada una; de esta manera, se disponen de manera organizada las funciones de cada interfaz dentro el modelo propuesto, a la hora de una futura implementación del mismo y sus interfaces.

A través de los diagramas de UML, se observarán las clases organizadas según lo que sus objetos “conocen” (características estructurales) y “pueden hacer” (características de comportamiento). Las características estructurales incluyen valores de atributos y enlaces.

Entre las definiciones básicas que se deben manejar para entender un diagrama UML están:

- **Atributo:** Un atributo es lo que un objeto de una clase conoce. Es un elemento de datos mantenido por el objeto.
- **Operación:** Una operación es, lo que el objeto de una clase puede hacer. Es una especificación de un servicio provisto por el objeto.
- **Método:** Un método es cómo un objeto de una clase hace su procesamiento. Es la implementación de un servicio provisto por el objeto [Sinan, 2003].

3.1. Justificación del Uso de Subsistemas e Interfaces en UML Versus el Uso de Paquetes

Mientras un paquete simplemente agrupa elementos, un subsistema agrupa elementos que juntos proveen servicios tales que otros elementos, pueden acceder sólo esos servicios y a ninguno de los elementos en sí.



Y mientras los paquetes permiten particionar el sistema en grupos lógicos y relacionar estos grupos lógicos, los subsistemas permiten considerar qué servicios estos grupos proveen uno al otro.

Para proveer un servicio un subsistema contiene: las operaciones, elementos de especificación e interfaces; éstos pueden ser desarrollados en paralelo porque cualquier interdependencia entre ellos, recae sólo en los servicios que provean. Debido a que los servicios de un subsistema son completamente especificados, se puede reutilizar el subsistema cuando sea que se necesiten servicios similares [Sinan, 2003].

3.2. UML y las Interfaces del Modelo

Las mayoría de las interfaces que se utilizan en los diagramas estructurales y secuenciales representantes del modelo planteado, pertenecen a las interfaces definidas en el documento “PSLX-02, APS Agent Model“ pero ordenadas y denominadas, de manera que puedan ser utilizadas en el modelo en cuestión.

No es necesario hacer uso de todas las interfaces definidas por PSLX, se mostrarán las que se consideren necesarias¹ para cada subsistema, además se asume que las interfaces ya están programadas adecuadamente bajo un esquema holónico, de esta manera se puede hacer énfasis en las asociaciones que permiten a los centros de decisiones colaborar entre si.

Más adelante se explicarán los servicios que proveen las interfaces que se utilizan en el modelo; además, se mostrarán diagramas en donde se indicarán las interacciones sectorizadas entre un subsistema externo y PPAP, luego entre PPAP y un subsistema externo y por último, entre dos subsistemas externos.

¹ De hecho se utilizaron varias interfaces adicionales que se consideraron necesarias, y que no estaban definidas por PSLX, véase anexo D



Es importante aclarar, que no se definirán las interfaces que conectan las clases o módulos entre sí y con el centro de gestiones correspondiente; se asume que éstas existen. Esto representa materia de estudio para otro proyecto.

3.3. Intercambio de Información

Cada intercambio de información que se explica a continuación, se llamará “*componente de información*”. Corresponde a un servicio principal provisto por una interfaz, o a uno necesario para complementar la formación de un servicio principal en una; relacionándose esta última directamente con algún subsistema.

Se explican todos los intercambios de información, como partes de la data que se envía entre subsistemas a través de las interfaces². No se explicarán las operaciones internas, que ejecuta cada interfaz para llevar los “objetos” de una interfaz a otra interfaz, o a un módulo en específico, ya que se asume que estas interfaces ya están programadas con los métodos y operaciones necesarias, para llevar a cabo esta transmisión de mensajes bajo un esquema holónico.

La forma en que se envía la información (a través de mensajes) entre los centros de decisión de los subsistemas y las interfaces correspondientes, y los algoritmos que manejan la secuencia en la transmisión de mensajes -como se muestra en los diagramas secuenciales en UML- definen la estructura de la mensajería que utiliza este modelo, y por ende sus protocolos. No obstante, debe tomarse en consideración, que la parte de los protocolos referente al formateo³ de los datos, de los objetos que se envían, será implementada en el capítulo siguiente (cap. IV)

El contenido de la información de intercambio entre los subsistemas: Ventas, Compras, Diseño, Producción y Planeación y Programación Avanzada de la Producción

² Que se podrán observar en los diagramas UML

³ Formateo no referente a la “presentación” de los datos, sino a la estructuración para ser entendidos dentro del sistema de información en donde se manejan.



(PPAP), puede ser clasificado, en varios componentes de información, que se mencionarán a continuación.

3.3.1. Componentes de Información

Puede considerarse que mucha de la información de intercambio, por ejemplo, entre PPAP y Compras (suplidor), se logra sustituir con el intercambio de información que existe entre PPAP y Ventas desde un punto de vista diferente (cambiando el origen y destino de la petición); por lo tanto, los componentes utilizados para transmitir información entre Compras y PPAP pueden sustituirse por los componentes de información utilizados entre PPAP y Ventas [PSLX-02, 2003].

En muchos casos ocurre lo expuesto en el párrafo preliminar, por esto, y tomando en consideración la explicación que se realizará en la sección final de este capítulo: “*Definición de Interfaces*”, es necesario aclarar que el inquisidor y suplidor del servicio (roles que se indican en cada componente de información) pueden y necesariamente cambian de perspectiva⁴; es decir, en donde desde un punto de vista uno pide un servicio y otro sirve, desde el punto de vista inverso, el que sirve ahora pide el servicio.

De esta manera, los componentes de información exponen -más que todo- lo que se quiere transmitir entre origen y destino, a través del servicio que se ofrece mediante las diversas interfaces ubicadas a lo largo de todos los subsistemas. Si en la explicación que sigue se especifican orígenes y destinos puntuales, es para resaltar mejor la función de ese componente de información, y además porque ocurre probablemente entre ellos (orígenes y destinos especificados) con mayor frecuencia que en otros casos. Algunos casos son exclusivos de los protagonistas que se nombran en cada definición, de cualquier manera, esto se especifica en la sección “*Definición de Interfaces*”.

Es importante resaltar además, que cada vez que se haga referencia a un intercambio de información en favor de un “cliente” en las definiciones que siguen, debe entenderse que se realiza a través de Ventas, que es el módulo a través del cual “cliente” entra en el modelo

⁴ Que depende netamente de quién inicie la comunicación y hacia quien va el mensaje



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

de sistema que se expone a los largo de este estudio; esta analogía también se aplica entre las referencias hechas a un “suplidor” y Compras.

Entre los componentes de información, que en su mayoría⁵ están definidos por PSLX [PSLX-02, 2003], y que conforman las interfaces que los subsistemas, Ventas, Compras, Diseño, Producción y PPAP pueden proveer, están:

Información de Estimación. La información de estimación, es una respuesta por parte de PPAP respecto al número de producto de cualquier ítem aplicable, fecha de entrega, cantidad de productos producibles, a una “petición de estimación” de un cliente. Una petición de estimación desde PPAP o cualquier otro centro de decisión o una configuración de estimación desde Ventas a cualquier centro de decisión, contiene además de lo especificado, la “previsión de ventas”. La información de estimación propia del cliente, puede ser enviada a Ventas.

Información de Orden. Esta es la información que representa el contenido de una orden del cliente. Cuando la orden es recibida (por Ventas y enviada a PPAP), PPAP usualmente contesta aceptación, reservación ó rechazo. También la información, como “cambio o cancelación de la orden” hecha, puede ser enviada desde un cliente (Ventas) a PPAP y desde PPAP a Producción.

Información No-Oficial. Información No-Oficial, es la información contenida en las órdenes que serán realizadas en un futuro cercano, enviada desde el cliente; no es una orden oficial.

Información de Opción-Datos. La información del contenido concreto de una especificación de producto en la información de orden puede ser enviada por separado del cliente (Ventas) a PPAP. Opción-Datos es la información acerca de la especificación del producto ordenado por un cliente. Opción-Datos es enviada en caso de una orden no oficial y también en caso de una orden definitiva. Inversamente, la información, “petición de

⁵ Los componentes de información que no están definidos por PSXL, son propuestos por el autor.



decisión Opción-Datos” para preguntar las opciones de datos, es enviada desde PPAP a un cliente (Ventas).

Información de Tiempo-De-Entrega (Duetime). Esta información de tiempo de entrega, es la información acerca del tiempo del que se dispone para entregar una orden a un cliente, ya sea una orden definitiva o no-oficial. Hay muchos casos en los que esta información es enviada desde PPAP a un cliente como respuesta a una petición de “tiempo de entrega”. Alguna de esta información de tiempo de entrega puede ser utilizada para prometer la “fecha de entrega” (due-date).

Información de Queja. Esta es la información enviada desde un cliente a PPAP (a través de Ventas), cuando cualquier falla ocurre luego de que el cliente recibe un producto (debe recordarse que, el cliente puede ser PPAP y el suplidor Compras, o el cliente puede ser Producción y el suplidor PPAP, etc.). Algunas medidas, como “procedimiento de queja” deben ser tomadas por PPAP respecto a esta queja.

Información de Notificación-Envío. Esta es la información para notificar que un producto es enviado para la orden definitiva. Cuando un cliente recibe la notificación, el cliente puede prepararse para la aceptación del producto de manera anticipada gracias a la recepción de esta información.

Información de Notificación-de-Recibo. Esta es la información para probar que el contenido y la cantidad de productos enviados a un cliente, son correctos luego de haber suplido el producto al mismo. Esta información es enviada desde un cliente a PPAP durante o luego de su entrega de productos. Esta información es utilizada para ajustar posteriormente las cuentas.

Información de Producto. Esta es la información que se envía a un cliente respecto a los datos de productos distribuidos, antes de recibir la orden del cliente (Ventas). Aquí se incluye, información sobre un nuevo producto, detenerse a la fabricación de un producto, su especificación detallada o su precio. Esta información puede ser enviada a un cliente por



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

decisión del lado de PPAP (a través de Ventas), puede ser enviada para responder a la consulta de un cliente.

Información de Almacén (Stock). Esta es la información para mostrar el depósito o almacén de productos distribuidos a un cliente. Abriendo esta información, el cliente puede juzgar si el producto puede ser suplido en un tiempo de entrega corto. Generalmente, esta información de stock es mostrada al cliente respondiendo a una “petición de inventario” hecha por el mismo. Desde otro punto de vista, esta información de stock abre el depósito de materiales del cliente, para ser conocido por PPAP. Sabiendo los productos de la compañía en stock, puede ser previsto en cierto modo, el momento y el número de productos en que un cliente puede ordenar.

Información de Plan. Esta información es adelantada por la información de depósito abierto, también abre el programa para mostrar el número de productos a ser terminados en el futuro cercano, excepto los productos ya terminados; de cualquier manera, este caso es entre empresas que cooperan de manera cercana.

El plan puede ser enviado desde PPAP a Ventas (y este a su vez a un cliente) regularmente, y los ítems requeridos pueden ser mostrados según la petición de un cliente.

Desde otro punto de vista, esta información es adelantada desde el stock abierto de información de Ventas (que refiere a la información de stock abierto del cliente), y abre el plan o programa de producción del cliente (a través de Ventas) para ser visto por PPAP. Entonces las órdenes futuras de un cliente pueden ser previstas de manera bastante precisa. Es posible decir, que la información no-oficial es una de las formas de abrir esta información de plan.

Información de Plaza-de-Producción. Esta plaza de producción es mostrada a un cliente como medidas para indicar la cantidad de productos a ser terminados sin tener que abrir la información del plan. La plaza de producción muestra sólo el número de productos a ser terminados en cada unidad de tiempo de producción (iteración) y encubre la información interna del programa.



Información de Demanda. Esta es la información de los planes futuros, que no va a ser reflejada, ni en el plan de producción del cliente, ni en el plan de ventas, ni en la previsión de ventas creada por un cliente, y que son mostradas a PPAP a través de Ventas. Puede decirse que esta información de demanda es mostrada para compartir la información con respecto a planeación colaborativa, previsión y reposición.

Información de Innovación. Se refiere a la información de los productos más innovadores, los más actualizados, que se encuentran o podrían salir al mercado, para abrir un nuevo sector en él o para competir en un sector que ya existe, pero en el que no se ha participado aún.

Información del Método-De-Manufactura. Esta es la información en relación a qué equipos y qué condiciones son necesarias para realizar los productos individuales. Esta información además provee restricciones detalladas de la producción individual y un indicador para calidad.

Información del Equipo-De-Producción. Esta es la información en relación a qué equipos deben ser usados para realizar productos. Esta información incluye aquella acerca del equipo alternativo.

Información de Manufactura-Horas-Hombre. Esta es la información acerca de cuántas horas-hombre son requeridas de acuerdo a la especificación del producto en cada proceso de producción.

Información de Diseño. Esta es la información acerca del diseño tal como estructuras del producto, partes, especificaciones e información de dibujo, etcétera. *La Matriz-De-Insumos* es administrada y modificada, con base en la información de la estructura del producto según sea la ocasión. La mayoría de la información es directamente enviada al campo de manufactura. Si el diseño es cambiado, esta nueva información de diseño debe ser enviada al campo manufacturero.



Información de Calidad. Como un resultado de la producción, La información acerca de fallas o “sectores de mejoramiento” es adquirida. Tal información, es enviada desde PPAP hacia Diseño, como la información de calidad.

Información de la Matriz-De-Insumos de Manufactura (BOM). Esta es la información de la *Matriz-De-Insumos* con la información original en el lado de manufactura o producción. Esta información difiere de la *Matriz-De-Insumos* hecha por la parte de diseño, ya que el contenido de la información de diseño puede ser cambiado cuando se toman en cuenta las posibilidades de manufactura.

Información de la Matriz-De-Insumos de Diseño (BOM). Esta es la información de la *Matriz-De-Insumos* original en el lado de diseño. Esta información puede ser modificada en base a las posibilidades de manufactura y a los nuevos materiales de innovación proporcionados por Ingeniería de Herramientas y Materiales.

Información de Innovación. Esta es la información acerca de las nuevas estructuras, herramientas, materiales, partes y especificaciones que se necesitan para sustituir la fabricación de los productos actuales o para elaborar nuevos. *La Matriz-De-Insumos de Diseño* es administrada y modificada, con base en la información de la estructura de los nuevos productos o de las nuevas herramientas y materiales utilizados, según sea la ocasión. La mayoría de la información es directamente enviada al campo de manufactura junto con la información de *Matriz-De-Insumos de Diseño y Especificación-del-Proceso*.

Producción provee una respuesta respecto a la asimilación de la información de innovación recibida, es decir, la adaptabilidad y factibilidad de las líneas de producción actuales a estos nuevos materiales y/o herramientas.

Información de Petición-Individual-De-Producto. Cuando diseño debe ser operado basado en peticiones individuales del cliente en la producción individual de “producir-por-orden”, la petición del cliente es enviada desde *Planeación y Programación*



de la Producción a Diseño, y desde Ventas a Diseño, como una petición-individual-de-producto.

Información de Opción-De-Producto-Individual. Como resultado de que el diseño progresa basado en la petición de un cliente, en la “producción por orden” individual, la opción individual a la petición del cliente es enviada desde *Diseño* a *Ventas* (hasta *Cliente*), y desde *Diseño* a *Planeación y Programación de la Producción*. La producción es ejecutada basada en el contenido de esta opción en el campo de producción.

Información de Progreso-De-Producción. Como resultado de la producción en el campo manufacturero, la información acerca de la cantidad de productos, es enviada a *Planeación y Programación Avanzada de la Producción* como progreso de producción. Hay casos donde el progreso de producción es obtenido en cualquier momento de la misma, y donde el progreso de producción es obtenido en un período de tiempo específico.

Información de Progreso-De-Culminación. Esta es la información relacionada con la terminación o no, del proceso de manufactura para el programa de manufactura en el campo de producción. Progreso-De-Culminación puede ser obtenido en cualquier momento en que la producción sea terminada, o puede ser obtenido en un período de tiempo específico.

Información del Progreso-De-Operación. El progreso de producción es el resultado de la información en el lado de producción, tal como, cantidad de ítems producidos. Por otra parte, el progreso-de-operación es la información resultante en la parte de equipamiento. Como resultado del término de la producción, la información acerca de las condiciones operacionales de las instalaciones, es enviada. El progreso-de-operación es usualmente enviado al haber sido recolectado para un período específico.

Información del Progreso-De-Inspección. El progreso de inspección muestra si el producto satisface la calidad requerida o no. El producto rechazado por pruebas, es abolido o reproducido. El momento de envío de progreso-de-inspección es el mismo que el de progreso-de-producción.



Información de Anormalidad-De-Instalación. Esta es una forma de progreso de operación de instalaciones. La información de anomalía-de-instalación es enviada a *Control de Ejecución* cuando alguna instalación opera de manera anormal.

Información de Alarma-De-Calidad. Esta es una forma de progreso-de-inspección. La anomalía puede ser notificada por decisión humana aunque el proceso de inspección de rutina no está especialmente establecido. La información acerca de la alarma-de-calidad es enviada a través de *Control Real* a tiempo real e inmediatamente *Control de Ejecución* toma medidas.

Información de Anormalidad-De-Progreso. El campo manufacturero observa los materiales que llegan retrasados, o los procesos específicos que terminan mucho después del tiempo esperado, y luego juzgan qué influencia tiene esto en la producción total. Las condiciones son inmediatamente enviadas a *Control de Ejecución* como una anomalía-de-progreso.

Información de Programa-De-Producción. Como un resultado del programa de producción, *Control de Ejecución* envía el “programa de producción” para cada instalación, piso de planta y/o trabajador, en la sección de producción. Esta programación, puede ser enviada en cualquier intervalo o período de tiempo, o puede ser enviada a tiempo real. Además, aquí se envía información acerca de la corrección o cancelación del primer programa de producción emitido o publicado.

Información de Programa-De-Envío. Esta es la información que indica el envío de productos. Ésta es enviada considerando la fecha de entrega que se especifica cuando los productos en *Almacén* son coordinados con las órdenes respectivas de despacho, o luego de recibir la información (por parte de progreso-de-producción) de que los productos finales están en el almacén de productos terminados.

Información de Instrucción-De-Liberación. Esta información ordena el envío del material del *Almacén* o el “trabajo en proceso” (work in progress) para ser manejado con la



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

finalidad de comenzar la producción. Este programa es dado para suministros que saldrán de la empresa o compañía.

Información de Programa-De-Mantenimiento. Esta es la información en la que *Control Real* ordena a la parte de *Producción* (a través de *Gestión de Mantenimiento*) realizar el mantenimiento de las instalaciones y procesos junto a operaciones relacionadas con *Producción*. El campo de *Producción* o manufactura puede ejecutar el mantenimiento por su propia cuenta, pero no es el objetivo de este programa.

Información de Petición-De-Mantenimiento. Aun si *Producción* realiza el mantenimiento independientemente, es necesario ajustar el mantenimiento a todo el programa de producción. *Producción* le pide a *Control de Ejecución* no usar el equipo por un período de tiempo (fijado para manufactura), con el fin de realizar un mantenimiento.

Información de Especificación-de-Proceso. *Ingeniería* envía la información de ingeniería requerida para establecer un nuevo proceso, mejorando el modelo para un nuevo producto y adaptándose a las posibilidades de sus instalaciones en el campo manufacturero. El contenido de la información enviada, no es reconocida en *Planeación y Programación de la Producción* por lo que la información de ingeniería enviada desde *Diseño* es entregada a *Producción* (como la especificación-de-proceso) tal y como fue recibida.

Información de Lote. Un “Lote” indica la existencia concreta de un “ítem” consumido o producido por una instrucción de producción, y corresponde a una parte del “almacén”. Más detalladamente, es una información de identificación, que indica un grupo de ítems producidos e ítems consumidos, cuando algún ítem es concretamente producido o consumido por una instrucción de producción. El almacén puede ser calculado agregando cada lote con fecha determinada,

Información de Tarea. Un tipo especial de “lote” es una “tarea”. Tarea indica la existencia concreta de un recurso, asignado por una instrucción de producción. Esto corresponde a una parte de capacidad del recurso requerido para ejecutar la instrucción de producción.



3.4. Protocolos para el Envío de Objetos de Información entre PPAP y demás Subsistemas

Aquí se define la forma en que se transmiten los mensajes entre los subsistemas; cuáles son las interfaces publicadas al exterior desde adentro (por parte de PPAP), y cuáles son publicadas al interior desde afuera (por parte de los otros subsistemas). La función provista por el sistema, concreta en que PPAP realiza cualquier procesamiento para el mensaje recibido desde cualquier subsistema externo a él, a través de la interfaz adecuada, y envía un mensaje de regreso de manera síncrona o asíncrona, siguiendo un camino, que contempla el recorrido del objeto, desde que sale de un centro de gestión o decisiones, hasta que llega a otro (naturalmente en un subsistema diferente), a través de ciertas interfaces según sea el caso. De igual manera para el caso inverso, en que cualquier subsistema externo realice el procesamiento para objetos provenientes de PPAP.

La petición, respuesta y cualquier otro tipo de mensaje que se envíe entre interfaces, se realiza en forma de “objetos-petición” y “objetos-respuesta”, que se generan gracias a la cooperación entre los módulos dentro de cada subsistema, su centro de decisiones correspondiente y la base de datos ubicada junto a dicho centro de decisiones.

La información que se pasa como “objetos de petición”, sirve para “obtener” o “configurar” (“get” ó “set” respectivamente) información de negocios en un subsistema específico. Un objeto de petición sale desde los centros de gestión ubicados en los subsistemas: **Compras, Ventas, Diseño, Producción y PPAP**, a las interfaces responsables de enviar la información hacia el centro de gestión objetivo. Éste responde con un “objeto de respuesta” que recorre “la misma ruta” o camino, -a través de las interfaces- que el objeto de petición, para así responder a cada solicitud en cada subsistema correspondiente.

3.4.1. Sistema de Intercambio de Mensajes

Los objetos recibidos por PPAP o un subsistema externo están (a grosso modo) divididos en tres “tipos de mensajes”:



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

- Mensaje para enviar los datos o información de negocios requerida por un subsistema (tipo de información de envío).
- Mensaje para obtener la información de negocios o datos de un subsistema requerido particular (tipo de información de búsqueda o petición).
- Mensaje para ejecutar el procesamiento de negocio requerido o procedimiento requerido por un subsistema (tipo de información de procesamiento).

Por otra parte, estos mensajes son desarrollados en varios tipos aplicables:

Sea el proceso de mensajería activado desde afuera o desde adentro, existen tres patrones “básicos” para los mensajes: aceptación (acepta), refutación (rechaza) y recibimiento (recibido) tal como la figura 3.1 muestra. Además, en la figura 3.2 se observa, que existen además, patrones en los que sólo un mensaje de “recibido” es retornado (para contestar a una petición), y una respuesta es “reservada” en el momento en el que se recibe la petición, entonces el mensaje se trata mas tarde, para dar una respuesta de aceptación o refutación⁶ [PSLX-02, 2003].

⁶ Esto se tratará con mayor detalle en el capítulo IV.

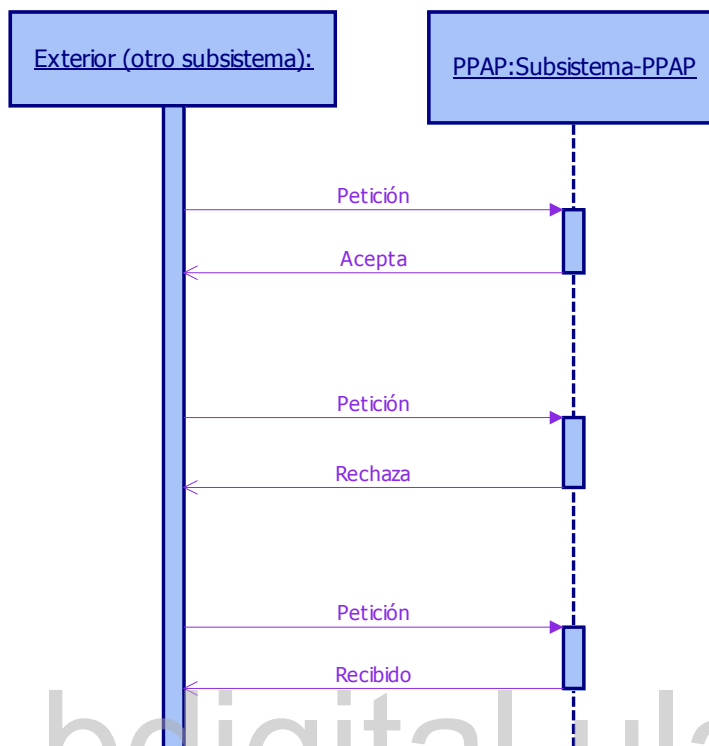


Figura 3.1. Patrones Básicos de Mensajes de Respuesta. Tomado de “PSLX Engineering Specification, APS Agent Model. PSLX-02” por PSLX Consortium. 2003.

El sistema de intercambio de mensajes en el que PPAP comienza la comunicación, es exactamente igual al descrito pero cambiando la perspectiva, es decir, “PPAP” sería ahora la instancia de subsistema colocada a la izquierda de ambas figuras 3.1 y 3.2, y la instancia “Exterior” estaría colocada a la derecha de ambas figuras.

El recorrido de los objetos mencionados, se describe en las secciones venideras del capítulo en cuestión, haciendo referencia a las figuras realizadas en UML “Diagramas Funcionales” y “Diagramas Estructurales”. El formato de los objetos se delinea en el capítulo IV: Descripción de objetos en XML.

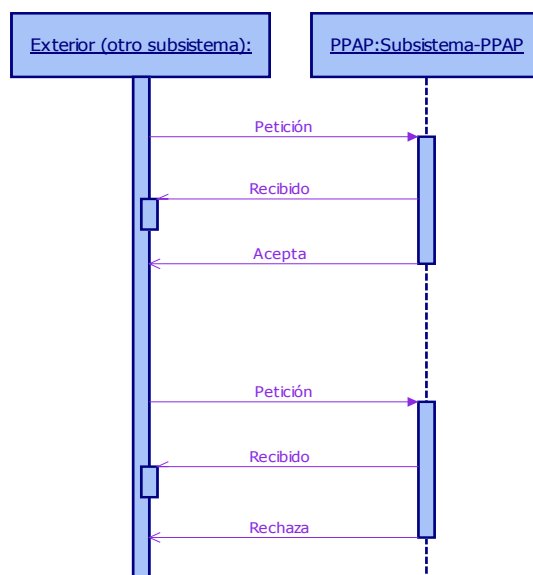


Figura 3.2. Patrones para Reservar una Respuesta. Tomado de “PSLX Engineering Specification, APS Agent Model. PSLX-02” por PSLX Consortium. 2003.

www.bdigital.ula.ve

3.4.2. Diagramas Funcionales

En esta sección se describirán los protocolos de vías o caminos de comunicación, de los objetos generados por el subsistema PPAP y sus subsistemas adyacentes. Se tomarán, como las pruebas de cooperación necesarias para comprobar el funcionamiento de la estructuración del modelo propuesto, tres ejemplos que resaltarán los tres casos generales de envío de objetos entre subsistemas.

- Ejemplo 1: Cómo se transmiten los mensajes desde un subsistema externo (Ventas) hacia otro subsistema externo (Diseño), y la manera en que estos últimos responden a las peticiones.
- Ejemplo 2: Cómo se transmiten los mensajes desde un subsistema externo (Ventas) hacia PPAP, y la manera en que éste responde a las peticiones.
- Ejemplo 3: Cómo se transmiten los mensajes desde PPAP hacia un subsistema externo (Ventas), y la manera en que estos responden a sus peticiones.



3.4.2.1. Ejemplo 1: Diagrama Secuencial VENTAS-DISEÑO

Desde un subsistema externo que realiza una petición, hacia otro subsistema externo (Subsistema Ventas =>Subsistema Diseño, véase figura 3.3):

- 1.) El “centro de gestiones” del subsistema externo -en este caso: Ventas (MBD) Centro de Decisiones- crea un objeto petición.
- 2.) El objeto petición se envía a la interfaz “Petición y Recepción de Objetos” del subsistema externo en cuestión (Petición-Recepción-Objetos-Ventas).
- 3.) Esta interfaz se encarga de enviar el objeto a la interfaz adecuada, bien sea otro subsistema externo o PPAP, en este caso, “D-Recepción y Respuesta de Peticiones-Ventas” (D-RRP-Ventas) que se encuentra en el subsistema Diseño.
- 4.) Luego, “D-Recepción y Respuesta de Peticiones-Ventas” estudia cuál interfaz debe utilizar para enviar la petición a “Diseño (MBD) Centro de Decisiones” (Opción, Orden, Innovación, etc.) y envía dicho objeto a través de la interfaz seleccionada.
- 5.) Una vez que el “Centro de Decisiones” crea el objeto respuesta, envía dicho objeto, a través de la interfaz desde donde recibió el objeto petición, a la interfaz “D-Recepción y Respuesta de Peticiones-Ventas” (D-RRP-Ventas) que se encuentra en el subsistema Diseño.
- 6.) Posteriormente esta interfaz envía el objeto respuesta hasta la interfaz “Petición y Recepción de Objetos” del subsistema externo que origina la petición (en este caso Petición-Recepción-Objetos-Ventas).
- 7.) Por último, esta interfaz envía el objeto respuesta al “centro de gestiones” del subsistema externo que generó la petición -Ventas (MBD) Centro de Decisiones-.

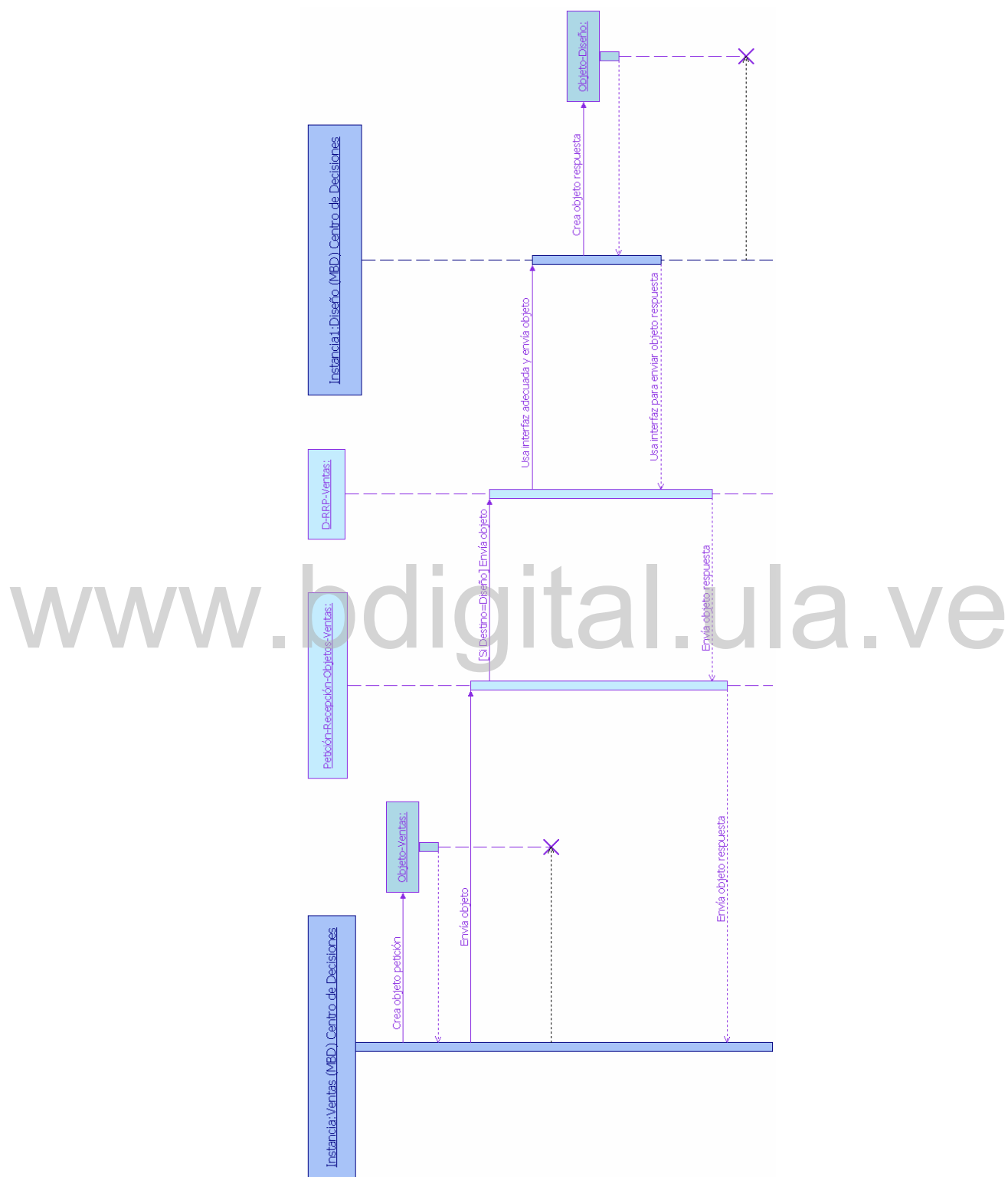


Figura 3.3. Diagrama Secuencial VENTAS-DISEÑO



3.4.2.2. Ejemplo 2: Diagrama Secuencial VENTAS-PPAP

Desde un subsistema externo que realiza una petición, hacia PPAP (Subsistema Ventas=>PPAP, véase figura 3.4):

- 1.) El “centro de gestiones” del subsistema externo -en este caso: Ventas (MBD) Centro de Decisiones- crea un objeto petición.
- 2.) El objeto petición se envía a la interfaz “Petición y Recepción de Objetos” del subsistema externo en cuestión (Petición-Recepción-Objetos-Ventas).
- 3.) Esta interfaz se encarga de enviar el objeto a la interfaz adecuada, bien sea otro subsistema externo o PPAP, en este caso, “Recepción y Respuesta de Peticiones- Ventas” (RRP-Ventas) que se encuentra en el subsistema PPAP.
- 4.) Luego, “Recepción y Respuesta de Peticiones-Ventas” estudia cuál interfaz debe utilizar para enviar la petición a “PPAP (MBD) Centro de Decisiones” (Plan, Programa, Estimación, etc.) y envía dicho objeto a través de la interfaz seleccionada.
- 5.) Una vez que el “Centro de Decisiones” crea el objeto respuesta, envía dicho objeto, a través de la interfaz desde donde recibió el objeto petición, a la interfaz “Recepción y Respuesta de Peticiones-Ventas” (RRP-Ventas) que se encuentra en el subsistema PPAP.
- 6.) Posteriormente esta interfaz envía el objeto respuesta hasta la interfaz “Petición y Recepción de Objetos” del subsistema externo (en este caso Petición-Recepción-Objetos-Ventas).
- 7.) Por último, esta interfaz envía el objeto respuesta al “centro de gestiones” del subsistema externo que generó la petición -Ventas (MBD) Centro de Decisiones-.

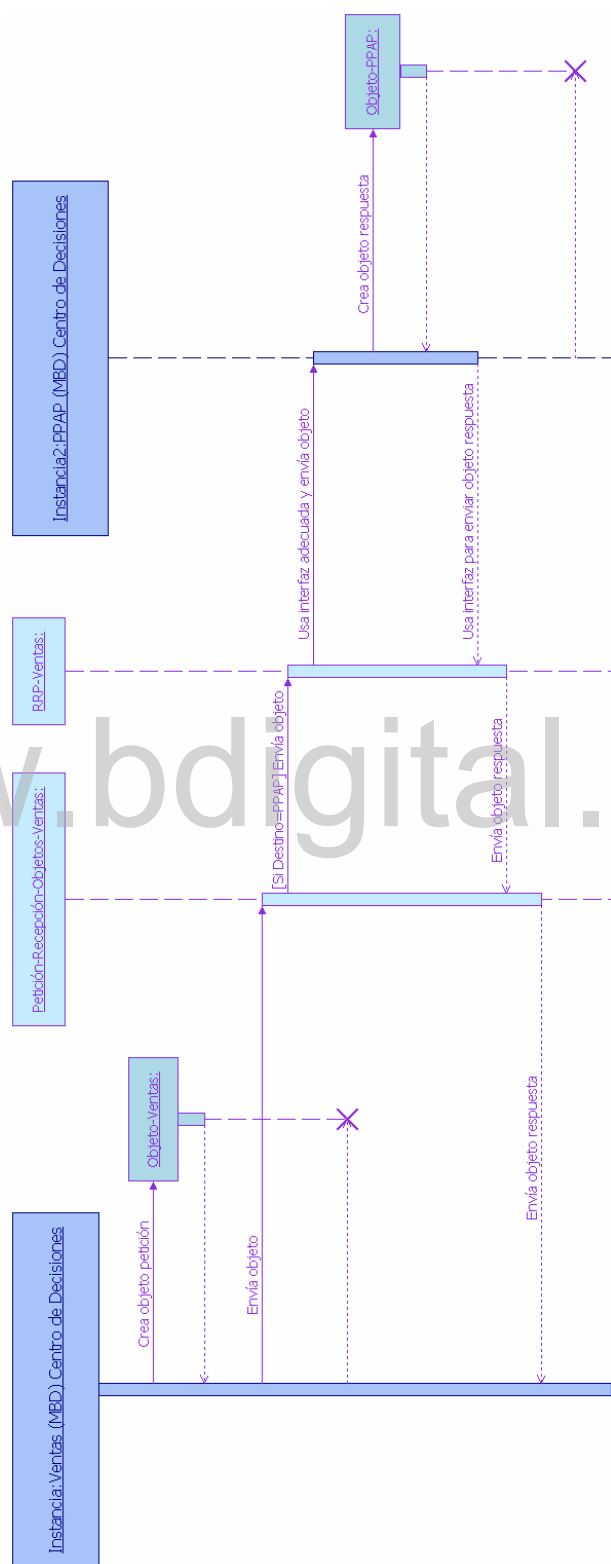


Figura 3.4. Diagrama Secuencial VENTAS-PPAP



3.4.2.3. Ejemplo 3: Diagrama Secuencial PPAP-VENTAS

Desde PPAP que realiza una petición, hacia un subsistema externo (PPAP=> Subsistema Ventas, véase figura 3.5):

- 1.) El “centro de gestiones” de PPAP -PPAP (MBD) Centro de Decisiones- crea un objeto petición.
- 2.) El objeto petición se envía a la interfaz “Petición y Recepción de Objetos” de PPAP (Petición-Recepción-Objetos-PPAP).
- 3.) Esta interfaz se encarga de enviar el objeto a la interfaz adecuada en un subsistema externo, en este caso, “V-Recepción y Respuesta de Peticiones-PPAP” (V-RRP-PPAP) que se encuentra en el subsistema Ventas.
- 4.) Luego, “V-Recepción y Respuesta de Peticiones-PPAP” estudia cuál interfaz debe utilizar para enviar la petición a “Ventas (MBD) Centro de Decisiones” (Plan, Programa, Producto, etc.) y envía dicho objeto a través de la interfaz seleccionada.
- 5.) Una vez que el “Centro de Decisiones” (de Ventas) crea el objeto respuesta, envía dicho objeto, a través de la interfaz desde donde recibió el objeto petición, a la interfaz “V-Recepción y Respuesta de Peticiones-PPAP” (V-RRP-PPAP) que se encuentra en el subsistema Ventas.
- 6.) Posteriormente esta interfaz envía el objeto respuesta hasta la interfaz “Petición y Recepción de Objetos” de PPAP (Petición-Recepción-Objetos-PPAP).
- 7.) Por último, esta interfaz envía el objeto respuesta al “centro de gestiones” de PPAP, que fue el generador de la petición -PPAP (MBD) Centro de Decisiones-.



