

UNIVERSIDAD DEL PAPALOAPAN



B2BOM **Business to Business Object Manager**

TESIS

Licenciatura en Ingeniería en Computación

PRESENTA

C. José Francisco Aburto Camacho

DIRECTOR DE TESIS

M.C. Eduardo Ortiz Hernández

UNIVERSIDAD DEL PAPALOAPAN



INGENIERÍA EN COMPUTACIÓN

LA PRESENTE TESIS TITULADA "Business to Business Object Manager" PRESENTADA POR EL SUSTENTANTE DE LICENCIATURA C. JOSÉ FRANCISCO ABURTO CAMACHO BAJO LA DIRECCIÓN DEL M.C. EDUARDO ORTIZ HERNÁNDEZ HA SIDO REVISADA Y ACEPTADA POR EL COMITÉ EXAMINADOR PARA SER DEFENDIDA EN EL EXAMEN PROFESIONAL Y OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN.

MC. EDUARDO ORTIZ
HERNÁNDEZ
Director de tesis

MC. ARIEL LÓPEZ
RODRÍGUEZ
Sinodal

DR. EDUARDO SÁNCHEZ
SOTO
Sinodal

MC. JOSÉ ANTONIO
ROSALES BARRALES
Sinodal

DEDICATORIA

A mis padres Arturo Aburto Zarate y Graciela Camacho Delgado que en todo momento siempre me alientan a seguir adelante, todo lo que soy se lo debo a ellos, por inculcar en mí la importancia de estudiar y ser un profesional.

A mi esposa Rosario Martínez Zúñiga que ha sido incondicional en motivarme, siempre ver las cosas positivamente y alentarme en cada momento.

A mi hijo Ian Adrián Aburto Martínez que ha sido la fuente y motor de inspiración para que finalice este proyecto.

A mi tío el Doctor Eliseo Camacho Delgado por siempre apoyarme, alentarme desde mi infancia hasta la carrera y siempre apoyar a mis padres.

AGRADECIMIENTOS

A Dios.

Por darme la paciencia y sabiduría para terminar esta etapa académica.

A mis padres.

Por siempre apoyarme en todo momento, estar al pendiente de mis estudios y por darme el regalo más importante de mi vida.

A profesores investigadores de la universidad.

Por ser pacientes y tener siempre la disponibilidad de enseñar, mostrando siempre gran profesionalismo en sus actividades.

Al M.C. Eduardo Ortiz Hernández.

Por aceptar ser el director de tesis, estar al pendiente de mis avances y revisiones.

Al M.C. Ariel López Rodríguez, M.C. Sergio Fabián Ruiz Paz, M.C. Eduardo Ortiz Hernández y M.C. José Antonio Rosales Barrales.

Fueron fundamentales sus enseñanzas académicas para implementar este proyecto y haber tenido éxito siendo el primero en mi vida profesional, sin mencionar al resto de los profesores que fueron el complemento de mi formación profesional y que durante el camino me apoyaron para terminar la carrera.

A familiares.

Por estar al pendiente de mis actividades académicas y personales, siempre alentándome para terminar mis estudios y ser ejemplo para las futuras generaciones.

A todos ustedes mi mayor reconocimiento y gratitud.

RESUMEN

Un sistema Business to Business (B2B) es aquel que gestiona documentos de forma electrónica para intercambiar información entre el ofertante y el consumidor; los datos de estos documentos ya están establecidos con anterioridad para concretar las reglas de negocio (por ejemplo, artículos, precios, tiempo estimado de entrega, datos de facturación, tipo de embarque, entre otros) al momento de enviar dicha información y evitando que exista un debate cuando se entregue la mercancía. El uso de sistemas B2B reduce el tiempo de gestión entre la compra y venta, ya que este proceso elimina posibles ocurrencias de errores humanos. Existen páginas Web conocidas que utilizan sistemas B2B en sus procedimientos, por ejemplo MercadoLibre, Amazon, entre otros.

Para intercambiar estos documentos electrónicamente es importante utilizar un sistema que realice esta tarea, ya sea de forma manual o automática. Existen varios canales para enviar documentos entre computadoras, por ejemplo, "IBM WebSphere Partner Gateway Express" mejor conocido como AS2 [4], cuya función principal es realizar dicho envío encriptado y firmado por certificados utilizando protocolos que mejor se adapten a las necesidades de los clientes.

Hoy en día existe un portal Web B2BOM (Business To Business Object Manager) realizado por la empresa consultora y desarrolladora de software para los proveedores de un cliente, donde se recibe una orden de compra y se genera un archivo embarque por orden de compra, con el fin de entregar la mercancía al Centro de Distribución de Mercaderías (MDC) del cliente. Este portal B2BOM tendrá un ajuste en el proceso que el cliente está implementando, y esto implica un ajuste en sus sistemas de software relacionados en su estructura y funcionamiento. El cliente solicita a la empresa de desarrollo que realice o incluya en el sistema los cambios requeridos en el proceso de entrega de la mercancía a sus centros de distribución; para ello ambas compañías se van a reunir para comprender con más detalle y analizar el nuevo proceso.

Actualmente, el sistema de software que se tiene funcionando con los proveedores no tiene la estructura o el diseño para modificaciones a futuro, incluye código basura, genera conexiones abiertas y una base de datos no relacionada. Es por esto que se llegó a la conclusión de desarrollar un nuevo sistema aprovechando el foro de un proceso nuevo, aplicando un modelo para el desarrollo del software y corrigiendo las limitaciones antes mencionadas.

ABSTRACT

A Business to Business system (B2B) is one that manages documents electronically to exchange information between the vendor and the consumer; this documents data are already established to specify business rules (for example articles, prices, estimated time of delivery, billing, type of shipment, etc.) when exchanging information and avoiding that a debate exists when the goods are delivered. Using B2B systems reduce the management time between buying and selling, as this process eliminates possible occurrences of human errors. There are known Web pages using a B2B exchange information such as MercadoLibre, Amazon and others.

In order to exchange these documents electronically it is important to use a system that performs this task either manually or automatically. There are several channels for sending documents between computers, for example, "IBM WebSphere Partner Gateway Express" better known as AS2 [4], whose main function is to send encrypted documents and certificates signed by using protocols that best suit the needs.

Nowadays exists a Web portal B2BOM (Business To Business Object Manager) conducted by the consulting and developer of Software Company, for use suppliers when a purchase order is received and a shipment file for purchase order is generated, in order to deliver the goods to Merchandise Distribution Center (MDC). This website will be adjusted in the process that the client is implementing, and this involves a setting in the software systems related in its structure and operation. The client requested to the development software company to perform this system changes required in the process of delivering the goods to its distribution centers; for this both companies will be meeting to obtain a complete analysis of the new process.

Currently, the software system working with suppliers does not have the structure or design for future modifications, includes garbage code, and generates open connections and a database disorganized. Therefore it is concluded to develop a new system including this new process, applying a software development model and correcting above limitations.

ÍNDICE GENERAL

CAPÍTULO 1. INTRODUCCIÓN	11
1.1 Antecedentes	12
1.2 Planteamiento del problema	15
1.3 Objetivo general	20
1.3.1 Objetivos específicos	20
1.4 Hipótesis	20
1.5 Justificación	20
1.6 Alcance y limitaciones	21
CAPÍTULO 2. MARCO TEÓRICO	25
2.1 Metodología de software	25
2.2 Sistema web	27
2.3 Base de datos	28
2.4 Arquitectura J2EE	29
2.5 Servidor de aplicaciones	31
2.5.1 Apache tomcat	31
2.6 Recursos y tiempo de la empresa desarrolladora	32
2.7 Otros sistemas B2B	34
CAPÍTULO 3. ANÁLISIS	36
3.1 Proceso embarque	36
3.2 Órdenes de compra e integración con su sistema interno	37
3.3 Documentos de intercambio electrónico	38
3.4 Personalizaciones	42
3.5 Módulos	43
3.6 Casos de uso	43
CAPÍTULO 4. DISEÑO	45
4.1 Diseño de alto nivel	45
4.2 Patrón de diseño Struts	46
4.3 Diagrama de paquetes	48
4.4 Diseño de la base de datos	52
4.5 Diseño de interfaz	54
CAPÍTULO 5. DESARROLLO	59
5.1 Puesto en desarrollo	59
5.2 Diagrama de clases	59
5.3 Colaboración y revisiones	69
CAPÍTULO 6. PRUEBAS	71
6.1 Pruebas internas	71
6.2 Pruebas de cliente solicitante	71
6.3 Pruebas de cliente piloto	71

CAPÍTULO 7. PUESTO EN PRODUCCIÓN	76
7.1 Soporte	76
CAPÍTULO 8. CONCLUSIONES	82
8.1 Conclusión	82
8.2 Trabajos futuros	83
9. REFERENCIAS	85
10. ANEXOS	87
10.1 PDF de la orden de compra.	87
10.2 PDF del Packing List	88
10.3 PDF Etiqueta.	89
10.4 XML de la orden de compra.	90
10.5 XSD de la orden de compra.	91
10.6 XML de Acuse.	93
10.7 XML del Booking	93
10.8 XSD del Booking	94
10.9 XML de estatus del Booking	95
10.10 XSD del estatus del Booking	95
10.11 XML del ASN	96
10.12 XSD del ASN	97
10.13 XML del estatus del ASN	98
10.14 XSD del estatus del ASN	98
10.15 XML del Consolidado	99
10.16 XSD del Consolidado	99
10.17 XML del estatus de Consolidado	100
10.18 XSD del estatus del Consolidado	100
10.19 XML del Order plan	101
10.20 XSD del Order Plan	103
10.21 PDF del Order Plan	104

ÍNDICE DE FIGURAS

Figura 1. Esquema de función B2BOM	14
Figura 2. Proceso actual	15
Figura 3. Proceso nuevo	16
Figura 4. Modelo en cascada	26
Figura 5. Arquitectura J2EE	30
Figura 6. Apache Tomcat	32
Figura 7. Plan de trabajo	33
Figura 8. Plan de actividades	34
Figura 9. Proceso embarque	37
Figura 10. Proceso de la orden de compra	39
Figura 11. Proceso Booking	40
Figura 12. Proceso ASN	41
Figura 13. Proceso Consolidado	42
Figura 14. Casos de uso	44
Figura 15. Diseño de sistema	45
Figura 16. Struts 2.0	47
Figura 17. Diagrama de paquetes de clases	49
Figura 18. Diagrama de paquetes Web	51
Figura 19. Base de datos	53
Figura 20. Inicio de sesión	55
Figura 21. MDC	55
Figura 22. Menús	56
Figura 23. Booking	56
Figura 24. ASN	57
Figura 25. Consolidado	57
Figura 26. Consolidado PDF	58
Figura 27. Diagrama de clases XML	60
Figura 28. Diagrama de clases PDF	61
Figura 29. Diagrama de clases	63
Figura 30. Diagrama de administración	64
Figura 31. Diagrama de filtros	65
Figura 32. Diagrama de clases Actions	66
Figura 33. Diagrama de clases DAO parte 1	67
Figura 34. Diagrama de clases DAO parte 2	68
Figura 35. Diagrama de clases DAO parte 3	68
Figura 36. Diagrama de clases DAO parte 4	69

ÍNDICE DE TABLAS

Tabla 1. Diccionario de tablas no relacionadas	54
Tabla 2. Diferencias de clases	61
Tabla 3. Tipos de sistema operativo	72
Tabla 4. Versiones de base de datos	72
Tabla 5. Características del servidor	72

CAPÍTULO 1. INTRODUCCIÓN

El intercambio electrónico de información hoy en día es una ventaja competitiva en los negocios; transferir datos entre compañías vinculadas hace que el proceso de información sea más rápido y ágil, como por ejemplo, la factura electrónica, las órdenes de compra, desgloses de pago, acuses de recibo y de envío, etcétera. Existen herramientas de envío que hacen este proceso más fácil, como por ejemplo Web Services y AS2. Cada envío de información electrónico entre compañías necesita un formato y estructura de archivo que tanto el emisor como el receptor de información entienda o interprete, como por ejemplo, XML, TXT, MOD, CSV, etcétera. Para realizar un sistema es importante analizar estos archivos y su estructura con el fin de gestionar la información y realizar un tratamiento adecuado para uno o más procesos dentro del sistema.

Las órdenes de compra junto con la factura electrónica son unos de los documentos que más demanda tienen para el intercambio electrónico; especialmente nos enfocaremos en la orden de compra y solo se hará mención de la facturación como documento de intercambio; en el proceso de la orden de compra el cliente envía órdenes de compra a sus proveedores por medio del software AS2 donde el proveedor lo recibe en su servidor de computadora, lo traduce y valida para realizar posteriormente el embarque de esa orden de compra, regresando un archivo al cliente indicando que se va a enviar de la orden de compra previamente enviada; es nuestra tarea realizar un sistema B2B (Business to Business) que reciba, gestione y envíe información entre el cliente y proveedor por medio de un sistema AS2.

Existen sistemas de software desarrollados de forma general que se adaptan a las necesidades de varios clientes y sistemas exclusivos que corresponden a un solo cliente adaptándose completamente a las restricciones implementadas. El sistema B2B que se va a desarrollar es exclusivo del cliente y solo funciona para sus proveedores y no cumple las reglas de negocio con otras cadenas comerciales; aunque el proveedor tiene la oportunidad de adquirir el sistema B2B o desarrollar uno propio que realice las validaciones correspondientes aplicadas por el cliente para embarcar la mercancía. Adicional el sistema B2B no es un software libre por lo que tiene una licencia y soporte anual desde el momento en que se adquiere el producto.

Web Services.- Es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

AS2.- Software que encripta y firma documentos de intercambio electrónico entre computadoras.

1.1 ANTECEDENTES

Las órdenes de compra existen desde hace tiempo, sin la necesidad de un papel de por medio; se realiza un vínculo entre el vendedor quien expone la mercancía al público y el comprador a quien le interesa el producto. En un principio de forma apalabrada únicamente, y tiempo después mediante el papel se especificaba la orden de compra como un comprobante, que ayudaba tanto al vendedor como al comprador tener un historial de venta y compra correspondiente para una mejor administración de los productos; a su vez, con la revolución electrónica se empezó a sustituir el papel por la información electrónica hasta los años 90s. Los grandes pedidos son enviados electrónicamente por vías como correo electrónico, archivos TXT, Excel o Word entre otros.

Detrás de un pedido (orden de compra) que solicita el cliente existe un gran número de procesos que intervienen para la compra y venta de mercancía. El flujo general se resume en los siguientes tres pasos:

1. Orden de compra.- El cliente genera un pedido de los artículos que necesita para surtir la tienda.
2. Factura.- El proveedor realiza su factura correspondiente del pedido.
3. Entrega.- El proveedor realiza la entrega de la mercancía a la tienda o centro de distribución solicitada.

Un ejemplo de este flujo es el que sigue: Un cliente entra a una tienda, busca y solicita un juguete (orden de compra), el vendedor enseña el artículo al cliente y decide comprarlo, el vendedor le genera un comprobante de compra (factura), al final el juguete es entregado al cliente en sus manos (entrega).

Al realizar esto en grandes compañías con gran cantidad de tiendas, estamos hablando de enormes cantidades de mercancía. Estos tres pasos son básicos, pero detrás de cada uno, como se mencionó anteriormente, existen numerosos procesos para realizar cada tarea. Cada proveedor y cliente lo manejan según sea su proceso de entrega interno, cada uno de diferente manera. Cuando se generan órdenes de compra es importante administrar el pedido y revisar que se va a enviar o embarcar en el camión que va directo al cliente.

La administración de estos procesos son llevados a cabo mediante tecnologías que ayudan a realizar el flujo casi de forma automática en todo el ámbito. El cliente genera una orden de compra que es publicada mediante un portal web o enviada a través de un sistema AS2, correo electrónico o Web Service; el proveedor lo recibe, descarga e ingresa a su sistema interno para la

administración del mismo y posteriormente genera la factura (la cual es un requerimiento oficial). El ingreso de la información al sistema interno del proveedor en la mayoría de los casos es de forma manual, resultando probablemente en errores humanos no deseados, como introducir el total de la orden, cantidades y precios. Otros proveedores traducen el archivo enviado de la orden de compra a un formato que su sistema interno interpreta y guarda de forma automática anulando estos errores.

Con la finalidad de automatizar y prevenir estos errores, el cliente creó un portal web llamado “Portal de proveedores”, siendo éste un módulo donde cada proveedor pueda iniciar sesión y realizar lo siguiente:

- Revisar las órdenes de compra nuevas.
- Descargar las órdenes de compra en PDF [Anexo 10.1].
- Generar el ASN [8] (lo que se va a embarcar o enviar) de cada orden de compra.
- Imprimir en PDF el Packing List [Anexo 10.2] (es una lista de artículos a embarcar) y las Etiquetas [Anexo 10.3] que corresponde a las tarimas de cada orden de compra.

Inicio de intercambio electrónico por el cliente

El SAT (Servicio de Administración Tributaria) inició con la facturación electrónica en el 2004, y en el 2006 el cliente adquirió un sistema de IBM llamada IBM WPG mejor conocido como AS2, el cual es un software de intercambio electrónico de archivos entre cliente-proveedor, tales como facturas electrónicas entre otros. A partir de este momento el cliente le solicitó a sus proveedores que envíen sus facturas electrónicas por medio del AS2 y les obligó a adquirir o desarrollar un sistema de intercambio electrónico.

Con el paso de los años el cliente creció y empezó a abrir más tiendas a nivel nacional, por lo que su portal empezaba a saturarse con las órdenes de compra. El cliente decidió eliminar la responsabilidad en gran parte y dejarles a los proveedores que realicen el embarque internamente y usen su lógica para entregar un archivo XML por medio del sistema AS2, que no es más que el ASN de la orden de compra. En su momento se desarrolló un sistema de software llamado B2BOMv2.0 (Business to Business Object Manager versión 2.0); este se analizó entre el cliente y la empresa desarrolladora para generar el archivo XML del ASN de cada orden de compra y enviarla por el AS2.

La idea principal del cliente es migrar a los proveedores que tienen grandes cantidades de órdenes de compra diariamente en el portal de proveedores, al sistema de AS2 de intercambio electrónico y usen un sistema B2BOM para la administración o gestión de los ASN de cada orden de compra y así generar los documentos de embarque. El cliente solo envía el XML de cada orden de compra por medio del AS2, el B2BOM del proveedor lo recibe, traduce y obtiene información necesaria para guardarlo en su sistema interno ERP (sistema interno de administración empresarial). La lógica que se sigue se muestra en la figura 1. Esquema de función B2BOM.

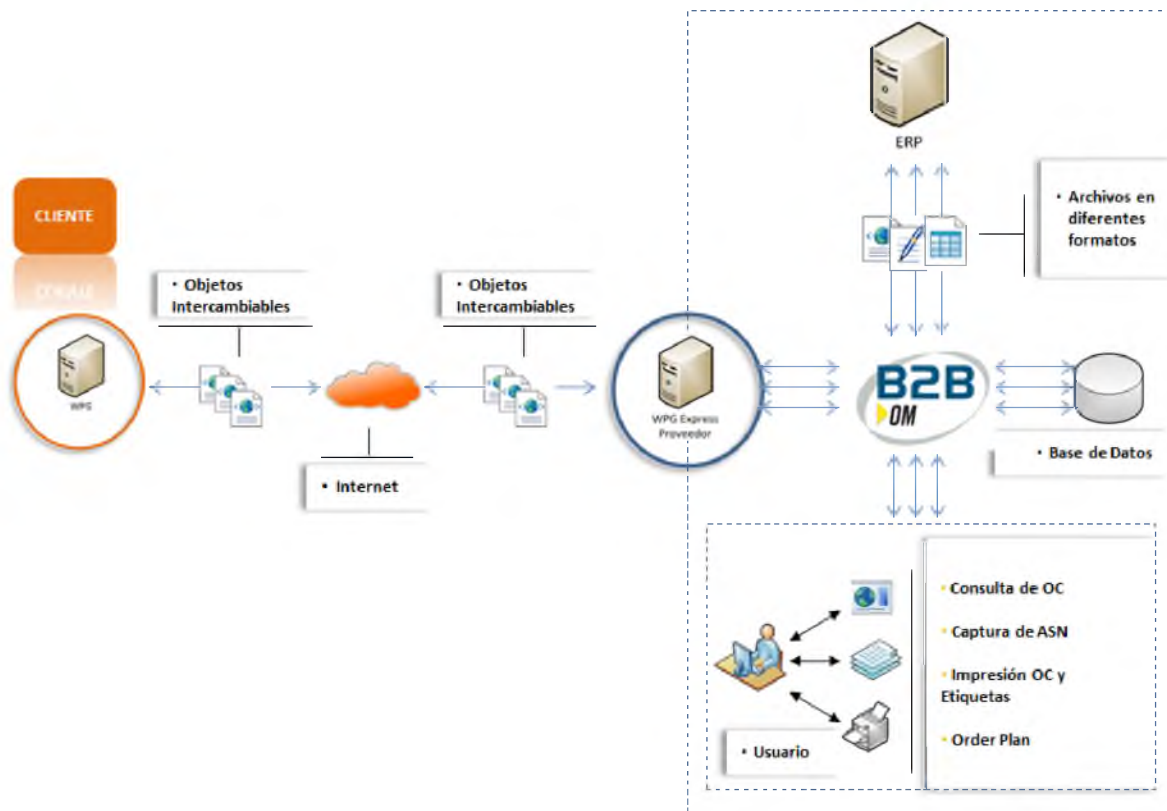


Figura 1. Esquema de función B2BOM.

Además de esto, desde el sistema de software B2BOM versión 2.0 (B2BOM v2.0) cada vez que llega una orden de compra se genera un archivo estructurado con información de la orden de compra y se deposita en una ruta que el proveedor define para que su sistema administrativo interno lo tome y se suba la información al ERP. Este archivo tiene el tipo y formato que el sistema interno puede traducir y guardar la información de la orden de compra.

1.2 PLANTEAMIENTO DEL PROBLEMA

Actualmente el cliente entrega órdenes de compra a sus proveedores por el canal AS2, el sistema B2BOM v2 recibe la orden de compra, la traduce, la guarda en la base de datos y genera un archivo en el formato que el proveedor lo requiere para subirlo al sistema interno, por ejemplo, SAP, Oracle, SAE, etc.

En el sistema B2BOM v2, el cliente genera un ASN por cada orden de compra para entregar la mercancía solicitada por su cliente. El proceso que hoy en día se realiza consiste en enviar una orden de compra y recibir un ASN para entregar la mercancía a un centro de distribución. En la figura 2. Proceso actual, se muestra como es el proceso actualmente siguiendo las flechas de arriba hacia abajo:

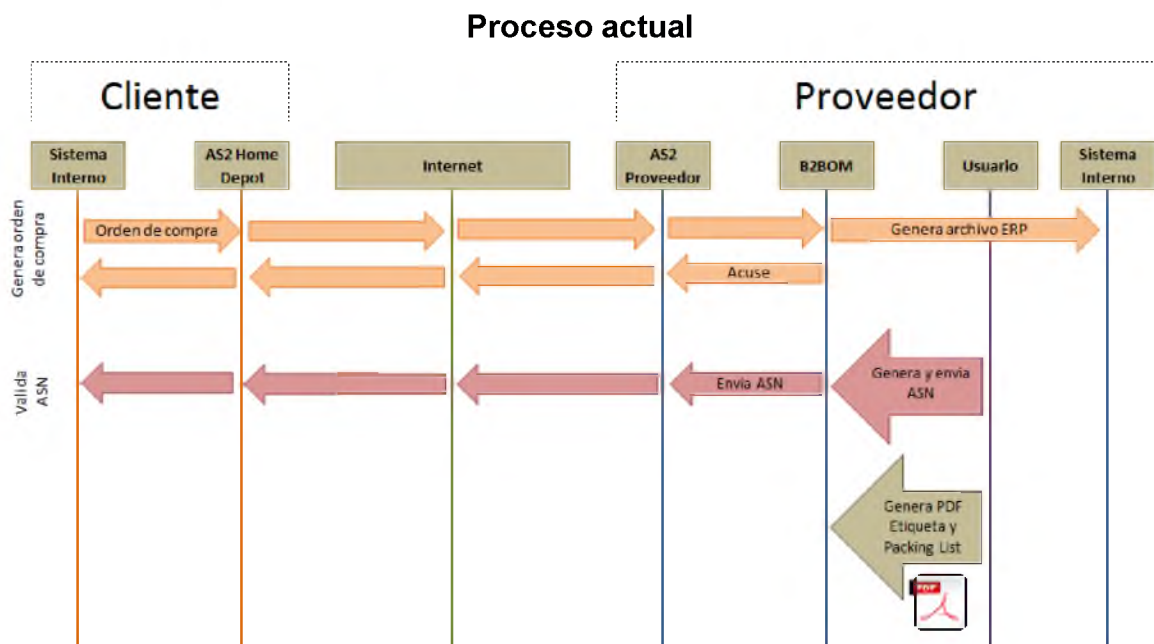


Figura 2. Proceso actual.

En resumen, el Cliente solo envía la orden de compra y el proveedor la recibe, lo guarda en el sistema interno ERP y desde el sistema B2BOM genera el ASN de la orden de compra y es enviado por el canal AS2 hacia el Cliente informando que se va a embarcar de la orden de compra. Con el nuevo esquema lo que requiere el Cliente es agrupar ASN de varias órdenes de compra para embarcar la mercancía, es decir, tener un control más estricto de lo que se envía en un tráiler o camión.

Nuevo esquema

El nuevo flujo para B2BOM v3 tiene dos procesos más, generar un Booking (conjunto de órdenes de compra) y realizar un Consolidado (conjunto de Bookings), anexo a la generación del ASN de versiones previas, quedando el proceso de entrega como sigue (ver figura 3):

PROCESO BOOKING.- Pre-notificación de embarque; es un conjunto de órdenes de compra pre-notificando la mercancía que va a ser enviada.

PROCESO ASN.- Cuando el booking sea ACEPTADO por el cliente a cada orden de compra se le genera un ASN (notificación de embarque) declarando lo que va a ser enviado al centro de distribución.

PROCESO CONSOLIDADO.- Cuando el booking y ASN son ACEPTADOS por el cliente se realiza el consolidado que es una agrupación de uno o más bookings que van a ser enviados al centro de distribución.

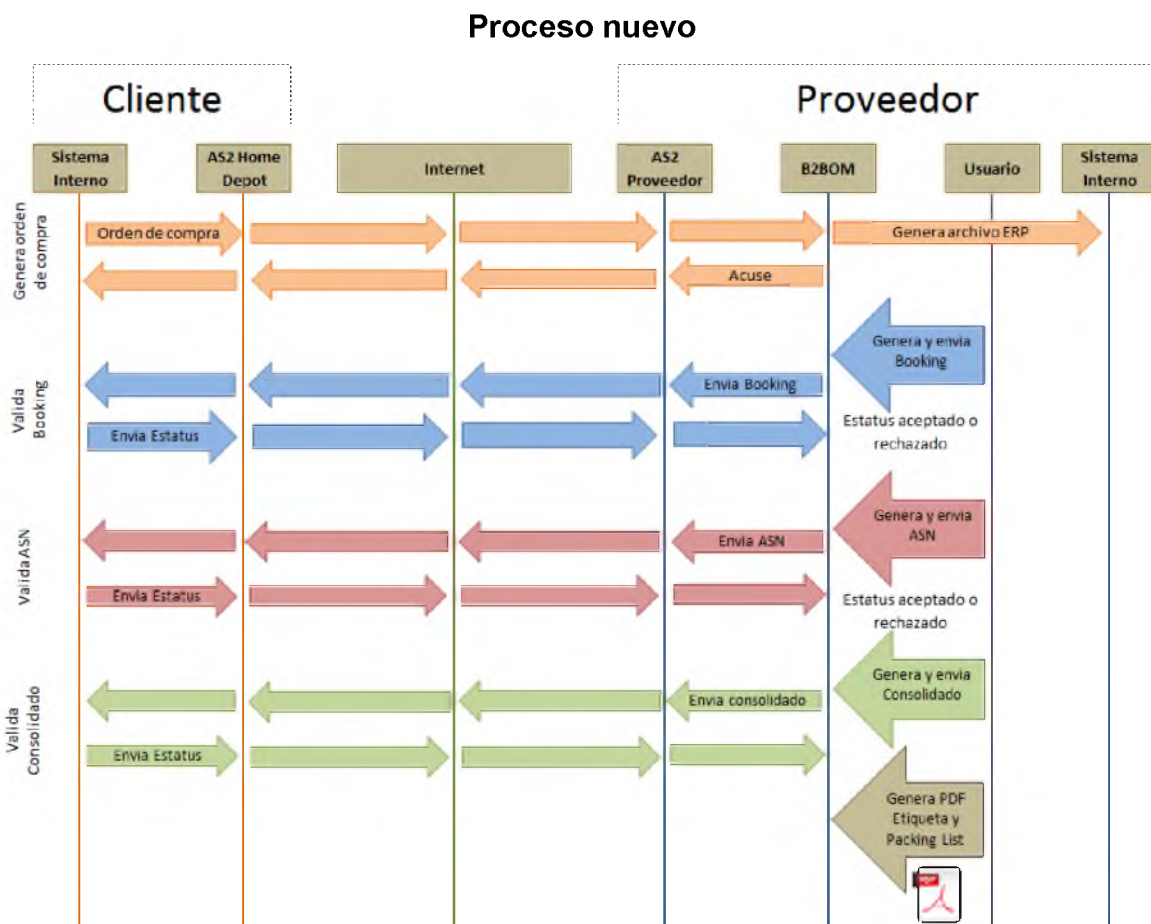


Figura 3. Proceso nuevo.

