

UNIVERSIDAD CATÓLICA SEDES SAPIENTIAE
FACULTAD DE INGENIERÍA



Implementación de aplicativo batch para la integración B2B en una
entidad bancaria, Lima, 2021

**TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR EL
TÍTULO PROFESIONAL DE INGENIERO INFORMÁTICO**

AUTOR

Jean Carlos José Yataco Saquicuray

REVISOR

Joel Benigno López Del Mar

Lima, Perú

2021

RESUMEN

El objetivo central de la investigación fue implementar un aplicativo batch para la integración Business to Business en un banco comercial, Lima-2021. Por consiguiente, este estudio tiene como diseño el descriptivo, exponiendo los resultados a través de la identificación de dos muestras, siendo la “muestra A” el fichero entregado por el cliente que es transformado al estándar del banco, “la muestra B” el fichero ya transformado y que es procesado por la plataforma B2B. Así mismo, se elaboraron escenarios de prueba partiendo desde antes que se incorpore el aplicativo hasta su inclusión de esta. Posteriormente, la inclusión del aplicativo fue exitosa porque logró integrarse al canal B2B, partiendo del esquema de desarrollo Quartz que a través de los componentes de su arquitectura brindaron automatización a los procesos de pago del banco; en cuanto a las comunicaciones a base de datos y servidores del banco fueron seguras a través del protocolo SFTP y JDBC. Por consiguiente, para las pruebas se utilizaron lotes de pago con hasta 9999 registros de proveedores teniendo como resultado los ficheros con el formato original que requería el cliente, y posteriormente brindando su conformidad. Estos detalles validan la integración, adaptabilidad y escalabilidad del aplicativo bancario.

PALABRAS CLAVE

- Aplicativo Batch
- Ingeniería Informática
- Marco de trabajo o Framework

ABSTRACT

The main objective of the research was to implement a batch application for Business to Business integration in a commercial bank, Lima-2021. Therefore, this study has a descriptive design, exposing the results through the identification of two samples, "sample A" being the file delivered by the client that is transformed to the bank's standard and "sample B" the file already transformed and processed by the B2B platform. Likewise, test scenarios were developed starting from before the application was incorporated until its inclusion. Subsequently, the inclusion of the application was successful because it managed to integrate into the B2B channel, based on the Quartz development scheme that, through its architecture components, provided automation to the bank's payment processes; As for the communications to the bank's database and servers, they were secure through the SFTP and JDBC protocol. Therefore, for the tests, payment batches were used with up to 9999 supplier records, resulting in the files with the original format required by the client, and subsequently providing their agreement. These details validate the integration, adaptability and scalability of the batch application.

INDICE

RESUMEN	2
PALABRAS CLAVE	2
ABSTRACT.....	3
INDICE	4
I. INTRODUCCIÓN	5
1. Antecedentes y Fundamentación Científica	5
2. Justificación de la Investigación	17
3. Problema.....	19
4. Conceptuación de las Variables.....	20
5. Objetivos	21
II. METODOLOGÍA DE LA INVESTIGACION	22
2.1. Tipo de Estudio.....	22
2.2. Diseño de Investigación	22
2.3. Método de Investigación.....	22
III. METODOLOGIA DE LA SOLUCION DEL PROBLEMA.....	23
3.1 Análisis Situacional	23
3.2. Alternativa de Solución	23
<i>Spring Batch</i>	23
3.3. Solución del Problema.....	25
3.4. Recursos requeridos.....	44
IV. ANALISIS Y PRESENTACION DE RESULTADO	47
V. CONCLUSIONES	62
VI. RECOMENDACIONES.....	63
VII. REFERENCIAS BIBLIOGRÁFICAS.....	64
ANEXOS	69

I. INTRODUCCIÓN

1. Antecedentes y Fundamentación Científica

1.1. Realidad Problemática

Actualmente, las empresas bancarias tienen informatizadas sus procesos de pagos a través de sus plataformas o sistemas web; por ello, se busca que sean adaptables, seguras y autónomas para la satisfacción y confianza del cliente.

En Sudamérica, la adaptabilidad de los aplicativos y plataformas Web de pago bancarias son vitales por la cantidad de clientes que poseen; además, que para que su desarrollo sea óptimo utilizan frameworks como Spring. Por consiguiente; los bancos buscan que sus sistemas tengan escalabilidad o adaptabilidad a las distintas tecnologías de sus clientes; Villegas (2018) en su tesis define a la escalabilidad como el valor agregado que adquiere la plataforma para un correcto funcionamiento dejando de lado los cambios en la demanda por parte de los clientes. Además, cabe mencionar que muchos de los clientes poseen sus propios tipos de sistemas informáticos y sus propios sistemas de pago a través de ficheros. Entonces, se considera implementar un aplicativo que pueda integrarse, comunicarse y complementarse con el sistema web del cliente; mejorando así los procesos de pago y la calidad del producto software.

