



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

Herramientas de integración de datos entre los sistemas de planificación de recursos empresariales (ERP) y sistemas de comercio electrónico.

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLINEA DE INVESTIGACIÓN: Aplicaciones WEB

AUTOR: Mauricio Esteban Pérez Nasser

TUTOR: Ing. Mg. Carlos Israel Núñez Miranda

Ambato – Ecuador
Junio/2017

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: **“Herramientas de integración de datos entre los sistemas de planificación de recursos empresariales (ERP) y sistemas de comercio electrónico.”**, del señor Mauricio Esteban Pérez Nasser, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato junio, 2017

EL TUTOR




Ing. Mg. Carlos Israel Miranda Núñez

AUTORÍA

El presente Proyecto de Investigación titulado: “**Herramientas de integración de datos entre los sistemas de planificación de recursos empresariales (ERP) y sistemas de comercio electrónico.**”, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato junio, 2017




Mauricio Esteban Pérez Nasser
CC: 180501068-1

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación, como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato junio, 2017



Mauricio Esteban Pérez Nasser
CC: 180501068-1

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes: Ing. Mg. Oswaldo Paredes e Ing. Mg. Edwin Hernando Buenaño Valencia, revisó y aprobó el Informe Final del Proyecto de Investigación titulado **“Herramientas de integración de datos entre los sistemas de planificación de recursos empresariales (ERP) y sistemas de comercio electrónico.”**, presentado por el señor Mauricio Esteban Pérez Nasser de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Mg. Elsa Pilar Urrutia Urrutia
PRESIDENTE DEL TRIBUNAL



Ing. Mg. Oswaldo Eduardo Paredes Ochoa
DOCENTE CALIFICADOR



Ing. Mg. Edwin Hernando Buenaño Valencia
DOCENTE CALIFICADOR

DEDICATORIA:

Dedicado a todas las personas que me brindaron su apoyo en cada etapa de mi vida, principalmente a mi Abuelita y Madre por formar mi carácter y brindarme su afecto, mi hermano por poder contar con él en todo momento.

A mi novia por todos los momentos que hemos compartido y su forma de ser conmigo.

A todas las personas con las que crecí y me enseñaron a ser mejor persona como mis amigos de la universidad y gimnasio.

Mauricio Esteban Pérez Nasser

AGRADECIMIENTO:

Agradezco a Dios por brindarme la fuerza y la su iluminación en decisiones durante el trayecto.

A mi familia, amigos por su constante apoyo y comprensión en momentos difíciles.

A AlquimiaSoft por brindarme la oportunidad y su confianza.

A todo el personal tanto docente como administrativo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por brindarme sus conocimientos y darme la oportunidad de haberme formado profesionalmente.

Por último, un sincero agradecimiento al Ing. Carlos Núñez por su guía en el desarrollo del proyecto y compartir sus conocimientos.

Mauricio Esteban Pérez Nasser

ÍNDICE DE CONTENIDOS

RESUMEN EJECUTIVO.....	xv
INTRODUCCIÓN.....	xvii
CAPÍTULO 1: EL PROBLEMA.....	1
1.1. Tema.....	1
1.2. Planteamiento del Problema.....	1
1.3. Delimitación.....	3
1.4. Justificación.....	3
1.5. Objetivos.....	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivo Específicos.....	4
CAPÍTULO 2: MARCO TEORICO.....	5
2.1. Antecedentes Investigativos.....	5
2.2. Fundamentación Teórica.....	6
2.2.1. Sincronización de la información.....	6
2.2.2. Software libre.....	7
2.2.3. Base de datos.....	7
2.2.4. Sistema de Gestión de Bases de Datos.....	8
2.2.5. Metodología Ágil.....	8
2.2.6. Manifiesto Ágil.....	9
2.2.7. Metodología de desarrollo XP.....	10
2.2.8. ERP.....	11
2.2.9. iDempiere ERP.....	12
2.2.10. Plug-in.....	13
2.2.11. JAR.....	13
2.2.12. Tareas Programadas (<i>Crontab</i>).....	14

2.2.13.	Comercio Electrónico	14
2.2.14.	Prestashop	15
2.2.15.	JMS	15
2.2.16.	Servidor de aplicaciones web	16
2.2.17.	Aplicación	16
2.2.18.	Aplicación Web	17
2.2.19.	Servicio Web.....	17
2.2.20.	REST.....	17
2.2.21.	HTTP	18
2.2.22.	XML.....	18
2.3.	Propuesta de solución.....	19
CAPÍTULO 3: METODOLOGÍA.....		20
3.1.	Modalidad de la Investigación	20
3.2.	Población y muestra.....	21
3.3.	Recolección de Información	21
3.4.	Procesamiento y Análisis de Datos.....	21
3.4.	Desarrollo del proyecto.....	21
CAPÍTULO 4: DESARROLLO DE LA PROPUESTA.....		23
4.1.	Fase I: Exploración	23
4.2.	Fase II: Planificación del proyecto	25
4.2.1.	Historias de Usuario.....	25
4.2.2.	Estimación de Historias de Usuario.....	29
4.3.	Fase III: Iteraciones.	30
4.3.1	Plan de Entrega.	30
4.3.2	Iteraciones.	30
4.4.	Fase IV: Producción.....	31

4.4.1. Diseño del modelo de sincronización.	31
4.4.2. Análisis de almacenamiento de datos de Atix ERP.	33
4.4.3. Análisis de almacenamiento de datos de tienda virtual Prestashop.	35
4.4.4. Diseño del modelo de sincronización.	41
4.4.5. Desarrollo de Plug-In en Atix ERP.	42
4.4.5.1 Lectura de categorías en Atix ERP.	44
4.4.5.2 Lectura de productos en Atix ERP.	46
4.4.3.5.3 Envío de información al sistema de mensajería JMS.	49
4.4.5.4 Lectura de Clientes y Pedidos.	52
4.4.5.4.1 Web Service de lectura de pedidos.	58
4.4.5.4.2 Web Service de lectura de clientes.	59
4.4.5.5 Lectura de Categorías y Productos e inserción.	60
4.4.6. Glosario de métodos.	73
4.4.6.1 Métodos Empleados en Atix ERP.	73
4.4.6.2 Métodos Empleados en JAR de sincronización en tienda virtual.	74
4.5. Fase V: Pruebas.	75
4.5.1. Pruebas de Caja Blanca.	75
4.5.1.1 Configuración para sincronizador de la tienda virtual.	75
4.5.1.1.1 Creación de script de ejecución.	76
4.5.1.1.2 Configuración de tarea crontab.	76
4.5.1.2 Prueba de envío de categorías y productos.	77
4.5.2. Pruebas de Caja Negra.	79
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	87
5.1 Conclusiones	87
5.1 Recomendaciones	88
Bibliografía:	89

ÍNDICE DE IMÁGENES

Imagen 1. Componentes de un Sistema de Recursos Empresariales.	12
Imagen 2. Diseño del modelo de sincronización.	32
Imagen 3. Pantalla principal del módulo de sincronización.	43
Imagen 4. Pantalla de búsqueda de productos del módulo de sincronización.	44
Imagen 5. Ventana de ejecución de proceso en segundo plano.	44
Imagen 6. Sentencia SQL para la consulta de Categorías.	45
Imagen 7. Ventana de Producto Atix ERP.	46
Imagen 8. Codificación para la detección de actualización de precio.	47
Imagen 9. Codificación para la detección de cambios en los atributos del producto.	48
Imagen 10. Sentencia SQL para selección de productos nuevos a enviar.	49
Imagen 11. Sentencia SQL para selección de productos actualizados a enviar.	49
Imagen 12. Codificación para el envío de información a la cola de mensajería.	50
Imagen 13. Interfaz de contenedores de colas de mensajería.	51
Imagen 14. Interfaz de la cola de mensajería pendiente.	51
Imagen 15. Interfaz del contenido del mensaje de sistema de mensajería.	52
Imagen 16. Diagrama de flujo de proceso para ingreso de pedidos.	54
Imagen 17. Codificación de consulta de datos del cliente por identificación.	55
Imagen 18. Codificación de lectura de líneas de pedido según identificador.	56
Imagen 19. Codificación del ingreso de pedido a Atix ERP.	57
Imagen 20. Web Service lectura de encabezados de órdenes.	58
Imagen 21. Web Service para obtención de identificación del cliente.	59
Imagen 22. Diagrama de flujo de proceso, lectura e ingreso de productos y categorías.	61
Imagen 23. Codificación para recepción de datos del sistema de mensajería.	63
Imagen 24. Codificación del método para el ingreso de categorías.	64
Imagen 25. Codificación de método para eliminación de caracteres especiales.	65

Imagen 26. Web Service de inserción de categorías en tienda virtual.....	65
Imagen 27. Codificación del método para ingreso de productos.....	66
Imagen 28. Web Service de inserción de productos en tienda virtual.	67
Imagen 29. Codificación del método para la actualización de productos.....	68
Imagen 30. Web Service de Consulta de ID producto.....	69
Imagen 31. Web Service de Actualización de Productos.	70
Imagen 32. Clase Product para encapsulamiento de información.	71
Imagen 33. Clase Category para encapsulamiento de información.	72
Imagen 34. Contenido del archivo sinc.sh.....	76
Imagen 35. Configuración de tarea programada (Crontab).	77
Imagen 36. Elección de opciones para prueba de envío de información.....	78
Imagen 37. Bloqueo de Interfaz en proceso de sincronización.	78
Imagen 38. Posibles caminos para la apertura y ejecución del módulo.....	79
Imagen 39. Prueba de apertura del módulo.	80
Imagen 40. Camino para la sincronización de productos.	80
Imagen 41. Prueba de pantalla de bloqueo del módulo.	81
Imagen 42. Información de cantidad de datos enviados presentado por Atix ERP.....	81
Imagen 43. Información de cantidad de datos recibidos por MqActive.	82
Imagen 44. Interfaz de tienda virtual sin contenido de categoría.	82
Imagen 45. Interfaz de tienda virtual sin contenido de productos.	83
Imagen 46. Interfaz de categorías sincronizadas en Prestashop.	84
Imagen 47. Categorías padres en tienda virtual.....	84
Imagen 48. Categorías pertenecientes a "Ferretería"	85
Imagen 49. Productos sincronizados en tienda virtual.....	85

ÍNDICE DE TABLAS

Tabla 1. Cuadro de preguntas a usuarios encargados de la sincronización manual.	24
Tabla 2. Plantilla de historia de Usuario.....	25
Tabla 3. Historia de Usuario: Envío de Información.....	26
Tabla 4. Historia de Usuario: Actualización de precios.	27
Tabla 5. Historia de Usuario: Ingreso de Pedidos.	27
Tabla 6. Historia de Usuario: Cantidad individual de productos.....	27
Tabla 7. Historia de Usuario: Ingreso de clientes nuevos.....	28
Tabla 8. Historia de Usuario: Ingreso de Categorías.	28
Tabla 9. Historia de Usuario: Actualización de nombres de productos.....	28
Tabla 10. Estimación de historias de Usuario.....	29
Tabla 11. Estimación de historias de Usuario.....	29
Tabla 12. Estimación de historias de Usuario.....	30
Tabla 13. Plan de Entrega.....	30
Tabla 14. Histórico de versiones en Historias de Usuario.....	31
Tabla 15. Estructura tabla productos Atix ERP.....	33
Tabla 16. Estructura tabla impuestos Atix ERP	34
Tabla 17. Estructura tabla categoría de productos Atix ERP.....	34
Tabla 18. Estructura de la tabla Unidad de medida	34
Tabla 19. Estructura de la tabla Lista de Precios	35
Tabla 20. Estructura de la tabla Producto	36
Tabla 21. Estructura de la tabla traducciones del producto	37
Tabla 22. Estructura de la tabla Tienda del Producto	37
Tabla 23. Estructura de la Tabla Cantidades de Productos.....	38
Tabla 24. Estructura de la tabla categoría del producto.....	38
Tabla 25. Estructura de la tabla Categorías	39

Tabla 26. Estructura de la tabla Traducciones de Categoría.....	39
Tabla 27. Estructura de la tabla Categorías de la Tienda.....	39
Tabla 28. Estructura de la tabla Grupo de Categoría.....	40
Tabla 29. Estructura de la tabla Cliente.....	40
Tabla 30. Estructura de la tabla Direcciones de Clientes.....	40
Tabla 31. Estructura de la tabla Ordenes de Venta.....	41
Tabla 32. Estructura de la tabla Detalle de Órdenes de Venta Prestashop	41
Tabla 33. Comparacion de datos enviados entre Interfaz y MqActive.....	82

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERIA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL

INGENIERIA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

“HERRAMIENTAS DE INTEGRACIÓN DE DATOS ENTRE LOS SISTEMAS DE PLANIFICACIÓN DE RECURSOS EMPRESARIALES (ERP) Y SISTEMAS DE COMERCIO ELECTRÓNICO.”

Autor: Mauricio Esteban Pérez Nasser.

Tutor: Ing. Mg. Carlos Israel Núñez Miranda.

Fecha: 04 de mayo de 2017.

RESUMEN EJECUTIVO

La constante evolución de los sistemas y aplicaciones web han impulsado el desarrollo de un sinnúmero de empresas e instituciones dedicadas a varias áreas comerciales, la manera substancial de solucionar problemas e impulsar mejoras hacen que día a día más empresas cambien el tradicionalismo por modernismo al momento de trabajar, por esta razón para seguir nutriendo de estos sistemas a la colectividad empresarial la Universidad Técnica de Ambato con su Carrera de Ingeniería en Sistemas Computacionales e Informáticos, orienta a sus estudiantes y futuros profesionales a ser partícipes en el desarrollo de las empresas e instituciones mediante aplicaciones que solucionen y faciliten la optimización de tiempo en los procesos que se realizan a diario.

La aplicación web que se detalla a continuación, tiene como fin impulsar el ERP de la empresa MegaProfer S.A, mediante una optimización de la comunicación que existe con su Tienda Virtual, la misma que procede a realizar múltiples pedidos de una gran variedad de productos, la aplicación web tomará este proceso obsoleto y lo adecuará al nuevo modelo ágil que exige una empresa en desarrollo como es MegaProfer S.A.

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTY OF ENGINEERING SYSTEMS, ELECTRONICS AND
INDUSTRIAL

ENGINEERING IN COMPUTATIONAL AND COMPUTER SYSTEMS

“DATA INTEGRATION TOOLS BETWEEN BUSINESS RESOURCE PLANNING
(ERP) SYSTEMS AND ELECTRONIC COMMERCE SYSTEMS.”

Author: Mauricio Esteban Pérez Nasser.

Tutor: Ing. Mg. Carlos Israel Núñez Miranda.

Date: May, 04 de de 2017.

EXECUTIVE SUMMARY

Ongoing evolution of computational systems and web applications has improved the growth and expansion of a great deal of companies and institutions, which are focused on various commercial enterprises. Nowadays technological challenges and the need of improvement, causes many companies to update their obsolete work models. This is why the support of this kind requirements is on demand. Universidad Técnica de Ambato and its Informatics Systems Engineering Career, encourage their future professionals to actively participate in the business expansion of local companies and institutions. In order to reach that goal, the development of fast, scalable and reliable applications is needed.

The web application detailed in this work, aims to improve the ERP of Megaprofer S.A. This will be achieved by optimizing the communication between the current database and its virtual store, which handles a great deal of requests from a variety of products. The web application will completely renovate this out of date process, fitting its performance to the requirements of Megaprofer S.A.

INTRODUCCIÓN

El desarrollo del informe final del presente trabajo de investigación se encuentra dividido en capítulos, los mismos que facilitan la comprensión del contenido del mismo:

El Capítulo I.- Denominado “EL PROBLEMA”, se encuentra el problema a investigar, la justificación respectiva de la investigación y el planteamiento de los objetivos a obtener tras la culminación del trabajo de investigación.

El Capítulo II.- Contiene el “MARCO TEÓRICO”, en donde se establece el conjunto de conocimientos en los cuales se sustenta la investigación, además de presentar investigaciones previas que sirven de soporte a la investigación y se establece la propuesta de solución del problema.

El Capítulo III.- Comprende la “METODOLOGÍA”, en la cual se especifica la metodología de investigación a utilizar, el proceso de recolección de la información, el procesamiento y análisis de la información recabada y define las etapas para el desarrollo del proyecto.

El Capítulo IV.- Consta del “DESARROLLO DE LA PROPUESTA”, que detalla el desarrollo del proyecto en base a la metodología establecida para la sincronización de sistemas empresariales y comercio electrónico.

El Capítulo V.- Abarca “CONCLUSIONES Y RECOMENDACIONES”, las cuales son determinadas en base a la culminación desarrollo del proyecto.

ANEXOS: Consta el manual del usuario como informativo al usuario final sobre el manejo del módulo desarrollado.

CAPÍTULO 1

EL PROBLEMA

1.1. Tema

“Herramientas de integración de datos entre los sistemas de planificación de recursos empresariales (ERP) y sistemas de comercio electrónico.”

1.2. Planteamiento del Problema

A nivel mundial las empresas se han adaptado y evolucionado junto a la tecnología utilizando nuevas herramientas de software que proveen una forma mejorada para gestionar los procedimientos de las mismas. Estas han sido de gran ayuda para mantener un registro de información y consolidación en las bases de datos para mantener información unificada y disponible. [1]

En toda empresa la publicación de sus productos y/o servicios ha sido fundamental para su evolución y desarrollo económico. Actualmente el manejo de esta información es administrado por sistemas, los cuales deben ser eficaces, precisos y rápidos, ya que el cliente busca un mejor servicio, ahorro de tiempo y dinero. [2]

Los sistemas que poseen esta información se encuentran distribuidos, y, por lo general manejan diferentes gestores de bases de datos, esquemas y codificación de lenguaje, estos deben ser sincronizados de alguna manera para mantener la integridad de los datos y disponibilidad de los mismos.

La sincronización de datos automática es la mejor manera de disponer información real y actualizada de la misma, esto implica que, esta debe ser confiable y estar disponible en el momento que el usuario la desee. Esto es posible gracias a herramientas de mensajería que facilita el intercambio de datos entre sistemas por medio de una plataforma intermedia. [3]

El grupo ferretero Megaprofer dispone de diferentes sistemas informáticos que soportan las actividades internas y de comercio de la empresa. Una de ellas es ATIX ERP (idempiere), un sistema de administración de recursos empresariales que contiene todas las transacciones comerciales de la empresa. El otro, una tienda de comercio electrónico (*ecommerce* Prestashop) con la función de ser un vendedor más para la empresa, proporcionando información sobre los productos que se disponen a la venta, recolectando los pedidos de los usuarios registrados.

Atix ERP debe contener toda la información de pedidos de los clientes, para poder proceder con su respectivo flujo de proceso y llegar a la facturación del mismo.

Para poder cumplir este objetivo se procede a realizar una sincronización manual, es decir, se leen los pedidos ingresados en la tienda virtual y son transcritos como órdenes de compra. Esto produce errores en la facturación, ya que se puede ingresar productos o cantidades diferentes a las solicitadas.

De igual manera para la publicación de los productos y categorías, los empleados se encargan de crearlos y modificarlos directamente en la tienda virtual. Lo cual produce errores de digitación de propiedades de los productos ocasionando toda una cadena de errores en los procesos de facturación.

Esta forma manual de ingreso y actualización de información en ambos sistemas provoca un retraso en la disponibilidad de información tanto para los consumidores como a la empresa.

Se debe sumar el cambio de personal que administre la tienda virtual, ya que para poder manipular la información se debe proceder con una capacitación previa sobre el mismo, la cual conlleva un tiempo aproximado de una semana, provocando retraso en el proceso de mantenimiento de información.

1.3. Delimitación

- Delimitación de contenidos:
 - Área Académica: Software
 - Línea de Investigación: Desarrollo de Software
 - Sublínea: Aplicaciones Web
- Delimitación Espacial: Este proyecto se desarrolla en el proveedor ferretero MegaProfer SA ubicado en la ciudad de Ambato.
- Delimitación temporal: El sistema se desarrolla desde diciembre de 2016 hasta junio de 2017.

1.4. Justificación

La sincronización de información en los sistemas empresariales, debe ser necesaria para mejorar la calidad del servicio que se proporciona a los clientes por parte de la empresa, para mantener los datos actualizados y disponibles en el momento que se los requiera, es imperioso brindar confiabilidad en la veracidad de la información y rapidez en la transacción a realizar.

El impacto será relevante ya que contribuye en la mejora y optimización de los procesos de la empresa MegaProfer, brindando a sus clientes datos actualizados, eliminando la inserción y actualización de datos manual.

Además, es factible su realización ya que tiene el apoyo de la Empresa de desarrollo de software AlquimiaSoft, encargada de desarrollar y administrar sistemas para MegaProfer. La cual cuenta con la disponibilidad de información y un entorno para realizar pruebas del sistema con el fin de optimizar su funcionamiento y corregir posibles fallas que se presenten en el proceso de mejora.

Debido a las diferentes fuentes de ingreso de información, cambio del personal administrador de los datos e información no centralizada, se produce incoherencia de datos entre los diferentes sistemas, provocando problemas en la empresa al no poseer información actualizada.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un sistema de sincronización de datos entre sistemas de planificación de recursos empresariales y sistemas de comercio electrónico mediante un sistema de mensajería.

1.5.2. Objetivo Específicos

1. Analizar el ingreso y procesamiento de la información de planificación empresarial y el comercio electrónico.
2. Diseñar un modelo (esquema de información de datos) que se adapte a los requerimientos de información entre las diferentes bases
3. Implementar el sistema de sincronización asincrónica de datos entre sistemas de planificación de recursos empresariales y sistemas de comercio electrónico mediante un sistema de mensajería.

CAPÍTULO 2

MARCO TEORICO

2.1. Antecedentes Investigativos

Previo a realizar la investigación que conlleva desarrollar una optimización de un proceso, es necesario conocer algunos antecedentes referentes al caso, algunos de los siguientes clarifican el objetivo y el fin de la investigación.

Jorge Eduardo Chicaiza Rugel, en 2005 emplea las herramientas propias de los gestores de bases de datos para replicar información entre diferentes motores luego de haber analizado a profundidad su funcionamiento. Utilizando los sistemas Administradores de Bases de Datos SQL Server 7.0 y Oracle8 Server Enterprise Edition. Concluyendo que con dichas herramientas y el conocimiento sobre las mismas la tarea de replicación se vuelve sumamente fácil y administrable por medio de un entorno gráfico.[4]

Paredes Guerrero Cristhian Marcelo, en 2012 detalla la metodología para implantar un ERP y su funcionamiento. Partiendo desde el análisis, diseño, desarrollo e implementación, con principal enfoque en los módulos contables y administrativos. Utilizando metodología OMT (Metodología de modelado de objetos) para su desarrollo. Una vez terminado su desarrollo se concluye que el sistema ERP logró automatizar procesos críticos para la empresa, los cuales anteriormente tenían retraso en su gestión. [5]

Chuquimarca Morales Edwin Adolfo, en 2003 detalla los procesos de diseño y construcción de un *e-commerce* (Sistema de comercio electrónico), junto a todas sus consideraciones en cuanto a requerimientos de información, para un buen desempeño en la web. Emplea lenguaje PHP y base de datos MySQL para la construcción del sitio web considerando todos los requerimientos y procesos que una tienda virtual conlleva.