El cliente entregó a la parte desarrolladora de software la siguiente información.

1.2.1 Proceso Booking:

El Booking sirve a Home Depot como filtro para prevenir discrepancias, identificar si el proveedor pudiera cometer algún error embarcando más cantidades de las declaradas en la Orden de Compra o que embarque antes o después de la fecha establecida, conocer con anticipación las cantidades y artículos que estará embarcando, confirmar que esté bien el término de flete en base a lo negociado, si es prepagado o por cobrar, el tamaño de tráiler que estará recibiendo el Centro de Distribución de Mercaderías (MDC) con la mercancía para estimar la carga de trabajo diaria, el total de tarimas que va a contener el embarque, y la fecha en que estará entregando el proveedor para anticipar la cita de descarga.

Puntos importantes acerca del Booking:

- ✓ *Se puede hacer el Booking por una o varias órdenes que vayan a ir juntas en un embarque.*
- ✓ *En un Booking se pueden agregar Órdenes de Compra de diferente número de proveedor, siempre y cuando se encuentren en el mismo grupo de proveedores consolidados (esto último es administrado por Home Depot).*
- ✓ *La regla es tener un Booking por embarque (1 Booking = 1 camión = 1 embarque).*
- ✓ *Una vez que una Orden de Compra ya esté incluida en un Booking finalizado, ya no habrá forma de ponerla en otro Booking.*
- ✓ *Las Órdenes de Compra que se asignen a un Booking deberán tener el mismo Destino (Delivery To), es decir que todas se entreguen en el mismo MDC.*
- ✓ *Las Órdenes de Compra que se asignen a un Booking deberán tener el mismo Término de Pago (FreightTerm).*
- ✓ *El Booking de una Orden de Compra deberá ser creado 7 días antes ó 30 días después respecto a su fecha de entrega (Expected Ship Date). Por ejemplo si la Fecha de entrega en la PO es el 30 de Noviembre, y hoy es 24 de Noviembre, se puede hacer el Booking porque se estaría adelantando menos de 7 días. Para el caso de 30 días después, el último día para hacer el Booking sería el 29 de Diciembre.*
- ✓ *Una vez que el Booking es finalizado, sólo podrán embarcar el total de piezas declaradas en el Booking, aún y cuando la PO requiera más artículos o mayor cantidad de los artículos declarados. Por ejemplo si una orden de compra demanda embarcar 20 piezas y en el Booking sólo ponen*

15, ya no van a poder hacer el embarque por 20 piezas sino por 15 o menos, el Booking será la base para hacer el ASN.

1.2.2 Proceso ASN:

El ASN (Advance Shipping Notice) es útil para identificar los embarques que están en tránsito, el número de órdenes de compra que están por recibir los Centros de Distribución, asignar citas a los embarques, identificar que artículos y cantidades van en cada tarima en el embarque, conocer el peso y cubicaje de cada tarima. Este ASN nos ayuda como Cadena de Suministro a dar un mejor servicio a nuestras tiendas, y distribuir más rápido sus productos a tiendas y consumidores finales.

La forma de crear el ASN que representa el embarque se basa en dos tipos de archivos, un ASN Consolidado y uno o varios ASN individuales (por orden de compra).

Puntos importantes a tomar en cuenta en el ASN por Orden Compra:

- ✓ Previo a la realización del ASN se debe tener aceptado el Booking de la Orden de Compra como requisito.
- ✓ El ASN de una Orden de Compra deberá ser creado 7 días antes ó 35 días después respecto a su fecha de entrega (Expected Ship Date).
- ✓ La cantidad máxima que se puede declarar por artículo en el ASN es la cantidad máxima que venga en su Booking.
- ✓ No se pueden enviar artículos en el ASN que no enviaron en su respectivo Booking de esa Orden de Compra.
- ✓ La cantidad máxima que se puede declarar por artículo en el Booking es la cantidad máxima que venga en la Orden de Compra, si por alguna razón la cantidad que se ordenó es diferente al vendor pack y se necesita poner más piezas que las ordenadas, se deberá contactar al planner o merchant para solicitarle una corrección a la Orden de Compra o una nueva orden de compra si no se puede corregir.
- ✓ Cuando un artículo vaya a ser embarcado, todos los campos a nivel artículo deben ser llenados en el Booking.
- ✓ El Peso y cubicaje por artículo debe ser mayor a cero.
- ✓ Para los proveedores de importación el peso del Booking no debe exceder a 20,412 Kg.

1.2.3 Proceso Consolidado:

Puntos importantes a tomar en cuenta en el Consolidado:

- ✓ *La regla es tener un Booking por ASN Consolidado (1 Booking = 1 camión = 1 embarque = 1 ASN Consolidado), sin embargo el cliente está preparado para juntar uno o varios Bookings en un sólo ASN Consolidado siempre y cuando sea posible cargarlo en un sólo camión.*
- ✓ *En un Consolidado se pueden agregar Bookings de diferente número de proveedor, siempre y cuando se encuentren en el mismo grupo de proveedores consolidados (este último es administrado por Home Depot).*
- ✓ *Una vez que un Booking está incluido en un ASN Consolidado, no hay manera de separarlo, ni las PO's que contenga ese Booking del ASN Consolidado.*
- ✓ *Una Orden de Compra que este en un Booking y ese Booking este en un ASN Consolidado finalizado no podrá ser embarcada en otro ASN Consolidado, si por alguna razón no van a embarcar una PO que está documentada en un Booking, no se pondría la Orden de compra en el ASN Consolidado y deben solicitar al planner o merchant que les realicen una nueva orden.*
- ✓ *Una vez finalizado el ASN Consolidado, no se debe embarcar otro ASN Consolidado en el mismo embarque (tráiler).*
- ✓ *Cuando realicen un **Consolidado**, todos los ASN de las Ordenes de compran que van en un mismo Consolidado deben de tener el mismo: Destino (DeliveryTo), Termino de pago de flete (FreightTerm), Carrier Type (ServiceType), Nombre del Transportista (ForeignCarrier), Equip Type (ContainerSize), BL / pro number / Guía de Embarque (BLno), Load Type (Comments).*
- ✓ *Para los proveedores de importación el peso del Consolidado no debe exceder a 20,412 Kg.*
- ✓ *Previo a la realización del Consolidado se debe tener aceptado el ASN de la Orden de Compra como requisito.*

El sistema actual (B2BOM v2.0) cumple con el proceso que se tiene solo del ASN, pero realizando un análisis, el sistema no tiene la estructura para realizar cambios en el futuro o lo que está solicitando el cliente; la lógica del embarque se encuentra en las páginas Web, ahí mismo se realizan las validaciones y conexiones a la base de datos, no existe una estructura de paquetes o un control del mismo, en pocas palabras, no cumple con el Modelo Vista Controlador (MVC).

Por lo que el sistema actual no se tomara como referencia para este nuevo requerimiento del cliente y se realizara un sistema totalmente nuevo.

1.3 OBJETIVO GENERAL

Desarrollar un sistema Web Business to Business que permita a los Proveedores de mercancías del Cliente estructurar las órdenes de compra mediante un portal interno y privado.

1.3.1 OBJETIVOS ESPECÍFICOS.

- Analizar y comprender el nuevo proceso.
- Analizar la integración de las órdenes de compra con el sistema administrativo del proveedor.
- Estructurar los procesos.
- Diseñar una base de datos centralizada.
- Realizar las restricciones correspondientes en cada proceso.
- Analizar la información relevante a mostrar en pantalla al usuario final.
- Realizar los PDF correspondientes para su impresión.
- Analizar cada documento de intercambio electrónico recibido y creado para su envío.
- Parametrizar el sistema según convenga al proveedor.

1.4 HIPÓTESIS

Los proveedores podrán administrar sus embarques desde un sistema Web con la aprobación y validez del Cliente en cada proceso y entrega de la mercancía desde un portal interno llamado B2BOM v3.

1.5 JUSTIFICACIÓN

La nueva integración del proceso que el Cliente va a realizar con sus proveedores tendrá un impacto desde el punto de entrega de la mercancía; la empresa de desarrollo será la única que irá de la mano con este nuevo cambio analizando y comprendiendo estos cambios en un sistema de software que será calificado en cada avance con el corporativo central en México ubicado en Monterrey, Nuevo León.

Una oportunidad para realizar un sistema que cumpla con el modelo de programación basado en la ingeniería de software, con la ventaja de tener un

sistema único para todos los proveedores y no uno por cada proveedor como hoy en día se tiene.

Si el cliente se actualiza, sus proveedores deben de ir de la mano con su cliente, es por eso que al realizar el sistema que cumple con los requisitos estipulados, el beneficio es para ambas partes. Por otro lado, para los proveedores es una gran ayuda y motivación el tener este sistema integrado a su sistemas internos, ya que contarán con la administración de sus embarques desde un sistema propio, podrían trabajarlo en el horario que deseen, así como subir las órdenes de compra al portal interno de la empresa sin errores humanos, ya que el sistema B2B tendrá interacción con el sistema ERP.

Por último y por no ser el menos importante, es una oportunidad de crecer en la confianza depositada por el cliente hacia la empresa para futuros proyectos, enlazándose para enfoques que ayuden a integrar de forma automática los procesos que se tienen; adicional existirá un beneficio para ambas partes, por un lado el cliente podría recomendar el software B2BOM para sus proveedores con la seguridad de que no existirá un problema para embarcar la mercancía y la empresa extendería la cartera de clientes (proveedores) que harán tener dos puntos claves en cualquier negocio que se dedique al desarrollo de software: confianza y nuevos proyectos.

1.6 ALCANCES Y LIMITACIONES

Cada software que está en producción tiene alcances y limitaciones, es decir, lo que hace y lo que no puede hacer, siempre tratando de que un sistema cumpla con las funciones principales y realizando las actividades para las que fue creado; es por eso que el cliente entregó un documento de lo que debería hacer el sistema y sus respectivas validaciones, estos puntos fueron tratados en el planteamiento del problema. Para realizar el sistema es importante que se tenga un control sobre los procesos que se van a ejecutar y registrar que usuario realizó una actividad importante como enviar un documento, este punto es relevante a la hora de tratar un soporte.

Alcances:

1.- Sistema Web.

Aplicación Web para acceder desde cualquier parte de la empresa, en la misma red y en la nube si así lo desea el cliente.

2.- Acceso por usuario y contraseña.

Acceder al sistema por medio de un usuario y contraseña (encriptado), es importante para saber quién se conectó, cuáles fueron sus actividades en el sistema imprimiendo sus acciones en un archivo log.

3.- Multiusuario.

No hay límite de usuarios; es responsabilidad del proveedor gestionar los usuarios registrados según sea la conveniencia del proceso de embarque y capacidad de la infraestructura de los servicios computacionales ofrecidos a los usuarios, por lo que pueden existir múltiples usuarios conectados al sistema para realizar algún proceso para la entrega de mercancía.

4.- Acceder por medio de un Centro de Distribución.

El usuario se conectará al sistema por medio de un Centro de Distribución (MDC) para separar y filtrar los procesos.

5.- Bitácoras de registros de cada proceso.

Para cada acción importante y relevante se imprimirá la acción en un archivo log, y este contendrá información de las peticiones al sistema Web: quién la realizo, a qué hora, qué eliminó, quien envió el documento, quien imprimió.

6.- Validación del XML a través de un XSD.

Un archivo XSD es un lenguaje de esquema que contiene líneas relevantes para comprobar si un archivo XML tiene la estructura correcta, en estos casos los archivos que se intercambiarán tanto para emisión y recepción de documentos.

7.- Generación de PDF especificados por el cliente.

El cliente entregará un layout de cómo deben de ser contruidos los PDF (Packing List y Etiqueta) para que el sistema B2B los genere de la misma forma.

8.- Personalizaciones por cliente.

Cada cliente puede necesitar una personalización al sistema sin afectar el proceso principal de embarque, ejemplo, un botón especial que realice alguna acción, algún tipo de ordenamiento al buscar información a la base.

9.- Sistema parametrizable.

El sistema no contendrá código duro para las restricciones, será 100% parametrizable adecuándolo a las exigencias del cliente y para su mayor comodidad.

10.- Comunicación con el sistema interno.

El sistema tendrá la capacidad de adecuarse a cada sistema administrativo, ya sea creando un archivo de la forma que el cliente lo desee para que el ERP lo pueda leer desde una ruta del servidor y guardar la información en el sistema, con la extensión que lo interprete (TXT, XML, MOD, CSV), o bien conectarse a una base de datos y vaciar la información.

11.- Envío de correo electrónico.

La aplicación tendrá la capacidad de enviar un correo electrónico cuando las órdenes de compra lleguen al sistema y cuando se reciban estatus por cada proceso enviado al cliente, con la intención de que el receptor (usuario) este enterado de las respuestas de su cliente sin necesidad de entrar al sistema.

12.- Órdenes de compra en formato XML o EDI X12 [1,2]

Al recibir las órdenes de compra llegaran en formato XML y si el proveedor lo requiere podrían llegar en formato EDI X12.

EDIX12 (Electronic Data Interchange), intercambio electrónico de datos en formato X12 (Estados Unidos).

13.- Framework Struts 2.

Se utilizará el framework Strust 2.

14.- Servidor de aplicaciones Apache Tomcat.

Se usará el mismo servidor de aplicaciones ya que es la que actualmente tienen en la actual versión B2BOM.

Limitaciones:

1.- Inventario.

El sistema no modificará el inventario de cada artículo a embarcar.

2.- Los correos electrónicos.

El sistema intentara enviar una sola vez, si no es exitoso, se perderá la información.

3.- Personalizaciones al sistema.

Estas se pueden agregar siempre y cuando no afecten el flujo principal del proceso y a conveniencia del proveedor.

4.- Sistema separado del AS2.

El sistema de software como el AS2 de intercambio electrónico de información, son totalmente independientes, es decir, uno no depende del otro para su funcionamiento, pero si son indispensables para el flujo de información hacia su cliente. Cada uno tiene su tarea correspondiente.

5.- Uso exclusivo.

El sistema es para realizar tareas de embarque exclusivo para el cliente, no administra otro tipo de archivos ni gestiona otras tareas que no sean las establecidas.

6.- Hilo.

El sistema contiene un hilo importante que en cada cierto tiempo se ejecuta, este funciona para monitorear una carpeta donde llegan los archivos XML de la orden de compra, solo se puede detener si el servidor de aplicaciones es detenido. Lo único configurable es el tiempo de ejecución.

CAPÍTULO 2. MARCO TEÓRICO

Teniendo en cuenta que el sistema actual no tiene la estructura o diseño para una implementación a futuro, es decir, con los nuevos requerimientos que el cliente solicita no es viable reutilizar el software actual, por lo que se optó por realizar uno nuevo que cumpla con la ingeniería de software, utilizando una metodología de software, diseño de bases de datos, diseño de paquetes, etcétera.

2.1 METODOLOGÍA DE SOFTWARE

Las aplicaciones de software regularmente son desarrollados mediante una metodología que ayudan a planificar, estructurar y gestionar el desarrollo de una aplicación con el fin de que la codificación sea simple y fácil de implementar; el trabajo más duro está en el análisis y diseño de software, es por ello que se debe utilizar una metodología dependiendo para dónde va el desarrollo del sistema y si es un software para un cliente en específico o para un mercado en general.

Existen varias metodologías de software que ayudan a gestionar un desarrollo:

Modelos tradicionales.

- Modelo en cascada.
- Modelo estructurado.
- Modelo basado en prototipos.

Modelos evolutivos.

- Modelo de desarrollo evolutivo (espiral).
- Modelo incremental.

Modelos ágiles.

- Proceso unificado (UML).
- Programación extrema (XP).
- Scrum.

Analizando cada metodología de software y teniendo en cuenta el tiempo de implementación el cual el margen es poco, se decidió utilizar el Modelo en Cascada, ya que para el cliente no es importante el cómo se realice el sistema siempre y cuando cumpla con las restricciones antes mencionadas que sirven como análisis del sistema B2BOM. Utilizar el método en cascada tiene una ventaja importante, ya se tienen los requerimientos desde un principio y no son cambiantes; al cliente no le interesa el diseño de interfaz y/o código de la

aplicación, siempre que cumpla con las restricciones antes mencionadas y en cada fase se puede realizar una entrega de avances al cliente. A continuación se explica a detalle el modelo en cascada.

Modelo en cascada.

En la figura 4, modelo en cascada, se puede observar el flujo del modelo en sus fases correspondientes que se va a implementar en este proyecto.

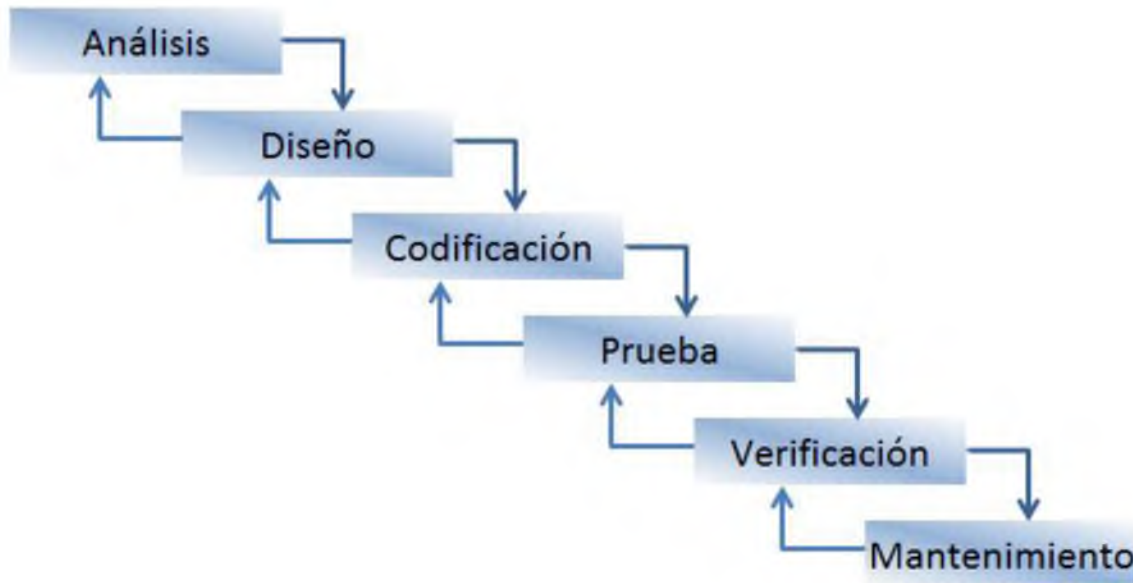


Figura 4. Modelo en cascada.

- 1) **Análisis de los requisitos del software.** El proceso de recopilación de información se intensifica en el software, se debe comprender el ámbito de la información, la lógica a seguir así como la función, el rendimiento y las interfaces requeridas.
- 2) **Diseño.** Se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que se comience con la codificación.
- 3) **Codificación.** El diseño debe traducirse en una forma legible para la máquina, el paso de codificación realiza esta tarea; si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.
- 4) **Prueba.** Una vez que se ha generado el código comienza la prueba del programa; se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que se requieren.

- 5) **Verificación.** Es la fase donde el usuario final prueba el sistema, para esto los desarrolladores debieron asegurarse que el sistema no tenga fallos.
- 6) **Mantenimiento.** El software sufrirá cambios después de que se entrega al cliente; los cambios ocurrirán debido a errores, a que el software deba adaptarse a cambios del entorno externo o debido a que el cliente requiera ampliaciones funcionales o rendimiento.

Ventajas:

- ✓ Se tiene todo organizado y no se mezclan las fases.
- ✓ Ayuda a localizar errores desde las primeras etapas del proyecto y a un bajo costo.
- ✓ Ayudan a minimizar los gastos de la planificación porque permite realizarla sin problemas.

Desventajas:

- ✓ Gran dependencia de los requerimientos iniciales.
- ✓ Difícilmente un cliente va a establecer al principio los requerimientos necesarios, lo que ocasiona un atraso en el proyecto, este modelo es muy restrictivo y no permite movilizarse entre fases.
- ✓ El modelo genera pocos signos de progreso hasta el final; esto puede dar la impresión de un desarrollo lento y el cliente puede dudar si su sistema puede ser entregado a tiempo.

2.2 SISTEMA WEB

Existen sistemas de software que son utilizados por medio de aplicación Web y aplicaciones de escritorio, este último requiere ser instalado en cada computadora en la que se va a usar, a diferencia del primero, que solo es instalado en un servidor o computadora que este en red en toda la compañía, y por medio de un navegador Web se puede acceder y realizar las tareas correspondientes. El funcionamiento de páginas dinámicas está basado en el almacenamiento de la información en la base de datos y en la facilidad de modificación de gestores de contenido. Las más comunes son, JSP, PHP y ASP. [6]. En el caso del sistema Web B2BOM estará en JSP.

- **JSP (Java Server Pages) [6].-** Es una tecnología desarrollada por Sun Microsystems. Permite incluir código Java en páginas HTML el cual es sustituido por el resultado de su ejecución y enviarla al cliente. Por lo general estos programas hacen consultas a bases de datos y dependiendo

del resultado que se despliegue será la información que se muestre a cada usuario según su petición. Estos archivos llevan extensión “.jsp”.

Un sistema tipo Web contiene páginas sin determinar, parcialmente o en su totalidad. El contenido final es generado cuando el usuario solicita la página al servidor Web. Las ventajas de este tipo de sitios son:

- Compatibilidad multiplataforma. Una de la ventaja de la aplicación web es que es multiplataforma, por lo que soporta cualquiera de los sistemas operativos.
- Actualización. Siempre están actualizadas con la última versión, ya que no es necesario cambiarla en cada computadora, solo se actualiza en el servidor donde está instalado.
- Acceso inmediato. Permite acceder de forma rápida y sin necesidad de esperar a que cargue el software.
- Misma información. Se guarda en una sola base de datos o archivos, por lo que la información guardada y modificada está centrada en un solo lado.
- Multiusuario. Varios usuarios pueden acceder al sistema de software y trabajar de forma simultánea sin afectar el trabajo de otros usuarios.
- Un sistema único en el server y no instalable en cada máquina que vaya a ser usada.

2.3 BASE DE DATOS

La información es la clave de cualquier intercambio de datos, cómo las personas se comunican y guardan en su memoria relatos, hechos, acciones y hasta el leer un libro, así funciona para las empresas, negocios, departamentos; el tener guardada la información empresarial como los empleados, sus sueldos, las facturas, las órdenes de compra, los artículos, los embarques, etcétera, es de vital importancia para poder generar una minería de datos en donde pueden buscar información del pasado y realizar estadísticas que ayuden a comparar el crecimiento del negocio. En empresas pequeñas, esta información en ocasiones se mantiene en archivos de tipo hojas de cálculo, trayendo consigo desventajas en su deficiente flexibilidad y acceso compartido por los usuarios. Otras empresas optimizan este almacenamiento usando una base de datos. Una base de datos es una estructura de tablas relacionadas entre sí para guardar información que es muy importante para una empresa o negocio; es una entidad en la cual se pueden almacenar datos de manera estructurada.

Para realizar esta lógica se usó una de las grandes ventajas que da el desarrollo de Java, el de usar clases abstractas, esto nos da una ventaja

importante ya que por medio de un parámetro se decide si el software utilizará MySQL, SQL Server, Oracle, DB2. Todo va a depender del requerimiento del proveedor. Desde un inicio se propone MySQL, por ser gratuito y no requiere de una licencia.

Gestor de base de datos

Para gestionar correctamente la información de una base de datos, acceso concurrente, seguridad y respaldo, entre otras, se usa un gestor de base de datos. La mayoría de los proveedores cuentan con gestores de base de datos, pero las empresas medianas regularmente no cuentan con un gestor en el que deben de invertir y pagar una licencia, es por eso que se llega a la conclusión de que el sistema B2B sea multibase de datos adaptándose a las necesidades de los proveedores. Existen gestores de base de datos en el mercado de forma gratuita o con licencia oficial, por ejemplo y los más conocidos son MySQL, Microsoft SQL Server, Oracle, PostgreSQL, Microsoft Access y DB2.

MySQL es la base de datos número uno para las aplicaciones basadas en la Web, utilizada por redes sociales y todas las propiedades Web más importantes y los inicios exitosos de manera virtual [5].

En el sistema B2BOM se utilizará el gestor de base de datos MySQL por tener antecedentes de uso con los proveedores.

2.4 ARQUITECTURA J2EE

La arquitectura J2EE (Java 2 Enterprise Edition) es un periodo de ejecución para gestionar los componentes de la aplicación desarrollados según las especificaciones del API (Servlet [6], JSP y JSTL) y destinados a proporcionar acceso a los API de J2EE. [6]. Cualquier desarrollo con la tecnología J2EE puede ser desplegada en cualquier servidor de aplicaciones o servidor Web que cumpla con el estándar. [6]

La siguiente figura 5. Arquitectura J2EE, se muestra los contenedores y sus API:

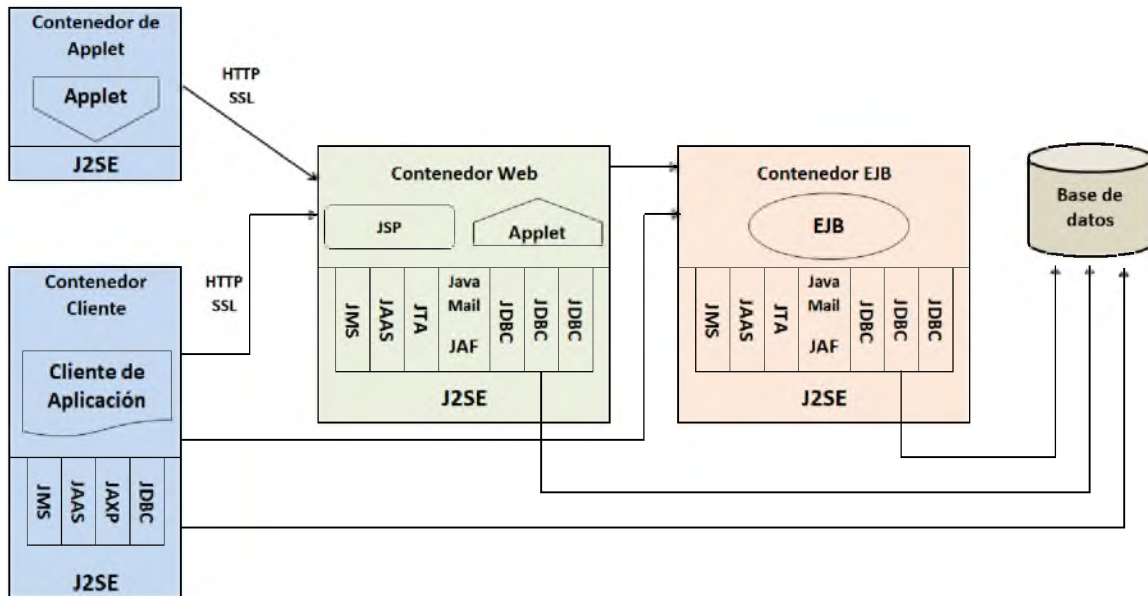


Figura 5. Arquitectura J2EE. [6]

Esta arquitectura muestra cuatro contenedores:

- Un contenedor Web para albergar servlets Java y páginas JSP.
- Un contenedor EJB para albergar componentes Enterprise JavaBean.
- Un contenedor applet para albergar applets Java.
- Un contenedor de aplicación cliente para albergar aplicaciones Java estándar.