En cuanto al uso del framework llamado Spring, esta es muy útil al momento de elaborar nuestras aplicaciones adaptables y automatizables con su módulo Spring Batch o lotes de pago; es así como Méndez (2016), define el framework Spring como un patrón de programación que da soporte al desarrollo de aplicaciones empresariales en el lenguaje de programación Java; el cual nació como una opción ligera a la compleja plataforma J2EE y ha ganado popularidad entre los programadores rápidamente. Además, las tecnologías seleccionadas para desarrollar un

aplicativo cumplen un papel importante a la hora de evaluar el nivel de calidad del producto software; es así como lo explican Piattini y Garzas (2010) que el proceso software para desarrollar y mantener un producto o servicio desempea por lo tanto un rol fundamental a la hora de determinar el nivel de calidad del producto o servicio, pero tambien en el coste de desarrollarlo, darle soporte y mantenerlo. Por ultimo, la calidad del aplicativo es importante para la satisfaccion y fidelidad del cliente; por ello, Calero, Moraga y Piattini (2010) consideran que las empresas buscan hacer la diferencia frente a sus competidores con la calidad de sus productos que ofrecen a sus clientes.

En el Peru, muchas empresas bancarias contratan los servicios de las Consultoras tecnologicas para desarrollar o dar mantenimiento a sus aplicativos, sistemas o plataformas informaticas de pago. Por consiguiente, los bancos buscan que sus sistemas de pago sean seguros y adaptables; por lo cual, si hay plataformas existentes con infraestructura de seguridad ya implementada y que ademas establece confianza con los clientes, las mejoras u optimizaciones de estos softwares deben seguir la linea de seguridad ya elaborada o desarrollada. Asimismo; hay que mencionar que las entidades bancarias poseen informacion confidencial y delicada de los multiples productos que ofrece la banca; por lo cual, los aplicativos de pago; son desarrollados con el marco de trabajo o framework Spring o Quartz. Entonces, en el presente informe se implementara un aplicativo que, mediante procesos ejecute lotes de pago representados por archivos o ficheros en un determinado tiempo, para mejorar la calidad y adaptabilidad de la plataforma Web que brinda el servicio de pagos a proveedores llamada Business to Bussines (B2B), el cual es parte de una entidad bancaria.

Para poder lograr la integracion del aplicativo con el producto software del banco (plataforma B2B), primero se eligio el modelo de arquitectura Plugin; ademas, el aplicativo

bachero utiliza la tecnología informática llamada Quartz que tiene como funcionalidad la calendarización o proyección de tareas, el cual dicha tecnología cumple con los requerimientos de la plataforma y su cliente. Por otro lado, se realizó la implementación de un servidor SFTP para alojar los ficheros brindados por el cliente y que serán extraídos por la plataforma B2B. Por lo tanto, este informe permite evaluar la eficiencia del aplicativo batch y la plataforma Business a través de los resultados o ficheros exitosos.

1.2. Antecedentes

API de servicios REST extensible dinámicamente basada en arquitectura Plug-in para procesamiento de imágenes recibidas desde dispositivos móviles

Fortes (2016), en su investigación de origen internacional propone implementar un API Rest basada en la arquitectura Plug-in para el procesamiento de imágenes en dispositivos móviles. En primer lugar, se aborda una problemática no muy común en el ambiente tecnológico actual que tiene como origen el intentar solucionar problemas que necesitan de cálculo y algoritmia compleja desde dispositivos móviles. Además, hay que mencionar que las características hardware de estos dispositivos no son los ideales para procesamientos exigentes en términos de procesador y memoria; por otro lado, si son idóneos al momento de brindar distintas aplicaciones de gran utilidad al usuario y también resaltar el uso de los componentes integrados al dispositivo como la cámara, GPS u otros. En base a lo mencionado, uno de los objetivos del proyecto ha sido la interoperabilidad total entre sistemas y tecnologías; además, se escogió el protocolo HTTP como medio de comunicación entre el cliente y el servidor; por ello, el móvil se ha simulado con una herramienta tecnológica que utiliza el protocolo mencionado anteriormente. En cuanto la aplicación de servidor, se ofrece una API de servicios web REST para la comunicación asíncrona con el cliente móvil y, además, brinda un diseño extensible dinámicamente, mediante una

arquitectura que se basa en el patrón de diseño de software Plug-in, para que los algoritmos de procesamiento de imagen puedan ser enlazados a la aplicación de servidor en tiempo real o de ejecución sin cambios en el código.

En cuanto a los resultados o ejecución de experimentos, esta ha sido satisfactoria de la aplicación Back-End desarrollada, el cual ha sido capaz de comunicarse con un cliente tecnológicamente no creíble como Fiddler, logrando la interoperabilidad que se buscaba mediante tecnologías conocidas; se ha logrado extenderse también la funcionalidad del aplicativo sin recompilaciones y en tiempo de ejecución solo agregando nuevos algoritmos compilados en un directorio específico y cambiando un parámetro en el XML externo de configuración y finalmente se ha comprobado como los datos entre el cliente y el servidor han sido los mínimos necesarios, sin la existencia overhead producido por metadatos excesivos agregados para el control o formato de las comunicaciones.