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

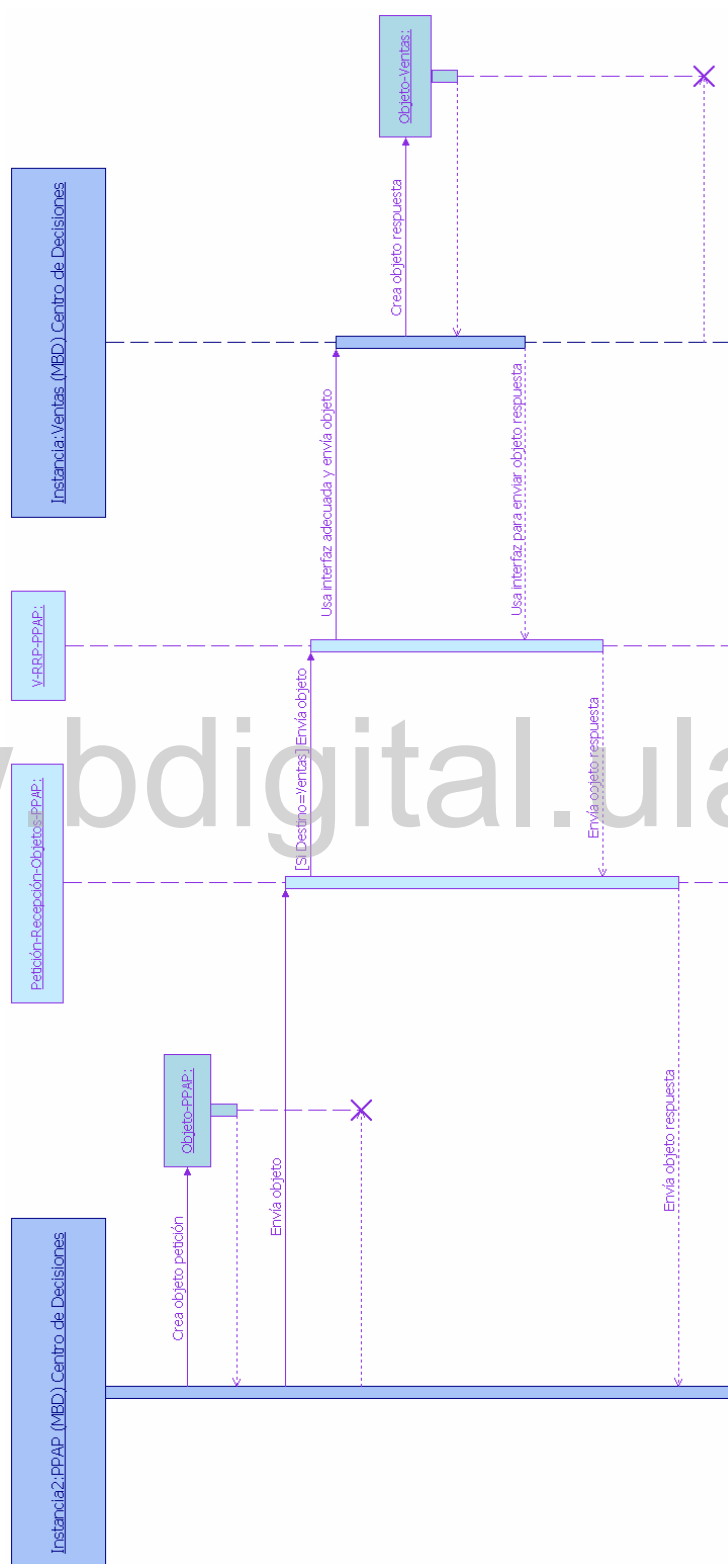


Figura 3.5. Diagrama Secuencial PPAP-VENTAS



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

En las figuras 3.6 y 3.7 se ilustra, a través de diagramas secuenciales, las posibilidades de peticiones que tienen los subsistemas externos, respecto a otros subsistemas (Ej. Ventas, puede hacer peticiones a Diseño ó PPAP).

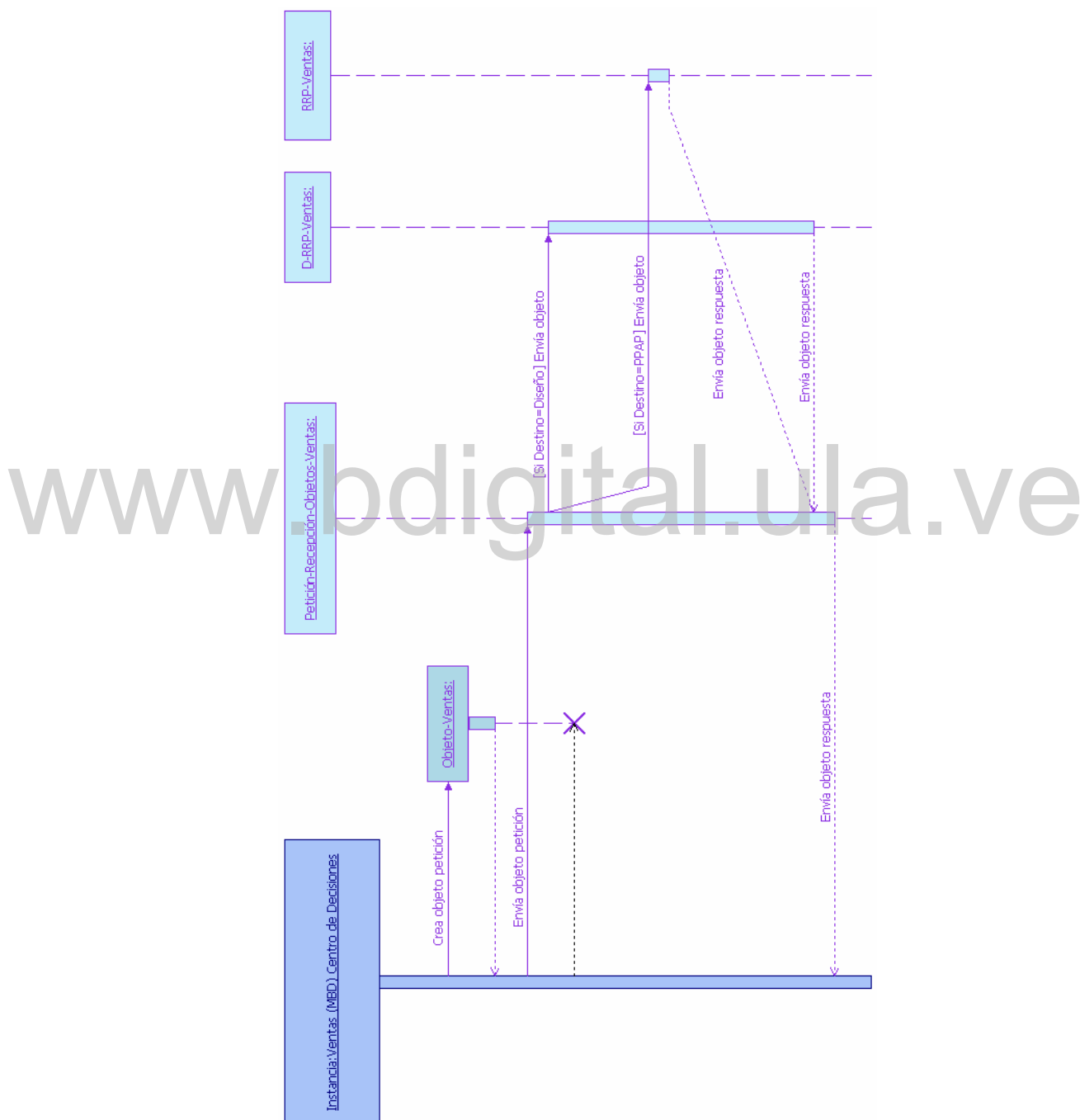


Figura 3.6. Diagrama Secuencial, Posibilidades de Petición de VENTAS



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

En la figura 3.7 se ilustra, a través de un diagrama secuencial, las posibilidades de peticiones que tiene PPAP. Puede hacer peticiones a Ventas, Diseño, Compras ó Producción).

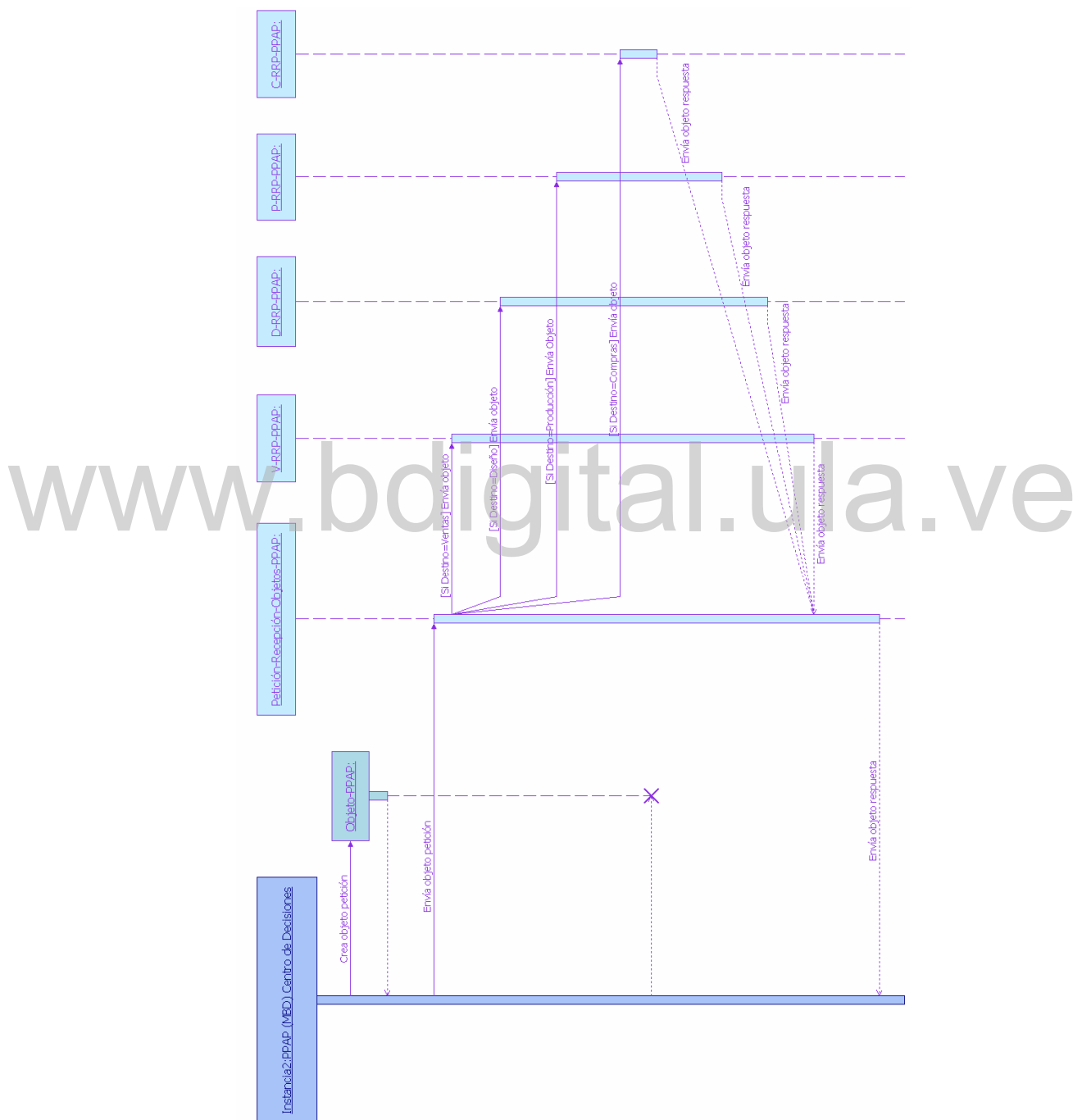


Figura 3.7. Diagrama Secuencial, Posibilidades de Peticiones de PPAP



En los diagramas presentados anteriormente, se puede observar que los protocolos de envío de objetos entre subsistemas, logran efectivamente llevar y traer los “objetos-petición” y “objetos-respuesta” enviados de un subsistema a otro, probando de esta manera su correcto funcionamiento. Ya sabiendo que los objetos que salen de un subsistema pueden llegar a otro, se describe a continuación la organización estructural que debe tener cada uno de los subsistemas propuesto en el modelo.

3.4.3. Diagramas Estructurales

Expresando la organización de las clases o módulos, interfaces de petición y recepción de objetos, interfaces de recepción y respuesta de peticiones, e interfaces de para el manejo de información de negocios, dentro de cada subsistema del modelo propuesto; se tiene un panorama de la complejidad que conlleva representar un sistema de producción holónico.

Ya tratados los conceptos básicos del modelado en UML, y probado el modelo desde el punto de vista secuencial (funcional), se muestran a continuación los diagramas estructurales, que exponen la constitución de cada subsistema explicado en el capítulo II:

- 1.) Subsistema Ventas, véase figura 3.8.
- 2.) Subsistema Diseño, véase figura 3.9
- 3.) Subsistema Compras, véase figura 3.10.
- 4.) Subsistema Producción, véase figura 3.11.
- 5.) Subsistema PPAP, véase figura 3.12.
- 6.) Todo el sistema, “Macro”, véase figura 3.13.



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

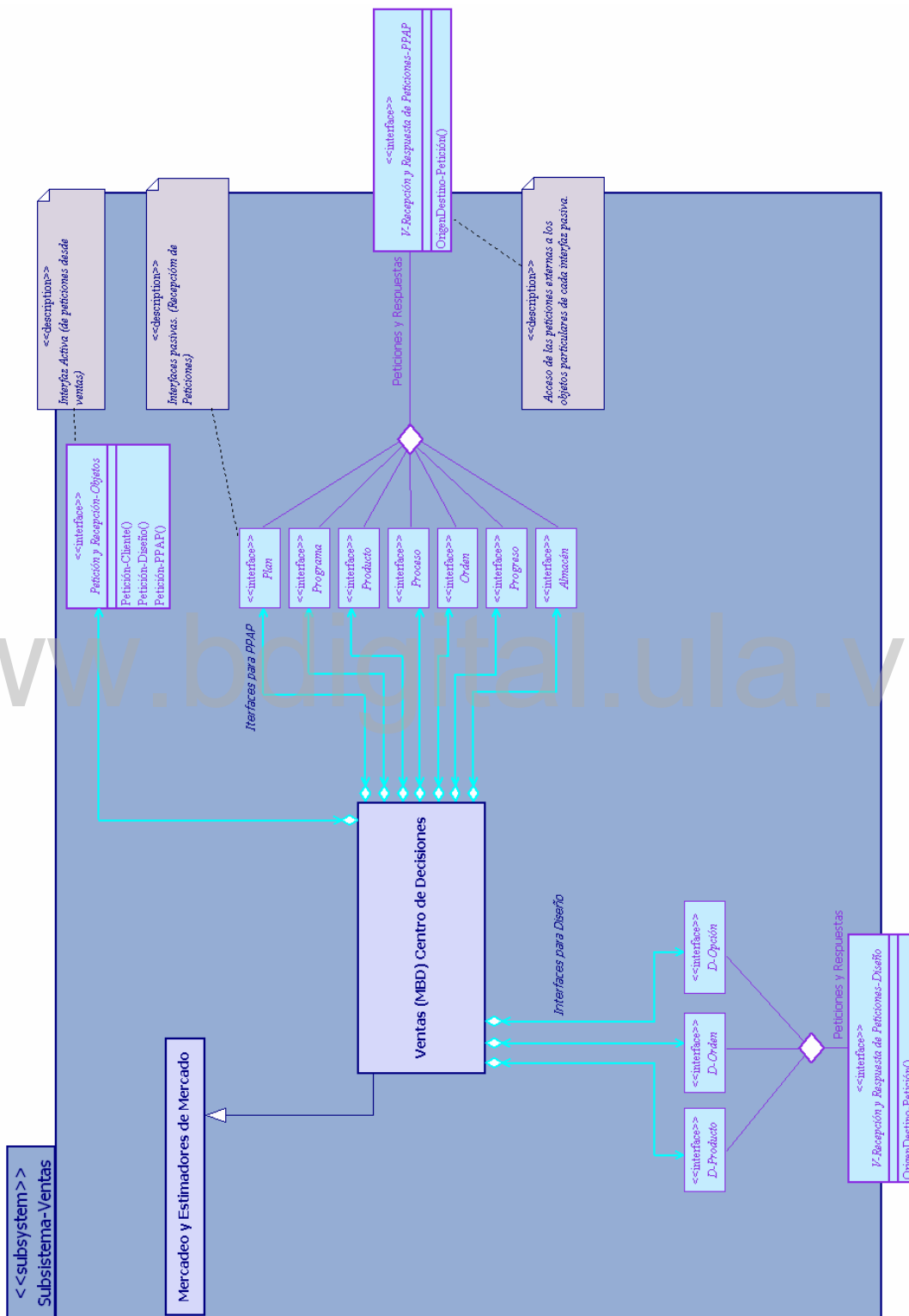


Figura 3.8. Diagrama Estructural VENTAS



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

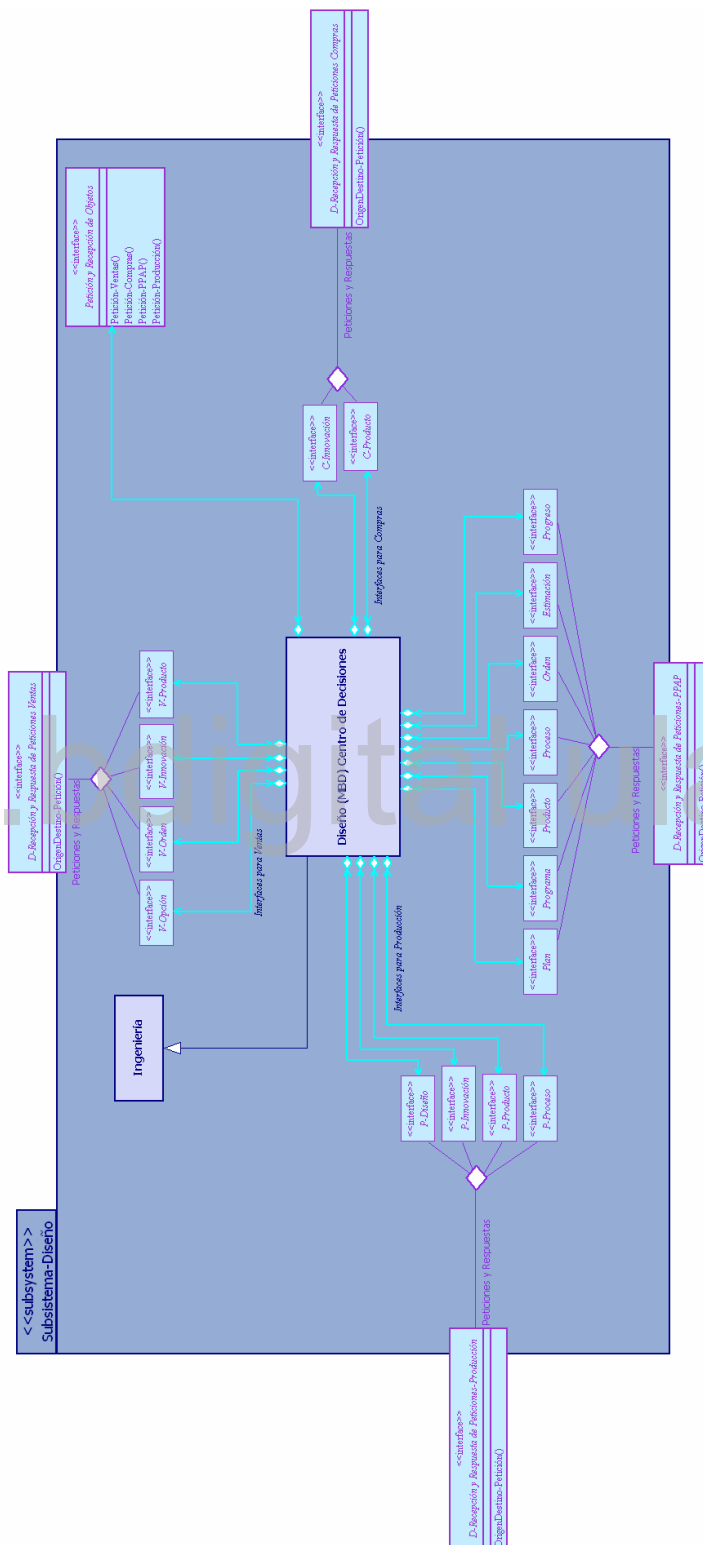


Figura 3.9. Diagrama Estructural DISEÑO



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

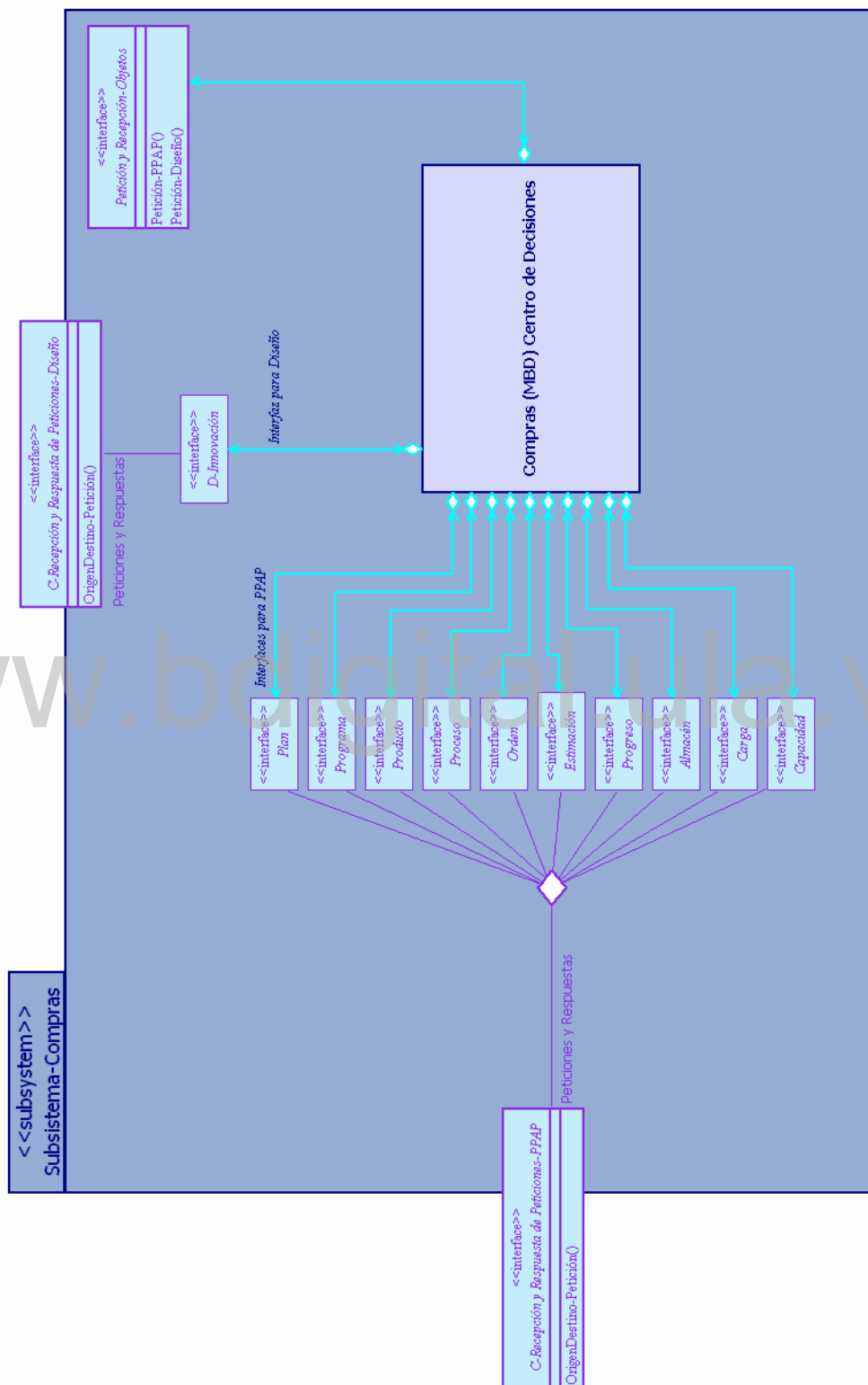


Figura 3.10. Diagrama Estructural COMPRAS



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

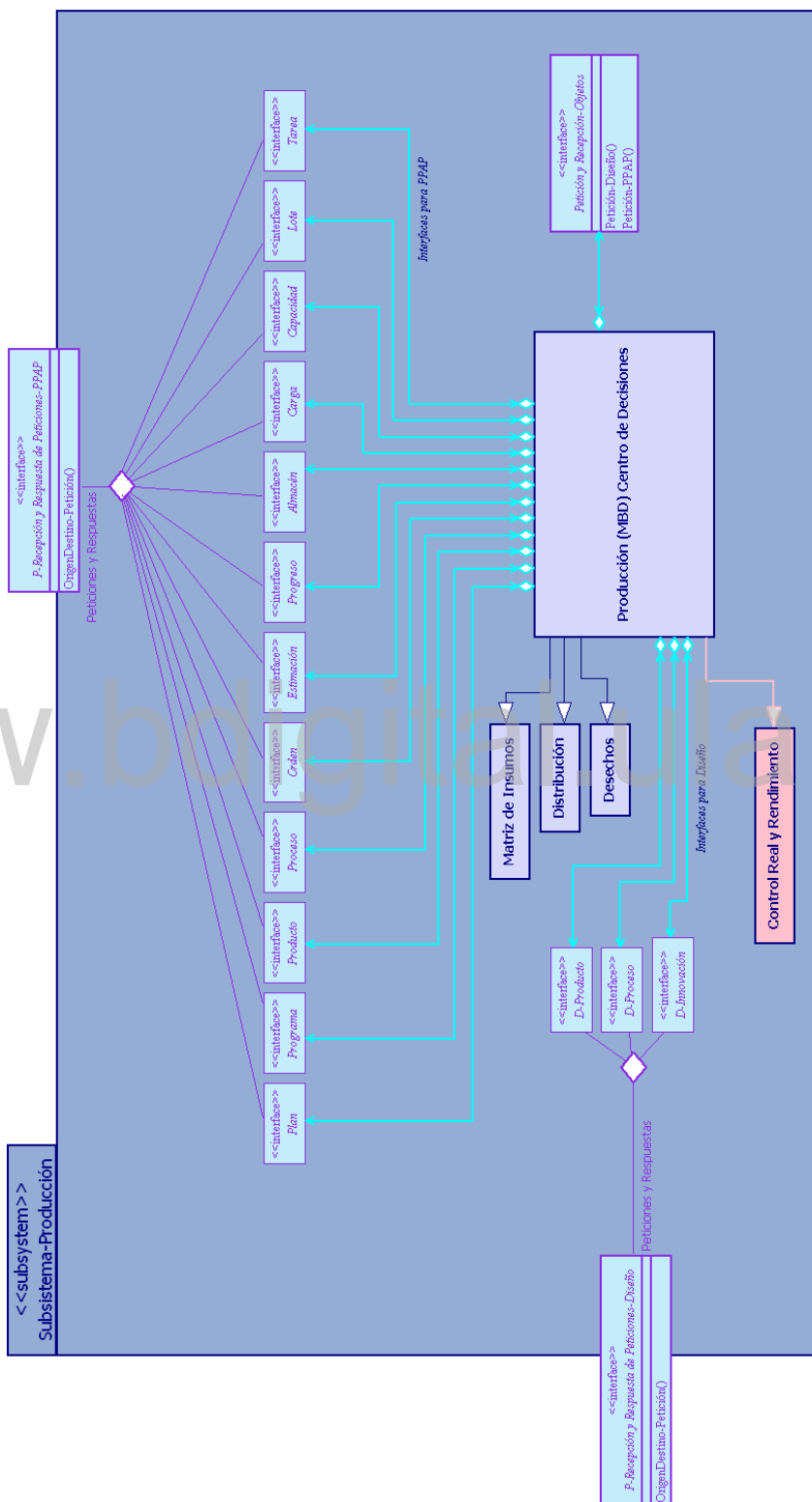


Figura 3.11. Diagrama Estructural PRODUCCIÓN



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

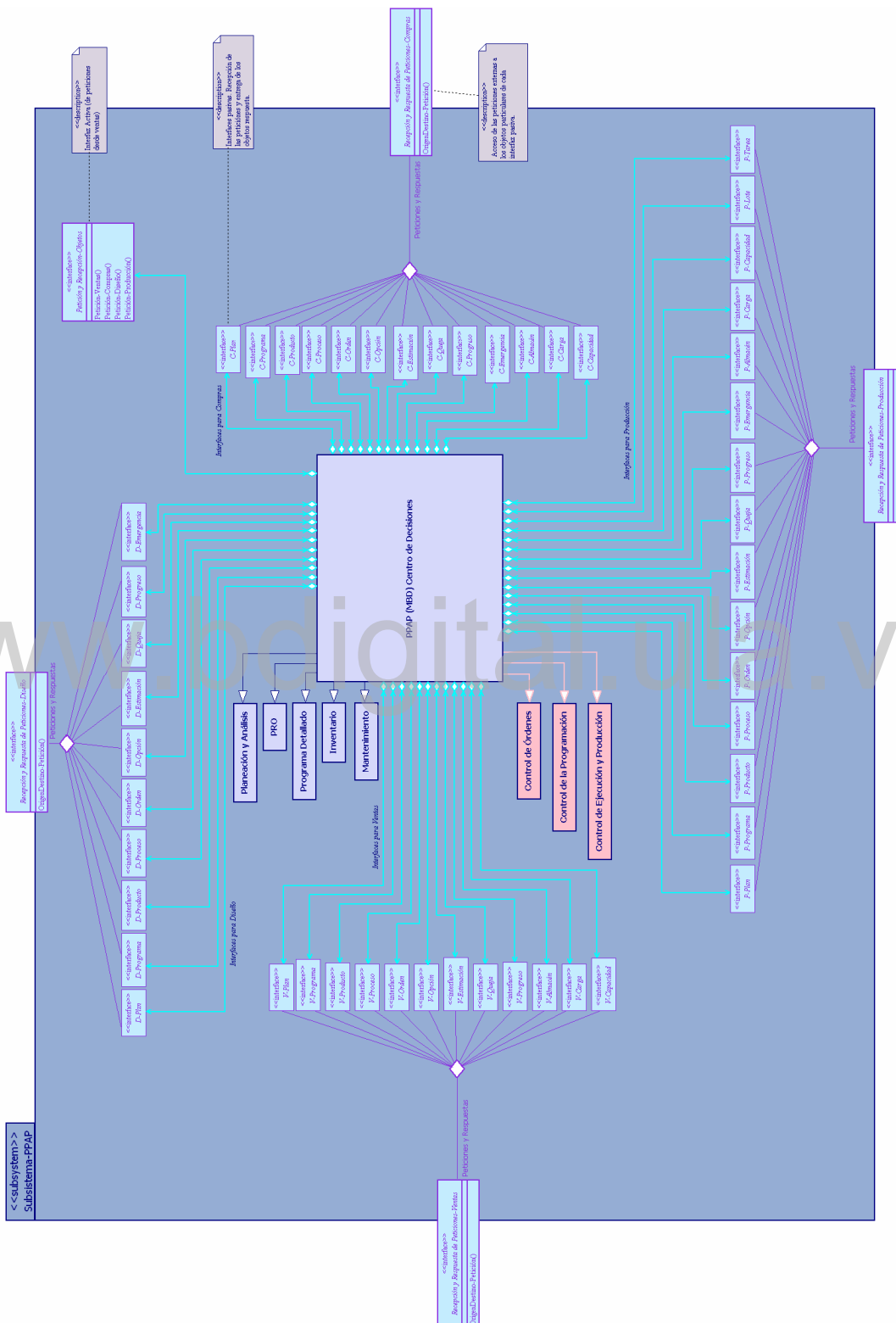


Figura 3.12. Diagrama Estructural PPAP

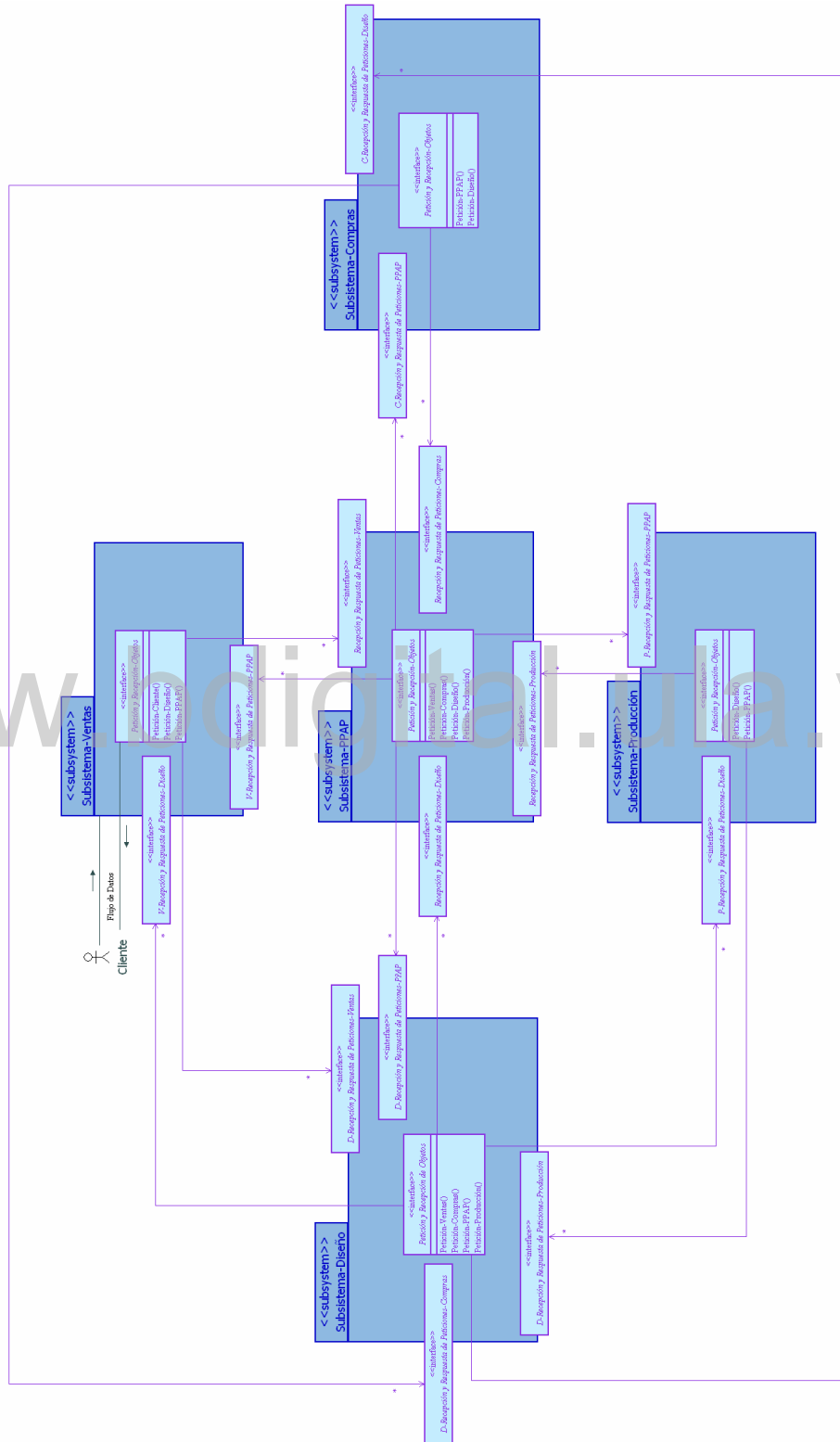


Figura 3.13. Diagrama Estructural “MACRO”



Como se observa en los diagramas estructurales del modelo en UML, a continuación se presentan interfaces que existen con el mismo nombre, para los diferentes componentes de negocios o subsistemas. Estas interfaces con el mismo nombre tienen básicamente la misma estructura, pero son implementadas ligeramente diferentes cuando los subsistemas objetivos son diferentes, es decir, su estructura se modificará según quién sea el que genere el mensaje, y hacia dónde vaya dicho mensaje.

3.5. Definición de Interfaces

A continuación se definirán las interfaces utilizadas en cada subsistema, que son en su mayoría propuestas por PSLX⁷. Se nombrarán los componentes de información que forman las interfaces, y las variaciones de sus funciones según quienes sean los orígenes y destinos, es decir, cuál de los protagonistas de la transmisión de mensajes inicia la comunicación, y quién responde. Es importante aclarar, que al hacerse referencia a un “cliente” o hacerse referencia a un “suministrador, surtidor o proveedor”, en los cuadros que se observan en el anexo D, se utilizan los subsistemas **Ventas** y **Compras** respectivamente, para llegar hasta ellos (clientes o proveedores).

Las interfaces utilizadas y sus componentes de información respectivos son:

- setPlan y getPlan (véase n.1, Anexo D-1)
 - Información de Plan.
- setPrograma y getPrograma (véase n.2, Anexo D-2)
 - Información de Programa-De-Envío.
 - Información de Programa-De-Producción.
 - Información de Instrucción-De-Liberación.
 - Información de Programa-De-Mantenimiento.
 - Información de Petición-De-Mantenimiento.

⁷ Las interfaces que no se encuentran especificadas por PSLX, son propuestas por el autor.



- Información de Demanda.
- setProducto y getProducto (véase n.3, Anexo D-3)
 - Información de Producto.
 - Información de la Matriz-De-Insumos de Manufactura (BOM).
 - Información de la Matriz-De-Insumos de Diseño (BOM).
 - Información de Diseño.
- setProceso y getProceso (véase n.4, Anexo D-4)
 - Información del Equipo-De-Producción.
 - Información del Método-De-Manufactura.
 - Información de Manufactura-Horas-Hombre.
 - Información de la Matriz-De-Insumos de Manufactura (BOM).
 - Información de la Matriz-De-Insumos de Diseño (BOM).
 - Información de Especificación-de-Proceso.
 - Información de Diseño.
- setOrden y getOrden (véase n.5, Anexo D-5)
 - Información de Orden.
 - Información No-Oficial.
 - Información de Opción-Datos.
 - Información de Tiempo-De-Entrega (Duetime).
 - Información de Notificación-Envío.
 - Información de Notificación-de-Recibo.
 - Información de Petición-De-Mantenimiento.
 - Información de Programa-De-Mantenimiento.
 - Información de Petición-Individual-De-Producto.



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

- setOpcion y getOpcion (véase n.6, Anexo D-6)
 - Información de Opción-Datos.
 - Información de Opción-De-Producto-Individual.
- setEstimacion y getEstimacion (véase n.7, Anexo D-7)
 - Información de Estimación.
 - Información de Plaza-de-Producción.
 - Información de Demanda.
- setQueja y getQueja (véase n.8, Anexo D-8)
 - Información de Queja.
- setProgreso y getProgreso (véase n.9, Anexo D-9)
 - Información de Calidad.
 - Información de Progreso-De-Producción.
 - Información de Progreso-De-Culminación.
 - Información del Progreso-De-Operación.
 - Información del Progreso-De-Inspección.
- setEmergencia y getEmergencia (véase n.10, Anexo D-10)
 - Información de Anormalidad-De-Instalación.
 - Información de Alarma-De-Calidad.
 - Información de Anormalidad-De-Progreso.
- setAlmacen y getAlmacen (véase n.11, Anexo D-11)
 - Información de Almacén (Stock).
- setCarga y getCarga (véase n.12, Anexo D-12)
 - Información de Progreso-De-Producción.
 - Información del Progreso-De-Operación.
- setCapacidad y getCapacidad (véase n.13, Anexo D-13)
 - Información de Estimación.
 - Información del Equipo-De-Producción.
 - Información de Manufactura-Horas-Hombre.
- setLote y getLote (véase n.14, Anexo D-14)



CAPÍTULO III DESCRIPCIÓN DE PROTOCOLOS

- Información de Lote.
- setTarea y getTarea (véase n.15, Anexo D-14)
 - Información de Tarea.
- setInnovacion y getInnovacion (véase n.16, Anexo D-15)
 - Información de Innovación.

www.bdigital.ula.ve

CAPÍTULO IV

DOCUMENTACIÓN DE PROTOCOLOS EN XML

www.bdigital.ula.ve

4.1. XML, Generalidades

En las siguientes tres secciones, se explicarán tres aspectos generales acerca del Lenguaje de Etiquetado Extensible (XML): “Qué es XML”, “beneficios del XML” y “lo que no es XML”.

4.1.1. ¿Qué es XML?

XML, (Extensible Markup Language) Lenguaje de Etiquetado Extensible, es un estándar respaldado por la W3C [Harold, 1999] para etiquetado de documentos. Define una sintaxis general, utilizada para enmarcar datos con etiquetas simples y legibles por humanos.

Provee un formato estándar para documentos, que es suficientemente flexible para ser personalizado para dominios como, diversos sitios web, intercambio de datos



electrónico (EDI), gráficos vectoriales, genealogías, listas de estado real, serialización de objetos, llamadas a procedimientos remotos (RPC), sistemas de correo de voz, y mas.

Se pueden escribir programas que interactúen y manipulen los datos encontrados en documentos XML; si se escriben, se debe acceder a un amplio rango de librerías libres de una variedad de lenguajes, que pueden leer y escribir XML, de tal manera que se puede hacer énfasis únicamente en las necesidades del programa.

También se puede usar software como “web browsers” (navegadores web) y editores de texto, para trabajar con documentos XML. Algunas herramientas están habilitadas para trabajar con cualquier documento XML, otras son personalizadas para soportar una aplicación XML particular, en un dominio en particular, pero la misma sintaxis es utilizada en todos los casos.

4.1.2. Los Beneficios del XML

XML es un lenguaje de “metaetiquetado” (metamarkup) [Elliott y W. Scott, 2004] para documentos de texto, en donde los datos son incluidos en documentos XML como líneas de texto y rodeados por etiquetas de texto que los describen.

La unidad básica de dato y etiqueta, es llamada “elemento”. La especificación XML define la sintaxis exacta que esta etiqueta debe seguir: cómo los elementos son delimitados por etiquetas, cómo se ve una etiqueta, qué nombres son aceptados para los elementos, dónde son colocados los atributos, entre otros. A primera vista, el etiquetado en un documento XML, se parece mucho al etiquetado en un documento HTML, pero hay diferencias cruciales.