Cada uno de estos componentes proporciona un entorno de periodo de ejecución para los respectivos componentes. En la figura anterior, los bloques verticales situados en la parte inferior de cada contenedor representan los API de J2EE. Además del acceso a estos API a nivel infraestructura, cada contenedor también implementa el correspondiente API específico a cada contenedor (API Java Servlet para contenedor Web y el API EJB para el contenedor EJB). [6]

Los rectángulos (Servlets, paginas JSP y los EJB) son los programas que se desarrollan y se albergan en estos contenedores mejor conocidos como componentes de aplicaciones. [6]

En esta arquitectura, existen dos tipos de clientes [6]:

- **Cientes Web**, que se ejecutan normalmente en navegadores Web. Para estos clientes, la interfaz de usuario es generada del lado del servidor como HTML o XML y es convertida por lo navegadores. Utilizan HTTP para

comunicarse con los contenedores Web que incluyen JSP y Servlet de Java.

- **Cliente EJB** son aplicaciones que acceden a componentes EJB en contenedores EJB.

Para nuestro sistema B2BOM utilizaremos contenedores Web con JSP para que el cliente acceda desde el navegador y pueda utilizar la aplicación para desarrollar o gestionar el embarque de la mercancía hacia el cliente.

2.5 SERVIDOR DE APLICACIONES

Un servidor de aplicaciones es un tipo de servidor que permite el procesamiento de datos de una aplicación cliente, es decir, un sistema Web, y sobre el cual están montados para que puedan funcionar de forma apropiada. Un servidor de aplicación ofrece tiempo de ejecución que admite la implementación y la administración eficaces de aplicaciones de negocio de alto rendimiento basadas en servidor. Estas aplicaciones son capaces de servir solicitudes desde sistemas cliente remotos, incluidos exploradores Web que se conectan desde Internet o desde una intranet o una red corporativa, y desde sistemas de equipos remotos que pueden enviar solicitudes como mensajes. Algunos de los servidores de aplicaciones más conocidos son Apache Tomcat [10], Jetty e IBM Websphere Application Server (mejor conocido como WAS).

2.5.1 APACHE TOMCAT

Es una implementación de código abierto de Java Servlets y Java Server Pages (JSP), es decir, un contenedor que soporta Servlets [7]; Los servlets que se ejecutan en Apache tomcat son programas pequeños de lado del servidor independientes de la plataforma que amplían programáticamente la funcionalidad del servidor Web. En API Java Servlet proporciona un marco sencillo para construir aplicaciones en estos servidores Web [6].

En la figura 6. Apache Tomcat, se puede observar cómo funciona el Apache Tomcat.

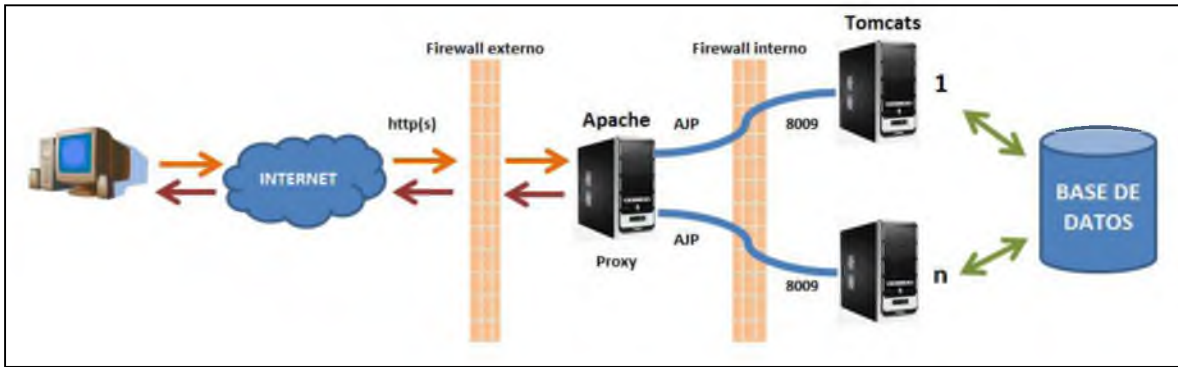


Figura 6. Apache Tomcat

En la figura anterior se observa lo siguiente:

1. Peticiones a través de Internet o intranet.
2. Apache es un servidor de aplicaciones.
3. Tomcat es un contenedor de Servlets.
4. Base de datos es donde está localizada la información que utilizan las aplicaciones.

Explicando a detalle, a través de una petición del usuario desde el sistema Web utilizando un protocolo HTTP o HTTPS, llega al Apache (Servidor) que a su vez envía peticiones por AJP (Apache Jserv Protocol de forma binaria a la aplicación Web) al servidor de aplicación Tomcat (donde está alojado el sistema B2BOM).

En resumen, el Apache Tomcat es un servidor donde estará el sistema B2BOM y es accedido desde cualquier red de la empresa o si se desea desde cualquier lugar con acceso a internet si así lo desea el proveedor.

2.6 RECURSOS Y TIEMPO DE LA EMPRESA DESARROLLADORA

Una vez obtenido y definidos los puntos anteriores como el servidor de aplicaciones, metodología de software, gestor de base de datos y el sistema Web, es importante enfatizar el responsable del proyecto desde un inicio; también cuantos desarrolladores estarán detrás del mismo y quien realizará los entregables en cada fase de implementación; también el tiempo y alcances.

En los desarrollos de la empresa están tres áreas involucradas en cada proyecto, ventas, líder de proyecto y desarrolladores; para este desarrollo de B2BOM v3 la empresa cuenta con personal limitado, por lo que solo un desarrollador toma el puesto de líder de proyecto y desarrollo en general; en

resumen, solo una persona será encargada de este proyecto atendiendo cada entregable, juntas y avances del mismo, así como también las pruebas internas, con el cliente piloto, salida a producción y documentación del sistema.

El tiempo de implementación del sistema B2BOM v3 son de tres meses y una semana completos con días hábiles, que van desde el análisis de los requerimientos hasta la salida a producción; el análisis, diseño del sistema y desarrollo del sistema son tiempos definidos por el líder del proyecto, las pruebas y salida a producción propuestas por el cliente. Las pruebas cliente y pruebas piloto va a depender del tiempo y disponibilidad del usuario final o personal correspondiente. En la figura 7. Plan de trabajo se puede observar los seis puntos que se van a realizar durante el proyecto.



Figura 7. Plan de trabajo.

1. **Análisis** del proyecto donde se identifica el problema, los recursos a utilizar y como se va a resolver.
2. Realizar el **diseño del sistema** con los componentes a utilizar como los diagramas de casos de uso, diagrama de clases y diseño de la base de datos.
3. **Desarrollo del sistema** empezando o atacando el núcleo del problema (Orden de compra, Booking, ASN y Consolidado).
4. Las **pruebas** que realiza el **cliente** junto con el equipo de desarrollo para validar cada proceso y que cumpla con las restricciones o reglas de negocio antes mencionadas.
5. Las **pruebas** que realiza el **proveedor** piloto donde valida cada proceso simulando el proceso final y realizando cada acción junto con el equipo de desarrollo.
6. Se establece una fecha de **salida a producción** del producto entre la empresa desarrolladora, el cliente y el proveedor.
7. **Monitorear** el proceso del proveedor en producción y realizando una bitácora de lo que realiza el sistema, sus virtudes y sus errores.

El plan de actividades es una lista que se entrega al cliente con las actividades que se van a realizar durante el proyecto con fechas estimadas de entrega. En la figura 8. Plan de actividades se muestra el inicio del proyecto hasta la fase de salida a producción.

PLAN DE ACTIVIDADES DEL 2 DE MAYO AL 12 DE AGOSTO DE 2011					
ACTIVIDAD	TEMA	FECHA INICIO	FECHA ENTREGA	DIAS	RESPONSABLE
1	Analisis del proyecto y problema	02/05/2011	06/05/2011	5	Desarrollador
2	Diseño del sistema				
3	Logica del proceso	09/05/2011	09/05/2011	1	Desarrollador
4	Diseño de casos de uso	10/05/2011	12/05/2011	3	Desarrollador
5	Diseño de base de datos	13/05/2011	18/05/2011	4	Desarrollador
6	Diagrama de clases	19/05/2011	20/05/2011	2	Desarrollador
7	Desarrollo del sistema				
8	Modulo ordencompra	23/05/2011	26/05/2011	4	Desarrollador
9	Modulo Booking	27/05/2011	03/06/2011	6	Desarrollador
10	Modulo ASN	06/06/2011	10/06/2011	5	Desarrollador
11	Modulo Consolidado	13/06/2011	17/06/2011	5	Desarrollador
12	Modulo catalogo	20/06/2011	24/06/2011	5	Desarrollador
13	Modulo administrativo	27/06/2011	01/07/2011	5	Desarrollador
14	Prueba cliente				
15	Prueba ordencompra	04/07/2011	04/07/2011	1	Desarrollador y cliente
16	Prueba Booking	05/07/2011	07/07/2011	3	Desarrollador y cliente
17	Prueba ASN	08/07/2011	12/07/2011	3	Desarrollador y cliente
18	Prueba Consolidado	13/07/2011	15/07/2011	3	Desarrollador y cliente
19	Prueba piloto				
20	Instalacion y configuracion del sistema	18/07/2011	18/07/2011	1	Desarrollador
21	Prueba ordencompra	19/07/2011	19/07/2011	1	Desarrollador y proveedor
22	Prueba Booking	20/07/2011	22/07/2011	3	Desarrollador y proveedor
23	Prueba ASN	25/07/2011	27/07/2011	3	Desarrollador y proveedor
24	Prueba Consolidado	28/07/2011	29/07/2011	2	Desarrollador y proveedor
25	Salida a producción				
26	Instalación y configuración del sistema	01/08/2011	02/08/2011	2	Desarrollador
27	Capacitación y entrega de manual de usuario	03/08/2011	04/08/2011	2	Desarrollador
28	En producción	05/08/2011	05/08/2011	1	Desarrollador, cliente y proveedor
29	Monitorear				
30	Atento a cualquier inconveniente del proveedor	08/08/2011	12/08/2011	5	Desarrollador y proveedor

75 Días hábiles del proyecto

Figura 8. Plan de actividades.

El plan de actividades incluye el número de actividad para identificar en donde está localizado el proyecto y sus avances, el tema de la actividad, las fechas de inicio y entrega de cada actividad, los días hábiles que toma realizar la actividad y el responsable de la actividad. Con esta tabla se lleva un control para las partes o personas involucradas para cada acción o actividad y prepararse cuando a cada responsable le toque el turno correspondiente.

2.7 OTROS SISTEMAS B2B.

En el mercado de software existen sistemas que realizan un mismo proceso pero que son desarrollados por diferentes compañías o distribuidores de software, como por ejemplo, SAE, SAP y ORACLE que son sistemas internos para

administrar una compañía, otro ejemplo es la paquetería de Office como Microsoft Office u Open Office, también sistemas de AS2 para el intercambio electrónico de información como IBM Wepshere o Medelson (software libre).

El sistema B2BOM como se mencionó anteriormente es exclusivo para proveedores del cliente y no funciona para otras cadenas comerciales o flujos de embarque de mercancía, es por eso que el sistema es lanzado como campaña por el cliente como el único software validado y liberado para uso comercial entre proveedores para embarcar la mercancía, aunque existen en el mercado dos sistemas B2B conocidos de uso exclusivo para proveedores del cliente aunque no está validado por este último y se desconoce qué tipo de sistemas son (Web, sistema de escritorio u otro tipo de sistema). Por último, el sistema B2BOM no es software libre por lo que tiene una licencia oficial anual y esto se establece desde la compra del producto.

CAPÍTULO 3. ANÁLISIS

Una vez que se obtuvieron los requerimientos, se realizó la etapa de análisis en la cual cada evento o proceso que contendrá la aplicación será inspeccionado conforme al flujo y reglas de negocio establecidos por el cliente. Para el sistema B2BOM se estudia el proceso de embarque, los diseños de los archivos XML y PDF, así como la integración de las órdenes de compra al sistema interno y personalizaciones del proveedor y los módulos que contendrá el sistema de software.

3.1 PROCESO DE EMBARQUE

Cada artículo que se compra se debe reemplazar lo antes posible para que el departamento o tienda cuente con la garantía de que tiene más artículos. Desde esa simple acción viene una ola de procesos para poder surtir una tienda y tener a diario mercancía lista para los consumidores.

Los negocios tienen proveedores que le surten mercancía desde una simple orden de compra. El proveedor realiza internamente un proceso para verificar el inventario y gestionar la orden de compra. En sus sistemas internos se visualiza si la mercancía solicitada puede ser embarcada y en qué tiempo. Usando su propia lógica para embarcar su mercancía (independiente de cada proveedor) realiza las tarimas y las coloca en el medio de transporte.

En un primer paso solo se revisan las órdenes de compra que van a ir en un camión y juntarlas en un Booking, aquí se define las cantidades que se van a embarcar y definir el tipo de empaque (bultos, rollos, caja o cartón) por cada artículo de cada orden de compra.

En una segunda fase se analiza cada embarque; esta fase es importante realizarla de forma genérica para que cada cliente pueda embarcar la mercancía sin problemas almacenándola en tarimas, las cuales son definidas por el usuario desde el sistema de software colocando cada artículo con su respectiva cantidad en cada tarima.

En una tercera fase se compone sólo de consolidar el camión, es decir, agrupar el Booking que van a ir en el camión y sean entregadas en el Centro de Distribución de Mercaderías.

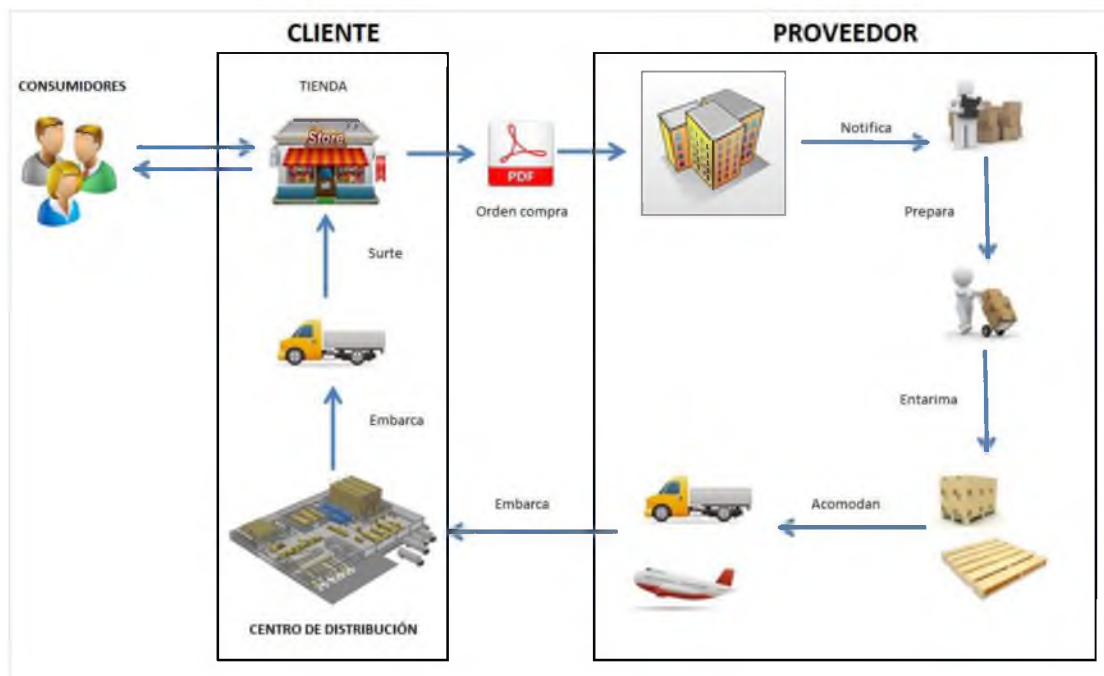


Figura 9. Proceso embarque.

En la figura 9. Proceso embarque, se puede observar a gran escala como es el proceso para la compra y venta de mercancía entre cliente-proveedor. El **consumidor** entra a la **tienda** y compra artículos para sus necesidades, esto afecta el inventario de la tienda, la cual debe solicitar mercancía para abastecer su negocio. Lo hace mediante una **orden_de_compra** que llega al proveedor; este **notifica** al área de bodega para revisar si hay mercancía, la bodega **prepara** los artículos y los acomoda en **tarimas** específicas, después **acomodan** esta tarimas en el tráiler y **embarca** la mercancía al centro de distribución del cliente; el cliente la recibe, administra y la **surte** a la tienda.

3.2 ÓRDENES DE COMPRA E INTEGRACION CON EL SISTEMA INTERNO

La administración de una empresa es clave para su funcionamiento, el gestionar cada departamento que la compone es necesario para administrar cada punto estratégico dentro del negocio, por ejemplo, departamento contable, departamento de sistemas, departamento de embarque, departamento administrativo, recursos humanos, empleados, sueldos y salarios, facturas, etc. Para gestionar todos los departamentos que la componen, existen sistemas en el mercado que realizan estas tareas (por ejemplo ORACLE, SAE y SAP).

Estos sistemas se realizaron con la idea de administrar casi el 100% de la empresa, pero no siempre se adapta a todas las necesidades, y realizan tareas de forma general, por lo que el usuario tiene que realizar actividades de forma

manual, en ocasiones se cometen errores no deseados; esto genera que puedan llegar a realizar doble trabajo, tener un problema interno en su departamento y llegar a provocar una inestabilidad a la hora de facturar o realizar cuentas contables.

Un ejemplo común es la captura manual de órdenes de compra, en las cuales, por algún error de dedo en la captura del precio unitario, precio total, IVA, número de proveedor, etcétera. Por lo que la mayoría de los sistemas ERP tienen la capacidad de leer un archivo que interprete la información necesaria para subirla al sistema de forma automática sin necesidad de que el usuario lo realice.

El B2B tiene la capacidad de generar un archivo en el formato que el sistema ERP del proveedor lo acepte, traduzca e inserte internamente. Esto es una ventaja importante para el proveedor y que ayuda a no cometer errores humanos.

¿Cuál es su función?

Al llegar las órdenes de compra el B2B lo valida, guarda y notifica al proveedor que llegaron ciertas órdenes de compra y adicional genera un archivo en el formato que su sistema interno lo pueda interpretar, traducir y guardar; el archivo se deposita en una ruta específica del servidor, ejemplo, un archivo en formato TXT, CSV, XML y MOD.

El sistema contempla otros tipos de procesos para integrar la orden de compra al sistema interno, ejemplo, procedimientos almacenados y Web services. Para realizar estas tareas es importante sincronizar las pruebas con los proveedores.

3.3 DOCUMENTOS DE INTERCAMBIO ELECTRÓNICO

Son documentos estructurados con datos necesarios para intercambiar información; para que el proveedor interactúe con el cliente sobre los embarques es necesario una comunicación, aquí es donde entra el sistema AS2, este se encarga de interactuar entre el servidor del cliente y el del proveedor, es decir, un intercambio electrónico de información en sus tres fases (Booking, ASN y Consolidado) así como la recepción de orden de compra en formato XML o EDI X12.

Orden de compra y Acuse

La orden de compra es enviada por el cliente a través del AS2 en formato XML [Anexo 10.4] o EDI X12 y el sistema B2B valida la estructura del archivo a

través de un XSD [Anexo 10.5] sin importar la información que contenga, si es correcta se inserta en la base de datos y realiza las tareas correspondientes, si no lo es se envía a una carpeta de error.

Cuando llegan las órdenes de compra al sistema es necesario notificar al cliente que si se recibió la orden de compra no importando si ya existe en el sistema B2B o esta errónea. El sistema debe generar un XML [Anexo 10.6] de respuesta por cada orden de compra.

En la figura 10. Proceso de la orden de compra, se muestra el flujo o proceso de la orden de compra y acuse de recibo.

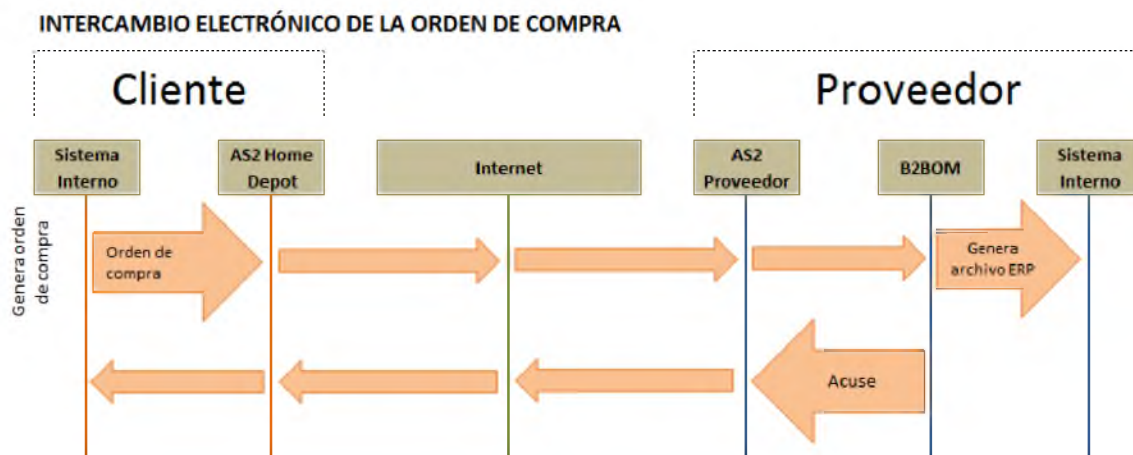


Figura 10. Proceso de la orden de compra.

Booking y Estatus Booking

El primer paso es la de seleccionar que órdenes de compra se van a enviar en el camión y colocar la información solicitada por el cliente en ese embarque como el tipo de carga, tipo de caja del tráiler o camioneta, el tipo de transporte, el transportista, número de embarque y las fechas estimadas de entrega. Al finalizar esta información es enviada por el usuario a través del sistema, aquí se genera un XML [Anexo 10.7] con información necesaria, se deposita en una ruta por default y se valida su estructura a través de un XSD [Anexo 10.8], si es inválido notifica al usuario, si es válido se mueve a una ruta específica en donde entra en acción el AS2, el cual lo toma, firma, encripta la información y la envía al cliente.

En respuesta al Booking enviado el cliente envía un XML [Anexo 10.9] que contiene si el documento fue aceptado o rechazado, este es validado

estructuralmente por un XSD [Anexo 10.10], si es correcto se traduce y actualiza el Booking, si es incorrecto lo mueve a una carpeta de error.

En la figura 11. Proceso Booking, se muestra el flujo o proceso del Booking y estatus del Booking.

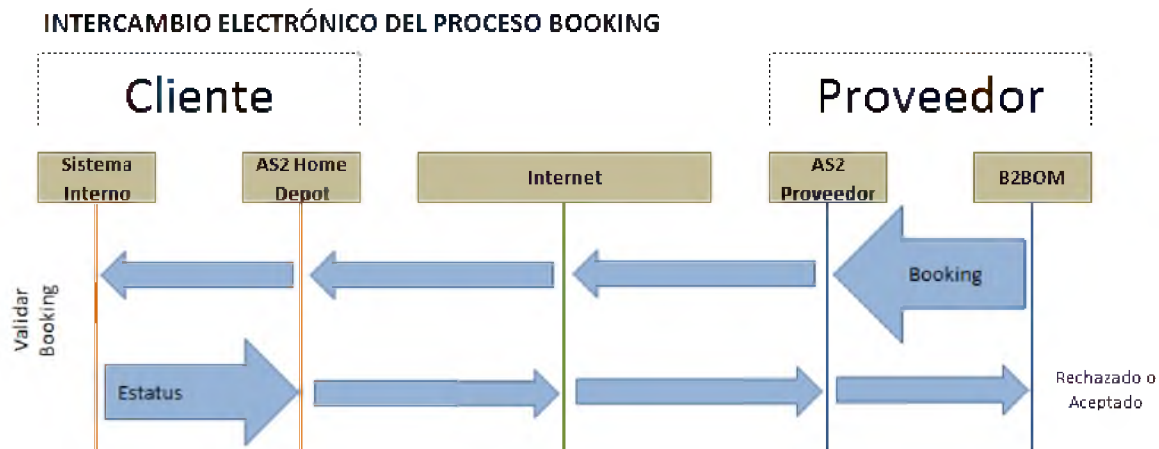


Figura 11. Proceso del Booking.

ASN y Estatus ASN

En esta fase el Booking ya debió haberse aceptado, por lo que se tiene que realizar cada ASN de cada orden de compra; aquí se divide la orden de compra y se define que se va a enviar en cada tarima especificando cada artículo y sus bultos. Una vez terminado todos los ASN de las órdenes de compra involucradas estas son enviados al cliente; así como en el mismo paso anterior se genera un XML [Anexo 10.11] con la información necesaria por cada ASN, se valida la estructura a través de un XSD [Anexo 10.12], si es correcta la estructura se copia a la ruta del AS2 para ser enviado. Si todos los ASN son aceptados se pasa a la siguiente fase de Consolidado, si uno o varios de los ASN son rechazados, se revisa el estado de rechazo, se modifica y se trata de enviar de nuevo.

En respuesta a los ASN enviados el cliente envía un XML [Anexo 10.13] por cada documento enviado, este contiene si el documento ASN fue aceptado o rechazado, este es validado estructuralmente por un XSD [Anexo 10.14], si es correcto se traduce y actualiza el ASN, si es incorrecto lo mueve a una carpeta de error.

En la figura 12. Proceso ASN, se muestra el flujo o proceso del ASN y estatus ASN.

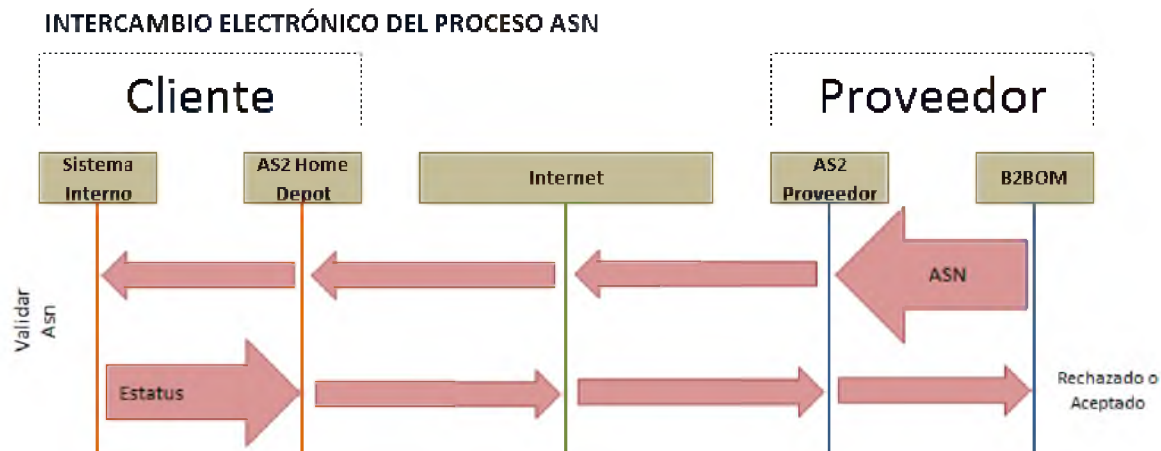


Figura 12. Proceso del ASN.

Consolidado y Estatus Consolidado

Se llega a esta parte cuando todos los ASN ya son aceptados por el cliente. Aquí se selecciona uno o varios Booking a embarcar en un solo camión. Se colocan las fechas más aproximadas a entregar la mercancía, una vez terminado se envía al cliente, se genera un XML [Anexo 10.15] y se valida la estructura a través de un XSD [Anexo 10.16] que contiene a detalle la información de lo que se va a embarcar. Si el consolidado es aceptado se genera en PDF un Packing List [Anexo 10.2] que contiene la lista de artículos por cada tarima de lo que se va a enviar y las Etiquetas [Anexo 10.3] que contienen un código de barras por cada tarima, así como su peso y volumen de todo el ASN.

En respuesta al consolidado enviado el cliente envía un XML [Anexo 10.17] que contiene si el documento consolidado fue aceptado o rechazado y es validado estructuralmente por un XSD [Anexo 10.18], si es correcto se traduce y actualiza el consolidado, si es incorrecto lo mueve a una carpeta de error.

En la figura 13. Proceso del consolidado, se muestra el flujo o proceso del Consolidado y estatus del consolidado.