Concluyendo que una tienda virtual proporciona ventajas a una empresa en el mundo de los negocios por medio de la presentación de sus productos y la competitividad de los precios. [6]

Matute Rivera Soraya Gabriela, en 2013 plantea la sincronización de información de diferente fuente por medio de servicios web a diferentes bases de datos para la automatización de los procesos de control áreas y recursos de una empresa. Por medio de sistemas de código abierto basados en lenguaje de desarrollo Java y base de datos en PostgreSQL. Concluyendo que la sincronización logro solucionar las ineficiencias detectadas en los procesos que anteriormente eran manuales, logrando un incremento de la productividad en cada una de las áreas involucradas [7]

Javier Alfonso Valdés, Ángela Gloria Gómez Peña, Luis Alberto Pimentel González en 2011 plantea una replicación de datos utilizando mensajería JMS(*Java Message Service*) para el envío de una gran cantidad de datos desde bases de datos relacionales, tomando en cuenta los requerimientos funcionales y no funcionales de los sistemas. Concluyendo que con la utilización de mensajería JMS y el desarrollo de un adecuado componente se garantiza la fiabilidad, calidad y consistencia de los datos. [8]

2.2. Fundamentación Teórica

2.2.1. Sincronización de la información

Consiste en mantener la información actualizada mediante procesos establecidos, que responden bajo el parámetro tiempo. Existen dos tiempos de sincronización que es síncrona y asíncrona.

- Mensajería Síncrona

“Mensajería síncrona implica procesos fuertemente acoplados, donde el envío y recepción de las aplicaciones se comunican directamente y ambos deben ser disponibles para que el intercambio de mensajes que se produzca.” [9]

- Mensajería Asincrónica

“Mensajería asincrónica implica procesos vagamente acoplados, donde el envío y recepción de las aplicaciones se comunican a través de una mensajería proveedor. La aplicación de envío es capaz de pasar los datos a la mensajería proveedor y luego continuar con su procesamiento. La aplicación receptora es capaz de conectarse al proveedor de mensajería, posiblemente en algún momento posterior en el tiempo, para recuperar los datos.” [9]

Claramente la sincronización síncrona tiene un mayor número de requerimientos, debido que debe cumplir con su objetivo directamente, tomando en cuenta que los dos sistemas estén disponibles.

2.2.2. Software libre

“El software libre está identificado por 4 principios básicos de libertad:

1. La libertad de ejecutar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades del usuario. El acceso al código fuente es una condición previa para esto.
3. La libertad de redistribuir copias para que pueda ayudar a los vecinos.
4. La libertad de mejorar el programa y publicar sus mejoras al público, de manera que beneficie a toda la comunidad. El acceso al código fuente es una condición previa para esto.” [10]

El uso de software libre beneficia a las empresas en costos de desarrollo, ya que no debe pagar por licencias. No requiere soporte pagado debido que existen grandes comunidades en internet dedicadas a brindar ayuda y publicar constantes mejoras y correcciones.

2.2.3. Base de datos

“Una Base de Datos (BD) es una colección o depósito de datos, almacenados en soporte secundario y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos y su

definición (estructura de la BD), única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual debe permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos.” [11]

De acuerdo al autor es un lugar de almacenamiento de información, la ventaja de usarla es que se controla la integridad y redundancia de los datos. También proveen medidas de seguridad a los datos como recuperación de los mismos para evitar pérdidas de información.

2.2.4. Sistema de Gestión de Bases de Datos

“Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto coordinado de programas, procedimientos, lenguajes, herramientas, etc., que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o administradores de una BD, los medios necesarios para describir y manipular los datos integrados en la BD, manteniendo su integridad, confidencialidad y disponibilidad. Entre los diferentes SGBD, se puede encontrar los siguientes:

- MySql.
- PostgreSQL
- Oracle
- Access
- Microsoft SQL Server, etc.” [11]

Los sistemas gestores de bases de datos son herramientas que proporcionan un fácil acceso y manipulación de las bases de datos, por medio de componentes gráficos. Estas permiten una fácil administración y mantenimiento, para personas con alto y bajo conocimiento en bases de datos.

2.2.5. Metodología Ágil

“En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de

la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.” [12]

El inicio de esta metodología es el Manifiesto Ágil, un documento que plasma esta filosofía.

2.2.6. Manifiesto Ágil

Comienza enumerando los principales valores del desarrollo ágil. Se valora:

- “Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Si se sigue un buen proceso de desarrollo, pero el equipo falla, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. No se necesitan desarrolladores brillantes, sino desarrolladores que se adapten bien al trabajo en equipo. Así mismo, las herramientas (compiladores, depuradores, control de versiones, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también puede afectar negativamente. En resumen, es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. Aunque se parte de la base que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma

inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.

- La colaboración con el cliente más que la negociación de un contrato. La característica particular del desarrollo de software hace que muchos proyectos hayan fracasado por intentar cumplir unos plazos y unos costes preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.” [12]

2.2.7. Metodología de desarrollo XP.

“Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes donde existe un alto riesgo técnico.” [12]

“Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:” [13]

1. El cliente especifica el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué desarrollar, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador desarrolla ese valor de negocio.
5. Se regresa al paso 1.

“En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.” [12]

“El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.” [14]

2.2.8. ERP

“Un ERP (*Enterprise Resource Planning*), es un sistema integrado de gestión que cubre de forma total diferentes áreas funcionales de una empresa. Es un sistema de información estructurado que permite la gestión integral de las diversas áreas empresariales. A continuación, se representa de forma gráfica, las diferentes posibles partes de un ERP.” [12]

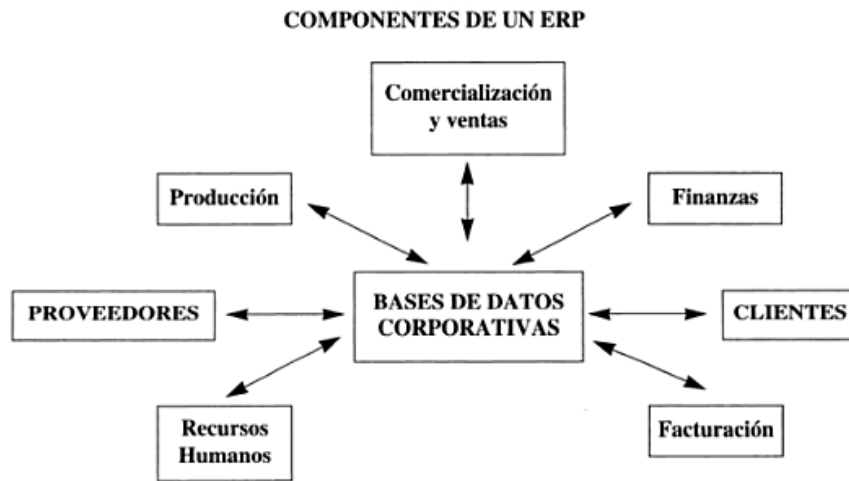


Imagen 1. Componentes de un Sistema de Recursos Empresariales.

Fuente: Carmen de Pablos, José Joaquín López-Hermoso, Santiago Martín-Romo, Sonia Medina

Los sistemas ERP implementan ciertas mejoras importantes al tratamiento de la información, que lo diferencia de otros tipos de sistemas existentes. [12]

2.2.9. iDempiere ERP

“iDempiere ERP CRM & SCM es una sofisticada solución de negocios que permite integrar y automatizar los procesos empresariales clave. Se ha posicionado a nivel internacional como la herramienta alternativa para empresas medianas y pequeñas, frente a los sistemas propietarios tradicionales, con alto costo de licencias y a los desarrollos locales, que no satisfacen las necesidades de los mercados globalizados.

iDempiere más allá de su riqueza funcional, puede incorporar cambios específicos del negocio y preservarlos para futuras versiones. Su filosofía permite realizar cambios “en caliente” (estando ya en producción) tales como: el modelo de negocio, la estructura de la información, el plan de cuentas, el tipo de moneda, regulaciones legales o impositivas, entre otros, sin afectar la operación.

iDempiere es independiente de la base de datos, no está asociado a ningún proveedor o implementador en particular, inclusive es posible acceder a su código fuente.

La comunidad de iDempiere asegura la evolución en el tiempo del sistema, su crecimiento y control, corroborando de esta manera su compromiso con el modelo de código abierto, de forma que se pueda obtener, redistribuir y modificar su código fuente. La contribución de la comunidad se verá reflejada en el compromiso con el fortalecimiento de la herramienta, la búsqueda de continuidad en el tiempo y la selección de partners basada en casos de éxito.” [13]

Alquimia Soft S.A nombra a iDempiere como “ATIX ERP”, que proviene de la palabra Inca “El que te puede ayudar”. La empresa desarrolla *plug-ins* para personalizar y adaptar a las diferentes necesidades de las empresas que requieren del sistema.

2.2.10. Plug-in

“Un Plug in es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software. En nuestro idioma, por lo tanto, puede nombrarse al plug in como un complemento.

Lo habitual es que el plug in sea ejecutado mediante el software principal, con el que interactúa a través de una cierta interfaz. En la actualidad, la mayoría de los programas trabajan con plug ins.

Una de las principales ventajas que ofrecen estos complementos es que facilitan la colaboración de desarrolladores externos con el software. Estos desarrolladores pueden realizar sus aportes a las funcionalidades a través de distintos plugins.

En ocasiones, puede surgir un conflicto entre un plugin y la aplicación principal que provoque diversos fallos. En estos casos, por lo general, el software brinda la opción de desactivar el plugin de manera temporal o incluso desinstalarlo.” [14]

2.2.11. JAR

“Es un formato de archivo de Java que permite agrupar varios archivos en un único archivo. Normalmente, un archivo JAR contiene los archivos de clase y recursos auxiliares y aplicaciones.

El formato de archivo JAR ofrece muchas ventajas:

- Seguridad: Se puede firmar digitalmente el contenido de un archivo JAR. Los usuarios que reconocen su firma a continuación, opcionalmente pueden otorgar privilegios de seguridad de software que no tendría de otra manera.
- Compresión: Permite comprimir sus archivos para un almacenamiento eficiente.
- Embalaje para las extensiones: El marco extensiones proporciona un medio por el cual se puede agregar funcionalidad a la plataforma central de Java, y el formato de archivo JAR define el embalaje de las extensiones. Al utilizar el formato de archivo JAR, puede convertir su software en extensiones también.
- Paquete Sellado: Los paquetes almacenados en archivos JAR pueden sellarse opcionalmente por lo que el paquete puede hacer cumplir la consistencia versión. Aislamiento de un paquete dentro de un archivo JAR significa que todas las clases definidas en ese paquete deben encontrarse en el mismo archivo JAR.
- Versiones del paquete: Un archivo JAR puede contener datos sobre los archivos que contiene, como proveedor y la versión.
- Portabilidad: El mecanismo para el manejo de archivos JAR es una parte estándar del núcleo de la API de la plataforma Java.” [18]

2.2.12. Tareas Programadas (*Crontab*)

Es un demonio (tarea) que ejecuta comandos programados y permite a usuarios de los sistemas operativos tipo Linux/Unix, ejecutar automáticamente comandos o shell-scripts (secuencia de comandos) a una hora y/o fecha específica.

Generalmente se utiliza para realizar tareas administrativas, como respaldos, pero puede utilizarse para ejecutar cualquier tarea. [19]

2.2.13. Comercio Electrónico

En inglés *e-Commerce* “se puede definir como la compra-venta de productos y servicios a través de sistemas electrónicos, principalmente Internet. En base a herramientas electrónicas que apoyan la compraventa de productos y servicios, independientemente de dónde se realice la transacción del pago final.

Una de las principales diferencias entre el comercio electrónico y el comercio “físico” es el uso de la tecnología.

En la mayor parte de las ocasiones, la principal barrera de entrada al comercio en internet es el desconocimiento de las tecnologías y funcionalidades que conforman la tienda y el propio medio en el que se va a desenvolver la empresa, a la hora de vender sus productos o servicios.” [13]

El comercio electrónico tiene como pilar fundamental al Internet y la tecnología para su funcionamiento. Su objetivo es la venta y promoción de productos/servicios por medios electrónicos llegando a un mayor número de clientes.

2.2.14. Prestashop

Es un software de comercio electrónico de licencia libre y código abierto, con una de las comunidades más grandes en el software libre. Está diseñado para ser fácil de utilizar, tomando como fuerte un diseño que se adapta a todo tipo de dispositivo para su correcta visualización.

Su funcionamiento y modelo de negocio se basa en módulos y plantillas, la gran mayoría de pago, ya que son desarrolladas por empresas externas y publicadas en su tienda de complementos. [14]

2.2.15. JMS

“JMS es la API Java estándar para aplicaciones para llevar a cabo mensajería. Se trata de mensajes como en dos "dominios" correspondiente a dos formas de utilizar la mensajería: punto a punto y de publicación / suscripción.

La especificación fue desarrollada por Sun Microsystems con la ayuda de IBM, otros proveedores de mensajería y otros proveedores de servidores de aplicaciones.

La norma define un conjunto de interfaces de Java que será ejecutado por el proveedor de mensajería, o "Proveedor", para usar la terminología de J2EE. Esto significa que las

aplicaciones escritas con la API JMS son independiente del proveedor, y puede ser redistribuido entre productos de proveedores sin modificación.” [15]

2.2.16. Servidor de aplicaciones web

“Un servidor de aplicaciones es el elemento (software) que es capaz de traducir las instrucciones y además comunicar con otros servidores (como por ejemplo los servidores de bases de datos) para extraer información de la empresa, que se necesita para resolver la petición.

Los servidores de aplicaciones trabajan en conjunto con los servidores web para que el proceso se haga de forma transparente al usuario; es decir el usuario pide el servicio a través, normalmente, de su navegador y el servidor web atiende la petición y pide al servidor de aplicaciones la traducción de la aplicación contenida a fin mostrar al usuario el resultado de forma entendible por su navegador (es decir en formato HTML).

A la forma de trabajar de un servidor de aplicaciones, se le conoce normalmente como arquitectura de tres capas (a veces se habla de más capas). Una primera capa es la del navegador que es capaz de traducir el llamado código del lado del cliente (HTML, JavaScript, CSS, Flash,).

La segunda capa la forma el servidor de aplicaciones en su labor de traducir código en el lado del servidor (JSP, PHP, Ruby on Rails, Cold Fussion...) y convertirlo al formato entendible por el navegador.

La tercera capa son todos los servicios a los que accede el servidor de aplicaciones que necesita para poder realizar la tarea encomendada a la aplicación (por ejemplo, el acceso a la base de datos).” [16]

2.2.17. Aplicación

“Una aplicación es un programa informático diseñado como una herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos o automatizar procesos.” [17]

2.2.18. Aplicación Web

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador1) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores. HTTP se sitúa en el nivel 7 (aplicación) del modelo OSI. [18]

2.2.19. Servicio Web

Un servicio Web (*WebService*) es un servicio ofrecido por una aplicación, que publica su lógica a clientes de cualquier plataforma por medio de una interfaz accesible a través de la red utilizando tecnologías (protocolos) estándar de Internet. [19]

Ventajas de los servicios Web:

- Ofrecen una “tecnología distribuida de componentes” optimizada.
- Evitan los problemas inherentes a la existencia de firewalls.
- Permiten una llamada sencilla de métodos.
- Los clientes (Consumidores) pueden estar en cualquier plataforma.
- Permiten centralizar datos. [19]

2.2.20. REST

“REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web.

REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen como los recursos son definidos y diseccionados. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP.

Cabe destacar que REST no es un estándar, es tan solo un estilo de arquitectura.

Aunque REST no es un estándar, está basado en estándares:

- HTTP
- URL
- Representación de los recursos: XML/HTML/GIF/JPEG/...
- Tipos MIME: text/xml, text/html.” [20]

2.2.21. HTTP

“HTTP (*hypertext transfer protocol*) es el protocolo base de la WWW. Se trata de un protocolo simple, orientado a conexión y sin estado. La razón de que esté orientado a conexión, es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, *transport control protocol*) de modo conectado, un protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir, en contraposición a los protocolos de datagrama o no orientados a conexión, que dividen los datos en pequeños paquetes (datagramas) y los envían, pudiendo llegar por vías diferentes del servidor al cliente. El protocolo no mantiene estado, es decir, cada transferencia de datos es una conexión independiente de la anterior, sin relación alguna entre ellas, hasta el punto de que para transferir una página web, se tiene que enviar el código HTML del texto, así como las imágenes que la componen, pues en la especificación inicial de HTTP, la 1.0, se abrían y usaban tantas conexiones como componentes tenía la página, transfiriéndose por cada conexión un componente (el texto de la página o cada una de las imágenes).” [21]

2.2.22. XML

“XML (*eXtensible Markup Language*) no es, como su nombre podría sugerir, un lenguaje de marcado. XML es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados.

Los documentos XML deben seguir una estructura estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar

correctamente "incluida" en otra. Además, los elementos con contenido, deben estar correctamente "cerrados".

Los documentos XML sólo permiten un elemento raíz, del que todos los demás sean parte. Es decir, la jerarquía de elementos de un documento XML bien-formado sólo puede tener un elemento inicial.” [22]

2.3. Propuesta de solución

La presente investigación propone realizar un módulo de sincronización web, utilizando mensajería (JMS) que permitirá actualizar información entre los sistemas de forma automática para solucionar los problemas de actualización e inserción de los datos en la empresa MegaProfer.

CAPÍTULO 3 METODOLOGÍA

3.1. Modalidad de la Investigación

La presente investigación es aplicada porque pretende automatizar el proceso de inserción y actualización entre sistemas empresariales en la empresa MegaProfer, manteniendo información actualizada en tiempo real, beneficiando a la toma de decisiones y llevando un óptimo control de ventas y publicación de productos.

Para la realización de la presente investigación se ha considerado utilizar las siguientes modalidades:

Modalidad Bibliográfica o Documental: Esta modalidad se debe aplicar porque en la investigación se recolecta información teórica de libros, libros digitales, artículos científicos, publicaciones, entre otros.

Modalidad de Campo: Esta también es aplicada en la investigación porque el investigador asistió directamente a la institución para recolectar información acerca de sus necesidades de información.

Se utilizó la metodología de programación XP debido que es un método ágil y permite al sistema adaptarse a cambios no previstos durante la etapa del diseño. Esta metodología es la utilizada por AlquimiaSoft S.A y requerida para el desarrollo de todos los proyectos planteados por la empresa.

3.2. Población y muestra

La población y muestra no son necesarios por ser la población demasiado pequeña y el tipo de investigación utilizada en el proyecto.

3.3. Recolección de Información

Para la investigación y desarrollo de la investigación se realizó una recolección de información directamente con los miembros que administran los sistemas de la empresa MegaProfer. Además de obtener información directamente de los esquemas de bases de datos relacionales que administran en base a campos requeridos, para esto se realizó una entrevista con cada uno de los operarios encargados del registro de la información, en la cual se recolectó datos muy relevantes e importantes que ayudaron a dar inicio a la investigación.

3.4. Procesamiento y Análisis de Datos

Una vez que el proceso de recolección de información concluyó, se aplicó los siguientes procedimientos:

- Identificar los campos relacionados a productos, categorías, clientes, direcciones en el sistema de comercio electrónico y ERP.
- Identificar el proceso de órdenes del sistema de comercio electrónico y datos relacionados con el mismo.
- Análisis de sintaxis de envío de información a través del sistema de mensajería JMS.
- Realizar una entrevista a los miembros de MegraProfer para determinar el tiempo de sincronización de datos.

3.4. Desarrollo del proyecto

Para el cumplimiento del desarrollo de este proyecto de investigación, se procedió en forma secuencial los siguientes métodos, procesos y/o actividades en base a la metodología de desarrollo ágil XP, que es la que actualmente maneja AlquimiaSoft S.A.

debido que su personal se encuentra capacitado para el uso de la misma y por lo tanto es impuesta para todo proyecto de desarrollo de software en la empresa:

- Fase 1. Exploración
 - Análisis del proceso de ingreso de información al sistema de comercio electrónico y ERP.
- Fase 2. Planificación del proyecto.
 - Recolección de historias de usuario del sistema Atix ERP.
 - Estimación de tiempo requerido para el desarrollo.
- Fase 3. Iteraciones
 - Definición del plan de entrega del módulo solicitado.
- Fase 4. Producción
 - Identificación de los datos requeridos en ambos sistemas para obtener información completa y precisa.
 - Diseño del modelo de sincronización.
 - Diseño y desarrollo de la aplicación, sentencias SQL para selección, inserción, actualización de datos.
 - Determinar y aplicar un glosario de términos para la determinación de métodos y clases para facilitar el mantenimiento del sistema
- Fase 5. Pruebas
 - Implementación de la aplicación.
 - Realización de pruebas caja blanca y caja negra.

CAPÍTULO 4

DESARROLLO DE LA PROPUESTA

Para el desarrollo del presente proyecto se utilizó la metodología XP, debido a que la metodología está diseñada para aceptar cambios en cualquier etapa de desarrollo y está orientada a la programación rápida, de calidad y es requerida por Alquimiasoft S.A.

4.1. Fase I: Exploración

En esta primera fase del proyecto se pretende realizar el análisis de los procesos que intervienen para mantener la tienda virtual y Atix ERP actualizados. Donde por medio de preguntas al usuario encargado, se obtendrá la información necesaria para automatizar el proceso de sincronización y ajustarlo a las necesidades de la empresa.

De acuerdo a las preguntas realizadas la información a ser utilizada para la sincronización es: Categorías, productos y pedidos. Esto bajo criterios establecidos, ya que la información de productos no puede ser mostrada en el momento de la sincronización sino a la hora establecida por el usuario para la sincronización.

A su vez la información de los pedidos realizados en la tienda virtual tiene que ser leída en el preciso momento que el usuario lo solicite, debido a requerimientos para procesos de facturación y una mejor interacción usuario-sistema.

Toda esta información debe mantener su integridad, ya que un dato erróneo puede provocar varios conflictos tanto para la empresa como para el usuario.

A su vez se debe evitar la duplicidad en los datos de pedidos leídos.

Entrevista dirigida a usuarios encargados de mantener sincronizados la tienda virtual y Atix ERP en MegaProfer S.A.	
Preguntas	Respuestas
¿Qué información requiere que se muestre en la tienda virtual?	Principalmente la información de los productos que se posee, menos el stock de cada producto. Esto también incluye a las categorías, ya que ayudan a la búsqueda por el menú de la tienda virtual.
¿Qué información requiere insertar en Atix ERP que se encuentre en la tienda virtual?	Los pedidos nuevos, esto es importante para que se pueda realizar los cobros o despachos de los productos solicitados.
¿Los procesos de sincronización deben ser inmediatos o deben esperar cierto tiempo para ser mostrados?	La información de los productos no se debe mostrar de manera inmediata, sería preferible establecer una hora exacta para que se muestren los nuevos valores. En cuanto a pedidos, estos si deben ser leídos de manera inmediata cuando se lo solicite.
¿Qué tan frecuente es el proceso de sincronización entre los sistemas?	Todos los días se actualizan precios de productos y se leen pedidos para que sean ingresados, este proceso manual lleva demasiado tiempo.
¿Qué grado de importancia tiene la información que se ingresa en ambos sistemas?	La información que se maneja en ambos sistemas es muy importante y valiosa para la empresa, un error puede provocar la pérdida de clientes.