Lo más importante es que XML es un metalenguaje, esto significa que “no” tiene un conjunto fijo de etiquetas y elementos que debieran trabajar para todo el mundo, en todas las áreas de interés, todo el tiempo. Cualquier intento en crear un conjunto finito de tales etiquetas está condenado al fracaso; en lugar de esto, XML permite a los desarrolladores y escritores, inventar los elementos que requieran en la medida que los vayan necesitando.



La “X” en XML significa “extensible”, es decir, que el lenguaje puede ser extendido y adaptado para cubrir muchas necesidades diferentes.

A pesar que el XML es bastante flexible en los elementos que permite, es bastante estricto en muchos otros aspectos, la especificación XML define la gramática para documentos XML, que dice: cuáles nombres de elementos son legales, cómo los atributos están atados a elementos, etcétera. Esta gramática es lo suficientemente específica para permitir el desarrollo de “XML parsers” [Elliott y W. Scott, 2004] que pueden leer cualquier documento XML. Los documentos que satisfacen esta gramática se dicen estar “bien formados”, por el contrario, los documentos que no están bien formados no están permitidos, así como a un programa en C no se le permite un error de sintaxis; por lo que los procesadores XML rechazan los documentos que contienen errores de “mal formación”.

Por razones de interoperabilidad, individuos u organizaciones pueden acordar usar sólo ciertas etiquetas; estos conjuntos de etiquetas son llamados aplicaciones XML. Una aplicación XML no es una aplicación de software que usa XML, tal como Mozilla o Microsoft Word; en lugar de ello, es una aplicación de XML en un dominio en particular, tal como gráficos vectoriales, cocina o como el caso en estudio: planificación y programación de la producción (producción planning and scheduling: pps).

En un documento XML bien formado, el etiquetado describe la semántica del documento, por ejemplo, puede indicar que un elemento es una fecha o una persona o un código de barras. En aplicaciones XML bien diseñadas, el etiquetado no dice nada acerca del formato del documento, esto significa, que no proporciona información acerca de si un elemento está presentado en **negritas**, en *cursivas* o sea un ítem lista. El XML es un lenguaje de etiquetado estructural y semántico, no un lenguaje de presentación.

Algunas aplicaciones XML, tales como XSL-FO (XSL Formatting Objects), están diseñadas para describir la presentación del texto, de cualquier manera, no se escribe un documento XSL-FO directamente, en cambio, se escribe un documento más semánticamente estructurado, y luego se usa una página de estilos (stylesheet) “XSL



Transformations” [Elliott y W. Scott, 2004] para cambiar un XML orientado a estructura en un XML orientado a presentación.

El etiquetado permitido en una aplicación XML en particular puede estar documentado en un esquema, e instancias particulares de un documento pueden ser comparadas al esquema. Los documentos que satisfacen el esquema se dicen ser válidos, mientras que los que no coinciden, son inválidos; esto significa que, para que un documento sea válido o no, depende de con cuál esquema se compare. No todos los documentos necesitan ser válidos, por muchas razones es suficiente con que el documento esté “bien formado”.

4.1.3. ¿Lo que no es XML?

Primero que todo, XML no es un lenguaje de programación. No existe tal cosa como un compilador que lea archivos XML y produzca código ejecutable. El XML puede ser utilizado como un formato para instrucciones que hacen que las cosas pasen, así como un programa tradicional puede leer un archivo de configuración de texto y tomar diferentes acciones dependiendo en lo que ve allí. Un documento XML aislado, simplemente “es”, no hace nada por si solo [Elliott y W. Scott, 2004].

Segundo, XML no es un protocolo de transporte de red, XML no envía datos a través de la red, más de lo que lo haría HTML. Los datos enviados a través de la red utilizando HTTP, FTP, NFS, o algún otro protocolo, pueden estar codificados en XML, pero debe haber un software fuera del documento XML que lo envíe.

Finalmente, XML no es una base de datos; no se va a reemplazar un servidor Oracle o MySQL, con XML. Una base de datos puede contener XML, pero la base de datos como tal, no es un documento XML; se puede almacenar información en una base de datos de un servidor o extraer información de una base de datos, en formato XML, pero para hacer esto se necesita estar corriendo un software escrito en un lenguaje de programación real como C o Java. Para almacenar XML en una base de datos, un software en el lado del cliente envía los datos en XML al servidor, utilizando un protocolo de red



establecido, como TCP/IP [Elliott y W. Scott, 2004]; el software en el lado del servidor recibirá los datos en XML, ejecuta un “parser” sobre ellos, y los almacena en la base de datos.

4.2. Reglas Generales para la Generación de Documentos en XML

En los cuadros mostrados en el Anexo D, se especifican las interfaces que se utilizan en este modelo con explicaciones simples de sus funciones principales. La aplicación que se desarrolle basada en lo especificado en dichos cuadros, debe implementar todas las funciones de estas interfaces, o en su defecto, retornar un mensaje de error diciendo que una interfaz no implementada está siendo llamada. En las secciones próximas se estudiarán los mensajes de error con más detalle.

Es importante destacar, que todos los elementos y respectivos atributos que aquí se utilizan¹, son extraídos de la especificación XML desarrollada por el consorcio PSLX [PSLX-04, 2003].

Existe un esquema (en vías de estandarización) desarrollado por el consorcio PSLX y el grupo OASIS², en donde se definen muchos elementos y atributos con fines de planificación y programación de la producción, y en donde además se hace referencia a otros esquemas, cruciales en la definición de elementos y atributos; pero no está completo. Este esquema estuvo disponible para su adquisición por muy poco tiempo, por los momentos no lo está. Muchas de las especificaciones a las que es difícil su acceso a través

¹ Para ejemplificar los documentos XML que viajan como objetos desde un centro de decisiones a otro, utilizando los protocolos desarrollados en el capítulo III

² PPS Part1 Committee Draft (Version1). Esta parte especifica elementos del núcleo XML, para componer documentos XML respecto a planificación y programación en industrias manufactureras. Este borrador fue aprobado el 14 de Agosto del 2005. Información disponible en: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pps



de las páginas Web indicadas en las referencias, pueden accederse a través del documento de especificaciones XML de PSLX [PSLX-04, 2003].

4.2.1. Sistema de Intercambio de Información en XML

Las interfaces utilizadas en este modelo consisten en un prefijo “set” o “get”, y a continuación una línea de caracteres junto a dichos prefijos. Los prefijos set y get, indican la acción realizada por la interfaz, y la línea de caracteres que le sigue indica el objetivo para la acción.

Un “mensaje de petición”, un “mensaje de respuesta” (puede ser aceptación o rechazo), un “mensaje de recibido” y un “mensaje de excepción” son definidos para cada una de estas interfaces.

El “mensaje de petición” para notificar al servidor de la petición de un cliente tiene la siguiente forma general [PSLX-04, 2003]:

```
<pslx type="request" id="ID del mensaje" action="nombre de la interfaz">  
.....  
información de petición  
.....  
</pslx>
```

El servidor ejecuta el proceso requerido para este “mensaje de petición” y retorna el resultado al cliente con un mensaje de respuesta. La forma general del mensaje de respuesta es [PSLX-04, 2003]:

```
<pslx type="response" id="ID del mensaje" ref="ID del mensaje de petición" action="nombre de la interfaz ">  
.....  
información de respuesta  
.....  
</pslx>
```

El servidor puede enviar un mensaje de excepción además de un mensaje de respuesta.

El servidor y el cliente deben retornar un mensaje de recibo (recibido), cuando una confirmación de “recibido” es configurada en un mensaje de petición o respuesta.



4.2.2. Ambiente Cliente-Servidor

En este inciso se explican detalladamente los mensajes que pueden enviarse entre el cliente y el servidor. El cliente en el modelo en estudio, siempre será el subsistema que inicie la comunicación (iniciador o starter), mientras que el servidor será siempre el subsistema que responda a dicha petición, no importa de cuál de los cinco subsistemas se trate, para todos es igual. Para la especificación de todos los casos que se muestran a continuación, se utilizó [PSLX-04, 2003].

4.2.2.1. Mensaje de Petición (Request Message) en el Lado del Cliente

La interfaz con un prefijo “set”, debe enviar información de negocios (generada en un centro de decisiones “cliente”), hacia el lado del centro de decisiones al que se inquiere: un centro de decisiones objetivo (servidor). La interfaz con un prefijo “get”, debe recibir u obtener información de negocios en lado del centro de decisiones al que se hace la solicitud (el centro de decisiones objetivo, servidor).

La información de negocios enviada con un “mensaje de petición” ó “request message”, es la información sobre elementos básicos, cliente, proveedor, ítem, recurso, lote, tarea, operación y orden, el parámetro de información en el plan y la información de expresión de cálculos. [PSLX-04, 2003]

Tomando en cuenta que todos los elementos y atributos utilizados para definir documentos referentes a planificación y programación de la producción, están referidos a los esquemas indicados desarrollados por OASIS y PSLX³; todos los elementos básicos y los elementos relacionados con el “plan” realizando un “mensaje de petición”, deben ser fijados en el nivel de etiquetas justo debajo (anidado o endentado) de la etiqueta <pslx>. Utilizar el “ID” (identificador) debe permitir a todos los datos individuales, ser identificados en la información de elemento básico e información de “plan”.

³ pps-schema-core-draft-1[1].1 Disponible en: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pps



El valor del atributo “type” en una etiqueta <pslx> debe ser fijado como “request” (petición) para mostrar un “mensaje de petición”. La cadena de caracteres que es única en la aplicación solicitada, debe ser configurada en el atributo “id” (identificador) como el código para identificar el mensaje. En el atributo “action” (acción) debe ser fijado el nombre de la interfaz.

Una “petición de recibido” o “receipt request” (que no es más que un acuse de recibo), puede ser agregada a un “mensaje de petición” para confirmar que la aplicación solicitada recibe dicho mensaje. Configurando el atributo “recibido” en “true” en la etiqueta <pslx>, puede cambiar un mensaje de petición al mensaje de petición con confirmación de recibo. De cualquier manera, en caso de comunicación síncrona, la confirmación de recibo no puede ser solicitada. Por ejemplo:

```
<pslx type="request" id="ID00001" action="getOrden" receipt="true">
```

```
.....  
información de petición  
.....
```

```
</pslx>
```

Agregar y corregir datos de elementos. La interfaz con el prefijo “set” solicita a un servidor agregar, corregir o borrar datos de elementos. Cuando se agregan los datos (data) del elemento, “create” (crear) debe ser fijado en el atributo “ac” en la etiqueta del elemento básico agregado o elemento del plan. Cuando se corrige o se borra la data, “revise” (revisar) y “delete” (borrar) deben ser fijados como los atributos de “ac” respectivamente. Ejemplo:

```
<pslx type="request" id="ID000011" action="setAlmacen">  
  <item name="teclado" ac="create">  
    <stock timeType="disc" relative="true">  
      <qty value="8000"/>  
    </stock>  
    <lotsize type="max" value="50"/>  
  </item>  
  <item name="mouse" ac="revise">  
    <stock>  
      <qty value="28000"/>  
    </stock>  
  </item>  
  <item name="memoria" ac="delete" type="material"/>  
</pslx>
```



CAPÍTULO IV DOCUMENTACIÓN DE PROTOCOLOS

El servidor que agrega, corrige y borra la data de los elementos, debe retornar solamente el ID (identificador) de la data de los elementos que pueden ser procesados normalmente como un mensaje de “respuesta” luego del procesamiento. Si el procesamiento requerido falla en una parte de la data de los elementos, la información de error –incluyendo el ID de la data del elemento fallido- es fijada en una etiqueta <error> y debe ser retornada al solicitante. Ejemplo:

```
<pslx type="response" id="RID000011" ref="ID000011" action="setAlmacen">  
  <error code="E02484" ID="memoria" severity="warning">el valor del atributo es ignorado</error>  
  <item name="teclado"/>  
  <item name="mouse"/>  
</pslx>
```

Cuando todas las solicitudes fallan, un “mensaje de excepción” debe ser retornado. Por otro lado, si solo una parte de las peticiones tiene éxito, un mensaje de respuesta debe ser devuelto; en otras palabras, la información de error puede ser especificada solo en un mensaje de excepción, y la información de advertencia (warning) puede ser especificada en un mensaje de excepción, respuesta y recibido.

Por ejemplo:

```
<pslx type="exception" ref="ID000011">  
  <error code="E02004" severity="error">El ID aplicable no puede ser encontrado</error>  
  <error code="E02015" severity="error">error en la especificacion de valor</error>  
  <error code="W08002" severity="warning">el valor del atributo es ignorado</error>  
</pslx>
```

Cuando un “mensaje de petición” es la adición de data de elementos (create), y los objetos adicionados existen en el servidor que recibe el mensaje, la información de advertencia debe ser fijada en una etiqueta <error> y debe ser retornada como una “mensaje de respuesta” como resultado del procesamiento. En tal caso, toda la data existente es borrada una vez y luego la nueva data fijada es agregada.

Cuando un “mensaje de petición” es la corrección de data de elementos, y no existen elementos a corregir en el lado del servidor, la data debe ser agregada como nueva data. Posteriormente la información de advertencia debe ser fijada en una etiqueta de



<error> en un “mensaje de respuesta” que debe ser retornado.

Cuando un mensaje de petición es la eliminación o borrado de la data de elementos y no existen los elementos a borrar en el servidor, la información de error debe ser fijada con una etiqueta de <error> y ser enviada al solicitante. Si todas las solicitudes fallan, un “mensaje de excepción” debe ser retornado. Por otra parte, si solo una parte de las peticiones tiene éxito, un “mensaje de respuesta” debe ser devuelto.

Cuando el servidor no tiene autoridad para agregar, corregir y borrar data de elementos, el mensaje de error es fijado con una etiqueta de <error> y luego un mensaje de excepción debe ser retornado.

Un “mensaje de petición” puede ejecutar solo una instrucción de adición, corrección y eliminación.

Cuando se corrige o borra un elemento básico, la data de elementos a ser corregido o borrado, no puede ser corregida o borrada en un atributo con nombre “query”.

Cómo corregir sub-elementos. Todos los sub-elementos y atributos de la data de elementos objetivo existentes en el servidor, son sobrescritos para su corrección. Sin embargo, los sub-elementos y los atributos que no están fijados como contenido a corregir, no son sobrescritos y entonces la información previa permanece como estaba. Si el valor no está asignado o fijado en el atributo, y además el valor omitido está definido, el valor omitido es fijado como nuevo.

Algunos elementos básicos de la data de elementos objetivo, pueden tener data múltiple como sub-elementos, de acuerdo al tipo de elemento básico [PSLX-04, 2003]. En tal caso, si el contenido de los sub-elementos recién especificados reemplaza el contenido de los sub-elementos existentes, o si el contenido es agregado como nuevo dejando el contenido de los sub-elementos existentes como estaba, depende de las circunstancias. Ejemplo: Para series de tiempo. Supongamos que el almacén para las fechas 1, 3 y 5 de noviembre está dispuesto en un servidor de la siguiente manera:



```
<pslx>
  <item name="producto-A">
    <stock><time value="2005-11-01T00:00:00"/><qty value="200"/></stock>
    <stock><time value="2003-11-03T00:00:00"/><qty value="260"/></stock>
    <stock><time value="2003-11-05T00:00:00"/><qty value="140"/></stock>
  </item>
</pslx>
```

Luego, desde el lado de un cliente es posible solicitar la especificación del almacén para el 2 de noviembre como sigue:

```
<pslx type="request" id="A001" action="setAlmacen">
  <item name="producto-A" ac="revise">
    <stock><time value="2003-11-02T00:00:00"/><qty value="290"/></stock>
  </item>
</pslx>
```

En este caso, se borra la data que exista especificada desde el 2 de noviembre en adelante, y el resultado se observa así:

```
<pslx>
  <item name="producto-A">
    <stock><time value="2003-11-01T00:00:00"/><qty value="200"/></stock>
    <stock><time value="2003-11-02T00:00:00"/><qty value="290"/></stock>
  </item>
</pslx>
```

Cuando se desea borrar toda la información de series de tiempo se especifica `<time ref="init"/>` y se omite la etiqueta `<qty>`, por ejemplo:

```
<pslx type="request" id="B001" action="setAlmacen">
  <item name="producto-B" ac="revise">
    <stock><time ref="init"/></stock>
  </item>
</pslx>
```

La solicitud para borrar un subelemento, no un elemento, puede ser especificada a través de la definición de un nuevo subelemento. En tal caso la eliminación del subelemento debe ser descrita antes del nuevo subelemento. Por ejemplo:

Todos los ítems que ya han sido especificados como ítems de producción de la operación-A son borrados y luego el ítem-X es especificado como nuevo:



```
<pslx type="request" id="P123" action="setProceso">  
  <operation name="operación-A" ac="revise">  
    <produce item="query"/>  
    <produce item="item-X"/>  
  </operation>  
</pslx>
```

En los demás casos que no se encuentran especificados, se definen los subelementos nuevos, y se sustituyen por los que ya están presentes.

4.2.2.2. *Reacción a Solicitudes (Request) en el Lado del Servidor*

El servidor que provee el servicio, debe responder a todos los “mensajes de petición” en la forma especificada a través de PSLX, para este modelo. Sin embargo, cuando el servicio específico no está implementado⁴, el mensaje acerca de la no implementación puede ser retornado como un error. Cuando se recibe un “mensaje de petición” que no está definido como en el inciso anterior, un mensaje de error debe ser retornado.

El servidor que recibe el “mensaje de petición” de la interfaz implementada, debe retornar un “mensaje de respuesta” o un “mensaje de excepción” luego del procesamiento de la petición.

Cuando una “petición de recibido” es agregada al “mensaje de petición”, un “mensaje de recibido” debe ser enviado antes de responder al “mensaje de petición”. Sin embargo, cuando un error ocurre antes de que se produzca el “mensaje de recibido”, sólo el “mensaje de excepción” es retornado y por lo tanto el “mensaje de recepción” puede ser omitido.

Mensaje de respuesta (respond message). Un “mensaje de respuesta” es utilizado para retornar el resultado de procesar un “mensaje de petición”, como respuesta. Cuando la solicitud o petición no puede ser satisfecha porque un usuario no tiene la autoridad, un “mensaje de excepción” debe ser retornado.

⁴ Ej. Cuando el mensaje de petición llama a una interfaz de un subsistema servidor, que no está implementada (que no existe para el momento)



El valor del atributo “type” (tipo) en una etiqueta <pslx>, debe ser fijado como “response” (respuesta) en un “mensaje de respuesta”. La cadena de caracteres única en la aplicación que responde, debe ser configurada en el atributo “id”. Por otra parte, el mismo valor del atributo “id” configurado o fijado en un “mensaje de petición”, debe ser fijado en el “ref attribute” (atributo de referencia) de dicho mensaje respuesta. Finalmente, el mismo nombre de la interfaz adecuada para el atributo “action” del “mensaje de petición”, debe ser fijado en el atributo “action” de mensaje en cuestión.

Una “petición de recibido” (o acuse de recibo) puede ser agregado a un “mensaje de respuesta” para confirmar que el mensaje llega al destinatario. Cuando el atributo “receipt” (recibido) en una etiqueta <pslx> es fijado como “true” (verdadero), la confirmación de recibo o “petición de recibido” es agregada al “mensaje de respuesta”; pero en caso de comunicación síncrona, la “petición de recibido” no puede ser solicitada.

El contenido de un error y una advertencia puede ser fijado en un “mensaje de respuesta” con una etiqueta <error> cuando la ocasión lo requiera. Cuando se incluye un error, el error es parte de un “mensaje de respuesta”, y cuando todo el procesamiento es un error, el error debe ser retornado con un “mensaje de excepción”. Ejemplo:

```
<pslx type="request" id="PRO001" action="setProgreso"/>
```

```
<pslx type="response" id="R001" ref="PRO001" action="setProgreso"/>
```

Mensaje de recibido (receipt message). Cuando un “mensaje de petición” o un “mensaje de respuesta” incluye una “petición de recibido”, el receptor del mensaje debe retornar un “mensaje de recibido” inmediatamente. “receipt” debe ser fijado en el atributo (type) de una etiqueta <pslx>, en el “mensaje de recibido”.

La cadena de caracteres única en el lado que envía, debe ser fijada en el atributo “id”. El valor del atributo “id” del mensaje que solicitó el acuse de recibo, debe ser especificado en el atributo “ref” (referencia); de la misma manera, el valor del atributo “action” del mensaje recibido, que es un nombre de interfaz, debe ser fijado en el atributo “action” del mensaje en cuestión. Ejemplo:



```
<pslx type="request" id="aaa" receipt="true" action="setPrograma"/>
```

```
<pslx type="receipt" id="bbb" ref="aaa" action="setPrograma"/>
```

Mensaje de excepción (exception message). Un “mensaje de excepción” debe ser enviado cuando se rechaza el mensaje recibido, cuando el servicio para el mensaje no está implementado, y cuando el contenido del mensaje contiene un error. En el “mensaje de excepción”, “exception” debe ser fijado en el atributo “type” en una etiqueta <pslx>. La cadena de caracteres única en el lado que envía debe ser fijada en el atributo “id”. El mismo valor “id” del mensaje recibido, debe ser especificado en el atributo “ref” del mensaje en cuestión. El valor del atributo “action” del mensaje recibido, que es un nombre de interfaz, debe ser fijado en el atributo “action”.

Los contenidos de excepción en un “mensaje de excepción” son configurados con una etiqueta <error>. La excepción es un error y puede contener además, advertencias. El valor del atributo “severity” (gravedad o severidad) debe ser “error” para los errores y “warning” para las advertencias. La información de error o advertencia múltiple puede ser configurada. Finalmente el “mensaje de excepción” debe tener al menos un error. Ejemplo:

```
<pslx type="exception" id="bbb" ref="aaa">  
  <error code="Z01006" severity="error">no existe el servicio aplicable setPrograma</error>  
</pslx>
```

Data permanente (permanent data). La data expresada puede hacerse permanente como un archivo, esto se logra de dos posibles maneras: omitiendo el valor del atributo “type” (tipo) o especificándolo como “file” (archivo) en una etiqueta <pslx>.

4.2.2.3. Comunicación Síncrona y Asíncrona

Sincronización. Las propiedades de sincronía se definen no sólo sobre los protocolos de comunicación que usa un sistema, sino también sobre las características de los procesos que trabajan en él. En sentido estricto, se dice que un sistema es síncrono si [Vassos y Toueg, 1993]:



CAPÍTULO IV DOCUMENTACIÓN DE PROTOCOLOS

- El retardo de transmisión de los mensajes (entre emisor y receptor) está acotado.
- Cada proceso dispone de un reloj cuya deriva respecto al tiempo global de referencia está acotada.
- Existe un límite superior y otro inferior en el tiempo que tarda en ejecutarse una instrucción en cada proceso.

Los sistemas *síncronos* son muy útiles porque permiten usar plazos para detectar fallos. Como en estos sistemas el retardo de un mensaje, en ausencia de fallos, está acotado, basta con que se detecte un retraso para que el receptor suponga que algo ha ido mal en el emisor o en el sistema de comunicaciones [González, 1998]. Además, en estos sistemas, incluso en presencia de fallos, los relojes de los procesos pueden mantenerse sincronizados, de forma que la deriva entre dos de ellos cualesquiera esté acotada [Lamport y Melliar-Smith, 1985]. De estos relojes se dice que están *sincronizados aproximadamente*, y son muy útiles para diseñar protocolos de comunicaciones. Para ciertas aplicaciones esta propiedad es imprescindible (por ejemplo, sistemas de control con requisitos de tiempo real).

Por otra parte, en un sistema *asíncrono* no se puede hacer ninguna suposición sobre el comportamiento temporal del sistema (retardos de mensajes, deriva de relojes, etc.). Este modelo es atractivo por su sencillez, y porque las soluciones que se encuentren para él serán válidas en prácticamente cualquier sistema. Además, en el mundo real, raramente es posible encontrar un sistema que se comporte siempre como sistema síncrono. Es habitual que, por el contrario, cargas inesperadas o puntuales conviertan un sistema en asíncrono, al menos en esos momentos. Sin embargo, las soluciones para sistemas asíncronos son también, habitualmente, más complejas, y a veces incluso no existen [González, 1998].

Este modelo soporta ambos métodos de comunicación, síncrona y asíncrona. Particularmente, existen tres tipos de comunicación asíncrona, tipo Push-Push (empujar-empujar), Push-Pull (empujar-halar) y Pull-Push (halar-empujar) [PSLX-04, 2003]. Cada subsistema debe declarar el método de comunicación aplicable, de manera tal que la



aplicación que se implemente en cada uno, conozca a través de qué método de comunicación específico se envían los mensajes.

Comunicación síncrona en PSLX. Cuando se ejecuta una comunicación síncrona, un “mensaje de petición” debe corresponder a una petición o solicitud, y un “mensaje de respuesta” ó “mensaje de excepción” debe corresponder a una respuesta. En este método de comunicación no se puede agregar un “mensaje de recibido”, por lo tanto, un mensaje de este tipo no está especificado en ninguna petición o respuesta. Si hay un error en un “mensaje de respuesta”, el error no puede ser notificado al servidor, y viceversa, si hay un error en un “mensaje de petición”, este no puede ser notificado al cliente.

Comunicación asíncrona en PSLX. La comunicación asíncrona que se va a utilizar en este modelo es de tipo “empujar” (Push-Pull) según [PSLX-04, 2003].

El tipo asíncrono “empujar” (Push-Pull) es el método de comunicación en donde un cliente es siempre un emisor (sender); por lo tanto un “mensaje de petición” es enviado en “Push” (como una petición o request) y un “mensaje de respuesta” es retornado en “Pull” (como una respuesta o response).

El método de comunicación asíncrona depende de quién envía (un mensaje de petición, mensaje de respuesta, etcétera), si un cliente ó un servidor; se ha decidido utilizar este método porque, como se observó en los modelos de UML en el capítulo III, cada subsistema posee interfaces para solicitar servicios de otros subsistemas, y también posee interfaces a través de las cuales otros pueden solicitar servicios, es decir, que los subsistemas pueden demandarse servicios entre sí. Entonces, quién es el cliente o el servidor, sólo depende del subsistema que “inicie” la comunicación (ver tablas del capítulo IV).



4.2.2.4 Comunicación Síncrona o Asíncrona Adaptada al Modelo Propuesto

En general, las diferencias principales entre el comportamiento de las interfaces de los subsistemas, una vez que se hayan puesto de acuerdo en transmitir según un método de comunicación u otro, son las siguientes:

- **Síncrona:** El subsistema que envía, debería esperar respuesta del subsistema objetivo luego de haberle enviado el mensaje, entre tanto, las interfaces de éste, no pueden hacer otro trabajo hasta que la respuesta llegue. A su vez, las interfaces receptoras en el subsistema objetivo, están siempre esperando por un mensaje y responden inmediatamente.
- **Asíncrona:** Las interfaces del subsistema que envía, pueden hacer otros trabajos luego de haber enviado el mensaje. Cuando la respuesta llegue, las interfaces detienen el trabajo actual para lidiar con la respuesta. Por su parte, las interfaces receptoras en el subsistema objetivo responden el mensaje luego de que sus trabajos actuales sean completados. Por lo tanto, se puede esperar una respuesta por mucho tiempo, dependiendo del nivel de carga que exista en el centro de decisiones del subsistema objetivo que genera la respuesta.

4.3. Generación de Documentos en XML

Cada uno de los patrones individuales que se observan en el método síncrono y asíncrono de tipo “empujar” (Push-Pull), corresponden al tipo de comunicación que pueden utilizar cada uno de los protocolos desarrollado en el capítulo III.

Para representar los métodos de comunicación desarrollados para los protocolos, en XML, se ejemplifican pequeños casos que se verán en secciones más adelante. Aquí se ilustran los diferentes elementos y atributos, que deben llevar los encabezados del



etiquetado de los objetos, para poder enviarse correctamente los mensajes entre los diversos subsistemas.

4.3.1. Esquema XML Estándar para la PPAP

Esta sección describe la “Especificación de Intercambio de Datos en XML”, para una futura implementación de cada interfaz⁵ definida en el modelo, en sistemas de aplicaciones individuales.

Ubicación de la especificación de intercambio de datos. Cuando dos programas de aplicaciones ejecutan comunicación de datos, la comunicación es manejada siendo dividida en algunas capas como muestra la figura 4.1. Es importante ubicar el esquema que se trabajará, en el ambiente de “capas para la comunicación de mensajes”. El rango de la especificación de intercambio para los sistemas distribuidos en donde se aplique dicha especificación, debe estar acotado por ciertas condiciones que se deben asumir, existen en el sistema. De tal manera, TCP/IP debe ser usado para la capa de red, HTML para la capa de transporte y SOAP para la capa de middleware. La forma de implementación en la capa de middleware e inferiores (véase figura 4.1), básicamente se separa de la especificación de intercambio de datos, definida en este capítulo, por lo que no depende ella [PSLX-04, 2003].

El esquema de datos referido en un sistema de computadora, necesita tener una visión “específica de aplicación” [PSLX White Paper]. En general, cada sistema de información tiene su propio esquema de implementación para representar data de negocios, en la forma de cada sistema de aplicación (ver figura 4.2). Para lograr la interoperabilidad de la data, se necesita que exista un tipo de modelo de datos, que comparta el formato común de los esquemas de implementación específica, es decir, un “esquema común en XML”. Este esquema común, representa información de negocios en el modelo de datos abstracto que se utilice para el modelo propuesto.

⁵ En un posible proyecto posterior.

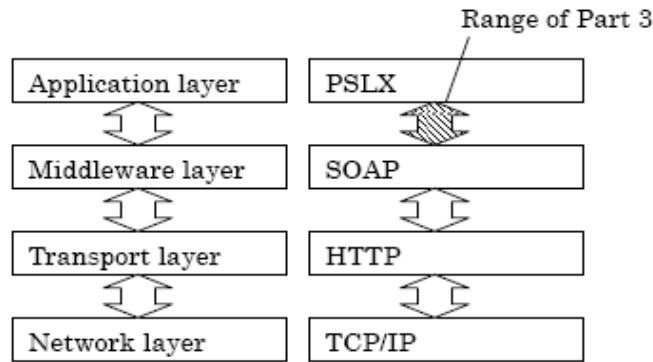


Figura 4.1. Capas para la Comunicación de Mensajes. Tomado de “XML Specification, PSLX-04” por PSLX Engineering Specification, PSLX Consortium. 2003.

PSLX define dos esquemas comunes para implementación: PSLX-RDB common schema y PSLX-XML common schema, naturalmente basados en la ontología PSLX. Para fines del modelo en estudio, se desarrolla el esquema común PSLX-XML.

www.bdigital.ula.ve

ISO Schema and meta schema PSLX Planning & Scheduling

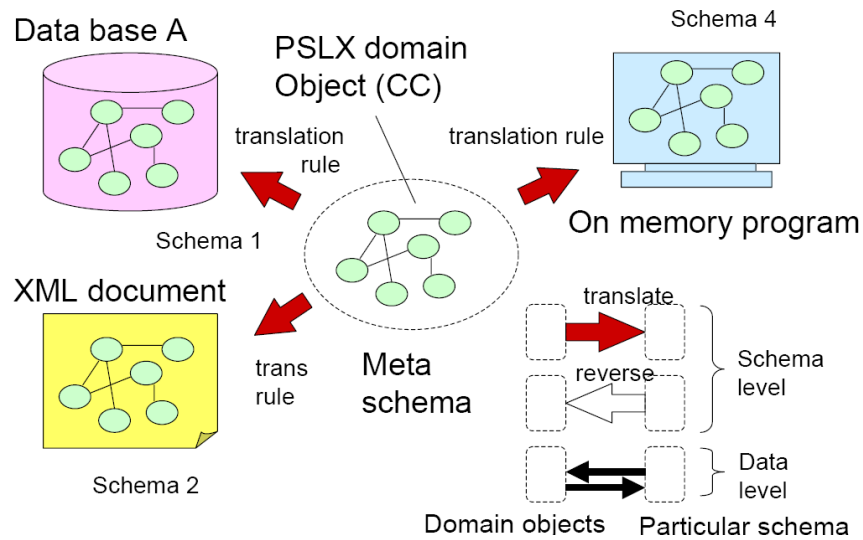


Figura 4.2. Esquema y Meta Esquema. Tomado de “PSLX-03, PSLX Domain Objects” por PSLX consortium. 2003.



4.3.1.1. Esquema Común PSLX-XML (PSLX-XML Common Schema)

Un esquema común PSLX-XML es definido para el intercambio de data entre diferentes aplicaciones, en diferentes actividades de negocios, por medio de la “comunicación de tipo mensajería” (messaging type communication) [PSLX White Paper]. Este tipo de comunicación se está volviendo popular aún en ambientes computacionales diferentes; por ejemplo, en las tecnologías de servicios-Web para servidores de aplicaciones Web y clientes en la Internet; en XML/EDI para el intercambio de información inter-empresa utilizando un formato de datos flexible, en vez del formato de datos de tamaño fijo del EDI convencional.

Las clases definidas en el esquema común PSLX-XML tienen estructuras jerárquicas, porque básicamente XML es un lenguaje de especificación de data para documentos. En el esquema, relaciones “tipo red” (network type) son implícitamente definidas para cada ítem de datos y sus referencias. Sin embargo, esas relaciones deberían ser manejadas en el modelo de dominio de PPAP (APS domain model)⁶, fuera de la definición formal del esquema XML.

El identificador (id) es descrito como un atributo de una clase, mientras que la data de referencia (ref) es también un atributo en la otra clase, a quien se hace referencia. Estas relaciones (en el esquema) son usualmente de una vía.

Las clases topes (pertenecientes al primer nivel de implementación)⁷ en el esquema común PSLX-XML, corresponden al nivel de las transacciones entre actividades de negocios. Para observar las clases y categorías del esquema común de PSLX, excluyendo las clases del primer nivel de implementación, véase Anexo A.

⁶ Debe recordarse que un modelo de dominio de PPAP, no es más que un modelo de datos abstracto general, que tiene un conjunto completo de objetos y sus relaciones, que funcionan a manera de vocabularios detallados, con el fin de utilizarlos en la planificación y programación de la producción.

⁷ Véase Anexo B



4.3.1.2. Esquema Ampliado con Base en las Especificaciones PSLX-04

El esquema que se utiliza para evaluar cualquier instancia de “PPAP” corresponde a un “trabajo de extensión” del esquema desarrollado e implementado en primera instancia por OASIS, PSLX y Nishioka Yasuyuki de la Universidad de Hosei (Japón) [Yasuyuki, OASIS y PSLX Consortium, 2005]. En los anexos⁸ se puede observar el esquema original (pps-schema-core-draft-1[1].1.xsd) y el esquema extendido (Esquema final PSLX-Tesis.xsd)

La extensión del esquema consiste en implementar todos los atributos, elementos y secuencias que se describen en el documento “Especificación Estándar XML” [PSLX-04, 2003], al “esquema pps núcleo” (pps-schema-core) publicado recientemente [Yasuyuki, OASIS y PSLX Consortium, 2005], de manera tal que cuando se validan instancias de información de negocios (que han de ir de un subsistema a otro), contra el esquema extendido editado por el autor, automáticamente se toman en cuenta “todas” las especificaciones de PSLX para la documentación de mensajes en XML; haciendo de estas instancias, documentos en XML válidos en la descripción de protocolos para el modelo desarrollado, basado en un sistema holónico de producción.

En esta extensión se definen exhaustivamente vocabularios, adicionales los del esquema original, sobre los siguientes tópicos:

- Definición de Data (Información) de Control.
- Definición de Data Relacionada con Unidad.
- Definición de Data Numérica.
- Definición de Data de Caracteres.
- Definición de Data de Tiempo.
- Definición de Data de Series de Tiempo.
- Definición de Data de Calendario.

⁸ Véase Anexos A y B respectivamente.



- Definición de Data de Atributos Relacionales.
- Información de Relación entre Elementos.
- Definición de Data de Restricción Extendida.
- Definición de Data de Eventos.
- Definición de Data de Elementos Básicos.
- Definición de Data Relacionada con Plan.
- Información para Solicitar Data.

Sobre la base teórica de todos los documentos de PSLX referidos en la bibliografía, se han corregido varias de las especificaciones que tenían algunas incongruencias. En el Anexo C, se describen ejemplos de expresiones que se consideran vitales en la información de negocios; éstas conforman el esqueleto de los mensajes y su combinación permite de manera ordenada y precisa, expresar un universo de situaciones dentro de una empresa. Dichos mensajes, se transmiten entre varios subsistemas usando los tipos de comunicación que se describen en los incisos posteriores. Es importante comparar los Anexos A y B, al igual que los Anexos E-1 y E-2 para poder observar con más claridad el trabajo de extensión realizado al esquema original.

Los tópicos cubiertos por las expresiones descritas en el Anexo C, son los siguientes:

- Elementos Básicos y Configuración.
- Expresión de Información Numérica.
- Manejo del Sistema de Unidades.
- Expresión de Información de Series de Tiempo.
- Expresión de Capacidad y Calendario.
- Expresión de Elementos Básicos.
- Expresión de Orden.



- Expresión de Operación.
- Precedencia de Operación.
- Lote y Tarea.
- Solicitud de Data.
- Agregación Parcial.
- Agregación Discreta.
- Cálculo por Parámetros.

Estas expresiones pueden servir como una especie de “guía” para los profesionales que desarrollen o manejen un software para PPAP, en el que se implementen todas las interfaces descritas en este modelo (o más inclusive), utilizando el “esquema extendido”.

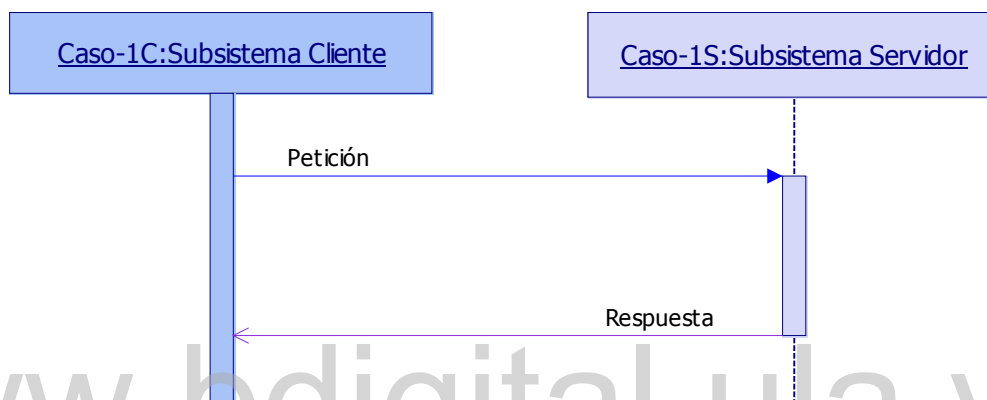
4.3.2. Mensajería Síncrona

Este proceso siempre es completado en un solo tiempo de envío-y-retorno en comunicación síncrona.