Implementación de un Sistema de Información para Controlar Solicitudes de Servicios Administrativos para una Entidad Financiera Lima - 2017

Dominguez (2017), en su tesis nacional tiene como objetivo implementar una aplicación informática con la ayuda de los recursos del ERP EasyWin y del lenguaje de programación Visual Basic .NET, sin dejar de lado su compatibilidad con el ERP antes mencionado y la base de datos implementada en Microsoft SQL Server 2014. El autor nos detalla la implementación de un sistema informático que controla las solicitudes de servicios académicos en una compañía financiera, el cual de esta manera apoya a los usuarios a gestionar los recursos que se brindarán para la ejecución de los servicios programados; teniendo como fin dar a los clientes una mejor atención solucionando inconformidades y obteniendo la fidelidad de estos. Así mismo, la

compañía financiera administrará la información utilizando el ERP EasyWin; también, se tiene la herramienta Nexus y CRM para la gestión de clientes pero que no tiene un módulo para la gestión de servicios académicos. Hay que mencionar que Nexus no es compatible con las bases de datos que ejecuta el ERP y que, como consecuencia, son independientes. En cuanto a la metodología de desarrollo se utiliza la metodología RAD o desarrollo rápido de aplicaciones, la cual consiste en la rápida construcción de software basada en componentes y permite construir una aplicación de buena y alto estándar de calidad. Además, este modelo RAD comprende de una fase de modelado de gestión, modelado de data, creación de aplicaciones y pruebas de entrega; así mismo, sus características son equipos híbridos o mixtos, herramientas especializadas y modelos iterativos. Por consiguiente, se tiene el uso de un equipo híbrido donde la colaboración desarrollador y usuario es recurrente y la fecha de entrega del producto es reducido.

En cuanto a los resultados, el manejo de solicitudes de servicios académicos ahora se realiza de manera automatizada, adquiriendo fiabilidad y rapidez en el uso de la data que se controla. Además, la aplicación se ha implementado utilizando los recursos humanos y informáticos propios de la entidad financiera, por tal motivo, el resultado de la implementación ha reducido costos y tiempo. Finalmente, la configuración del nuevo sistema utilizando el ERP EasyWin con su componente EasyPro realiza una integración rápida y fácil, el cual evita los errores de ejecución de la aplicación informática por su compatibilidad.

Integración de sistemas con ERP SAP y su impacto en la satisfacción de los clientes de Decor Center, 2017

Vilchez (2019) en su objetivo de la investigación nacional busca determinar el impacto de la unión de un sistema del exterior con el Sistema Empresarial SAP en la satisfacción de los

compradores de Decor Center S.A. en su proceso de valor y oferta apuntando al año 2017; por tal motivo, la muestra de estudio y análisis es el proceso de valoración y venta de la entidad Decor S.A. En cuanto a la metodología implementada en la exploración fue el hipótesis-deductivo y explicativo. Por consiguiente, se usó para su propósito el planteamiento pre-experimental, el mismo que extrajo la información en un tiempo específico que se plasmó al aplicar la herramienta: temario para el proyecto de unión de una estructura informática externa con el ERP, y su huella en el agrado a los usuarios de la Sociedad Anónima Center en el 2017; este cuestionario estuvo elaborado con catorce interrogantes con progresión de Likert, en el mismo los clientes de la entidad Decor otorgaron los suficientes indicadores de las variables unificación y complacencia del usuario, todo en base a la evaluación de sus distintas propiedades, el cual sus frutos se presentan gráficamente y de forma textual. En cuanto a la validez de la herramienta recolectora de data cualitativa, que en este escenario serían las preguntas, se aplicó el llamado “juicio de expertos”; realizado por profesionales con grado académico de doctor y maestría de la universidad para su revisión y calificación. Por consiguiente, los profesionales expertos validaron la claridad, relevancia y pertinencia de los ítems correspondientes a cada propiedad de los objetos de exploración. Así mismo, para los resultados se realizó la prueba de normalidad, el análisis descriptivo, el pre y pro test de objetos y las pruebas de hipótesis, conformadas por la prueba de hipótesis general, la de dimensión de fiabilidad, dimensión comunicación y dimensión capacidad de respuesta.

En cuanto a las conclusiones, primero se determina que no hay impresión en el gozo de los compradores de Center Sociedad A. en el periodo de estudio, otorgado un nivel de significancia idéntico a 0.051 el mismo que es inmejorable al p valor ($\text{Sig. } 0.051 > 0.05$); además, en el desenlace hay impresión para la unificación de un sistema del exterior con el Sistema de Planificación en la

magnitud de fiabilidad, en la dimensión de comunicación y en la propiedad de nivel de respuesta del usuario Decor Centro.

Metodología ágil Iconix en la calidad del producto software

Flores (2019), en su estudio de origen nacional tiene como finalidad optimizar la calidad del producto informático, adaptando la norma ágil ICONIX en el ciclo de análisis de requisitos, diseño y desarrollo, a través de métodos probados de ingeniería de software, el cual la tesis se realiza en la capital peruana. Así mismo, el tipo o método de investigación es prospectivo, analítico, modo longitudinal y de nivel aplicativo. En cuanto a las variables, la variable independiente es la disciplina Iconix (v1) y la variable número dos es la condición del software (v2). Por el lado de las técnicas de investigación, se usaron las técnicas de encuesta para recopilar información y evaluar los procesos Iconix; el análisis documental para obtener información sobre métricas en referencia a las reglas NTP ISO-IEC y el análisis estadístico para las mediciones de los atributos de índole del artículo tecnológico. También, los instrumentos de investigación para evaluar están en línea a la metodología ágil y la estructura de clase adaptada por la Norma técnica Peruana ISO e Indecopi. En cuanto a los procedimientos, se realiza la estrategia de test hipotética en el cual se formula la hipótesis nula (H_0) y la hipótesis alterna (H_a); también se realiza técnicas de procesamiento de datos, diseño estadístico, técnicas de análisis y traducción de la data abstraída de los instrumentos de investigación antes mencionados. Por consiguiente, se utilizó como diseño investigativo el cuasiexperimental, cortando las tareas de la regla ágil antes mencionada, estas tareas intervenidas con procedimientos de ingeniería programática, buscan mejorar la naturaleza del fruto de equipamiento lógico, después, se elabora documentos de preguntas a personas con expertis en el desarrollo ágil de soporte lógico, pre y post de la interferencia al criterio metodológico, la data obtenida es procesado mediante las pruebas o distribución de probabilidad

T-Student, análisis de varianza y la homocedasticidad, con el fin de contrastar las hipótesis de estudio que se formularon.