Tabla 1. Cuadro de preguntas a usuarios encargados de la sincronización manual.

Elaborado por: El Investigador

4.2. Fase II: Planificación del proyecto

4.2.1. Historias de Usuario

En esta sección se definen las historias de los usuarios, en este caso tomadas directamente de usuarios del sistema ATIX ERP. Estas historias no requieren un lenguaje técnico, se emplea un lenguaje natural que exprese la necesidad del usuario en pocas palabras.

A la vez se estima un tiempo para el desarrollo y la prioridad que la historia de usuario posee. Estas historias no son complejas ya que a nivel de usuario el proceso de sincronizar datos en un proceso simple, esto tiene como resultado que no exista un listado extenso de historias.

A continuación, se establecerá la plantilla con la cual se recolectarán las historias de los usuarios:

Historia de Usuario	
Número:	Nombre:
Usuario:	
Puntos Estimados:	Iteración Asignada:
Prioridad en Negocio (Alta/Media/Baja):	Riesgo en Desarrollo (Alto/Medio/Bajo):
Descripción:	
Observaciones:	

Tabla 2. Plantilla de historia de Usuario.

Elaborado por: El Investigador

Esta plantilla consta de los siguientes elementos:

- **Número:**
Es el identificador numérico de la historia del usuario.
- **Nombre:**
Es el título que posee la historia del usuario, está relacionada con el caso descrito.
- **Usuario:**
Es la persona que relata la historia.

- **Puntos estimados:**
Es el tiempo estimado en días para concluir la historia, se trabaja solo con días enteros, si una historia tiene un día y medio inmediatamente se aproxima a dos días.
- **Iteración Asignada:**
Es el número de iteración en que se establece la historia de usuario.
- **Prioridad en Negocio (Alta/Media/Baja):**
Se le asigna una calificación a la prioridad en base a la necesidad del negocio.
- **Riesgo en Desarrollo (Alto/Medio/Bajo):**
Se establece el valor del riesgo al momento de desarrollar dicha historia.
- **Descripción:**
En esta sección va la historia del usuario en sus propias palabras. Estableciendo el que hacer y no el cómo hacer.
- **Observaciones:**
Se detalla los procesos que tienen relación con dicha historia de usuario.

A continuación, se muestran las historias de usuarios, recolectadas con los encargados del manejo de ingreso de pedidos a Atix ERP y de igual manera con el ingreso de productos a Prestashop *e-commerce*.

Historia de Usuario	
Número: 1	Nombre: Envío de Información.
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Primera
Prioridad en Negocio (Alta/Media/Baja): Alta	Riesgo en Desarrollo (Alto/Medio/Bajo): Alto
Descripción: Como persona encargada de ingresar productos en la tienda virtual deseo seleccionar los productos en Atix que van a estar disponibles y que estos aparezcan, no de manera inmediata, sino a una hora específica, ya que se puede lanzar promociones y quisiera que estas se muestren al siguiente día. Lo más importante del producto es su código, nombre, categoría, precio e imagen.	
Observaciones: La prioridad del negocio es alta, debido a que la publicación oportuna de sus productos puede incrementar sus ventas. El riesgo de desarrollo es Alto, debido que se debe mantener la integridad de la información y la correcta publicación en la tienda web en el tiempo establecido.	

Tabla 3. Historia de Usuario: Envío de Información.

Elaborada por el investigador

Historia de Usuario	
Número:2	Nombre: Actualización de precios
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Primera
Prioridad en Negocio (Alta/Media/Baja): Alta	Riesgo en Desarrollo (Alto/Medio/Bajo): Alto
Descripción: Los cambios en los precios son constantes, por lo tanto hay que tener la tienda virtual con los mismos precios del Atix, quisiera que estos cambios sean automáticos, pero no puede ser en el preciso momento, ya que puede dañar una compra potencial. Quiera que los cambios de precios sean reflejados el día siguiente.	
Observaciones: Prioridad del negocio es alta, debido que los cambios en los precios puede provocar inconvenientes con los compradores y causar una caída en ventas. El riesgo de desarrollo es Alto, debido que el precio debe ser el correcto y estar siempre actualizado.	

Tabla 4. Historia de Usuario: Actualización de precios.

Elaborada por el Investigador

Historia de Usuario	
Número: 3	Nombre: Ingreso de pedidos
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Segunda
Prioridad en Negocio (Alta/Media/Baja): Alta	Riesgo en Desarrollo (Alto/Medio/Bajo): Alta
Descripción: Actualmente los pedidos son ingresados manualmente al Atix, quisiera que con solo darle la orden al sistema todos los pedidos pendientes se inserten y pasen a su respectivo departamento en caso que sea necesario una aprobación por crédito o por stock.	
Observaciones: La prioridad del negocio es alta debido que el ingreso de pedidos de clientes es fundamental en el negocio. El riesgo de desarrollo es alto, ya que se debe brindar la información correcta en el momento solicitado por el usuario.	

Tabla 5. Historia de Usuario: Ingreso de Pedidos.

Elaborado por el Investigador

Historia de Usuario	
Número: 4	Nombre: Cantidad individual de productos
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Primera
Prioridad en Negocio (Alta/Media/Baja): Baja	Riesgo en Desarrollo (Alto/Medio/Bajo): Baja
Descripción: No deseamos que el producto sea enviado con su cantidad real de stock ya que esto nos podría ocasionar problemas. Por lo tanto todos los productos en la tienda virtual tendrían un stock ilimitado.	
Observaciones: La prioridad del negocio es baja, ya que no es una información que se deba publicar. El riesgo de desarrollo es bajo, debido que el sistema no deberá considerar la cantidad de productos.	

Tabla 6. Historia de Usuario: Cantidad individual de productos.

Elaborada por el Investigador

Historia de Usuario	
Número: 5	Nombre: Control de Clientes
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Segunda
Prioridad en Negocio (Alta/Media/Baja): Alta	Riesgo en Desarrollo (Alto/Medio/Bajo): Medio
Descripción: Los pedidos deben estar relacionados a un cliente que ya exista en nuestro ERP, para esto nosotros solicitamos la información directamente, así tenemos información valida y real. En caso que el cliente no exista en Atix, el pedido de este cliente será omitido.	
Observaciones: La prioridad del negocio es alta debido que la información de clientes de la tienda virtual debe concordar con Atix ERP. El riesgo de desarrollo es Medio, debido que todos los pedidos en la tienda virtual tiene obligatoriamente un cliente relacionado.	

Tabla 7. Historia de Usuario: Ingreso de clientes nuevos.

Elaborado por el Investigador.

Historia de Usuario	
Número: 6	Nombre: Ingreso de Categorías
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Primera
Prioridad en Negocio (Alta/Media/Baja): Baja	Riesgo en Desarrollo (Alto/Medio/Bajo): Baja
Descripción: Deseo que las categorías de los productos se inserten automáticamente ya que actualmente tengo que crear la categoría y luego el producto. Actualmente las categorías no están bien definidas, el nombre de la categoría no es actualizado, por lo tanto solo se requiere enviar las nuevas categorías. Quisiera tener todas las que tengo en Atix.	
Observaciones: La prioridad del negocio es baja, debido que la categoría es solo un dato para la organización de la información, los clientes tienden a buscar directamente el producto por su nombre. El riesgo en desarrollo es bajo, debido que si no se encuentra la categoría del producto en la tienda, este será asignado a una categoría por defecto, esto no afectara al producto.	

Tabla 8. Historia de Usuario: Ingreso de Categorías.

Elaborado por el Investigador

Historia de Usuario	
Número: 7	Nombre: Actualización de nombres de productos
Usuario: Mauricio Guato	
Puntos Estimados: NA	Iteración Asignada: Tercera
Prioridad en Negocio (Alta/Media/Baja): Media	Riesgo en Desarrollo (Alto/Medio/Bajo): Media
Descripción: En ocasiones los nombre de los productos son modificados para hacerlos más comprensibles, o también se añade una descripción. Quiera que esto también sea reflejado en la tienda virtual.	
Observaciones: La prioridad del negocio es media, ya que la búsqueda en la tienda virtual también se puede realizar por códigos de referencia de productos. El riesgo de desarrollo es medio, debido que son datos secundarios de un producto.	

Tabla 9. Historia de Usuario: Actualización de nombres de productos.

Elaborado por el Investigador

4.2.2. Estimación de Historias de Usuario

Al ser un sistema de sincronización se ha realizado la estimación en módulos dependientes, que van ser desarrollados dentro de Atix ERP y en el servidor de la tienda electrónica. Estos cumplirán las funciones detalladas previamente.

Para la estimación en días, se consideró un trabajo de 4 horas por día en el desarrollo de la historia de usuario.

Módulos relacionados a productos

Modulo	Nº	Historia del usuario	Tiempo Estimado	
			Días Estimados	Horas Estimadas
Productos	1	Envío de Información	4	16
	2	Actualización de precios	3	10
	4	Cantidad individual de productos	2	8
	6	Ingreso de Categorías	3	10
	7	Actualización de nombres de productos	2	8
Tiempo Total Estimado:			14	52

Tabla 10. Estimación de historias de Usuario.

Módulo de productos.

Elaborado por el Investigador.

Módulos relacionados a clientes

Modulo	Nº	Historia del usuario	Tiempo Estimado	
			Días Estimados	Horas Estimadas
Cientes	5	Control de Clientes nuevos	4	16
Tiempo Total Estimado:			4	16

Tabla 11. Estimación de historias de Usuario.

Módulo de Clientes.

Elaborado por el Investigador

Módulos relacionados a pedidos

Modulo	Nº	Historia del usuario	Tiempo Estimado	
			Días Estimados	Horas Estimadas
Pedidos	3	Ingreso de pedidos	10	40
Tiempo Total Estimado:			10	40

Tabla 12. Estimación de historias de Usuario.

Módulo de pedidos.

Elaborado por el Investigador.

4.3. Fase III: Iteraciones.

4.3.1 Plan de Entrega.

En la fase de implementación se planteará la presentación de un plan de entrega del producto ajustado a las funciones de los módulos, para la correcta comprensión de la actual fase de desarrollo.

Módulo	Nº	Historia de Usuario	Tiempo Estimado		Interacción Asignada			Entrega Asignada		
			Días Estimados	Horas estimadas	1	2	3	1	2	3
Productos	1	Envió de información	4	16	X			X		
	2	Actualización de precios	3	10	X			X		
	4	Cantidad individual de productos	2	8	X			X		
	6	Ingreso de categorías	3	10	X			X		
	7	Actualización de nombre de productos	2	8			X			X
Clientes	5	Ingreso de clientes	4	16		X			X	
Pedidos	3	Ingreso de pedidos	10	40		X			X	

Tabla 13. Plan de Entrega.

Elaborado por: El Investigador

4.3.2 Iteraciones.

En la presente fase, cada historia de usuario es representada y estimada por el tiempo en desarrollo de cada una. Estos tiempos se plantean de acuerdo a la comunicación entre el

cliente final y desarrollador. Se definen en días de 4 horas de trabajo debido a disponibilidad de la empresa.

Iteración	Nº	Historia del Usuario	Prioridad (Entrega)	Actividad (Nueva - Ajuste)	Dependencia (Nº Historia Usuario)	Riesgo (Alto - Medio - Bajo)	Versión	Estado de desarrollo	Pruebas
Primera	1	Envío de Información	1	Nueva	NA	Alta	1	Completo	Aprobado
	2	Actualización de precios	1	Nueva	1	Alta	1	Completo	Aprobado
	4	Cantidad individual de productos	1	Nueva	NA	Baja	1	Completo	Aprobado
	6	Ingreso de categorías	1	Nueva	1	Baja	1	Completo	Aprobado
Segunda	5	Control de clientes	1	Nueva	1	Media	1	Completo	Aprobado
	3	Ingreso de pedidos	1	Nueva	NA	Alta	1	Completo	Aprobado
Tercera	7	Actualización de nombre de productos	1	Nueva	1	Media	1	Completo	Aprobado

Tabla 14. Histórico de versiones en Historias de Usuario.

Elaborado por: El Investigador

4.4. Fase IV: Producción.

4.4.1. Diseño del modelo de sincronización.

Tomando en cuenta que es un sistema de sincronización asíncrona se ha decidido dividir el sistema y tener una estructura en la cual la información pueda permanecer temporalmente antes de ser publicada.

Esto se logra gracias a un sistema de mensajería JMS que será el encargado de almacenar temporalmente la información, para que en el momento indicado esta sea leída e ingresada en la tienda virtual.

El sistema será dividido en dos componentes principales, el primero se encontrará con plug-in en Atix y el segundo como un JAR en el servidor de la tienda virtual, que se ejecutará como tarea del sistema operativo cada cierto tiempo.

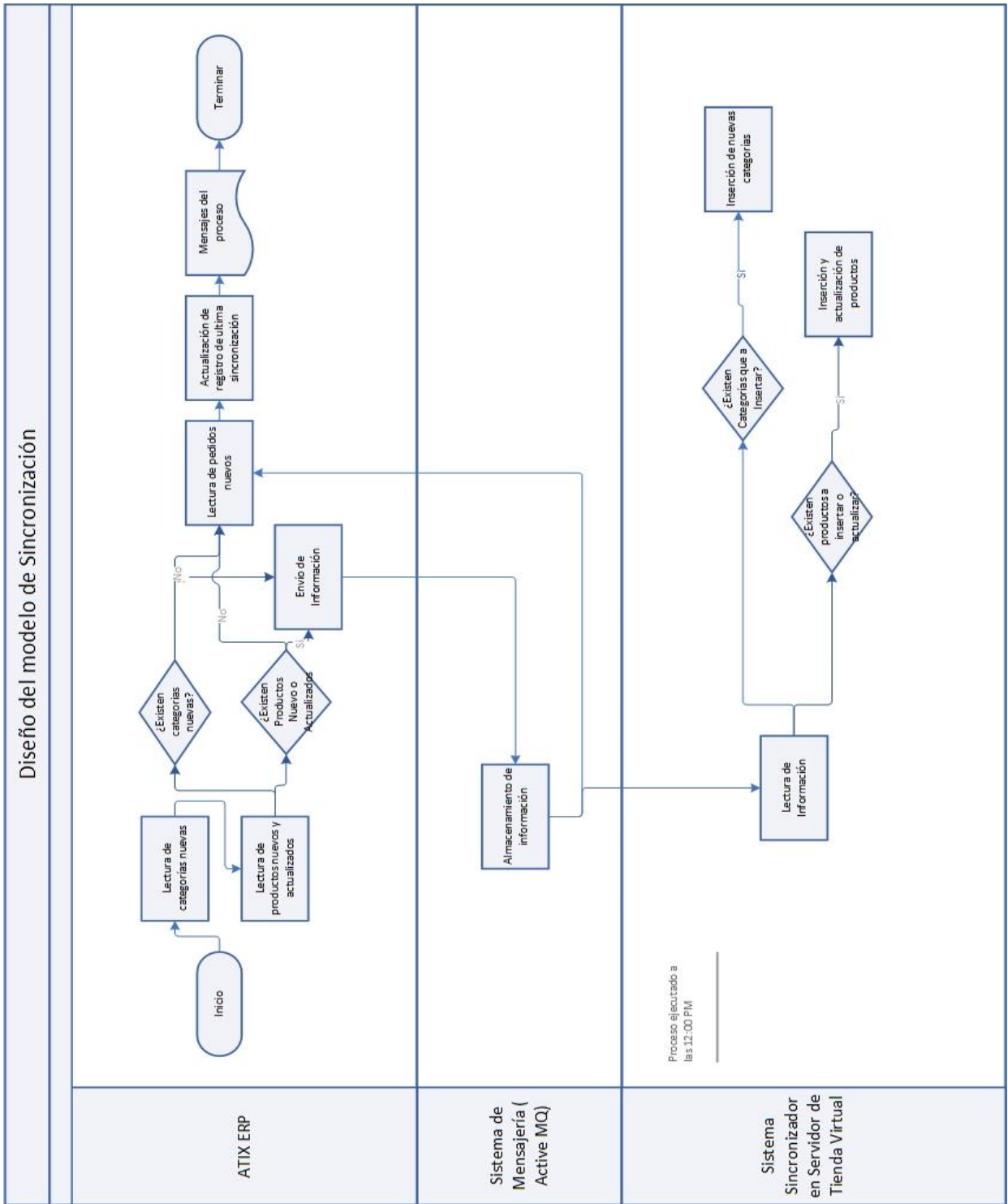


Imagen 2. Diseño del modelo de sincronización.

Elaborado por: El Investigador

Para poder realizar lo planteado, previamente se debe analizar la estructura de la base de datos, tanto como de Atix ERP y de la tienda virtual Prestashop. Todo esto para tener en cuenta de donde se va a tomar la información y donde va a ser almacenada.

4.4.2. Análisis de almacenamiento de datos de Atix ERP.

Para la comprensión del almacenamiento de información, se realizó una investigación al esquema de base de datos del sistema ATIX ERP y mediante el diseño de la tabla de productos, se ha podido conocer las columnas que contiene y como está relacionada con otras entidades. Al ser un sistema ERP, la cantidad de columnas sobrepasa a las requeridas por el sistema E-Comerce, por lo cual se detallará la información más relevante para el sistema de sincronización.

Nombre de la tabla: m_product

Descripción: Posee información de los productos.

Campo	Tipo	Descripción
m_product_id	numeric (10,0)	Clave primaria del producto
Isactive	character(1)	Producto activo o no
Created	Timestamp	Fecha de creación
Createdby	numeric (10,0)	Usuario que ingresó el producto
updated	Timestamp	Fecha de actualización
updatedby	numeric (10,0)	Usuario que actualizo el producto
Value	character varying(40)	Código de referencia del producto
Name	character varying(255)	Nombre
Description	character varying(255)	Descripción
c_uom_id	numeric (10,0)	Clave foránea, representa la unidad de medida
m_product_category_id	numeric (10,0)	Categoría a la cual pertenece
Weight	Numeric	Peso
c_taxcategory_id	numeric (10,0)	Clave foránea para el impuesto
Imageurl	character varying(120)	Enlace de la imagen del producto
Descriptionurl	character varying(120)	Descripción del enlace de la imagen

Tabla 15. Estructura tabla productos Atix ERP.

Elaborado por el Investigador

Tabla: c_taxcategory

Descripción: Almacena los porcentajes de impuestos.

Campo	Tipo	Descripción
c_taxcategory_id	numeric(10,0)	Clave primaria de la tabla
isactive	character(1)	Impuesto activo o no
Name	character varying(60)	Nombre de impuesto
description	character varying(255)	Descripción del impuesto

Tabla 16. Estructura tabla impuestos Atix ERP

Elaborado por el Investigador

Tabla: m_product_category

Descripción: Posee las categorías de los productos

Campo	Tipo	Descripción
m_product_category_id	numeric (10,0)	Clave primaria de la tabla
Isactive	character(1)	Categoría activa o no
Value	character varying(40)	Código de la categoría
Name	character varying(60)	Nombre de la categoría
description	character varying(255)	Descripción
Isdefault	character(1)	Si va por defecto o no
m_product_category_parent_id	numeric(10,0)	Padre de la categoría

Tabla 17. Estructura tabla categoría de productos Atix ERP

Elaborado por el Investigador

Tabla: c_uom

Descripción: Almacena la unidad de medida de los productos.

Campo	Tipo	Descripción
c_uom_id	numeric(10,0)	Clave primaria de la tabla
Isactive	character(1)	Si está activo o no
Uomsymbol	character varying(10)	Símbolo de la unidad
Name	character varying(60)	Nombre de la unidad
Description	character varying(255)	Descripción de la unidad

Tabla 18. Estructura de la tabla Unidad de medida

Atix ERP

Elaborado por el Investigador

Tabla: m_product_price

Descripción: Almacena la lista de precios de los productos.

Campo	Tipo	Descripción
m_pricelist_version_id	numeric(10,0)	Clave primaria de la tabla
m_product_id	numeric(10,0)	Clave foránea del código de producto
Isactive	character(1)	Precio activo o no
Created	Timestamp	Fecha de creación
pricelist	Numeric	Código de la lista de precio
pricestd	Numeric	Precio base
pricelimit	Numeric	Precio limite

Tabla 19. Estructura de la tabla Lista de Precios

Atix ERP

Elaborado por el Investigador

4.4.3. Análisis de almacenamiento de datos de tienda virtual Prestashop.

Para poder acceder al esquema de base de datos de Prestashop se puede proceder a la instalación y con el uso de herramientas acceder a la base, la cual está abierta a cambios por ser sistema de código abierto.

Este sistema distribuye la información del producto en diferentes tablas, esto se lo pudo deducir al ingresar productos por medio de su interfaz gráfica y al análisis de las relaciones de la tabla en la base de datos. Para este propósito se utiliza “Mysql workbench” software libre que proporciona herramientas para la gestión y administración visual de bases de datos hechas en MySQL Server.

El nombre de la tabla se antepone de un prefijo que se elige durante la instalación, debido que el mismo esquema puede tener diferentes sistemas Prestashop instalados. En este caso el prefijo es “ps_”.

De igual manera se procederá a listar las tablas más importantes que poseen la información requerida para poder sincronizar los sistemas.

Tabla: ps_product

Descripción: Almacena todas las propiedades principales de todos los productos de la tienda, entre los campos más importantes están el código de referencia del producto, código EAN (usado para código de barras), nombre, precio, activo.

Campo	Tipo	Descripción
id_product	Integer	Clave primaria del producto
id_supplier	Integer	Clave foránea de la identificación del proveedor
id_manufacturer	Integer	Clave foránea de la identificación del fabricante
id_category_default	Integer	Clave foránea de la categoría por defecto
id_shop_default	Integer	Clave foránea de la tienda que pertenece el producto
id_tax_rules_group	Integer	Clave foránea del impuesto del producto
ean13	varchar (13)	Código de barras EAN del producto
Upc	varchar(12)	Código de barras UPC del producto
Unity	varchar(255)	Unidad de venta
Price	decimal(20,6)	Precio unitario
supplier_reference	varchar(32)	Código de referencia del proveedor para el producto
Location	varchar(64)	Ubicación del producto
Active	Tinyint	Producto activo o no ('1' Si, '2' No)
redirect_type	Enum	Página para redireccionar el producto
Indexed	Tinyint	Producto Indexado o no('1' Si, '2' No)
cache_default_attribute	Integer	Atributo por defecto del cache, estandar '0'
date_add	Datetime	Fecha de ingreso
date_upd	Datetime	Fecha de actualización
Reference	varchar(32)	Código de referencia del producto

Tabla 20. Estructura de la tabla Producto

Prestashop

Elaborado por el Investigador

Tabla: ps_product_lang

Descripción: Esta tabla posee la descripción corta y larga de cada producto, la cual aparecerá en la tienda al seleccionar un producto, a su vez también posee los textos que facilitarían la exploración por medio de la herramienta integrada para búsqueda de productos.