4.3.2.1. Caso 1. Petición-Respuesta

Al observar la figura 4.2, se puede apreciar que un subsistema cliente (cualquiera de los cinco propuestos en el modelo) realiza una “petición” (envía un mensaje de petición) a un subsistema servidor, quien responde con una “respuesta” (envía un mensaje de respuesta) al subsistema cliente, en donde no es posible agregar un “mensaje de recibido”.



www.bdigital.ula.ve

Figura 4.3. Caso 1

Una instancia del caso en XML, se representaría de la siguiente manera:

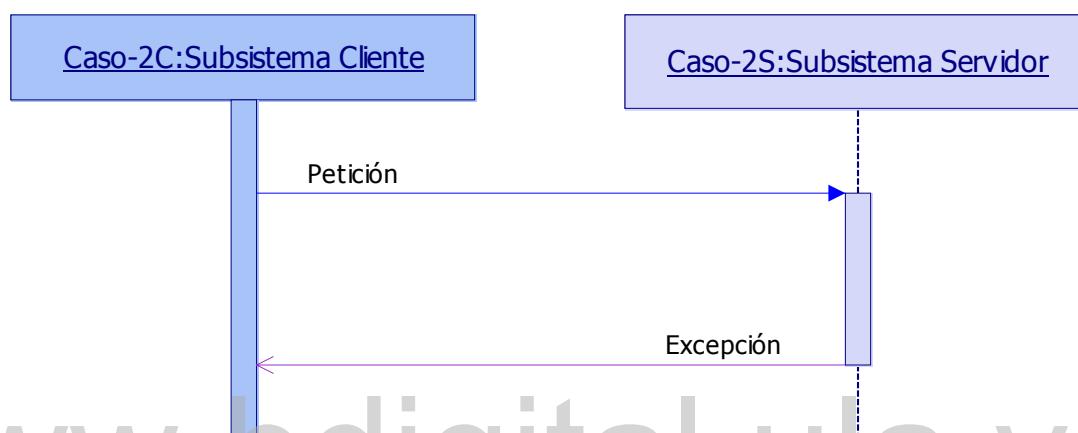
```

Ciente                                     Servidor
<pslx type="request" id="caso1C" action="getCarga">
<!--Contenido-->
</pslx>
<pslx type="response" id="caso1S" ref="caso1C" action="getCarga">
<!--Contenido-->
</pslx>
```



4.3.2.2. Caso 2. Petición-Excepción

En la figura 4.3 se puede apreciar que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con una “excepción” al subsistema cliente, en donde no es posible agregar un “mensaje de recibido”.



www.bdigital.ula.ve

Figura 4.4. Caso 2

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```
<pslx type="request" id="caso2C" action="getAlmacen">
<!--Contenido-->
</pslx>
```



```
<pslx type="exception" id="caso2S" ref="caso2C" action="getAlmacen">
<!--Contenido-->
</pslx>
```




4.3.3. Mensajería Asíncrona

4.3.3.1. Caso 3. Petición-Respuesta (patrón petición-respuesta)

Al observar la figura 4.4 se puede apreciar, que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con una “respuesta” al subsistema cliente.



Figura 4.5. Caso 3

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

```

<pslx type="request" id="caso3C" action="setEstimacion">
<!--Contenido-->
</pslx>
    
```

Servidor

```

<pslx type="response" id="caso3S" ref="caso3C" action="setEstimacion">
<!--Contenido-->
</pslx>
    
```



4.3.3.2. Caso 4. *Petición-Respuesta-Recibido (patrón petición-respuesta)*

En la figura 4.5 se puede apreciar que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con una “respuesta” al subsistema cliente, y éste devuelve un “mensaje de recibido” al servidor.

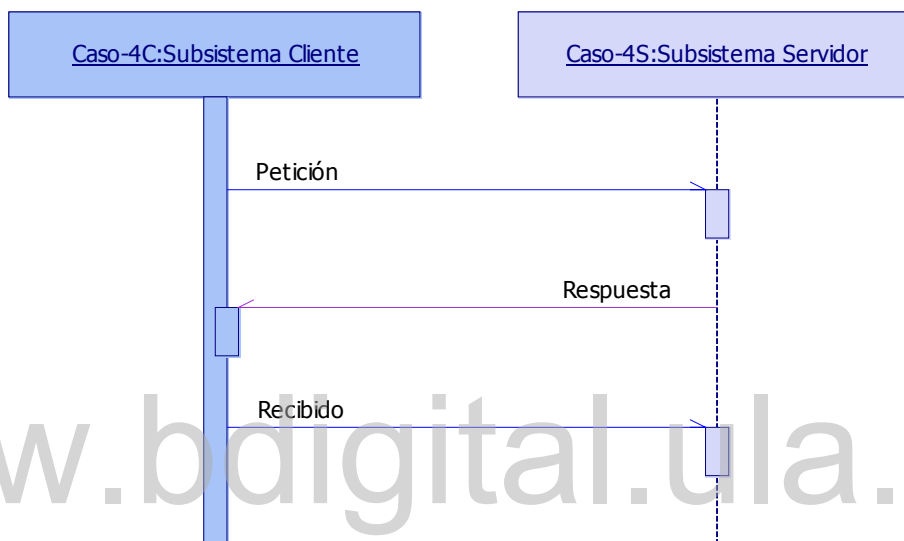


Figura 4.6. Caso 4

Una instancia del caso en XML, se representaría de la siguiente manera:

<u>Cliente</u>	<u>Servidor</u>
<pre> <pslx type="request" id="caso4C" action="setLote"> <!--Contenido--> </pslx> </pre>	<pre> <pslx type="response" id="caso4S" ref="caso4C" receipt="true" action="setLote"> <!--Contenido--> </pslx> </pre>
<pre> <pslx type="receipt" id="caso4CR" ref="caso4S" action="setLote"> <!--Contenido--> </pslx> </pre>	



4.3.3.3. Caso 5. *Petición-Recibido-Respuesta (patrón petición-respuesta)*

Al observar la figura 4.6 se puede apreciar, que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde primero con un “mensaje de recibido” al subsistema cliente, y luego con una “respuesta”.

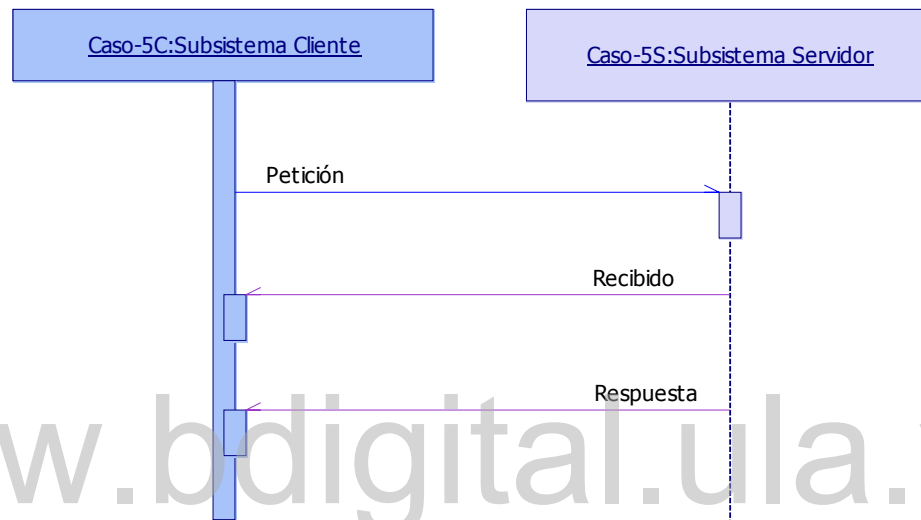


Figura 4.7. Caso 5

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```

<pslx type="request" id="caso5C" receipt="true" action="setProceso">
<!--Contenido-->
</pslx>
    
```

→

```

<pslx type="receipt" id="caso5SR" ref="caso5C" action="setProceso">
<!--Contenido-->
</pslx>
    
```

←

```

<pslx type="response" id="caso5S" ref="caso5C" action="setProceso">
<!--Contenido-->
</pslx>
    
```

←



4.3.3.4. Caso 6. *Petición-Recibido-Respuesta-Recibido (patrón petición-respuesta)*

En la figura 4.7 se aprecia que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde primero con un “mensaje de recibido” al subsistema cliente, y luego con una “respuesta”; finalmente el cliente envía un “mensaje de recibido” al servidor.

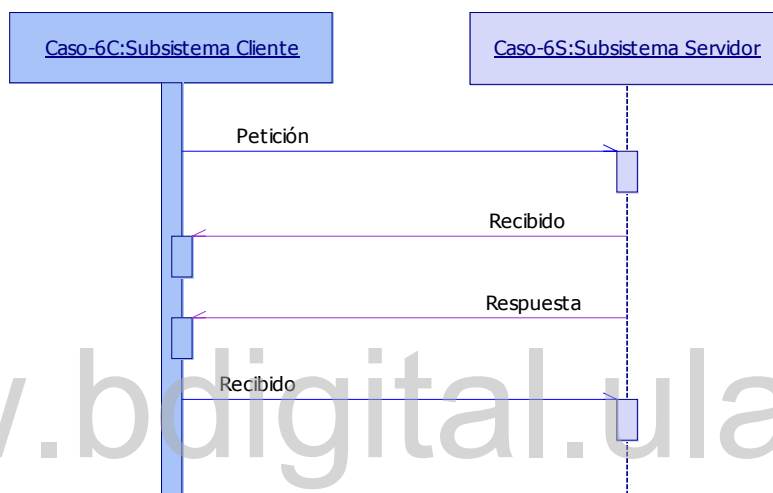
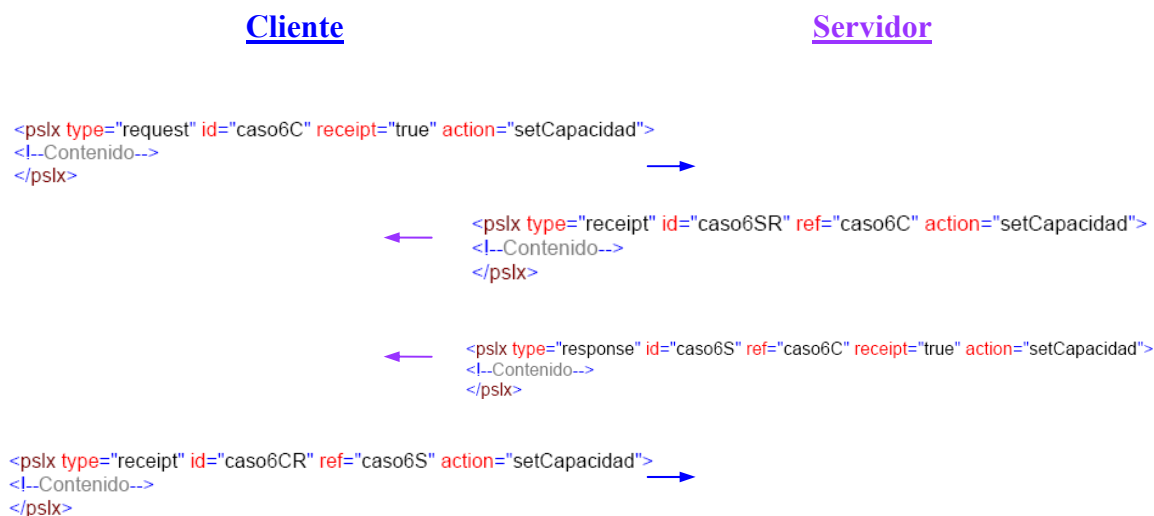


Figura 4.8. Caso 6

Una instancia del caso en XML, se representaría de la siguiente manera:





4.3.3.5. Caso 7. *Petición-Excepción (patrón petición-excepción)*

En la figura 4.8 se aprecia que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con una excepción.

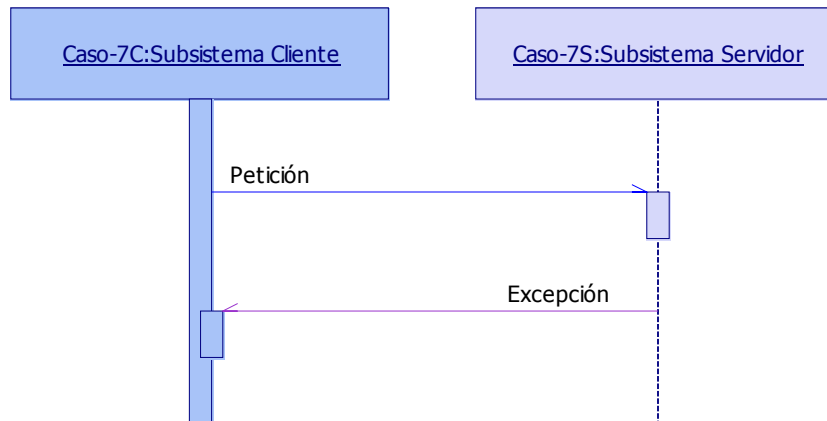


Figura 4.9. Caso 7

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```

<pslx type="request" id="caso7C" action="getTarea">
<!--Contenido-->
</pslx>
    
```

```

<pslx type="exception" id="caso7S" ref="caso7C" action="getTarea">
<!--Contenido-->
</pslx>
    
```



4.3.3.6. *Caso 8. Petición-Recibido-Excepción (patrón petición-excepción)*

Al apreciar la figura 4.9 se observar, que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con un “mensaje de recibido” y luego con una “excepción”.

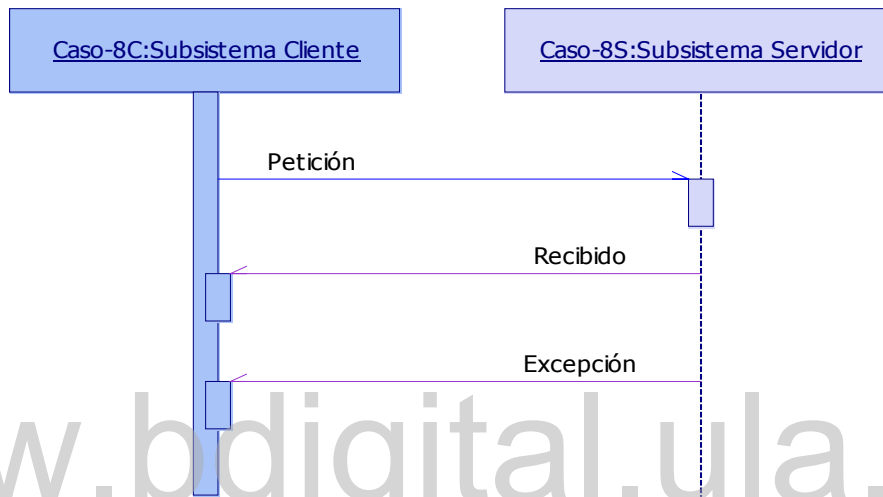


Figura 4.10. Caso 8

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```

<pslx type="request" id="caso8C" receipt="true" action="setProducto">
<!--Contenido-->
</pslx>
    
```

→

```

<pslx type="receipt" id="caso8SR" ref="caso8C" action="setProducto">
<!--Contenido-->
</pslx>
    
```

←

```

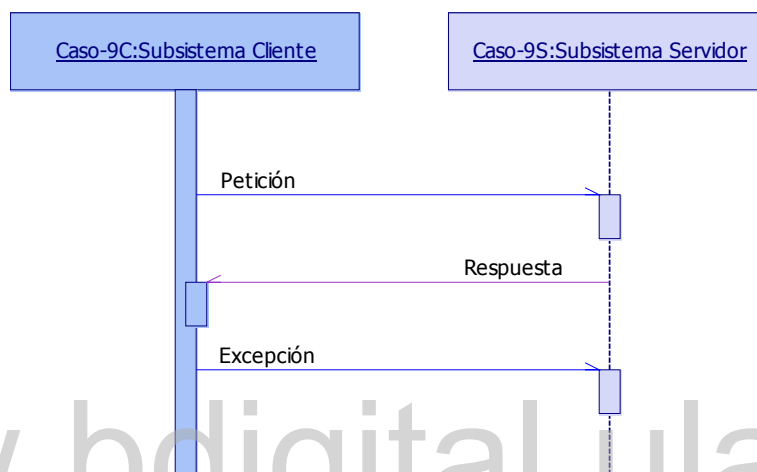
<pslx type="exception" id="caso8S" ref="caso8C" action="setProducto">
<!--Contenido-->
</pslx>
    
```

←



4.3.3.7. Caso 9. Petición-Respuesta-Excepción (patrón petición-respuesta-excepción)

Al apreciar la figura 4.10 se observar, que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con una “respuesta” al cliente; finalmente el cliente envía una “excepción” al servidor.



www.bdigital.ula.ve

Figura 4.11. Caso 9

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```
<pslx type="request" id="caso9C" action="setOpcion">
<!--Contenido-->
</pslx>
```

```
<pslx type="response" id="caso9S" ref="caso9C" action="setOpcion">
<!--Contenido-->
</pslx>
```

```
<pslx type="exception" id="caso9CE" ref="caso9S" action="setOpcion">
<!--Contenido-->
</pslx>
```



4.3.3.8. Caso 10. Petición-Recibido-Respuesta-Excepción (patrón petición-respuesta-excepción)

A través de la figura 4.11 se observar, que un subsistema cliente realiza una “petición” a un subsistema servidor, quien responde con un “mensaje de recibido”, y luego con una “respuesta” al cliente; finalmente el cliente envía una “excepción” al servidor.

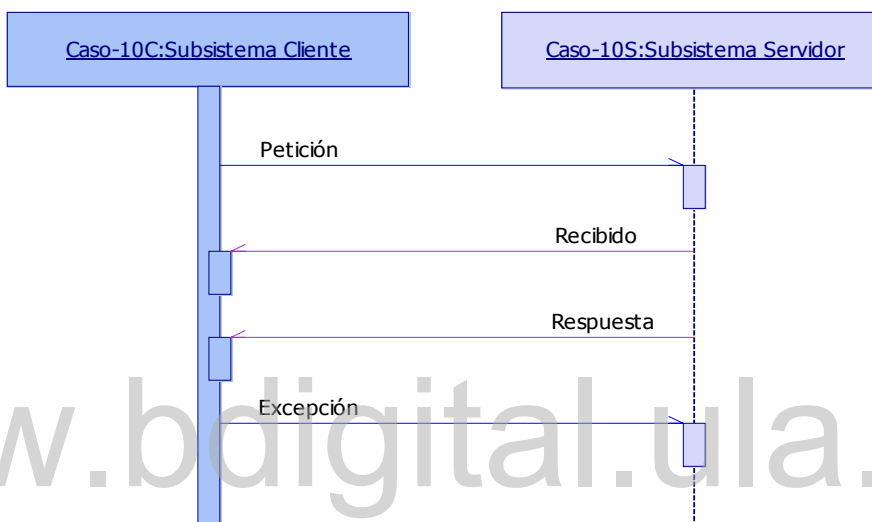


Figura 4.12. Caso 10

Una instancia del caso en XML, se representaría de la siguiente manera:

Cliente

Servidor

```

<pslx type="request" id="caso10C" receipt="true" action="getProgreso">
<!--Contenido-->
</pslx>
<pslx type="receipt" id="caso10SR" ref="caso10C" action="getProgreso">
<!--Contenido-->
</pslx>
<pslx type="response" id="caso10S" ref="caso10C" action="getProgreso">
<!--Contenido-->
</pslx>
<pslx type="exception" id="caso10CE" ref="caso10S" action="getProgreso">
<!--Contenido-->
</pslx>
  
```


CONCLUSIONES

El modelo propuesto, estructurado a manera de un sistema distribuido bajo un esquema holónico de manufactura, sirve como un esqueleto sobre el que se puede desarrollar un “modelo abstracto de datos” bien definido para cada empresa, que permite modelar las actividades de negocios particulares a ella. Una gran ventaja que proporciona este modelo, es que no limita la interacción entre los subsistemas a una relación padre-hijo, sino que define un conjunto de relaciones y asociaciones que les permiten comunicarse repetida y eficientemente, sin ir en detrimento del control que deben mantener para lograr sus objetivos efectivamente. Por tal motivo, se vence de manera absoluta la rigidez de los sistemas jerárquicos, y al mismo tiempo lo incontrolable de los sistemas heterárquicos.

Al desglosar estructural y funcionalmente el modelo propuesto utilizando los diagramas en UML; se pudo observar cómo la disposición de los subsistemas, los centros de gestiones y sus bases de datos, los módulos asociados, y los controles que las rigen; fue lograda de una forma bastante adaptada a la realidad existente en el ámbito de la manufacturación. También su estructura se ajusta bien a las tendencias futuras de las necesidades de la nueva industria; esto es, gracias al fortalecimiento de las debilidades que han embargado la empresa manufacturera. Entre los efectos más resaltantes de dicho fortalecimiento se tienen:

- 1.) Elaboración viable de algoritmos más novedosos para el desarrollo de las actividades de negocios. Éstos son posibles a través de una gestión en línea más eficiente, en donde cada subsistema tiene conocimiento de la situación de la fábrica

a tiempo real, gracias a la estandarización de los datos que constituyen la información de negocios, a través de un esquema común que es entendido por todo el sistema distribuido, indiferentemente de la cantidad de aplicaciones individuales diferentes que contenga; en otras palabras, establece la integración de aplicaciones entre diferentes actividades de negocio.

2.) Flexibilidad y robustez en el mejoramiento continuo de los procesos de negocios y los procesos de ingeniería, ya que la migración de datos de un sistema previo a uno nuevo, no justificará un cambio de su formato.

De este trabajo también se puede deducir, que hay facilidad para utilizar el modelo propuesto en una compañía en particular, ya que mejoraría directamente su capacidad funcional y los sistemas de planificación de recursos convencionales; además, si no hay interés de la empresa en desarrollar una nueva ontología que rijan el modelo, no importa porque éste puede desempeñarse bajo la ontología PSLX y el “modelo de dominio de PPAP (APS)”, que la organización propone como base para especificar modelos abstractos de datos.

El esquema final ampliado, que se desarrolló según el borrador (draft) original, es muy completo en relación al original. Posee todos los elementos necesarios para la generación de actividades de negocios, desde las más sencillas, hasta las más complejas. Aunque el esquema fue ampliado en base a las especificaciones de XML descritas por PSLX, en éste se corrigieron una serie de errores de congruencia que poseían las especificaciones originales. Con el esquema arreglado, y un buen editor de XML (como el XMLSpy) es muy sencillo manipular la información de negocios, e incluso cambiar elementos y atributos según se considere necesario en las particularidades de cada empresa.

El hecho de que el esquema original haya estado tan poco tiempo disponible en la Internet (ya no lo está, al menos de manera gratuita), y considerando que ser miembro de OASIS (la organización que publicó el borrador) implica cancelar cuotas por la medida mínima de 3000 US\$ y 4000 US\$ anuales para particulares; sugiero que debe pensarse en

este esquema ampliado, como un soporte importante para el desarrollo de otros esquemas para la planificación y programación de la producción.

www.bdigital.ula.ve

RECOMENDACIONES

Pueden hacerse varias recomendaciones a este trabajo:

Definir la distribución, funcionamiento y variantes de las interfaces, que en el modelo propuesto, comunican los centros de gestión con las clases adyacentes, y también las que comunican las clases entre sí.

En el elemento primitivo “ítem”, descrito en el esquema ampliado, debería clasificarse la forma y el estado en que éstos se almacenan, por ejemplo: en estado líquido, sólido o gaseoso; perecedero o no perecedero, de índole electrónico, mecánico, o eléctrico, de manipulación manual o mecánica, entre otros.

Podría definirse un elemento administrativo “stumble” (tropiezo) que funcione bajo la gestión de control de ejecución, para detener órdenes y procedimientos con una interrupción brusca, y podría ser implementado en la interfaz “setEmergencia”. También podría especificarse un elemento administrativo “rock” (piedra) que podría funcionar como un obstáculo, es decir, como algo que no permite a un evento ocurrir.

Se sugiere traducir todo el esquema ampliado al castellano para facilitar su implantación y entendimiento.

BIBLIOGRAFÍA

[Benoit, 2000] Benoit, Marchal. 2000. *XML by example*. Indianapolis, Indiana 462901, QUE.

[Boggino, 2005] Boggino G., Adriana. 2005. *Anemona: Una Metodología Multi Agente para Sistemas Holónicos de Fabricación*. Tesis de Doctorado, Doctorado en Informática, Universidad Politécnica de Valencia. Valencia-España.

[Cárdenas, 2004] Alfonso Araujo Cárdenas. 2004. *Sistemas Distribuidos*. Disponible en: <http://mx.geocities.com/alfonsoaraujocardenas/sistemasdistribuidos.html>.

[Chacón, Altamiranda y Colina] Chacón E., Altamiranda E. y Colina E. *Estructuras Teleinformáticas para la Implantación de Sistemas de Monitoreo y Supervisión de Sistemas de Producción*. Pamplona – Colombia.

[Deitel 2001] Deitel, P. 2001. *XML, How to Program*. Prentice Hall. Deitel & Associates Inc.

[Dye, 1999] Dye, Charles. 1999. *Oracle Distributed Systems*. 1st Edition. Pp 548.

[Elliott y W. Scott, 2004] Elliott R. Harold y W. Scott. 2004. *XML in a Nutshell. 3rd Edition*. Publicado por O'Reilly Media, Inc. United States of America.

[Enslow,1978] Enslow, P.H. *What is a “distributed” data processing system?*

- [FEMZ, 1999] Federación de Empresarios del Metal de Zaragoza (FEMZ). 1999. *Curso de Gestión de Producción*. Zaragoza, España. Disponible en: <http://www.femz.es/cursos/Produccion/tema01/portada01.htm>.
- [González, 1998] González Barahona, Jesús M. 1998. *Arquitectura de Comunicaciones para Grupos de Procesos Replicados*. Tesis de Doctorado, Postgrado de E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid. Madrid-España. Disponible: http://gsync.escet.urjc.es/simple_com/phd-thesis-es/phd-thesis.html.
- [Harold, 1999] Harold E.R. 1999. *XML Bible*. Foster City. CA: IDG Books Worldwide.
- [HMS, 1994] HMS, P. R. 1994. *HMS Requirements*. <http://hms.ifw.uniannover.de/>: HMS Server.
- [Iwata y Onosato, 1994] Iwata, K., y Onosato, M. 1994. Random Manufacturing System: a New Concept of Manufacturing Systems for production to order. *Annals of the CIRP* 43/1/1994.
- [Kalpakjian y Schmid, 2002] Kalpakjian, S. y Schmid, S. 2002. *Manufactura, ingeniería y tecnología*. Pearson Educación.
- [Kimura, 1993] Kimura, F. 1993. A Product and Process Model for Virtual Manufacturing Systems. *Annals of the CIRP* 42/1/1993.
- [Koestler, 1971] Koestler, A. 1971. *The Ghost in the Machine*. Arkana Books.
- [Medina, 2004] Medina Devia, Judith J. 2004. *Automatización Integrada de Procesos de Producción Continua Bajo Arquitectura Holárquica*. Tesis de Maestría, Postgrado de Automatización e Instrumentación, Universidad de los Andes. Mérida-Venezuela.
- [Lamport y Melliar-Smith, 1985] Lamport L. y Melliar-Smith P.M. 1985. *Synchronizing clocks in the presence of faults*. *Journal of the ACM*, 32(1):52-78.

- [OASIS, 2004] OASIS. 2004. *PPS (Production Planning and Scheduling) Part 1: Core Elements*. Working Draft 01-b. Editado por Yasuyuki Nishioka, PSLX Consortium, Hosei University. Disponible en: <http://www.oasis-open.org/apps/org/workgroup/pps/>.
- [Okino, 1993] Okino, N. 1993. Bionic Manufacturing Systems. *Flexible Manufacturing Systems, Past, Present, Future* 73–95.
- [Porter, 1985] Porter, Michael. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY The Free Press.
- [PSLX-02, 2003] PSLX Consortium. 2003. *PSLX Engineering Specification, APS Agent Model*. PSLX-02, Recommendation, Version 1.0. Disponible en: <http://www.pslx.org/en/publications.html>.
- [PSLX White Paper, 2005] PSLX White Paper, PSLX Consortium. 2005. *Advanced Planning and Scheduling (APS) Conceptual Definition and Implementation*. Disponible en: <http://www.pslx.org/en/publications.html>.
- [PSLX-03, 2003] PSLX-03, 2003 *PSLX Domain Objects*. Disponible en: <http://www.pslx.org/en/publications.html>.
- [PSLX-04, 2003] PSLX Engineering Specification, PSLX Consortium. 2003. *XML Specification*. PSLX-04 versión 1.0. Disponible en: <http://www.pslx.org/en/publications.html>.
- [Riggs, 2001] Riggs, James L. 2001. *Sistemas de producción: planeación, análisis y control*. Editorial Limusa. México.
- [Seidel, 1994] Seidel, D. 1994. HMS - Strategies. *Vol0, Overview, IMS/HMS TC5 deliverable*.
- [Sevilla, 2005] Sevilla Ruiz, Diego. *Sistemas Distribuidos*. DITEC, Facultad de Informática. Murcia-España. Disponible en: <http://ditec.um.es/ssdd/>.

- [Shen y Norrie, 1999] Shen, W., y Norrie, D. 1999. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems, an Internatinal Journal* 1(2):129–156.
- [Sinan, 2003] Sinan Si, Alhir. 2003. *Learning UML*. O'Reilly.
- [Sistemas Distribuidos, 2003] Sistemas Distribuidos. [Documento en línea]. 2003. Disponible en: <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>.
- [Synthesis C.I., 2004] Synthesis Consultores Internacionales. 2004. *La Cadena de Valor de Michael Porter*. San Salvador, El Salvador, Disponible: <http://www.synthesisci.com/irf/consultas/2004/08/170804.htm>.
- [Tanenbaum y Maarten, 2001] Tanenbaum A. y Maarten V. 2001. *Distributed Systems: Principles and Paradigms*. 1st Edition. Pp. 803.
- [Ueda, 1993] Ueda, K. 1993. A Genetic Approach toward Future Manufacturing Systems. *Flexible Manufacturing Systems, Past, Present, Future* 221–228.
- [Vassos y Toueg, 1993] Vassos, Hadzilacos y Toueg Sam. 1993. *Fault-tolerant broadcasts and related problems*. Pp 97-145.
- [Warnecke, 1992] Warnecke, H. 1992. *Die Fraktale Fabrik, Revolution Unternehmenskultur*. Springer Verlag.
- [Wikipedia, Cadena del valor] Wikipedia. *Cadena de valor*. Disponible en: http://es.wikipedia.org/wiki/Cadena_de_valor.
- [Wyns, 1999] Wyns, Jo. 1999. *Reference Architecture for Holonic Manufacturing Systems*. Katholieke Universiteit Leuven, Heverlee (Leuven). Belgium, Maart.

°[Yasuyuki, OASIS y PSLX Consortium, 2005] Yasuyuki Nishioka, OASIS y PSLX Consortium. 2005. *pps-schema-core-draft-1[1].1*. Disponible en: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pps.

www.bdigital.ula.ve

ANEXO A

ESQUEMA ORIGINAL PSLX

www.bdigital.ula.ve
(pps-schema-core-draft-1[1].1.xsd)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Nishioka Yasuyuki (Hosei University) -->
3 <xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:pps="
http://www.oasis-open.org/committees/pps/schema" targetNamespace="http://www.oasis-open.org/committees/pps/schema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
4 <!--
5 administrative elements
6 -->
7 <xsd:complexType name="scaleType">
8 <xsd:sequence>
9 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
10 </xsd:sequence>
11 <xsd:attribute name="name" type="xsd:string" use="required"/>
12 <xsd:attribute name="value" type="xsd:duration"/>
13 <xsd:attribute name="base" type="xsd:dateTime"/>
14 </xsd:complexType>
15 <xsd:complexType name="stoneType">
16 <xsd:sequence>
17 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
18 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
19 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
20 </xsd:sequence>
21 <xsd:attribute name="name" type="xsd:string" use="required"/>
22 </xsd:complexType>
23 <!--
24 primitive elements
25 -->
26 <xsd:complexType name="customerType">
27 <xsd:sequence>
28 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
29 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
30 <xsd:element name="calendar" type="pps:calendarType" minOccurs="0" maxOccurs="unbounded"/>
31 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
32 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
33 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
34 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
35 </xsd:sequence>
36 <xsd:attribute name="name" type="xsd:string" use="required"/>
37 <xsd:attribute name="parent" type="xsd:string"/>
38 <xsd:attribute name="type" type="xsd:string"/>
39 </xsd:complexType>
40 <xsd:complexType name="supplierType">
41 <xsd:sequence>
42 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
43 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
44 <xsd:element name="calendar" type="pps:calendarType" minOccurs="0" maxOccurs="unbounded"/>
45 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
46 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
47 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
48 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
49 </xsd:sequence>
50 <xsd:attribute name="name" type="xsd:string" use="required"/>
51 <xsd:attribute name="parent" type="xsd:string"/>
52 <xsd:attribute name="type" type="xsd:string"/>
53 </xsd:complexType>
54 <xsd:complexType name="itemType">
55 <xsd:sequence>
56 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
57 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
58 <xsd:element name="produced" type="pps:producedType" minOccurs="0" maxOccurs="unbounded"/>
59 <xsd:element name="consumed" type="pps:consumedType" minOccurs="0" maxOccurs="unbounded"/>
60 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
61 <xsd:element name="available" type="pps:availableType" minOccurs="0" maxOccurs="unbounded"/>
62 <xsd:element name="stock" type="pps:stockType" minOccurs="0" maxOccurs="unbounded"/>
63 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
64 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
65 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
66 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
67 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
68 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
69 </xsd:sequence>
70 <xsd:attribute name="name" type="xsd:string" use="required"/>
71 <xsd:attribute name="parent" type="xsd:string"/>
72 <xsd:attribute name="type" type="xsd:string"/>
73 </xsd:complexType>
74 <xsd:complexType name="resourceType">
75 <xsd:sequence>
76 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
77 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
78 <xsd:element name="assigned" type="pps:assignedType" minOccurs="0" maxOccurs="unbounded"/>
79 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
80 <xsd:element name="calendar" type="pps:calendarType" minOccurs="0" maxOccurs="unbounded"/>
81 <xsd:element name="available" type="pps:availableType" minOccurs="0" maxOccurs="unbounded"/>
82 <xsd:element name="load" type="pps:loadType" minOccurs="0" maxOccurs="unbounded"/>
83 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
84 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
85 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
86 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
87 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
88 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
89 </xsd:sequence>
90 </xsd:complexType>
91 </xsd:schema>

```

```

89 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
90 </xsd:sequence>
91 <xsd:attribute name="name" type="xsd:string" use="required"/>
92 <xsd:attribute name="parent" type="xsd:string"/>
93 <xsd:attribute name="type" type="xsd:string"/>
94 </xsd:complexType>
95 <xsd:complexType name="operationType">
96 <xsd:sequence>
97 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
98 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
99 <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
100 <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
101 <xsd:element name="assign" type="pps:assignType" minOccurs="0" maxOccurs="unbounded"/>
102 <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
103 <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
104 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
105 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
106 <xsd:element name="start" type="pps:startType" minOccurs="0"/>
107 <xsd:element name="end" type="pps:endType" minOccurs="0"/>
108 <xsd:element name="event" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
109 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
110 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
111 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
112 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
113 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
114 </xsd:sequence>
115 <xsd:attribute name="name" type="xsd:string" use="required"/>
116 <xsd:attribute name="parent" type="xsd:string"/>
117 <xsd:attribute name="type" type="xsd:string"/>
118 </xsd:complexType>
119 <xsd:complexType name="orderType">
120 <xsd:sequence>
121 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
122 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
123 <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
124 <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
125 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
126 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
127 <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
128 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
129 <xsd:element name="start" type="pps:startType" minOccurs="0"/>
130 <xsd:element name="end" type="pps:endType" minOccurs="0"/>
131 <xsd:element name="release" type="pps:releaseType" minOccurs="0"/>
132 <xsd:element name="duetime" type="pps:duetimeType" minOccurs="0"/>
133 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
134 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
135 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
136 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
137 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
138 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
139 </xsd:sequence>
140 <xsd:attribute name="name" type="xsd:string" use="required"/>
141 <xsd:attribute name="parent" type="xsd:string"/>
142 <xsd:attribute name="type" type="xsd:string"/>
143 <xsd:attribute name="customer" type="xsd:string"/>
144 <xsd:attribute name="supplier" type="xsd:string"/>
145 <xsd:attribute name="item" type="xsd:string"/>
146 <xsd:attribute name="resource" type="xsd:string"/>
147 <xsd:attribute name="operation" type="xsd:string"/>
148 </xsd:complexType>
149 <xsd:complexType name="dispatchType">
150 <xsd:sequence>
151 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
152 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
153 <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
154 <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
155 <xsd:element name="assign" type="pps:assignType" minOccurs="0" maxOccurs="unbounded"/>
156 <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
157 <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
158 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
159 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
160 <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
161 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
162 <xsd:element name="start" type="pps:startType" minOccurs="0"/>
163 <xsd:element name="end" type="pps:endType" minOccurs="0"/>
164 <xsd:element name="release" type="pps:releaseType" minOccurs="0"/>
165 <xsd:element name="duetime" type="pps:duetimeType" minOccurs="0"/>
166 <xsd:element name="event" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
167 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
168 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
169 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
170 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
171 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
172 </xsd:sequence>
173 <xsd:attribute name="name" type="xsd:string" use="required"/>
174 <xsd:attribute name="parent" type="xsd:string"/>
175 <xsd:attribute name="type" type="xsd:string"/>
176 <xsd:attribute name="operation" type="xsd:string"/>
177 </xsd:complexType>
178 <xsd:complexType name="lotType">

```

```

179 <xsd:sequence>
180 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
181 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
182 <xsd:element name="produced" type="pps:producedType" minOccurs="0" maxOccurs="unbounded"/>
183 <xsd:element name="consumed" type="pps:consumedType" minOccurs="0" maxOccurs="unbounded"/>
184 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
185 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
186 <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
187 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
188 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
189 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
190 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
191 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
192 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
193 </xsd:sequence>
194 <xsd:attribute name="name" type="xsd:string" use="required"/>
195 <xsd:attribute name="parent" type="xsd:string"/>
196 <xsd:attribute name="type" type="xsd:string"/>
197 <xsd:attribute name="item" type="xsd:string"/>
198 </xsd:complexType>
199 <xsd:complexType name="taskType">
200 <xsd:sequence>
201 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
202 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
203 <xsd:element name="assigned" type="pps:assignedType" minOccurs="0" maxOccurs="unbounded"/>
204 <xsd:element name="relation" type="pps:relationType" minOccurs="0" maxOccurs="unbounded"/>
205 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
206 <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
207 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
208 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
209 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
210 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
211 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
212 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
213 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
214 </xsd:sequence>
215 <xsd:attribute name="name" type="xsd:string" use="required"/>
216 <xsd:attribute name="parent" type="xsd:string"/>
217 <xsd:attribute name="type" type="xsd:string"/>
218 <xsd:attribute name="resource" type="xsd:string"/>
219 </xsd:complexType>
220 <!--
221 relational elements between primitives
222 -->
223 <xsd:complexType name="composeType">
224 <xsd:sequence>
225 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
226 </xsd:sequence>
227 <xsd:attribute name="parent" type="xsd:string"/>
228 <xsd:attribute name="type" type="xsd:string"/>
229 </xsd:complexType>
230 <xsd:complexType name="composedType">
231 <xsd:sequence>
232 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
233 </xsd:sequence>
234 <xsd:attribute name="parent" type="xsd:string"/>
235 <xsd:attribute name="type" type="xsd:string"/>
236 </xsd:complexType>
237 <xsd:complexType name="produceType">
238 <xsd:sequence>
239 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
240 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
241 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
242 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
243 </xsd:sequence>
244 <xsd:attribute name="type" type="xsd:string"/>
245 <xsd:attribute name="item" type="xsd:string"/>
246 <xsd:attribute name="lot" type="xsd:string"/>
247 </xsd:complexType>
248 <xsd:complexType name="producedType">
249 <xsd:sequence>
250 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
251 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
252 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
253 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
254 </xsd:sequence>
255 <xsd:attribute name="type" type="xsd:string"/>
256 <xsd:attribute name="operation" type="xsd:string"/>
257 <xsd:attribute name="dispatch" type="xsd:string"/>
258 </xsd:complexType>
259 <xsd:complexType name="consumeType">
260 <xsd:sequence>
261 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
262 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
263 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
264 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
265 </xsd:sequence>
266 <xsd:attribute name="type" type="xsd:string"/>
267 <xsd:attribute name="item" type="xsd:string"/>
268 <xsd:attribute name="lot" type="xsd:string"/>

```



```

269 </xsd:complexType>
270 <xsd:complexType name="consumedType">
271   <xsd:sequence>
272     <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
273     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
274     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
275     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
276   </xsd:sequence>
277   <xsd:attribute name="type" type="xsd:string"/>
278   <xsd:attribute name="operation" type="xsd:string"/>
279   <xsd:attribute name="dispatch" type="xsd:string"/>
280 </xsd:complexType>
281 <xsd:complexType name="assignType">
282   <xsd:sequence>
283     <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
284     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
285     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
286     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
287   </xsd:sequence>
288   <xsd:attribute name="type" type="xsd:string"/>
289   <xsd:attribute name="resource" type="xsd:string"/>
290   <xsd:attribute name="task" type="xsd:string"/>
291 </xsd:complexType>
292 <xsd:complexType name="assignedType">
293   <xsd:sequence>
294     <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
295     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
296     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
297     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
298   </xsd:sequence>
299   <xsd:attribute name="type" type="xsd:string"/>
300   <xsd:attribute name="operation" type="xsd:string"/>
301   <xsd:attribute name="dispatch" type="xsd:string"/>
302 </xsd:complexType>
303 <xsd:complexType name="predecessorType">
304   <xsd:sequence>
305     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
306     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
307   </xsd:sequence>
308   <xsd:attribute name="type" type="xsd:string"/>
309   <xsd:attribute name="stone" type="xsd:string"/>
310   <xsd:attribute name="operation" type="xsd:string"/>
311   <xsd:attribute name="order" type="xsd:string"/>
312   <xsd:attribute name="dispatch" type="xsd:string"/>
313   <xsd:attribute name="item" type="xsd:string"/>
314   <xsd:attribute name="lot" type="xsd:string"/>
315 </xsd:complexType>
316 <xsd:complexType name="successorType">
317   <xsd:sequence>
318     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
319     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
320   </xsd:sequence>
321   <xsd:attribute name="type" type="xsd:string"/>
322   <xsd:attribute name="stone" type="xsd:string"/>
323   <xsd:attribute name="operation" type="xsd:string"/>
324   <xsd:attribute name="order" type="xsd:string"/>
325   <xsd:attribute name="dispatch" type="xsd:string"/>
326   <xsd:attribute name="item" type="xsd:string"/>
327   <xsd:attribute name="lot" type="xsd:string"/>
328 </xsd:complexType>
329 <xsd:complexType name="relationType">
330   <xsd:sequence>
331     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
332     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
333   </xsd:sequence>
334   <xsd:attribute name="name" type="xsd:string" use="required"/>
335   <xsd:attribute name="parent" type="xsd:string"/>
336   <xsd:attribute name="type" type="xsd:string"/>
337 </xsd:complexType>
338 <!--
339 property elements of primitives
340 -->
341 <xsd:complexType name="locationType">
342   <xsd:sequence>
343     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
344     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
345     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
346   </xsd:sequence>
347   <xsd:attribute name="name" type="xsd:string" use="required"/>
348   <xsd:attribute name="type" type="xsd:string"/>
349 </xsd:complexType>
350 <xsd:complexType name="calendarType">
351   <xsd:sequence>
352     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
353     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
354     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
355   </xsd:sequence>
356   <xsd:attribute name="type" type="xsd:string"/>
357 </xsd:complexType>
358 <xsd:complexType name="availableType">