Por ello, el autor establece que la intervención a la etapa de analítica de requisitos con la adaptación de ingeniería de requisitos, la etapa de planteamiento mediante la inclusión de patrones de boceto, la etapa de implementación vía incorporación de técnicas de programación orientada a objetos con las normas Solid; mejoran significativamente la calidad o clase del producto programático.

Desarrollo de una app web para el control de las cotizaciones y las postventas de los proyectos inmobiliarios en la empresa Zoom Constructora S.A

Caicedo (2019), desde Colombia nos comenta que el sistema a desarrollar tiene como objetivo general gestionar las cotizaciones y las postventas de cada proyecto que tenga la entidad constructora llamada Zoom S.A, optimizando la búsqueda de información en documento físicos que muchas veces generan problemas al centrarse en obtener información de manera ágil y rápida; además, se busca controlar y gestionar los flujos de procesos en las tomas de decisiones al momento de publicar una campaña comercial o cuando se tenga que realizar un contrato a un proveedor para la construcción del proyecto; con lo siguiente, saber que los informes llevan a que se puedan prever situaciones de acuerdo al histórico que se puede manejar. En cuanto a la investigación, se detalla que se realizaron estudios sobre el estado del arte sobre plataformas web para la gestión de información y para determinar las funciones necesarias del aplicativo software se analizó y se diseñó con un lenguaje de modelado, ejecutando tecnologías web y elaborando las pruebas pertinentes para un correcto funcionamiento teniendo como finalidad entregar un producto de buena calidad. En cuanto a la implementación de este proyecto se utilizó la metodología de

desarrollo RUP realizando cada una de sus etapas, entregando así un software fiable, robusto, de buena calidad y fácil de usar que se adecua a las necesidades actuales y con la disposición de crecimiento si la entidad desea implementar nuevas opciones que puedan conectarse y trabajar en colaboración con las ya implementadas. Por tal motivo, se implementó una base de datos y una interfaz web que brinda la gestión y administración de la información de manera sistematizada y ágil, cabe mencionar que se aprovechó los recursos que estaban a la mano con la que contaba la constructora Zoom y obteniendo al máximo la ayuda de las personas que amablemente brindaron su colaboración para la elaboración de este proyecto.

Finalmente se concluyó, que el sistema de información implementado en esta investigación cumplió con todos los objetivos señalados y las expectativas que fueron pronosticadas para su diseño. En primer lugar, se logró mejorar el registro de las cotizaciones y postventas en la entidad Zoom. En segundo lugar, se brinda una aplicación con capacidad de guardar gran cantidad de información, con soporte estable y funcional. Por último, se logró generar un producto útil.

1.3. Fundamentación Científica

En cuanto a las bases teóricas, se ha efectuado una búsqueda de conceptos de diferentes autores con los cuales se fundamentan las variables Aplicación e Integración de sistemas.

1.3.1 Definiciones

García y Munilla (2005), define a la integración entre aplicaciones a una generación de soluciones software que nacen en los años noventa; que usan tecnología de integración propietaria para la transformación de datos y la creación de adaptadores entre aplicativos.

Roldán, Valderas y Torres (2018), define la aplicación Spring Boot como una aplicación Java clásica, ósea que su ejecución se da a través de su método llamado main.

Sánchez y Montes (2013), nos define al programa, como toda la información que engloba al código y los datos, guardada en disco de una aplicación que soluciona una necesidad específica para los usuarios. En cuanto al proceso, cuando el software entra en ejecución, diremos de una manera muy concreta que es un proceso; en resumen, el proceso es un software en funcionamiento. Además, cabe mencionar que los procesos trabajan de manera individual, así ejecuten el mismo programa; por consiguiente, convivirán múltiples procesos que ejecuten el mismo software, aunque con diferentes datos y en distintos tiempos de ejecución.

Raya, Raya y Zurdo (2014), define al sistema informático como un conjunto de componentes vinculados, un sistema común utiliza una computadora que hace uso de dispositivos programables para obtener, guardar y procesar datos, así mismo, el ordenador con el usuario que lo controla y los periféricos que están alrededor, son el ejemplo de un sistema informático. En cuanto a la estructura, este sistema se puede estructurar en partes, aunque funcionalmente indivisible; en caso contrario, de que se puede dividir, pierde alguna de sus propiedades esenciales. Por ello, un sistema o plataforma informática sin alguna de sus partes perdería su funcionalidad. Cabe mencionar que estos sistemas se componen de tres elementos básicos, el primero es un componente físico o hardware. Segundo, un componente lógico o software; el componente incluye software de base y aplicación. Por último, el componente humano que incluye a las personas en las fases de análisis y desarrollo.