Campo	Tipo	Descripción
id_product	integer	Clave foránea del producto
id_shop	integer	Clave foránea de la tienda
id_lang	integer	Clave primaria de la tabla
description_short	Text	Descripción corta del producto
link_rewrite	varchar(128)	Link sobre escrito para optimizar búsqueda
name	varchar(128)	Nombre del producto
description	Text	Descripción completa del producto
meta_description	varchar(255)	Descripción para optimizar la búsqueda de buscadores externos
meta_keywords	varchar(255)	Palabras clave para optimizar la búsqueda de buscadores
meta_title	varchar(128)	Título para optimizar la búsqueda
available_now	varchar(255)	Descripción del producto disponible
available_later	varchar(255)	Descripción de fecha/motivos de disponibilidad después

Tabla 21. Estructura de la tabla traducciones del producto

Prestashop

Elaborado por el Investigador

Tabla: ps_product_shop

Descripción: Debido a que soporta tiendas múltiples, en esta tabla se especifica a que tienda pertenece cada producto, puede pertenecer a un o varias tiendas.

Campo	Tipo	Descripción
id_product	Integer	Clave foránea del producto
id_shop	Integer	Clave primaria de la tabla
id_category_default	Integer	Clave foránea de la categoría por defecto
id_tax_rules_group	Integer	Clave foránea del impuesto
Price	decimal(20,6)	Precio del producto
Active	tiny int	Estado, activo o no ('1' Si, '0' No)
redirect_type	Emun	Página de redirección
Unity	varchar(255)	Unidad del producto
cache_default_attribute	Integer	Atributo por defecto del cache, estándar '0'
Indexed	tiny int	Producto Indexado o no('1' Si, '2' No)
date_add	Datetime	Fecha de ingreso
date_upd	Datetime	Fecha de actualización

Tabla 22. Estructura de la tabla Tienda del Producto

Prestashop

Elaborado por el Investigador

Tabla: ps_stock_available

Descripción: Almacena la cantidad disponible de cada producto, también el número mínimo de unidades, con la cual el sistema bloqueara la compra cuando la cantidad de producto llegue a este indicador. Este bloqueo es configurable, ya que se puede proceder con la facturación de productos en cantidades negativas.

Campo	Tipo	Descripción
id_stock_available	Integer	Clave primaria de la tabla
id_product	Integer	Clave foranea del producto
id_shop	Integer	Clave foranea de la tienda
Quantity	Integer	Cantidad de productos disponibles
out_of_stock	tiny int	Cantidad minima para que el producto este no disponible
id_shop_group	Integer	Clave foranea del grupo de la tienda que pertenece
id_product_attribute	Integer	Clave foranea de los atributos del producto

Tabla 23. Estructura de la Tabla Cantidades de Productos

Prestashop

Elaborado por el Investigador

Tabla: ps_category_product

Descripción: En esta tabla se asocia una categoría con un producto, también se asigna una posición en la cual asomara el producto en el listado. Esta tabla principalmente posee claves foráneas, ya que un producto puede pertenecer a varias categorías. Prestashop utiliza esto atributos para poder mejor las búsquedas dentro de la tienda o a su vez formar el menú para la presentación de productos en la página.

Campo	Tipo	Descripción
id_category	Integer	Clave foránea de la categoría
id_product	Integer	Clave foránea del producto
Position	Integer	Posición de la categoría

Tabla 24. Estructura de la tabla categoría del producto

Prestashop

Elaborado por el Investigador

Tabla: ps_category

Descripción: Posee todas las categorías de los productos, estas categorías deben pertenecer a un padre, esto ayuda a formar un árbol, en caso que la categoría sea la principal, esta tendrá de padre al “Inicio”.

Campo	Tipo	Descripción
id_category	Integer	Clave primaria de la tabla
id_parent	Integer	Clave foránea del padre de la categoría
id_shop_default	Integer	Clave foránea de la tienda
level_depth	tiny int	Nivel de profundidad para el árbol de categorías
active	tiny int	Categoría activa o no ('1' Si, '2' No)
date_add	date time	Fecha de inserción
date_upd	date time	Fecha de actualización
position	Integer	Posición de la categoría
is_root_category	tiny int	Categoría Principal ('1' Si, '2' No)

Tabla 25. Estructura de la tabla Categorías

Prestashop

Elaborado por el Investigador

Tabla: ps_category_lang

Descripción: De igual manera de los productos, cada categoría debe tener su descripción corta y larga, también un texto de enlace para ayudar en las búsquedas.

Campo	Tipo	Descripción
id_category	Integer	Clave foránea de la categoría
id_shop	Integer	Clave foránea de la tienda
id_lang	Integer	Clave primaria de la tabla
Name	varchar(128)	Nombre de la categoría
description	Text	Descripción
link_rewrite	varchar(128)	Enlace de la categoría

Tabla 26. Estructura de la tabla Traducciones de Categoría

Prestashop

Elaborado por el Investigador

Tabla: ps_category_shop

Descripción: Define a que tienda pertenece la categoría.

Campo	Tipo	Descripción
id_category	Integer	Clave foránea de la categoría
id_shop	Integer	Clave foránea de la tienda
position	Integer	Posición

Tabla 27. Estructura de la tabla Categorías de la Tienda

Prestashop

Elaborado por el Investigador

Tabla: ps_category_group

Descripción: Las categorías pueden ser asociadas a grupos de clientes, en estos tenemos por ejemplo a “Visitantes”, “Registrados” y “Exclusivos”.

Campo	Tipo	Descripción
id_category	Integer	Clave foránea de la categoría
id_group	Integer	Clave foránea del grupo

Tabla 28. Estructura de la tabla Grupo de Categoría

Prestashop

Elaborado por el Investigador

Tabla: ps_customer

Descripción: Almacena la información de todos los clientes registrados, como nombre, teléfono, sexo, fecha de nacimiento.

Campo	Tipo	Descripción
id_customer	Integer	Clave primaria de la tabla
firstname	vachar(32)	Nombres
lastname	varchar(32)	Apellidos
email	varchar(128)	Correo electrónico
birthday	Date	Fecha de nacimiento

Tabla 29. Estructura de la tabla Cliente

Prestashop

Elaborado por el Investigador

Tabla: ps_address

Descripción: Debido que los clientes pueden tener diferentes direcciones, existe esta tabla para almacenar todas las direcciones que registren los clientes.

Campo	Tipo	Descripción
id_address	Integer	Clave primaria de la tabla
id_customer	Integer	Clave foránea del cliente
company	varchar(64)	Compañía
address1	varchar(128)	Dirección 1
city	varchar(64)	Ciudad
phone	varchar(32)	Teléfono fijo
phone_mobile	varchar(32)	Teléfono móvil

Tabla 30. Estructura de la tabla Direcciones de Clientes

Elaborado por el Investigador

Tabla: ps_orders

Descripción: Guarda todas las ordenes hechas por los clientes, aquí se encuentra la cantidad de artículos, total a pagar, impuestos y el costo del envío en caso de ser necesario.

Campo	Tipo	Descripción
id_order	Integer	Clave primaria de la tabla
id_customer	Integer	Clave foránea del cliente
id_address_delivery	Integer	Clave foránea de la dirección
date_add	Datetime	Fecha que se ingresó el pedido
Payment	varchar(255)	Forma de pago
total_products	decimal(20,6)	Valor total del pedido sin impuestos
total_products_wt	decimal(20,6)	Valor total del pedido con impuestos

Tabla 31. Estructura de la tabla Ordenes de Venta

Prestashop

Elaborado por el Investigador

Tabla: ps_order_detail

Descripción: Tiene el detalle de cada orden, donde están descritos los productos y las cantidades solicitadas.

Campo	Tipo	Descripción
id_order_detail	Integer	Clave primaria de la tabla
id_order	Integer	Clave foránea de la orden
product_id	Integer	Clave foránea del producto
product_name	varchar(255)	Nombre del producto
product_quantity	Integer	Cantidad
product_reference	varchar(32)	Código de referencia del producto
unit_price_tax_excl	decimal(20,6)	Precio unitario sin impuestos
unit_price_tax_incl	decimal(20,6)	Precio unitario con impuestos
total_price_tax_excl	decimal(20,6)	Precio total sin impuestos
total_price_tax_incl	decimal(20,6)	Precio total con impuestos

Tabla 32. Estructura de la tabla Detalle de Órdenes de Venta Prestashop

Elaborado por el Investigador

4.4.4. Diseño del modelo de sincronización.

Por ser dos sistemas diferentes y poseer requerimientos de una sincronización asincrónica, se decidió realizar dos módulos, uno como Plug-in en Atix ERP y otro un JAR en el lado de la tienda virtual. Estos dos sistemas tomarán como intermediario a un sistema de mensajería JMS que será el encargado de almacenar temporalmente la información, al

emplear esta herramienta se logra rapidez en la sincronización, ya que la forma tradicional sería por servicios web, pero este proceso es costoso en tiempo y memoria del servidor. A continuación, se detalla el proceso de sincronización y las funciones de cada módulo.

1. Plug-in en Atix ERP.

Funciones:

- Leer categorías y productos nuevos con la condición que estos sean destinados a mostrarse en la tienda virtual y enviarlos al sistema de mensajería JMS.
- Leer productos y precios actualizados y enviarlos.
- Seleccionar un solo producto para poder enviarlo.
- Funciones lectura de pedidos.
 - Leer clientes que posean pedidos, si no existe crearlo.
 - Leer pedidos que sean nuevos.
 - Actualizar historial de sincronizaciones para control de lectura.

2. JAR en servidor de la tienda virtual.

Funciones:

- Leer productos y categorías del sistema de mensajería JMS e insertarlos en la tienda virtual por servicios web.
- Guardar historial de sincronización.

El sistema de mensajería JMS solo será el encargado de recibir información y mantenerla disponible en todo momento hasta que la información sea solicitada.

4.4.5. Desarrollo de Plug-In en Atix ERP.

Es necesario que el módulo permita manejar al usuario el proceso de sincronización, de forma fácil y natural.

En Atix ERP primeramente se crea un proceso, por medio del sistema a través de un rol de usuario de administrador de sistema. En este indicamos los parámetros que tendrá, se asigna los nombres de los parámetros y del proceso, junto con las traducciones de los campos tomando en cuenta que es un sistema multilingüaje.

El proceso constará de los siguientes parámetros:

- **sync_actualizar_info:** Este parámetro es creado como tipo booleano, cuando su valor sea verdadero se procederá a enviar información actualizada de todos los productos.
- **sync_productos_webshop:** Este parámetro posee el código del producto, es un tipo de parámetro que permite desplegar un cuadro de búsqueda y seleccionar el producto a sincronizar, en caso de que este se encuentre vacío, se procederá a enviar todos los productos que no se encuentren en el sistema de comercio electrónico.
- **sync_pedidos_webshop:** Es un parámetro de tipo booleano, el cual dará la orden de leer e insertar los clientes y pedidos ingresados que se encuentran pendientes.

Atix ERP interpreta los valores de los parámetros como componentes visuales, por lo cual la interfaz es automáticamente creada, la cual quedaría de la siguiente manera:

Pantalla principal:

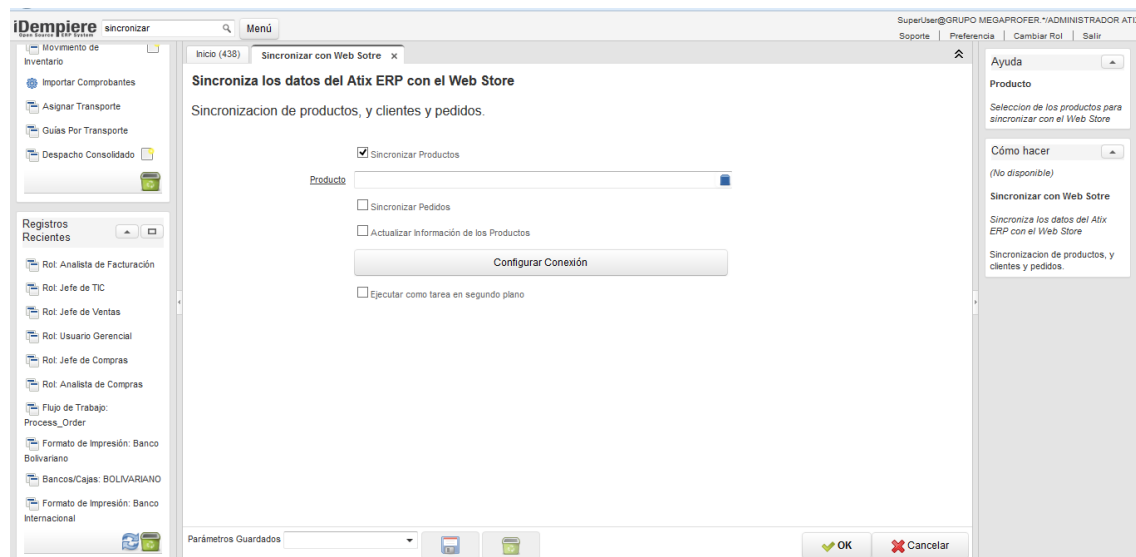


Imagen 3. Pantalla principal del módulo de sincronización.

Elaborado por el Investigador

Pantalla de búsqueda de producto:

Almacén	Descripción	Sustituto	Producto Relacionado	Disponible para Promesa	Precio
CAD AMBATO				15.914,00	15.914,00
CAD PRO 1				6,00	6,00
Σ				15.920,00	15.920,00

Imagen 4. Pantalla de búsqueda de productos del módulo de sincronización.

Elaborado por el Investigador

Por defecto todos los procesos en Atix ERP pueden ser ejecutados en segundo plano y notificar al usuario, en el panel principal o por correo electrónico, cuando la tarea haya sido terminada.

Ejecutar como tarea en segundo plano

Tipo de Notificación: **Email**

- Aviso
- Email
- Email+Aviso

Imagen 5. Ventana de ejecución de proceso en segundo plano.

Atix ERP

4.4.5.1 Lectura de categorías en Atix ERP.

Como se mencionó anteriormente Atix ERP almacena los datos en diferentes tablas para brindar una total flexibilidad a las características y una rápida adaptabilidad al negocio.

Primeramente, se requiere enviar la categoría del producto, esta es importante para una correcta organización de información y formación del menú en la tienda electrónica Prestashop.

Tomando en cuenta que las categorías pueden ser padres o hijos se realizó la sentencia SQL para leer la información de manera correcta, primeramente, se toma en cuenta a los padres y luego a los hijos. También se procede a estructurar los datos, quitando espacios innecesarios y enviándolos tal y como los requiere la tienda virtual Prestashop.

Toda la información tomada en la sentencia SQL será enviada, se delimitará cada campo con el carácter ‘>’ que no es utilizado para nombres o códigos en idempiere y que se encuentra bloqueado su uso en el sistema.

Sentencia SQL:

```
Select c.m_product_category_id,c.name,description ,lower(replace(replace(trim(name), '_',' '), ' ','-')) as link,
(Select name from adempiere.m_product_category cp where c.m_product_category_parent_id = cp.m_product_category_id) as parent
From adempiere.m_product_category c
Order by c.m_product_category_id asc;
```

Imagen 6. Sentencia SQL para la consulta de Categorías.

Elaborador por: El Investigador

En la cual se selecciona la clave del producto, nombres y su padre. En ATIX ERP, los nombres de las categorías se encuentran formadas por sus padres, es decir el nombre de una categoría sería, por ejemplo: Taladros/Bosh. Para una mejor presentación de los datos, solo se utilizará el nombre final, es por eso que se hace un “juego” con la información para solo mantener el nombre final de la categoría, ya que por referenciación se sabe a qué categoría padre pertenece.

Al final se forma un link, en caso de que nombre de la categoría tenga espacios, por ejemplo el link de la categoría: “Bosh portátil” sería: “Bosh-portatil”. Esto se hace debido que la tienda de comercio electrónico utiliza la búsqueda por medio del link y no por el nombre de la categoría.

4.4.5.2 Lectura de productos en Atix ERP.

Los productos están formados por código referencial, nombre, categoría, impuesto, imagen y una lista de precios.

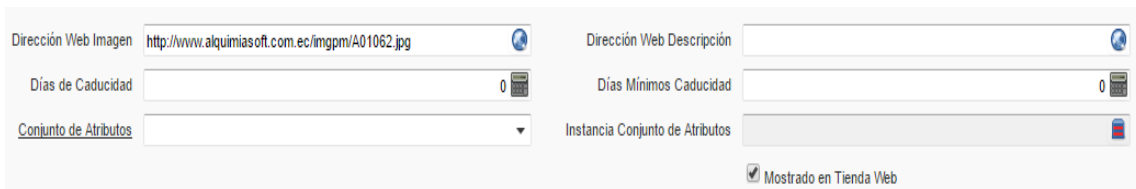
La imagen es una URL, debido a que almacenar una dirección de texto es más rápido que almacenar un dato binario en la base. Las imágenes se encuentran alojadas en un servidor externo de lectura pública.

La lista es definida previamente por los usuarios encargados del sistema, en este caso se va a utilizar el último precio establecido para ser enviado a la tienda virtual.

Para la selección de productos, se procede a su lectura, tomando en cuenta tres criterios:

1. El producto este seleccionado para aparecer en la tienda.
2. El producto ya haya sido sincronizado
3. El producto ya se encuentre sincronizado y sus datos hayan sido actualizados.

Para el primer criterio dentro de los atributos del Producto se encuentra un campo a nivel de interfaz de usuario que se llama “Mostrado en Tienda Web”, este a nivel de base de datos es nombrado: “*isselfservice*”. Cuando este se encuentre activo será tomado en cuenta para la sincronización.



The screenshot shows a web interface for product management. It features several input fields and a checkbox. The fields include: 'Dirección Web Imagen' with the value 'http://www.alquimiasoft.com.ec/imgpm/A01062.jpg', 'Dirección Web Descripción' (empty), 'Días de Caducidad' with the value '0', 'Días Mínimos Caducidad' with the value '0', 'Conjunto de Atributos' (dropdown menu), and 'Instancia Conjunto de Atributos' (dropdown menu). At the bottom right, there is a checkbox labeled 'Mostrado en Tienda Web' which is checked.

Imagen 7. Ventana de Producto Atix ERP.

Campo Mostrado en Tienda Web.

Elaborado por el Investigador.

Una vez que todos los productos hayan sido leídos y enviados para su sincronización, se procede a actualizar el campo “was_syn” que fue creado con el propósito de almacenar un dato *boolean* donde ‘N’ significa que no ha sido actualizado y ‘Y’ que ya se procedió con su envío.

Para el caso dos y tres se toma en cuenta que el producto ya haya sido sincronizado y que el campo creado en el diccionario de datos llamado 'was_upd' que es de tipo booleano este en estado 'Y'.

En Atix ERP, para detectar que un producto haya sido actualizado, se tomó en cuenta si el precio, nombre, categoría, descripción o imagen varió.

En caso de que el precio haya sido actualizado, existe una clase dentro del Plug-in de sincronización que se encarga de detectar si hay un cambio en la lista de precios y actualizar el campo en la tabla 'm_product', el código es el siguiente:

```
import org.compiere.model.MProduct;

public class EPriceUpdate extends CustomEventHandler {

    @Override
    protected void doHandleEvent() {
        MProductPrice mproductprice = (MProductPrice)getPO();
        MProduct product =MProduct.get(mproductprice.getCtx(), mproductprice.getM_Product_ID());
        product.setValueOfColumn("was_upd", true);
        product.save();
    }
}
```

Imagen 8. Codificación para la detección de actualización de precio.

Elaborado por: El investigador

Esta clase extiende de un Evento Personalizado, el cual detecta cambios, en el evento se especifica la tabla (entidad) de la cual se desea obtener la acción y a continuación se instancia la clase m_producto para proceder con la actualización del campo 'was_upd'.

Al momento de detectar cambios en el nombre del producto, se procede de diferente manera, ya que si se utiliza el código anterior orientado a la tabla "m_product" producirá un bucle infinito, ya que detectaría un cambio, actualizaría el campo y esa actualización sería un cambio con la cual se enviaría a actualizar nuevamente al campo y de esta forma indefinidamente.

Por lo tanto, se procedió a realizar una condición que comprobará si el campo ya fue actualizado. La primera vez pasará dos veces por la condición, la segunda vez si ya está actualizado no actualizará nada.

```
import org.compiere.model.MProduct;

import ec.com.alquimiasoft.base.CustomEventHandler;

public class EProductUpdate extends CustomEventHandler {

    @Override
    protected void doHandleEvent() {

        MProduct product = (MProduct) getPO();
        boolean status = product._getValueAsBoolean("was_upd");
        if(status != true){
            product._setValueOfColumn("was_upd", true);
            product.save();
        }

    }

}
```

Imagen 9. Codificación para la detección de cambios en los atributos del producto.

Elaborado por: El Investigador

De igual manera se instancia la clase producto, se obtiene el valor del campo “was_upd” para proceder a verificar si está activo el campo, si no lo está, se procede a la actualización y al guardado de la información.

Finalmente se procede a realizar la consulta SQL para obtener los productos, primeramente, de los nuevos, seguido por los actualizados. De igual manera delimitado cada campo por el carácter ‘>’ para su separación.

Sentencia SQL productos nuevos:

```
SELECT m_product_id,p.value as id_item ,p.name as nombre,t.name as prod_grab_iva,
(Select pricelist
from adempiere.m_productprice where m_product_id = p.m_product_id
order by pricelist desc OFFSET 0 FETCH FIRST 1 ROWS ONLY ) as preciobase,
p.description as desc_alt, lower(Replace(p.name,' ','-')) as link,
c.name as m_product_category, p.imageurl
FROM adempiere.m_product p
INNER JOIN adempiere.m_product_category c on (p.m_product_category_id = c.m_product_category_id)
INNER JOIN adempiere.c_taxcategory t on(p.c_taxcategory_id = t.c_taxcategory_id)
WHERE p.isselfservice like 'Y' and p.was_syn like 'N';
```

Imagen 10. Sentencia SQL para selección de productos nuevos a enviar.

Elaborado por: El Investigador

Sentencia SQL productos actualizados:

```
SELECT m_product_id,name as nombre,
(Select pricelist
from adempiere.m_productprice
where m_product_id = p.m_product_id
order by pricelist desc OFFSET 0 FETCH FIRST 1 ROWS ONLY ) as preciobase,
description as desc_alt, lower(Replace(name,' ','-')) as link,
m_product_category_id
FROM adempiere.m_product p
WHERE p.was_upd like 'Y' and p.isselfservice like 'Y';
```

Imagen 11. Sentencia SQL para selección de productos actualizados a enviar.