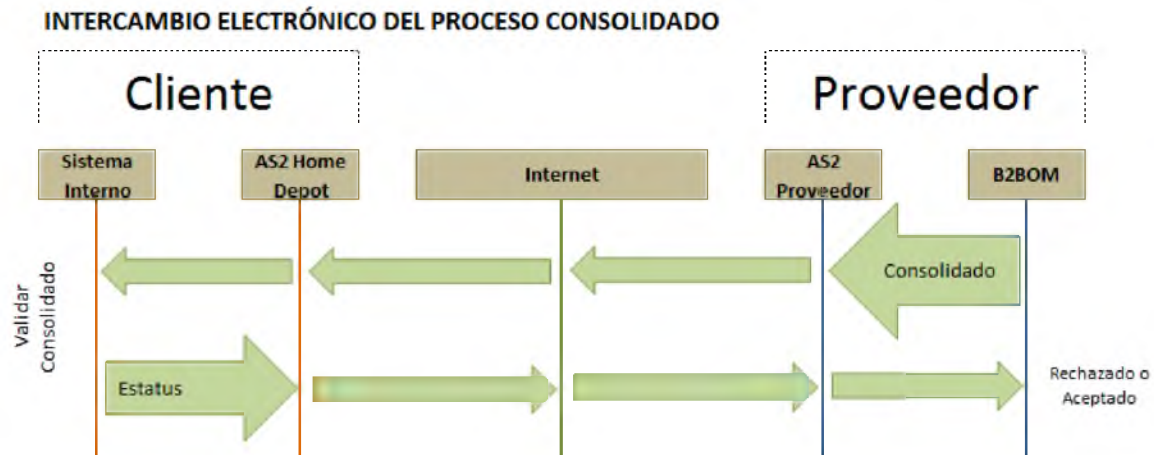


Figura 13. Proceso del consolidado.

Order Plan

Es un archivo XML [Anexo 10.19] que contiene un catálogo de artículos y las cantidades de lo que podría llegar a solicitar el cliente al proveedor, es decir, un plan de compra que podría llegar a realizar en las próximas 12 semanas. Este archivo XML es validado por un XSD [Anexo 10.20] para validar la estructura del archivo.

3.4 PERSONALIZACIONES

El análisis que se hace para desarrollar sistemas de software que se van a comercializar se realizan con la intención de cumplir las necesidades de la mayoría de los clientes; el sistema B2BOM fue realizado bajo esta premisa para que los proveedores se adapten a un mismo proceso de embarque. Sin embargo, existen proveedores que necesitan alguna función extra que pueda ayudarlos a embarcar de forma más rápida y sin contratiempos, entre otras; es por ello que se analizó la parte de incluir personalizaciones al producto de software, es decir, si el proveedor solicita alguna función extra en el sistema de software sin alterar el flujo principal de embarque. Las funciones personalizables a implementar son:

- ✓ En la lista de órdenes de compra colocar un botón que al seleccionar órdenes de compra pueda reenviar el correo electrónico a los destinatarios previamente configurados.
- ✓ En el módulo o sección de ASN, cuando este sea aceptado por el cliente el sistema deberá ser capaz de generar una hoja de recolección en PDF de forma especial con la información de las tarimas y cantidades; esto ayuda a embarcar de forma más rápida la mercancía.

- ✓ Que el sistema de software no reciba actualizaciones de las órdenes de compra (que esta parte esté bloqueada).
- ✓ Que el listado de los artículos estén ordenados primero por número interno y después por el modelo.
- ✓ El listado de órdenes de compra estén ordenados por fecha de entrega y número de proveedor.
- ✓ Si el proveedor utiliza archivos EDI de órdenes de compra puede descargar el archivo y no el XML.

3.5 MÓDULOS

Analizando la documentación entregada por el cliente, el sistema se dividió en siete módulos, siendo los cuatro primeros el flujo principal del embarque de la mercancía y el resto como apoyo para gestionar el sistema desde un punto administrativo.

1. Módulo Orden de compra.
2. Módulo Booking.
3. Módulo ASN.
4. Módulo Consolidado y diseño de PDF.
5. Módulo ERP.
6. Módulo Administración.
7. Módulo Catalogo.

En el plan de trabajo mencionado anteriormente el Módulo ERP no está incluido; cada proveedor tiene un diseño distinto para su sistema ERP, por lo que para cada implementación (instalación, configuración y capacitación) del sistema con un proveedor se realiza el desarrollo para generar la implementación del archivo o conexión al ERP para subir las órdenes de compra de forma automática.

3.6 CASOS DE USO

Los casos de uso son módulos que el sistema ejecuta en cada paso del flujo de un proceso; en la Figura 14. Casos de uso, se puede observar los módulos del sistema que el cliente va a ejecutar en el B2BOM y lo que el sistema ejecuta por cada módulo.

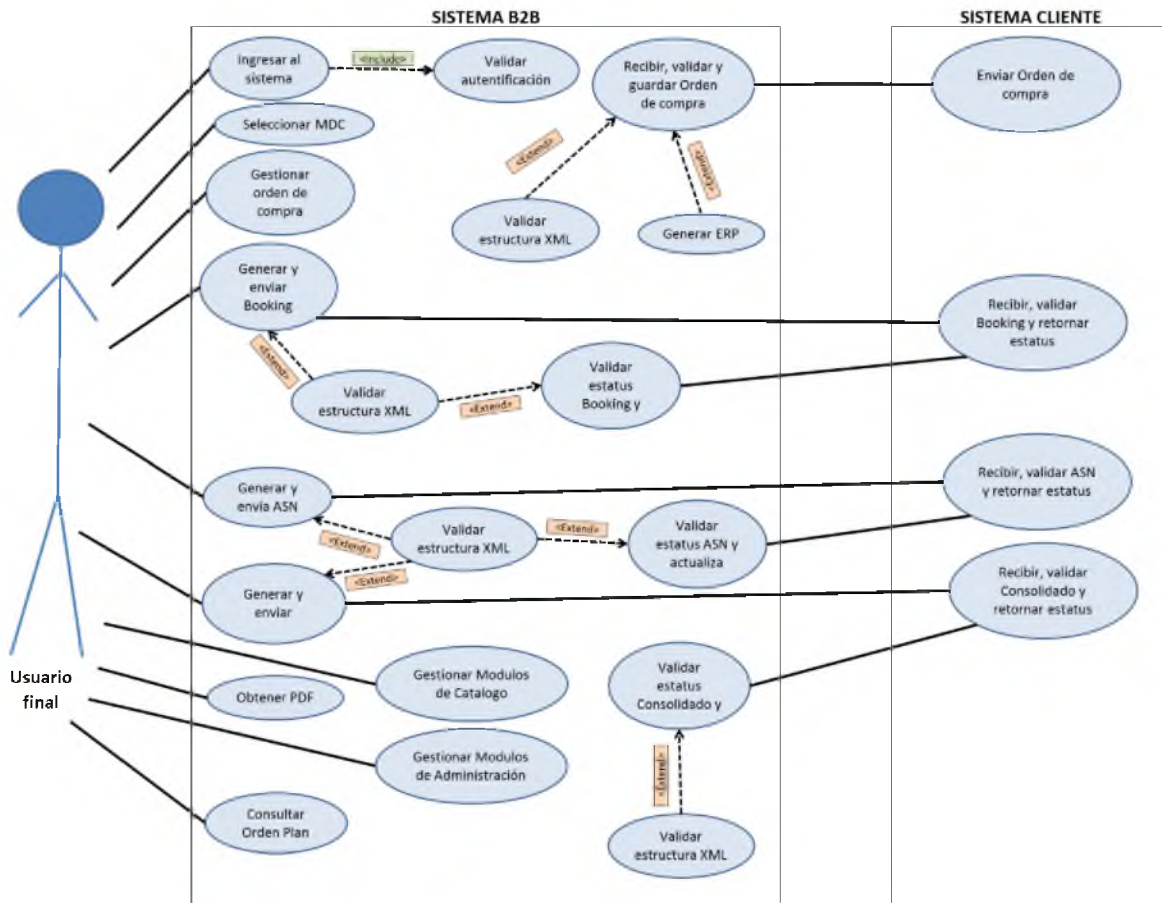


Figura 14. Casos de uso.

De forma general el usuario realiza las siguientes actividades:

- Ingresa al sistema.
- Selecciona el MDC.
- Gestiona la orden de compra.
- Genera el Booking y lo envía.
- Genera el ASN y lo envía.
- Genera el Consolidado y lo envía.
- Gestiona el módulo administrativo.
- Gestiona el módulo catálogo.
- Consulta el order plan.

Adicional, el sistema realiza validaciones por cada documento enviado o recibido, tanto en estructura como en la información que contiene.

CAPÍTULO 4. DISEÑO

Los sistemas de software regularmente pasan por una etapa de diseño, especificando sus componentes, módulos, paquetes, base de datos, tecnologías entre otros; para nuestro sistema B2BOM es importante diseñar la estructura del sistema dividiéndolo en subsistemas, el framework a utilizar, el diseño de la base de datos, los diagramas de clases, diagramas Web e interfaz. Esta etapa es importante porque da una idea general para la estructura de nuestro software y sobre todo al momento de programar. Los diseños son importantes para tener organizado el sistema, también ayuda para cambios futuros, es decir, si en algún momento se requiere añadir, eliminar o actualizar algún módulo, framework o alguna tecnología no sea difícil realizar esta tarea.

4.1 DISEÑO DE ALTO NIVEL

El diseño de un software tiene como fundamento organizar la estructura del sistema y dividir sus funciones en módulos o componentes; cada uno de estos realiza una función importante dentro del software; para el B2BOM es importante dividir los tres procesos de embarque (Booking, ASN y Consolidado); también dividir otras funciones como generar los XML, generar los PDF, los catálogos y el módulo administrativo. En la figura 8, Diseño de sistema, se puede observar los módulos o componentes del sistema.

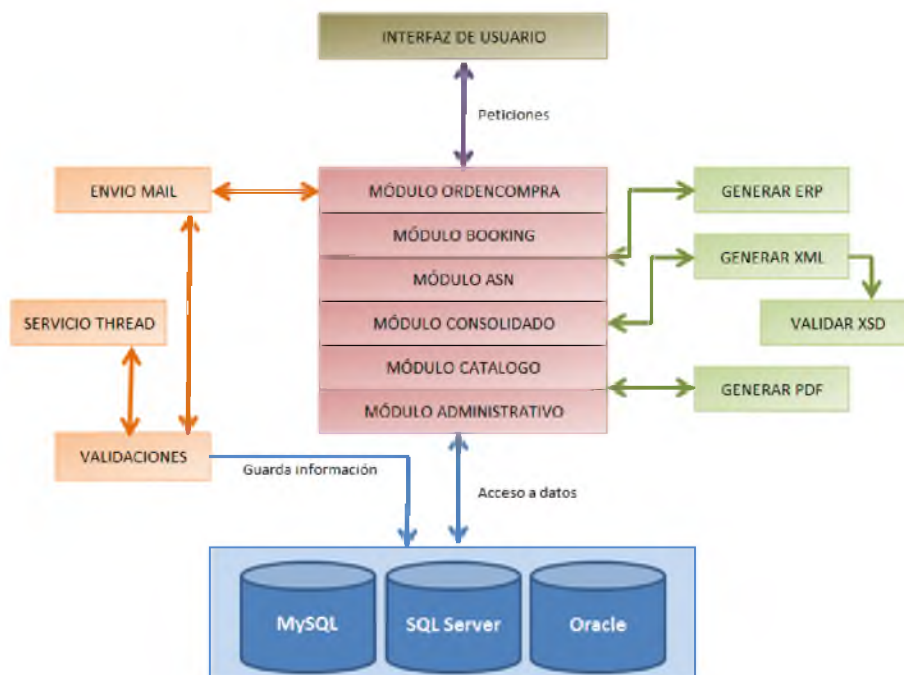


Figura 15, Diseño de sistema.

La estructura de la figura 8, Diseño de sistema, se observa a primera instancia la Interfaz de usuario, donde está el paquete de los archivos Web, lo que ve el usuario y donde realiza peticiones, es decir, las páginas JSP. Después están los módulos de orden de compra, booking, asn, consolidado, catálogo y administración; estos módulos reciben la petición del usuario desde la interfaz y dependiendo de que acción va a realizar ejecuta otros módulos.

Ejemplo de cuatro flujos del sistema B2BOM.

1. Si la petición del usuario es enviar un booking al cliente, este pasa por el Módulo Booking, utiliza el módulo Generar XML que a su vez utiliza la validación del archivo creado por medio de un XSD.
2. Si se desea enviar el archivo ERP, desde la petición del usuario utiliza el Módulo Orden de compra que a su vez utiliza el modulo Generar ERP.
3. Si el usuario desea generar un PDF, desde la petición de la interfaz utiliza el Módulo Consolidado que a su vez utiliza el Módulo Generar PDF y de este modo obtiene el PDF deseado.
4. El servicio Thread es un hilo que se ejecuta por default cada minuto (es parametrizable) valida los archivo XML entrantes y se guarda la información en la base de datos.

4.2 FRAMEWORK STRUTS

Los framework son herramientas de apoyo para los desarrolladores y son diseñados con la intención de facilitar el código del software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los detalles de bajo nivel de proveer un sistema funcional. Existen varios frameworks que apoyan al desarrollo de software entre ellos estan:

1. Hibernate.
2. Spring.
3. Struts
4. JSF.
5. ZKoss.
6. Primefaces.

Al desarrollar el sistema de software se tiene que adaptar a la imagen y productos Web de la empresa, estos diseños forman parte del sello de la compañía; la decisión de utilizar Struts y el diseño de las páginas Web es irrevocable, por lo que se debe utilizar el framework Struts 2.0 para el desarrollo de aplicaciones Web en JSP el cual hace que la implementación de las mismas sea más sencilla, más rápida y con menos complicaciones, al igual que más robustas y flexibles.

Struts 2 [3] es un framework de presentación, dentro de las capas en las que se divide una aplicación en la arquitectura JEE (Java Enterprise Edition), el cual implementa el controlador del patrón de diseño MVC (Modelo Vista Controlador), y que podemos configurar de varias maneras; además proporciona algunos componentes para la capa de vista.

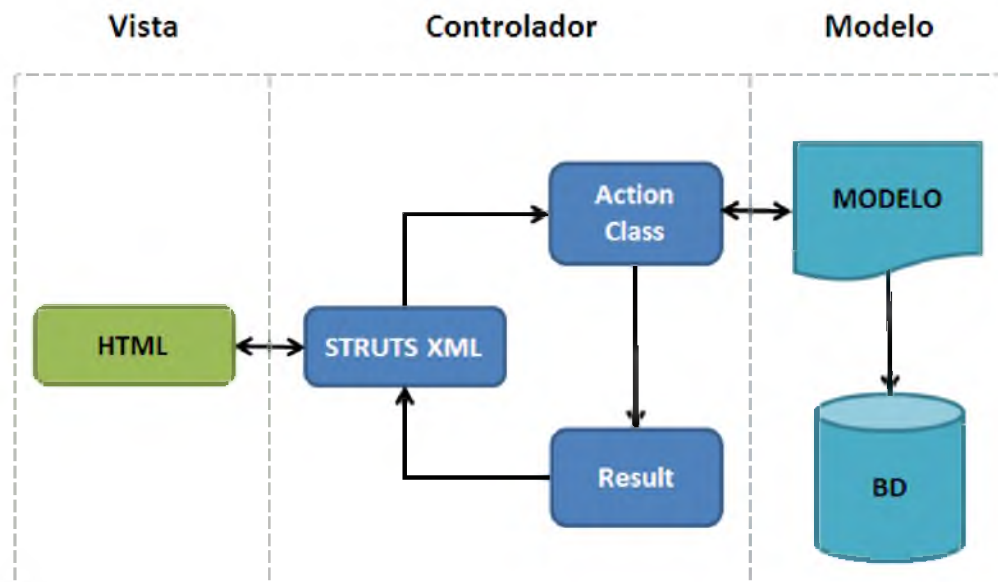


Figura 16. Struts 2.0

La figura 16. Struts 2.0, muestra el esquema del framework que trabaja de forma práctica y sencilla.

Vista. Es lo que ve el usuario y donde realiza las peticiones al sistema.

Controlador. Contiene la lógica de cada uno de los procesos.

Modelo. Es la información que se procesa, guarda, actualiza y elimina de la base de datos.

En la **vista** a través de una página HTML transformada en JSP se lanza una petición ejecutada por el usuario utilizando estructuras XML que son filtros entre las páginas y clases Java. En la parte del **controlador** que realiza una petición por

medio del filtro de un XML llegan a las clases Java y estas a través de un Bean (objeto) van a la base de datos a administrar la información en la parte del **modelo**, que es la encargada de administrar las conexiones a la base de datos y realizar las tareas correspondientes, es decir, administra la información en la base de datos.

4.3 DIAGRAMA DE PAQUETES

La organización para el desarrollo de un sistema es importante para identificar antes de visualizar el código que acción o acciones realiza esa estructura; cualquier sistema se debe de dividir en unidades más pequeñas, de modo que las personas puedan trabajar con una unidad de información limitada, a la vez y de modo que los equipos de trabajo no interfieran con el trabajo de otros. La gestión del modelo consiste en paquetes y relaciones de dependencia entre paquetes [10].

Estos diagramas especifican la organización que tiene el sistema; en el cual se pueden especificar dependencias, accesos o generalizaciones, este último es el más común; con esto se puede separar la lógica del sistema, la conexión a la base de datos, las herramientas o utilidades, versiones, etc. Contiene elementos del modelo al más alto nivel, tales como clases Java y sus relaciones, diagramas de casos de uso, interacciones y colaboraciones [10].

En la figura 17. Diagrama de paquetes, se puede observar la cabecera como paquete principal (com.azurian.b2b.mx), esto quiere decir que el sistema es comercial (com), empresa desarrolladora (azurian), nombre del sistema (b2b) y es desarrollado en México (mx). Después de esto se despliega varios paquetes que conforman el sistema.

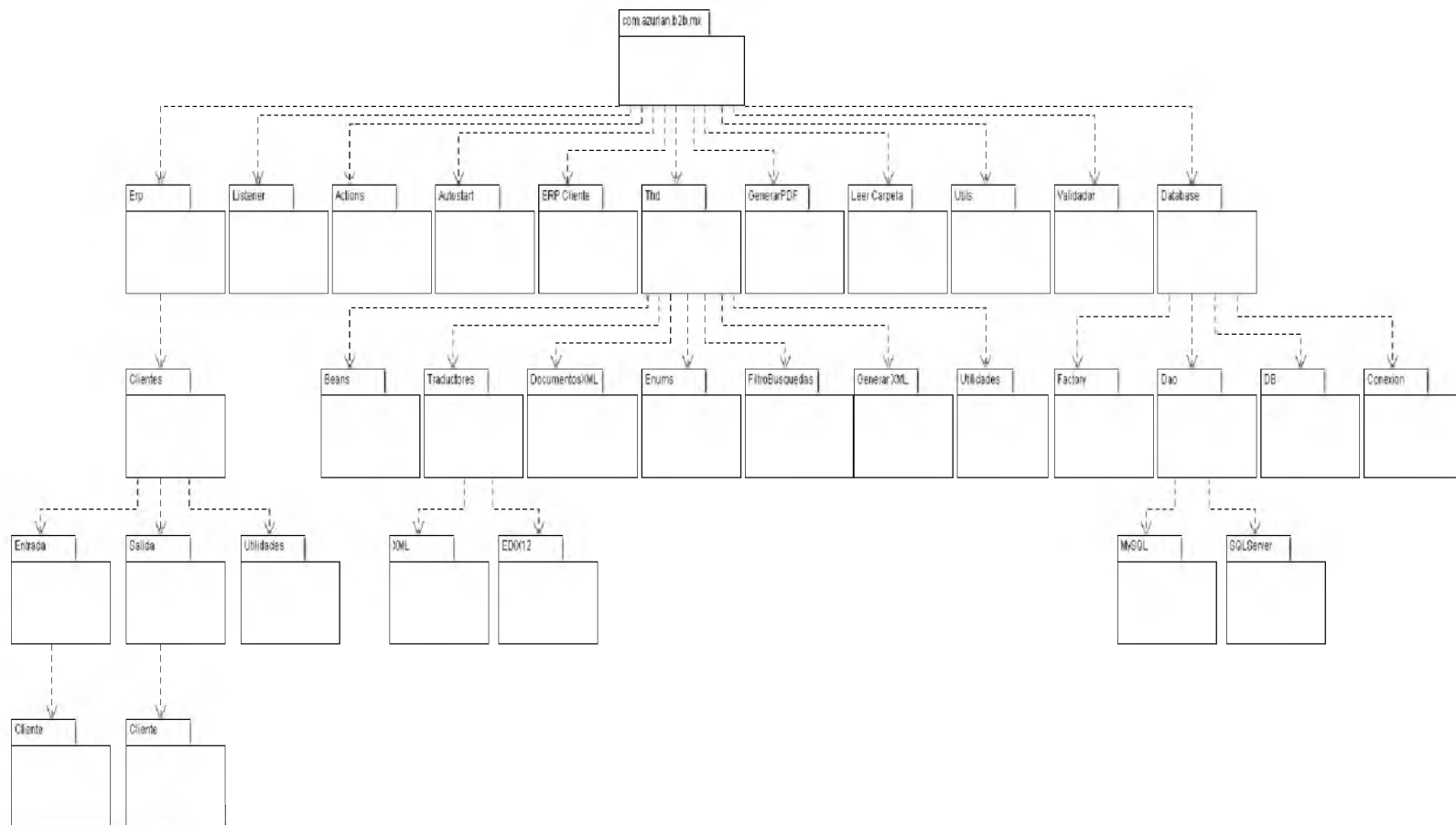
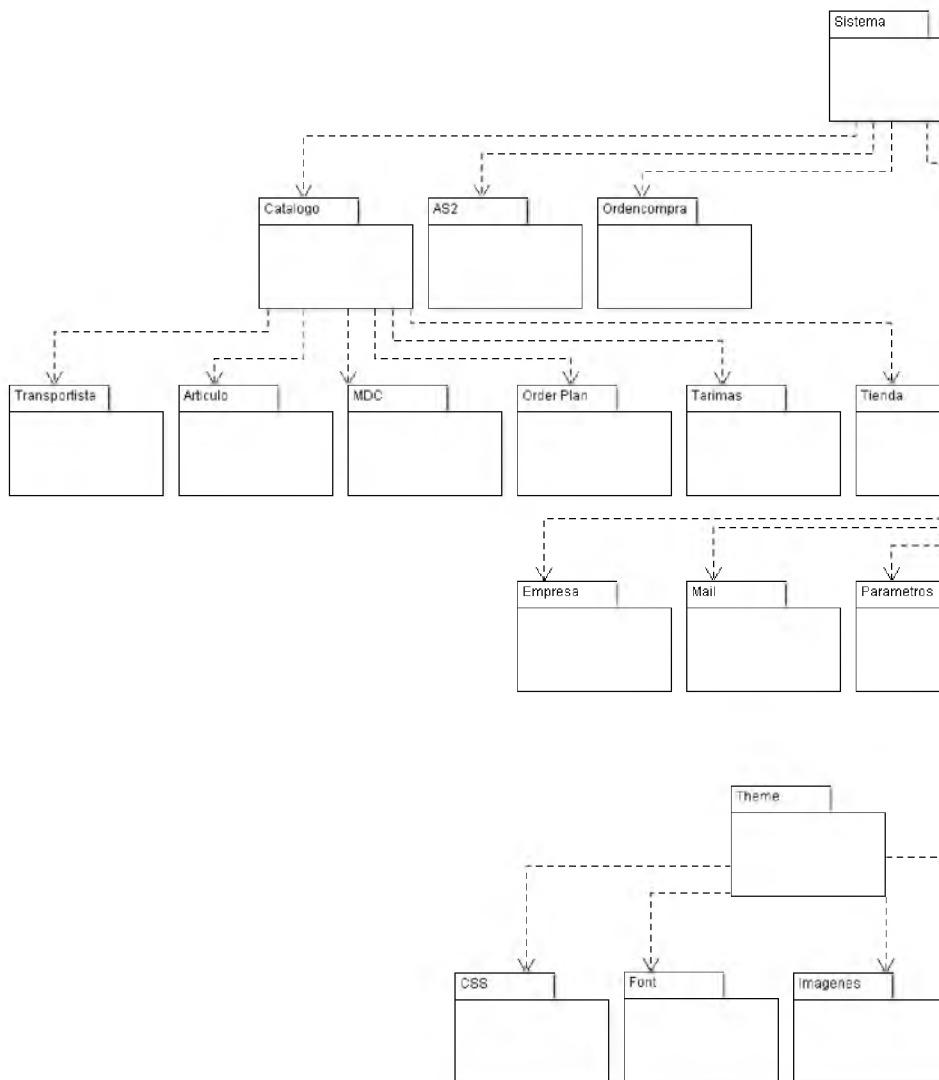


Figura 17. Diagrama de paquetes.

- **Erp.-** En este paquete se genera el archivo ERP especificado por el proveedor de cada orden de compra y subirlo al sistema administrativo interno. Accesa al paquete Cliente, después a Salida y por último a Cliente, aquí son depositados cada paquete con el nombre del cliente y su clase java para crear el documento.
- **Listener.-** Al iniciar el servidor de aplicaciones, se cargan las variables de contexto así como las personalizaciones del proveedor, si llegara a tener.
- **Actions.-** Cada acción del usuario que realiza en el B2BOM, se ejecuta un método de una clase asociada al módulo que está realizando, y devuelve un resultado a la página web.
- **Autostart.-** Cuando se inicia el servidor de aplicación se ejecuta un hilo (demonio) que va a estar escuchando una ruta en específico, en este caso es donde llegan las órdenes de compra, este hilo se ejecuta cada cierto tiempo.
- **ERP Cliente.-** Las órdenes de compra llegan y después de ser validadas y guardadas en la base de datos son enviadas a este paquete que las administra y consulta a que proveedor se le va a generar el archivo ERP, después se manda a llamar al paquete Erp.
- **Generar PDF.-** Aquí se generan los PDF del sistema, cada clase recibe un bean y devuelve un archivo en PDF.
- **Leer carpeta.-** Se ejecuta cada vez que se detecta alguna orden de compra, traduce, valida y aquí manda a llamar al Database para guardar la información.
- **Utilidades.-** Herramientas de apoyo en la lógica del sistema.
- **Validador.-** En el paquete Actions cuando el cliente realiza una acción, se validan componentes (beans) del Booking, ASN, Orden de compra y Consolidado. Por ejemplo el peso del Booking no debe exceder por lo estipulado, la fecha de las órdenes de compra no deben exceder del límite, etc.
- **Database.-** Aquí se utilizan clases abstractas que mediante un parámetro se identifica si el cliente utiliza como gestor de base de datos MySQL o SQL Server.
- **Thd.-** Aquí es donde están todos los objetos, traductores de XML o EDIX12 de las órdenes de compra, los objetos de los documentos XML, los filtros de búsqueda, generar los XML (Booking, ASN y Consolidado) y las herramientas o utilidades que usan estos paquetes.

La figura 18. Diagrama de paquetes Web, se muestra la distribución de las páginas Web (JSP), así como las herramientas de estilos, imágenes, tipos de letra y javascript. En el paquete Sistema se inicia sesión, se selecciona el MDC y se muestran los menús.



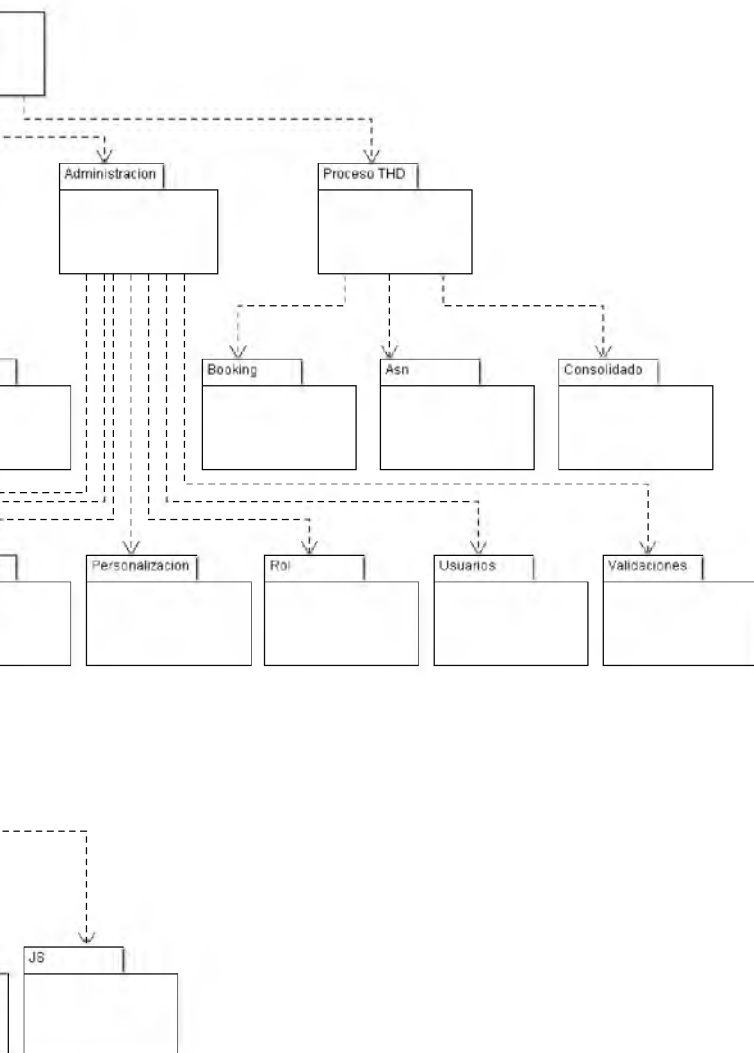


Figura 18. Diagrama de paquetes Web.