Ferrer (2012), en su libro, nos brindan conceptos generales de la arquitectura de aplicativos webs, este tipo de aplicaciones usan los conocidos cliente ligeros o livianos, estos no

ejecutan cantidad de tareas de procesamiento para hacer funcionar el aplicativo. Además, desde el enfoque arquitectónico aparecen dos partes; el primero es el cliente, donde el usuario final utiliza la aplicación vía un navegador como Firefox o Explorer. Por consiguiente, desde este cliente web, el usuario interactúa con el software que está localizada en un servidor, donde se hospedan realmente los datos, reglas y lógica de la aplicación. En cuanto al concepto de integración, el autor menciona que la combinación de servicios de las computadoras, las aplicaciones y la extensión de las facultades de comunicación con las aplicativos tipo web que hacen posible Internet, son la plataforma ideal que sirve de infraestructura para los usuarios de tipo empresarial, personal o institucional.

Gisbert (2015), define el protocolo SFTP como el resultado de la ejecución de FTP sobre SSH; en síntesis, SFTP es un protocolo totalmente nuevo. Además, el autor menciona que el SFTP permita llevar a cabo el tipo de mantenimiento de archivos o ficheros básico que es útil cuando se trabaja con jerarquía de ficheros.

Talledo (2015), en su libro habla de la evolución de las aplicaciones informáticas; y nos dice, que la evolución ha estado por momentos convirtiéndose en una revolución; por ello, lo que actualmente está vigente, el día de mañana estará desfasado o mejor dicho descartado.

1.3.2 Características

López, Soltero, Sánchez, Moreno, Bollati y Vara (2016), nos detallan que una de las principales características y ventajas de las aplicaciones web, es la centralización del software y por este motivo se hacen más fácil las tareas de mantenimiento y actualización.

Talledo (2017), en otros de sus libros nos menciona que las aplicaciones informáticas poseen unos requisitos previos para su instalación, estos requisitos son definidos en lo que corresponde a configuración del hardware, sistemas operativos, controladores, versión mínima del sistema operativo, firmwares, librerías de software. Por ello, si no se cumplen con estos requisitos el fabricante o desarrollador no garantiza su buen funcionamiento.

Piattini y Garzás (2010), considera que la implementación de un sistema software requiere del uso y cooperación entre muchas tecnologías para obtener la funcionalidad especificada para el sistema; ocurre entonces con las aplicaciones que se implementan reutilizando componentes o sistemas externos. Así mismo, Kendall y Kendall (2005), nos menciona que según vayan adaptándose y difundan las nuevas tecnologías; parte de la labor de los analistas de sistemas se centrará en la integración de los sistemas tradicionales con los nuevos.

Molina y Baena (2007), nos dice que la instalación y puesta en marcha de las aplicaciones no es una tarea sencilla, así mismo, una de las fases más sensibles es el estudio de que herramienta necesita la entidad o empresa. Por consiguiente, comprobar la necesidad de la herramienta para su instalación y por último instalarla y ponerla en funcionamiento para toda la entidad.

1.3.3 Objetivos

Morales (2010), detalla que los fines de un sitio o sistema web deben estar alineados con lo que busca la empresa e integrar estas necesidades en la estrategia del negocio; por consiguiente, la forma más clara de definirlos es buscar en los objetivos de la empresa y en función de ellos se da inicio a la implementación de los objetivos.

Moreno (2012), nos define que la meta del sistema informático es el dar soporte al procesado, guardado, entrada y salida de datos que normalmente forman parte de un modelo de información general o concreto. Para ello, se adquiere una cantidad de recursos que cambian en función de la aplicación que se le brinda al mismo.

2. Justificación de la Investigación

La integración Business to Business es esencial para el rendimiento y escalabilidad de la plataforma del banco. Actualmente las entidades bancarias realizan integraciones entre sistemas con sus clientes, pero no necesariamente los sistemas de ambas partes se compenetran en su totalidad. Para estos casos los sistemas o plataformas bancarias deben ser adaptables a estos pequeños detalles pero que pueden dar grandes problemas en el futuro.

Por consiguiente, se buscó solucionar el problema de integración de una plataforma existente con el sistema externo del cliente; por tal motivo se puede considerar una investigación práctica.

La investigación tiene como objetivo implementar un aplicativo que pueda realizar la integración de un nuevo sistema con la plataforma existente llamada B2B sin generar impacto alguno, ya que la modificación en la actual plataforma generaba un gran impacto en el flujo de los procesos de pagos, el cual incluía modificaciones en las validaciones de lotes de pago y encriptación de lotes; además, los clientes ya adaptados a esta plataforma también eran afectados.

Las implicancias de este estudio es que brindará la solución al problema de integración con un sistema SAP, teniendo en cuenta la adaptabilidad, confiabilidad, alcance y seguridad; que optimizará la plataforma B2B del banco.

En cuanto a la arquitectura de desarrollo, se aplicó un modelo de arquitectura Plug-in para una rápida integración del aplicativo con la plataforma, este nuevo componente realiza la transformación de estructura de los ficheros que utiliza el cliente; además este modelo se acomoda a las políticas de seguridad del Banco y no genera impacto alguno en el flujo de procesos de pago.

El presente estudio tiene gran importancia en el crecimiento y alcance de la plataforma B2B, ya que una vez implementada se podrá agregar funcionalidades en la estructura del código a nivel aplicativo para los nuevos sistemas de los futuros clientes. Por consiguiente, la arquitectura base del marco Quartz usada en el aplicativo, brinda la capacidad de integración y comunicación con el B2B ya que ambos procesan ficheros en un determinado tiempo.