Elaborado por: El Investigador

En caso de requerir enviar un solo producto se concatena al SQL de productos nuevos en la cláusula *where* la condición:

and m_product_id =

Ya que el parámetro que se especificó previamente, para el envío de un solo producto, retorna el identificador único del registro.

4.4.3.5.3 Envío de información al sistema de mensajería JMS.

Una vez que la información esté preparada para ser enviada, se procede a llamar a la clase *Send*, que tiene la conexión a la cola de mensajería:

```

public class Send {
    private ConnectionFactory factory = null;
    private Connection connection = null;
    private Session session = null;
    private Destination destination = null;
    private MessageProducer producer = null;
    private String cola="";
    private String broker="tcp://149.56.141.14:61616";
    public Send(String cola) {
        this.cola = cola;
    }
    public void sendMessage(ArrayList<String> data) {
        try {
            factory = new ActiveMQConnectionFactory(broker);
            connection = factory.createConnection();
            connection.start();
            session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
            destination = session.createQueue(cola);
            producer = session.createProducer(destination);
            TextMessage message = session.createTextMessage();
            String elemento;
            for(int i=0 ; i<data.size();i++){
                elemento=data.get(i).toString();
                message.setText(elemento);
                message.setJMSType("String");
                producer.send(message);
            }
            connection.close();
            connection=null;
        } catch (JMSException e) {
            e.printStackTrace();
        }
    }
}

```

Imagen 12. Codificación para el envío de información a la cola de mensajería.

Elaborado por: El Investigador

Donde se procede a instanciar todos los objetos y se asigna una dirección a la variable “*broker*”, que es la dirección del servidor de mensajería, junto con su puerto de conexión.

Esta clase se instancia con una variable tipo cadena de texto, ésta va a ser el nombre de la cola donde se almacenarán todos los datos relacionados. Principalmente existirán dos nombres de colas, que sería:

- M_Product: Almacena la información de productos.
- M_Product_Category: Almacena información de las categorías.

La clase posee el método “sendMessage”, que recibe un objeto de tipo ArrayList, esta poseerá toda la información que va a contener la cola de datos.

Una vez que la información sea enviada a la cola de mensajería, ésta tendrá toda la información disponible para ser leída por la tienda virtual.

Queues

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
M_Product	1469	0	1469	0	Browse Active Consumers Active Producers atom rsi	Send To Purge Delete
M_Product_Category	643	0	643	0	Browse Active Consumers Active Producers atom rsi	Send To Purge Delete

Imagen 13. Interfaz de contenedores de colas de mensajería.

Elaborado por: El Investigador

Se indica la pantalla de la cola de mensajería, donde muestra la cantidad de elementos que contiene cada cola e indica que ninguno ha sido leído aún.

Dentro de la cola “M_Product” se tiene:

Browse M_Product

Message ID ↑	Correlation ID	Persistence	Priority	Redelivered	Reply To	Timestamp	Type	Operations
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1		Persistent	4	false		2016-10-27 23:38:02:873 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:10		Persistent	4	false		2016-10-27 23:38:04:032 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:100		Persistent	4	false		2016-10-27 23:38:19:932 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1000		Persistent	4	false		2016-10-27 23:41:26:750 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1001		Persistent	4	false		2016-10-27 23:41:26:881 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1002		Persistent	4	false		2016-10-27 23:41:27:021 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1003		Persistent	4	false		2016-10-27 23:41:27:168 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1004		Persistent	4	false		2016-10-27 23:41:27:332 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1005		Persistent	4	false		2016-10-27 23:41:27:461 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1006		Persistent	4	false		2016-10-27 23:41:27:581 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1007		Persistent	4	false		2016-10-27 23:41:27:711 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1008		Persistent	4	false		2016-10-27 23:41:27:881 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1009		Persistent	4	false		2016-10-27 23:41:28:011 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:101		Persistent	4	false		2016-10-27 23:38:20:052 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1010		Persistent	4	false		2016-10-27 23:41:28:141 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1011		Persistent	4	false		2016-10-27 23:41:28:260 EDT	String	Delete
ID:Dev-Lap-58575-1477625671883-0:1:1:1:1012		Persistent	4	false		2016-10-27 23:41:28:381 EDT	String	Delete

Imagen 14. Interfaz de la cola de mensajería pendiente.

Elaborado por: El Investigador

La primera columna es un identificador único para cada mensaje, en el segundo se tiene la persistencia, que indica que estos datos persistirán hasta que sean leídos. También consta la prioridad del mensaje, la fecha del envío y el tipo de dato.

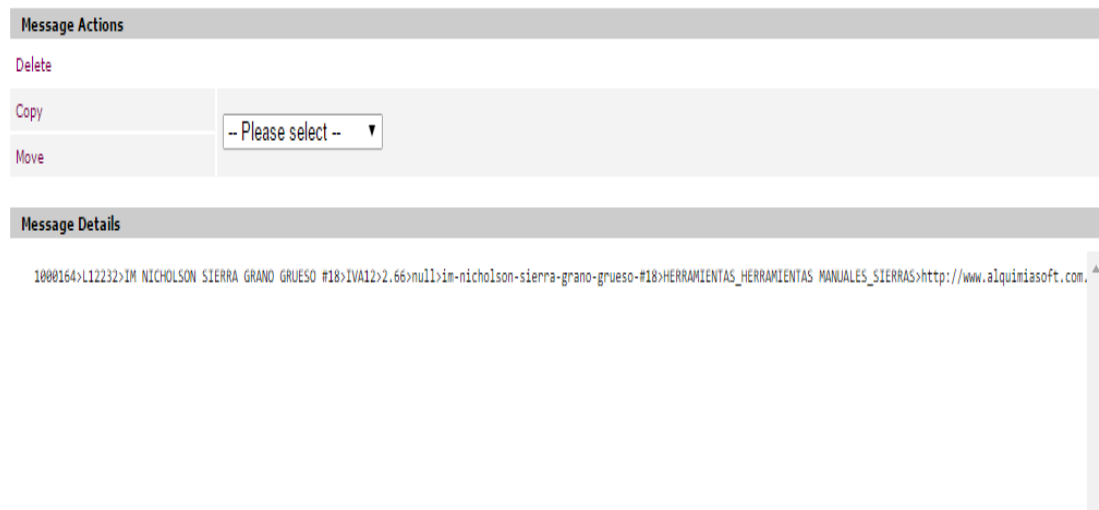


Imagen 15. Interfaz del contenido del mensaje de sistema de mensajería.

Elaborado por: El investigador

Dentro del mensaje se tiene el contenido, que fue previamente especificado en las sentencias de lectura. Se encuentra de la misma manera que fue enviada, delimitado entre campos por un carácter especial especificado previamente.

4.4.5.4 Lectura de Clientes y Pedidos.

Para este proceso se optó por utilizar directamente los servicios web de la tienda virtual Prestashop, ya que el requerimiento establece que los pedidos sean leídos en el momento solicitado. Para lo cual se empleó una clase que utiliza los métodos brindados por Prestashop para acceder a los servicios web, en esta solo basta con establecer la dirección IP y la clave de acceso proporcionada por la tienda misma.

Los campos necesarios para el Atix ERP en cuanto a un pedido son:

- Cliente
- Almacén

- Fecha
- Total Descuento
- Total IVA
- Total Factura
- Detalle de Factura
 - Código referencial del producto
 - Nombre del producto
 - Precio
 - Impuesto
 - Total

En cuanto al almacén se puede usar uno por defecto, ya que posteriormente habrá una decisión por parte del usuario encargado, sobre de qué lugar se enviará la mercadería, este proceso ya es externo al de sincronización.

El resto de campos se los puede obtener directamente del pedido de la tienda virtual Prestashop. El proceso se detalla a continuación:

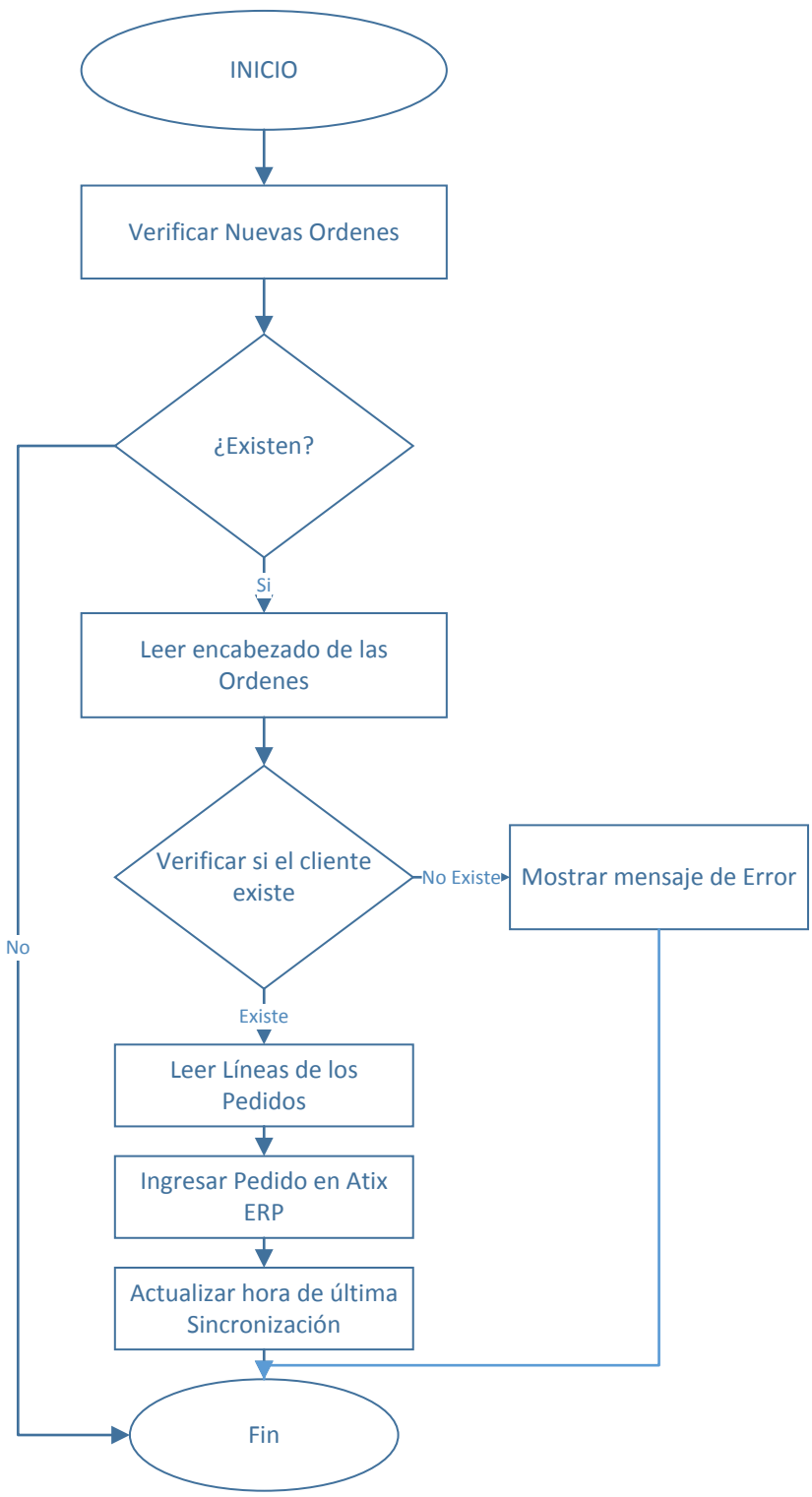


Imagen 16. Diagrama de flujo de proceso para ingreso de pedidos.

Elaborado por: El Investigador

Para verificar si existen nuevas órdenes en el sistema, se realiza una consulta a la tabla de sincronizaciones, consultando la última hora de sincronización y enviándola como parámetro para recibir todos los pedidos a partir de esa fecha y hora.

En caso de existir, se procede a traer únicamente el encabezado del pedido, ya que de esta forma se puede proceder a verificar la existencia del cliente. Esto se realiza por medio de una consulta a la tabla “C_BPartner” con la condición que la identificación del cliente exista, esto retorna un objeto tipo C_BPartner que contiene toda la información del cliente.

```
PO C_BPartner = new Query(Env.getCtx(), "C_BPartner", "TaxID = ?", get_TrxName())
    .setParameters(new Object[] { rucCliente })
    .setOnlyActiveRecords(true).setClient_ID().first();
```

Imagen 17. Codificación de consulta de datos del cliente por identificación.

Elaborado por: El Investigador

En caso de no existir, se procederá a imprimir un mensaje de error, indicando que el cliente no existe en el sistema Atix ERP.

Una vez realizada la comprobación, se procede a leer las líneas del pedido para su ingreso, los datos requeridos para Atix ERP son:

- Código de referencia del producto
- Precio unitario
- Cantidad
- Descuento

Sin embargo, se procede a leer más campos, en caso que el sistema requiera más información sobre los pedidos a futuro. Estos son traídos por medio del servicio web, enviando como parámetro el identificador del pedido.


```

public ArrayList<Order_detail> selectOrderLines(String id) throws PrestaShopWebserviceException,
TransformerException, IOException{
    HashMap<String,Object> getSchemaOpt = new HashMap<String,Object>();
    getSchemaOpt.put("url",shopUrl+"/api/order_details/?display=[id,product_name,product_quantity,product_price," +
        "product_reference,unit_price_tax_incl,unit_price_tax_excl," +
        "total_price_tax_excl,total_price_tax_incl]&filter[id_order]=[\"+id+\"];");
    Document schema = ws.get(getSchemaOpt);
    ArrayList<Order_detail> array_orderd = new ArrayList<Order_detail>();
    int size=schema.getElementsByTagName("id").getLength();
    for(int i=0; i<size; i++){
        Order_detail order_detail= new Order_detail();

        order_detail.setID(Integer.valueOf(schema.getElementsByTagName("id").item(i).
            gettextContent()));
        order_detail.setProduct_reference(schema.getElementsByTagName("product_reference").item(i).
            gettextContent());
        order_detail.setProduct_name(schema.getElementsByTagName("product_name").item(i).
            gettextContent());
        order_detail.setQuantity(Double.valueOf(schema.getElementsByTagName("product_quantity").item(i).
            gettextContent()));
        order_detail.setPrice_unit_without_tax(Double.valueOf(schema.getElementsByTagName("unit_price_tax_excl").item(i).
            gettextContent()));
        order_detail.setPrice_unit_with_tax(Double.valueOf(schema.getElementsByTagName("unit_price_tax_incl").item(i).
            gettextContent()));
        order_detail.setTotal_without_tax(Double.valueOf(schema.getElementsByTagName("total_price_tax_excl").item(i).
            gettextContent()));
        order_detail.setTotal_with_tax(Double.valueOf(schema.getElementsByTagName("total_price_tax_incl").item(i).
            gettextContent()));
        array_orderd.add(order_detail);
    }
    return array_orderd;
}

```

Imagen 18. Codificación de lectura de líneas de pedido según identificador.

Elaborado por El Investigador

Una vez obtenido el encabezado y el detalle del pedido, se procede a ingresarlo al sistema por medio del objeto MOrder. Este requiere valores adicionales como el almacén destino, la organización, lista de precios utilizada, reglas del pedido, estado del documento y estado del mismo.

Estos valores ya están inicializados por defecto, ya que el sistema Atix ERP ya se encuentra implementado y funcionando en la empresa, por lo tanto, solo es necesario acceder a estos valores por medio de objetos.

```

private String crearPedido(int C_BPartner_ID, Timestamp fecha, String descripcion, Integer C_PaymentTerm_ID,
    String paymentRule, List<Producto> listadoProductos, int SalesRep_ID, String DocumentNo, String trxName) {
    Trx trx = null;
    String mensaje = "";
    try {
        trx = Trx.get(trxName, false);

        MOrder order = new MOrder(Env.getCtx(), 0, trx.getTrxName());
        order.setDocumentNo(DocumentNo);
        order.setDateOrdered(fecha);
        order.setAD_Org_ID(AD_Org_ID);
        order.setDatePromised(fecha);
        order.setIsSOTrx(true);
        order.setC_DocTypeTarget_ID(C_DocType_ID);
        order.setC_BPartner_ID(C_BPartner_ID);
        order.setDescription(descripcion);
        order.setM_Warehouse_ID(M_Warehouse_ID);
        order.setM_PriceList_ID(M_PriceList_ID);
        order.setSalesRep_ID(SalesRep_ID);
        order.setPaymentRule(paymentRule);
        order.setC_PaymentTerm_ID(C_PaymentTerm_ID);
        order.setInvoiceRule(MOrder.INVOICERULE_Immediate);
        order.setDocStatus(MOrder.STATUS_Drafted);
        order.setDocAction(MOrder.ACTION_Prepare);
        order.setDeliveryViaRule(MOrder.DELIVERYVIARULE_Pickup);
        order.setM_Shipper_ID(1000000);
        //order.setAD_User_ID(getAD_User_ID());
        order.saveEx();

        for (Producto p : listadoProductos)
            saveDetailProductLine(order, p, trx);

        order.load(get_TrxName());
        if (order.getGrandTotal().compareTo(Env.ZERO)==0){
            order.delete(true);
            return "<br> Error en el Pedido: " + DocumentNo + " "+Msg.translate(getCtx(), "NoLines");
        }

        trx.commit();

        String result = callProcessOrder(order);
        if (result.contains("error"))
            return mensaje = "<br> Error en el Pedido: " + DocumentNo + result;
        else
            readOrder(order.getC_Order_ID());

        updateFildSyncOrdersProlan(DocumentNo);
    } catch (Exception e) {
        mensaje = "<br> Error en el Pedido: " + DocumentNo + " " +e.getMessage();
        log.log(Level.SEVERE, "Error en el Pedido: " + DocumentNo + " " +e.getMessage());
    } finally {
    }
    return mensaje;
}

```

Imagen 19. Codificación del ingreso de pedido a Atix ERP.

Elaborado por: El Investigador.

Este método requiere todo lo consultado anteriormente, en caso de ocurrir algún problema con algún pedido, este será mostrado en pantalla, para mantener un registro apropiado del mismo.

4.4.5.4.1 Web Service de lectura de pedidos

Como previamente se especificó, en primer punto se procede a leer el encabezado de los pedidos, esto para mantener la condición de sincronización en base a la última lectura y la identificación del pedido, para poder obtener sus valores y los productos que contiene.

Este proceso se divide en dos métodos:

1. getOrdersHeader
2. getOrderDetail

El primer método es el encargado de leer los valores como el total de la compra, IVA, total de productos que graban impuesto y que no graban, descuento, fecha, cliente.

El segundo en base al identificador del pedido “id” procede a traer los detalles del mismo, como código de referencia, nombre, cantidad, precio unitario con y sin impuestos, el total de la línea del detalle.

```
public ArrayList<Order> Orderes() throws PrestaShopWebserviceException, TransformerException, IOException{

    HashMap<String, Object> getSchemaOpt = new HashMap<String, Object>();
    getSchemaOpt.put("url", shopUrl+"/api/orders/?display=[id,id_customer,id_address_delivery,reference,total_discounts," +
        "total_paid_tax_excl,total_paid_tax_incl]");
    Document schema = ws.get(getSchemaOpt);
    int size=schema.getElementsByTagName("id").getLength();
    ArrayList<Order> array_order = new ArrayList<Order>();

    for(int i=0; i<size; i++){

        Order order = new Order();

        order.setId(Integer.valueOf(schema.getElementsByTagName("id").item(i).
            getTextNodeContent()));
        order.setReference(schema.getElementsByTagName("reference").item(i).
            getTextNodeContent());
        order.setCustomer_id(selectTaxIdCustomer(schema.getElementsByTagName("id_customer").item(i).
            getTextNodeContent()));
        order.setTotal_discounts(Double.valueOf(schema.getElementsByTagName("total_discounts").item(i).
            getTextNodeContent()));
        order.setTotal_without_taxes(Double.valueOf(schema.getElementsByTagName("total_paid_tax_excl").item(i).
            getTextNodeContent()));
        order.setTotal_with_taxes(Double.valueOf(schema.getElementsByTagName("total_paid_tax_incl").item(i).
            getTextNodeContent()));

        array_order.add(order);

    }
    return array_order;
}
```

Imagen 20. Web Service lectura de encabezados de órdenes.

Elaborado por El Investigador.

Todos estos valores son retornados en objetos, para poder interpretarlos y utilizarlos de manera simple.

4.4.5.4.2 Web Service de lectura de clientes

Al utilizar el servicio web de lectura de pedidos, se obtiene el ID del cliente, que es la identificación única para la tienda virtual, con este campo se puede consultar la información como nombres, dirección y RUC/CI. En este caso en especial, lo único necesario es la cedula o RUC del comprador, por lo tanto, se procede a obtener esta información en base al identificador.

```
private String selectTaxIdCustomer(String id) throws PrestaShopWebserviceException,
                                                TransformerException, IOException{

    HashMap<String,Object> getSchemaOpt = new HashMap<String,Object>();
    getSchemaOpt.put("url",shopUrl+"/api/addresses/?display=" +
                    "[id,dni,id_customer]&filter[id_customer]=[+id+]");
    Document schema = ws.get(getSchemaOpt);
    int size = schema.getElementsByTagName("id").getLength();
    String identity = "";
    for(int i = 0; i<size; i++){
        identity=schema.getElementsByTagName("dni").item(0).getTextContent();
    }
    return identity;
}
```

Imagen 21. Web Service para obtención de identificación del cliente.

Elaborado por: El investigador.

Donde “shopUrl” posee el valor de la dirección del sitio web, seguido por el formato proporcionado por Prestashop para lectura de clientes, añadiendo un filtro, en este caso el “id_customer”. Esta información es leída de la tabla “addresses” que contiene la identificación del comprador.

El resultado de esta consulta dará un documento, por lo tanto, se procede a obtener su información mediante el nombre de la etiqueta que lo contiene, en este caso, la identificación del cliente es almacenada en el campo “dni” por lo tanto ,se obtiene el primer resultado encontrado y se lo retorna, ya que la consulta se la realizo por la clave primaria y no existirían más resultados.

4.4.5.5 Lectura de Categorías y Productos e inserción.

Debido a los requerimientos del sistema, para la parte de inserción de categorías y productos en la tienda virtual Prestashop, se programó un JAR que se ejecuta como tarea programada en el lado del servidor de la tienda.

Este es el encargado de leer categorías y productos nuevos en la cola de mensajería de mqActive e ingresarlo a la tienda virtual Prestashop mediante el uso de servicios web para mantener la integridad de la base de datos y mantener todos los procesos propios de la tienda.

Este sistema sincronizador llevará un registro de conexiones e ingreso de datos conocido como LOG, este será utilizado para optimización del sistema e información sobre su correcto funcionamiento.

El sistema se basa en el siguiente diagrama para su funcionamiento.