```

```

359 <xsd:sequence>
360 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
361 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
362 </xsd:sequence>
363 <xsd:attribute name="type" type="xsd:string"/>
364 </xsd:complexType>
365 <xsd:complexType name="progressType">
366 <xsd:sequence>
367 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
368 <xsd:element name="char" type="pps:charType" minOccurs="0"/>
369 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
370 </xsd:sequence>
371 <xsd:attribute name="name" type="xsd:string" use="required"/>
372 <xsd:attribute name="type" type="xsd:string"/>
373 </xsd:complexType>
374 <xsd:complexType name="stockType">
375 <xsd:sequence>
376 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
377 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
378 </xsd:sequence>
379 <xsd:attribute name="type" type="xsd:string"/>
380 </xsd:complexType>
381 <xsd:complexType name="loadType">
382 <xsd:sequence>
383 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
384 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
385 </xsd:sequence>
386 <xsd:attribute name="type" type="xsd:string"/>
387 </xsd:complexType>
388 <xsd:complexType name="specType">
389 <xsd:sequence>
390 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
391 <xsd:element name="char" type="pps:charType" minOccurs="0"/>
392 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
393 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
394 </xsd:sequence>
395 <xsd:attribute name="name" type="xsd:string" use="required"/>
396 <xsd:attribute name="type" type="xsd:string"/>
397 </xsd:complexType>
398 <!--
399 temporal events on primitives
400 -->
401 <xsd:complexType name="startType">
402 <xsd:sequence>
403 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
404 </xsd:sequence>
405 </xsd:complexType>
406 <xsd:complexType name="endType">
407 <xsd:sequence>
408 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
409 </xsd:sequence>
410 </xsd:complexType>
411 <xsd:complexType name="releaseType">
412 <xsd:sequence>
413 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
414 </xsd:sequence>
415 </xsd:complexType>
416 <xsd:complexType name="duetimeType">
417 <xsd:sequence>
418 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
419 </xsd:sequence>
420 </xsd:complexType>
421 <xsd:complexType name="eventType">
422 <xsd:sequence>
423 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
424 </xsd:sequence>
425 <xsd:attribute name="name" type="xsd:string" use="required"/>
426 </xsd:complexType>
427 <!--
428 Basic data elements
429 -->
430 <xsd:complexType name="qtyType">
431 <xsd:sequence>
432 <xsd:element name="min" type="pps:minType" minOccurs="0"/>
433 <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
434 </xsd:sequence>
435 <xsd:attribute name="value" type="xsd:double"/>
436 <xsd:attribute name="unit" type="xsd:string"/>
437 <xsd:attribute name="base" type="xsd:double"/>
438 </xsd:complexType>
439 <xsd:complexType name="priceType">
440 <xsd:sequence>
441 <xsd:element name="min" type="pps:minType" minOccurs="0"/>
442 <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
443 </xsd:sequence>
444 <xsd:attribute name="value" type="xsd:double"/>
445 <xsd:attribute name="unit" type="xsd:string"/>
446 <xsd:attribute name="base" type="xsd:double"/>
447 </xsd:complexType>
448 <xsd:complexType name="charType">

```

www.bdigital.ula.ve

```

449 <xsd:sequence>
450 <xsd:element name="enumerate" type="pps:enumerateType" minOccurs="0" maxOccurs="unbounded"/>
451 </xsd:sequence>
452 <xsd:attribute name="value" type="xsd:string"/>
453 </xsd:complexType>
454 <xsd:complexType name="durationType">
455 <xsd:sequence>
456 <xsd:element name="min" type="pps:minType" minOccurs="0"/>
457 <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
458 <xsd:element name="shortest" type="pps:shortestType" minOccurs="0"/>
459 <xsd:element name="longest" type="pps:longestType" minOccurs="0"/>
460 </xsd:sequence>
461 <xsd:attribute name="value" type="xsd:duration"/>
462 <xsd:attribute name="count" type="xsd:long"/>
463 <xsd:attribute name="scale" type="xsd:string"/>
464 </xsd:complexType>
465 <xsd:complexType name="timeType">
466 <xsd:sequence>
467 <xsd:element name="min" type="pps:minType" minOccurs="0"/>
468 <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
469 <xsd:element name="earliest" type="pps:earliestType" minOccurs="0"/>
470 <xsd:element name="latest" type="pps:latestType" minOccurs="0"/>
471 </xsd:sequence>
472 <xsd:attribute name="value" type="xsd:dateTime"/>
473 <xsd:attribute name="count" type="xsd:long"/>
474 <xsd:attribute name="scale" type="xsd:string"/>
475 <xsd:attribute name="stone" type="xsd:string"/>
476 </xsd:complexType>
477 <!--
478 administrative properties
479 -->
480 <xsd:complexType name="priorityType">
481 <xsd:sequence>
482 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
483 <xsd:element name="char" type="pps:charType" minOccurs="0"/>
484 </xsd:sequence>
485 </xsd:complexType>
486 <xsd:complexType name="displayType">
487 <xsd:sequence>
488 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
489 <xsd:element name="char" type="pps:charType" minOccurs="0"/>
490 </xsd:sequence>
491 <xsd:attribute name="name" type="xsd:string" use="required"/>
492 </xsd:complexType>
493 <xsd:complexType name="descriptionType"/>
494 <!--
495 data auxiliary elements for query
496 -->
497 <xsd:complexType name="minType">
498 <xsd:attribute name="value" type="xsd:double"/>
499 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
500 </xsd:complexType>
501 <xsd:complexType name="maxType">
502 <xsd:attribute name="value" type="xsd:double"/>
503 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
504 </xsd:complexType>
505 <xsd:complexType name="earliestType">
506 <xsd:attribute name="value" type="xsd:dateTime"/>
507 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
508 </xsd:complexType>
509 <xsd:complexType name="latestType">
510 <xsd:attribute name="value" type="xsd:dateTime"/>
511 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
512 </xsd:complexType>
513 <xsd:complexType name="shortestType">
514 <xsd:attribute name="value" type="xsd:duration"/>
515 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
516 </xsd:complexType>
517 <xsd:complexType name="longestType">
518 <xsd:attribute name="value" type="xsd:duration"/>
519 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
520 </xsd:complexType>
521 <xsd:complexType name="enumerateType">
522 <xsd:attribute name="value" type="xsd:string"/>
523 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
524 </xsd:complexType>
525 </xsd:schema>
526

```


ANEXO B

ESQUEMA FINAL PSLX-TESIS

www.bdigital.ula.ve

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- editado con XMLSPY v2005 rel. 3 U (http://www.xmlspy.com) por Héctor Zambrano (Universidad de los Andes) -->
3 <xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:pps="
http://www.oasis-open.org/committees/pps/schema" xmlns:pslx="http://www.oasis-open.org/committees/pps/schema-instance" targetNamespace="
http://www.oasis-open.org/committees/pps/schema" elementFormDefault="qualified" attributeFormDefault="unqualified">
4 <!--
5 Elemento raíz (root element)
6 -->
7 <xsd:element name="ppap"/>
8 <!--
9 1er Nivel de Implementación PSLX (first implementation level)
10 -->
11 <xsd:element name="pslx">
12 <xsd:complexType>
13 <xsd:sequence>
14 <xsd:element name="unit" type="pps:unitType" minOccurs="0" maxOccurs="unbounded"/>
15 <xsd:element name="timeScale" type="pps:timeScaleConversionType" minOccurs="0" maxOccurs="unbounded"/>
16 <xsd:element name="error" type="pps:errorType" minOccurs="0" maxOccurs="unbounded"/>
17 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
18 <xsd:element name="shift" type="pps:shiftType" minOccurs="0" maxOccurs="unbounded"/>
19 <xsd:element name="profile" type="pps:profileType" minOccurs="0"/>
20 <xsd:element name="customer" type="pps:customerType" minOccurs="0" maxOccurs="unbounded"/>
21 <xsd:element name="supplier" type="pps:supplierType" minOccurs="0" maxOccurs="unbounded"/>
22 <xsd:element name="item" type="pps:itemType" minOccurs="0" maxOccurs="unbounded"/>
23 <xsd:element name="resource" type="pps:resourceType" minOccurs="0" maxOccurs="unbounded"/>
24 <xsd:element name="order" type="pps:orderType" minOccurs="0" maxOccurs="unbounded"/>
25 <xsd:element name="operation" type="pps:operationType" minOccurs="0" maxOccurs="unbounded"/>
26 <xsd:element name="lot" type="pps:lotType" minOccurs="0" maxOccurs="unbounded"/>
27 <xsd:element name="task" type="pps:taskType" minOccurs="0" maxOccurs="unbounded"/>
28 <xsd:element name="event" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
29 <xsd:element name="parameter" type="pps:parameterType" minOccurs="0" maxOccurs="unbounded"/>
30 <xsd:element name="expression" type="pps:expressionType" minOccurs="0" maxOccurs="unbounded"/>
31 </xsd:sequence>
32 <xsd:attribute name="type" default="file">
33 <xsd:simpleType>
34 <xsd:restriction base="xsd:string">
35 <xsd:enumeration value="request"/>
36 <xsd:enumeration value="response"/>
37 <xsd:enumeration value="receipt"/>
38 <xsd:enumeration value="exception"/>
39 <xsd:enumeration value="file"/>
40 </xsd:restriction>
41 </xsd:simpleType>
42 </xsd:attribute>
43 <xsd:attribute name="id" type="xsd:ID"/>
44 <xsd:attribute name="ref" type="xsd:IDREF"/>
45 <xsd:attribute name="action">
46 <xsd:simpleType>
47 <xsd:restriction base="xsd:string">
48 <xsd:enumeration value="setPlan"/>
49 <xsd:enumeration value="getPlan"/>
50 <xsd:enumeration value="setPrograma"/>
51 <xsd:enumeration value="getPrograma"/>
52 <xsd:enumeration value="setProducto"/>
53 <xsd:enumeration value="getProducto"/>
54 <xsd:enumeration value="setProceso"/>
55 <xsd:enumeration value="getProceso"/>
56 <xsd:enumeration value="setOrden"/>
57 <xsd:enumeration value="getOrden"/>
58 <xsd:enumeration value="setOpcion"/>
59 <xsd:enumeration value="getOpcion"/>
60 <xsd:enumeration value="setEstimacion"/>
61 <xsd:enumeration value="getEstimacion"/>
62 <xsd:enumeration value="setQueja"/>
63 <xsd:enumeration value="getQueja"/>
64 <xsd:enumeration value="setProgreso"/>
65 <xsd:enumeration value="getProgreso"/>
66 <xsd:enumeration value="setEmergencia"/>
67 <xsd:enumeration value="getEmergencia"/>
68 <xsd:enumeration value="setAlmacen"/>
69 <xsd:enumeration value="getAlmacen"/>
70 <xsd:enumeration value="setCarga"/>
71 <xsd:enumeration value="getCarga"/>
72 <xsd:enumeration value="setCapacidad"/>
73 <xsd:enumeration value="getCapacidad"/>
74 <xsd:enumeration value="setLote"/>
75 <xsd:enumeration value="getLote"/>
76 <xsd:enumeration value="setTarea"/>
77 <xsd:enumeration value="getTarea"/>
78 <xsd:enumeration value="setInnovacion"/>
79 <xsd:enumeration value="getInnovacion"/>
80 </xsd:restriction>
81 </xsd:simpleType>
82 </xsd:attribute>
83 <xsd:attribute name="receipt" type="xsd:boolean" default="false"/>
84 </xsd:complexType>
85 </xsd:element>
86 <!--
87 error information (información de error)
88 -->

```

```

89 <xsd:complexType name="errorType" mixed="true">
90 <xsd:attribute name="code" type="xsd:string"/>
91 <xsd:attribute name="ID" type="xsd:IDREF"/>
92 <xsd:attribute name="severity" default="error">
93 <xsd:simpleType>
94 <xsd:restriction base="xsd:string">
95 <xsd:enumeration value="warning"/>
96 <xsd:enumeration value="error"/>
97 </xsd:restriction>
98 </xsd:simpleType>
99 </xsd:attribute>
100 </xsd:complexType>
101 <!--
102 profile data (información del perfil)
103 -->
104 <xsd:complexType name="profileType">
105 <xsd:sequence>
106 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
107 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
108 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
109 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
110 </xsd:sequence>
111 <xsd:attribute name="name" type="xsd:string"/>
112 <xsd:attribute name="author" type="xsd:string"/>
113 <xsd:attribute name="version" type="xsd:string"/>
114 <xsd:attribute name="create" type="xsd:dateTime"/>
115 <xsd:attribute name="expire" type="xsd:dateTime"/>
116 <xsd:attribute name="current" type="xsd:dateTime"/>
117 <xsd:attribute name="start" type="xsd:dateTime"/>
118 <xsd:attribute name="end" type="xsd:dateTime"/>
119 </xsd:complexType>
120 <!--
121 administrative elements (elementos administrativos)
122 -->
123 <xsd:complexType name="scaleType">
124 <xsd:sequence>
125 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
126 </xsd:sequence>
127 <xsd:attribute name="name" type="xsd:ID" use="required"/>
128 <xsd:attribute name="value" type="xsd:duration"/>
129 <xsd:attribute name="base" type="xsd:dateTime"/>
130 </xsd:complexType>
131 <xsd:complexType name="stoneType">
132 <xsd:sequence>
133 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
134 <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
135 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
136 </xsd:sequence>
137 <xsd:attribute name="name" type="xsd:string" use="required"/>
138 </xsd:complexType>
139 <!--
140 primitive elements (elementos primitivos)
141 -->
142 <xsd:complexType name="customerType">
143 <xsd:sequence>
144 <xsd:element name="query" type="pps:queryType" minOccurs="0"/>
145 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
146 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
147 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
148 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
149 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
150 </xsd:sequence>
151 <xsd:attribute name="name" type="xsd:ID" use="required"/>
152 <xsd:attribute name="ac">
153 <xsd:simpleType>
154 <xsd:restriction base="xsd:string">
155 <xsd:enumeration value="create"/>
156 <xsd:enumeration value="delete"/>
157 <xsd:enumeration value="revise"/>
158 </xsd:restriction>
159 </xsd:simpleType>
160 </xsd:attribute>
161 <xsd:attribute name="calculation">
162 <xsd:simpleType>
163 <xsd:restriction base="xsd:string">
164 <xsd:enumeration value="sum"/>
165 <xsd:enumeration value="ave"/>
166 <xsd:enumeration value="cnt"/>
167 <xsd:enumeration value="max"/>
168 <xsd:enumeration value="min"/>
169 <xsd:enumeration value="earliest"/>
170 <xsd:enumeration value="latest"/>
171 <xsd:enumeration value="shortest"/>
172 <xsd:enumeration value="longest"/>
173 </xsd:restriction>
174 </xsd:simpleType>
175 </xsd:attribute>
176 </xsd:complexType>
177 <xsd:complexType name="supplierType">
178 <xsd:sequence>

```

```

179 <xsd:element name="query" type="pps:queryType" minOccurs="0"/>
180 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
181 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
182 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
183 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
184 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
185 </xsd:sequence>
186 <xsd:attribute name="name" type="xsd:ID" use="required"/>
187 <xsd:attribute name="ac">
188 <xsd:simpleType>
189 <xsd:restriction base="xsd:string">
190 <xsd:enumeration value="create"/>
191 <xsd:enumeration value="delete"/>
192 <xsd:enumeration value="revise"/>
193 </xsd:restriction>
194 </xsd:simpleType>
195 </xsd:attribute>
196 <xsd:attribute name="calculation">
197 <xsd:simpleType>
198 <xsd:restriction base="xsd:string">
199 <xsd:enumeration value="sum"/>
200 <xsd:enumeration value="ave"/>
201 <xsd:enumeration value="cnt"/>
202 <xsd:enumeration value="max"/>
203 <xsd:enumeration value="min"/>
204 <xsd:enumeration value="earliest"/>
205 <xsd:enumeration value="latest"/>
206 <xsd:enumeration value="shortest"/>
207 <xsd:enumeration value="longest"/>
208 </xsd:restriction>
209 </xsd:simpleType>
210 </xsd:attribute>
211 </xsd:complexType>
212 <!--Item is produced or consumed by operation. To put it concretely, product, sub assembly,
213 work in process, parts, raw material, and material correspond to item-->
214 <!--Item es producido y consumido por operaci3n. Concretamente, producto, sub-ensamblado,
215 trabajo-en-proceso, partes, materia prima, y material correspondiente a item-->
216 <xsd:complexType name="itemType">
217 <xsd:sequence>
218 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
219 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
220 <xsd:element name="query" type="pps:queryType" minOccurs="0"/>
221 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
222 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
223 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
224 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
225 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
226 <xsd:element name="calendar" type="pps:calendarType" minOccurs="0" maxOccurs="unbounded"/>
227 <xsd:element name="capacity" type="pps:capacityType" minOccurs="0" maxOccurs="unbounded"/>
228 <xsd:element name="stock" type="pps:stockType" minOccurs="0" maxOccurs="unbounded"/>
229 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
230 <xsd:element name="partof" type="pps:partofType" minOccurs="0" maxOccurs="unbounded"/>
231 <xsd:element name="lotsize" type="pps:lotsizeType" minOccurs="0" maxOccurs="unbounded"/>
232 <xsd:element name="interval" type="pps:intervalType" minOccurs="0" maxOccurs="unbounded"/>
233 <xsd:element name="produced" type="pps:producedType" minOccurs="0" maxOccurs="unbounded"/>
234 <xsd:element name="consumed" type="pps:consumedType" minOccurs="0" maxOccurs="unbounded"/>
235 <xsd:element name="available" type="pps:availableType" minOccurs="0" maxOccurs="unbounded"/>
236 </xsd:sequence>
237 <xsd:attribute name="name" type="xsd:ID" use="required"/>
238 <xsd:attribute name="ac">
239 <xsd:simpleType>
240 <xsd:restriction base="xsd:string">
241 <xsd:enumeration value="create"/>
242 <xsd:enumeration value="delete"/>
243 <xsd:enumeration value="revise"/>
244 </xsd:restriction>
245 </xsd:simpleType>
246 </xsd:attribute>
247 <xsd:attribute name="calculation">
248 <xsd:simpleType>
249 <xsd:restriction base="xsd:string">
250 <xsd:enumeration value="sum"/>
251 <xsd:enumeration value="ave"/>
252 <xsd:enumeration value="cnt"/>
253 <xsd:enumeration value="max"/>
254 <xsd:enumeration value="min"/>
255 <xsd:enumeration value="earliest"/>
256 <xsd:enumeration value="latest"/>
257 <xsd:enumeration value="shortest"/>
258 <xsd:enumeration value="longest"/>
259 </xsd:restriction>
260 </xsd:simpleType>
261 </xsd:attribute>
262 <xsd:attribute name="master" type="xsd:boolean"/>
263 <xsd:attribute name="parent" type="xsd:IDREF"/>
264 <xsd:attribute name="type">
265 <xsd:simpleType>
266 <xsd:restriction base="xsd:string">
267 <xsd:enumeration value="product"/>
268 <xsd:enumeration value="material"/>

```

```

269     <xsd:enumeration value="phantom"/>
270     <xsd:enumeration value="module"/>
271     <xsd:enumeration value="recipe"/>
272   </xsd:restriction>
273 </xsd:simpleType>
274 </xsd:attribute>
275 </xsd:complexType>
276 <xsd:complexType name="resourceType">
277   <xsd:sequence>
278     <xsd:element name="ev" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
279     <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
280     <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
281     <xsd:element name="query" type="pps:queryType" minOccurs="0"/>
282     <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
283     <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
284     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
285     <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
286     <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
287     <xsd:element name="calendar" type="pps:calendarType" minOccurs="0" maxOccurs="unbounded"/>
288     <xsd:element name="capacity" type="pps:capacityType" minOccurs="0" maxOccurs="unbounded"/>
289     <xsd:element name="load" type="pps:loadType" minOccurs="0" maxOccurs="unbounded"/>
290     <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
291     <xsd:element name="partof" type="pps:partofType" minOccurs="0" maxOccurs="unbounded"/>
292     <xsd:element name="assigned" type="pps:assignedType" minOccurs="0" maxOccurs="unbounded"/>
293     <xsd:element name="task" type="pps:taskType" minOccurs="0" maxOccurs="unbounded"/>
294     <xsd:element name="tasksize" type="pps:tasksizetype" minOccurs="0" maxOccurs="unbounded"/>
295     <xsd:element name="changeover" type="pps:changeoverType" minOccurs="0" maxOccurs="unbounded"/>
296   </xsd:sequence>
297   <xsd:attribute name="name" type="xsd:ID" use="required"/>
298   <xsd:attribute name="ac">
299     <xsd:simpleType>
300       <xsd:restriction base="xsd:string">
301         <xsd:enumeration value="create"/>
302         <xsd:enumeration value="delete"/>
303         <xsd:enumeration value="revise"/>
304       </xsd:restriction>
305     </xsd:simpleType>
306   </xsd:attribute>
307   <xsd:attribute name="calculation">
308     <xsd:simpleType>
309       <xsd:restriction base="xsd:string">
310         <xsd:enumeration value="sum"/>
311         <xsd:enumeration value="ave"/>
312         <xsd:enumeration value="cnt"/>
313         <xsd:enumeration value="max"/>
314         <xsd:enumeration value="min"/>
315         <xsd:enumeration value="earliest"/>
316         <xsd:enumeration value="latest"/>
317         <xsd:enumeration value="shortest"/>
318         <xsd:enumeration value="longest"/>
319       </xsd:restriction>
320     </xsd:simpleType>
321   </xsd:attribute>
322   <xsd:attribute name="master" type="xsd:boolean"/>
323   <xsd:attribute name="parent" type="xsd:IDREF"/>
324   <xsd:attribute name="type">
325     <xsd:simpleType>
326       <xsd:restriction base="xsd:string">
327         <xsd:enumeration value="personnel"/>
328         <xsd:enumeration value="equipment"/>
329         <xsd:enumeration value="subsidiary"/>
330         <xsd:enumeration value="utility"/>
331       </xsd:restriction>
332     </xsd:simpleType>
333   </xsd:attribute>
334   <xsd:attribute name="multiplier" type="xsd:IDREF"/>
335 </xsd:complexType>
336 <xsd:complexType name="operationType">
337   <xsd:sequence>
338     <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
339     <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
340     <xsd:element name="query" type="pps:queryType" minOccurs="0" maxOccurs="unbounded"/>
341     <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
342     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
343     <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
344     <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
345     <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
346     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
347     <xsd:element name="start" type="pps:startType" minOccurs="0"/>
348     <xsd:element name="end" type="pps:endType" minOccurs="0"/>
349     <xsd:element name="event" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
350     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
351     <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
352     <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
353     <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
354     <xsd:element name="assign" type="pps:assignType" minOccurs="0" maxOccurs="unbounded"/>
355     <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
356     <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
357     <xsd:element name="partof" type="pps:partofType" minOccurs="0" maxOccurs="unbounded"/>
358     <xsd:element name="lotsize" type="pps:lotsizetype" minOccurs="0" maxOccurs="unbounded"/>

```

www.bdigital.ula.ve


```

359 <xsd:element name="tasksize" type="pps:tasksizeType" minOccurs="0" maxOccurs="unbounded"/>
360 <xsd:element name="changeover" type="pps:changeoverType" minOccurs="0" maxOccurs="unbounded"/>
361 <xsd:element name="interval" type="pps:intervalType" minOccurs="0" maxOccurs="unbounded"/>
362 </xsd:sequence>
363 <xsd:attribute name="name" type="xsd:ID" use="required"/>
364 <xsd:attribute name="ac">
365 <xsd:simpleType>
366 <xsd:restriction base="xsd:string">
367 <xsd:enumeration value="create"/>
368 <xsd:enumeration value="delete"/>
369 <xsd:enumeration value="revise"/>
370 </xsd:restriction>
371 </xsd:simpleType>
372 </xsd:attribute>
373 <xsd:attribute name="calculation">
374 <xsd:simpleType>
375 <xsd:restriction base="xsd:string">
376 <xsd:enumeration value="sum"/>
377 <xsd:enumeration value="ave"/>
378 <xsd:enumeration value="cnt"/>
379 <xsd:enumeration value="max"/>
380 <xsd:enumeration value="min"/>
381 <xsd:enumeration value="earliest"/>
382 <xsd:enumeration value="latest"/>
383 <xsd:enumeration value="shortest"/>
384 <xsd:enumeration value="longest"/>
385 </xsd:restriction>
386 </xsd:simpleType>
387 </xsd:attribute>
388 <xsd:attribute name="master" type="xsd:boolean"/>
389 <xsd:attribute name="parent" type="xsd:IDREF"/>
390 <xsd:attribute name="order" type="xsd:IDREF"/>
391 <xsd:attribute name="direction" default="forward">
392 <xsd:simpleType>
393 <xsd:restriction base="xsd:string">
394 <xsd:enumeration value="forward"/>
395 <xsd:enumeration value="backward"/>
396 </xsd:restriction>
397 </xsd:simpleType>
398 </xsd:attribute>
399 <xsd:attribute name="type" default="make">
400 <xsd:simpleType>
401 <xsd:restriction base="xsd:string">
402 <xsd:enumeration value="make"/>
403 <xsd:enumeration value="stock"/>
404 <xsd:enumeration value="move"/>
405 <xsd:enumeration value="check"/>
406 <xsd:enumeration value="care"/>
407 </xsd:restriction>
408 </xsd:simpleType>
409 </xsd:attribute>
410 </xsd:complexType>
411 <xsd:complexType name="orderType">
412 <xsd:sequence>
413 <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
414 <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
415 <xsd:element name="query" type="pps:queryType" minOccurs="0" maxOccurs="unbounded"/>
416 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
417 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
418 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
419 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
420 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
421 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
422 <xsd:element name="release" type="pps:releaseType" minOccurs="0"/>
423 <xsd:element name="duetime" type="pps:duetimeType" minOccurs="0"/>
424 <xsd:element name="progress" type="pps:progressType" minOccurs="0" maxOccurs="unbounded"/>
425 <xsd:element name="partof" type="pps:partofType" minOccurs="0" maxOccurs="unbounded"/>
426 <xsd:element name="pegging" type="pps:peggingType" minOccurs="0" maxOccurs="unbounded"/>
427 <xsd:element name="lotsize" type="pps:lotsizeType" minOccurs="0" maxOccurs="unbounded"/>
428 <xsd:element name="tasksize" type="pps:tasksizeType" minOccurs="0" maxOccurs="unbounded"/>
429 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
430 </xsd:sequence>
431 <xsd:attribute name="name" type="xsd:ID" use="required"/>
432 <xsd:attribute name="ac">
433 <xsd:simpleType>
434 <xsd:restriction base="xsd:string">
435 <xsd:enumeration value="create"/>
436 <xsd:enumeration value="delete"/>
437 <xsd:enumeration value="revise"/>
438 </xsd:restriction>
439 </xsd:simpleType>
440 </xsd:attribute>
441 <xsd:attribute name="calculation">
442 <xsd:simpleType>
443 <xsd:restriction base="xsd:string">
444 <xsd:enumeration value="sum"/>
445 <xsd:enumeration value="ave"/>
446 <xsd:enumeration value="cnt"/>
447 <xsd:enumeration value="max"/>
448 <xsd:enumeration value="min"/>

```

www.abdigital.ula.ve

```

449     <xsd:enumeration value="earliest"/>
450     <xsd:enumeration value="latest"/>
451     <xsd:enumeration value="shortest"/>
452     <xsd:enumeration value="longest"/>
453   </xsd:restriction>
454 </xsd:simpleType>
455 </xsd:attribute>
456 <xsd:attribute name="master" type="xsd:boolean"/>
457 <xsd:attribute name="parent" type="xsd:IDREF"/>
458 <xsd:attribute name="customer" type="xsd:IDREF"/>
459 <xsd:attribute name="supplier" type="xsd:IDREF"/>
460 <xsd:attribute name="item" type="xsd:IDREF"/>
461 <xsd:attribute name="resource" type="xsd:IDREF"/>
462 <xsd:attribute name="operation" type="xsd:IDREF"/>
463 <xsd:attribute name="status" default="fixed">
464   <xsd:simpleType>
465     <xsd:restriction base="xsd:string">
466       <xsd:enumeration value="fixed"/>
467       <xsd:enumeration value="forecast"/>
468       <xsd:enumeration value="unofficial"/>
469       <xsd:enumeration value="temporal"/>
470       <xsd:enumeration value="done"/>
471       <xsd:enumeration value="cancel"/>
472     </xsd:restriction>
473   </xsd:simpleType>
474 </xsd:attribute>
475 <xsd:attribute name="direction" default="forward">
476   <xsd:simpleType>
477     <xsd:restriction base="xsd:string">
478       <xsd:enumeration value="forward"/>
479       <xsd:enumeration value="backward"/>
480     </xsd:restriction>
481   </xsd:simpleType>
482 </xsd:attribute>
483 <xsd:attribute name="multiplier" type="xsd:IDREF"/>
484 <xsd:attribute name="type">
485   <xsd:simpleType>
486     <xsd:restriction base="xsd:string">
487       <xsd:enumeration value="customer"/>
488       <xsd:enumeration value="purchase"/>
489       <xsd:enumeration value="production"/>
490       <xsd:enumeration value="maintenance"/>
491       <xsd:enumeration value="design"/>
492       <xsd:enumeration value="work"/>
493     </xsd:restriction>
494   </xsd:simpleType>
495 </xsd:attribute>
496 </xsd:complexType>
497 <xsd:complexType name="lotType">
498   <xsd:sequence>
499     <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
500     <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
501     <xsd:element name="query" type="pps:queryType" minOccurs="0" maxOccurs="unbounded"/>
502     <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
503     <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
504     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
505     <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
506     <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
507     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
508     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
509     <xsd:element name="produced" type="pps:producedType" minOccurs="0" maxOccurs="unbounded"/>
510     <xsd:element name="consumed" type="pps:consumedType" minOccurs="0" maxOccurs="unbounded"/>
511     <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
512     <xsd:element name="tracking" type="pps:trackingType" minOccurs="0" maxOccurs="unbounded"/>
513   </xsd:sequence>
514   <xsd:attribute name="name" type="xsd:ID" use="required"/>
515   <xsd:attribute name="ac">
516     <xsd:simpleType>
517       <xsd:restriction base="xsd:string">
518         <xsd:enumeration value="create"/>
519         <xsd:enumeration value="delete"/>
520         <xsd:enumeration value="revise"/>
521       </xsd:restriction>
522     </xsd:simpleType>
523   </xsd:attribute>
524   <xsd:attribute name="calculation">
525     <xsd:simpleType>
526       <xsd:restriction base="xsd:string">
527         <xsd:enumeration value="sum"/>
528         <xsd:enumeration value="ave"/>
529         <xsd:enumeration value="cnt"/>
530         <xsd:enumeration value="max"/>
531         <xsd:enumeration value="min"/>
532         <xsd:enumeration value="earliest"/>
533         <xsd:enumeration value="latest"/>
534         <xsd:enumeration value="shortest"/>
535         <xsd:enumeration value="longest"/>
536       </xsd:restriction>
537     </xsd:simpleType>
538   </xsd:attribute>

```

```

539 <xsd:attribute name="item" type="xsd:IDREF"/>
540 <xsd:attribute name="order" type="xsd:IDREF"/>
541 <xsd:attribute name="type" default="produced">
542   <xsd:simpleType>
543     <xsd:restriction base="xsd:string">
544       <xsd:enumeration value="produced"/>
545       <xsd:enumeration value="consumed"/>
546     </xsd:restriction>
547   </xsd:simpleType>
548 </xsd:attribute>
549 </xsd:complexType>
550 <xsd:complexType name="taskType">
551   <xsd:sequence>
552     <xsd:element name="compose" type="pps:composeType" minOccurs="0" maxOccurs="unbounded"/>
553     <xsd:element name="composed" type="pps:composedType" minOccurs="0" maxOccurs="unbounded"/>
554     <xsd:element name="query" type="pps:queryType" minOccurs="0" maxOccurs="unbounded"/>
555     <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
556     <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
557     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
558     <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
559     <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
560     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
561     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
562     <xsd:element name="assigned" type="pps:assignedType" minOccurs="0" maxOccurs="unbounded"/>
563     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
564     <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
565     <xsd:element name="tracking" type="pps:trackingType" minOccurs="0" maxOccurs="unbounded"/>
566   </xsd:sequence>
567   <xsd:attribute name="name" type="xsd:ID" use="required"/>
568   <xsd:attribute name="ac">
569     <xsd:simpleType>
570       <xsd:restriction base="xsd:string">
571         <xsd:enumeration value="create"/>
572         <xsd:enumeration value="delete"/>
573         <xsd:enumeration value="revise"/>
574       </xsd:restriction>
575     </xsd:simpleType>
576   </xsd:attribute>
577   <xsd:attribute name="calculation">
578     <xsd:simpleType>
579       <xsd:restriction base="xsd:string">
580         <xsd:enumeration value="sum"/>
581         <xsd:enumeration value="ave"/>
582         <xsd:enumeration value="cnt"/>
583         <xsd:enumeration value="max"/>
584         <xsd:enumeration value="min"/>
585         <xsd:enumeration value="earliest"/>
586         <xsd:enumeration value="latest"/>
587         <xsd:enumeration value="shortest"/>
588         <xsd:enumeration value="longest"/>
589       </xsd:restriction>
590     </xsd:simpleType>
591   </xsd:attribute>
592   <xsd:attribute name="resource" type="xsd:IDREF"/>
593   <xsd:attribute name="order" type="xsd:IDREF"/>
594   <xsd:attribute name="multiplier" type="xsd:IDREF"/>
595   <xsd:attribute name="type" default="produced">
596     <xsd:simpleType>
597       <xsd:restriction base="xsd:string">
598         <xsd:enumeration value="produced"/>
599         <xsd:enumeration value="consumed"/>
600       </xsd:restriction>
601     </xsd:simpleType>
602   </xsd:attribute>
603 </xsd:complexType>
604 <!--

```

605 relational elements between primitives (elementos relacionales entre primitivos)

606 -->

```

607 <xsd:complexType name="composeType">
608   <xsd:sequence>
609     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
610   </xsd:sequence>
611   <xsd:attribute name="parent" type="xsd:IDREF"/>
612   <xsd:attribute name="type">
613     <xsd:simpleType>
614       <xsd:restriction base="xsd:string">
615         <xsd:enumeration value="and"/>
616         <xsd:enumeration value="or"/>
617       </xsd:restriction>
618     </xsd:simpleType>
619   </xsd:attribute>
620 </xsd:complexType>
621 <xsd:complexType name="composedType">
622   <xsd:sequence>
623     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
624   </xsd:sequence>
625   <xsd:attribute name="parent" type="xsd:IDREF"/>
626   <xsd:attribute name="type">
627     <xsd:simpleType>
628       <xsd:restriction base="xsd:string">

```



```
C:\629 <xsd:enumeration value="and"/>
630 <xsd:enumeration value="or"/>
631 </xsd:restriction>
632 </xsd:simpleType>
633 </xsd:attribute>
634 </xsd:complexType>
635 <xsd:complexType name="produceType">
636 <xsd:sequence>
637 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
638 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
639 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
640 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
641 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
642 </xsd:sequence>
643 <xsd:attribute name="type">
644 <xsd:simpleType>
645 <xsd:restriction base="xsd:string">
646 <xsd:enumeration value="start"/>
647 <xsd:enumeration value="end"/>
648 </xsd:restriction>
649 </xsd:simpleType>
650 </xsd:attribute>
651 <xsd:attribute name="timeType" default="disc">
652 <xsd:simpleType>
653 <xsd:restriction base="xsd:string">
654 <xsd:enumeration value="disc"/>
655 <xsd:enumeration value="cont"/>
656 </xsd:restriction>
657 </xsd:simpleType>
658 </xsd:attribute>
659 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
660 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
661 <xsd:attribute name="item" type="xsd:IDREF"/>
662 <xsd:attribute name="lot" type="xsd:IDREF"/>
663 </xsd:complexType>
664 <xsd:complexType name="producedType">
665 <xsd:sequence>
666 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
667 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
668 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
669 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
670 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
671 </xsd:sequence>
672 <xsd:attribute name="type">
673 <xsd:simpleType>
674 <xsd:restriction base="xsd:string">
675 <xsd:enumeration value="start"/>
676 <xsd:enumeration value="end"/>
677 </xsd:restriction>
678 </xsd:simpleType>
679 </xsd:attribute>
680 <xsd:attribute name="timeType" default="disc">
681 <xsd:simpleType>
682 <xsd:restriction base="xsd:string">
683 <xsd:enumeration value="disc"/>
684 <xsd:enumeration value="cont"/>
685 </xsd:restriction>
686 </xsd:simpleType>
687 </xsd:attribute>
688 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
689 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
690 <xsd:attribute name="operation" type="xsd:IDREF"/>
691 <xsd:attribute name="dispatch" type="xsd:IDREF"/>
692 </xsd:complexType>
693 <xsd:complexType name="consumeType">
694 <xsd:sequence>
695 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
696 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
697 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
698 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
699 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
700 </xsd:sequence>
701 <xsd:attribute name="type">
702 <xsd:simpleType>
703 <xsd:restriction base="xsd:string">
704 <xsd:enumeration value="start"/>
705 <xsd:enumeration value="end"/>
706 </xsd:restriction>
707 </xsd:simpleType>
708 </xsd:attribute>
709 <xsd:attribute name="timeType" default="disc">
710 <xsd:simpleType>
711 <xsd:restriction base="xsd:string">
712 <xsd:enumeration value="disc"/>
713 <xsd:enumeration value="cont"/>
714 </xsd:restriction>
715 </xsd:simpleType>
716 </xsd:attribute>
717 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
718 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
```

```

719 <xsd:attribute name="item" type="xsd:IDREF"/>
720 <xsd:attribute name="lot" type="xsd:IDREF"/>
721 </xsd:complexType>
722 <xsd:complexType name="consumedType">
723 <xsd:sequence>
724 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
725 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
726 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
727 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
728 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
729 </xsd:sequence>
730 <xsd:attribute name="type">
731 <xsd:simpleType>
732 <xsd:restriction base="xsd:string">
733 <xsd:enumeration value="start"/>
734 <xsd:enumeration value="end"/>
735 </xsd:restriction>
736 </xsd:simpleType>
737 </xsd:attribute>
738 <xsd:attribute name="timeType" default="disc">
739 <xsd:simpleType>
740 <xsd:restriction base="xsd:string">
741 <xsd:enumeration value="disc"/>
742 <xsd:enumeration value="cont"/>
743 </xsd:restriction>
744 </xsd:simpleType>
745 </xsd:attribute>
746 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
747 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
748 <xsd:attribute name="operation" type="xsd:IDREF"/>
749 <xsd:attribute name="dispatch" type="xsd:IDREF"/>
750 </xsd:complexType>
751 <xsd:complexType name="assignType">
752 <xsd:sequence>
753 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
754 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
755 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
756 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
757 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
758 </xsd:sequence>
759 <xsd:attribute name="type">
760 <xsd:simpleType>
761 <xsd:restriction base="xsd:string">
762 <xsd:enumeration value="start"/>
763 <xsd:enumeration value="end"/>
764 </xsd:restriction>
765 </xsd:simpleType>
766 </xsd:attribute>
767 <xsd:attribute name="timeType" default="disc">
768 <xsd:simpleType>
769 <xsd:restriction base="xsd:string">
770 <xsd:enumeration value="disc"/>
771 <xsd:enumeration value="cont"/>
772 </xsd:restriction>
773 </xsd:simpleType>
774 </xsd:attribute>
775 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
776 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
777 <xsd:attribute name="resource" type="xsd:IDREF"/>
778 <xsd:attribute name="task" type="xsd:IDREF"/>
779 </xsd:complexType>
780 <xsd:complexType name="assignedType">
781 <xsd:sequence>
782 <xsd:element name="location" type="pps:locationType" minOccurs="0" maxOccurs="unbounded"/>
783 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
784 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
785 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
786 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
787 </xsd:sequence>
788 <xsd:attribute name="type">
789 <xsd:simpleType>
790 <xsd:restriction base="xsd:string">
791 <xsd:enumeration value="start"/>
792 <xsd:enumeration value="end"/>
793 </xsd:restriction>
794 </xsd:simpleType>
795 </xsd:attribute>
796 <xsd:attribute name="timeType" default="disc">
797 <xsd:simpleType>
798 <xsd:restriction base="xsd:string">
799 <xsd:enumeration value="disc"/>
800 <xsd:enumeration value="cont"/>
801 </xsd:restriction>
802 </xsd:simpleType>
803 </xsd:attribute>
804 <xsd:attribute name="nth" type="xsd:integer" default="1"/>
805 <xsd:attribute name="sel" type="xsd:integer" default="1"/>
806 <xsd:attribute name="operation" type="xsd:IDREF"/>
807 <xsd:attribute name="dispatch" type="xsd:IDREF"/>
808 </xsd:complexType>

```