Por otro lado, en la investigación se tiene como pilar fundamental la seguridad informática, pues los beneficiados serán el cliente y el banco; por el lado del cliente se brindará un único acceso para ingresar a la red bancaria y por el lado del cliente, se implementará un “firewall” o mejor llamado cortafuegos para mayor seguridad en los pedidos de información de datos y traslado de ficheros.

En cuanto a la satisfacción, fidelidad y empatía con el cliente, si requieren realizar algún cambio en sus procesos de pago a sus proveedores, brindarán los requerimientos con la confianza de que la modificación e integración será factible.

Por consiguiente, en la actualidad las empresas buscan adaptar nuevos modelos arquitectónicos que tengan la capacidad de adaptarse o acoplarse rápidamente a lo que demande el cliente; pues, lo que se evita es querer molestar al cliente proponiéndoles que sus sistemas tengan ligeras modificaciones. Por ello, se está buscando implementar modelos arquitectónicos que

puedan integrarse se forma eficiente y así generar plataformas óptimas con los que los clientes se sientan satisfechos.

3. Problema

3.1. Problema General

¿Cómo la implementación del aplicativo batch integra el Business to Business en una entidad bancaria, Lima - 2021?

3.2. Problema Específico

¿Cómo la implementación del modelo Quartz Scheduler integra el Business to Business en una entidad bancaria, Lima - 2021?

¿Cómo la implementación de procesos batch integra el Business to Business en una entidad bancaria, Lima - 2021?

¿Cómo la implementación del servicio de transferencia de ficheros integra el Business to Business en una entidad bancaria, Lima - 2021?

¿Cómo la implementación del framework MyBatis integra el Business to Business en una entidad bancaria, Lima - 2021?

4. Conceptuación de las Variables

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES
IMPLEMENTACIÓN DE APLICATIVO BATCH PARA LA INTEGRACIÓN BUSINESS TO
BUSINESS EN UNA ENTIDAD BANCARIA, LIMA, 2021

Variable	Definición conceptual	Definición Operacional	Dimensiones	Escala de medición
V. 1. Aplicativo Batch	Son aplicaciones conducidas por los datos. (lan Sommerville, 2005).	Es la tecnología informática basada en procesos automatizados.	Modelo Quartz Scheduler	Ordinal
			Procesos batch Servicio de transferencia de ficheros Framework MyBatis	
V. 2. Integración Business to Business	La integración de aplicaciones es una generación de soluciones software que usan tecnología de integración para la transformación de datos y la creación de adaptadores entre aplicativos. (García & Munilla, 2005).	Resultado general sobre la adaptación del sistema del cliente.	Empatía satisfacción Confianza	Ordinal

5. Objetivos

5.1. Objetivo General

Implementar un aplicativo batch en el Business to Business de una entidad bancaria, Lima -2021.

5.2. Objetivo Específico

Implementar el modelo Quartz Scheduler para integrar el Business to Business en una entidad bancaria, Lima -2021.

Implementar procesos batch para integrar el Business to Business en una entidad bancaria, Lima -2021.

Implementar un servicio de transferencia de ficheros para integrar el Business to Business en una entidad bancaria, Lima -2021.

Implementar el framework MyBatis para integrar el Business to Business en una entidad bancaria, Lima -2021.

II. METODOLOGÍA DE LA INVESTIGACION

2.1. Tipo de Estudio

En el presente informe, el tipo de estudio es aplicada debido a la evaluación de la eficiencia del aplicativo o variable independiente, ya que se identificó que la plataforma existente del banco no tiene la capacidad de adaptarse al sistema del nuevo cliente. Por un lado, el sistema no posee las herramientas que requiere el cliente para procesar sus lotes de pago y tampoco la arquitectura implementada, además de que se busca la adaptación del sistema hacia el cliente y no lo contrario.

Ñaupas, Mejía, Novoa y Villagómez (2014), consideran que la presente investigación es tecnológica porque su producto no es conocimiento puro, científico sino tecnológico.

2.2. Diseño de Investigación

El diseño de investigación es descriptivo, ya que se detalla la situación del canal o plataforma bancaria, los motivos de no poderse integrar con el nuevo sistema y se describe la infraestructura actual que posee la plataforma para la obtención de resultados.

2.3. Método de Investigación

Se aplica el método descriptivo. Por un lado, se tiene la variable X: Implementación de aplicativo batch y la variable Y: integración B2B, la variable x es implementada y verificada en los entornos de prueba; y en cuanto a la variable y, se mide el alcance que poseen sus funcionalidades, tecnologías alojadas y los recursos que consume el B2B.

III. METODOLOGIA DE LA SOLUCION DEL PROBLEMA

3.1 Análisis Situacional

Actualmente el banco comercial - Sede Perú de la presente investigación, tiene como visión ganarse la fidelidad y confianza de los clientes, accionistas, empleados y de la sociedad. Por consiguiente, esto se ve reflejado en la gran relación que tiene con sus clientes, además, los procesos de los contratos establecidos, la información que se les brinda a los clientes, están correctamente implementadas teniendo como consecuencia la estrecha confianza de los productos que brinda el banco internacional del presente informe. Las plataformas o canales informáticos esta correctamente implementadas a nivel de desarrollo y seguridad, cabe mencionar que los procesos de pagos a proveedores cuentan con un mantenimiento constante para frenar las posibles incidencias con el cliente; además, de ser automatizables; también, las plataformas poseen interfaces graficas actualizadas y muy interactivas que facilitan el uso de esta; bajo estos detalles, los resultados son la confianza y fiabilidad de parte del cliente.