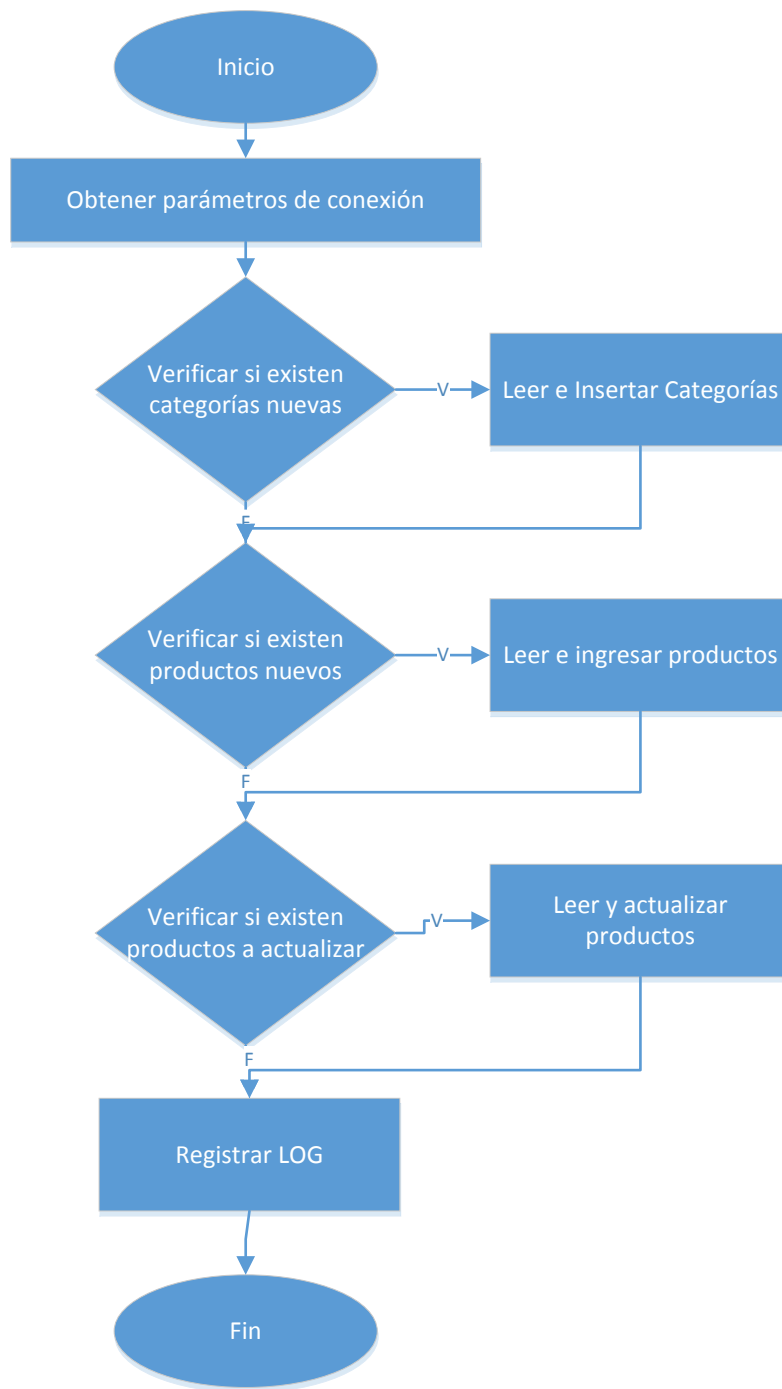


Imagen 22. Diagrama de flujo de proceso, lectura e ingreso de productos y categorías.

Elaborado por: El Investigador

La obtención de parámetros de sincronización es para proveer portabilidad y adaptabilidad al sistema ya que la contraseña del servidor de mensajería puede cambiar o en si su dirección IP.

Por lo tanto, se han tomado en cuenta los siguientes valores para a ser obtenidos al momento de iniciar la sincronización:

- mqDireccion: Dirección IP del sistema de mensajería.
- usu_mqactive: Usuario del sistema de mensajería.
- contr_mqactive: Contraseña del sistema de mensajería.
- ip_webstore: Dirección IP del servidor de la tienda virtual.
- key_webstore: Clave proporcionada por la tienda virtual para su conexión mediante servicios web.
- queueProduct: Nombre de la cola del sistema de mensajería que posee los datos de los productos nuevos.
- queueProductUpdate: Nombre de la cola del sistema de mensajería que posee los datos de los productos que ya han sido sincronizados, pero con sus nuevos valores.
- queueCategory: Nombre de la cola del sistema de mensajería que posee los datos de las nuevas categorías.

Estos valores se encuentran en un archivo plano, ubicado junto al sistema de sincronizador en una carpeta de acceso privado del servidor para proporcionar seguridad a la información contenida.

Una vez obtenidos los parámetros de sincronización se procede a establecer la conexión al sistema de mensajería. De igual manera que se empleó una clase para el envío de información hacia esta, también se usó una clase dedicada para la recepción de información que se muestra a continuación:

```

public class MqRead {
    private ConnectionFactory factory = null;
    private Connection connection = null;
    private Session session = null;
    private Destination destination = null;
    private MessageConsumer consumer = null;
    private static final int TIMEOUT = 1000;
    public MqRead() {

    }
    public ArrayList<String> receiveMessage(String cola) {
        ArrayList<String> elementos = new ArrayList<String>();
        try {
            factory = new ActiveMQConnectionFactory(Parameters.getMqDireccion());
            connection = factory.createConnection();
            connection.start();
            session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
            destination = session.createQueue(cola);
            consumer = session.createConsumer(destination);
            Message message;
            while ((message = consumer.receive(TIMEOUT)) != null) {
                TextMessage text = (TextMessage) message;
                elementos.add(text.getText());
            }
            connection.close();
            connection=null;

        } catch (JMSEException e) {
            e.printStackTrace();
        }
        return elementos;
    }
}

```

Imagen 23. Codificación para recepción de datos del sistema de mensajería.

Elaborado por: El Investigador

Como previamente se definió, el primer proceso es la consulta de datos en la cola que posee los datos de las nuevas categorías, esta se nombró previamente como: “M_Product_Category”, y este nombre es obtenido desde el archivo de configuraciones.

El proceso de definir si la cola de datos tiene información y el de ingresarla, se define en la misma clase, ya que solo es una condición del proceso. El método es el siguiente:


```

public String insertCategories() throws PrestaShopWebserviceException, TransformerException, IOException {
    MqRead rev= new MqRead();
    ArrayList<String> objectCategory = new ArrayList<String>();
    objectCategory =rev.receiveMessage(Parameters.getQueueCategory());

    if(objectCategory.size()>0){
        WSCategory wsC= new WSCategory(Parameters.getIp_webstore(), Parameters.getKey_webstore());
        for (int i=0; i<objectCategory.size();i++){
            Category objCategory = new Category();
            String item[] = objectCategory.get(i).split(">");

            objCategory.setName(item[1].toString());
            objCategory.setDescription(item[2].toString());
            objCategory.setLink_rewrite(removeSpeciaChar(item[1].toString().replace(" ", "-").toLowerCase()));
            objCategory.setId_parent(wsC.getIdParentCategory(item[4].toString()));

            try{
                wsC.addCategory(objCategory);
                Log.info(" Categ Ok: "+i + " : "+item[1].toString());
            }
            catch (Exception e){
                Log.error("Error Categoría: "+item[1].toString()+" error:"+e.getMessage());
            }
        }
    }else
        Log.info("Ninguna categoría que ingresar");

    return "Ok";
}

```

Imagen 24. Codificación del método para el ingreso de categorías.

Elaborado por: El Investigador.

En método presentado se utilizarán dos funciones propias de java y un método especialmente creado para el valor de “link-rewrite”, este es un valor de la tienda Prestashop, que no puede contener espacios, por lo tanto, se ingresan “-”.

También se debe considerar que al momento de leer el mensaje de la cola este viene delimitado por el carácter “>” por lo cual se debe proceder a la separación en campos utilizando el método “Split”.

A la vez no puede contener ningún carácter especial el campo “link-rewrite”, caso contrario lanza excepción al ingresar este atributo y la sincronización de ese registro no se completa.

Por lo tanto, se procede a quitar los caracteres especiales del valor obtenido con el siguiente método:

```

public static String removeSpeciaChar(String input) {

    String original = "áàãéèëíîïóðöúûüññÁÀÁÉÈËÏÎÓÐÖÚÛÜŃÇ(),/._";
    String ascii = "aaaaeeiiiioooouuunAAAEIIIIOOOUUUŃc--C ---";
    String output = input;
    for (int i=0; i<original.length(); i++) {
        output = output.replace(original.charAt(i), ascii.charAt(i));
    }
    return output;
}

```

Imagen 25. Codificación de método para eliminación de caracteres especiales.

Elaborado por: El Investigador

En el presente método se incluyó todos caracteres detectados en nombres de categorías, junto con su correspondiente reemplazo por un carácter simple.

Una vez formada la información para su ingreso, se procede a la llamada del servicio web para la inserción de las categorías obtenidas.

Esta tomará los valores previamente establecidos en el archivo de configuración, para obtener la dirección IP junto con sus credenciales para el acceso y el objeto tipo categoría que contiene la información obtenida.

```

public void addCategory(Category obCategory)
    throws PrestaShopWebserviceException, TransformerException,
    IOException {

    HashMap<String, Object> getSchemaOpt = new HashMap<>();
    getSchemaOpt.put("url", shopUrl+"/api/categories?schema=blank");
    Document schema = ws.get(getSchemaOpt);

    schema.getElementsByTagName("id_parent").item(0).setTextContent(obCategory.getId_parent());
    schema.getElementsByTagName("id_shop_default").item(0).setTextContent("1");
    schema.getElementsByTagName("active").item(0).setTextContent("1");

    Element name = (Element) schema.getElementsByTagName("name").item(0).getFirstChild();
    name.appendChild(schema.createCDATASection(obCategory.getName()));
    name.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element description = (Element) schema.getElementsByTagName("description").item(0).getFirstChild();
    description.appendChild(schema.createCDATASection(obCategory.getDescription()));
    description.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element link_rewrite = (Element) schema.getElementsByTagName("link_rewrite").item(0).getFirstChild();
    link_rewrite.appendChild(schema.createCDATASection(removeSpeciaChar(obCategory.getLink_rewrite())));
    link_rewrite.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    HashMap<String, Object> categoryOpt = new HashMap<String, Object>();
    categoryOpt.put("resource", "categories");
    categoryOpt.put("postXml", ws.DocumentToString(schema));
}

```

Imagen 26. Web Service de inserción de categorías en tienda virtual.

Elaborado por: El Investigador.

En el caso de productos se mantendrán las mismas condiciones, se verificará si existen productos a actualizar y seguido por los productos nuevos, esto debido que un producto que no existía en la tienda virtual fue actualizado y puesto para que se muestre en la tienda, en caso que la actualización se diera primero, éste producto no sería encontrado.

Los datos requeridos para el servicio web son:

- Reference: Código de referencia
- Name: Nombre del producto
- TaxID: Código del impuesto del producto.
- Price: Precio de venta del producto incluido impuesto.
- Description Short: Descripción corta del producto, esta aparecerá alado de la imagen del producto.
- Description: Descripción larga del producto, esta aparecerá en la parte inferior de la imagen del producto.
- Category: Categoría a la que pertenece el producto.
- Link image: Dirección URL de la imagen del producto.

```
public String insertProductos() throws PrestaShopWebserviceException, TransformerException, IOException {
    MqRead rev= new MqRead();
    ArrayList<String> objectProducts = new ArrayList<String>();
    objectProducts =rev.receiveMessage(Parameters.getQueueProduct());
    if(objectProducts.size()>0){
        WSPProduct wsP= new WSPProduct(Parameters.getIp_webstore(), Parameters.getKey_webstore());
        for (int i=0; i<objectProducts.size();i++){
            Product objProduct = new Product();
            String item[] = objectProducts.get(i).split(">");
            objProduct.setReference(item[1].toString());
            objProduct.setName(item[2].toString());
            objProduct.setTaxid(item[3].toString());
            objProduct.setPrice(item[4].toString());
            objProduct.setDescription_short(item[5].toString());
            objProduct.setDescription(item[6].toString());
            objProduct.setCategory(wsP.getIdParentCategory(item[7].toString()));
            objProduct.setLink_image(item[8].toString());

            try{
                wsP.addProduct(objProduct);
                Log.info("Producto Ins Ok: "+item[1].toString());
            }
            catch (Exception e){
                System.out.println(e.getMessage());
                Log.error("Error en Producto: "+item[1].toString()+ " Error: "+e.getMessage());
            }
        }
    }
    }else
    Log.info("Ningun producto que insertar");
    return "Ok";
}
```

Imagen 27. Codificación del método para ingreso de productos.

Elaborado por: El Investigador

El presente método hace la llamada al web Service de inserción de productos, el cual se detalla a continuación, este utiliza los valores obtenidos por el método “insertProducts()”.

```

public void addProduct(Product objProduct)
    throws PrestaShopWebserviceException, TransformerException,
        IOException {

    HashMap<String, Object> getSchemaOpt = new HashMap<String, Object>();
    getSchemaOpt.put("url", shopUrl + "/api/products?schema=blank");
    Document schema = ws.get(getSchemaOpt);

    schema.getElementsByTagName("id_category_default").item(0).setTextContent("2");
    schema.getElementsByTagName("price").item(0).setTextContent(objProduct.getPrice());
    schema.getElementsByTagName("active").item(0).setTextContent("1");
    schema.getElementsByTagName("available_for_order").item(0).setTextContent("1");
    schema.getElementsByTagName("show_price").item(0).setTextContent("1");
    schema.getElementsByTagName("indexed").item(0).setTextContent("1");
    schema.getElementsByTagName("id_tax_rules_group").item(0).setTextContent(getTaxIdProduct(objProduct.getTaxid()));

    Element name = (Element) schema.getElementsByTagName("name").item(0).getFirstChild();
    name.appendChild(schema.createCDATASection(removeSpecialCharName(objProduct.getName())));
    name.setAttribute("id", "1");
    name.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element reference = (Element) schema.getElementsByTagName("reference").item(0);
    reference.appendChild(schema.createCDATASection(objProduct.getReference()));
    reference.setAttribute("id", "1");
    reference.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element category = schema.createElement("category");
    Element catId = schema.createElement("id");
    catId.setTextContent(objProduct.getCategory());
    category.appendChild(catId);
    schema.getElementsByTagName("categories").item(0).appendChild(category);

    Element description = (Element) schema.getElementsByTagName("description").item(0).getFirstChild();
    description.appendChild(schema.createCDATASection(objProduct.getDescription()));
    description.setAttribute("id", "1");
    description.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element description_short = (Element) schema.getElementsByTagName("description_short").item(0).getFirstChild();
    description_short.appendChild(schema.createCDATASection(objProduct.getDescription_short()));
    description_short.setAttribute("id", "1");
    description_short.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element available_now = (Element) schema.getElementsByTagName("available_now").item(0).getFirstChild();
    available_now.appendChild(schema.createCDATASection("Disponible ahora"));
    available_now.setAttribute("id", "1");
    available_now.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element available_later = (Element) schema.getElementsByTagName("available_later").item(0).getFirstChild();
    available_later.appendChild(schema.createCDATASection("Disponible después"));
    available_later.setAttribute("id", "1");
    available_later.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    HashMap<String, Object> productOpt = new HashMap<String, Object>();
    productOpt.put("resource", "products");
    productOpt.put("postXml", ws.DocumentToString(schema));
    Document product = ws.add(productOpt);

    if (objProduct.getLink_image() != null) {
        Document productImg = ws.addImg(objProduct.getLink_image(), Integer.valueOf(
            getCharacterDataFromElement((Element) product.getElementsByTagName("id").item(0))));
        ws.DocumentToString(productImg);
    }
}

```

Imagen 28. Web Service de inserción de productos en tienda virtual.

Elaborado por: El Investigador

De similar manera se codifica el método encargado de la actualización de productos, la diferencia es el llamado del servicio web, en este caso se emplea “updateProduct” el cual cumple con la función requerida.

```

public String updateProductos() throws PrestaShopWebServiceException, TransformerException, IOException {

    MqRead rev= new MqRead();
    ArrayList<String> objectProductsUpdate = new ArrayList<String>();
    objectProductsUpdate =rev.receiveMessage(Parameters.getQueueProductUpdate());

    if(objectProductsUpdate.size(>0){
        WSProduct wsP= new WSProduct(Parameters.getIp_webstore(), Parameters.getKey_webstore());
        for (int i=0; i<objectProductsUpdate.size();i++){

            Product objProduct = new Product();
            String item[] = objectProductsUpdate.get(i).split(">");
            objProduct.setReference(item[2].toString().substring(1));
            objProduct.setName(item[3].toString().substring(1));
            objProduct.setTaxid(item[4].toString().substring(1));
            objProduct.setPrice(item[5].toString().substring(1));
            objProduct.setDescription_short(item[6].toString().substring(1));
            objProduct.setDescription(item[7].toString().substring(1));
            objProduct.setCategory(wsP.getIdParentCategory(item[8].toString().substring(1)));
            try{
                wsP.updateProduct(objProduct);
                Log.info("Producto Upd Ok: "+item[1].toString());
            }
            catch (Exception e){
                System.out.println(e.getMessage());
                Log.error("Error en Producto: "+item[1].toString()+ " Error: "+e.getMessage());
            }
        }
    }else
        Log.info("Ningun producto que actualizar");
    return "Ok";
}
}

```

Imagen 29. Codificación del método para la actualización de productos.

Elaborado por: El Investigador

En el caso del Web Service de actualización de productos, es necesario enviar toda la información. Antes de proceder a ejecutar el proceso, se debe realizar previamente la consulta del “ID” propio de la tienda virtual en base a la referencia de producto para establecer el registro a ser modificado.

```

private String getIdProduct(String reference) throws PrestaShopWebserviceException{
    String id = "";
    reference = reference.replace(" ", "%20");
    HashMap<String, Object> getSchemaOpt = new HashMap<String, Object>();
    getSchemaOpt.put("url", shopUrl
        + "/api/products/?display=[id]&filter[reference]=[ " + reference + " ]");
    Document schema = ws.get(getSchemaOpt);
    id = schema.getElementsByTagName("id").item(0).getTextContent();

    return id;
}

```

Imagen 30. Web Service de Consulta de ID producto.

Elaborado por: El Investigador.

Luego de obtener el identificador, se puede proceder con la actualización correspondiente al registro, para esto el web Service tiene una ligera modificación donde el ID pasa a ser un parámetro más del método. Este se encuentra establecido tanto en la información del producto, como en el momento de establecer la conexión y enviar los datos, esta es la manera de funcionar de la tienda virtual Prestashop.

```

public void updateProduct(Product objProduct) throws PrestaShopWebserviceException, TransformerException,
IOException{

    String product_id = getIdProduct(objProduct.getReference());

    HashMap<String, Object> getSchemaOpt = new HashMap<String, Object>();
    getSchemaOpt.put("url", shopUrl+"/api/products?schema=blank");
    Document schema = ws.get(getSchemaOpt);

    schema.getElementsByTagName("id_category_default").item(0).setTextContent("2");
    schema.getElementsByTagName("price").item(0).setTextContent(objProduct.getPrice());
    schema.getElementsByTagName("active").item(0).setTextContent("1");
    schema.getElementsByTagName("available_for_order").item(0).setTextContent("1");
    schema.getElementsByTagName("show_price").item(0).setTextContent("1");
    schema.getElementsByTagName("indexed").item(0).setTextContent("1");
    schema.getElementsByTagName("id_tax_rules_group").item(0).setTextContent(getTaxIdProduct(objProduct.getTaxid()));
    schema.getElementsByTagName("id").item(0).setTextContent(product_id);

    Element name = (Element) schema.getElementsByTagName("name").item(0).getFirstChild();
    name.appendChild(schema.createCDATASection(removeSpecialCharName(objProduct.getName())));
    name.setAttribute("id", "1");
    name.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

    Element reference = (Element) schema.getElementsByTagName("reference").item(0);
    reference.appendChild(schema.createCDATASection(objProduct.getReference()));
    reference.setAttribute("id", "1");
    reference.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);
}

```

```

Element category = schema.createElement("category");
Element catId = schema.createElement("id");
catId.setTextContent(objProduct.getCategory());
category.appendChild(catId);
schema.getElementsByTagName("categories").item(0).appendChild(category);

Element description = (Element) schema.getElementsByTagName("description").item(0).getFirstChild();
description.appendChild(schema.createCDATASection(objProduct.getDescription()));
description.setAttribute("id", "1");
description.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

Element description_short = (Element) schema.getElementsByTagName("description_short").item(0).getFirstChild();
description_short.appendChild(schema.createCDATASection(objProduct.getDescription_short()));
description_short.setAttribute("id", "1");
description_short.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

Element link_rewrite = (Element) schema.getElementsByTagName("link_rewrite").item(0).getFirstChild();
link_rewrite.appendChild(schema.createCDATASection(objProduct.getLink_rewrite()));
link_rewrite.setAttribute("id", "1");
link_rewrite.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

Element available_now = (Element) schema.getElementsByTagName("available_now").item(0).getFirstChild();
available_now.appendChild(schema.createCDATASection("Disponible ahora"));
available_now.setAttribute("id", "1");
available_now.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

Element available_later = (Element) schema.getElementsByTagName("available_later").item(0).getFirstChild();
available_later.appendChild(schema.createCDATASection("Disponible después"));
available_later.setAttribute("id", "1");
available_later.setAttribute("xlink:href", this.shopUrl+"/api/languages/"+1);

HashMap<String, Object> productOpt = new HashMap<String, Object>();
productOpt.put("resource", "products");
productOpt.put("id", product_id);
productOpt.put("putXml", ws.DocumentToString(schema));
Document product = ws.edit(productOpt);

Document productImg = ws.addImg(objProduct.getLink_image(), Integer.valueOf(getCharacterDataFromElement((Element)
ws.DocumentToString(productImg)));
}

```

Imagen 31. Web Service de Actualización de Productos.

Elaborado por: El Investigador.

Tanto en el método de sincronizar categorías como el de productos el servicio web requiere un objeto tipo “*Category*” y “*Product*”, motivo por el cual se creó las clases de los mismos que contienen todas las propiedades que se requieren.

Las clases poseen métodos get y set los cuales sirven para obtener la información y asignarla. De esta manera los datos de un producto o una categoría se envía encapsulada y contiene toda la información necesaria.