De izquierda a derecha se muestra lo siguiente en Sistema:

- **Catalogo.-** Aquí se muestran los catálogos de cada módulo, el transportista, los artículos, centro de distribución (MDC), order plan, las tarimas y las tiendas.
- **AS2.-** Cuando existe un error en la transmisión de un archivo por el canal del AS2, aquí se muestra la carpeta de errores, donde se puede seleccionar y reenviar el o los documentos.
- **Orden de compra.-** Se gestiona la orden de compra.
- **Administración.-** Se gestiona los usuarios, roles, validaciones, parámetros, correo, empresas y personalizaciones.
- **Proceso THD.-** Se crea, administra, elimina y envía los procesos del cliente para el embarque. (Booking, ASN y Consolidado).

Al nivel del Sistema el paquete Theme, contiene las herramientas que se usan para dar estilos, tipos de letra, imágenes que se usan y validaciones javascript.

4.4 DISEÑO DE LA BASE DE DATOS

Para realizar una base de datos, es importante empezar por un diseño que nos ayude a plasmar las ideas de nuestros procesos y englobe todo lo necesario e importante para una administración clara y eficiente.

El diseño se hizo analizando a primera instancia del proceso principal del embarque, Booking, ASN y Consolidado, después con la parte de los canales, catálogos y por último el módulo administrativo. En la figura 19. Base de datos, se puede observar la inclusión de nuestro proceso principal y también el diseño administrativo.

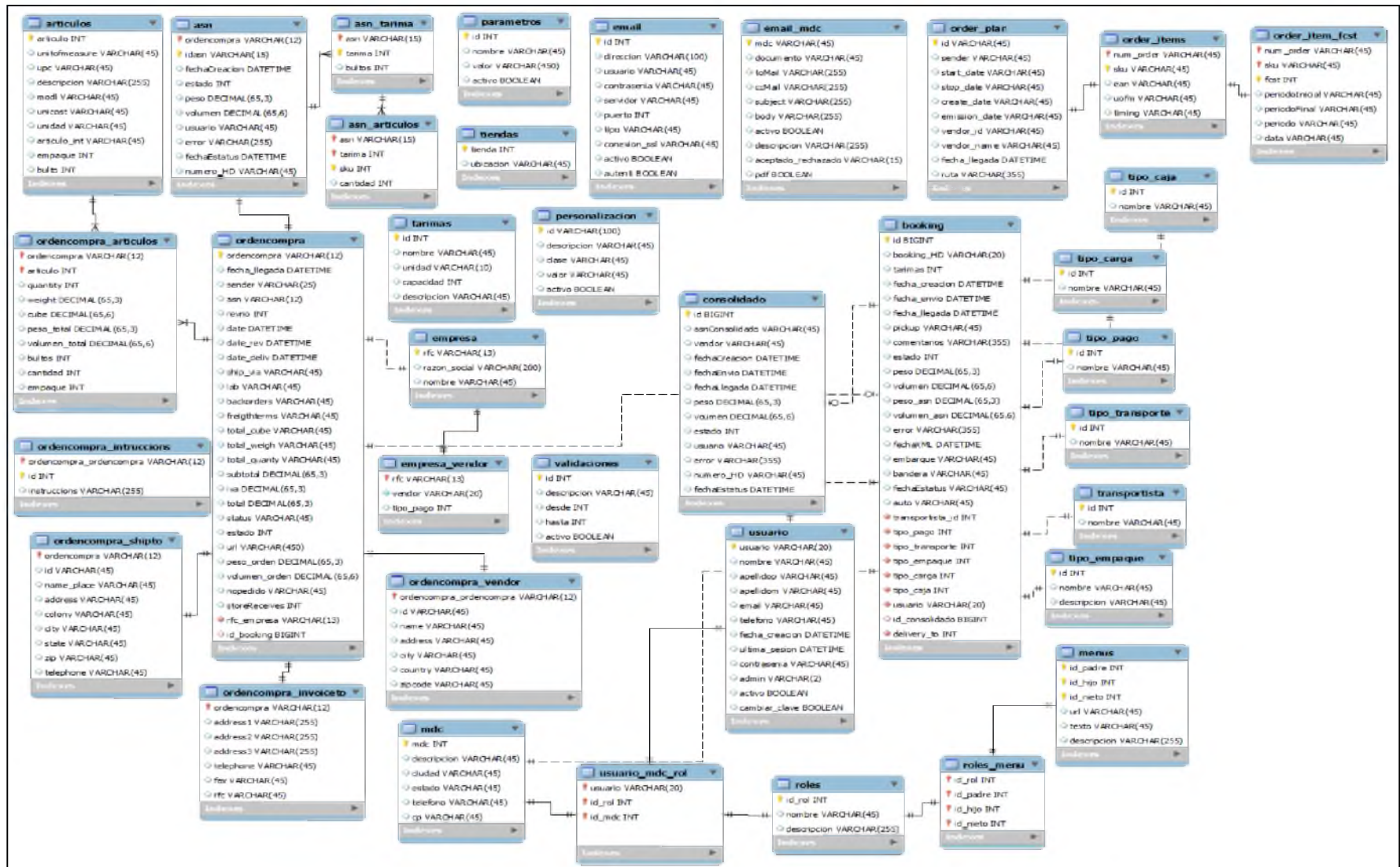


Figura 19. Base de datos.

Como se puede observar hay tablas que no están relacionadas, esto debido a que el sistema es multiempresa, es decir, no importa de qué empresa vas a agregar tu mercancía a un embarque, ejemplo, si la orden de compra A pertenece a la empresa AA y la orden de compra B pertenece a la empresa BB, puedes agregar las órdenes A y B a un embarque (Booking) sin importar a que empresa pertenece. A continuación en la tabla 1. Diccionario de tablas no relacionadas, se muestran las tablas que no tienen relación:

Tabla	Descripción
Parámetros	Los parámetros son de forma general, solo se necesita una llave primaria.
Tienda	Lista de tiendas activas por parte del cliente, no se relaciona con la orden de compra, porque no siempre existe una tienda, se da de alta desde el sistema; el cliente envía un correo al proveedor con la notificación de una apertura de tienda.
Email	Información de la cuenta de correo electrónico para envío de información, solo se necesita un registro.
Email_mdc	No tiene relación con la tabla anterior, no es necesario, ya que solo debe existir una conexión.
Tarimas	Lista de tarimas que se usan para construir el ASN de la orden de compra. No es necesario relacionarla, es de forma general y se usa para generar el ASN, no se lleva un histórico.
Peronalizaciones	Lista de registros que se usan para que el sistema realice ciertas actividades o procesos según sea cada proveedor. Tampoco es necesario relacionarla, ya que es de forma general y es utilizada en todo el proceso.

Tabla 1. Diccionario de tablas no relacionadas.

Cabe mencionar que el diseño del mismo no cumple con la funcionalidad de ser multiprovedores pero si multiempresa que pueden embarcar como una sola compañía. Es por eso que el diseño se analizó desde el punto de vista de un solo proveedor con varias empresas y un solo embarque.

4.5 DISEÑO DE LA INTERFAZ

Los sistemas que los usuarios usan para realizar sus actividades diarias regularmente lo identifican por como luce, los colores que tiene, botones e imágenes si es que contiene; desde la distancia se puede observar que sistema está usando el usuario sin la necesidad de preguntar. El diseño de la interfaz es un proceso que regularmente el desarrollador no está incluido pero si el diseñador,

que se encarga de implementar una estructura única en el cual contenga un sello de la compañía, sea vistoso y amigable para el usuario, con el fin de ayudar a complementar una fase en la parte comercial, el producto salga a la venta y tenga un sello en el mercado.

Para el sistema B2BOM el diseño de las interfaces no fue un problema, se tiene una estructura y un diseño único para los sistemas Web, por lo que esta etapa de diseño es práctico de implementar. A continuación aprovecharemos las interfaces para recorrer lo que el usuario final visualiza para el embarque de la mercancía. En la figura 20. Inicio de sesión. La ventana solicita al usuario un identificador con su respectiva clave para acceder al portal.



Figura 20. Inicio de sesión

Cuando un usuario ingresa sus credenciales el sistema lo envía a elegir qué MDC va a trabajar o realizar el embarque como en la figura 21. MDC. Solo existen dos, MDC MONTERREY y MDC MEXICO.

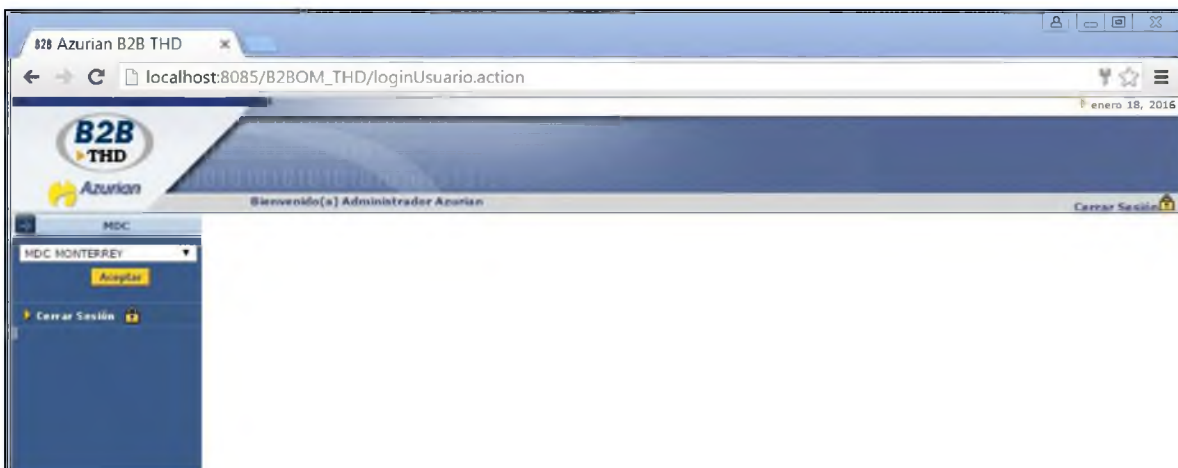


Figura 21. MDC

Al elegir el MDC el sistema lo enviará al usuario a realizar el proceso de embarque de las órdenes de compra como se muestra en la figura 22. Menús.

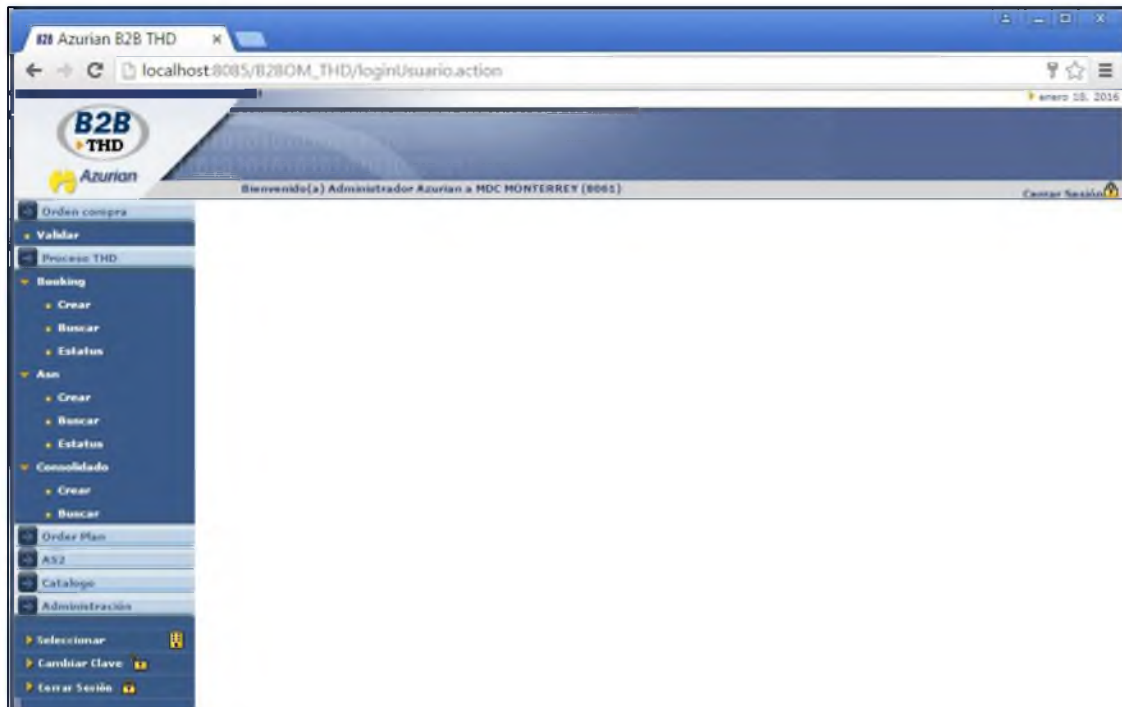
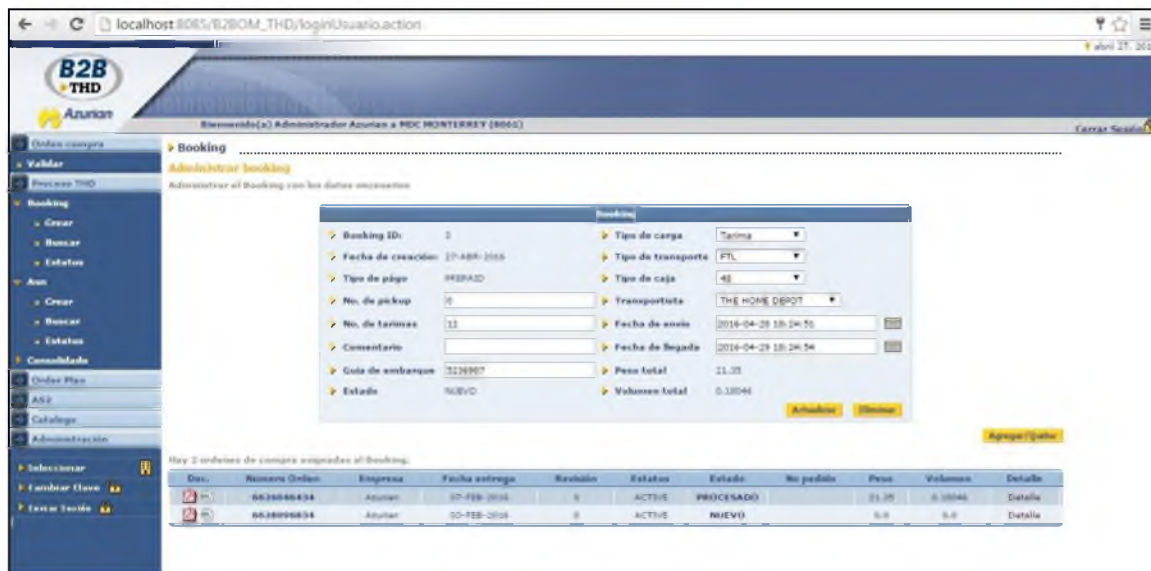


Figura 22. Menús

Booking

Una vez que el usuario accedió a los menús, puede empezar a realizar el proceso de embarque de la mercancía. Para esto se debe ir al menú Proceso THD, seleccionar Booking y presionar la opción Crear. Aquí selecciona las órdenes de compra que van a formar parte del Booking y después debe administrar el Booking como se muestra en la figura 23. Booking.



Asn

Quando el Booking es aceptado por el cliente se libera el módulo para generar ASN por cada orden de compra, eligiendo la tarima y colocando las cantidades correspondientes por cada una de ellas, en la figura 24. ASN, se muestra el entarimado de una primera tarima.

**Consolidado**

Cuando los ASN de cada orden de compra fueron enviados y aceptados cada uno se libera el módulo de Consolidado; es aquí donde se agregan cuantos Booking van a ir en el embarque (como se ha mencionado anteriormente un Booking es igual a un Consolidado que este respectivamente es igual a un Embarque). En la figura 25. Consolidado, se puede observar la parte administrativa de un consolidado con un Booking agregado y listo para su envío.

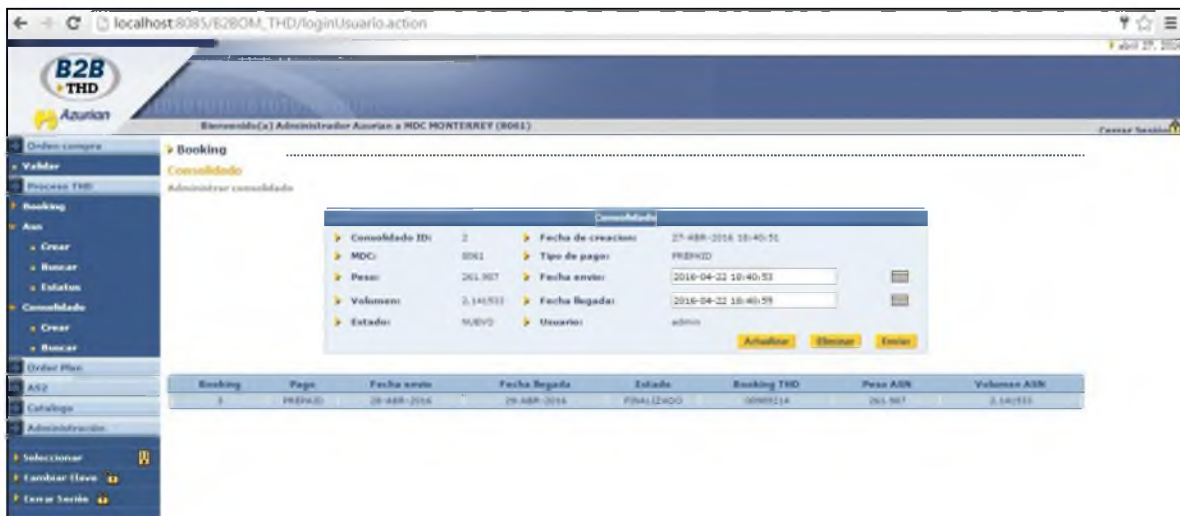


Figura 25. Consolidado.

Documentos PDF

El último paso para terminar el flujo de embarque es obtener los documentos PDF (siempre y cuando el Consolidado sea aceptado por el cliente) que van a ir con el embarque, el Packing List y las Etiquetas, estas van a ir pegadas en cada tarima de cada orden de compra. En la figura 26. Consolidado PDF, se observan los documentos listos para ser extraídos y ser impresos.

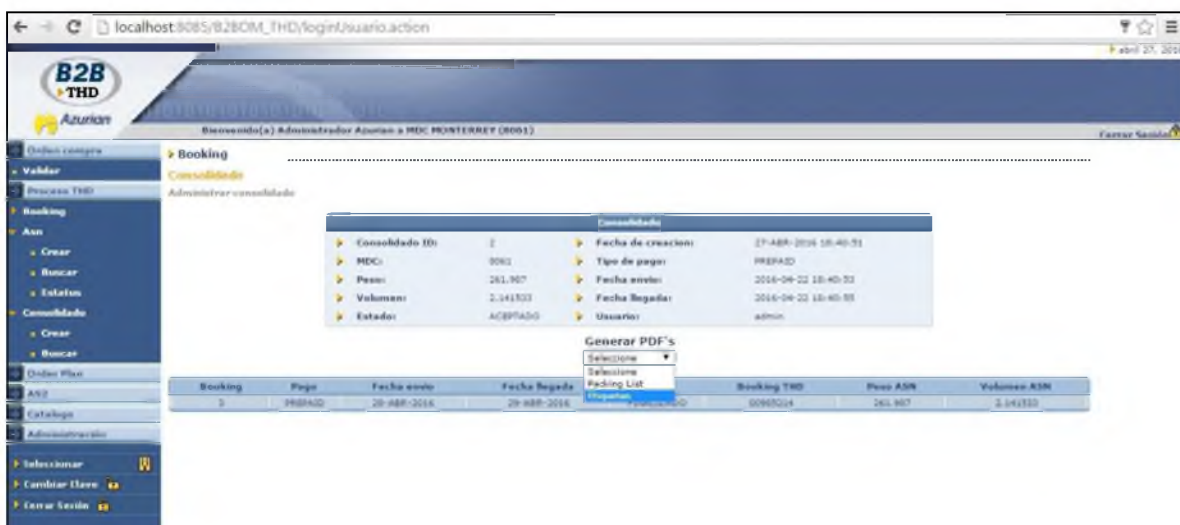


Figura 26. Consolidado PDF

CAPITULO 5. DESARROLLO

5.1 PUESTO EN DESARROLLO

Una vez concluido la fase de diseño en donde se especificaron los objetos, base de datos, paquetes, etcétera, se tiene que crear un ambiente del proyecto de forma interna que los desarrolladores van a utilizar, usar una interfaz de desarrollo (Netbeans, Eclipse, IBM Rational, etcétera), un gestor de base de datos y si existe más de un desarrollador es importante utilizar un Sistema de Control de Versiones (SVN) para trabajar sobre un mismo proyecto y así se tenga organizado el desarrollo. En el caso del sistema B2BOM a pesar de ser un solo desarrollador, es necesario utilizar un SVN para mantener el código respaldado por si se llegaran a dañar equipos o eliminar códigos, además si se intentó crear un módulo y este no funciona como lo esperado tener la oportunidad de regresar a una versión anterior mediante el SVN sin la necesidad de eliminar y realizar un esfuerzo mayor.

El principio que se tomó para empezar fue la de realizar los módulos principales del flujo de embarque, ya que es el núcleo del sistema B2BOM, el desarrollo del código tomó el siguiente orden:

1. Módulo orden de compra.
2. Módulo Booking.
3. Módulo ASN.
4. Módulo Consolidado y diseño de PDF.
5. Módulo ERP.
6. Módulo Order plan.
7. Módulo administración.
8. Módulo catálogo.

Empezar con el núcleo del sistema es importante para atender el principal detalle; conforme se va avanzando e integrando con los demás módulos se pueden encontrar algunos detalles que impliquen modificar algún parámetro para un mejor rendimiento del software a diferencia de si se empieza al revés, esto duplicaría el esfuerzo realizado y no se estaría atacando lo que realmente tiene importancia.

5.2 DIAGRAMA DE CLASES

Los diagramas de clases son aquellos que especifican la estructura de un sistema, sus atributos y métodos, ayudan a simplificar la programación del mismo y a verificar el diseño del sistema relacionándolos para que se construyan modelos eficientes y estructurados que se van a utilizar a lo largo de la programación.

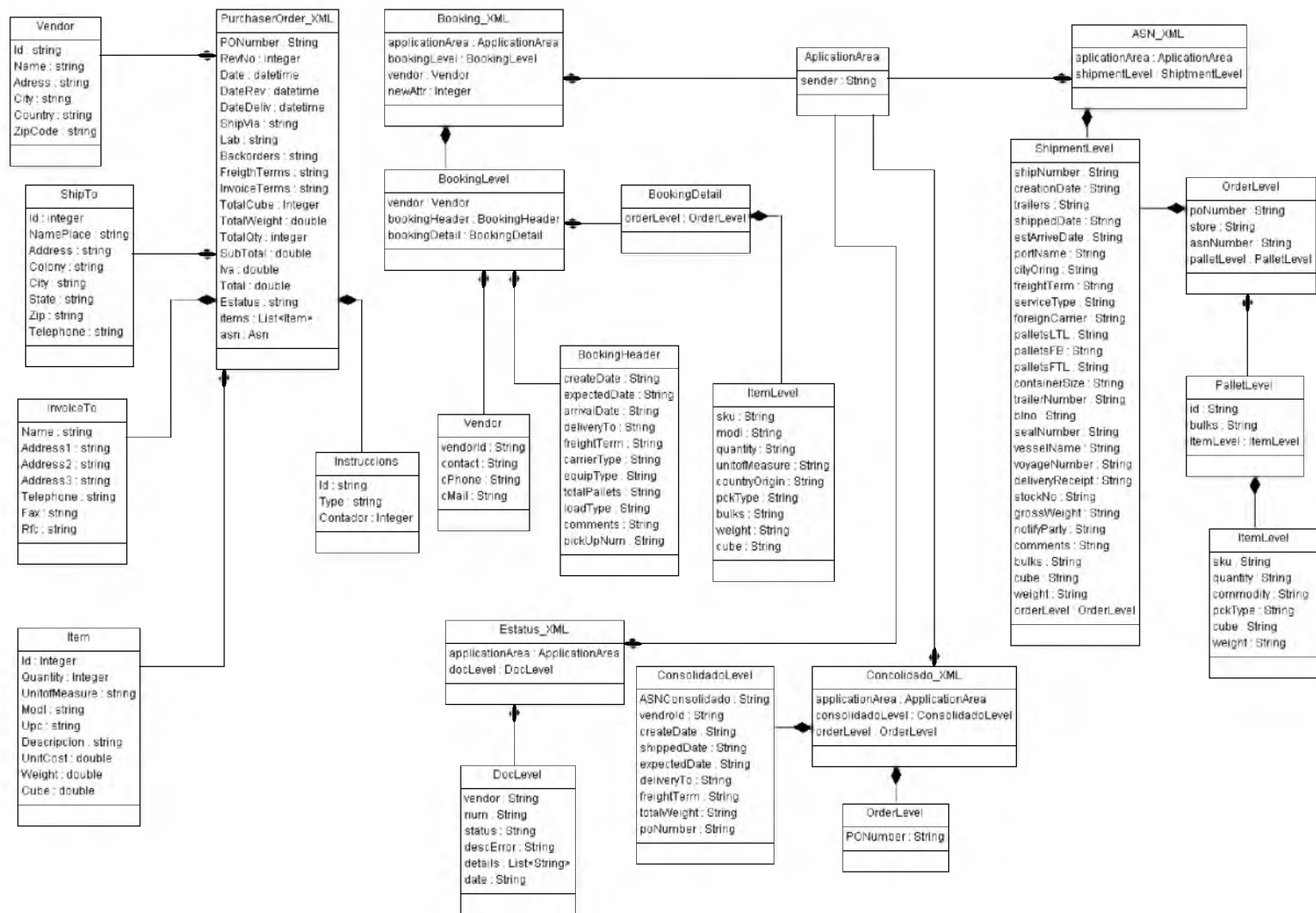


Figura 27. Diagrama de clases XML.

Objetos XML y PDF

Como se mencionó anteriormente, las clases que nos ayudan a generar los XML de intercambio electrónico son objetos que tienen la estructura de las mismas con el mismo nombre al que pertenece la etiqueta, esto con el fin de solo llamar al método correspondiente, enviarle la clase validada y evitar errores; los objetos Booking_XML, ASN_XML, Consolidado_XML y Estatus_XML son objetos que contienen la estructura del XML a utilizar en el sistema y solo son utilizados para crear los documentos XML. Adicional se crearon clases con la estructura de los PDF (Etiqueta y PackingList) para un mejor tratamiento a la hora de generar los archivos; se utiliza el mismo procedimiento que los objetos XML, se llama al método correspondiente, se envía la clase y se genera el archivo PDF. En la figura 28, Diagrama de clases PDF, se observan los objetos para construir los PDF; la clase PurchaserOrder XML pertenece a una clase de objeto XML.

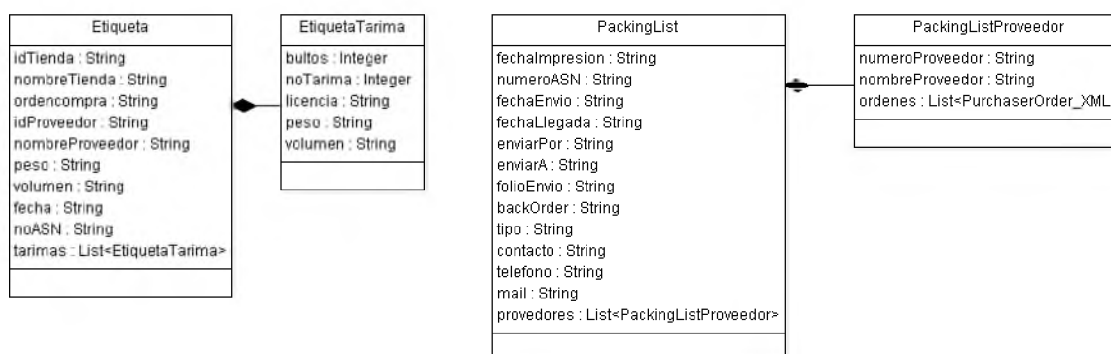


Figura 28. Diagrama de clases PDF.