Por otro lado, constantemente la empresa bancaria está optimizando sus plataformas principales, adicionándoles nuevas tecnologías, esto a nivel de software de seguridad y de desarrollo. Por consiguiente, en el presente año se integró un consorcio empresarial, el cual posee un sistema informático que tiene dificultades para acoplarse a la red de proyectos que brindan el proceso de pagos; así mismo, se planteó la solución requerida.

3.2. Alternativa de Solución

Spring Batch

En cuanto a la alternativa de solución, se propuso implementar el aplicativo por lotes con la tecnología Spring Batch del framework ¹Spring.

Walls (2015), dice que este marco de trabajo por lotes es muy utilizado para desarrollar aplicaciones sólidas y robustas por lotes; siendo primordiales para las operaciones del día a día de los sistemas empresariales. Spring batch posee funciones reutilizables que son importantes en el procesado de gran cantidad de datos, esto incluye la administración de transacciones, las estadísticas de los procesos de trabajo, el reinicio de estas, la omisión y la gestión de recursos. Además, brinda funciones y servicios técnicos eficientes que permitirán trabajos por lotes de gran volumen y buen rendimiento con sus técnicas de particionamiento del framework. Finalmente, algunas ventajas y características del framework son:

Robustez y fiabilidad: Rastreo constante de lo que se realiza por parte de los componentes y el framework brinda los datos de ejecución al reiniciar los trabajos (Jobs) en el escenario de un error.

Estrategias de escala: El Spring batch procesa elementos particionados o mejor llamado “in chunks”.

Gradle

Es una herramienta tecnológica con la cual podemos automatizar la compilación del código, esta herramienta es un gestor de dependencias estable y trabaja con scripts de compilación que se escriben con el uso de Groovy o Kotlin DSL². Además, Gradle tiene la capacidad de

¹ Marco de trabajo para el desarrollo de aplicaciones.

² Lenguajes de programación.

recompilar las clases que han sido modificadas en el código fuente o también si se ha modificado las rutas de estas.

Así mismo Roldan, Valderas y Torres (2018) nos dicen que Gradle hace que la construcción de nuestro código sea automático y que esta herramienta al ser comparada en los aportes de otras herramientas tecnológicas como Maven o Ant; Gradle es más óptima brindando un lenguaje más simple y claro al momento de especificar la construcción a realizar y de soportar múltiples lenguajes de programación siendo una de ellas Java.

3.3. Solución del Problema

Como solución al problema se implementó un aplicativo de tipo bachero utilizando el entorno de desarrollo Eclipse, el lenguaje de programación usado fue Java; para el proceso de desarrollo, se utilizó el framework Spring con la tecnología Spring Boot y se utilizó como metodología de solución el framework Quartz para programar los procesos batch divididos por trabajos o Jobs³.

En cuanto a la metodología, usamos el esquema Quartz Scheduler para desarrollar cuatro Jobs automatizables que procesan lotes de pago de tipo Confirming cerrado, el primer Job se encarga de las transformaciones o conversiones de la estructura de los ficheros que otorga el cliente, esta internamente valida los campos de estructura del archivo y si es correcta, ingresará a la carpeta donde el Job número dos empieza a ejecutarse, pero en caso el fichero no cumpla con las propiedades establecidas por el Job, esta realiza la devolución del fichero con su respectivo mensaje de error.

³ Componentes de ejecución.

Para el Job número dos, esta se activa cuando detecta un fichero procesado correctamente por el Job transformador en la carpeta configurada por el archivo de propiedades del aplicativo. Este Job es primordial ya que es el nexo entre el aplicativo Batch y la plataforma B2B porque este Job tiene la responsabilidad de dejar los ficheros en la carpeta que apunta el Business to Business y luego se encarga de obtener los ficheros ya procesados por la plataforma.

En cuanto al tercer Job, esta se activa cuando detecta ficheros en la carpeta que corresponde a los ya procesados por la plataforma B2B. La función de este Job es realizar una reconversión de la estructura de los ficheros hacia la estructura inicial que brindo el cliente con sus ficheros.

Finalmente, el cuarto Job procesa los ficheros que otorga el core⁴ bancario automáticamente cuando se procesan los ficheros con el tercer Job. Estos ficheros contendrán los estados de pago de cada lote.

Por otro lado, se utilizó MyBatis para la creación de queries y uso de repositorio; y se crearon tablas en una Base de datos Relacional Oracle. Cabe resaltar que la base de datos es parte de la plataforma Business y toda la información guardada en la base de datos se hace vía comunicación JDBC.

Por consiguiente, el aplicativo está alojado como archivo JAR⁵ en un servidor SFTP, donde también se alojan los ficheros del cliente. Como la muestra la Figura 1, el sistema externo del cliente es el encargado de dejar los ficheros en el servidor, esto lo realiza conectándose con credenciales únicas a la red privada virtual (VPN) del banco; este proceso es vía comunicación SFTP. Este servidor aloja los ficheros ingresados por el cliente para ser procesados con sus

⁴ Sistema utilizado en el banco.