En la clase “*Product*” tenemos:

```
public class Product {  
    private String id;  
    private String reference;  
    private String name;  
    private String category;  
    private String price;  
    private String description_short;  
    private String taxid;  
    private String link_rewrite;  
    private String link_image;  
    public String getId() {  
        return id;  
    }  
    public void setId(String id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String getCategory() {  
        return category;  
    }  
    public void setCategory(String category) {  
        this.category = category;  
    }  
    public String getPrice() {  
        return price;  
    }  
    public void setPrice(String price) {  
        this.price = price;  
    }  
    public String getDescription_short() {  
        return description_short;  
    }  
    public void setDescription_short(String description_short) {  
        this.description_short = description_short;  
    }  
    public String getDescription() {  
        return description;  
    }  
    public String getDescription() {  
        return description;  
    }  
    public void setDescription(String description) {  
        this.description = description;  
    }  
    public String getReference() {  
        return reference;  
    }  
    public void setReference(String reference) {  
        this.reference = reference;  
    }  
    public String getTaxid() {  
        return taxid;  
    }  
    public void setTaxid(String taxid) {  
        this.taxid = taxid;  
    }  
    public String getLink_rewrite() {  
        return link_rewrite;  
    }  
    public void setLink_rewrite(String link_rewrite) {  
        this.link_rewrite = link_rewrite;  
    }  
    public String getLink_image() {  
        return link_image;  
    }  
    public void setLink_image(String link_image) {  
        this.link_image = link_image;  
    }  
}
```

Imagen 32. Clase *Product* para encapsulamiento de información.

Elaborado por: El Investigador.

En la clase “*Product*” se puede notar que todos los datos son de tipo *String* (Cadena de caracteres) lo cual es lógico, debido a que son características del producto, pero en precio no, esto se mantiene como texto, debido a que el servicio web de la tienda virtual Prestashop no distingue entre tipos de datos.

En la clase “*Category*” tenemos:

```
public class Category {  
  
    private String id_parent;  
    private String name;  
    private String description;  
    private String link_rewrite;  
  
    public String getId_parent() {  
        return id_parent;  
    }  
    public void setId_parent(String id_parent) {  
        this.id_parent = id_parent;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String getDescription() {  
        return description;  
    }  
    public void setDescription(String description) {  
        this.description = description;  
    }  
    public String getLink_rewrite() {  
        return link_rewrite;  
    }  
    public void setLink_rewrite(String link_rewrite) {  
        this.link_rewrite = link_rewrite;  
    }  
}
```

Imagen 33. Clase Category para encapsulamiento de información.

Elaborado por: El Investigador.

La clase “*Category*” también posee todos sus tipos de datos como *String*, pero con menos elementos que la clase “*Product*”, debido a que la función de las categorías es únicamente organizar el menú en la pantalla principal de la tienda virtual Prestashop.

4.4.6. Glosario de métodos.

El objetivo de emplear un glosario de métodos, es facilitar el mantenimiento del código, corregir errores y mejorar sus funciones. Por lo tanto, se divide en los métodos contenidos por el Plug-in en Atix ERP y por el JAR de sincronización en la tienda virtual, todos estos contenidos por una clase.

4.4.6.1 Métodos Empleados en Atix ERP.

- Send
 - sendMessage(). Recibe como parámetro un ArrayList de cadenas de texto para enviar al sistema de mensajería MqActive.
- ReadOrdersPresta
 - orders(). Retorna un ArrayList con las órdenes leídas.
 - selectOrderLines(). Retorna un ArrayList con los detalles de la orden, en base a la orden que recibe como parámetro de entrada.
 - selectClient(). Obtiene los datos del cliente en base al ID del mismo.
 - selectAddress(). Obtiene la dirección que el cliente especificó en el pedido realizado en la tienda virtual.
 - selectTaxIdCustomer(). Obtiene el número de documento de identidad del cliente en base a su ID interno.
- Update
 - UpdateSyncProduct(). Actualiza el campo de la tabla producto a sincronizado para que no vuelva a ser leído en caso que no haya sido actualizado.
- PWebshop
 - prepare(). Obtiene los valores de los parámetros ingresados por la interfaz gráfica.
 - doIt(). Ejecuta los procesos en base a los parámetros leídos.
 - sendProducts(). Lee los productos nuevos a ser sincronizados y los envía al sistema de mensajería MqActive.

- sendCategories(). Lee las categorías a ser sincronizadas y las envía al sistema de mensajería MqActive.
- updateProducts(). Lee los productos que ya estén sincronizados y que hayan sido actualizados para posteriormente enviarlos al sistema de mensajería MqActive.
- createOrder(). Crea una orden en Atix ERP para poder ingresar sus componentes.
- saveDetailProductLine(). Guarda los detalles (productos) de los pedidos en la misma orden creada por createOrder().
- callProcessOrder(). Procesa el pedido ingresado para que se verifiquen diferentes condiciones establecidas en diferentes procesos de AtixERP.
- getOrderDetail(). Obtiene los detalles de los pedidos directamente de la tienda virtual.
- setProductDetail(). Procesa la información leída por el método getOrderDetail() para asignarle el formato requerido por Atix ERP.
- getOrdersHeader(). Obtiene los encabezados de las órdenes nuevas desde la tienda virtual.
- EPriceUpdate
 - doHandleEvent(). Evento especial para detectar un cambio en la lista de precios del producto y actualizar el campo en la tabla padre.
- EProductUpdate
 - doHandleEvent(). Evento especial para detectar un cambio en los atributos del producto y actualizar el campo en la misma tabla con el objetivo de ser incluido en el proceso de sincronización.

4.4.6.2 Métodos Empleados en JAR de sincronización en tienda virtual.

- Main
 - main(). Inicializa el archivo de log, parámetros de lectura y procesos de sincronizacion.
- MqRead

- receiveMessage(). Recibe como parámetro el nombre de la cola a leer y retorna un ArrayList de los datos encontrados.
- Parameters
 - getParameters(). Lee los parámetros del archivo de configuración y los asigna a variables accesibles en todo el sistema.
- InsertProducts
 - insertProducts(). Lee los productos nuevos desde la cola de mensajería MqActive y los inserta en la tienda virtual por medio de Web Service.
 - updateProducts(). Lee los productos actualizados desde la cola de mensajería MqActive y los actualiza en la tienda virtual por medio de Web Service.
- InsertCategories
 - insertCategories(). Lee las categorías nuevas desde el sistema de mensajería MqActive y las inserta en la tienda virtual por medio de Web Service.
 - removeSpecialChar(). Recibe una cadena de texto y elimina los caracteres especiales, retornando una cadena sin ellos.

4.5. Fase V: Pruebas.

4.5.1. Pruebas de Caja Blanca

En esta fase del desarrollo del presente proyecto se pretende realizar las pruebas del sistema, con el objetivo de documentar los resultados obtenidos.

Previamente es necesario realizar las configuraciones pertinentes para que el sistema funcione, por lo tanto, en un servidor de pruebas con un sistema de comercio electrónico idéntico al de producción se procede a realizar los ajustes.

4.5.1.1 Configuración para sincronizador de la tienda virtual.

Como se describió previamente, el proceso sincronizador se encuentra dividido en tres secciones, el Plug-in dentro de Atix ERP, el servidor de mensajería y el sistema sincronizador de productos y categorías asincrónico.

Este último se debe encontrar dentro del servidor de la tienda virtual, por diferentes motivos como seguridad, velocidad de inserción de información.

Este sistema como se detalló anteriormente se desarrolló en java, por lo cual el resultado de la compilación es un archivo .JAR el cual va a ser ejecutado por medio de una tarea *crontab* que viene a ser una utilidad de sistemas operativos Linux que facilita la ejecución de procesos a un determinado tiempo.

Antes de configurar el *crontab* se debe crear un archivo .sh que viene a ser un *script* (secuencia de ordenes) ejecutable en Linux.

4.5.1.1.1 Creación de script de ejecución.

Este archivo será creado en la misma carpeta donde se encuentra el sistema de sincronización, por motivos de facilidad de mantenimiento. Este contendrá el comando para llamar al sistema sincronizador y será llamado sinc.sh.

El contenido será el siguiente:

```
java -jar /opt/sincronizador/sinc.jar
```

Imagen 34. Contenido del archivo sinc.sh.

Elaborado por el Investigador

Donde el comando para ejecutar archivos compilados en java es: “java -jar” y seguido por el archivo junto con su ruta, este fue colocado en la carpeta “opt/sincronizador”.

4.5.1.1.2 Configuración de tarea crontab.

Una tarea *crontab* requiere una sintaxis específica, la cual está conformada por minutos, horas, días, mes y días de la semana junto con la ubicación del script a ejecutar.

Los valores están dentro del siguiente rango:

- Minutos: 0 a 59.
- Horas: 0 a 23.
- Días: 1 a 31.

- Mes: 1 a 12.
- Días de la semana: 0 a 6.

Comenzando con 0 que equivalente a Domingo.

Tomando los parámetros indicados en cuenta, se configuró el crontab para ejecutarse todos los días, a las 24H00, esto bajo petición de los usuarios. Esta es la hora indicada para que todos los productos y categorías se sincronicen y la información se publique en la tienda virtual.

```
0 23 * * * /opt/sincronizador/sinc.sh
```

Imagen 35. Configuración de tarea programada (Crontab).

Elaborado por el investigador.

4.5.1.2 Prueba de envió de categorías y productos.

Para verificar el funcionamiento del envío de información, se procede a utilizar una tienda virtual copia de la original, por medio de esto se garantiza el funcionamiento del sistema en producción en caso de ser exitoso.

La primera prueba de envío se procede a enviar todos los datos de categorías y productos destinados a ser sincronizados, de esta manera se garantiza el funcionamiento ante una carga grande de información.

Para realizarla se procede a escoger las opciones de Sincronizar productos y Sincronizar todos los productos, para ejecutar la tarea ser presiona en “Ok”.

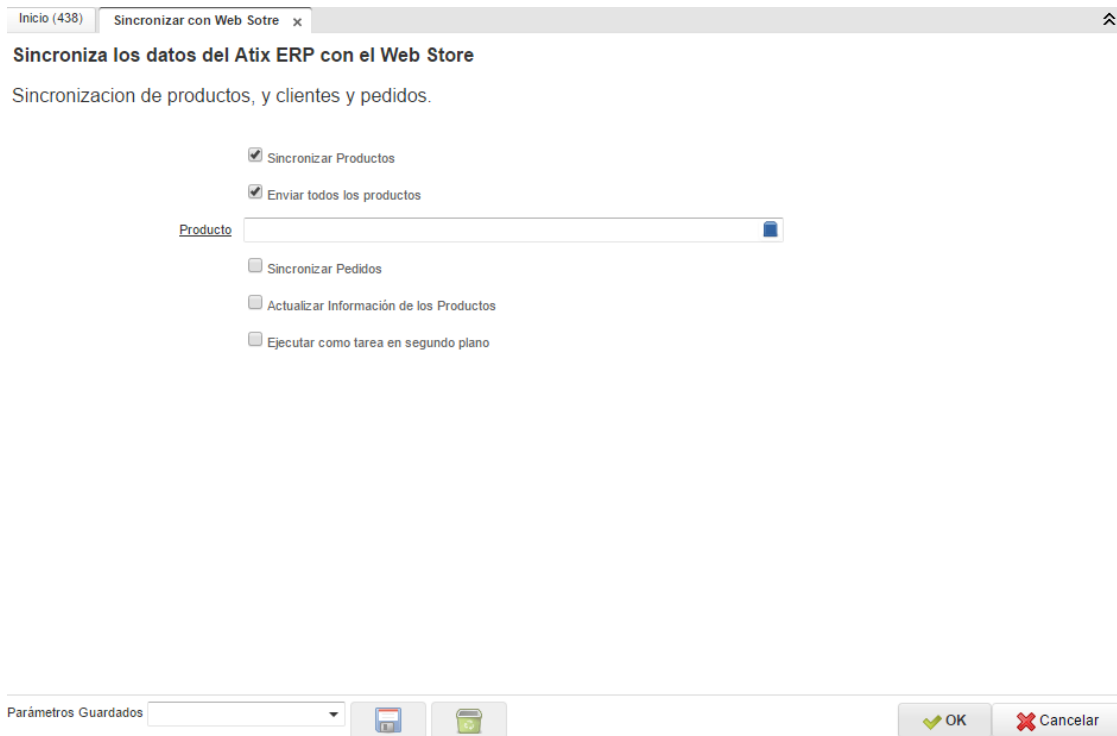


Imagen 36. Elección de opciones para prueba de envío de información.

Elaborado por: El Investigador.

Al momento de ejecutar se presentará la siguiente información, la cual bloquea la pantalla para no permitir modificaciones en el proceso de envío.

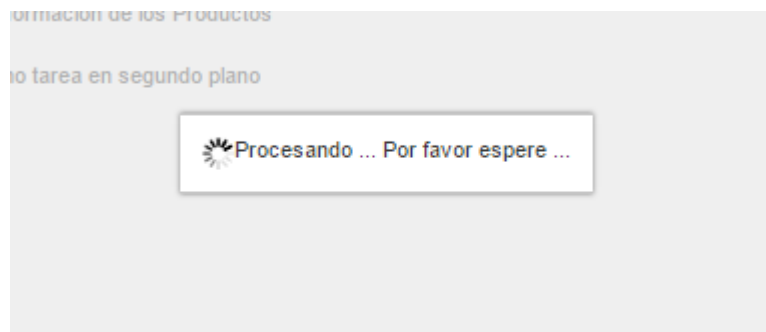


Imagen 37. Bloqueo de Interfaz en proceso de sincronización.

Elaborado por: El Investigador.

4.5.2. Pruebas de Caja Negra

El propósito de las presentes pruebas es tener un registro de los resultados de las entradas y salidas de los procesos que integra el sistema. Para poder lograr el objetivo principal de una prueba de caja negra, se pidió el apoyo de un usuario que no había utilizado antes Atix ERP. Este usuario sería el encargado de intentar realizar los procesos de enviar y recibir información de prueba de la tienda virtual.

El primer caso de la prueba es la apertura del módulo y la elección de la función a ejecutar en el cual se pueden obtener los siguientes caminos:

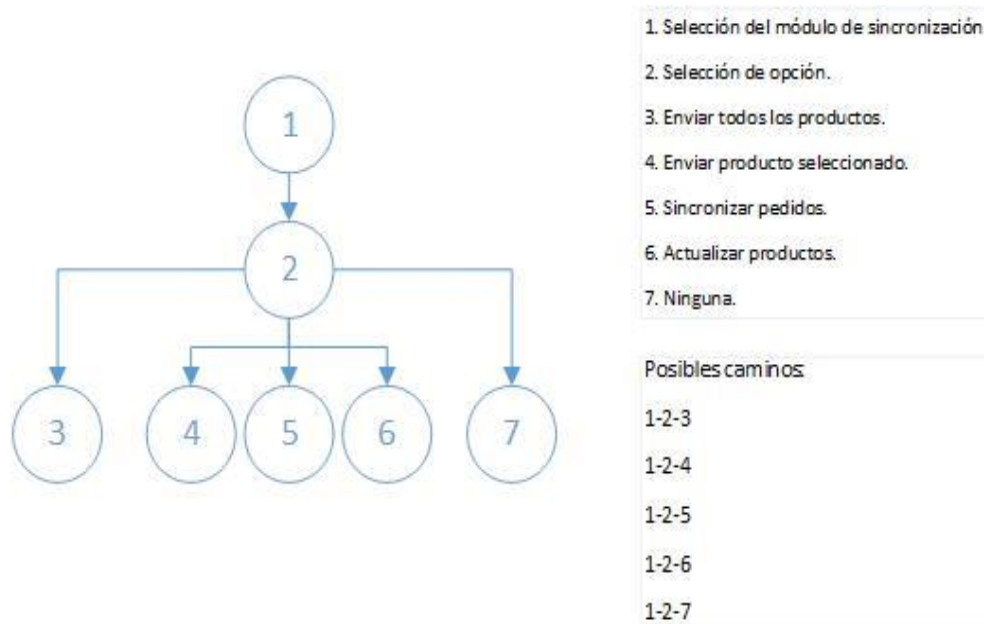


Imagen 38. Posibles caminos para la apertura y ejecución del módulo.

Elaborado por: El Investigador

- Pantalla de apertura del módulo con sus respectivas opciones:

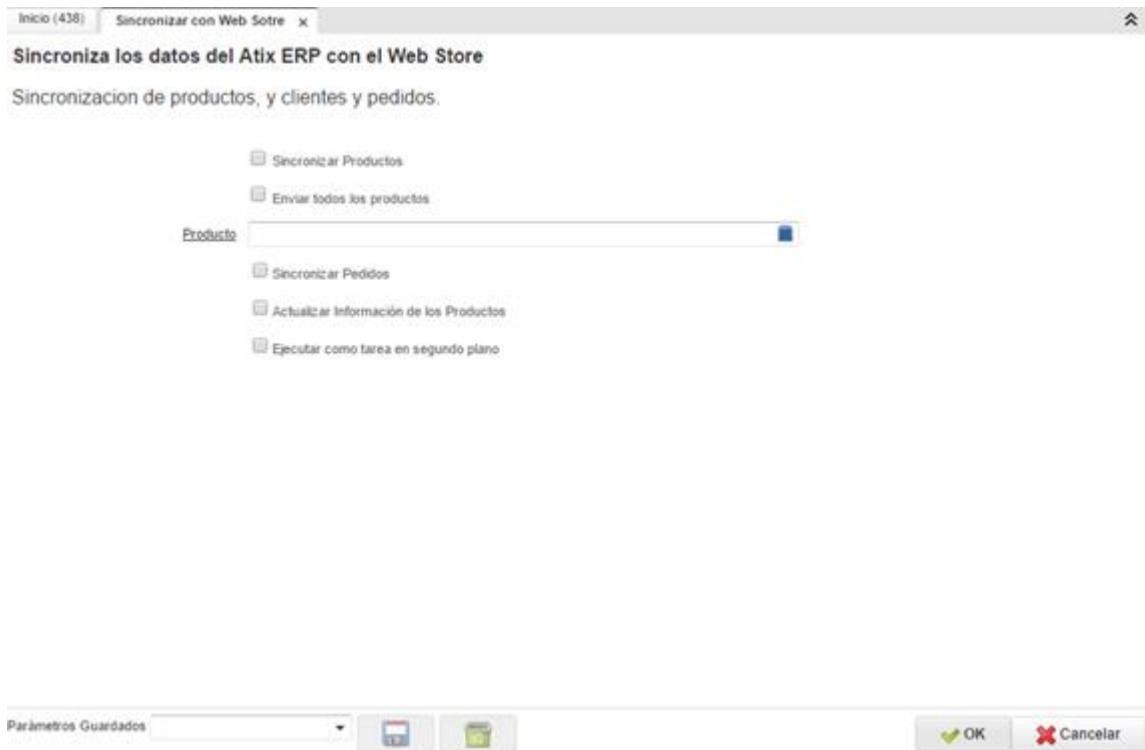


Imagen 39. Prueba de apertura del módulo.

Elaborado por: El Investigador

En función a la solicitud hacia el usuario de sincronizar todos los productos el camino a seguir es: “1-2-3”. Dentro del cual se realiza el siguiente flujo de procesos:

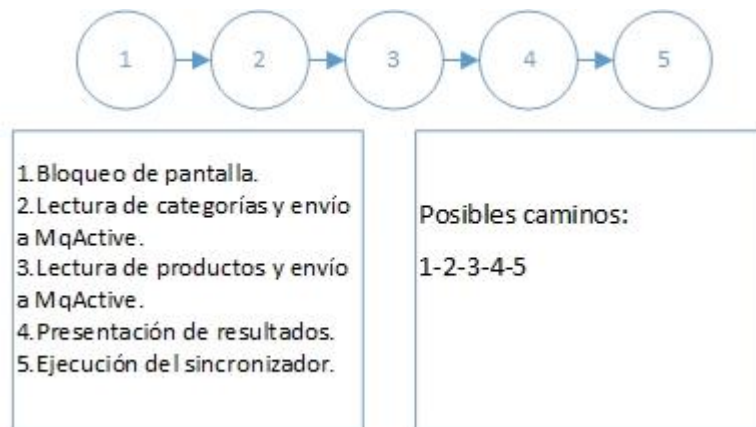


Imagen 40. Camino para la sincronización de productos.

Elaborado por: El Investigador.

- Pantalla de bloqueo durante la ejecución del módulo:

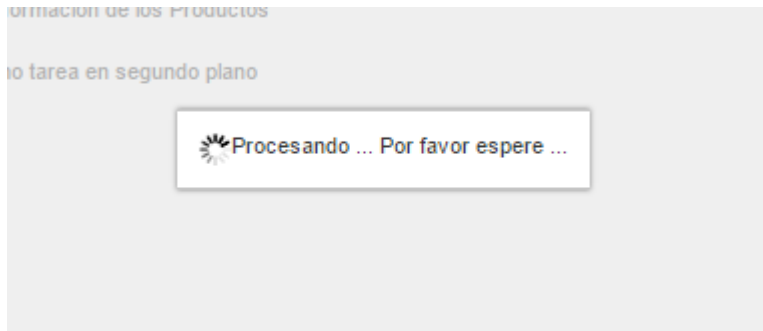


Imagen 41. Prueba de pantalla de bloqueo del módulo.

Elaborado por: El Investigador.

Al momento de concluir la tarea, se presentará el número de productos y categorías enviadas, para verificar que toda la información este en el servidor de mensajería MqActive, se procede a comprar los números presentados en la interfaz del Plug in con la del servidor de mensajería.

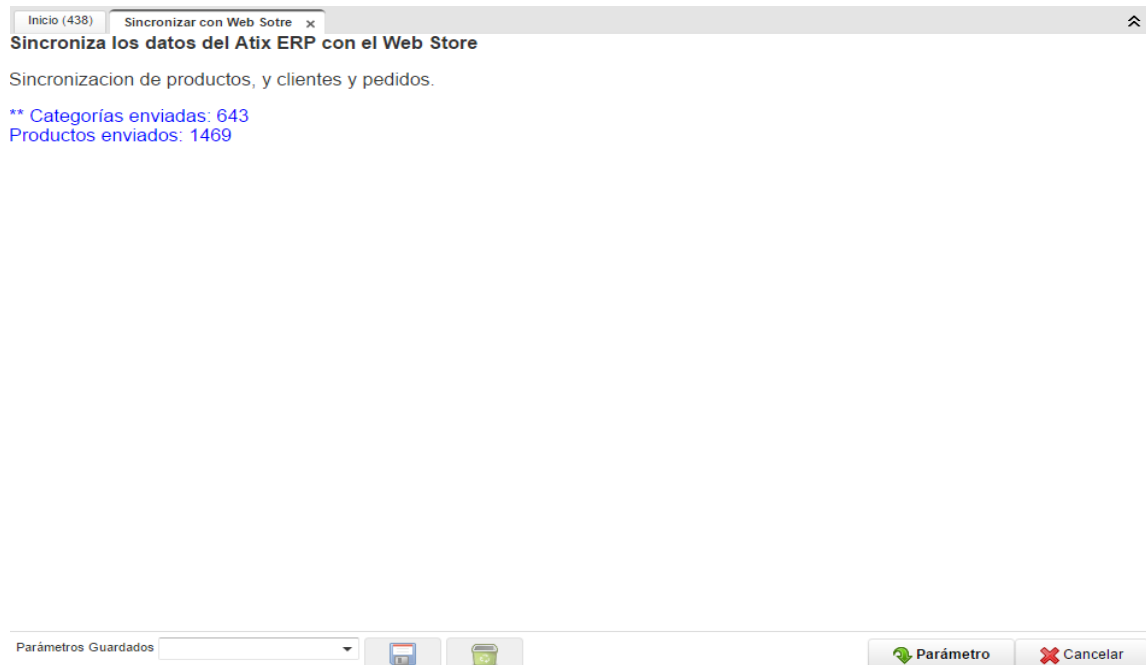


Imagen 42. Información de cantidad de datos enviados presentado por Atix ERP.