```
C:809 <xsd:complexType name="predecessorType">
810 <xsd:sequence>
811 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
812 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
813 </xsd:sequence>
814 <xsd:attribute name="type" default="FS">
815 <xsd:simpleType>
816 <xsd:restriction base="xsd:string">
817 <xsd:enumeration value="FS"/>
818 <xsd:enumeration value="FF"/>
819 <xsd:enumeration value="SS"/>
820 </xsd:restriction>
821 </xsd:simpleType>
822 </xsd:attribute>
823 <xsd:attribute name="stone" type="xsd:string"/>
824 <xsd:attribute name="operation" type="xsd:IDREF"/>
825 <xsd:attribute name="order" type="xsd:IDREF"/>
826 <xsd:attribute name="dispatch" type="xsd:IDREF"/>
827 <xsd:attribute name="item" type="xsd:IDREF"/>
828 <xsd:attribute name="lot" type="xsd:IDREF"/>
829 <xsd:attribute name="event" type="xsd:IDREF"/>
830 </xsd:complexType>
831 <xsd:complexType name="successorType">
832 <xsd:sequence>
833 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
834 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
835 </xsd:sequence>
836 <xsd:attribute name="type" default="FS">
837 <xsd:simpleType>
838 <xsd:restriction base="xsd:string">
839 <xsd:enumeration value="FS"/>
840 <xsd:enumeration value="FF"/>
841 <xsd:enumeration value="SS"/>
842 </xsd:restriction>
843 </xsd:simpleType>
844 </xsd:attribute>
845 <xsd:attribute name="stone" type="xsd:string"/>
846 <xsd:attribute name="operation" type="xsd:IDREF"/>
847 <xsd:attribute name="order" type="xsd:IDREF"/>
848 <xsd:attribute name="dispatch" type="xsd:IDREF"/>
849 <xsd:attribute name="item" type="xsd:IDREF"/>
850 <xsd:attribute name="lot" type="xsd:IDREF"/>
851 <xsd:attribute name="event" type="xsd:IDREF"/>
852 </xsd:complexType>
853 <xsd:complexType name="partofType">
854 <xsd:sequence>
855 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
856 </xsd:sequence>
857 <xsd:attribute name="operation" type="xsd:IDREFS"/>
858 <xsd:attribute name="item" type="xsd:IDREFS"/>
859 <xsd:attribute name="resource" type="xsd:IDREFS"/>
860 <xsd:attribute name="order" type="xsd:IDREFS"/>
861 <xsd:attribute name="customer" type="xsd:IDREFS"/>
862 <xsd:attribute name="supplier" type="xsd:IDREFS"/>
863 </xsd:complexType>
864 <xsd:complexType name="peggingType">
865 <xsd:sequence>
866 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
867 </xsd:sequence>
868 <xsd:attribute name="order" type="xsd:IDREF" use="required"/>
869 <xsd:attribute name="type" default="produce">
870 <xsd:simpleType>
871 <xsd:restriction base="xsd:string">
872 <xsd:enumeration value="produce"/>
873 <xsd:enumeration value="consume"/>
874 <xsd:enumeration value="upper"/>
875 <xsd:enumeration value="lower"/>
876 </xsd:restriction>
877 </xsd:simpleType>
878 </xsd:attribute>
879 <xsd:attribute name="final" type="xsd:boolean" default="false"/>
880 </xsd:complexType>
881 <xsd:complexType name="trackingType">
882 <xsd:sequence>
883 <xsd:element name="qty" type="pps:qtyType"/>
884 </xsd:sequence>
885 <xsd:attribute name="lot" type="xsd:IDREF"/>
886 <xsd:attribute name="task" type="xsd:IDREF"/>
887 </xsd:complexType>
888 <!--
889 property elements of primitives (elementos propiedades de primitivos)
890 -->
891 <xsd:complexType name="locationType">
892 <xsd:sequence>
893 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
894 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
895 <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
896 </xsd:sequence>
897 <xsd:attribute name="name" use="required">
898 <xsd:simpleType>
```

www.bdigital.ula.ve

C. 899

900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988

```
<xsd:restriction base="xsd:string">  
  <xsd:enumeration value="address"/>  
  <xsd:enumeration value="code"/>  
  <xsd:enumeration value="site"/>  
  <xsd:enumeration value="area"/>  
  <xsd:enumeration value="row"/>  
  <xsd:enumeration value="column"/>  
</xsd:restriction>  
</xsd:simpleType>  
</xsd:attribute>  
<xsd:attribute name="timeType" default="disc">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="disc"/>  
      <xsd:enumeration value="cont"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:attribute>  
<xsd:attribute name="relative" default="false">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="true"/>  
      <xsd:enumeration value="false"/>  
      <xsd:enumeration value="relative-gradual"/>  
      <xsd:enumeration value="absolute-gradual"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:attribute>  
</xsd:complexType>  
<xsd:complexType name="shiftType">  
  <xsd:sequence>  
    <xsd:element name="capacity" type="pps:capacityType" maxOccurs="unbounded"/>  
  </xsd:sequence>  
  <xsd:attribute name="name" type="xsd:ID" use="required"/>  
</xsd:complexType>  
<xsd:complexType name="calendarType">  
  <xsd:sequence>  
    <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>  
    <xsd:element name="char" minOccurs="0">  
      <xsd:simpleType>  
        <xsd:restriction base="xsd:string">  
          <xsd:enumeration value="on"/>  
          <xsd:enumeration value="off"/>  
          <xsd:enumeration value="extend"/>  
          <xsd:enumeration value="1-shift"/>  
          <xsd:enumeration value="2-shift"/>  
          <xsd:enumeration value="3-shift"/>  
        </xsd:restriction>  
      </xsd:simpleType>  
    </xsd:element>  
  </xsd:sequence>  
  <xsd:attribute name="shift" type="xsd:IDREF" use="required"/>  
  <xsd:attribute name="time" type="xsd:dateTime" use="required"/>  
  <xsd:attribute name="relative" default="false">  
    <xsd:simpleType>  
      <xsd:restriction base="xsd:string">  
        <xsd:enumeration value="true"/>  
        <xsd:enumeration value="false"/>  
        <xsd:enumeration value="relative-gradual"/>  
        <xsd:enumeration value="absolute-gradual"/>  
      </xsd:restriction>  
    </xsd:simpleType>  
  </xsd:attribute>  
</xsd:complexType>  
<xsd:complexType name="availableType">  
  <xsd:sequence>  
    <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>  
    <xsd:element name="time" type="pps:timeType" minOccurs="0"/>  
  </xsd:sequence>  
  <xsd:attribute name="relative" default="false">  
    <xsd:simpleType>  
      <xsd:restriction base="xsd:string">  
        <xsd:enumeration value="true"/>  
        <xsd:enumeration value="false"/>  
        <xsd:enumeration value="relative-gradual"/>  
        <xsd:enumeration value="absolute-gradual"/>  
      </xsd:restriction>  
    </xsd:simpleType>  
  </xsd:attribute>  
</xsd:complexType>  
<xsd:complexType name="progressType">  
  <xsd:sequence>  
    <xsd:element name="time" type="pps:timeType" minOccurs="0"/>  
    <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>  
  </xsd:sequence>  
  <xsd:attribute name="name" use="optional">  
    <xsd:simpleType>  
      <xsd:restriction base="xsd:string">  
        <xsd:enumeration value="rate"/>  
        <xsd:enumeration value="yield"/>  
      </xsd:restriction>  
    </xsd:simpleType>
```



```

989     <xsd:enumeration value="crap"/>
990     </xsd:restriction>
991 </xsd:simpleType>
992 </xsd:attribute>
993 <xsd:attribute name="status" use="optional">
994   <xsd:simpleType>
995     <xsd:restriction base="xsd:string">
996       <xsd:enumeration value="completed"/>
997       <xsd:enumeration value="canceled"/>
998       <xsd:enumeration value="suspended"/>
999       <xsd:enumeration value="started"/>
1000      <xsd:enumeration value="ready"/>
1001      <xsd:enumeration value="created"/>
1002    </xsd:restriction>
1003  </xsd:simpleType>
1004 </xsd:attribute>
1005 <xsd:attribute name="relative" default="false">
1006   <xsd:simpleType>
1007     <xsd:restriction base="xsd:string">
1008       <xsd:enumeration value="true"/>
1009       <xsd:enumeration value="false"/>
1010       <xsd:enumeration value="relative-gradual"/>
1011       <xsd:enumeration value="absolute-gradual"/>
1012     </xsd:restriction>
1013   </xsd:simpleType>
1014 </xsd:attribute>
1015 </xsd:complexType>
1016 <xsd:complexType name="stockType">
1017   <xsd:sequence>
1018     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
1019     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1020   </xsd:sequence>
1021   <xsd:attribute name="timeType" default="disc">
1022     <xsd:simpleType>
1023       <xsd:restriction base="xsd:string">
1024         <xsd:enumeration value="disc"/>
1025         <xsd:enumeration value="cont"/>
1026       </xsd:restriction>
1027     </xsd:simpleType>
1028   </xsd:attribute>
1029   <xsd:attribute name="relative" default="false">
1030     <xsd:simpleType>
1031       <xsd:restriction base="xsd:string">
1032         <xsd:enumeration value="true"/>
1033         <xsd:enumeration value="false"/>
1034         <xsd:enumeration value="relative-gradual"/>
1035         <xsd:enumeration value="absolute-gradual"/>
1036       </xsd:restriction>
1037     </xsd:simpleType>
1038   </xsd:attribute>
1039 </xsd:complexType>
1040 <xsd:complexType name="loadType">
1041   <xsd:sequence>
1042     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
1043     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1044   </xsd:sequence>
1045   <xsd:attribute name="timeType" default="disc">
1046     <xsd:simpleType>
1047       <xsd:restriction base="xsd:string">
1048         <xsd:enumeration value="disc"/>
1049         <xsd:enumeration value="cont"/>
1050       </xsd:restriction>
1051     </xsd:simpleType>
1052   </xsd:attribute>
1053   <xsd:attribute name="relative" default="false">
1054     <xsd:simpleType>
1055       <xsd:restriction base="xsd:string">
1056         <xsd:enumeration value="true"/>
1057         <xsd:enumeration value="false"/>
1058         <xsd:enumeration value="relative-gradual"/>
1059         <xsd:enumeration value="absolute-gradual"/>
1060       </xsd:restriction>
1061     </xsd:simpleType>
1062   </xsd:attribute>
1063 </xsd:complexType>
1064 <xsd:complexType name="specType">
1065   <xsd:sequence>
1066     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1067     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
1068     <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
1069     <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
1070   </xsd:sequence>
1071   <xsd:attribute name="name" type="xsd:string" use="required"/>
1072   <xsd:attribute name="timeType" default="disc">
1073     <xsd:simpleType>
1074       <xsd:restriction base="xsd:string">
1075         <xsd:enumeration value="disc"/>
1076         <xsd:enumeration value="cont"/>
1077       </xsd:restriction>
1078     </xsd:simpleType>

```

www.bdigital.ula.ve

C
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168

```
</xsd:attribute>  
<xsd:attribute name="relative" default="false">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="true"/>  
      <xsd:enumeration value="false"/>  
      <xsd:enumeration value="relative-gradual"/>  
      <xsd:enumeration value="absolute-gradual"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:attribute>  
</xsd:complexType>  
<xsd:complexType name="capacityType">  
  <xsd:sequence>  
    <xsd:element name="time" type="pps:timeType"/>  
    <xsd:element name="qty" type="pps:qtyType"/>  
  </xsd:sequence>  
  <xsd:attribute name="timeType" default="disc">  
    <xsd:simpleType>  
      <xsd:restriction base="xsd:string">  
        <xsd:enumeration value="disc"/>  
        <xsd:enumeration value="cont"/>  
      </xsd:restriction>  
    </xsd:simpleType>  
  </xsd:attribute>  
  <xsd:attribute name="direccion" default="upper">  
    <xsd:simpleType>  
      <xsd:restriction base="xsd:string">  
        <xsd:enumeration value="upper"/>  
        <xsd:enumeration value="lower"/>  
      </xsd:restriction>  
    </xsd:simpleType>  
  </xsd:attribute>  
<xsd:attribute name="relative" default="false">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="true"/>  
      <xsd:enumeration value="false"/>  
      <xsd:enumeration value="relative-gradual"/>  
      <xsd:enumeration value="absolute-gradual"/>  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:attribute>  
</xsd:complexType>  
<!--  
temporal events on primitives (eventos temporales en primitivos)  
-->  
<xsd:complexType name="startType">  
  <xsd:sequence>  
    <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>  
    <xsd:element name="display" type="pps:displayType" minOccurs="0"/>  
    <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>  
    <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="price" type="pps:priceType" minOccurs="0"/>  
    <xsd:element name="time" type="pps:timeType" minOccurs="0"/>  
    <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="action" type="pps:actionType" minOccurs="0" maxOccurs="unbounded"/>  
  </xsd:sequence>  
</xsd:complexType>  
<xsd:complexType name="endType">  
  <xsd:sequence>  
    <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>  
    <xsd:element name="display" type="pps:displayType" minOccurs="0"/>  
    <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>  
    <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="price" type="pps:priceType" minOccurs="0"/>  
    <xsd:element name="time" type="pps:timeType" minOccurs="0"/>  
    <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="action" type="pps:actionType" minOccurs="0" maxOccurs="unbounded"/>  
  </xsd:sequence>  
</xsd:complexType>  
<xsd:complexType name="releaseType">  
  <xsd:sequence>  
    <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>  
    <xsd:element name="display" type="pps:displayType" minOccurs="0"/>  
    <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>  
    <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="price" type="pps:priceType" minOccurs="0"/>  
    <xsd:element name="time" type="pps:timeType" minOccurs="0"/>  
    <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>  
    <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
```

www.bdigital.ula.ve

```

1169 <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
1170 <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>
1171 <xsd:element name="action" type="pps:actionType" minOccurs="0" maxOccurs="unbounded"/>
1172 </xsd:sequence>
1173 </xsd:complexType>
1174 <xsd:complexType name="duetimeType">
1175 <xsd:sequence>
1176 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
1177 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
1178 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1179 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
1180 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
1181 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
1182 <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
1183 <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
1184 <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
1185 <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
1186 <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>
1187 <xsd:element name="action" type="pps:actionType" minOccurs="0" maxOccurs="unbounded"/>
1188 </xsd:sequence>
1189 </xsd:complexType>
1190 <xsd:complexType name="eventType">
1191 <xsd:sequence>
1192 <xsd:element name="query" type="pps:queryType" minOccurs="0" maxOccurs="unbounded"/>
1193 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
1194 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
1195 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1196 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
1197 <xsd:element name="price" type="pps:priceType" minOccurs="0"/>
1198 <xsd:element name="time" type="pps:timeType" minOccurs="0"/>
1199 <xsd:element name="predecessor" type="pps:predecessorType" minOccurs="0" maxOccurs="unbounded"/>
1200 <xsd:element name="successor" type="pps:successorType" minOccurs="0" maxOccurs="unbounded"/>
1201 <xsd:element name="produce" type="pps:produceType" minOccurs="0" maxOccurs="unbounded"/>
1202 <xsd:element name="consume" type="pps:consumeType" minOccurs="0" maxOccurs="unbounded"/>
1203 <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>
1204 <xsd:element name="action" type="pps:actionType" minOccurs="0" maxOccurs="unbounded"/>
1205 </xsd:sequence>
1206 <xsd:attribute name="name" type="xsd:ID" use="required"/>
1207 <xsd:attribute name="pre_name">
1208 <xsd:simpleType>
1209 <xsd:restriction base="xsd:string">
1210 <xsd:enumeration value="run"/>
1211 <xsd:enumeration value="complete"/>
1212 <xsd:enumeration value="terminate-n"/>
1213 <xsd:enumeration value="restart-n"/>
1214 <xsd:enumeration value="arrive"/>
1215 <xsd:enumeration value="leave"/>
1216 </xsd:restriction>
1217 </xsd:simpleType>
1218 </xsd:attribute>
1219 <xsd:attribute name="ac">
1220 <xsd:simpleType>
1221 <xsd:restriction base="xsd:string">
1222 <xsd:enumeration value="create"/>
1223 <xsd:enumeration value="delete"/>
1224 <xsd:enumeration value="revise"/>
1225 </xsd:restriction>
1226 </xsd:simpleType>
1227 </xsd:attribute>
1228 <xsd:attribute name="calculation">
1229 <xsd:simpleType>
1230 <xsd:restriction base="xsd:string">
1231 <xsd:enumeration value="sum"/>
1232 <xsd:enumeration value="ave"/>
1233 <xsd:enumeration value="cnt"/>
1234 <xsd:enumeration value="min"/>
1235 <xsd:enumeration value="max"/>
1236 <xsd:enumeration value="earliest"/>
1237 <xsd:enumeration value="latest"/>
1238 <xsd:enumeration value="shortest"/>
1239 <xsd:enumeration value="longest"/>
1240 </xsd:restriction>
1241 </xsd:simpleType>
1242 </xsd:attribute>
1243 <xsd:attribute name="master" type="xsd:boolean"/>
1244 <xsd:attribute name="parent" type="xsd:IDREF"/>
1245 <xsd:attribute name="resource" type="xsd:IDREF"/>
1246 </xsd:complexType>
1247 <!--
1248 basic data elements (elementos información o data básica)
1249 -->
1250 <xsd:complexType name="qtyType">
1251 <xsd:sequence>
1252 <xsd:element name="min" type="pps:minType" minOccurs="0"/>
1253 <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
1254 <xsd:element name="base" type="pps:baseType" minOccurs="0"/>
1255 </xsd:sequence>
1256 <xsd:attribute name="value" type="xsd:double"/>
1257 <xsd:attribute name="unit" type="xsd:IDREF"/>
1258 </xsd:complexType>

```

```

1259 <xsd:complexType name="priceType">
1260   <xsd:sequence>
1261     <xsd:element name="min" type="pps:minType" minOccurs="0"/>
1262     <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
1263     <xsd:element name="base" type="pps:baseType" minOccurs="0"/>
1264   </xsd:sequence>
1265   <xsd:attribute name="value" type="xsd:double"/>
1266   <xsd:attribute name="unit" type="xsd:IDREF"/>
1267 </xsd:complexType>
1268 <xsd:complexType name="charType">
1269   <xsd:sequence>
1270     <xsd:element name="enumerate" type="pps:enumerateType" minOccurs="0" maxOccurs="unbounded"/>
1271   </xsd:sequence>
1272   <xsd:attribute name="value" type="xsd:string"/>
1273 </xsd:complexType>
1274 <xsd:complexType name="addressType">
1275   <xsd:sequence>
1276     <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
1277     <xsd:element name="display" type="pps:displayType" minOccurs="0" maxOccurs="unbounded"/>
1278     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1279     <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
1280   </xsd:sequence>
1281   <xsd:attribute name="value" type="xsd:string"/>
1282   <xsd:attribute name="uri" type="xsd:anyURI"/>
1283   <xsd:attribute name="east" type="xsd:double"/>
1284   <xsd:attribute name="south" type="xsd:double"/>
1285   <xsd:attribute name="height" type="xsd:double"/>
1286 </xsd:complexType>
1287 <xsd:complexType name="durationType">
1288   <xsd:sequence>
1289     <xsd:element name="min" type="pps:minType" minOccurs="0"/>
1290     <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
1291     <xsd:element name="shortest" type="pps:shortestType" minOccurs="0"/>
1292     <xsd:element name="longest" type="pps:longestType" minOccurs="0"/>
1293   </xsd:sequence>
1294   <xsd:attribute name="value" type="xsd:duration"/>
1295   <xsd:attribute name="count" type="xsd:long"/>
1296   <xsd:attribute name="unit" type="xsd:IDREF"/>
1297   <xsd:attribute name="type" default="longer"/>
1298   <xsd:simpleType>
1299     <xsd:restriction base="xsd:string">
1300       <xsd:enumeration value="fixed"/>
1301       <xsd:enumeration value="longer"/>
1302       <xsd:enumeration value="shorter"/>
1303     </xsd:restriction>
1304   </xsd:simpleType>
1305 </xsd:attribute>
1306   <xsd:attribute name="net" type="xsd:boolean"/>
1307 </xsd:complexType>
1308 <xsd:complexType name="timeType">
1309   <xsd:sequence>
1310     <xsd:element name="min" type="pps:minType" minOccurs="0"/>
1311     <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
1312     <xsd:element name="earliest" type="pps:earliestType" minOccurs="0"/>
1313     <xsd:element name="latest" type="pps:latestType" minOccurs="0"/>
1314   </xsd:sequence>
1315   <xsd:attribute name="value" type="xsd:dateTime"/>
1316   <xsd:attribute name="count" type="xsd:long"/>
1317   <xsd:attribute name="timeScale" type="xsd:IDREF"/>
1318   <xsd:attribute name="stone" type="xsd:string"/>
1319   <xsd:attribute name="ref">
1320     <xsd:simpleType>
1321       <xsd:restriction base="xsd:string">
1322         <xsd:enumeration value="init"/>
1323         <xsd:enumeration value="final"/>
1324         <xsd:enumeration value="now"/>
1325       </xsd:restriction>
1326     </xsd:simpleType>
1327   </xsd:attribute>
1328 </xsd:complexType>
1329 <!--
1330 administrative properties (propiedades administrativas)
1331 -->
1332 <xsd:complexType name="priorityType">
1333   <xsd:choice>
1334     <xsd:element name="qty" minOccurs="0" maxOccurs="unbounded">
1335       <xsd:complexType>
1336         <xsd:sequence>
1337           <xsd:element name="min" type="pps:minType" minOccurs="0"/>
1338           <xsd:element name="max" type="pps:maxType" minOccurs="0"/>
1339         </xsd:sequence>
1340         <xsd:attribute name="value">
1341           <xsd:simpleType>
1342             <xsd:restriction base="xsd:double">
1343               <xsd:minInclusive value="-1"/>
1344               <xsd:maxInclusive value="1"/>
1345             </xsd:restriction>
1346           </xsd:simpleType>
1347         </xsd:attribute>
1348       </xsd:complexType>

```



```

1349 </xsd:element>
1350 <xsd:element name="char" minOccurs="0" maxOccurs="unbounded">
1351 <xsd:complexType>
1352 <xsd:attribute name="value">
1353 <xsd:simpleType>
1354 <xsd:restriction base="xsd:ID">
1355 <xsd:enumeration value="before"/>
1356 <xsd:enumeration value="after"/>
1357 </xsd:restriction>
1358 </xsd:simpleType>
1359 </xsd:attribute>
1360 </xsd:complexType>
1361 </xsd:element>
1362 </xsd:choice>
1363 </xsd:complexType>
1364 <xsd:complexType name="displayType">
1365 <xsd:sequence>
1366 <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1367 <xsd:element name="char" type="pps:charType" minOccurs="0"/>
1368 </xsd:sequence>
1369 <xsd:attribute name="name" type="xsd:string" use="required"/>
1370 </xsd:complexType>
1371 <xsd:complexType name="descriptionType">
1372 <xsd:sequence>
1373 <xsd:element name="description" type="xsd:string"/>
1374 </xsd:sequence>
1375 </xsd:complexType>
1376 <!--
1377 information for querying data (información para la solicitud de información o data)
1378 -->
1379 <xsd:complexType name="queryType">
1380 <xsd:attribute name="select" type="xsd:string"/>
1381 <xsd:attribute name="pre_select">
1382 <xsd:simpleType>
1383 <xsd:restriction base="xsd:string">
1384 <xsd:enumeration value="produce"/>
1385 <xsd:enumeration value="consume"/>
1386 <xsd:enumeration value="tracking"/>
1387 <xsd:enumeration value="pegging"/>
1388 <xsd:enumeration value="predecessor"/>
1389 <xsd:enumeration value="sucesor"/>
1390 <xsd:enumeration value="partof"/>
1391 <xsd:enumeration value="stock"/>
1392 <!--manufacturing BOM (BOM de manufactura)-->
1393 <!--tracking (rastreado)-->
1394 <!--pegging (marcado)-->
1395 <!--precedence (precedencia)-->
1396 <!--parent-children (padre-hijos)-->
1397 <!--stock (almacén)-->
1398 </xsd:restriction>
1399 </xsd:simpleType>
1400 </xsd:attribute>
1401 <xsd:attribute name="depth" type="xsd:int" default="0"/>
1402 <xsd:attribute name="start" type="xsd:dateTime"/>
1403 <xsd:attribute name="end" type="xsd:dateTime"/>
1404 <xsd:attribute name="calculation" default="end">
1405 <xsd:simpleType>
1406 <xsd:restriction base="xsd:string">
1407 <xsd:enumeration value="start"/>
1408 <xsd:enumeration value="end"/>
1409 <xsd:enumeration value="sum"/>
1410 <xsd:enumeration value="ave"/>
1411 <xsd:enumeration value="min"/>
1412 <xsd:enumeration value="max"/>
1413 <xsd:enumeration value="cnt"/>
1414 </xsd:restriction>
1415 </xsd:simpleType>
1416 </xsd:attribute>
1417 <xsd:attribute name="scale" type="xsd:IDREF"/>
1418 </xsd:complexType>
1419 <!--
1420 data auxiliary elements for query (elementos auxiliares de información o data para solicitud)
1421 -->
1422 <xsd:complexType name="minType">
1423 <xsd:attribute name="value" type="xsd:double" use="required"/>
1424 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1425 </xsd:complexType>
1426 <xsd:complexType name="maxType">
1427 <xsd:attribute name="value" type="xsd:double" use="required"/>
1428 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1429 </xsd:complexType>
1430 <xsd:complexType name="earliestType">
1431 <xsd:attribute name="value" type="xsd:dateTime" use="required"/>
1432 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1433 </xsd:complexType>
1434 <xsd:complexType name="latestType">
1435 <xsd:attribute name="value" type="xsd:dateTime" use="required"/>
1436 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1437 </xsd:complexType>
1438 <xsd:complexType name="shortestType">

```

www.bdigital.ula.ve

```

1439 <xsd:attribute name="value" type="xsd:duration" use="required"/>
1440 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1441 </xsd:complexType>
1442 <xsd:complexType name="longestType">
1443 <xsd:attribute name="value" type="xsd:duration" use="required"/>
1444 <xsd:attribute name="exclusive" type="xsd:boolean" default="false"/>
1445 </xsd:complexType>
1446 <xsd:complexType name="enumerateType">
1447 <xsd:attribute name="value" type="xsd:string" use="required"/>
1448 <xsd:attribute name="exclusive" type="xsd:string" default="no"/>
1449 </xsd:complexType>
1450 <!--
1451 data related to unit (Información o data relacionada a unidad)
1452 -->
1453 <xsd:complexType name="unitType">
1454 <xsd:sequence>
1455 <xsd:element name="translate" type="pps:translateType" minOccurs="0" maxOccurs="unbounded"/>
1456 </xsd:sequence>
1457 <xsd:attribute name="pre-det-name">
1458 <xsd:annotation>
1459 <xsd:documentation>
1460 Es: Si se desea agregar otro valor predeterminado de "id de unidad" simplemente utilice otra "enumeración"
1461 En: If you want to add another "unit value id", simply use another "enumeration" value
1462 </xsd:documentation>
1463 </xsd:annotation>
1464 <xsd:simpleType>
1465 <xsd:restriction base="xsd:ID">
1466 <xsd:enumeration value="sec"/>
1467 <xsd:enumeration value="min"/>
1468 <xsd:enumeration value="half-hour"/>
1469 <xsd:enumeration value="hour"/>
1470 <xsd:enumeration value="day"/>
1471 <xsd:enumeration value="week"/>
1472 <xsd:enumeration value="ten_d"/>
1473 <xsd:enumeration value="fifteen_d"/>
1474 <xsd:enumeration value="month"/>
1475 <xsd:enumeration value="year"/>
1476 <xsd:enumeration value="US_Dollar"/>
1477 <xsd:enumeration value="CN_Dollar"/>
1478 <xsd:enumeration value="Yen"/>
1479 <xsd:enumeration value="Bs"/>
1480 <xsd:enumeration value="Pnd"/>
1481 <xsd:enumeration value="Chelines"/>
1482 <xsd:enumeration value="EUR"/>
1483 <xsd:enumeration value="Pesos"/>
1484 <xsd:enumeration value="piece"/>
1485 <xsd:enumeration value="person"/>
1486 <xsd:enumeration value="degree"/>
1487 <xsd:enumeration value="degree_C"/>
1488 </xsd:restriction>
1489 </xsd:simpleType>
1490 <xsd:attribute>
1491 <xsd:attribute name="name" type="xsd:string" default="min"/>
1492 </xsd:complexType>
1493 <xsd:complexType name="translateType">
1494 <xsd:sequence>
1495 <xsd:element name="base" type="pps:baseType" minOccurs="0"/>
1496 </xsd:sequence>
1497 <xsd:attribute name="unit" type="xsd:IDREF" use="required"/>
1498 <xsd:attribute name="value" type="xsd:double" use="required"/>
1499 <xsd:attribute name="origin" type="xsd:double"/>
1500 </xsd:complexType>
1501 <xsd:complexType name="timeScaleConversionType">
1502 <xsd:attribute name="name" type="xsd:ID" use="required"/>
1503 <xsd:attribute name="unit" type="xsd:IDREF"/>
1504 <xsd:attribute name="value" default="1">
1505 <xsd:simpleType>
1506 <xsd:restriction base="xsd:double">
1507 <xsd:minExclusive value="0"/>
1508 </xsd:restriction>
1509 </xsd:simpleType>
1510 </xsd:attribute>
1511 <xsd:attribute name="origin" type="xsd:dateTime"/>
1512 <xsd:attribute name="type" default="disc">
1513 <xsd:simpleType>
1514 <xsd:restriction base="xsd:string">
1515 <xsd:enumeration value="disc"/>
1516 <xsd:enumeration value="cont"/>
1517 </xsd:restriction>
1518 </xsd:simpleType>
1519 </xsd:attribute>
1520 </xsd:complexType>
1521 <xsd:complexType name="baseType">
1522 <xsd:attribute name="value" type="xsd:double"/>
1523 </xsd:complexType>
1524 <!--
1525 definition of extended constraint data (definición de información o data de restricción extendida)
1526 -->
1527 <xsd:complexType name="lotsizeType">
1528 <xsd:sequence>

```

```

1529     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1530 </xsd:sequence>
1531 <xsd:attribute name="unit" type="xsd:IDREF"/>
1532 <xsd:attribute name="type" default="fixed">
1533   <xsd:simpleType>
1534     <xsd:restriction base="xsd:string">
1535       <xsd:enumeration value="fixed"/>
1536       <xsd:enumeration value="min"/>
1537       <xsd:enumeration value="max"/>
1538       <xsd:enumeration value="unit"/>
1539     </xsd:restriction>
1540   </xsd:simpleType>
1541 </xsd:attribute>
1542 <xsd:attribute name="value" type="xsd:double"/>
1543 </xsd:complexType>
1544 <xsd:complexType name="tasksizeType">
1545   <xsd:sequence>
1546     <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1547   </xsd:sequence>
1548   <xsd:attribute name="unit" type="xsd:IDREF"/>
1549   <xsd:attribute name="type" default="fixed">
1550     <xsd:simpleType>
1551       <xsd:restriction base="xsd:string">
1552         <xsd:enumeration value="fixed"/>
1553         <xsd:enumeration value="min"/>
1554         <xsd:enumeration value="max"/>
1555         <xsd:enumeration value="unit"/>
1556       </xsd:restriction>
1557     </xsd:simpleType>
1558   </xsd:attribute>
1559   <xsd:attribute name="value" type="xsd:double"/>
1560 </xsd:complexType>
1561 <xsd:complexType name="conditionType">
1562   <xsd:sequence>
1563     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
1564     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1565     <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
1566   </xsd:sequence>
1567   <xsd:attribute name="item" type="xsd:IDREF"/>
1568   <xsd:attribute name="resource" type="xsd:IDREF"/>
1569   <xsd:attribute name="pre_spec">
1570     <xsd:simpleType>
1571       <xsd:restriction base="xsd:string">
1572         <xsd:enumeration value="stock"/>
1573         <xsd:enumeration value="load"/>
1574         <xsd:enumeration value="capacity"/>
1575         <xsd:enumeration value="progress"/>
1576         <xsd:enumeration value="location"/>
1577       </xsd:restriction>
1578     </xsd:simpleType>
1579   </xsd:attribute>
1580   <xsd:attribute name="spec" type="xsd:string"/>
1581   <xsd:attribute name="type" default="NE">
1582     <xsd:simpleType>
1583       <xsd:restriction base="xsd:string">
1584         <xsd:enumeration value="EQ"/>
1585         <xsd:enumeration value="NE"/>
1586         <xsd:enumeration value="GT"/>
1587         <xsd:enumeration value="GE"/>
1588         <xsd:enumeration value="LT"/>
1589         <xsd:enumeration value="LE"/>
1590       </xsd:restriction>
1591     </xsd:simpleType>
1592   </xsd:attribute>
1593   <xsd:attribute name="direction" default="cmb">
1594     <xsd:simpleType>
1595       <xsd:restriction base="xsd:string">
1596         <xsd:enumeration value="pre"/>
1597         <xsd:enumeration value="suc"/>
1598         <xsd:enumeration value="cmb"/>
1599       </xsd:restriction>
1600     </xsd:simpleType>
1601   </xsd:attribute>
1602 </xsd:complexType>
1603 <xsd:complexType name="actionType">
1604   <xsd:sequence>
1605     <xsd:element name="char" type="pps:charType" minOccurs="0"/>
1606     <xsd:element name="qty" type="pps:qtyType" minOccurs="0"/>
1607     <xsd:element name="address" type="pps:addressType" minOccurs="0"/>
1608   </xsd:sequence>
1609   <xsd:attribute name="item" type="xsd:IDREF"/>
1610   <xsd:attribute name="resource" type="xsd:IDREF"/>
1611   <xsd:attribute name="pre_spec">
1612     <xsd:simpleType>
1613       <xsd:restriction base="xsd:string">
1614         <xsd:enumeration value="stock"/>
1615         <xsd:enumeration value="load"/>
1616         <xsd:enumeration value="capacity"/>
1617         <xsd:enumeration value="progress"/>
1618         <xsd:enumeration value="location"/>

```

www.abdigital.ula.ve

```

1619 </xsd:restriction>
1620 </xsd:simpleType>
1621 </xsd:attribute>
1622 <xsd:attribute name="spec" type="xsd:string"/>
1623 <xsd:attribute name="relative" type="xsd:boolean" default="false"/>
1624 </xsd:complexType>
1625 <xsd:complexType name="changeoverType">
1626 <xsd:sequence>
1627 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1628 <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>
1629 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
1630 </xsd:sequence>
1631 <xsd:attribute name="operation" type="xsd:IDREF"/>
1632 </xsd:complexType>
1633 <xsd:complexType name="intervalType">
1634 <xsd:sequence>
1635 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1636 <xsd:element name="condition" type="pps:conditionType" minOccurs="0" maxOccurs="unbounded"/>
1637 <xsd:element name="duration" type="pps:durationType" minOccurs="0"/>
1638 </xsd:sequence>
1639 <xsd:attribute name="operation" type="xsd:IDREF"/>
1640 </xsd:complexType>
1641 <!--
1642 definition of data related with plan (definición de información o data relacionada con plan)
1643 -->
1644 <xsd:complexType name="expressionType">
1645 <xsd:sequence>
1646 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
1647 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
1648 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1649 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
1650 <xsd:element name="op" type="pps:opType" minOccurs="0" maxOccurs="unbounded"/>
1651 </xsd:sequence>
1652 <xsd:attribute name="name" type="xsd:ID" use="required"/>
1653 <xsd:attribute name="ac">
1654 <xsd:simpleType>
1655 <xsd:restriction base="xsd:string">
1656 <xsd:enumeration value="create"/>
1657 <xsd:enumeration value="delete"/>
1658 <xsd:enumeration value="revise"/>
1659 </xsd:restriction>
1660 </xsd:simpleType>
1661 </xsd:attribute>
1662 <xsd:attribute name="rhd" type="xsd:double" default="0"/>
1663 <xsd:attribute name="value" type="xsd:double" default="0"/>
1664 <xsd:attribute name="violation" type="xsd:double" default="1"/>
1665 <xsd:attribute name="unit" type="xsd:IDREF"/>
1666 <xsd:attribute name="type" default="EQ">
1667 <xsd:simpleType>
1668 <xsd:restriction base="xsd:string">
1669 <xsd:enumeration value="EQ"/>
1670 <xsd:enumeration value="NE"/>
1671 <xsd:enumeration value="GT"/>
1672 <xsd:enumeration value="GE"/>
1673 <xsd:enumeration value="LT"/>
1674 <xsd:enumeration value="LE"/>
1675 <xsd:enumeration value="MAX"/>
1676 <xsd:enumeration value="MIN"/>
1677 </xsd:restriction>
1678 </xsd:simpleType>
1679 </xsd:attribute>
1680 </xsd:complexType>
1681 <xsd:complexType name="opType">
1682 <xsd:attribute name="no" type="xsd:long" default="1"/>
1683 <xsd:attribute name="operator" default="+">
1684 <xsd:simpleType>
1685 <xsd:restriction base="xsd:string">
1686 <xsd:enumeration value="+"/>
1687 <xsd:enumeration value="-"/>
1688 <xsd:enumeration value="*"/>
1689 <xsd:enumeration value="/">
1690 </xsd:restriction>
1691 </xsd:simpleType>
1692 </xsd:attribute>
1693 <xsd:attribute name="value" type="xsd:double"/>
1694 <xsd:attribute name="parameter" type="xsd:IDREF"/>
1695 <xsd:attribute name="expression" type="xsd:IDREF"/>
1696 </xsd:complexType>
1697 <xsd:complexType name="parameterType">
1698 <xsd:sequence>
1699 <xsd:element name="description" type="pps:descriptionType" minOccurs="0"/>
1700 <xsd:element name="display" type="pps:displayType" minOccurs="0"/>
1701 <xsd:element name="priority" type="pps:priorityType" minOccurs="0"/>
1702 <xsd:element name="spec" type="pps:specType" minOccurs="0" maxOccurs="unbounded"/>
1703 <xsd:element name="op" type="pps:opType" minOccurs="0" maxOccurs="unbounded"/>
1704 <xsd:element name="customer" type="pps:customerType" minOccurs="0" maxOccurs="unbounded"/>
1705 <xsd:element name="supplier" type="pps:supplierType" minOccurs="0" maxOccurs="unbounded"/>
1706 <xsd:element name="item" type="pps:itemType" minOccurs="0" maxOccurs="unbounded"/>
1707 <xsd:element name="resource" type="pps:lotType" minOccurs="0" maxOccurs="unbounded"/>
1708 <xsd:element name="lot" type="pps:lotType" minOccurs="0" maxOccurs="unbounded"/>

```