Diagrama de clases del sistema

La separación de las clases del sistema a los objetos XML consiste en administrar de mejor forma cada elemento con su respectivo valor. En la figura 13, Diagramas de clases, se puede observar las clases principales del flujo del proceso principal de embarque (Ordencompra, Booking, ASN y Consolidado) que se utilizan para la interacción del usuario con el sistema.

Clase XML	Clase estándar	Descripción
PurchserOrder_XML		El objeto XML del documento.
Booking_XML	Booking	
Asn_XML	Asn	
Consolidado_XML	Consolidado	

Tabla 2. Diferencias de clases.

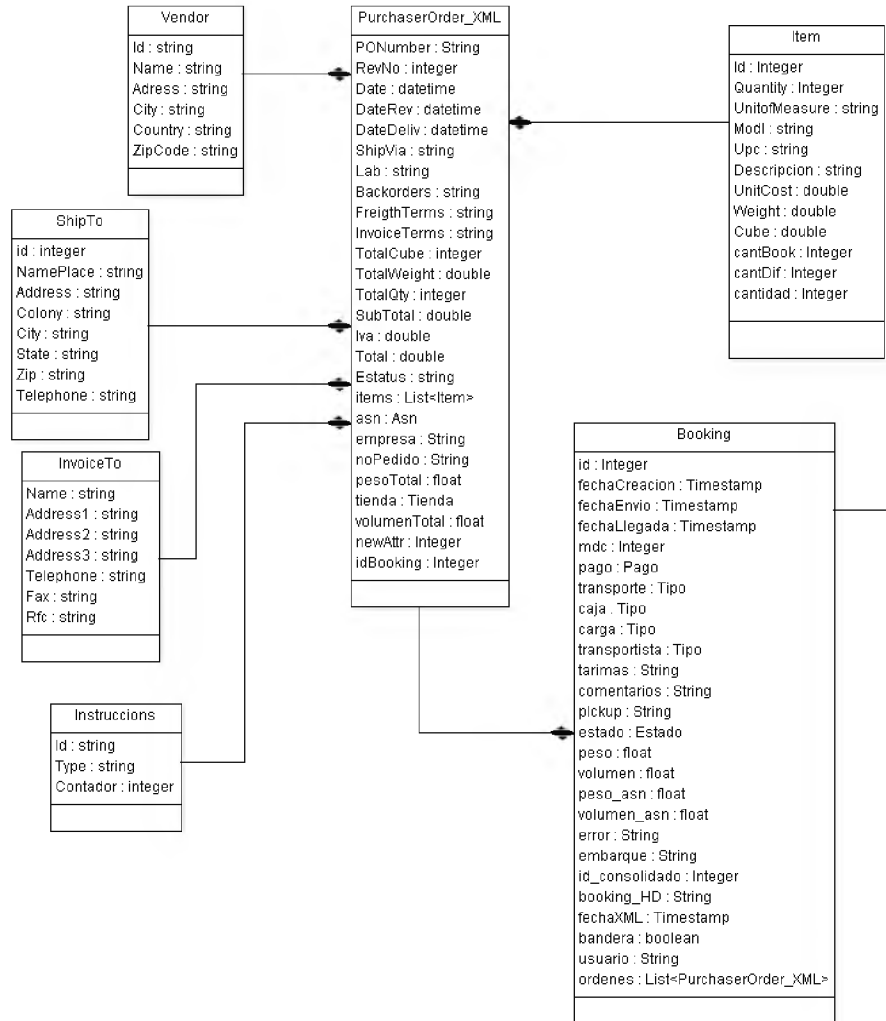
En la tabla 2. Diferencias de clases, se observan las clases estándar que corresponden a la interacción entre el usuario y el sistema, sin ninguna estructura que corresponda a un documento XML; estas clases se utilizan por todo el flujo del proceso de embarque hasta que son enviados al cliente, momento en el cual se utilizan los objetos que dan forma al XML.

Hacer énfasis en la diferenciación de estas clases nos ayudan a una mejor administración sobre todo en código, por lo que es una manera fácil de entender que al momento de crear un archivo físico se entrega un objeto en forma de XML a un método que se encargará de entregar información en cada etiqueta del archivo a crear o se envía un objeto vacío a un método para la recepción del documento y mientras sea traducido el archivo el objeto se vaya llenando conforme a su etiqueta; para los documentos de intercambio electrónico ésta es la manera de generar cada archivo Booking, Asn y Consolidado en formato XML para la emisión, y para la recepción de documentos como Orden de compra y Order plan.

En la figura 29. Diagrama de clases, se puede observar cuatro clases principales:

1. PurchaserOrder_XML.- Contiene la estructura del XML de la orden de compra añadiendo campos adicionales que nos ayudan a darle un tratamiento durante todo el proceso; esta clase se usa para la recepción del documento, emisión y ser utilizado en la interacción con el usuario.
2. Booking, Asn y Consolidado-. Objetos que solo se utilizan para la interacción entre el sistema, el usuario y no es funcional para crear el documento XML del Booking.

Los objetos restantes apoyan la estructura de los objetos antes mencionados, las clases Vendor, ShipTo, InvoiceTo, Instruccions e Item completan el objeto PurchaserOrder_XML, el objeto Tarima es parte del objeto Asn. En resumen, estas clases solo funcionan para la interacción entre el sistema y el usuario a excepción del objeto PurchaserOrder_XML que funciona para ambos procesos.



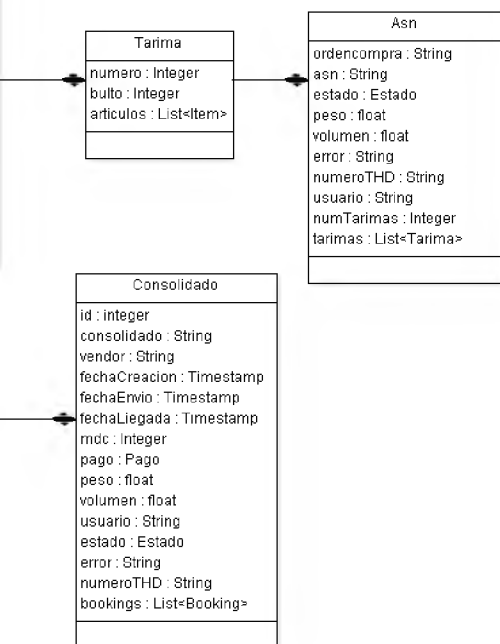


Figura 29. Diagrama de clases.

Administración y catálogos

La gestión de un sistema es importante para determinar ciertas acciones dentro del mismo, controlar el proceso a la que está destinado el sistema así como gestionar hasta qué punto puede llegar hacer uso el usuario, determinar la configuración; el sistema B2BOM está dividido en dos partes, administración y catálogo, en donde el administrador del sistema puede controlar accesos, permisos, correos electrónicos, configuración de correo, empresas, centros de distribución entre otros módulos. En la figura 30, Diagrama de clases de administración, se puede observar las clases que nos ayudan a gestionar los módulos administrativos y de catálogos.

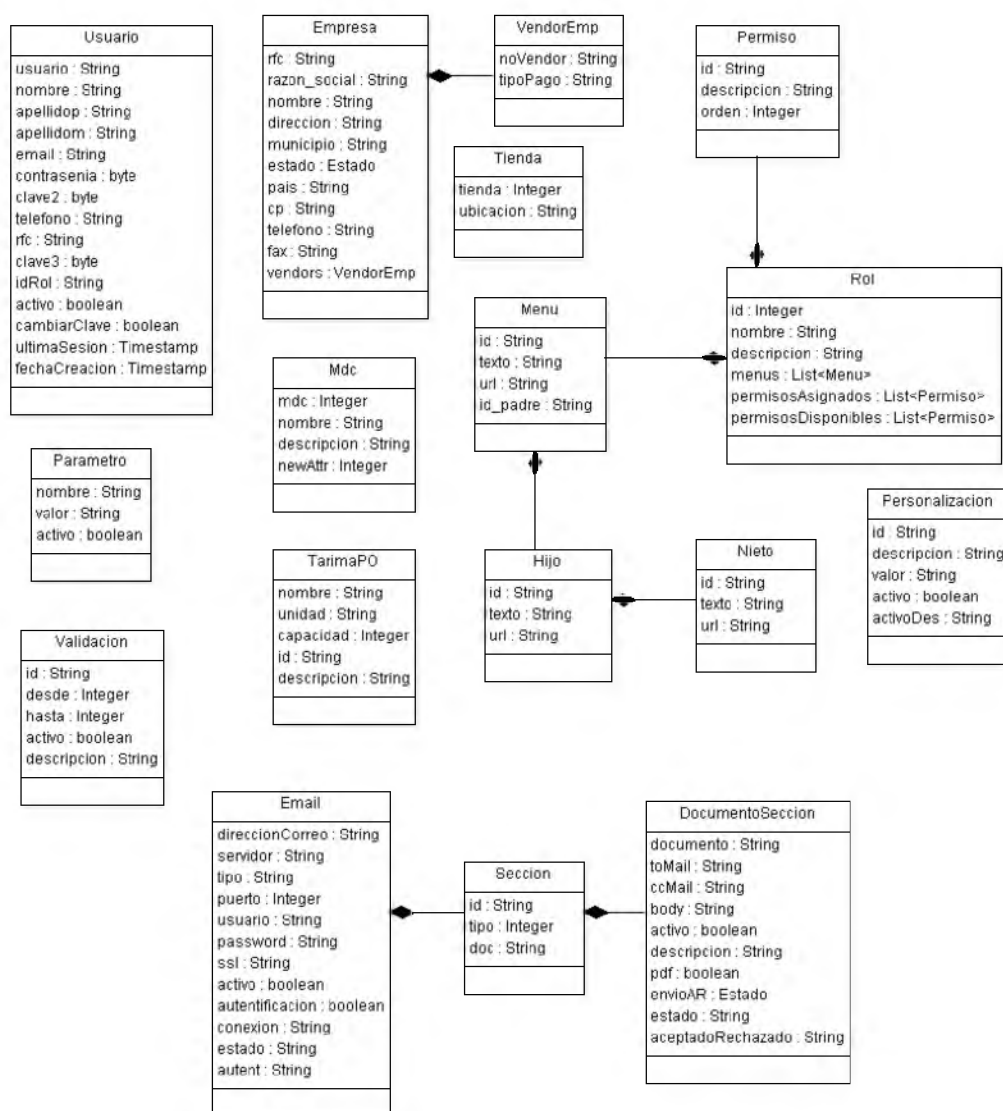


Figura 30. Diagrama de clases de administración.

Filtros de búsqueda

Las búsquedas son de gran ayuda para obtener resultados deseados, es decir, el sistema B2BOM guarda gran cantidad de información para el intercambio de documentos entre el cliente y el proveedor, y el cargar toda la información se requiere de una memoria importante en un servidor, es por eso que para una mejor y estricta búsqueda se utilizan en cada módulo filtros de búsqueda para los procesos de Orden de compra, Booking, Asn, Consolidado y catálogo de artículos, el cual trae como resultado los objetos mostrados en la figura 31, Diagrama de clases de filtros. En la figura 31. Diagrama de clases de filtros, se puede observar las clases que se usan para la búsqueda por cada módulo en específico.

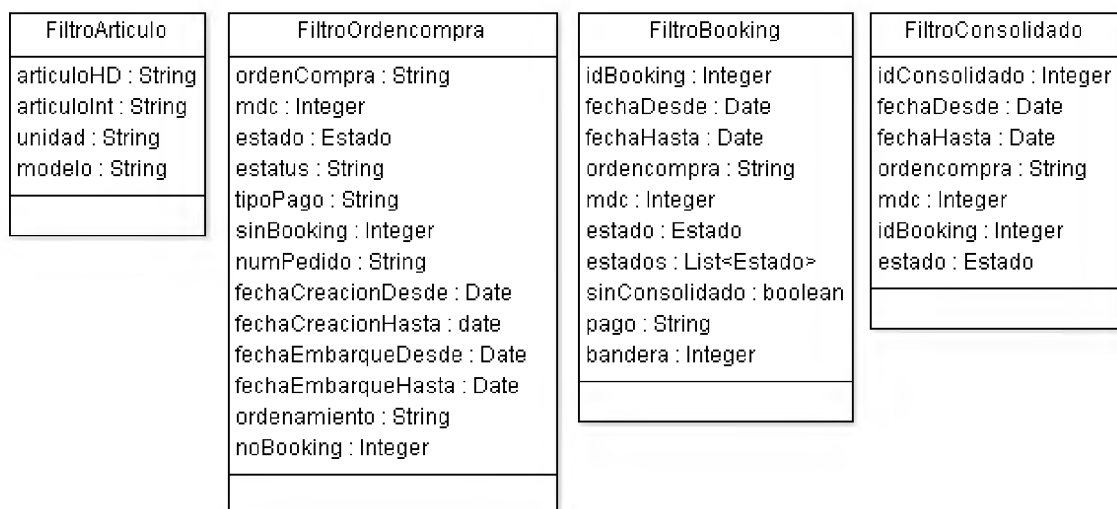


Figura 31. Diagrama de clases de filtros.

Actions

Los eventos son aquellos donde realmente interactúa el usuario con el sistema; son las acciones que se realizan durante el uso del sistema para obtener una respuesta de la acción correspondiente. Cada vez que el usuario da un click a un evento se ejecuta una acción y realiza un proceso. Estas clases se conocen en Struts como Action (heredando de la clase de ActionSupport) y son filtros entre la página web y la clase Java, es decir, todas las variables que se declaran en estas clases pueden ser vistas por las páginas Web a las que están asociadas. En la figura 32. Diagrama de clases Actions, se observa el proceso principal de embarque de la mercancía (Ordencompra, Booking, Asn y Consolidado).

OrdencompraAction
ordenesForm : List<PurchaserOrder_XML> ordenForm : PurchaserOrder_XML articulosForm : List<Item> articuloForm : Item empaqueForm : Tipo empaquesForm : List<Tipo>
execute() buscar() detalleOrden() actualizarDetalleOrden() regresarLista() eliminar() updateStatus() generarPDFS() generarExcel() envioMail() envioErpOrden() obtenerPdf() obtenerXml() obtenerEdi()

BookingAction
ordenesForm : List<PurchaserOrder_XML> ordenesDisponiblesForm : List<PurchaserOrder_XML> ordenesAsignadasForm : List<PurchaserOrder_XML> bookingForm : Booking cajas : List<Tipo> empaques : List<Tipo> transportes : List<Tipo> transportistas : List<Tipo> bookingsForm : List<Booking> cargas : List<Tipo> pago : List<Pago> filtroBooking : FiltroBooking
execute() buscar() crear() buscarOrdenes() asignarCrearBooking() buscarBooking() actualizarBooking() estatus() detalleOrden() actualizarDetalleOrden() eliminar() enviar() regresarBooking() asignarOrdenesBooking() asignarOrdenes() desligarOrdenes() obtenerPdf() obtenerXml() obtenerEdi()

AsnAction
bookingForm : Booking bookingsForm : List<Booking> ordenesForm : List<PurchaserOrder_XML> ordenForm : PurchaserOrder_XML tarimaForm : Tarima tarimasForm : List<Tarima> asnForm : Asn filtroAsn : FiltroAsn transportistasForm : List<Tipo> posicion : Integer
execute() buscar() buscarBooking() listarBooking() obtenerBooking() ordenAsn() consultar() next() before() tarima() entarimar() eliminar() finalizar() enviarASNS() enviarASN() estatus() cancelar() activar() actualizarBooking()

ConsolidadoAction
bookingForm : Booking bookingsForm : List<Booking> consolidadoForm : Consolidado consolidadosForm : List<Consolidado> documentoPdf : byte
execute() listarBookings() asignarCrear() buscar() buscarConsolidado() actualizar() eliminar() administrar() enviar() generaPDF() getPackingList() getEtiqueta()

Figura 32. Diagrama de clases Actions

Acceso a la base de datos DAO

La interacción con la base de datos se realizan a través de clases abstractas y dependiendo de la configuración del sistema se accede al gestor utilizado, por ejemplo, la clase B2BFactory realiza la comparación del gestor a utilizar y regresa una instancia de la clase que se ocupara para realizar el proceso en el gestor utilizado. En la siguiente línea veremos cómo se accede a la base de datos:

```
B2BFactory.getInstance().getTienda().add(Tienda tienda);
```

B2BFactory.- La clase que realiza la comparación del gestor a utilizar y tiene todas las clases abstractas.

getInstance().- Método que realiza la comparación del gestor a utilizar y asigna la instancia.

getTienda().- Clase abstracta que contiene la instancia a la clase TiendaDAOImpl.

add().- Método que agrega una tienda a la base de datos instanciada.

En la figura 33. Diagrama de clases DAO parte 1, se puede observar las dos clases que realizan la tarea de comparar y realizar la instancia correspondiente.

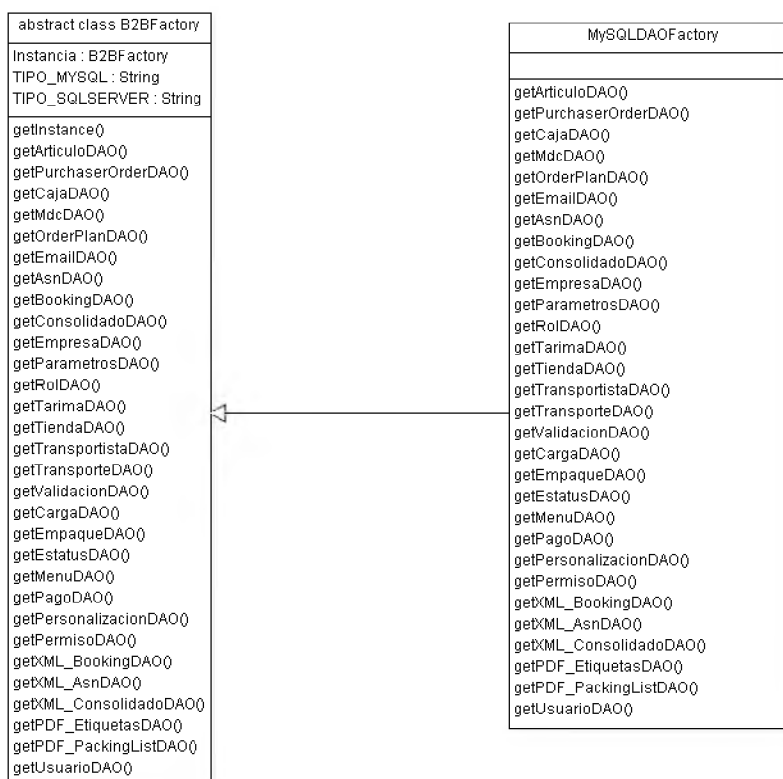


Figura 33. Diagrama de clases DAO parte 1.

En las figuras 34. Diagrama de clases DAO parte 2, 35. Diagrama de clases DAO parte 3 y 36. Diagrama de clases DAO parte 4 (clases separadas en 3 figuras para una mejor comprensión) se observa las clases declaradas como “*abstract interface*” para realizar la instancia a la clase que corresponde según sea el gestor a la base de datos.

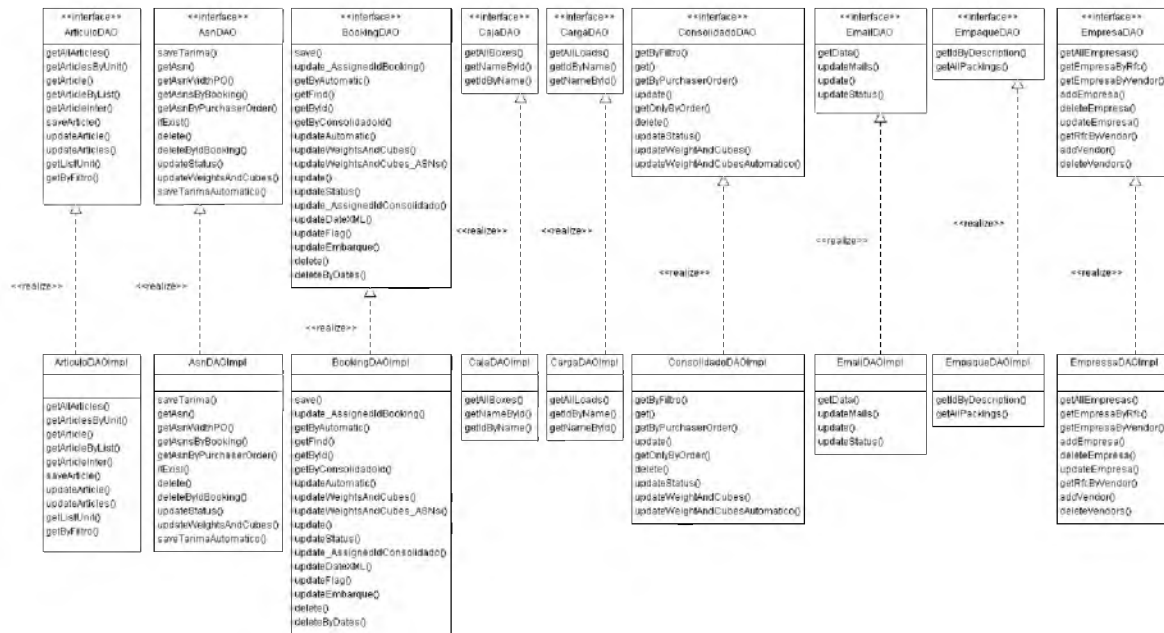


Figura 34. Diagrama de clases DAO parte 2

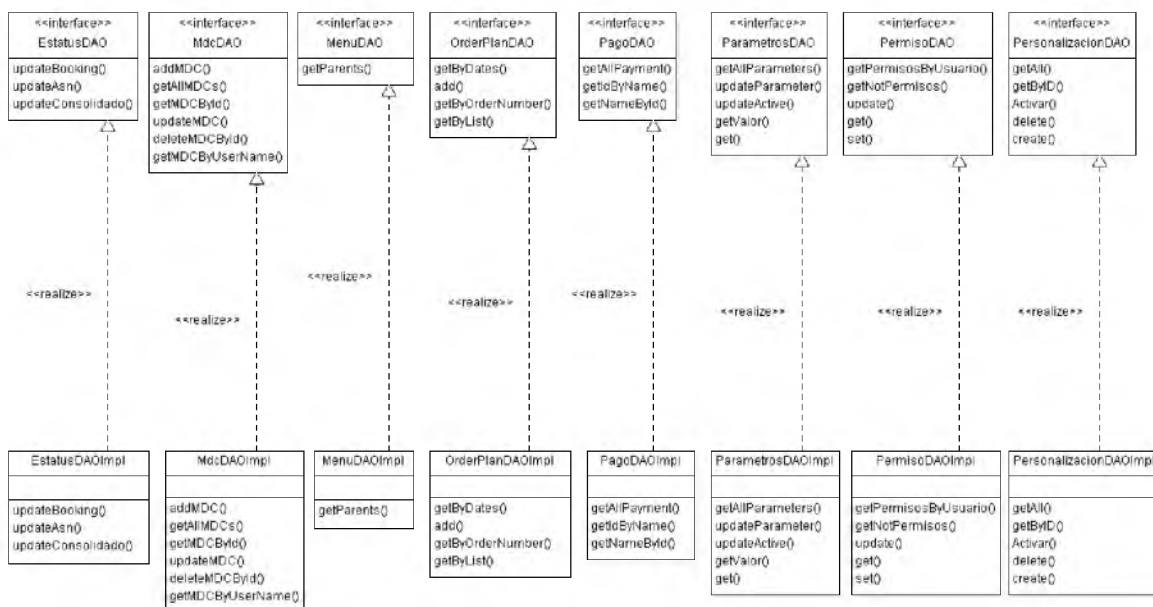


Figura 35. Diagrama de clases DAO parte 3

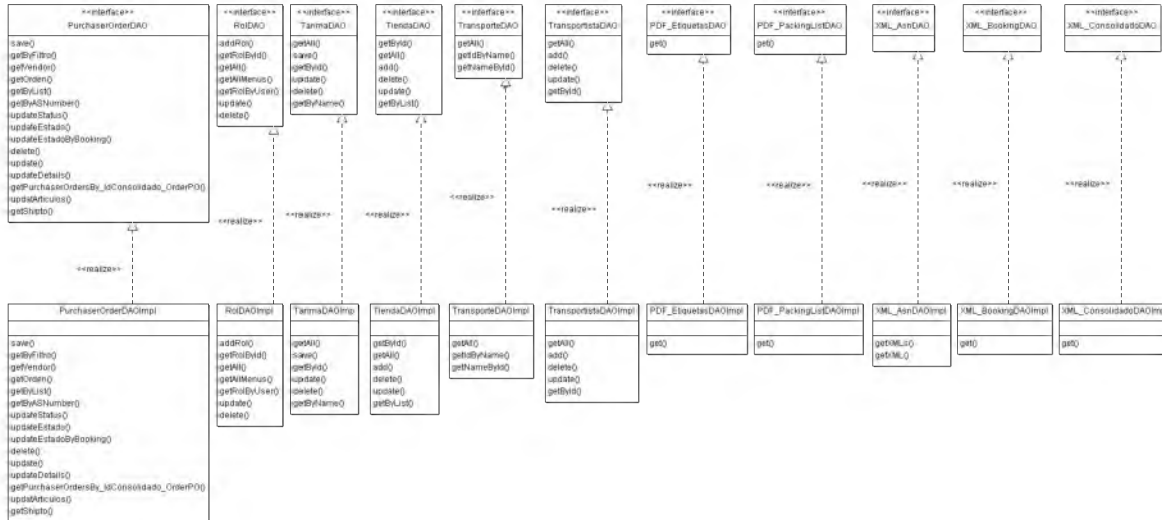


Figura 36. Diagrama de clases DAO parte 4

Con estas clases abstractas se cumple el objetivo de que el sistema sea multi base de datos, es decir, en el mismo sistema se puede utilizar el gestor de base de datos que el proveedor requiera para su instalación. Por ahora se tienen dos gestores de base de datos MySQL y SQL Server.

Si un proveedor solicita que el sistema este montado en un gestor diferente solo se implementa las clases abstractas y se añade un paquete con terminación de gestor a utilizar; para esto se necesitan hacer pruebas con el gestor nuevo aunque utilicen las mismas sentencias SQL.

5.3 COLABORACIÓN Y REVISIONES

Los grandes sistemas están realizados con colaboración de quiénes lo van a utilizar, parte de una idea principal de hacia donde se quiere llegar y qué se quiere hacer. Además en la mayoría de las ocasiones se involucran personajes para una mejor comprensión y desarrollo.

En nuestro caso, la principal ayuda viene del cliente, del departamento de logística que se encarga de realizar toda la lógica de embarque de la mercancía, rutas, tiendas, centros de distribución, sistemas. Se realizaron juntas constantes sobre cómo se utilizaría las órdenes de compra para embarcar la mercancía, que documentos se utilizarían y cuál era el propósito de cada proceso hasta llegar al centro de distribución. Además, en cada avance significativo se realizaron juntas para validar el sistema Web, apuntar ajustes y mejoras; siempre con la intención de apoyar al usuario en el camino a la lógica del embarque y poder entregar la mercancía detalladamente.

Anteriormente se ha comentado que el cliente no desea un software a la medida, es decir, no importa cómo se realice siempre y cuando cumpla con las características o reglas de negocio establecidas anteriormente, por lo que las revisiones y juntas con el cliente se atacó el flujo principal.

CAPÍTULO 6. PRUEBAS

El desarrollo de un sistema no es complejo si se tiene las herramientas necesarias de análisis que previamente se realizaron. En este caso es el documento emitido por el cliente con la lógica de negocio y restricciones en cada proceso de embarque.

Se tienen tres fases de pruebas.

- 1) Pruebas internas.
- 2) Pruebas de cliente solicitante.
- 3) Pruebas de cliente piloto.

Al desarrollar cada módulo este va de la mano con la documentación entregada por el cliente, especificando las reglas de negocio que se tiene en cada uno de los puntos clave.