Elaborado por: El Investigador.

Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Queue Name

Queues

Name ↓	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
M_Product	1469	0	1469	0	Browse Active Consumers atom rss	Send To Purge Delete
M_Product_Category	643	0	643	0	Browse Active Consumers atom rss	Send To Purge Delete

Imagen 43. Información de cantidad de datos recibidos por MqActive.

Elaborado por: El Investigador.

	Número Presentado en Interfaz Atix ERP	Número Presentado en MqActive
Categorías	643	643
Productos	1469	1469

Tabla 33. Comparación de datos enviados entre Interfaz y MqActive.

Elaborado por: El Investigador.

En el lado del servidor de la tienda virtual se procede a eliminar todos los datos de categorías y productos, para verificar que la sincronización sea completa.

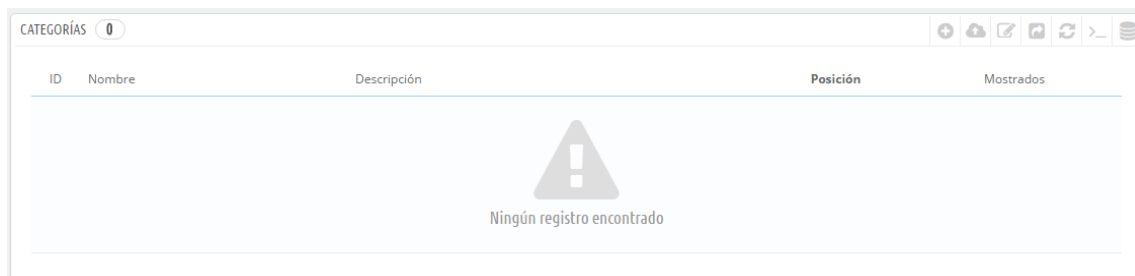


Imagen 44. Interfaz de tienda virtual sin contenido de categoría.

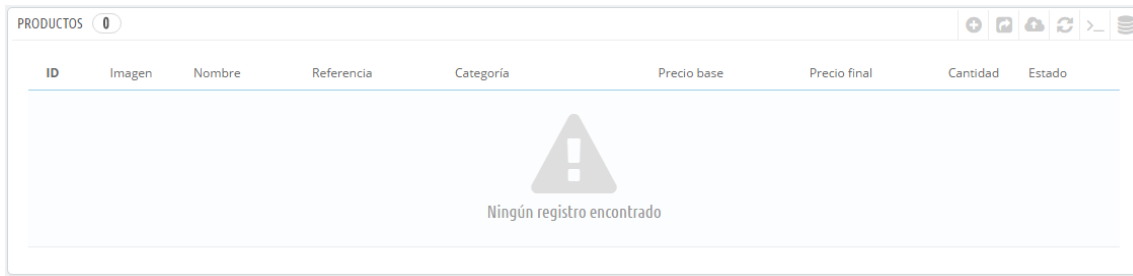


Imagen 45. Interfaz de tienda virtual sin contenido de productos.

Se procede a configurar la tarea cron con motivos de pruebas, para que sea ejecutada cada 10 minutos, de esta manera comprobar que esta se encuentre correctamente configurada y verificar la información presentada en la tienda virtual.

Después de ser ejecutada la sincronización se procede a la verificación de los datos en la tienda virtual por medio del uso de la interfaz gráfica de la tienda virtual. Para acceder al sitio se ingresa en el navegador al panel de administración por medio de la dirección con la siguiente sintaxis:

IP/carpeta_Admin

Donde “IP” es la dirección web de la tienda y “carpeta_Admin” es la carpeta donde se encuentra los archivos de configuración de la tienda.

Una vez ingresado se procede a verificar el número de categorías:

ID	Nombre	Descripción	Posición	Mostrados	
1598	Standard	null	1	✓	Modificar
1599	Trees	null	2	✓	Modificar
1600	Bushes	null	3	✓	Modificar
1601	Tools	Garden Tools	4	✓	Modificar
1602	Chemicals	Lawn Chemicals	5	✓	Modificar
1603	Patio	Patio and Lawn Furniture	6	✓	Modificar
1604	Documentation	Documentation Download	7	✓	Modificar
1605	Training	null	8	✓	Modificar
1606	Assembly	null	9	✓	Modificar

Imagen 46. Interfaz de categorías sincronizadas en Prestashop.




La presente interfaz muestra un total de 27 categorías, estas solo son las categorías padres, las cuales contienen categorías hijas. Una categoría padre se diferencia únicamente por la opción de modificación, esta debe decir “ver”, ya que al dar clic se presentan las categorías hijas.

1612	FERRETERIA	null	15	✓	Ver
1613	HERRAMIENTAS	null	16	✓	Ver
1614	CONSTRUCCION	null	17	✓	Ver
1615	ELECTRICIDAD	null	18	✓	Ver
1616	LIQUIDACION	null	19	✓	Ver
1617	ESPECIAL	null	20	✓	Ver
1618	GASFITERIA Y ACABADOS	null	21	✓	Ver
1619	PINTURAS Y AFINES	null	22	✓	Ver

Imagen 47. Categorías padres en tienda virtual.

Al desplegar uno de estos elementos, por ejemplo “Ferretería” se muestran las categorías que contienen, las cuales también pueden ser categorías padres.

FERRETERIA

 Añadir nueva categoría
  Módulos y Servicios recomendados
  Ayuda

<input type="checkbox"/>	1630	FERRETERIA_RUEDAS Y GARRUCHAS	null	+ 3	✓	Modificar
<input type="checkbox"/>	1631	FERRETERIA_ABRASIVOS	null	+ 4	✓	Modificar
<input type="checkbox"/>	1632	FERRETERIA_ADHESIVOS GENERAL	null	+ 5	✓	Modificar
<input type="checkbox"/>	1633	FERRETERIA_CADENAS TRAMPAS Y NYLONS	null	+ 6	✓	Modificar
<input type="checkbox"/>	1634	FERRETERIA_CAJAS Y ORGANIZADORES	null	+ 7	✓	Modificar
<input type="checkbox"/>	1635	FERRETERIA_CANDADOS CERRADURAS Y HERRAJES	null	+ 8	✓	Modificar
<input type="checkbox"/>	1636	FERRETERIA_FIJACIONES	null	+ 9	✓	Modificar
<input type="checkbox"/>	1637	FERRETERIA_GANCHOS ROPA	null	+ 10	✓	Ver
<input type="checkbox"/>	1638	FERRETERIA_GAS	null	+ 11	✓	Ver
<input type="checkbox"/>	1639	FERRETERIA_MANGUERAS	null	+ 12	✓	Ver
<input type="checkbox"/>	1640	FERRETERIA_PEGAMENTOS Y SOLDADURAS	null	+ 13	✓	Ver
<input type="checkbox"/>	1641	FERRETERIA_HERRAMIENTAS MANUALES	null	+ 14	✓	Ver

Imagen 48. Categorías pertenecientes a "Ferretería"

En cuanto a productos, el listado de la tienda virtual muestra el total de elementos sin importar su clasificación, por lo tanto, se puede verificar que la sincronización fue exitosa, comparando cifras de datos enviados contra sincronizados.

PRODUCTOS 1469

ID	Imagen	Nombre	Referencia	Categoría	Precio base	Precio final	Cantidad	Estado	
761		IM NICHOLSON SIERRA GRANO GRUESO 18	L12232	Inicio	2,66 \$	2,98 \$	0	✓	Modificar
762		ADAPTADOR DE 3 A 2 COOPER 419GY	A01062	Inicio	0,77 \$	0,86 \$	0	✓	Modificar
763		ADAPTADOR DE CAUCHO 50 MM A 1 1/4"	A01064	Inicio	0,27 \$	0,30 \$	0	✓	Modificar
764		ME IM ADAPTADOR NEGR DE REDONDO A PLANO PBP047	A01065	Inicio	0,19 \$	0,22 \$	0	✓	Modificar
765		DW ATORNILLADOR DW255 VVR 6AMP	A01066	Inicio	119,70 \$	134,07 \$	0	✓	Modificar

Imagen 49. Productos sincronizados en tienda virtual.

La cifra presentada en la parte izquierda superior de la “Imagen 45” muestra un total de 1469 productos ingresados, la cual cuadra con el número de productos inicialmente enviados.

De acuerdo a las pruebas realizadas en el módulo en cuanto al correcto funcionamiento de sus funciones, califica para el uso de los usuarios finales.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se analizó el ingreso y procesamiento de información en Atix ERP y la tienda virtual por medio del ingreso de información a través de su interfaz gráfica y consultas SQL a la base de datos, obteniendo un esquema claro de la distribución de la información y campos clave para la sincronización.
- Por medio del diseño de un modelo de sincronización se obtuvo una idea clara del flujo de información, facilitando la programación del módulo y mantenimiento del mismo.
- En base a los datos recolectados en la metodología aplicada, se desarrollaron las herramientas capaces de llevar a cabo el proceso de sincronización entre Atix ERP y la tienda virtual utilizando el sistema de mensajería de java (JMS).
- El error de duplicidad de pedidos fue eliminado, ya que al disponer un tipo de documento único y sincronizar directamente el identificador del pedido, este no podrá ser repetido por el control de restricciones, evitando problemas de facturación.
- Se puede llegar a una integración más compleja y completa para brindar mayor información al usuario final por medio de la tienda virtual, aumentando información sobre los atributos del producto como disponibilidad de colores o tamaños en ciertos casos.
- Atix ERP (iDempiere) es un sistema completo, que por código java se puede extender sin problemas, hasta abastecer todas las necesidades de información de la empresa mediante la correcta programación y el uso de estándares para mantener un código legible.

5.1 Recomendaciones

- Al personal encargado de la creación/modificación de productos en ATIX ERP:
 - Las imágenes de los productos subidas al servidor no deben tener un tamaño mayor a 400 KB, esto debido a que el proceso de visualización de productos en la tienda virtual Prestashop debe ser rápida, si la página tarda en cargarse, es probable que se haya perdido un cliente.
 - Para seleccionar los productos a sincronizarse, deben estar conscientes que este va a mostrarse en la tienda virtual, la cual brinda una imagen de la empresa, por lo tanto, la descripción del producto o a la vez su nombre no debe contener faltas de ortografía o error en su precio.
- Al personal encargado de ingreso de pedidos, se recomienda capacitar a nuevos usuarios que utilicen el modulo por medio del manual del usuario o por el personal de AlquimiaSoft S.A.
- Se sugiere al personal a cargo del servidor revisar los LOG del sincronizador y reportar al personal de AlquimiaSoft S.A. en caso de encontrar alguna irregularidad. También mantener un respaldo de los mismos.

Bibliografía:

- [1] F. A. R. Albores, «Gestiopolis,» 01 Septiembre 2014. [En línea]. Available: <http://www.gestiopolis.com/tecnologias-emergentes-de-la-informacion-y-desarrollo-de-software/>.
- [2] J. Fleitman, La importancia de los sistemas de información y control en la empresa, 2000.
- [3] www.informatica.com, «Integración de datos,» [En línea]. Available: <https://www.informatica.com/es/products/data-integration.html>.
- [4] J. E. C. Rugel, «Análisis de la Replicación de Bases de Datos Relaciones Caso Práctico,» 2005.
- [5] P. G. C. Marcelo, «Análisis, diseño, desarrollo e implementación de un ERP (Enterprise Resource Planning) "ACSOFT" de los módulos administrativo y contable para la empresa Disprolim dedicada al sector institucional y químico,» 2012.
- [6] C. M. E. Adolfo, «Diseño y construcción de un E-Commerce para ventas virtuales de CD`s y DVD`s,» 2003.
- [7] c. d. á. y. r. d. u. e. g. c. t. i. t. i. y. t. (. b. o. s. Solución informática para la automatización de los procesos, «Matute Rivera Soroya Gabriela,» 2013.
- [8] Javier Alfonso Valdés, Ángela Gloria Gómez Peña, Luis Alberto Pimentel Gonzáles, «Modulo para la transmisión de datos de gran tamaño para el sistema Reko,» 2011.
- [9] IBM, WebSphere Application Server V7 Messaging Administration, 2005.
- [10] «opensuse,» [En línea]. Available: http://es.opensuse.org/Software_libre_y_de_c%C3%B3digo_abierto.

- [11] Dolores Cuadra, Elena Castro y Paloma Martínez, Diseño de Base de Datos, Centro Nacional de Información y Comunicación Educativa, 2004.
- [12] C. P. Patricio Letelier, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 2006.
- [13] Jeffries R., Anderson A., Hendrickson C, Extreme Programming Installed, Addison-Wesley, 2001.
- [14] K. Beck, Extreme Programming Explained. Embrace Change, Pearson Education, 2000.
- [15] Carmen de Pablos, José Joaquín López-Hermoso, Santiago Martin-Romo, Sonia Medina, Informática y comunicaciones en la empresa, 2004.
- [16] «Elite System Consultoria en Informática,» 19 Septiembre 2016. [En línea]. Available: <http://www.elitesystem.ec/index.php/que-es-adempiere>.
- [17] Julián Pérez Porto, María Merino, «Definicion.de,» 2013. [En línea]. Available: <http://definicion.de/plugin/>.
- [18] Oracle, «docs.oracle,» [En línea]. Available: <https://docs.oracle.com/javase/tutorial/deployment/jar/>.
- [19] L. Carábula, Sistemas Operativos III, 2013.
- [20] A. E. d. I. E. D. (adigital), Libro blanco del comercio electrónico, 2012.
- [21] «Prestashop,» [En línea]. Available: <https://www.prestashop.com/es/faq>.
- [22] S. Rowles, Introduction to JMS and XMS Application Programming, IBM.

- [23] J. S. Asenjo, Aplicaciones Web, 2012.
- [24] G. C. Roque, Principios básicos de Informática, Madrid, 2007.
- [25] S. L. Mora, Programación de aplicaciones web, 2002.
- [26] Marco Besteiro, Miguel Rodríguez, «Universidad del País Vasco,» [En línea]. Available:
<http://www.ehu.eus/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>.
- [27] R. N. Marset, Modelado, Diseño e Implementación de Servicios Web, 2007.
- [28] C. Mateu, Desarrollo de Aplicaciones Web, Barcelona: UOC, 2004.
- [29] Alfredo Reino Romero, Serviweb, Ismael Zori, Manual de XML.
- [30] Patricio Letelier, Carmen Penadés, «Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» 2006.

ANEXOS

Anexo 1. MANUAL DEL USUARIO

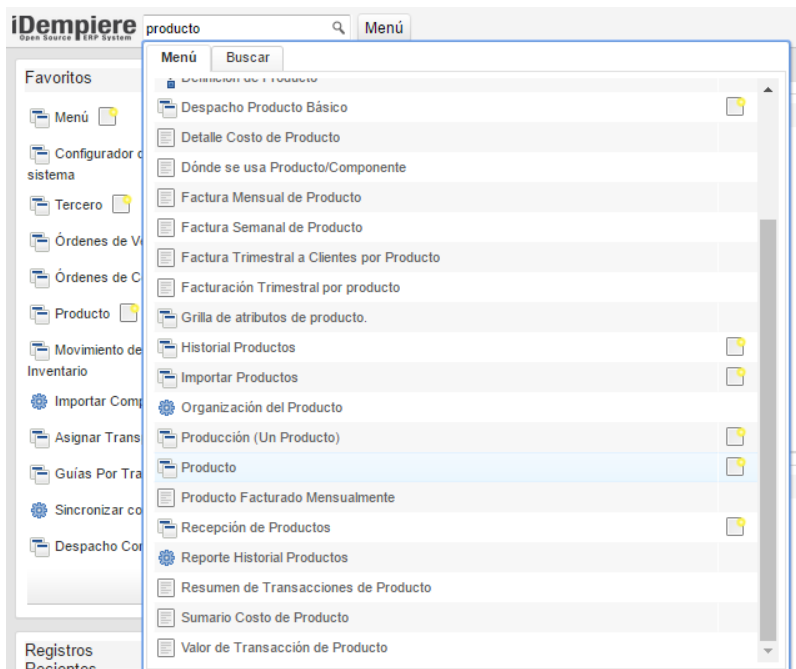
Este manual tiene el objetivo de proporcionar a los usuarios encargados de pedidos y publicación de productos en la tienda virtual, información sobre el módulo integrado a Atix ERP que se encarga de realizar estas acciones.

Antes de iniciar se debe tomar en cuenta:

- Una vez que el usuario haya decidido actualizar los productos o agregar nuevos, estos cambios serán visualizados en la tienda virtual a las 12 am del día que se realice la acción, no es inmediato debido que los cambios no deben ser vistos por los usuarios de la tienda virtual.
- Al momento de leer pedidos, éstos serán traídos inmediatamente desde la tienda, el tiempo de lectura puede variar dependiendo de la cantidad.
- No se leerán pedidos ya registrados, solo nuevos hasta el momento de la petición.
- Es posible ejecutar esta tarea en segundo plano, si así lo desea, Atix ERP notificará al terminar el proceso en la pantalla de inicio o en su correo electrónico.
- Los pedidos que tengan clientes no registrados en Atix serán ignorados.
- Las imágenes deben estar previamente cargadas en el servidor dispuesto para este motivo.

1. Cómo establecer que productos van a ser mostrados en la tienda virtual.

Una vez ya ingresados en el sistema ATIX ERP, se procede a buscar en el menú la ventana de “Producto”.

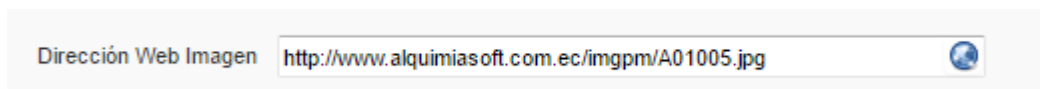


Una vez ingresado, se procede a buscar el producto a sincronizar, es posible buscar mediante código, nombre o descripción.

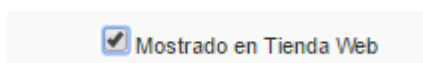
The image shows a search form titled 'Encontrar registro' (Find record) with an 'Avanzado' (Advanced) tab. The form contains three input fields: 'Código' (Code) with the value 'A01003', 'Nombre' (Name), and 'Descripción' (Description). Below the input fields is a large empty area for search results. At the bottom of the form, there are two buttons: a green checkmark button and a red 'X' button.

Al encontrar el producto, se procede a verificar que tenga una imagen asociada y que toda su información se encuentre correcta.

El campo de la imagen se puede establecer en el campo “Dirección Web de la Imagen”. Esta dirección debe tener un formato similar al del ejemplo.



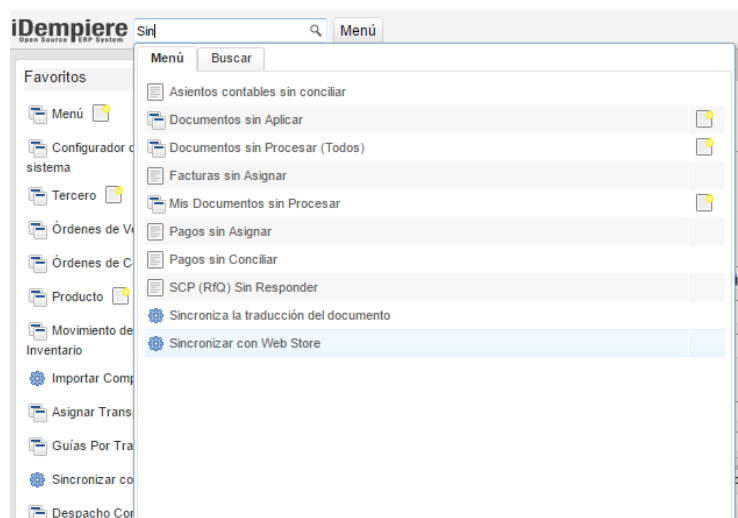
Una vez verificada la información, el campo para especificar que el producto va a ser mostrado en la tienda virtual se llama “Mostrado en la tienda web”. Se debe proceder a activarlo.



Este proceso es importante debido que se puede verificar previamente los productos que se van a enviar a la tienda virtual y que su información es correcta.

2. Sincronizar productos y pedidos.

Para realizar esta acción, en el menú se busca la ventana “Sincronizar con Web Store”.



Se presentará la siguiente pantalla con las siguientes opciones:

Sincroniza los datos del Atix ERP con el Web Store

Sincronización de productos, y clientes y pedidos.

Sincronizar Productos



Enviar todos los productos

Producto

Sincronizar Pedidos

Actualizar Información de los Productos

Ejecutar como tarea en segundo plano

Parámetros Guardados  

- Sincronizar Productos: Habilita las opciones de enviar todos los productos o enviar un solo producto.
 - Enviar todos los Productos: Envía todos los productos que previamente hayan sido marcados, con la opción de “Mostrado en la tienda web”.
 - Producto: Mediante una ventana de búsqueda, permite seleccionar el producto que únicamente va a ser enviado a la tienda virtual:

Información de Producto

Código Nombre UPC/EAN Todos los criterios

Número de Referencia (SKU) Almacén Proveedor

Conjunto de Versión de Lista

D	Código	Nombre	Proveedor
<input type="checkbox"/>	D04475	IM DISCO DIAMANTE DHINO 4 1/2 CONTIN 115MM	Consumidor F
<input type="checkbox"/>	T19440	IM EMT TUBO 1/2*1.07MM*3MT GALVANIZADO	Consumidor F
<input type="checkbox"/>	C03121AM	CABLE CABLEC INSTALACION AUTOM #18 AWG GPT 600V AM	CABLES ELEC
<input checked="" type="checkbox"/>	C04806	ABRO CINTA ELECTRICA VERDE 10Y 19MM ET-912 (500)	ABRODESIVC
<input type="checkbox"/>	L13199	FV VALVULA ESFERICA 1/2 0655 10 13 CR LIVIANA	F.V. AREA AN
<input type="checkbox"/>	C04018	CABLE CABLEC TTU 4 AWG 7H CA04	CABLES ELEC

1 / 278 [1 - 25 / 6944]

- Sincronizar Pedidos: Lee todos los pedidos nuevos en la tienda virtual y los ingresa en Atix ERP.
- Actualizar información de los productos: Envía información de los productos actualizados con sus nuevos valores.
- Ejecutar como tarea en segundo plano: Corre el proceso ocultando la ventana y notificando al usuario al momento de terminar, las opciones son:

Tipo de Notificación

- Aviso
- E-Mail**
- E-Mail+Aviso

- Aviso: Muestra el mensaje en la pantalla principal de ATIX ERP.
- Email: Envía un correo electrónico al usuario notificando que la tarea ha sido terminada.
- Email +Aviso: Combina las alertas de Email y Aviso.

Una vez seleccionadas las tareas a realizar se procede a dar clic en el botón “Ok” ubicado en la parte inferior derecha.