```
1709 <xsd:element name="task" type="pps:taskType" minOccurs="0" maxOccurs="unbounded"/>
1710 <xsd:element name="event" type="pps:eventType" minOccurs="0" maxOccurs="unbounded"/>
1711 <xsd:element name="operation" type="pps:operationType" minOccurs="0" maxOccurs="unbounded"/>
1712 <xsd:element name="order" type="pps:orderType" minOccurs="0" maxOccurs="unbounded"/>
1713 </xsd:sequence>
1714 <xsd:attribute name="name" type="xsd:ID" use="required"/>
1715 <xsd:attribute name="ac">
1716 <xsd:simpleType>
1717 <xsd:restriction base="xsd:string">
1718 <xsd:enumeration value="create"/>
1719 <xsd:enumeration value="delete"/>
1720 <xsd:enumeration value="revise"/>
1721 </xsd:restriction>
1722 </xsd:simpleType>
1723 </xsd:attribute>
1724 <xsd:attribute name="value" type="xsd:double"/>
1725 <xsd:attribute name="unit" type="xsd:IDREF"/>
1726 <xsd:attribute name="index" type="xsd:long" default="0"/>
1727 <xsd:attribute name="type" default="variable">
1728 <xsd:simpleType>
1729 <xsd:restriction base="xsd:string">
1730 <xsd:enumeration value="variable"/>
1731 <xsd:enumeration value="constant"/>
1732 </xsd:restriction>
1733 </xsd:simpleType>
1734 </xsd:attribute>
1735 </xsd:complexType>
1736 </xsd:schema>
1737
```

www.bdigital.ula.ve

ANEXO C

GUÍA DE INSTANCIAS

www.bdigital.ula.ve


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ppap xmlns="http://www.oasis-open.org/committees/pps/schema" xmlns:pslx="http://www.oasis-open.org/committees/pps/schema-instance" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.oasis-open.org/committees/pps/schema
3 ESQUEM~1.XSD">
4 <!--
5 Información de Nivel Tope: pslx, error y perfí
6 |-->
7 <!--
8 Petición y error
9 -->
10 <!--Esta es la data para notificar el contenido
11 del error a la parte, cliente o suplidor, cuando una falla
12 como un error ocurre. Existen "advertencia" y "error",
13 el primero permite continuar la operación y el segundo no-->
14 <!--Expresando Perfil. Especifica información sobre toda la data PSLX-->
15 <pslx type="request" id="A0000158" receipt="true" action="getAlmacen">
16 <profile name="Aquí se expresa el problema, por ejemplo: la representación de las interfaces del modelo"
17 version="versión 0.1" create="2003-06-26T00:00:00" expire="2010-06-01T00:00:00"
18 start="2005-10-17T10:00:00" current="2005-10-26T10:34:00" end="2005-10-31T23:59:59">
19 </profile>
20 </pslx>
21 <!--Especificando error-->
22 <pslx type="exception" ref="A0000158">
23 <error code="E02004" severity="error">El ID aplicable no puede ser encontrado</error>
24 <error code="E02015" severity="error">error en la especificacion de valor</error>
25 <error code="W08002" severity="warning">el valor del atributo es ignorado</error>
26 </pslx>
27 <!--
28 Expresando Cantidad en Fracciones
29 -->
30 <!--Producción de 3000 cordones para zapatos deportivos en una semana-->
31 <pslx>
32 <unit pre-det-name="week"/>
33 <item name="cordones"/>
34 <operation name="fabricar-zapato">
35 <produce item="cordones">
36 <qty value="1" unit="week">
37 <base value="3000"/>
38 </qty>
39 </produce>
40 </operation>
41 </pslx>
42 <!--
43 Expresando fecha constantes
44 -->
45 <!--Esto ejemplifica que el almacén tiene 100 unidades al "inicia" la planificación-->
46 <pslx>
47 <item name="procesador">
48 <stock>
49 <time ref="init"/>
50 <qty value="100"/>
51 </stock>
52 </item>
53 </pslx>
54 <!--
55 Expresando tiempo (duración ó duration)
56 -->
57 <!--1 hora-->
58 <pslx>
59 <unit pre-det-name="hour"/>
60 <operation name="ensamblado_1">
61 <!--ambos ensamblados 1 y 2, duran 1 hora-->
62 <duration value="PT1H"/>
63 </operation>
64 <operation name="ensamblado_2">
65 <duration count="1" unit="hour"/>
66 </operation>
67 </pslx>
68 <!--1 hora-->
69 <pslx>
70 <unit pre-det-name="half-hour"/>
71 <operation name="cases">
72 <duration count="2" unit="half-hour"/>
73 </operation>
74 </pslx>
75 <!--
76 Si una unidad no se encuentra pre determinada, (traducción)
77 la unidad se puede expresar en función de la original de la siguiente manera
78 -->
79 <pslx>
80 <unit pre-det-name="fifteen_d"/>
81 <unit name="five_d">
82 <translate unit="fifteen_d" value="0.3333333333"/>
83 </unit>
84 <!--0-->
85 <unit name="fortyfive_d"/>
86 <operation name="ejemplo-traducción">
87 <duration count="3" unit="fifteen_d"/>
88 </operation>
89 <!--Se usa para valores multiples de la unidad original -->

```

```

90 </pslx>
91 <!--
92 Expresando La escala de tiempo
93 -->
94 <pslx>
95 <unit pre-det-name="min"/>
96 <timeScale name="DiezMinutos" unit="min" value="10" origin="2005-10-26T17:48:00" type="disc"/>
97 <!--Y así se usa la escala-->
98 <operation name="Ejemplo">
99 <start>
100 <!--<time count="20" timeScale="DiezMinutos"></time-->
101 <time value="2003-04-08T03:20:00"/>
102 </start>
103 </operation>
104 </pslx>
105 <!--
106 Especificando sistema de Unidades
107 -->
108 <pslx type="request">
109 <unit pre-det-name="month"/>
110 <order name="cantidad">
111 <qty value="40" unit="month"/>
112 </order>
113 </pslx>
114 <pslx>
115 <unit pre-det-name="EUR"/>
116 <operation name="prueba">
117 <price value="5000" unit="EUR"/>
118 </operation>
119 </pslx>
120 <pslx>
121 <unit pre-det-name="piece"/>
122 <operation name="prueba1">
123 <qty value="10" unit="piece"/>
124 </operation>
125 </pslx>
126 <!--
127 Expresando series de tiempo
128 -->
129 <pslx>
130 <unit pre-det-name="degree"/>
131 <item name="ProductoA">
132 <spec name="temperature">
133 <qty value="10" unit="degree"/>
134 <time value="2003-04-10T09:00:00"/>
135 </spec>
136 <spec name="temperature">
137 <qty value="25" unit="degree"/>
138 <time value="2003-04-10T15:00:00"/>
139 </spec>
140 <!--Si no se especifica la etiqueta tiempo, la especificación de tomar correspondiente al tiempo de planificación-->
141 <spec name="especificación A">
142 <qty value="100"/>
143 </spec>
144 <!--Ambas declaraciones asumen lo mismo-->
145 <spec name="especificación A">
146 <qty value="100"/>
147 <time ref="init"/>
148 </spec>
149 <!--También se puede expresar especificación con la etiqueta "char"-->
150 <spec name="especificación B">
151 <qty value="150"/>
152 <char value="normal"/>
153 <time value="2005-10-27T12:46:00"/>
154 </spec>
155 </item>
156 </pslx>
157 <!--
158 Especificando Almacén y Carga
159 -->
160 <!--Almacén-->
161 <pslx>
162 <unit pre-det-name="person"/>
163 <item name="ProductoB">
164 <stock>
165 <time value="2003-04-10T00:00:00"/>
166 <qty value="150"/>
167 </stock>
168 <stock>
169 <time value="2003-04-11T00:00:00"/>
170 <qty value="130"/>
171 </stock>
172 <!--Valor relativo del almacén-->
173 <stock relative="true">
174 <time value="2003-04-10T00:00:00"/>
175 <qty value="150"/>
176 </stock>
177 <stock relative="true">
178 <time value="2003-04-11T00:00:00"/>
179 <qty value="20"/>

```



```

180 </stock>
181 </item>
182 <!--
183 Carga
184 -->
185 <!--El ejemplo muestra que son necesitados
186 2 trabajadores como recursos entre
187 9:00 y 14:00 y 5 hasta las 16:00-->
188 <resource name="trabajo">
189 <load>
190 <time value="2003-04-19T09:00:00"/>
191 <qty value="2" unit="person"/>
192 </load>
193 <load>
194 <time value="2003-04-19T14:00:00"/>
195 <qty value="5" unit="person"/>
196 </load>
197 <load>
198 <time value="2003-04-19T16:00:00"/>
199 <qty value="0" unit="person"/>
200 </load>
201 <!--Carga relativa-->
202 <load relative="true">
203 <time value="2003-04-19T09:00:00"/>
204 <qty value="2" unit="person"/>
205 </load>
206 <load relative="true">
207 <time value="2003-04-19T14:00:00"/>
208 <qty value="3" unit="person"/>
209 </load>
210 <load relative="true">
211 <time value="2003-04-19T16:00:00"/>
212 <qty value="-5" unit="person"/>
213 </load>
214 </resource>
215 </pslx>
216 <!--
217 Operación de Progreso
218 -->
219 <!--El siguiente ejemplo muestra que la operación P001
220 está completada a 0%, 80% y 100%
221 según las fechas correspondientes-->
222 <pslx>
223 <operation name="P001">
224 <progress>
225 <time value="2003-04-10T18:00:00"/>
226 <qty value="0.0"/>
227 </progress>
228 <progress>
229 <time value="2003-04-11T18:00:00"/>
230 <qty value="0.8"/>
231 </progress>
232 <progress>
233 <time value="2003-04-12T18:00:00"/>
234 <qty value="1.0"/>
235 </progress>
236 </operation>
237 </pslx>
238 <!--Aquí se muestra que la orden K001 es aceptada el 3 de Abril,
239 lista para comenzar el 5, comenzó el 7 y finalizó el 9
240 -->
241 <pslx>
242 <order name="K001" type="production">
243 <progress status="created">
244 <time value="2003-04-03T18:00:00"/>
245 </progress>
246 <progress status="ready">
247 <time value="2003-04-05T18:00:00"/>
248 </progress>
249 <progress status="started">
250 <time value="2003-04-07T18:00:00"/>
251 </progress>
252 <progress status="completed">
253 <time value="2003-04-09T18:00:00"/>
254 </progress>
255 </order>
256 </pslx>
257 <!--
258 Expresando localización
259 -->
260 <pslx>
261 <resource name="carro_transporte">
262 <location name="address">
263 <time value="2003-04-10T13:00:00"/>
264 <address value="place A"/>
265 </location>
266 <location name="address">
267 <time value="2003-04-10T16:00:00"/>
268 <address value="place B"/>
269 </location>

```

```

270 </resource>
271 </pslx>
272 <!--
273 Expresando Capacidad y Calendario
274 -->
275 <!--El valor capacidad muestra los límites superiores e inferiores a ser especificados
276 en el item (para valor de almacén especificado en "item")
277 o recurso (para el valor de carga especificado en "recurso").
278 El valor capacidad puede ser especificado con la etiqueta <capacity>-->
279 <pslx>
280 <!--Especificando calendario 1-->
281 <shift name="DíaDeTrabajo8h">
282 <capacity>
283 <time value="2000-01-01T00:00:00"/>
284 <qty value="0"/>
285 </capacity>
286 <capacity>
287 <time value="2000-01-01T08:00:00"/>
288 <qty value="1"/>
289 </capacity>
290 <capacity>
291 <time value="2000-01-01T18:00:00"/>
292 <qty value="0"/>
293 </capacity>
294 <capacity>
295 <time value="2000-01-01T23:59:59"/>
296 <qty value="0"/>
297 </capacity>
298 </shift>
299 <!--Especificando calendario 2. También se puede expresar sin la última ni la primera línea-->
300 <shift name="DíaDeTrabajo9h">
301 <capacity>
302 <time value="2000-01-01T00:00:00"/>
303 <qty value="0"/>
304 </capacity>
305 <capacity>
306 <time value="2000-01-01T07:00:00"/>
307 <qty value="1"/>
308 </capacity>
309 <capacity>
310 <time value="2000-01-01T18:00:00"/>
311 <qty value="0"/>
312 </capacity>
313 </shift>
314 <!--Especificando un día libre-->
315 <shift name="DíaLibre">
316 <capacity>
317 <time value="2000-01-01T00:00:00"/>
318 <qty value="0"/>
319 </capacity>
320 <capacity>
321 <time value="2000-01-01T23:59:59"/>
322 <qty value="0"/>
323 </capacity>
324 </shift>
325 <!--Especificando calendario 3 (previos definidos 1 y 2)-->
326 <resource name="FábricaA">
327 <calendar shift="DíaDeTrabajo8h" time="2001-09-01T00:00:00"/>
328 <calendar shift="DíaDeTrabajo8h" time="2001-09-02T00:00:00"/>
329 <calendar shift="DíaDeTrabajo9h" time="2001-09-03T00:00:00"/>
330 <calendar shift="DíaLibre" time="2005-09-04T00:00:00"/>
331 </resource>
332 <!--Especificando Capacidad-->
333 <!--Este ejemplo muestra que existen 2 trabajadores de 8:00 a 12:00, 2 de 12:00 a 13:00 y 7 de 13:00 a 17:00-->
334 <resource name="Trabajo">
335 <capacity>
336 <time value="2003-04-10T08:00:00"/>
337 <qty value="5" unit="person"/>
338 </capacity>
339 <capacity>
340 <time value="2003-04-10T12:00:00"/>
341 <qty value="2" unit="person"/>
342 </capacity>
343 <capacity>
344 <time value="2003-04-10T13:00:00"/>
345 <qty value="7" unit="person"/>
346 </capacity>
347 <capacity>
348 <time value="2003-04-10T17:00:00"/>
349 <qty value="0" unit="person"/>
350 </capacity>
351 </resource>
352 </pslx>
353 <!--
354 Expresando elementos "básicos" <event> (evento), <dispatch> (despacho),
355 <operation> (operación), <item> (item), <event> (evento) y <order> (orden)
356 -->
357 <!--Usando el atributo "master"-->
358 <pslx>
359 <operation name="cortando" master="true"/>

```

www.bdigital.ula.ve

```

360 <operation name="cortando_001" parent="cortando"/>
361 </pslx>
362 <!--
363 Especificando Inclusión Item
364 -->
365 <!--Este ejemplo muestra que el Producto-X, está formado
366 por una configuración del Producto-A y Producto-B-->
367 <pslx>
368 <item name="Producto-X"/>
369 <item name="Producto-A">
370 <partof item="Producto-X"/>
371 </item>
372 <item name="Producto-B">
373 <partof item="Producto-X"/>
374 </item>
375 <!--Este ejemplo muestra la cantidad de items "hijos" que forman a un "padre"-->
376 <item name="partePadre"/>
377 <item name="partesHijas">
378 <partof item="partePadre">
379 <qty value="10"/>
380 </partof>
381 </item>
382 <!--Muestra de Inclusión en "recurso", en donde la "máquinaA"
383 es un elemento que forma a la líneaB-->
384 <resource name="líneaB"/>
385 <resource name="máquinaA">
386 <partof resource="líneaB"/>
387 </resource>
388 <!--En el caso de "operación" es similar, por ejemplo:
389 "especificaciónA", "procesamientoB" y "transportaciónC"
390 se componen en una "operaciónP"-->
391 <operation name="operaciónP"/>
392 <operation name="especificaciónA">
393 <partof operation="operaciónP"/>
394 </operation>
395 <operation name="procesamientoB">
396 <partof operation="operaciónP"/>
397 </operation>
398 <operation name="transportaciónC">
399 <partof operation="operaciónP"/>
400 </operation>
401 </pslx>
402 <!--La relación de inclusión de orden es utilizada para administrar
403 una orden individual definida en cada "item" como una sola "orden".
404 Por ejemplo: Cuando se ordenan 10 unidades del "producto A"
405 y 20 del "producto B" el contenido de la orden se expresa como sigue:-->
406 <pslx>
407 <item name="producto-A"/>
408 <item name="producto-B"/>
409 <order name="K002" type="purchase"/>
410 <order name="K002-1" item="producto-A" type="purchase">
411 <qty value="10" unit="piece"/>
412 <partof order="K002"/>
413 </order>
414 <order name="K002-2" item="producto-B" type="purchase">
415 <qty value="20" unit="piece"/>
416 <partof order="K002"/>
417 </order>
418 </pslx>
419 <!--
420 Definición de data de restricción extendida "action" (acción)
421 -->
422 <!--La información que puede ser cambiada por <action> (acción) es:
423 valor del almacén (stock value), valor de carga (load value),
424 valor de capacidad (capacity value), progreso (progress),
425 localización (location) y la especificación originalmente indicada en una etiqueta-->
426 <!--Lo siguiente expresa la acción para hacer que un "producto X" aumente su temperatura
427 en +5° C.-->
428 <pslx>
429 <unit pre-det-name="degree_C"/>
430 <item name="productoX"/>
431 <event name="SubirTemperatura">
432 <action item="productoX" spec="temperature" relative="true">
433 <qty value="5" unit="degree_C"/>
434 </action>
435 </event>
436 </pslx>
437 <!--Lo siguiente expresa que el valor de la especificación "status"
438 de la "máquina A" está especificada como "off" (apagada)-->
439 <pslx>
440 <resource name="máquina_A"/>
441 <event name="mantenimiento">
442 <action resource="máquina_A" spec="status">
443 <char value="off"/>
444 </action>
445 </event>
446 </pslx>
447 <!--
448 Especificando "condición" (condición)
449 -->

```

www.digital.ula.ve

```

450 <!--La condición especifica el artículo (item) o el recurso (resource) previsto y un nombre
451 de especificación, y especifica la manera de comparar el valor de la
452 especificación y el valor especificado con el "char" (caracteres), "qty" (cantidad) o "address" (dirección)
453 a través de el atributo "type"-->
454 <pslx>
455 <!--Ejemplo con item-->
456 <item name="producto_Y">
457   <interval>
458     <condition item="producto_Y" spec="stock">
459       <char value="full"/>
460     </condition>
461   </interval>
462 </item>
463 <!--La condición de que una "máquina_C" esté lista (ready), se expresa así-->
464 <resource name="máquina_C">
465   <changeover>
466     <condition resource="máquina_C" spec="status" type="EQ">
467       <char value="ready"/>
468     </condition>
469   </changeover>
470 </resource>
471 </pslx>
472 <!--Caso en el que todo es chequeado en el 2do piso de la compañía-X-->
473 <pslx>
474   <order name="comienzo" type="work" resource="máquina_C">
475     <release>
476       <condition spec="location" type="EQ">
477         <address value="compañía-X" height="2"/>
478       </condition>
479     </release>
480   </order>
481 </pslx>
482 <!--
483 Especificando Orden
484 -->
485 <!--Existen tres tipos de órdenes en general: "orden de item" (item order),
486 orden de recursos (resource order) y "orden de operación" (operation order)-->
487 <pslx>
488   <unit pre-det-name="day"/>
489 <!--
490 Orden de Item (Item order)
491 -->
492 <!--El tiempo de entrega de la orden (duetime)
493 que es la última fecha (latest date) de completación de operación, para satisfacer
494 la petición puede ser especificada con <duetime>.
495 Orden para 150 unidades del producto-X
496 para el mediodía del 20 de Octubre del 2005-->
497 <item name="producto-X"/>
498 <order name="K003" item="producto-X">
499   <qty value="150"/>
500   <duetime>
501     <time value="2005-10-29T12:00:00"/>
502   </duetime>
503 </order>
504 <!--También el tiempo de entrega de la orden (order duetime) puede ser especificado
505 con una etiqueta de salida de orden <release> ,
506 quien es la fecha más temprana de comienzo (earliest start date)
507 de operación para realizar la orden.
508 -->
509 <order name="k003" item="producto-X">
510   <qty value="150"/>
511   <release>
512     <time value="2005-10-29T12:00:00"/>
513   </release>
514 </order>
515 <!--
516 Orden de Recurso (Resource Order)
517 -->
518 <!--La petición de que 15 trabajadores vengan a ayudar
519 para el 11 de Octubre y hasta el 13 de Octubre se expresa así-->
520 <order name="K004" resource="trabajo">
521   <qty value="15" unit="person"/>
522   <release>
523     <time value="2003-04-11T00:00:00"/>
524   </release>
525   <duetime>
526     <time value="2003-04-14T00:00:00"/>
527   </duetime>
528 <!--Esto indica que puede haber un valor máximo de 5 trabajadores por día-->
529   <tasksize type="max" unit="day" value="5"/>
530 </order>
531 <!--En el ejemplo anterior, no se especifica qué cantidad de personas deben venir por día
532 sino el total, si se quisiera especificar, se utilizaría el atributo "multiplier", si se utilizara como unidad "día",
533 entonces cada día de los tres días del ejemplo, deberían venir 15 personas.-->
534 <!--Lo siguiente expresa número de "unidades de operación". El tamaño del lote (lot) o tarea (task)
535 puede ser aumentado o disminuido por el número de unidades de esta operación-->
536 <order name="k004" operation="cortando">
537   <qty value="5" unit="máquina_A"/>
538 </order>
539 </pslx>

```

www.modigital.ula.ve

```

540 <!--
541 Orden de Operación (Operation Order)
542 -->
543 <!--Corresponde a instrucción de operación
544 y directamente solicita ejecución de operación.-->
545 <!--Por ejemplo, en caso de que el número de "producto A" producido
546 sea definido como 2 en una etiqueta <operation> como la información de operación,
547 si el número de "unidades de orden de operación" es 3,
548 el tamaño de lote (lotsize) del "producto A"es 6
549 -->
550 <!--Asumiendo que la "máquina A" se necesita para producir el "producto A"-->
551 <pslx>
552 <order name="K005" operation="operaciónP01">
553 <qty value="3"/>
554 </order>
555 <operation name="operaciónP01">
556 <produce item="producto-A">
557 <qty value="2" unit="máquina-A"/>
558 </produce>
559 </operation>
560 </pslx>
561 <!--
562 Expresando Precio de la orden
563 -->
564 <!--El precio puede ser expresado también para <item>
565 <resource>, <operation> y <event>. El caso donde el precio de unidad de un "producto D"
566 es 2 millones de Bs, y 3 productos D son ordenados se expresa como sigue:-->
567 <pslx>
568 <unit pre-det-name="Bs"/>
569 <item name="producto-D">
570 <price value="2000000.00" unit="Bs"/>
571 </item>
572 <order name="K006" item="producto-D">
573 <price value="6000000.00" unit="Bs"/>
574 <qty value="3" unit="piece"/>
575 </order>
576 </pslx>
577 <!--
578 Expresando Cliente y Supliador (customer and supplier)
579 -->
580 <!--Información de orden en general puede expresarse como sigue-->
581 <pslx>
582 <customer name="G001">
583 <spec name="TEL">
584 <char value="0058-274-2639326"/>
585 </spec>
586 <spec name="CEL">
587 <char value="0058-414-7242341"/>
588 </spec>
589 <spec name="persona a cargo">
590 <char value="Héctor Zambrano"/>
591 </spec>
592 </customer>
593 <order name="Guitarras-Guibson-ES335" customer="G001" status="fixed" ac="create">
594 </order>
595 </pslx>
596 <!--
597 Expresando categorización a través de asociación o marcado (Pegging)
598 -->
599 <!--
600 Por ejemplo, si dos ordenes (orden B y orden C) son publicadas
601 para un proceso de producción A para satisfacer una orden A de un cliente, la
602 orden B y orden C son asociadas a la orden A asi:-->
603 <pslx>
604 <order name="orden_A"/>
605 <order name="orden_B">
606 <pegging order="orden_A"/>
607 </order>
608 <order name="orden_C">
609 <pegging order="orden_A"/>
610 </order>
611 <!--La orden X produce 10 productos para la "orden de cliente A"
612 y 15 productos para la "orden de cliente B"-->
613 <order name="cliente_orden_A"/>
614 <order name="cliente_orden_B"/>
615 <order name="orden_X">
616 <pegging order="cliente_orden_A">
617 <qty value="10"/>
618 </pegging>
619 <pegging order="cliente_orden_B">
620 <qty value="15"/>
621 </pegging>
622 </order>
623 <!-- "pegging" tiene una dirección, el lado cercano al cliente es tratado como "lower",
624 y el lado cercano al procesamiento en el lado de la materia prima, es tratado como "upper"-->
625 <order name="orden_A">
626 <pegging order="orden_X" type="upper">
627 <qty value="10"/>
628 </pegging>
629 </order>

```

```

630 </pslx>
631 <!--
632 Especificando Operación.
633 -->
634 <pslx>
635 <!--Cuando se instancia una operación,
636 respecto de una operación abstracta master, se indica lo siguiente:-->
637 <operation name="Ensamblaje_Computadora" master="true"/>
638 <operation name="Ensamblaje_TarjetaMadre" parent="Ensamblaje_Computadora"/>
639 <operation name="hacia_adelante" direction="forward">
640 <description>
641 <description>Hacia adelante significa que la operación comenzará lo antes posible,
642 y hacia atrás (backward) significa que la operación comenzará
643 lo más lejos que lo permita &lt;duetime&gt;
644 </description>
645 </description>
646 </operation>
647 <!--Para la producción ordinaria, el atributo "type" contiene make, para almacenamiento "stock",
648 para mover "move" y para revisar "check"-->
649 <operation name="producción_ordinaria" type="make"/>
650 </pslx>
651 <!--La información de "orden" que hace referencia a
652 producción de operaciones se expresa así:-->
653 <pslx>
654 <order name="Instrucciones_de_operación"/>
655 <operation name="Instrucción_A01" order="Instrucciones_de_operación"/>
656 </pslx>
657 <!--
658 Tiempo de operación (operation time)
659 -->
660 <!--El ejemplo siguiente muestra que el tiempo de operación es de 120 minutos y "fijo".
661 En este caso, la unidad básica de tiempo de operación está en términos de segundos (sec)
662 y entonces, 120 minutos son cambiados en términos de segundos-->
663 <pslx>
664 <operation name="operación-A">
665 <duration value="PT7200S" type="fixed"/>
666 </operation>
667 <!--El siguiente ejemplo muestra que la cantidad de la operación B001 es 3 unidades
668 y la unidad de tiempo de operación es 30 minutos (1800 segundos) y entonces 90 minutos
669 son requeridos como la cantidad total.-->
670 <operation name="operación-B" master="true">
671 <duration value="PT1800S"/>
672 </operation>
673 <operation name="B001" parent="operación-B">
674 <qty value="3"/>
675 <start>
676 <time value="2003-04-10T00:00:00"/>
677 </start>
678 <end>
679 <time value="2003-04-10T01:30:00"/>
680 </end>
681 </operation>
682 </pslx>
683 <!--
684 Expresando Evento
685 -->
686 <pslx>
687 <!--no_usar_para_producción_de_órdenes-->
688 <order name="release">
689 <release/>
690 </order>
691 <order name="duetime">
692 <duetime/>
693 </order>
694 <!--no_usar_para_producción_de_operaciones-->
695 <operation name="start">
696 <start/>
697 </operation>
698 <operation name="end">
699 <end/>
700 </operation>
701 </pslx>
702 <!--
703 Produciendo y Consumiendo Item
704 -->
705 <pslx>
706 <!--El caso donde la operación-A utiliza 10 unidades de materia prima Y
707 y produce una unidad de producto-x se expresa así:-->
708 <item name="producto-x"/>
709 <item name="materia_prima_Y"/>
710 <item name="material-raro"/>
711 <item name="partes-Z"/>
712 <operation name="operation-A" master="true">
713 <produce item="producto-x"/>
714 <consume item="materia_prima_Y">
715 <qty value="10"/>
716 </consume>
717 </operation>
718 <!--cuando se producen o consumen "múltiples tipos items" al mismo tiempo,
719 el atributo "nth" es especificado cambiando su valor en enteros continuos

```



```

720 comenzando en "1". La operación B produce producto-x y material raro, y consume
721 materia prima Y y partes Z al mismo tiempo-->
722 <operation name="operación_B" master="true">
723   <produce item="producto-x" nth="1"/>
724   <produce item="material-raro" nth="2"/>
725   <consume item="materia_prima_Y" nth="1"/>
726   <consume item="partes-Z" nth="2"/>
727 </operation>
728 </pslx>
729 <!--Si una operación es la información de instancia, el atributo "lot" (lote)
730 puede ser especificado. Ejemplo, cuando la operación-F produce 2 unidades
731 del producto-X, una instancia de la operación F (F001) produce un lote de
732 producto-X (X001)-->
733 <pslx>
734   <operation name="operación-F"/>
735   <operation name="F001" parent="operación-F">
736     <produce item="operación-F" lot="X001">
737       <qty value="2" unit="producto-X"/>
738     </produce>
739   </operation>
740   <lot name="X001"/>
741 </pslx>
742 <!--
743 Asignación de Recursos
744 -->
745   <!--Una "operación G" utiliza una "máquina A"-->
746   <pslx>
747     <resource name="máquina-A"/>
748     <resource name="máquina-B"/>
749     <operation name="operación-G">
750       <assign resource="máquina-A"/>
751     </operation>
752   <!--La operación-H necesita 3 trabajadores-->
753   <operation name="operación-H">
754     <assign resource="trabajo">
755       <qty value="3" unit="person"/>
756     </assign>
757   </operation>
758   <!--Cuando "alguna" (no ambas) máquina X y máquina Yes utilizada,
759   La etiqueta es especificada como sigue. En este caso nth="1" es el
760   valor de abreviación y puede ser omitido.-->
761   <operation name="operación-I">
762     <assign resource="máquina-A" nth="1"/>
763     <assign resource="máquina-B" nth="1"/>
764   </operation>
765 </pslx>
766 <!--
767 Carga de recursos
768 -->
769   <!--En el ejemplo se muestra como gradualmente el nivel de carga sube
770   gradualmente desde 0 por la primera hora, hasta la "operación estable",
771   en donde se mantiene la carga fija durante 4 horas, luego el nivel baja
772   gradualmente de nuevo durante una hora, y la operación finaliza.-->
773   <pslx>
774     <resource name="máquina-X"/>
775     <operation name="comienzo-operación">
776       <!--Se asume que la carga comienza en 0-->
777       <start>
778         <time value="2002-09-01T00:00:00"/>
779       </start>
780       <end>
781         <time value="2002-09-01T01:00:00"/>
782       </end>
783       <assign timeType="cont" resource="máquina-X">
784         <qty value="100"/>
785     <!--La carga subió gradualmente durante 1h, hasta llegar a 100 -->
786     </assign>
787   </operation>
788   <operation name="operación-estable">
789     <start>
790       <time value="2002-09-01T01:00:00"/>
791     </start>
792     <end>
793       <time value="2002-09-01T05:00:00"/>
794     </end>
795     <assign timeType="disc" resource="máquina-X">
796       <qty value="100"/>
797   <!--La carga se mantiene por 4 horas-->
798   </assign>
799 </operation>
800 <operation name="operación-final">
801   <start>
802     <time value="2002-09-01T05:00:00"/>
803   </start>
804   <end>
805     <time value="2002-09-01T06:00:00"/>
806   </end>
807   <assign timeType="cont" resource="máquina-X">
808     <qty value="-100"/>
809   <!--La carga baja a 0 gradualmente, durante 1h-->

```

www.bdigital.ula.ve

```

810 </assign>
811 </operation>
812 </pslx>
813 <!--El atributo "nth" expresa qué puede ser seleccionado al mismo tiempo,
814 producción, consumo y asignación según la variedad de ítems y recursos,
815 para cada caso. Cuando el valor de "nth" es el mismo, uno debe ser seleccionado
816 para una agenda ejecutable. Este valor puede ser especificado independientemente
817 en <produce>, <consume> y <assign>.-->
818 <!--En el siguiente ejemplo, la operación-J produce artículo-A "y" artículo-B juntos
819 pero la operación-K produce artículo-C "ó" artículo D porque ambos llevan el valor "1"
820 en el atributo "nth".-->
821 <pslx>
822 <item name="artículo-A1"/>
823 <item name="artículo-A2"/>
824 <item name="artículo-B1"/>
825 <item name="artículo-B2"/>
826 <item name="artículo-A"/>
827 <item name="artículo-B"/>
828 <item name="artículo-C"/>
829 <item name="artículo-D"/>
830 <operation name="operación-X">
831 <produce item="artículo-A" nth="1"/>
832 <produce item="artículo-B" nth="2"/>
833 </operation>
834 <operation name="operación-Y">
835 <produce item="artículo-C" nth="1"/>
836 <produce item="artículo-D" nth="1"/>
837 </operation>
838 <!--El atributo "sel" es utilizado para la asociación de selecciones individuales
839 y para expresar la complicada relación de combinar "AND" y "OR"-->
840 <!--Lo siguiente muestra que cuando la operación-W selecciona
841 el número "1" para "sel", el artículo B1 es producido del artículo A1,
842 y cuando selecciona el número "2" para "sel", el artículo B2 es producido del A2.-->
843 <operation name="operación-W">
844 <produce item="artículo-B1" sel="1"/>
845 <produce item="artículo-B2" sel="2"/>
846 <consume item="artículo-A1" sel="1"/>
847 <consume item="artículo-A2" sel="2"/>
848 </operation>
849 </pslx>

```

```

850 <!--
851 Precedencia de Operación
852 -->

```

```

853 <!--El siguiente ejemplo expresa que el sucesor de la operación-K es
854 la operación-M. El valor del atributo "type" (tipo) es omitido porque es "FS".-->
855 <pslx>
856 <operation name="operación-K">
857 <successor operation="operación-M"/>
858 </operation>
859 <!--Esta relación puede ser representada desde el punto de vista de la operación-M
860 y estas relaciones tienen exactamente el mismo significado-->
861 <operation name="operación-M">
862 <predecessor operation="operación-K"/>
863 </operation>
864 <!--Omitiendo <duration> significa que el valor es "0" o mas.
865 En el ejemplo siguiente se muestra que entre el final de la operación-N
866 y el comienzo de la operación-O deben haber 60 minutos o mas.-->
867 <operation name="operación-N">
868 <successor operation="operación-O">
869 <duration value="PT3600S"/>
870 </successor>
871 </operation>
872 <operation name="operación-O"/>
873 <!--En el ejemplo de arriba, si el tiempo debe ser sólo 60 minutos,
874 el atributo "type" (tipo) de <duration> debe ser "fixed" (fijo).
875 Si el tiempo debe ser 60 minutos o menos, no 60 y mas, entonces "short" (corto)
876 es especificado en el atributo "type" (tipo)-->
877 </pslx>

```

```

878 <!--
879 Operation Changeover
880 -->
881 <!--(Relación entre operaciones ejecutadas de forma continua en un recurso)-->
882 <!--Cuando la "condición" de <chageover> (cambio) toma en cuenta sólo
883 la operación siguiente ó la operación anterior, el atributo "direction" es
884 "suc" o "pre" respectivamente; y cuando toma en cuenta ambas es "cmb".-->
885 <!--El siguiente ejemplo muestra que si el valor "setup" del sucesor (actual) es "yes",
886 10 minutos son mantenidos para el setup (configuración).-->
887 <pslx>
888 <resource name="máquina-01"/>
889 <operation name="setup">
890 <changeover>
891 <condition resource="máquina-01" spec="specified" type="EQ" direction="pre">
892 <char value="yes"/>
893 </condition>
894 <duration value="PT600S"/>
895 </changeover>
896 </operation>
897
898 </pslx>

```

```

899 <!--Esto significa que "limpia", manejado por una serie de cambios

```


H.900 operacionales, toma 60 min en configurarse. En tal caso, no se usa <qty> ni <char>
901 en la etiqueta <condition> porque el contenido de las especificaciones de predecesor y sucesor
902 son comparados-->

```
903 <pslx>  
904 <resource name="máquina-02">  
905 <changeover>  
906 <condition spec="limpia" type="NE" direction="cmb"/>  
907 <duration value="PT360S"/>  
908 </changeover>  
909 </resource>
```

```
910 </pslx>  
911 <!--En este ejemplo se ilustra la situación en la que se necesitan 20 min  
912 para cambiar de "tipo-A" a "tipo-B" y 30 minutos para cambiar de "tipo-B" a "tipo-C"-->
```

```
913 <pslx>  
914 <resource name="Pega-Suelas">  
915 <changeover>  
916 <condition spec="tipo" type="EQ" direction="pre">  
917 <char value="A"/>  
918 </condition>  
919 <condition spec="tipo" type="EQ" direction="suc">  
920 <char value="B"/>  
921 </condition>  
922 <duration value="PT120S"/>  
923 </changeover>  
924 <changeover>  
925 <condition spec="tipo" type="EQ" direction="pre">  
926 <char value="B"/>  
927 </condition>  
928 <condition spec="tipo" type="EQ" direction="suc">  
929 <char value="C"/>  
930 </condition>  
931 <duration value="PT180S"/>  
932 </changeover>  
933 </resource>
```

```
934 </pslx>  
935 <!--Cuando el atributo operación es especificado en una etiqueta <changeover>  
936 indicando "cambio" significa que la operación especificada es requerida cuando la condición  
937 especificada con <condition> es satisfecha. Pero el caso del ejemplo corresponde al caso  
938 en donde la configuración debe ser "operada" separadamente-->
```

```
939 <!--  
940 Item Interval Period (período de intervalo de tiempo)
```

```
941 -->  
942 <!--Esta información especifica el intervalo de tiempo, desde que una operación produce un item  
943 al tiempo en que otra operación consume el item y el item ha desaparecido.  
944 gone -->  
945 <!--la etiqueta <changeover> que indica data de cambio es definida en una etiqueta <resource>  
946 indicando recurso, mientras que una etiqueta <interval> que indica data de intervalo es definida  
947 en una etiqueta <item> indicando item-->  
948 <!--El estatus en donde un producto-A puede ser guardado en el almacén por 10 días o menos  
949 y no puede ser guardado por más de 10 días.-->
```

```
950 <pslx>  
951 <item name="producto-P">  
952 <interval>  
953 <duration count="10" unit="day" type="shorter"/>  
954 </interval>  
955 </item>
```

```
956 </pslx>  
957 <!--El siguiente ejemplo muestra que el intervalo de 20 horas debe  
958 ser establecido sólo para "procesamiento especial"-->
```

```
959 <pslx>  
960 <item name="producto-Q">  
961 <interval>  
962 <condition item="producto-Q" spec="método de procesamietno" type="EQ" direction="pre">  
963 <char value="procesamiento especial"/>  
964 </condition>  
965 <duration count="20" unit="hour"/>  
966 </interval>  
967 </item>
```

```
968 </pslx>  
969 <!--Cuando el atributo operación es especificado en una etiqueta <interval>  
970 indicando "intervalo" significa que la operación especificada es requerida  
971 para satisfacer la condición especificada con <condition>-->
```

```
972 <!--  
973 Lot and Task (lote y tarea)
```

```
974 -->  
975 <!--Lote <lot> es el item consumido o producido por la instancia una operación-->  
976 <!--El ID para la identificación del lote es expresado con el atributo "nombre".  
977 La cantidad de lotes producidos o consumidos es expresado con <qty>.  
978 El lugar donde el lote es producido o consumido es expresado con <address>.  
979 El atributo "type" (tipo) expresa si el lote es producido o consumido por una operación.  
980 La relación entre el lote producido y el consumido puede ser especificada con  
981 <tracking> como la información de rastreo-->  
982 <!--Cuando la operación-C produce 2 producto-Y, la instrucción de producción concreta es C001  
983 y el lote que indica la producción concreta del producto-Y es Y001, se expresa como sigue-->
```

```
984  
985 <pslx>  
986 <item name="producto-Y" type="phantom">  
987 <produced operation="operación-C">  
988 <qty value="2"/>  
989 </produced>
```

```

990 </item>
991 <operation name="operación-C" master="true"> <!--forma alterna de especificar lote-->
992   <produce item="producto-Y">
993     <qty value="2" unit="piece"/>
994   </produce>
995 </operation>
996 <operation name="C001" parent="operación-C">
997   <produce item="producto-Y" lot="Y001"/>
998 </operation>
999 <lot name="Y001" type="produced"> <!--forma alterna de especificar lote-->
1000   <qty value="2" unit="piece"/>
1001 </lot>
1002 </pslx>
1003 <!--
1004 Lotsize (tamaño del lote)
1005 -->
1006 <!--El tamaño del lote es una restricción que se utiliza en información
1007 de <item>, <operation> y <order>.
1008 Cuando el <lotsize> compite entre si, primero toma prioridad data de orden,
1009 luego data de operación y por último data de item.
1010 La información de instancia siempre tiene prioridad sobre la información master.
1011 Se puede utilizar otra opción, que es jerarquizar por medio de la etiqueta <priority-->
1012 <!--El atributo "type" (tipo) pueden: "fixed" (el valor especificado es fijo),
1013 min (mínimo tamaño del lote), max (máximo tamaño del lote) ó
1014 unit (unidad múltiplo del valor especificado).-->
1015 <pslx>
1016   <item name="producto-W ">
1017     <lotsize value="30" type="min"/>
1018     <lotsize value="200" type="max"/>
1019     <lotsize value="10" type="unit"/>
1020   </item>
1021 </pslx>
1022 <!--En el ejemplo anterior, el tamaño mínimo del lote es 30, el tamaño máximo es 200
1023 y la unidad del tamaño del lote es 10. Si la orden es 20, el tamaño del lote es 30,
1024 y si la orden es 32, el tamaño del lote es 40 (porque el valor de la unidad es 10).
1025 Si la orden es 230, dos instrucciones de producción son producidas;
1026 una con tamaño del lote 200 y otra con tamaño del lote 30.-->
1027 <!--
1028 Informaton on task (información sobre tarea)
1029 -->
1030 <!--Task (tarea) indica el estado donde una operación es asignada
1031 a un recurso(s) y utiliza(n) la capacidad del mismo-->
1032 <!--La cantidad de tareas (task) es especificada con la etiqueta <qty>.
1033 Las tareas suficientes para el valor de capacidad requerido, son individualmente
1034 especificadas a cada recurso asignado con operación.
1035 El lugar donde la tarea es producida, es decir, la localización del recurso objetivo,
1036 puede ser especificada en cada tarea con <address>
1037 Cuando la unidad de tiempo es especificada en el atributo "multiplier" (multiplicador),
1038 el valor significa la "cantidad de trabajo" (amount of work)-->
1039 <pslx>
1040 <!--Cuando la operación-P utiliza la máquina-X, el uso concreto de la
1041 máquina "X002" se expresa así-->
1042 <operation name="operación-P">
1043   <assign resource="máquina-X">
1044     <qty value="3"/>
1045   </assign>
1046 </operation>
1047 <task name="X002" resource="máquina-X">
1048   <qty value="3"/>
1049 </task>
1050 </pslx>
1051 <!--La restricción de el tamaño de la tarea (tasksize) puede ser especificada
1052 para una <resource>, <operation> y <order>. La capacidad del recurso es
1053 usaa siguiendo las reglas que especifica <tasksize> -->
1054 <!--Si el número de trabajadores a destajo es 10 y el valor máximo del tamaño
1055 de la tarea (tasksize) es 4, la agenda de trabajo para 10 trabajadores
1056 es 4 personas en el primer día, 4 personas en el segundo día y 2 en el tercero-->
1057 <pslx>
1058   <resource name="trabajador-destajo" type="personnel" multiplier="day">
1059     <tasksize value="4" type="max"/>
1060   </resource>
1061   <operation name="operación-D">
1062     <assign resource="trabajador-destajo"/>
1063   </operation>
1064   <task name="operación_D-001" resource="trabajador-destajo">
1065     <qty value="4"/>
1066   </task>
1067   <task name="operación_D-002" resource="trabajador-destajo">
1068     <qty value="4"/>
1069   </task>
1070   <task name="operación_D-003" resource="trabajador-destajo">
1071     <qty value="2"/>
1072   </task>
1073 </pslx>
1074 <!--Cuando la tarea especificada en operación compite con la especificada en
1075 recurso y orden, el orden de prioridad es orden, operación y recurso.
1076 La información de instancias siempre tiene prioridad sobre la información "master"-->
1077 <!--
1078 Lot Tracking and Task Tracking (rastreo de lote y tarea)
1079 -->