6.1 PRUEBAS INTERNAS

Una vez terminado el módulo se hacen estas pruebas donde se detecta si realmente el modulo realiza el proceso correcto, de no ser así, se vuelve a analizar y codificar de nuevo, para volver a realizar pruebas. Una vez terminado cada módulo principal y aprobado internamente que el sistema realiza correctamente lo estipulado en el documento de análisis, se pasa a la siguiente fase, pruebas de cliente solicitante.

6.2 PRUEBAS DE CLIENTE SOLICITANTE

En esta fase se realizaron juntas con el cliente que solicito el sistema web, para mostrar avances significativos, al igual que la prueba anterior, se realizan todas las acciones posibles en cada módulo, así el cliente tiene la oportunidad de analizar, comentar y detectar cada aspecto del módulo. Una vez terminado las pruebas con el cliente se pasa a la siguiente fase, prueba cliente piloto.

6.3 PRUEBAS DE CLIENTE PILOTO

Cuando los sistemas de software pasan por la fase de pruebas llega la etapa de verificación, es decir, que el usuario final usa el sistema simulando el flujo principal y definiendo los datos de entrada esperando los resultados deseados; el equipo de desarrollo de software debe estar seguro que el sistema no tenga errores. Es importante gestionar la infraestructura donde se va a montar el producto en una etapa de desarrollo, es decir, todo aquel software, archivos y

configuración con el que va a convivir el sistema de software para un funcionamiento correcto.

Para el sistema B2BOM se tienen unos requisitos mínimos a considerar para un funcionamiento óptimo que ayudan al usuario final tener una experiencia confiable para sus actividades diarias. Los requisitos mínimos a considerar para la instalación del sistema B2BOM son:

Sistema operativo.

Windows server	Linux	AIX
2003	Cualquier distribución de servidor	5.3
2008		6.1
2012		7.1

Tabla 3. Tipos de sistema operativo.

Compatibilidad de base de datos.

MySQL (Incluido)	Microsoft	Oracle	DB2
Community 5.6	SQL Server 2005	Versión 10g	9x
	SQL Server 2008	Versión 11g	Versión express
	SQL Server 2012	Versión express	
	Versión express		

Tabla 4. Versiones de base de datos.

Requerimientos del servidor de desarrollo.

Mínimo 2 GB de RAM
80 GB de disco duro
Procesador: si es físico: 1 procesador, si es virtual 2 procesadores.
Versión de java 1.7 o posterior.
Conexión a su servidor de correo: Exchange, Lotus, Gmail, etc.

Tabla 5. Características de servidor.

Para realizar esta fase de pruebas se utilizó la modalidad de cliente piloto, el cual está dispuesto detectar y mejorar aquellos puntos que están vulnerables en el proceso. Este eligió desde el catálogo de proveedores uno que apoye a validar todo el proceso de embarque en el software.

Algunas de las ventajas de realizar pruebas mediante un cliente piloto son:

- ✓ Valida cada proceso.
- ✓ Detecta errores.
- ✓ Comenta mejoras en los puntos débiles.
- ✓ Promueve el sistema.

Una vez seleccionado el cliente piloto, se realizó la instalación del producto en un ambiente de desarrollo que anteriormente se le solicitó con las características y requisitos mínimos para un funcionamiento correcto. Así mismo se estipuló una fecha en que las tres partes involucradas puedan trabajar y realizar las pruebas.

Como sucede con cualquier desarrollo, se encontraron errores en el proceso los cuales se solucionaron, en este caso fueron:

- 1) **Error:** El peso en el documento del XML del Consolidado con una ligera diferencia a la real.

Solución: En el peso del consolidado se detectó que el error estaba en la clase java de la conexión a la base de datos. Al obtener el resultado de la base de datos y empezar a llenar el objeto se utiliza la propiedad "getFloat()" para obtener el peso, ejemplo, el resultado real debería ser 23,987.567 pero la propiedad nos extraía 23,987.568, por lo que existe un ligero error y por esto se rechazaba el documento Consolidado. Se resolvió utilizando la propiedad "getString()" el cual si extraía el valor correcto, después se convirtió a la propiedad Float.

- 2) **Error:** Una etiqueta del XML mal colocada en el documento Booking.

Solución: El XML utiliza etiquetas en la estructura, por un error de programación se escribió una letra incorrectamente y esto rechazaba el documento Booking, se corrigió.

- 3) **Error:** Fallas en el demonio que escucha las carpetas para su procesamiento.

Solución: En el demonio o hilo que escucha una carpeta para leer archivos XML había un error al momento de que se ejecutara, por diferencias entre un servidor de aplicaciones binario y uno normal no se detectó que en uno se ejecutaba dos veces el hilo o demonio, por lo que se cambió y se corrigió descargando otra versión del servidor de aplicaciones.

- 4) **Error:** Falla en el envío de correo electrónico.

Solución: Para el funcionamiento de un envío de correo electrónico, este depende de la configuración, en nuestras pruebas utilizamos configuración básica, se solicitó y configuró la información del correo del proveedor, y al momento de las pruebas y envío de correo se detectó que en nuestras pruebas internas la autenticación se colocó de forma verdadera desde un inicio y en código duro, pero la configuración del cliente no necesitaba esa propiedad, por lo que se añadió a la configuración del sistema y se solucionó.

- 5) **Error:** Falla en la lógica de la agrupación de órdenes de compra al Booking según la fecha de entrega de cada orden de compra.

Solución: Una de las restricciones o reglas de negocio implementadas por el cliente es que las órdenes de compra deben estar agrupadas en un Booking; las mismas órdenes de compra no pueden estar en otro Booking, para liberar la orden de compra ligada a un Booking es necesario eliminarla del mismo o eliminar el Booking. El sistema B2BOM no validaba que la orden de compra estaba en un Booking previamente cuando dos o más usuarios estaban simultáneamente agregando la misma orden de compra a dos Bookings diferentes, por lo que antes de agregar las órdenes de compra se realiza un comparativo de que cada orden de compra debe estar libre para asociarla a un nuevo booking.

- 6) **Error:** Falla en la generación del XML del Booking que lleva una variable con valor null.

Solución: Cuando los archivos XML son creados a partir del sistema B2BOM para el intercambio de documentos no se aceptan valores null, es decir, si una variable del objeto que se entrega al método de generar XML lleva valor null, el archivo XML no se crea; se corrigió realizando una validación de cada variable que contenga un valor, si existe lo contrario envía un error de que el XML no se puede construir.

- 7) **Error:** Falla al iniciar el servidor de aplicaciones Apache Tomcat.

Solución: El servidor de aplicaciones Apache Tomcat inicia desde la aplicación instalada Java en el servidor o computadora donde se ejecuta la aplicación, la compatibilidad es importante; el sistema B2BOM está desarrollado con la versión Java 1.7, en el servidor del proveedor tenían la versión Java 1.6, se notificó al proveedor que cambiara la versión de Java a una más reciente.

- 8) **Error:** Agregar o eliminar órdenes de compra a un Booking previamente aceptado por el cliente.

Solución: Cuando el Booking es aceptado por el cliente no existe la posibilidad de eliminar o agregar órdenes de compra al Booking, por un detalle en la programación si existía la posibilidad de eliminar órdenes de compra al Booking. Se corrigió el detalle agregando una validación en la programación que ayudó a corregir este problema.

- 9) **Error:** Falla en el envío masivo de ASN y envío individual del ASN.

Solución: Si el usuario realizó 50 ASN que corresponden a 50 órdenes de compra, enviar una por una es un proceso largo y tedioso, por lo que se tiene un botón que genera los 50 XML de las 50 ASN construidas; cuando alguno de los ASN es rechazado por el cliente, el B2BOM tiene la posibilidad de trabajar las que están rechazadas y volver a enviarlo, este envío no se efectuaba correctamente, un error en dos variables en la programación que apoyan a la construcción del ASN fue corregido.

- 10) **Error:** Fallas en las páginas JSP enviando un error de ejecución.

Solución: Las paginas JSP también tienen errores y se encontraron algunos que al momento de una petición del usuario al sistema B2BOM arrojaba el error de ejecución. Esto sucedía por que el método que es llamado desde el JSP a las clases Java fue cambiado o alguna variable no contenía un valor, por lo que desde la página JSP se realizó una validación que las variables deberían contener un valor.

- 11) **Error:** Falla en el script de SQL Server.

Solución: Al obtener el respaldo de la base de datos de pruebas del ambiente del desarrollador y montarlo en el ambiente del proveedor no se cargó correctamente enviando una falla de versiones; desde el diseño del modelo de SQL Server se generó el script de las tablas y sus relaciones, desde la aplicación SQL Server se ejecutó el script y las tablas fueron creadas exitosamente.

CAPÍTULO 7. PUESTO EN PRODUCCIÓN

Al pasar la fase de pruebas internas y con los proveedores, las tres partes involucradas (cliente-proveedor-empresa desarrolladora) deben seleccionar una fecha de salida a producción, es decir, un uso real y comercial. Antes de la fecha de salida se deben realizar tareas importantes para que el ambiente general esté listo, es decir, se debe realizar una instalación en el servidor de producción que cumpla con las características del servidor de desarrollo, se debe realizar una capacitación a los usuarios finales, entregar el manual de usuario y configurar el sistema para su uso final.

Es importante tener en cuenta los siguientes puntos:

- ✓ Revisar el catálogo de artículos cliente-proveedor.
- ✓ Configurar el sistema (usuarios y canales).
- ✓ Colocar los archivos necesarios en el server.
- ✓ Realizar un corte de órdenes de compra actuales y las órdenes que no se han embarcado se deben de solicitar un reenvío al cliente con el nuevo formato (XML).
- ✓ Capacitación terminada a los usuarios.

Una vez instalado fue importante comprobar el funcionamiento del primer embarque, con información real y validar los compromisos de entrega de mercancía. Es por ello que se le promete al cliente apoyarlo en su primera entrega, es decir, realizar cada paso en el sistema hasta el final del flujo del proceso. Si todo salió bien y sin contratiempos se da por terminado la instalación del producto y el proveedor pasa a una instancia de soporte.

7.1 SOPORTE

Todos los sistemas que hay en el mercado y con licencia oficial tienen por obligación un soporte desde la compra hasta la fecha establecida por el vendedor. Un soporte es donde el usuario detecta algo extraño en el proceso, o no ha recibido un archivo, no se ha enviado un correo, etcétera, algo que el usuario no puede resolver y necesita ayuda; aquí entra el apoyo del área de soporte del vendedor en donde ayuda al cliente a resolver el incidente. Existen varios medios para realizar un soporte, correo electrónico, chat, software de aplicación y la más común vía telefónica.

Al momento de la compra del sistema B2BOM el punto del soporte es importante detallarlo en el contrato del producto y que alcance va a tener; para

esto se le ofrecen distintos tipos de soporte al cliente con sus características y al final él decide el que más le conviene; y como regla interna de la compañía solo se da asistencia técnica al producto adquirido sin intervenir en otras aplicaciones externas aunque formen parte de un proceso.

Se debe tener en cuenta que en el flujo de intercambio electrónico existen otras aplicaciones que conviven en el proceso de embarque de la mercancía, el B2BOM, AS2 del proveedor y AS2 del cliente; por lo que si uno de estos tres falla, entonces el proceso no se cumple y se solicita soporte indicando la incidencia del mismo; de estas tres herramientas dos están de lado del proveedor (B2BOM y AS2 proveedor) y la empresa cuenta con uno que es seguro el B2BOM siendo el AS2 una opción de compra. Por lo que el soporte no siempre es del lado del B2BOM, sino de las otras aplicaciones que conviven en el flujo del proceso y se atenderá a cada solicitud dependiendo del software adquirido.

Modelos de soporte.

Remoto. Se realiza mediante una aplicación de software el acceso al servidor donde se requiere el soporte, en la mayoría de los casos donde está instalado el producto.

Asistencia telefónica. El más común que se utiliza, el cliente realiza una llamada al proveedor para reportar el incidente y recibir la asistencia.

Presencial.- Se envía un analista a las instalaciones donde se requiere el soporte para revisar y resolver la incidencia.

Niveles de soporte.

Cuando el cliente levanta un soporte especificando la incidencia se escala a un nivel básico, donde los analistas verifican el detalle levantado y concluyen si lo pueden resolver o escalarlo a un segundo nivel y así sucesivamente hasta llegar al punto de no poder resolver el detalle. La compañía cuenta con tres niveles de soporte.

Nivel 1.- La incidencia levantada es más una duda o falta de conocimiento del proceso del producto, acceso al sistema, etcétera. Este nivel no requiere acceso al servidor del cliente.

Nivel 2.- La incidencia levantada es un problema en el flujo del proceso, sistema caído, sin acceso al sistema, no se ven alguna información, lentitud del software, etcétera. Este nivel si es necesario el acceso al servidor del cliente.

Nivel 3.- La incidencia levantada requiere la intervención de los desarrolladores para revisar el detalle, cuando se llega a este nivel es porque el nivel 2 no puede resolverlo y el problema es en algunos casos incorregible. Se analiza si el detalle es una incidencia interna (B2BOM) o externa (aplicaciones externas al producto comprado).

Interna. Un detalle en el código o flujo del proceso principal.

Externa. Daños en el disco duro, sistema operativo, etcétera.

Tipos de soporte.

- Soporte 8x5 remoto.- Incluye soporte remoto y/o telefónico los cinco días laborales de la semana (lunes a viernes) con horario de oficina.
- Soporte 24x7 remoto.- Incluye soporte remoto y/o telefónico los siete días de la semana y las 24 horas del día.
- Soporte por evento remoto.- Cada llamada o incidencia levantada genera horas de inversión para resolver el detalle, se define un costo por hora desde el principio.
- Soporte 8x5 en sitio.- Incluye el envío de un analista en las instalaciones donde se requiere el soporte para resolver el incidente.
- Soporte por evento en sitio.- Se envía un analista a las instalaciones donde se requiere el soporte, se define un costo por hora desde el principio.

Soportes más frecuentes.

Órdenes no se ven en B2BOM.

Pueden ser por varias razones.

- *Nivel de soporte 1.*

El cliente no ha enviado la orden de compra por el canal de AS2.

El portal de proveedores es un recurso del proveedor para consultar las órdenes de compra creadas por el cliente, esto no quiere decir que la orden de compra ya se encuentre en el sistema B2BOM, tiene que esperar un lote de envío que se ejecuta tres veces al día. Se le notifica al usuario que debe esperar para el envío de la orden de compra o solicitar un reenvío de la orden de compra a su cliente.

- *Nivel de soporte 2.*

Documentos de intercambio electrónico no están en línea o va dirigida a otro MDC.

Las órdenes de compra suelen tener modificaciones, esto sucede porque el cliente detecta que cometió un error y la modifico, por lo que puede no ser

encontrada en el sistema B2BOM y se movió a otro MDC o cambio de estado. Se le notifica al usuario como localizar la orden de compra por medio del portal B2BOM.

No se genera archivo ERP.

- *Nivel de soporte 2.*

Ruta de envío del documento ERP inválido o sin permisos de escritura.

La mayoría de los proveedores tienen estructurados sus servidores, uno pertenece al sistema interno, otro al B2BOM, al AS2 y a la facturación electrónica; por lo que es importante puntualizar las rutas utilizadas para el intercambio electrónico entre los sistemas; los proveedores pueden cambiar las rutas a su conveniencia pero estas están sin permiso de escritura, lo cual sucede principalmente en servidores basados en Unix. Se le comenta al usuario el cambio de la ruta y se le notifica el detalle del porqué no escribe el archivo ERP.

Booking rechazado.

- *Nivel de soporte 2.*

Validaciones en el sistema B2BOM desactivados y el cliente rechazó el documento por no cumplir con la regla de negocio.

En el sistema B2BOM el proveedor puede modificar las validaciones para realizar algunas tareas antes de tiempo, como por ejemplo, deshabilitar el tiempo de generación de Booking y así adelantar hacer el Booking, al ser enviado el cliente envía un error ya que no cumple con la validación previa. Se le notifica al usuario su error y se le explica lo que más conviene.

- *Nivel de soporte 2.*

Usuario envió el Booking en más de una ocasión.

Existe un bug aún no solucionado por parte del cliente que consiste en que cada documento enviado (Booking, ASN y Consolidado) el sistema B2BOM espera una respuesta; esta respuesta no es enviada pero si registrada en el cliente, por lo que si el documento se reenvía de nuevo el cliente regresa un error. Se le notifica al usuario el detalle, se realiza una llamada al cliente para que reenvíe de regreso la respuesta en forma correcta.

- *Nivel de soporte 1.*

Fechas de entrega de cada orden de compra no cumplen con las regla de negocio.

Al igual que el primer punto, el usuario deshabilita la validación de la fecha estimada de entrega por cada orden de compra, por lo que sí es enviado en un Booking este marca un error, que la orden de compra no puede ser embarcada, porque no cumple con la fecha correcta; se le notifica al usuario que debe esperar para reenviar el documento Booking.

ASN ya existe.

- *Nivel de soporte 2.*

El usuario envió el ASN en más de una ocasión.

El bug mencionado anteriormente puede suceder en cualquier punto del proceso de envío y recepción de documentos. El usuario envió en más de una ocasión el documento ASN y el cliente responde que ya existe el ASN de la orden de compra; se le notifica al usuario el detalle y se cambia el estado del ASN para que continúe con el proceso.

Usuario no accede al sistema.

- *Nivel de soporte 2.*

Servidor de aplicaciones está detenido.

Al reiniciar el servidor puede causar que algunas aplicaciones no inicien; en este caso el servidor de aplicaciones no inició, se le notifica al usuario y departamento de sistemas del proveedor para revisar la incidencia y se le explica el detalle.

- *Nivel de soporte 2.*

Usuario está inactivo en el sistema.

Puede ser por dos causas, una porque el administrador de la aplicación lo desactivó desde el portal B2BOM o se perdió la conexión a la base de datos, este último sucede con la base de datos instalada en otro servidor, por lo que no puede existir conexión entre la aplicación y sistema Web. Se le notifica al usuario el detalle junto con el departamento de sistemas del proveedor para resolver el incidente.

- *Nivel de soporte 2.*

Base de datos detenido.

Al igual que el punto uno, el servicio de la base de datos esta caído, por lo que es importante iniciarlo. Esta parte es notificada al usuario.

No se recibe estatus de los documentos de intercambio electrónico.

- *Nivel de soporte 2.*
Los sistemas de AS2 (cliente-proveedor) tienen problemas de comunicación.
Cada sistema depende de la administración del cliente o proveedor, por lo que descuidar el mantenimiento del mismo puede provocar esta incidencia; Esto sucede con frecuencia, (en automático cambio la IP pública del servidor y/o el sistema AS2 está detenido).
- *Nivel de soporte 2.*
Certificados vencidos por el AS2 del proveedor.
Los certificados tienen una vigencia tanto para el cliente y proveedor, por lo que hay que tener en cuenta este detalle es vital para no perder comunicación. Se le apoya al usuario y se crea un certificado nuevo notificándole al cliente para su aprobación.
- *Nivel de soporte 2.*
El servidor no tiene internet.
Esto sucede cuando la red está abajo y el sistema B2BOM al ser una aplicación Web, no se puede acceder. Se le notifica al usuario y se revisa la incidencia con el departamento de sistemas.

Orden de compra cancelada.

- *Nivel de soporte 2.*
El cliente canceló una orden de compra pero el proveedor no fue notificado de dicha cancelación.
Las órdenes de compra pueden ser canceladas por el cliente, pero sin notificarle al proveedor, por lo que esta orden de compra sigue vigente en el sistema B2BOM y es utilizada en el proceso de embarque; se le notifica al usuario que realice una conferencia con las tres partes involucradas (cliente-proveedor-empresa de soporte) para revisar la incidencia y resolverla.

CAPÍTULO 8. CONCLUSIONES

8.1 CONCLUSIÓN

Los embarques son complejos en la forma de la administración de la misma, aunque la lógica es la misma cada proveedor tiene, según considera que sea, la forma más eficiente de embarcar o la que se ajusta según sus necesidades; al cliente no le interesa como el proveedor embarca, solo necesita la mercancía en su bodega y él le dará el tratamiento necesario para distribuirla a las tiendas. El sistema B2B está hecho a la medida del cliente y no del proveedor (el que embarca), es por esa lógica que todos deben ajustarse a ese proceso usen o no el sistema B2B.

El objetivo principal fue la de generar un sistema de software con los cambios que el cliente está realizando; se realizó el B2BOM con las especificaciones del cliente con las reglas de negocio y se cumplió al generar un sistema privado, de acceso inmediato y con los documentos imprimibles especificados para cada embarque. Así mismo se cumplieron los objetivos específicos:

- **Analizar y comprender el nuevo proceso.**
Cada junta diaria, semanal y cada llamada telefónica sirvieron para entender más a fondo el problema, comprender el objetivo principal y las reglas de negocio que contendría el sistema.
- **Analizar la integración de las órdenes de compra con el sistema administrativo del proveedor.**
Se cumplió al entender que cada cliente tiene una forma diferente de integrar las órdenes de compra al sistema interno, es por eso que utilizar clases abstractas, nos ayudaron a resolver cada integración según sea el cliente que esté utilizando el sistema.
- **Estructurar los procesos.**
Cada proceso de embarque es significativo, si uno no es aprobado no se puede avanzar al siguiente, es por eso que se separaron los 3 procesos principales identificados desde el B2BOM.
- **Diseñar una base de datos.**
Se cumple con un diseño funcional y estructurado que se ajusta solo para un portal interno y no multi proveedores.

- **Realizar las restricciones correspondientes en cada proceso.**
Se aplicaron exitosamente las reglas de negocio expuestas por el cliente para cada proceso, aunque el cliente cumple con las reglas desde sus sistemas internos cuando los documentos son enviados, el B2B logra cumplir estas exigencias mucho antes de que salgan de él.
- **Analizar la información relevante a mostrar en pantalla al usuario final.**
Se analizó internamente que cada ventana mostrara información relevante de cada proceso y fueron de la satisfacción del usuario final.
- **Realizar los PDF correspondientes para su impresión.**
Los PDF fueron exitosamente validados y diseñados de forma cuidadosa para cumplir con la exigencia del cliente.
- **Analizar cada documento creado para su envío.**
Las estructuras de cada documento en formato XML fueron detenidamente revisados y se llegó a la conclusión de utilizar XSD para validar que el archivo tenga la estructura correcta y solo queda para la revisión si existiera algún error en la información.
- **Parametrizar el sistema según convenga al proveedor.**
El sistema es 100% parametrizable, sin la necesidad de modificar código, el B2BOM se ajusta a las necesidades del proveedor y del usuario, modificando y/o eliminado parámetros para una mejor experiencia.

Cada proyecto lleva tiempo de análisis, desarrollo y pruebas hasta la salida a producción, en medio de estos procesos llegan a surgir ideas que puedan mejorar un módulo o el sistema como tal y así llevar a una mejor experiencia al usuario, pero en este tipo de proyectos existen fechas compromiso que hacen que cada día valga la pena el avance de desarrollo. Este sistema en su nombre oficial lleva el nombre de “B2BOM” Business To Business Object Manager 3.0.

Es importante versionar los sistemas desarrollados para que la empresa, cliente, proveedores y sobre todo el desarrollador identifique el sistema y sus cualidades; Teniendo en cuenta que utilizar la ingeniería de software así como tener organizado el sistema B2BOM nos da la oportunidad de seguir mejorando el sistema según sea la conveniencia o deseo de la empresa o cliente.

8.2 TRABAJOS FUTUROS

En varios trabajos, proyectos o investigaciones, la continuidad es importante para darle un seguimiento a un tema en específico; para nuestro caso el sistema B2B cumple con los requerimientos solicitados por el cliente desde un principio y dejando abierta la posibilidad de mejorar algunos procesos o módulos que contiene el sistema; a continuación se describen algunos puntos que pueden ser implementados para futuras versiones del sistema B2B:

- **Automatización del embarque.** Se pueden realizar los tres procesos de embarque (Booking, ASN y Consolidado) de forma automática sin la intervención del usuario final; las órdenes de compra están guardadas en el sistema interno del proveedor por lo que tiene la capacidad de generar un archivo estructurado con la información del embarque y que el sistema B2B lo capture, traduzca, valide y envíe los documentos de intercambio electrónico con el cliente de forma automática.
- **Sistema B2B en la nube.** La posibilidad de que el sistema sea instalado en la nube y desde ahí los proveedores puedan realizar los embarques desde un portal externo y de acceso las 24 horas del día; la ventaja que se obtendría es que los proveedores no le darían mantenimiento al servidor, base de datos o la infraestructura de la misma. La base de datos no es multiproveedores, por lo que se debe de realizar un ajuste a la misma y modificar todas las consultas a la base; también revisar cómo integrar la información de las órdenes de compra al sistema interno del proveedor.
- **Separar hilos.** Se puede separar en tres hilos el proceso de recepción de órdenes de compra, uno que corresponda a la traducción, validación y guardado en la base de datos, otro que genere el archivo para el sistema interno ERP y por último el envío de correo electrónico a los usuarios registrados; estos procesos están en un solo hilo de ejecución.

9. REFERENCIAS

- [1] Cursos de Posgrados de Supply Chain Management y Logística. Intercambio Electrónico de Datos (Electronic Data Interchange) en formato X12 (EDI X12 850 y 860) para órdenes de compra.
- [2] EDI Standards.
<http://ecomgx17.ecomtoday.com/edi/>
Internet (15/01/2015).
- [3] Universidad de Alicante. Dept. Ciencia de la computación en Inteligencia Artificial. Introducción a Struts2.
<http://expertojava.ua.es/j2ee/publico/struts-2010-11/sesion04-struts-apuntes.html>
26/06/2014
- [4] IBM WebSphere Partner Gateway - Express.
<http://www-01.ibm.com/software/integration/wspartnergateway/express/>
Internet 16/01/2015.
- [5] MYSQL, The world's most popular open source database.
<http://www.mysql.com/>
Internet.
- [6] Allamaraju, S., Beust, C., Davies, J. (Ed.) (2002). Programación Java Server con J2EE Edición 1.3, España: Editorial Anaya Multimedia.
- [7] Apache tomcat. Tomcat, Apache tomcat, Jakarta Tomcat.
<http://tomcat.apache.org/>
The Apache Software Foundation
- [8] Silvain Petit (09/06/2014 11:10) Foro de Taulia. ASN ¿Qué es un Aviso previo al embarque (ASN)?.
<https://support.taulia.com/hc/es/articles/202742659--Qu%C3%A9-es-un-Aviso-previo-al-embarque-ASN-FAQ-S198-> Sylvain Petit (09/06/2014 11:10).
- [9] Servidor de aplicaciones. @ 2016 Microsoft.
[https://technet.microsoft.com/es-mx/library/cc754024\(v=ws.10\).aspx](https://technet.microsoft.com/es-mx/library/cc754024(v=ws.10).aspx)

- [10] El Lenguaje Unificado de Modelado. Manual de referencia.
James Rumbaugh, Ivar Jacobson y Grady Booch.
Edición Addison Wesley

ANEXO 10.2 PDF PACKING LIST

Archivo en formato PDF que genera el sistema B2B de cada embarque, es decir, de cada consolidado.

	Packing List
---	-----------------------

Número de ASN: 0005530468
Fecha envío: 08-oct-2015

Fecha estimada de llegada: 23-oct-2015

Enviar por: USE HD GUIA DE EMBARQUE
Enviar a: DAL TILE MEXICO S DE RL DE CV
Folio de envío: 0005530468

Back Order: NO
Tipo: PREPAID

Contacto proveedor: Administrador
Telefono contacto: 81818181818
Email contacto: jeburto@azurlan.com

Proveedor 509170 DAL TILE MEXICO S DE RL DE CV

Orden de compra	Tienda	Artículo	Modelo	Descripción	UPC	Licencia Termino	Cantidad	Auditoria
8061071271	8061 MDC MTY	809741	ZDC1720HD1P2	ZDC1720HD1P2 PISO VINTAGE BROWN 18X	7-33057-54683-1	000553046801	60	☐ ☐ ☐
		211580	ZDM7720HD1P2	ZDM7720HD1P2 PISO WOODLAND RED OAK	0733057541942	000553046802	60	☐ ☐ ☐
		809738	ZDM0720HD1P2	ZDM0720HD1P2 PISO HERITAGE BLEND 18	0733057546883	000553046803	60	☐ ☐ ☐
8061071278	8061 MDC MTY	211580	ZDM7720HD1P2	ZDM7720HD1P2 PISO WOODLAND RED OAK	0733057541942	000553047501	200	☐ ☐ ☐
		211583	ZDM8720HD1P2	ZDM8720HD1P2 PISO HERITAGE COFFEE 1	7-33057-54195-8	000553047501	780	☐ ☐ ☐
		211580	ZDM7720HD1P2	ZDM7720HD1P2 PISO WOODLAND RED OAK	0733057541942	000553047502	520	☐ ☐ ☐

COMENTARIOS:

--

AUDITORIA

NOMBRE COMPLETO: _____

FECHA: _____

HORA: _____

FIRMA: _____

ANEXO 10.3 PDF ETIQUETA

Archivo en formato PDF que genera el B2B por cada tarima de cada orden de compra en el ASN.

	ASN	 0 0 0 6 3 5 2 3 8 3 8823 GUADALAJARA
8823 GUADALAJARA		
Orden de Compra		
6799-123662		 6 7 9 9 1 2 3 6 6 2
Proveedor		
100352	PROVEEDOR SA DE CV	
Licencia tarima	Tarima	Bultos
 M000635238301	1/2	1188
	Peso(kg)	Fecha
	23760.000	11-DIC-2016
	Vol. (m3)	
	12.224520	

ANEXO 10.4 XML DE LA ORDEN DE COMPRA

Archivo de la orden de compra enviada por el cliente al proveedor.

```
<?xml version="1.0" encoding="utf-8"?><PurchaseOrder>
  <Sender>ZZ-homedepotmxt</Sender>
  <PO>
    <PONumber>6799123661</PONumber>
    <ASN>0006352383</ASN>
    <RevNo>0</RevNo>
    <Date>11/16/16</Date>
    <DateRev>11/16/16</DateRev>
    <DateDeliv>11/28/16</DateDeliv>
    <ShipVia />
    <Lab>ORIGIN</Lab>
    <BackOrders>CANCEL</BackOrders>
    <FreightTerms>COLLECT</FreightTerms>
    <InvoiceTerms>NET 60 DAYS </InvoiceTerms>
    <TotalCube>13.34</TotalCube>
    <TotalWeight>25920</TotalWeight>
    <Vendor>
      <VendorId>100352-03</VendorId>
      <VendorName>PROVEEDOR SA DE CV</VendorName>
      <VendorAddress>RM25. CARR. HUICHAPAN-QUERETARO</VendorAddress>
      <VendorCity>HUICHAPAN</VendorCity>
      <VendorCountry>HGO</VendorCountry>
      <VendorZipCode>42410</VendorZipCode>
    </Vendor>
    <StoreReceives>8799</StoreReceives>
    <ShipTo>
      <ShipToId>8823</ShipToId>
      <ShipToNamePlace>GUADALAJARA </ShipToNamePlace>
      <ShipToAddress>PARQUE INDUSTRIAL </ShipToAddress>
      <ShipToColony>AUTOPISTA JOROBAS TULA KM 23.5 </ShipToColony>
      <ShipToCity>GUADALAJARA</ShipToCity>
      <ShipToState>JL</ShipToState>
      <ShipToZip>42970</ShipToZip>
      <ShipToTelephone />
    </ShipTo>
    <Instructions>
      <Instruction Id="000" Type="Special" />
      <Instruction Id="000" Type="Special" />
      <Instruction Id="000" Type="Shipping" />
    </Instructions>
    <Items>
      <Item>
        <Id>115924</Id>
        <Quantity>1188</Quantity>
        <UnitofMeasure>EA</UnitofMeasure>
        <modl>AB.ADHE.GRIS.20KG.RAPIS54</modl>
        <upc>0769695000039</upc>
        <Descr>AB.ADHE.GRIS.20KG.RAPIS54 ADHESIVO RAPIDO GRIS 20 </Descr>
        <UnitCost>50.65</UnitCost>
        <Weight>20</Weight>
        <Cube>.01029</Cube>
      </Item>
    </Items>
    <TotalQty>1296</TotalQty>
    <SubTotal>68312.16</SubTotal>
    <Iva>10929.95</Iva>
    <Total Currency="PESOS">79242.11</Total>
    <InvoiceTo>
      <InvoiceName>Cliente S. de R.L.</InvoiceName>
      <InvoiceAddress1>Av. Ricardo Margain No. 605</InvoiceAddress1>
      <InvoiceAddress2>Col. Santa Pedro</InvoiceAddress2>
      <InvoiceAddress3>San Pedro Garza Garcia , Nuevo Leon C.P.66200 Mexico</InvoiceAddress3>
      <InvoiceTelephone>(8) 155-99-99</InvoiceTelephone>
      <InvoiceFax>(8) 155-99-99</InvoiceFax>
      <InvoiceRfc>HDM-562485-DE3</InvoiceRfc>
    </InvoiceTo>
    <Status>ACTIVE</Status>
  </PO>
</PurchaseOrder>
```

ANEXO 10.5 XSD DE LA ORDEN DE COMPRA

Archivo XSD para validar la estructura del XML de la orden de compra.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PurchaseOrder">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="Sender"/>
        <xs:element name="PO">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:long" name="PONumber"/>
              <xs:element type="xs:string" name="ASN"/>
              <xs:element type="xs:byte" name="RevNo"/>
              <xs:element type="xs:string" name="Date"/>
              <xs:element type="xs:string" name="DateRev"/>
              <xs:element type="xs:string" name="DateDeliv"/>
              <xs:element type="xs:string" name="ShipVia"/>
              <xs:element type="xs:string" name="Lab"/>
              <xs:element type="xs:string" name="BackOrders"/>
              <xs:element type="xs:string" name="FreigthTerms"/>
              <xs:element type="xs:string" name="InvoiceTerms"/>
              <xs:element type="xs:float" name="TotalCube"/>
              <xs:element type="xs:float" name="TotalWeight"/>
              <xs:element name="Vendor">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="VendorId"/>
                    <xs:element type="xs:string" name="VendorName"/>
                    <xs:element type="xs:string" name="VendorAddress"/>
                    <xs:element type="xs:string" name="VendorCity"/>
                    <xs:element type="xs:string" name="VendorCountry"/>
                    <xs:element type="xs:int" name="VendorZipCode"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:int" name="StoreReceives"/>
              <xs:element name="ShipTo">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:int" name="ShipToId"/>
                    <xs:element type="xs:string" name="ShipToNamePlace"/>
                    <xs:element type="xs:string" name="ShipToAddress"/>
                    <xs:element type="xs:string" name="ShipToColony"/>
                    <xs:element type="xs:string" name="ShipToCity"/>
                    <xs:element type="xs:string" name="ShipToState"/>
                    <xs:element type="xs:int" name="ShipToZip"/>
                    <xs:element type="xs:string" name="ShipToTelephone"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="Instructions">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Instruction" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:string">
                <xs:attribute type="xs:string" name="Id" use="optional"/>
                <xs:attribute type="xs:string" name="Type" use="optional"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Items" maxOccurs="unbounded" minOccurs="1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Item">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:int" name="Id"/>
              <xs:element type="xs:float" name="Quantity"/>
              <xs:element type="xs:string" name="UnitofMeasure"/>
              <xs:element type="xs:string" name="modl"/>
              <xs:element type="xs:long" name="upc"/>
              <xs:element type="xs:string" name="Descr"/>
              <xs:element type="xs:float" name="UnitCost"/>
              <xs:element type="xs:float" name="Weight"/>
              <xs:element type="xs:float" name="Cube"/>
              <xs:element type="xs:string" name="Tab"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element type="xs:float" name="TotalQnty"/>
  <xs:element type="xs:float" name="SubTotal"/>
  <xs:element type="xs:float" name="Iva"/>
  <xs:element name="Total">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:float">
          <xs:attribute type="xs:string" name="Currency"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="InvoiceTo">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="InvoiceName"/>
        <xs:element type="xs:string" name="InvoiceAddress1"/>
        <xs:element type="xs:string" name="InvoiceAddress2"/>
        <xs:element type="xs:string" name="InvoiceAddress3"/>
        <xs:element type="xs:string" name="InvoiceTelephone"/>
        <xs:element type="xs:string" name="InvoiceFax"/>
        <xs:element type="xs:string" name="InvoiceRfc"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element type="xs:string" name="Status"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

ANEXO 10.6 XML DE ACUSE

Archivo XML que se crea y envía al cliente por cada orden de compra recibida.

```
<POAck>
  <Sender>azurian1</Sender>
  <POAcknow>
    <TSetPurposeCode>00</TSetPurposeCode>
    <AcknowledgmentType>AK</AcknowledgmentType>
    <Receives>ZZ-homedepotmxt</Receives>
    <PurchaseOrderNum>6082102947</PurchaseOrderNum>
    <PODate>2016-02-16 16:49:49</PODate>
    <AcknowledgmentDate>2016-02-16 16:49:49</AcknowledgmentDate>
  </POAcknow>
</POAck>
```

ANEXO 10.7 XML DEL BOOKING

Archivo XML que se crea y se envía al cliente sobre el embarque.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Booking>
  <ApplicationArea>
    <Sender>proveedor1</Sender>
  </ApplicationArea>
  <BookingLevel>
    <Vendor>
      <VendorId>100352</VendorId>
      <Contact>PROVEEDOR SA DE CV</Contact>
      <CPhone>81818181818</CPhone>
      <CMail>soportemexico@azurian.com</CMail>
    </Vendor>
    <BookingHeader>
      <CreateDate>12/09/16 15:12</CreateDate>
      <ExpectedDate>12/10/16 15:12</ExpectedDate>
      <ArrivalDate>12/24/16 15:12</ArrivalDate>
      <DeliveryTo>8799</DeliveryTo>
      <FreightTerm>COLLECT</FreightTerm>
      <CarrierType>FTL</CarrierType>
      <EquipType>48</EquipType>
      <TotalPallets>0</TotalPallets>
      <LoadType>Tarima</LoadType>
      <Comments/>
      <PickUpNum>12</PickUpNum>
    </BookingHeader>
    <BookingDetail>
      <OrderLevel>
        <PoNumber>6799123662</PoNumber>
        <ItemLevel>
          <Sku>115924</Sku>
          <Modl>AB.ADHE.GRIS.20KG.RAPIS54</Modl>
          <Quantity>1188</Quantity>
          <UnitofMeasure>EA</UnitofMeasure>
          <CountryOrigin/>
          <PckType>BDL</PckType>
          <Bulks>1188</Bulks>
          <Weight>20.0</Weight>
          <Cube>0.01029</Cube>
        </ItemLevel>
        <ItemLevel>
          <Sku>115926</Sku>
          <Modl>AB.ADHE.GRIS.20KG.PORCS54</Modl>
          <Quantity>108</Quantity>
          <UnitofMeasure>EA</UnitofMeasure>
          <CountryOrigin/>
          <PckType>BDL</PckType>
          <Bulks>108</Bulks>
          <Weight>20.0</Weight>
          <Cube>0.01029</Cube>
        </ItemLevel>
      </OrderLevel>
    </BookingDetail>
  </BookingLevel>
</Booking>
```

ANEXO 10.8 XSD DEL BOOKING

Archivo XSD que valida la estructura del XML que genera el B2B para su envío.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Booking">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="BookingLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Vendor">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="CreateDate"/>
                    <xs:element type="xs:string" name="ExpectedDate"/>
                    <xs:element type="xs:string" name="ArrivalDate"/>
                    <xs:element type="xs:int" name="DeliveryTo"/>
                    <xs:element type="xs:string" name="FreightTerm"/>
                    <xs:element type="xs:string" name="CarrierType"/>
                    <xs:element type="xs:string" name="EquipType"/>
                    <xs:element type="xs:string" name="TotalPallets"/>
                    <xs:element type="xs:string" name="LoadType"/>
                    <xs:element type="xs:string" name="Comments"/>
                    <xs:element type="xs:string" name="PickUpNum"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="BookingDetail">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="OrderLevel" maxOccurs="unbounded" minOccurs="1">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="PoNumber"/>
                          <xs:element name="ItemLevel" maxOccurs="unbounded" minOccurs="1">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:int" name="Sku"/>
                                <xs:element type="xs:string" name="Mod1"/>
                                <xs:element type="xs:int" name="Quantity"/>
                                <xs:element type="xs:string" name="UnitofMeasure"/>
                                <xs:element type="xs:string" name="CountryOrigin"/>
                                <xs:element type="xs:string" name="PckType"/>
                                <xs:element type="xs:int" name="Bulks"/>
                                <xs:element type="xs:int" name="Bulks"/>
                                <xs:element type="xs:float" name="Weight"/>
                                <xs:element type="xs:float" name="Cube"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.9 XML DE ESTATUS DEL BOOKING

Archivo en formato XML que envía el Cliente por respuesta a cada Booking enviado.

```
<?xml version="1.0" encoding="utf-8"?><StatusBooking>
  <ApplicationArea>
    <Sender>ZZ-cliente</Sender>
  </ApplicationArea>
  <DocLevel>
    <Vendor>505400</Vendor>
    <Num>0000090077</Num>
    <Status>A</Status>
    <DescError />
    <Detail>
      <PO>6082080939</PO>
      <PO>7701000009</PO>
    </Detail>
    <Date>04/23/14</Date>
  </DocLevel>
</StatusBooking>
```

ANEXO 10.10 XSD DEL ESTATUS DEL BOOKING

Archivo XSD que valida la estructura del Estatus del Booking enviado por el Cliente.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="StatusBooking">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DocLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:int" name="Vendor"/>
              <xs:element type="xs:string" name="Num"/>
              <xs:element type="xs:string" name="Status"/>
              <xs:element type="xs:string" name="DescError"/>
              <xs:element name="Detail">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:long" name="PO" maxOccurs="unbounded" minOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="Date"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.11 XML DEL ASN

Archivo en formato xml que genera el B2B del ASN de cada orden de compra y que se envía al Cliente.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ASN>
  <ApplicationArea>
    <Sender>proveedor1</Sender>
  </ApplicationArea>
  <ShipmentLevel>
    <ShipNumber>0006352383</ShipNumber>
    <CreationDate>2016-12-09 15:23:25</CreationDate>
    <Trailers>0000</Trailers>
    <ShippedDate>2016-12-10 15:12:45</ShippedDate>
    <EstArriveDate>2016-12-24 15:12:47</EstArriveDate>
    <PortName/>
    <CityOrigin/>
    <FreightTerm/>
    <ServiceType>FTL</ServiceType>
    <ForeignCarrier>PROFIO</ForeignCarrier>
    <PalletsLTL/>
    <PalletsFB/>
    <PalletsFTL>2</PalletsFTL>
    <ContainerSize>48</ContainerSize>
    <TrailerNumber/>
    <BLno>72752</BLno>
    <SealNumber/>
    <VesselName/>
    <VoyageNumber/>
    <DeliveryReceipt/>
    <StockNo/>
    <GrossWeight/>
    <NotifyParty/>
    <Comments>Tarima</Comments>
    <Bulks/>
    <Cube/>
    <Weight/>
    <OrderLevel>
      <PONumber>6799128662</PONumber>
      <Store>8828</Store>
      <ASNNumber>0006352383</ASNNumber>
      <PalletLevel>
        <Id>000635238301</Id>
        <Bulks>1188</Bulks>
        <ItemLevel>
          <Sku>115924</Sku>
          <Quantity>1188</Quantity>
          <Commodity/>
          <PckType/>
          <Cube/>
          <Weight/>
        </ItemLevel>
      </PalletLevel>
      <PalletLevel>
        <Id>000635238302</Id>
        <Bulks>108</Bulks>
        <ItemLevel>
          <Sku>115926</Sku>
          <Quantity>108</Quantity>
          <Commodity/>
          <PckType/>
          <Cube/>
          <Weight/>
        </ItemLevel>
      </PalletLevel>
    </OrderLevel>
  </ShipmentLevel>
</ASN>
```

ANEXO 10.12 XSD DEL ASN

Archivo XSD que valida el XML del ASN creado por cada orden de compra para su envío.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ASN">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ShipmentLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="ShipNumber"/>
              <xs:element type="xs:string" name="CreationDate"/>
              <xs:element type="xs:string" name="Trailers"/>
              <xs:element type="xs:string" name="ShippedDate"/>
              <xs:element type="xs:string" name="EstArriveDate"/>
              <xs:element type="xs:string" name="PortName"/>
              <xs:element type="xs:string" name="CityOrigin"/>
              <xs:element type="xs:string" name="FreightTerm"/>
              <xs:element type="xs:string" name="ServiceType"/>
              <xs:element type="xs:string" name="ForeignCarrier"/>
              <xs:element type="xs:string" name="PalletsLTL"/>
              <xs:element type="xs:string" name="PalletsFB"/>
              <xs:element type="xs:string" name="PalletsFTL"/>
              <xs:element type="xs:string" name="ContainersSize"/>
              <xs:element type="xs:string" name="TrailerNumber"/>
              <xs:element type="xs:string" name="BLno"/>
              <xs:element type="xs:string" name="SealNumber"/>
              <xs:element type="xs:string" name="VesselName"/>
              <xs:element type="xs:string" name="VoyageNumber"/>
              <xs:element type="xs:string" name="DeliveryReceipt"/>
              <xs:element type="xs:string" name="StockNo"/>
              <xs:element type="xs:string" name="GrossWeight"/>
              <xs:element type="xs:string" name="NotifyParty"/>
              <xs:element type="xs:string" name="Comments"/>
              <xs:element type="xs:string" name="Bulks"/>
              <xs:element type="xs:string" name="Cube"/>
              <xs:element type="xs:string" name="Weight"/>
              <xs:element name="OrderLevel">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:long" name="PONumber"/>
                    <xs:element type="xs:int" name="Store"/>
                    <xs:element type="xs:string" name="ASNNumber"/>
                    <xs:element name="PalletLevel" maxOccurs="unbounded" minOccurs="1">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="Id"/>
                          <xs:element type="xs:int" name="Bulks"/>
                          <xs:element name="ItemLevel" maxOccurs="unbounded" minOccurs="1">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:int" name="Sku"/>
                                <xs:element type="xs:int" name="Quantity"/>
                                <xs:element type="xs:string" name="Commodity"/>
                                <xs:element type="xs:string" name="PckType"/>
                                <xs:element type="xs:string" name="Cube"/>
                                <xs:element type="xs:string" name="Weight"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.13 XML DEL ESTATUS DEL ASN

Archivo en formato XML que el Cliente envía por cada XML enviado.

```
<?xml version="1.0" encoding="utf-8"?><StatusASN>
  <ApplicationArea>
    <Sender>ZZ-homedepotwxt</Sender>
  </ApplicationArea>
  <DocLevel>
    <Vendor>505400</Vendor>
    <Num>0004753983</Num>
    <Status>A</Status>
    <DescError></DescError>
    <Detail>
      <PO>7701000009</PO>
    </Detail>
    <Date>05/20/14</Date>
  </DocLevel>
</StatusASN>
```

ANEXO 10.14 XSD DEL ESTATUS DEL ASN

Archivo XSD para validar el XML del estatus enviado por el Cliente por cada ASN.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="StatusASN">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DocLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:int" name="Vendor"/>
              <xs:element type="xs:string" name="Num"/>
              <xs:element type="xs:string" name="Status"/>
              <xs:element type="xs:string" name="DescError"/>
              <xs:element name="Detail">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:long" name="PO"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="Date"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.15 XML DEL CONSOLIDADO

Archivo en formato XML que genera el B2B por cada Consolidado enviado.

```
▼<Consolidado>
  ▼<ApplicationArea>
    <Sender>azurian1</Sender>
  </ApplicationArea>
  ▼<ConsolidadoLevel>
    <ASNConsolidado>0005530477</ASNConsolidado>
    <VendorId>509170</VendorId>
    <CreateDate>01/19/16 16:53</CreateDate>
    <ShippedDate>01/20/16 16:53</ShippedDate>
    <ExpectedDate>01/22/16 16:53</ExpectedDate>
    <DeliveryTo>8061</DeliveryTo>
    <FreightTerm>PREPAID</FreightTerm>
    <TotalWeight>11175.000</TotalWeight>
  </ConsolidadoLevel>
  ▼<OrderLevel>
    <PONumber>6061071280</PONumber>
    <PONumber>6062071290</PONumber>
    <PONumber>6161071283</PONumber>
  </OrderLevel>
</Consolidado>
```

ANEXO 10.16 XSD DEL CONSOLIDADO

Archivo XSD que valida la estructura del XML del Consolidado enviado.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:element name="Consolidado">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="ConsolidadoLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="ASNConsolidado"/>
              <xs:element type="xs:int" name="VendorId"/>
              <xs:element type="xs:string" name="CreateDate"/>
              <xs:element type="xs:string" name="ShippedDate"/>
              <xs:element type="xs:string" name="ExpectedDate"/>
              <xs:element type="xs:int" name="DeliveryTo"/>
              <xs:element type="xs:string" name="FreightTerm"/>
              <xs:element type="xs:float" name="TotalWeight"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="OrderLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:long" name="PONumber" maxOccurs="unbounded" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.17 XML DEL ESTATUS DEL CONSOLIDADO

Archivo en formato XML que envía el Cliente por cada Consolidado enviado.

```
<?xml version="1.0" encoding="utf-8"?><StatusConsolidado>
  <ApplicationArea>
    <Sender>ZZ-homedepotmxr</Sender>
  </ApplicationArea>
  <DocLevel>
    <Vendor>505400</Vendor>
    <Num>0004741242</Num>
    <Status>A</Status>
    <DescError />
    <Detail>
      <PO>6649026030</PO>
      <PO>6724309522</PO>
    </Detail>
    <Date>04/23/14</Date>
  </DocLevel>
</StatusConsolidado>
```

ANEXO 10.18 XSD DEL ESTATUS DEL CONSOLIDADO

Archivo XSD que valida la estructura del XML del estatus del consolidado.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="StatusConsolidado">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ApplicationArea">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Sender"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="DocLevel">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:int" name="Vendor"/>
              <xs:element type="xs:int" name="Num"/>
              <xs:element type="xs:string" name="Status"/>
              <xs:element type="xs:string" name="DescError"/>
              <xs:element name="Detail">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:long" name="PO" maxOccurs="unbounded" minOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="Date"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.19 XML DEL ORDER PLAN

Archivo en formato XML que envía el Cliente cada mes.

```
<OrderPlan>
  <Sender>ZZ-homedepotmxt</Sender>
  <FcstHStartDate>09/29/2014</FcstHStartDate>
  <FcstHStopDate>12/21/2014</FcstHStopDate>
  <CreateDate>10/02/2014</CreateDate>
  <Vendor>
    <VendorId>907695</VendorId>
    <VendorName>BRUNNEN INTERNACIONAL S.A. DE C.V.</VendorName>
  </Vendor>
  <NumOrderPlan>10022014907695</NumOrderPlan>
  <DateEmission>10/02/2014</DateEmission>
  <Items>
    <SKU>101542</SKU>
    <EAN>0614922201973</EAN>
    <Uofm>EA</Uofm>
    <Timing>Weekly</Timing>
    <Fcst>
      <PeriodIntialDate>09/29/2014</PeriodIntialDate>
      <FinalDate>10/05/2014</FinalDate>
      <Period>1</Period>
      <FcstData>5</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>10/06/2014</PeriodIntialDate>
      <FinalDate>10/12/2014</FinalDate>
      <Period>2</Period>
      <FcstData>0</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>10/13/2014</PeriodIntialDate>
      <FinalDate>10/19/2014</FinalDate>
      <Period>3</Period>
      <FcstData>0</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>10/20/2014</PeriodIntialDate>
      <FinalDate>10/26/2014</FinalDate>
      <PeriodIntialDate>10/20/2014</PeriodIntialDate>
      <FinalDate>10/26/2014</FinalDate>
      <Period>4</Period>
      <FcstData>28</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>10/27/2014</PeriodIntialDate>
      <FinalDate>11/02/2014</FinalDate>
      <Period>5</Period>
      <FcstData>72</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>11/03/2014</PeriodIntialDate>
      <FinalDate>11/09/2014</FinalDate>
      <Period>6</Period>
      <FcstData>52</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>11/10/2014</PeriodIntialDate>
      <FinalDate>11/16/2014</FinalDate>
      <Period>7</Period>
      <FcstData>28</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>11/17/2014</PeriodIntialDate>
      <FinalDate>11/23/2014</FinalDate>
      <Period>8</Period>
      <FcstData>52</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>11/24/2014</PeriodIntialDate>
      <FinalDate>11/30/2014</FinalDate>
      <Period>9</Period>
      <FcstData>28</FcstData>
    </Fcst>
    <Fcst>
      <PeriodIntialDate>12/01/2014</PeriodIntialDate>
      <FinalDate>12/07/2014</FinalDate>
      <Period>10</Period>
      <FcstData>36</FcstData>
    </Fcst>
  </Items>
</OrderPlan>
```


ANEXO 10.20 XSD DEL ORDER PLAN

Archivo XSD que valida la estructura del XML del Order Plan.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="OrderPlan">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="Sender"/>
        <xs:element type="xs:string" name="FcstHStartDate"/>
        <xs:element type="xs:string" name="FcstHStopDate"/>
        <xs:element type="xs:string" name="CreateDate"/>
        <xs:element name="Vendor">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="VendorId"/>
              <xs:element type="xs:string" name="VendorName"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element type="xs:long" name="NumOrderPlan"/>
        <xs:element type="xs:string" name="DateEmission"/>
        <xs:element name="Items" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:int" name="SKU"/>
              <xs:element type="xs:string" name="EAN"/>
              <xs:element type="xs:string" name="Uofm"/>
              <xs:element type="xs:string" name="Timing"/>
              <xs:element name="Fcst" maxOccurs="unbounded" minOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="PeriodIntialDate"/>
                    <xs:element type="xs:string" name="FinalDate"/>
                    <xs:element type="xs:int" name="Period"/>
                    <xs:element type="xs:int" name="FcstData"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

ANEXO 10.21 PDF DEL ORDER PLAN

Archivo PDF que genera el B2B por cada XML del order plan que envía el Cliente.



ORDER PLAN 29-Sep-2014 - 21-Dic-2014

PROVEEDOR: 907695
NO. DE ORDER PLAN: 10022014907695
FECHA: 02-Oct-2014

SKU	29-Sep-2014 05-Oct-2014	06-Oct-2014 12-Oct-2014	13-Oct-2014 19-Oct-2014	20-Oct-2014 26-Oct-2014	27-Oct-2014 02-Nov-2014	03-Nov-2014 09-Nov-2014	10-Nov-2014 16-Nov-2014	17-Nov-2014 23-Nov-2014	24-Nov-2014 30-Nov-2014	01-Dic-2014 07-Dic-2014	08-Dic-2014 14-Dic-2014	15-Dic-2014 21-Dic-2014
101542	5	0	0	11	0	59	43	68	40	68	50	51
105554	24	36	20	28	72	52	28	52	28	36	40	56
105555	33	33	30	27	18	27	48	36	45	24	33	51
105556	16	8	20	12	16	20	8	36	20	20	12	36
167850	5	10	12	27	14	267	249	197	285	253	181	219
210627	6	11	39	55	6	329	200	256	270	241	260	240
210651	15	23	25	18	24	211	206	199	193	235	162	197
244192	48	240	216	168	108	1224	900	1752	924	1356	1296	1308
244474	32	52	48	68	60	464	488	464	420	524	444	480
244831	12	26	48	36	31	248	225	223	310	207	239	205
247971	18	18	28	25	9	151	135	143	145	154	129	145
248541	108	168	312	228	144	1296	1056	1392	1284	1620	1500	936
248647	30	86	142	141	37	477	546	535	543	513	527	455
250810	25	48	81	122	18	457	435	443	458	433	429	417
252190	28	36	56	68	0	332	280	344	304	336	368	248
252900	0	16	68	68	20	272	372	236	256	448	192	300
260234	44	96	132	192	32	564	544	408	428	436	476	408
262775	144	276	492	492	162	1782	2076	2184	1614	2736	1692	1746
264114	246	360	792	726	624	2988	3018	2748	2928	3036	2676	2598
264680	24	36	80	144	12	480	540	708	336	624	492	384
264881	24	36	42	60	18	264	282	282	378	198	276	300
302037	27	6	17	10	6	242	383	295	312	218	270	327
304671	15	15	27	28	8	189	196	327	78	135	160	207
730646	29	15	30	24	28	184	159	157	166	118	121	107
738329	132	204	168	288	288	1188	1008	1248	1308	1044	1212	1236
774365	48	240	300	144	0	1080	600	1572	1284	828	1068	876
790595	60	144	348	360	120	1008	1164	1704	1080	756	1152	1272
820472	11	12	19	24	10	226	233	241	296	277	252	190
850362	7	21	19	21	9	574	271	442	519	308	181	341
862391	3	8	17	7	3	231	162	248	214	251	161	181
871625	39	44	28	26	28	278	617	395	455	358	312	349
871921	0	7	8	17	30	354	386	306	521	235	290	369
891398	11	11	18	30	13	325	214	249	245	240	301	166
949149	6	2	19	12	29	430	250	626	374	180	401	360