```

1080 <!--El rastreo de lote (lot tracking) está disponible al asociar un lote de producción
1081 con un lote de consumo en el mismo ítem. Esta es la información que apunta desde
1082 el lote de producción hacia el lote de consumo o viceversa. Cada correspondencia
1083 puede ser especificada con una relación de cantidad usando <qty>.-->
1084 <!--Por ejemplo, si el procesamiento anterior, operación-1, es ejecutado 3 veces
1085 y el procesamiento siguiente, operación-2 es ejecutado 2 veces en el producto
1086 medio A (phantom), el siguiente rastreo de datos puede ser definido
1087 (ver figura 7 "lot tracking", PSLX Consortium,
1088 XML Standard Specification, pág 116/221)-->

```
1089 <pslx>  
1090 <lot name="P1-001">  
1091 <qty value="10"/>  
1092 </lot>  
1093 <lot name="P1-002">  
1094 <qty value="10"/>  
1095 </lot>  
1096 <lot name="P1-003">  
1097 <qty value="10"/>  
1098 </lot>  
1099 <lot name="P2-001" type="consumed">  
1100 <qty value="12"/>  
1101 <tracking lot="P1-001">  
1102 <qty value="10"/>  
1103 </tracking>  
1104 <tracking lot="P1-002">  
1105 <qty value="2"/>  
1106 </tracking>  
1107 </lot>  
1108 <lot name="P2-002" type="consumed">  
1109 <qty value="12"/>  
1110 <tracking lot="P1-002">  
1111 <qty value="8"/>  
1112 </tracking>  
1113 <tracking lot="P1-003">  
1114 <qty value="4"/>  
1115 </tracking>  
1116 </lot>  
1117 </pslx>
```

1118 <!--Ejemplo de rastreo de tareas-->
1119 <!--ver figura 8 "task tracking", PSLX Consortium,
1120 XML Standard Specification, pág 117/221)-->

```
1121 <pslx>  
1122 <resource name="recurso-x"/>  
1123 <task name="Q1-001" resource="recurso-x">  
1124 <qty value="10"/>  
1125 </task>  
1126 <task name="Q2-001" type="consumed">  
1127 <tracking lot="Q1-001">  
1128 <qty value="20"/>  
1129 </tracking>  
1130 </task>  
1131 <task name="Q2-002" type="consumed">  
1132 <tracking lot="Q1-001">  
1133 <qty value="20"></qty>  
1134 </tracking>  
1135 </task>  
1136 <task name="Q2-003" type="consumed">  
1137 <tracking lot="Q1-001">  
1138 <qty value="20"></qty>  
1139 </tracking>  
1140 </task>  
1141 <task name="Q2-004" type="consumed">  
1142 <tracking lot="Q1-001">  
1143 <qty value="20"></qty>  
1144 </tracking>  
1145 </task>  
1146 <task name="Q2-005" type="consumed">  
1147 <tracking lot="Q1-001">  
1148 <qty value="20"></qty>  
1149 </tracking>  
1150 </task>  
1151 </pslx>
```

1152 <!--
1153 Inquiring Data (solicitando información o data
1154 -->

```
1155 <!--Available Range of Value (rango disponible de valor)-->  
1156 <!--El rango de valor debe ser especificado cuando se solicita data.  
1157 Pueden ser especificados rangos para <qty>, <price>, <priority> y <duration>  
1158 para data numérica; y para información de carácter <char>; igualmente  
1159 la fecha más temprana y la más tardía se expresan con <time>.  
1160 Los valores son especificados para las etiquetas anteriores con el atributo "value",  
1161 y el rango con los atributos "min" y "max"-->  
1162 <!--Por ejemplo, cuando el rango desde 1.000.000,00 Bs a 2.000.000,00 es especificado,  
1163 La etiqueta se expresa como sigue-->
```

```
1164 <pslx>  
1165 <item name="Computadora-Básica">  
1166 <price unit="Bs">  
1167 <min value="1000000"/>  
1168 <max value="2000000"/>  
1169 </price>
```

```

1170 </item>
1171 </pslx>
1172 <!--Cuando el valor numérico no se incluye para comparar, el atributo "exclusive"
1173 (exclusivo) puede ser utilizado.
1174 Por ejemplo, cuando el valor está por debajo de 2 millones (sin incluir el extremo),
1175 la etiqueta es especificada utilizando el atributo "exclusive"-->
1176 <pslx>
1177 <order name="Computadora-B">
1178 <price unit="Bs">
1179 <min value="1000000"/>
1180 <max value="2000000" exclusive="true"/>
1181 </price>
1182 </order>
1183 </pslx>
1184 <!--En caso de información de "caracteres", El valor propuesto seleccionable
1185 puede ser especificado con una etiqueta <enumerate>.
1186 El siguiente ejemplo expresa el "ensamblaje", que consiste en 2 cadenas de
1187 caracteres, "sí" y "no"-->
1188 <pslx>
1189 <operation name="Ensamblaje">
1190 <spec name="comienzo">
1191 <char>
1192 <enumerate value="yes"/>
1193 <enumerate value="no"/>
1194 </char>
1195 <!-- La etiqueta <earliest> tag indica la fecha más temprana y <latest> la más tardía,
1196 y pueden ser utilizadas para especificar el rango de fecha disponible.-->
1197 <!--El siguiente ejemplo muestra el período desde las 8 am
1198 (hora más temprana de comienzo del ensamblaje), hasta las 7 pm
1199 (hora más tardía de finalización del ensamblaje)el 11/03/2005-->
1200 <time>
1201 <earliest value="2003-04-10T08:00:00"/>
1202 <latest value="2003-04-10T19:00:00"/>
1203 </time>
1204 </spec>
1205 </operation>
1206 </pslx>
1207 <!--Cuando se solicita el contenido de data con un mensaje de petición (request message),
1208 el contenido a ser inquirido debe ser especificado concretamente. La data de elementos
1209 básicos, "item", "resource", "lot", "task", "operation", "event", "customer", "supplier" y "order" puede
1210 ser inquirida, también pueden serla la información de plan, "expression" y "parameter"-->
1211 <!--Cuando se solicita data, la etiqueta del elemento a ser inquirido es especificada, y el ID
1212 de la data debe ser especificado en el valor del atributo "name" (nombre); ahora bien, cuando se
1213 utiliza "query" (solicitud) como valor del atributo "name", toda la data que se ajuste a la condición
1214 de devolución, es retornada. Cuando se restringen los elementos solicitados, la condición
1215 de devolución es especificada, para esto, en la etiqueta del elemento solicitado se especifica,
1216 el atributo o subelemento con el nombre "query"-->
1217 <!--Ejemplo, si se buscan los productos cuyos precios (precio unidad) son 1.000.000 Bs y menores
1218 la etiqueta se especifica como sigue-->
1219 <pslx>
1220 <item name="query.">
1221 <price>
1222 <max value="1000"/> <!--condición-->
1223 </price>
1224 <!--Cuando el valor es una cadena de caracteres, varios valores propuestos pueden ser
1225 especificados con un <enumerate>. En el siguiente ejemplo, los colores "amarillo", "azul"
1226 y "rojo" son seleccionados-->
1227 <spec name="color">
1228 <char>
1229 <enumerate value="amarillo"/> <!--condición-->
1230 <enumerate value="azul"/>
1231 <enumerate value="rojo"/>
1232 </char>
1233 </spec>
1234 </item>
1235 </pslx>
1236 <!--Cuando el mismo valor de elemento básico es especificado varias veces,
1237 la relación entre ellos es "OR"-->
1238 <!-- La etiqueta <query> es usada para especificar la data del elemento
1239 que va a ser solicitada. El item seleccionado con el atributo "select" en
1240 esta etiqueta es seleccionado como el ítem a inquirir. La etiqueta de un
1241 subelemento del elemento puede ser especificada también en este atributo.
1242 Si una etiqueta <query> no es especificada, sólo el ID de la data seleccionada
1243 es retornada. No obstante <query> no puede ser especificado para solicitudes de
1244 <parameter> ni <expression>, ya que en estos casos, toda la data es retornada-->
1245 <!--En el siguiente caso, la información acerca de la fecha de comienzo y la
1246 fecha de finalización de la operación "OP001" (ID del elemento básico) y el recurso
1247 asignado para la misma son retornados, a través de la siguiente solicitud-->
1248 <pslx>
1249 <operation name="OP001"> <!--ID-->
1250 <query select="start"/>
1251 <query select="end"/>
1252 <query select="assign"/>
1253 </operation>
1254 </pslx>
1255 <!--Cuando "all" es especificado como valor del atributo "select",
1256 toda la información del elemento es retornada.-->
1257 <!--Cuando "spec" es especificado como valor del atributo "select",
1258 toda la información de especificación es retornada.-->
1259 <!--

```

1260 Inquiring The Contents of Subelement (solicitando el contenido de subelementos)

```
1261 -->
1262 <!--La respuesta a la solicitud anterior es la siguiente:-->
1263 <pslx>
1264 <!--No se puede repetir el ID en este mismo archivo (OP001),
1265 se coloca el punto para poder ejemplificar solicitud y respuesta-->
1266 <operation name="OP001.">
1267 <start>
1268 <time value="2005-11-03T12:00:00"/>
1269 </start>
1270 <end>
1271 <time value="2003-11-03T14:00:00"/>
1272 </end>
1273 <assign resource="máquina-A">
1274 <qty value="1"/>
1275 </assign>
1276 </operation>
1277 </pslx>
1278 <!--Cuando existen algunos subelementos iguales en el elemento básico solicitado,
1279 toda la información de los subelementos es retornada. De cualquier manera, el período
1280 puede ser limitado con el atributo "start" y "end" en una etiqueta <query> o <profile>
1281 para información de series de tiempo tales como: información de especificación,
1282 localización, progreso, carga, almacén y calendario.-->
1283 <!--Lo siguiente expresa que la cantidad de producto-X producido por la máquina-A
1284 por 5 días desde el 7 de noviembre 2005 hasta el 12 de noviembre del mismo año,
1285 es solicitada. -->
1286 <pslx>
1287 <operation name="cálculo" calculation="sum">
1288 <query pre_select="produce" start="2005-11-7T00:00:00" end="2003-04-12T00:00:00"/>
1289 <assign resource="máquina-A"/>
1290 <produce lot="producto-X"/>
1291 </operation>
1292 </pslx>
1293 <!--En tal caso, la información es devuelta con una etiqueta <spec> como resultado
1294 del cálculo-->
1295 <pslx>
1296 <item name="producto-A.">
1297 <spec name="produce">
1298 <qty value="123000"/>
1299 </spec>
1300 </item>
1301 </pslx>
```

1302 <!--
1303 Partial Aggregation (agregación parcial)

```
1304 -->
1305 <pslx>
1306 <lot name="producto-Z">
1307 <query pre_select="consume" start="2005-11-15T00:00:00" end="2005-11-15T01:00:00" calculation="end"/>
1308 <produced operation="operación-A"/>
1309 </lot>
1310 </pslx>
```

1311 <!--
1312 How to Aggregate Discretely (cómo agregar discretamente)

```
1313 -->
1314 <!--El caso donde la cantidad de producto-X producido por la máquina-A
1315 por 5 días, desde el 7 hasta el 11 de noviembre es agregada cada día, se expresa así:-->
1316 <pslx>
1317 <timeScale name="daily" unit="day" type="disc"/>
1318 <operation name="query." calculation="sum">
1319 <query pre_select="produce" scale="daily" start="2003-04-7T00:00:00" end="2003-04-12T00:00:00"/>
1320 <assign resource="máquina-A"/>
1321 <produce lot="producto-X"/>
1322 </operation>
1323 </pslx>
1324 <!--El resultado de la agregación sería-->
1325 <pslx>
1326 <operation name="query" calculation="sum">
1327 <spec name="produce"><qty value="200"/><time count="0"/></spec>
1328 <spec name="produce"><qty value="320"/><time count="1"/></spec>
1329 <spec name="produce"><qty value="140"/><time count="2"/></spec>
1330 <spec name="produce"><qty value="230"/><time count="3"/></spec>
1331 <spec name="produce"><qty value="210"/><time count="4"/></spec>
1332 <spec name="produce"><qty value="130"/><time count="5"/></spec>
1333 </operation>
1334 </pslx>
```

1335 <!--
1336 Calculating by Parameter

```
1337 -->
1338 <!--El parámetro que indica la cantidad promedio del almacén en noviembre es:-->
1339 <pslx>
1340 <parameter name="Valor_promedio_del_almacen_en_noviembre">
1341 <item name="query" calculation="ave">
1342 <query pre_select="stock" calculation="ave" start="2005-11-01T00:00:00" end="2003-12-01T00:00:00"/>
1343 </item>
1344 </parameter>
1345 </pslx>
1346 <!--Cuando el valor promedio del almacén claculado anteriormente es dividido en cada semana
1347 Los diferentes valores son expresado cambiando el valor del atributo "index" en cada período:-->
1348 <pslx>
1349 <parameter name="Valor_promedio_del_almacen_en_noviembre_sem1" index="0" value="340"/>
```

```
1350 <parameter name="Valor_promedio_del_almacen_en_noviembre_sem2" index="1" value="520"/>
1351 <parameter name="Valor_promedio_del_almacen_en_noviembre_sem3" index="2" value="200"/>
1352 <parameter name="Valor_promedio_del_almacen_en_noviembre_sem4" index="3" value="170"/>
1353 </pslx>
1354 </ppap>
1355
```

www.bdigital.ula.ve

ANEXO D

DEFINICIÓN DE LAS INTERFACES SEGÚN SU UBICACIÓN EN CADA SUBSISTEMA

www.bdigital.ula.ve

**[ANEXO D-1]
[Cuadro de Interfaces]**

Cuadro 1

Interfaces Plan

SetPlan	GetPlan	Origen-Iniciador	Destino
Especificar información de pronóstico del cliente (Ventas).	Mostrar la programación de suministro de PPAP.	Ventas.	PPAP.
Especificar la programación de suministro del suplidor (Compras) en PPAP.	Proveer la planificación de producción de PPAP a un suplidor (Compras).	Compras.	PPAP.
Especificar el parámetro de “plan” específico acerca de Diseño, en PPAP.	Inquiere el parámetro de “plan” específico de PPAP.	Diseño	PPAP
Especificar el parámetro de “plan” específico acerca de Producción, en PPAP.	Inquiere el parámetro de “plan” específico de PPAP.	Producción	PPAP
Muestra la programación de suministro creada por PPAP, a un cliente (Ventas).	Inquiere la planeación de demanda poseída por un cliente (Ventas).	PPAP	Ventas
Especifica el “plan” de producción creado por PPAP a un suplidor (Compras)	Inquiere la información de programación de suministro de un suplidor (Compras).	PPAP	Compras
Especifica la información acerca de un parámetro de Diseño.	Inquiere información acerca de un parámetro de Diseño.	PPAP	Diseño
Especifica la información acerca de un parámetro de Producción o manufactura.	Inquiere información acerca de un parámetro Producción o manufactura.	PPAP	Producción

**[ANEXO D-2]
[Cuadro de Interfaces]**

Cuadro 2

Interfaces Programa

SetPrograma	GetPrograma	Origen-Iniciador	Destino
Especificar el programa del cliente (Ventas) en PPAP.	Mostrar El programa de PPAP a un cliente (Ventas).	Ventas.	PPAP.
Especificar el programa de un proveedor (Compras) en PPAP y usar el encadenamiento. Hacer preparaciones para recibir productos, enviando el programa-de-envío.	Mostrar las partes relacionadas del programa en PPAP a un proveedor (Compras).	Compras.	PPAP.
Especificar el programa de operaciones de Diseño en PPAP.	Inquiere el programa creado por PPAP. La operación de Diseño puede estar incluida en el programa inquirido.	Diseño	PPAP
Especificar el programa de demanda originalmente creado o el programa definitivo, en PPAP.	El lado de manufactura adopta el programa creado por PPAP.	Producción	PPAP
Especifica el programa creado por PPAP para un cliente (Ventas). Este mensaje puede ser utilizado para enviar el programa-de-envío, cuando se haga el envío	Inquiere el programa de producción poseído por un cliente (Ventas).	PPAP	Ventas
Especifica el programa de producción para un proveedor (Compras)	Inquiere el programa de producción de un proveedor (Compras)	PPAP	Compras
Especifica el programa de Diseño originalmente creado en PPAP.	Especifica el programa de producción, incluyendo Diseño.	PPAP	Diseño
Especifica el programa de manufactura o Producción.	Inquiere el programa de producción originalmente creado por el lado de manufactura o Producción.	PPAP	Producción

[ANEXO D-3]
[Cuadro de Interfaces]

Cuadro 3

Interfaces Producto

SetProducto	GetProducto	Origen-Iniciador	Destino
No existe.	Mostrar la información de producto a la petición de un cliente (Ventas).	Ventas.	PPAP.
Registrar la información de producto poseída por un suplidor (Compras) como el catálogo de información o lista de precios, en PPAP.	El suplidor (Compras) pregunta a PPAP información de producto para la producción, si es necesario.	Compras.	PPAP.
Especifica la información de producto en PPAP, como resultado de un Diseño. El BOM de Producción es especificado aquí.	Inquirir información de producto especificada en PPAP.	Diseño	PPAP
Especificar en PPAP la información de producto (información del BOM de manufactura) requerida en el lado de Producción.	Adoptar la información de producto de PPAP (información del BOM de manufactura).	Producción	PPAP
Registrar la información de catálogo de productos de la propia compañía, para un cliente (Ventas).	No existe.	PPAP	Ventas
Especificar la información acerca del producto producido a un suplidor. Ambos SetProducto y SetProceso pueden mandar el BOM de manufactura.	Inquirir la información de un producto de un suplidor.	PPAP	Compras
No existe.	Inquirir el modelo de un producto como resultado de un diseño. Ambos GetProducto y GetProceso pueden inquirir el BOM de manufactura.	PPAP	Diseño

No existe.	Inquirir la información de producto originalmente diseñada por el lado de manufactura (Producción). Ambos GetProducto y GetProceso pueden inquirir el BOM de manufactura.	PPAP	Producción
Describir información sobre un producto, su especificación detallada.	No existe.	Diseño	Ventas
No existe.	Inquirir información sobre un producto, su especificación detallada.	Ventas	Diseño
Describir información sobre un producto, su especificación detallada de diseño.	Inquirir la información de producto originalmente diseñada por el lado de manufactura (Producción).	Diseño	Producción
No existe.	Inquirir la información de producto originalmente diseñada por el lado de Diseño.	Producción	Diseño
Registrar la información de producto poseída por un suplidor (Compras).	Inquirir la información de producto originalmente diseñada por el lado de Diseño.	Compras	Diseño

**[ANEXO D-4]
[Cuadro de Interfaces]**

Cuadro 4

Interfaces Proceso

SetProceso	GetProceso	Origen-Iniciador	Destino
No existe.	Proveer la información de procesos de manufactura a la petición, tales como: las instalaciones poseídas.	Ventas.	PPAP.
El proveedor (Compras) registra la información de procesos de producción en PPAP, tales como: su propio equipamiento y la ingeniería más fuerte.	El proveedor quiere la información de procesos de producción de PPAP, para la manufactura.	Compras.	PPAP.
Especificar la información de procesos de producción como resultado de un diseño. La información de los recursos de producción y las formas de producción, son especificadas aquí.	Inquirir la información de procesos de producción de PPAP	Diseño	PPAP
Especificar la información relacionada con procesos de producción, tal como: equipos originalmente requeridos en el lado de manufactura o restricciones.	Adoptar la información relacionada con procesos de producción tal como: equipamiento o restricciones.	Producción	PPAP
Registrar el equipamiento o los procesos de producción de la propia compañía, para un cliente (usando a Ventas).	No existe.	PPAP	Ventas
Especificar la información de procesos de producción a tener, el contenido de producción para un proveedor (Compras). La información de operación y la información de eventos pueden	Inquirir la información de procesos de producción para un proveedor (Compras).	PPAP	Compras

ser enviadas. Tanto SetProducto como SetProceso pueden enviar el BOM de producción.			
No existe.	Inquirir la información de procesos de producción como resultado de un diseño. Ambos GetProducto y GetProceso pueden inquirir la información del BOM de manufactura.	PPAP	Diseño
No existe.	Inquirir la información de procesos de producción acerca de instalaciones u operaciones, originalmente diseñadas por el lado de manufactura. Ambos GetProducto y GetProceso pueden inquirir la información del BOM de manufactura.	PPAP	Producción
No existe.	Inquirir la información de procesos de producción como resultado de un diseño. Puede inquirir la información del BOM de manufactura.	Producción	Diseño
Registrar la información de procesos de producción como resultado de un diseño. Puede enviar la información del BOM de manufactura.	Inquirir los procesos utilizados para la manufacturación.	Diseño	Producción

**[ANEXO D-5]
[Cuadro de Interfaces]**

Cuadro 5

Interfaces Orden

SetOrden	GetOrden	Origen-Iniciador	Destino
Especificar una orden de un cliente (Ventas). La orden del cliente incluye la precisión de una orden definitiva o una orden no-oficial.	El cliente inquiriere en PPAP, el tiempo de entrega de una orden o la cantidad de productos que pueden ser provistos.	Ventas.	PPAP.
El suplidor (Compras) promete un tiempo de entrega y una cantidad de producción a PPAP.	El suplidor (Compras) inquiriere la información de orden para su propia compañía. Las órdenes son: una orden no-oficial aparte de una orden definitiva.	Compras.	PPAP.
Notificar al contenido que las opciones se decidieron en el diseño.	Recibir la petición a diseñar. Cuando el diseño es requerido para la orden del cliente, PPAP crea una petición de diseño al lado de Diseño.	Diseño	PPAP
Pide a PPAP reflejar la operaciones necesarias (como mantenimiento) en el lado de manufactura del plan completo.	Adoptar de PPAP, la diversa información de programas de producción para Manufactura (Producción) tal como: programa de producción, programa de envío, instrucción de liberación y otras operaciones de mantenimiento.	Producción	PPAP
No existe.	Extraer la orden desde un cliente (Ventas) a la propia compañía.	PPAP	Ventas
Publicar una orden a un proveedor (Compras).	No existe.	PPAP	Compras
Enviar la solicitud de diseño a (Diseño) basada en el plan base creado por PPAP	No existe.	PPAP	Diseño

<p>Especificar el orden de producción para el lado de manufactura (Producción). La información de “cambio de orden” y “cancelación de orden” puede ser enviada para el cambio o cancelación de la orden respectivamente.</p>	<p>Recibir la orden solicitada desde el lado de manufactura. Los objetivos son órdenes de mantenimiento, programa de producción, etc.</p>	<p>PPAP</p>	<p>Producción</p>
<p>Especificar el orden de diseño de un producto, para fines promocionales y publicitarios.</p>	<p>Inquirir información respecto de las necesidades de productos y/o servicios (en materia de Diseño) en el mercado, usando como eje las cantidades y tipos de órdenes que se realizan.</p>	<p>Diseño</p>	<p>Ventas</p>
<p>Suministra información respecto de las necesidades de productos y/o servicios del mercado (en materia de Diseño), usando como eje las cantidades y tipos de órdenes que se realizan.</p>	<p>Inquirir información del orden de diseño de un producto, para fines promocionales y publicitarios.</p>	<p>Ventas</p>	<p>Diseño</p>

**[ANEXO D-6]
[Cuadro de Interfaces]**

Cuadro 6

Interfaces Opción

SetOpción	GetOpción	Origen-Iniciador	Destino
<p>Especificar la opción en una orden, adicionalmente a la orden. Esta opción se utiliza cuando la opción es decidida después que la orden.</p>	<p>No existe.</p>	<p>Ventas.</p>	<p>PPAP.</p>
<p>No existe.</p>	<p>El proveedor o suplidor (Compras) inquiera los datos de opción de la orden para su propia compañía. Cuando la opción no está decidida, este mensaje puede además solicitar la decisión de los datos de opción.</p>	<p>Compras.</p>	<p>PPAP.</p>
<p>Diseñar la opción para satisfacer la solicitud de cada orden, y especificar los resultados</p>	<p>Inquirir el contenido de la opción especificada por el cliente mismo (Ventas). Este mensaje puede ser requerido para el diseño de otras partes.</p>	<p>Diseño</p>	<p>PPAP</p>
<p>No existe.</p>	<p>Adoptar la información de datos de opción, como contenido del programa de producción proveniente de PPAP.</p>	<p>Producción</p>	<p>PPAP</p>
<p>Especificar las opciones de precio, promoción, producto y plaza a Diseño.</p>	<p>Inquirir el contenido de las opciones de diseño disponibles para un producto.</p>	<p>Ventas</p>	<p>Diseño</p>
<p>Suministrar información del contenido de las opciones de diseño disponibles para un producto.</p>	<p>Inquirir información de precio, promoción, producto y plaza (las cuatro P) para tomarla en consideración en los diseños.</p>	<p>Diseño</p>	<p>Ventas</p>

**[ANEXO D-7]
[Cuadro de Interfaces]**

Cuadro 7

Interfaces Estimación

SetEstimación	GetEstimación	Origen-Iniciador	Destino
No existe.	El cliente (Ventas) solicita a PPAP estimar un precio, cantidad y tiempo de entrega de un producto.	Ventas.	PPAP.
El proveedor (Compras) especifica en PPAP, información de estimación tal como tiempo de entrega.	No existe.	Compras.	PPAP.
Especifica información de estimación como: producción hora-persona o precio, dada una solicitud o petición.	No existe.	Diseño	PPAP
Especificar a PPAP, el tiempo requerido para la producción y la cantidad de productos producibles.	No existe.	Producción	PPAP
No existe.	Obtener información de estimación del proveedor (Compras) acerca del tiempo de entrega o cantidad.	PPAP	Compras
No existe.	Inquirir la información de estimación acerca de horas-persona, tiempo de entrega y cantidad de productos producibles para producción.	PPAP	Producción

**[ANEXO D-8]
[Cuadro de Interfaces]**

Cuadro 8

Interfaces Queja

SetQueja	GetQueja	Origen-Iniciador	Destino
Mostrar la queja contra un producto o servicio recibido por un cliente (Ventas), a PPAP.	No existe.	Ventas.	PPAP.
No existe.	Recibir la información de queja contra un proveedor (Compras), desde PPAP.	Compras.	PPAP.
Especificar mensaje cuando existe un problema en el proceso de diseño que nunca puede ser solucionado, y la base de solicitud del cliente debe ser cambiada.	Recoger la información de queja desde PPAP y tomar las medidas necesarias.	Diseño	PPAP
Especificar este mensaje en PPAP cuando existe cualquier defecto en la orden recibida	Recoger, del cliente o de otros, la información relacionada a defectos, y luego tomar las medidas contra el defecto.	Producción	PPAP

**[ANEXO D-9]
[Cuadro de Interfaces]**

Cuadro 9

Interfaces Progreso

SetProgreso	GetProgreso	Origen-Iniciador	Destino
El cliente (Ventas) reporta el progreso del suministro y recibimiento de un producto, a PPAP.	El cliente (Ventas) pregunta el progreso de la orden relacionada u operaciones, a PPAP.	Ventas.	PPAP.
Especificar la información del progreso de producción actual in PPAP, para la orden recibida.	Inquirir el hecho de que un producto es suministrado por un proveedor, y es aceptado.	Compras.	PPAP.
Especificar el progreso para la operación de diseño instruida por PPAP.	No existe.	Diseño	PPAP
Especificar el progreso de las condiciones relacionadas con operaciones en PPAP, tales como: progreso de producción, progreso de culminación y progreso de inspección.	No existe.	Producción	PPAP
Especificar las condiciones de progreso a un cliente (Ventas). Este mensaje puede ser utilizado para enviar la información acorde al despacho de productos.	Preguntar al cliente (Ventas) los resultados de operaciones como suministro o aceptación de un producto.	PPAP	Ventas
Reportar al surtidor (Compras) el estado de suministro o el estado de recepción, de productos recibidos.	Inquirir la información relacionada a los progresos de producción de un proveedor (Compras).	PPAP	Compras
No existe.	Inquirir el estado de progreso del diseño.	PPAP	Diseño
No existe.	Pregunta el estado de progreso al lado de manufactura.	PPAP	Producción

**[ANEXO D-10]
[Cuadro de Interfaces]**

Cuadro 10

Interfaces Emergencia

SetEmergencia	GetEmergencia	Origen-Iniciador	Destino
Notificar a PPAP que emergencias como problemas y retraso en la producción ocurren durante la producción del proveedor.	No existe.	Compras.	PPAP.
Notificar PPAP del estado en necesidad de medidas urgentes, por la ocurrencia de serios errores de diseño.	No existe.	Diseño	PPAP
Notificar a PPAP que una emergencia inesperada ocurre, como fallas de equipo, problemas de calidad y un gran retraso en la producción.	No existe.	Producción	PPAP

**[ANEXO D-11]
[Cuadro de Interfaces]**

Cuadro 11

Interfaces Almacén

SetAlmacén	GetAlmacén	Origen-Iniciador	Destino
Especificar el estado del almacén del producto provisto por PPAP, que el cliente tiene actualmente.	El cliente (Ventas) pregunta el valor del almacén presente o futuro, poseído por PPAP.	Ventas.	PPAP.
El reporta a PPAP, el “valor del almacén” de los ítems poseídos por el proveedor (Compras) y que son suministrados a PPAP.	El proveedor (Compras) inquiriere el “valor del almacén” administrado por PPAP.	Compras.	PPAP.
Especificar el “valor revisión actual del almacén” o el “valor de pronóstico” para el estado del almacén.	Inquirir el valor teórico del almacén creado por PPAP.	Producción	PPAP
No existe.	Inquirir el valor del almacén de los productos de la propia compañía, en el almacén de materiales del cliente (Ventas).	PPAP	Ventas
No existe.	Inquirir el valor presente o futuro del almacén, poseído por un proveedor (Compras).	PPAP	Compras
No existe.	Inquirir la información de almacén tomada por el lado de manufactura.	PPAP	Producción

**[ANEXO D-12]
[Cuadro de Interfaces]**

Cuadro 12

Interfaces Carga

SetCarga	GetCarga	Origen-Iniciador	Destino
No existe.	El cliente (Ventas) pregunta a PPAP la carga presente o futura.	Ventas.	PPAP.
El surtidor o proveedor registra su carga en PPAP.	No existe.	Compras.	PPAP.
Especificar en PPAP, el valor actual o el valor pronosticado de la carga de los recursos de producción, como equipamiento o trabajadores. El progreso de operación es especificado usando este mensaje.	Inquirir el valor teórico de carga de recursos especificado por PPAP.	Producción	PPAP
No existe.	Inquirir la carga presente o futura de un proveedor (Compras).	PPAP	Compras
No existe.	Inquirir la información de carga tomada del lado de manufactura.	PPAP	Producción

**[ANEXO D-13]
[Cuadro de Interfaces]**

Cuadro 13

Interfaces Capacidad

SetCapacidad	GetCapacidad	Origen-Iniciador	Destino
No existe.	El cliente (Ventas) pregunta la capacidad de producción de PPAP. La capacidad de producción corresponde a un “puesto” en la producción.	Ventas.	PPAP.
Registrar la capacidad de producción de suplidor en PPAP.	No existe.	Compras.	PPAP.
El lado de manufactura o Producción específica en PPAP, la capacidad de producción presente o futura.	Inquirir el valor teórico de la capacidad de producción creada por PPAP.	Producción	PPAP
No existe.	Inquirir la capacidad de producción presente o futura de un proveedor. La capacidad e producción corresponde a un “puesto de producción” en la producción “tipo-booking”.	PPAP	Compras
No existe.	Inquirir la información de capacidad de producción tomada por el lado de manufactura.	PPAP	Producción

[ANEXO D-14]
[Cuadros de Interfaces]

Cuadro 14

Interfaces Lote

SetLote	GetLote	Origen-Iniciador	Destino
Especificar en PPAP, información relacionada con “lote de producción”.	Inquirir la información relacionada con el “lote de producción” poseído por PPAP.	Producción	PPAP
No existe.	Inquirir información de “lote de producción”.	PPAP	Producción

Cuadro 15

Interfaces Tarea

SetTarea	GetTarea	Origen-Iniciador	Destino
Especificar en PPAP, la información relacionada con “tarea”.	Inquirir la información relacionada con la “tarea”, poseída por PPAP.	Producción	PPAP
No existe.	Inquirir la información de tarea para recursos.	PPAP	Producción

[ANEXO D-15]
[Cuadro de Interfaces]

Cuadro 16

Interfaces Innovación

SetInnovación	GetInnovación	Origen-Iniciador	Destino
Registrar la información de los productos más innovadores, los más actualizados, que se encuentran o podrían salir al mercado.	Buscar una propuesta en diseño, para competir con los productos innovadores.	Ventas	Diseño
Especificar la información acerca de los nuevos materiales y partes que se necesitan para sustituir los utilizados en la fabricación de los productos actuales, o para elaborar nuevos.	No existe.	Compras	Diseño
No existe.	Pedir al surtidor o proveedor (Compras), información acerca de materiales y procesos de fabricación innovadores.	Diseño	Compras
Especificar la información acerca de las nuevas estructuras, herramientas y especificaciones que se necesitan para sustituir los procesos de fabricación actuales o para elaborar nuevos.	Inquirir una respuesta respecto a la asimilación de la información de innovación recibida y la factibilidad de las líneas de producción actuales a estas nuevas herramientas.	Diseño	Producción
Registrar en Diseño una respuesta respecto a la asimilación de la información de innovación recibida, y la factibilidad de las líneas de producción actuales a estas nuevas herramientas.	Inquirir información acerca de las nuevas estructuras, herramientas y especificaciones de procesos de fabricación.	Producción	Diseño

ANEXO E

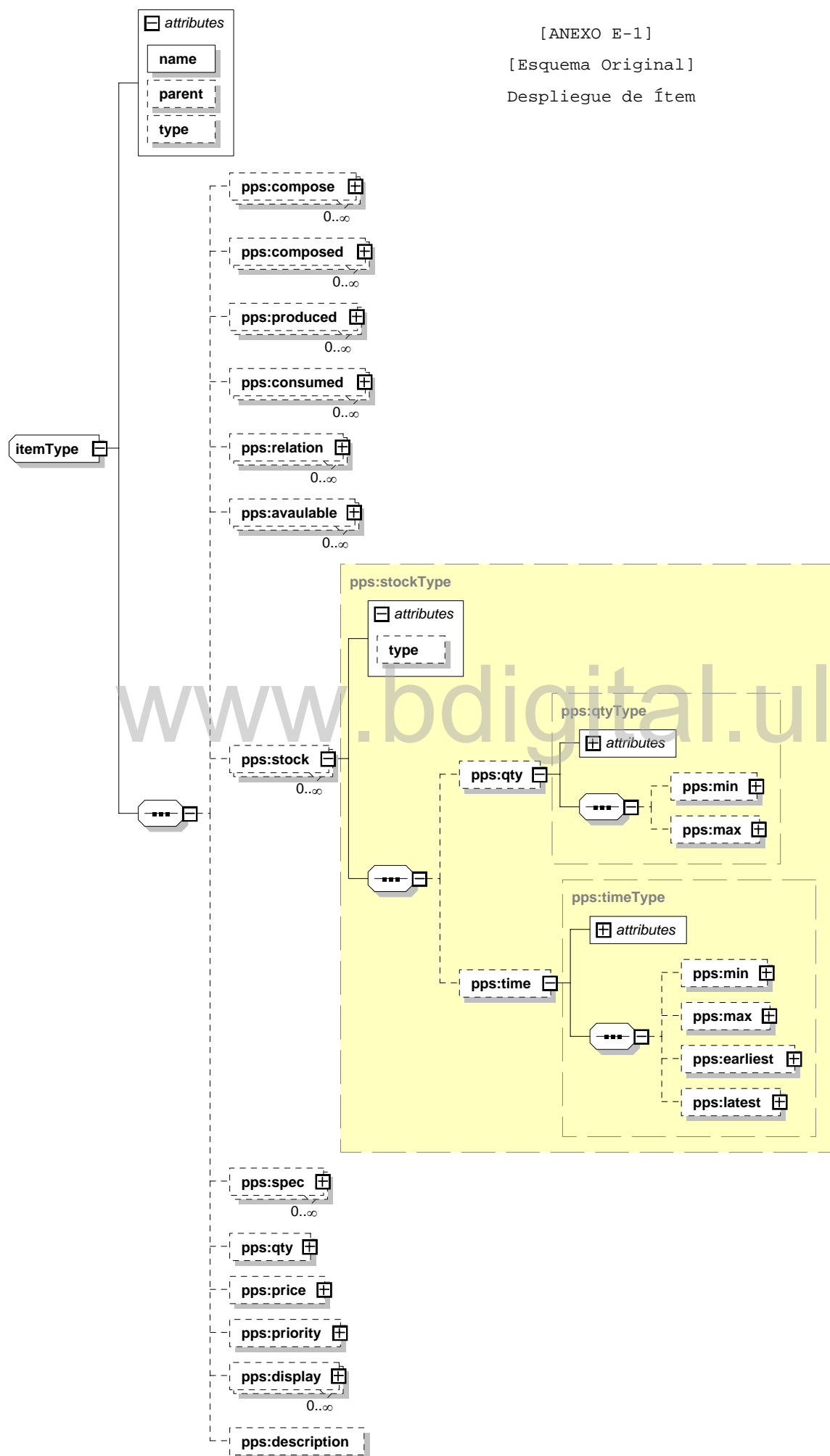
ALGUNOS ESQUEMAS DESPLEGADOS

www.bdigital.ula.ve

[ANEXO E-1]

[Esquema Original]

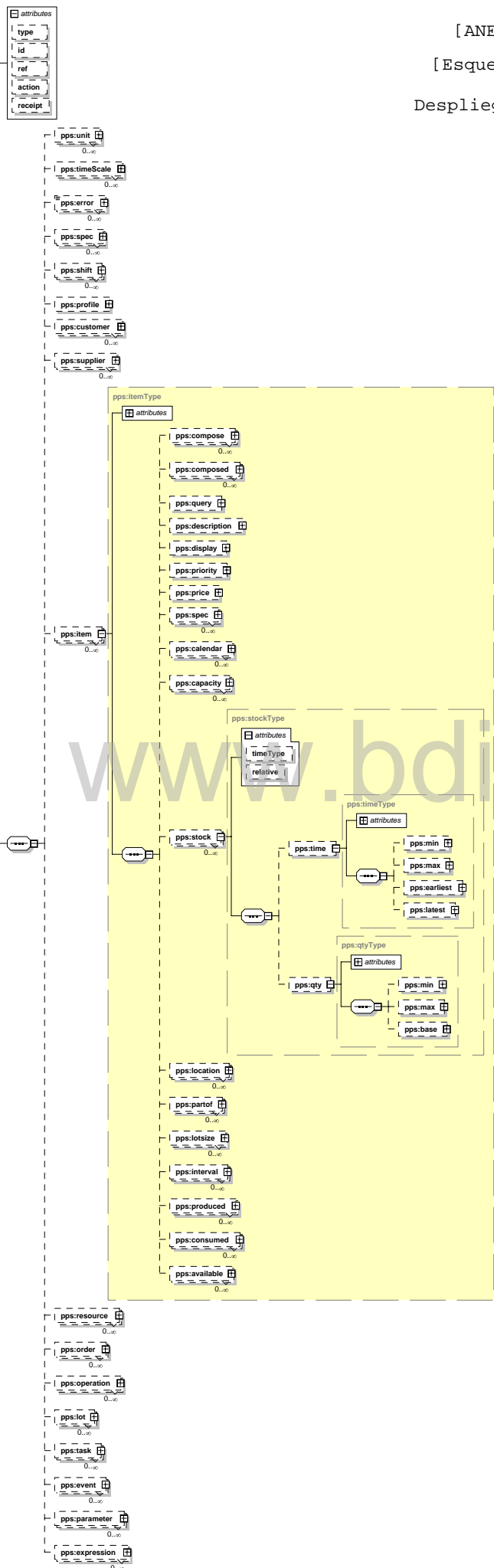
Despliegue de ítem



[ANEXO E-2]

[Esquema Final]

Despliegue de "Ítem"



www.bdigital.ula.ve

[ANEXO E-3]

[Esquema Final]

Elementos del 1er Nivel de Implementación