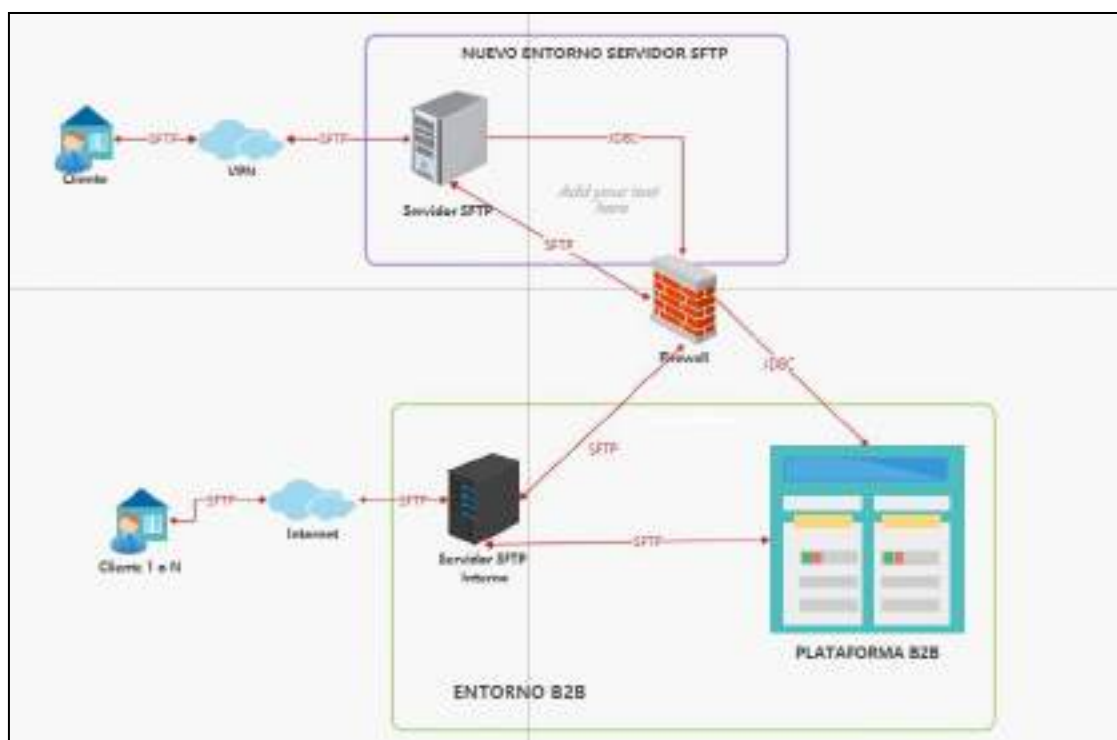
⁵ Tipo de archivo que ejecuta aplicaciones Java

respectivas validaciones, luego los ficheros son enviados a un servidor SFTP Interno del entorno del banco, a través del Firewall bancario, luego estos archivos son enviados a la plataforma B2B y finalmente esta plataforma devuelve el fichero ingresado inicialmente con sus respectivos mensajes sea satisfactorio o no al servidor SFTP, y de manera también automatizada el sistema core del banco registra el fichero ingresado, genera los ficheros que representan el proceso de pago y son enviados al servidor.

Finalmente, el aplicativo y la plataforma B2B están orquestadas y conectadas a través del modelo de arquitectura Plug-in; en cuanto a la infraestructura, arquitectura y tecnologías mencionadas se detallarán en las siguientes líneas.

Figura 1

Diagrama de Infraestructura B2B Confirming Cerrado



Nota. Elaboración propia.

Arquitectura de Solución

El modelo arquitectónico que se aplicó es el Plug-in, plugin significa extensión, el cual se refiere al componente o aplicativo que se conecta a otra para aportar una nueva función y que por lo general es muy específica. En contraste, la aplicación implementada será la que optimizará la plataforma B2Ba través de la arquitectura Plug-in, instalándose y comunicándose para realizar las transformaciones de ficheros correspondientes del producto Confirming Cerrado del banco.

Red Privada Virtual (VPN)

Este tipo de red virtual permitirá el acceso del cliente hacia el servidor SFTP de forma segura, a través de un túnel donde pasa la información y donde existen flujos de encriptación. Así mismo, Corrales, Beltrán y Guzmán (2006) definen la VPN como al sistema que simula una red privada sobre una red de tipo pública como el Internet y la comunicación se realiza a través de vías virtuales entre dos puntos donde se negocian esquemas de encriptación y autenticación que aseguran la confidencialidad de los datos.

SFTP (SSH Protocolo de Transferencia de archivos)

En el traslado de archivos remotos se utilizó SFTP. Software de traslado de ficheros o archivos alojado en un servidor y que se ejecuta a través de una conexión SSH. Por otro lado, Gisbert (2015) lo define como un programa de transferencia de archivos interactivo, esta se conecta y se registra en un host especificado para entrar en un modo interactivo como FTP (Protocolo de Transferencia de Archivos).

SSH (Cubierta Segura)

Los Shell son los programas informáticos que son primordiales para la invocación y ejecución de los diferentes programas disponibles en un ordenador. Por consiguiente, el Secure Shell es un protocolo que se utiliza en el acceso remoto a un servidor de manera cifrada. Algunas características del SSH son:

- a) Posee autenticación.
- b) Brinda confidencialidad e Integridad de la información.
- c) Posee versiones para los sistemas operativos IOS, Windows, Linux.
- d) Evita la suplantación de dominios.
- e) Permite tunelización para otros protocolos como FTP y SMTP.

Según Andreu (2010), el funcionamiento del SSH se da la siguiente manera:

- a) Apertura de conexión TCP por el puerto 22 que da inicio el cliente.
- b) Se negocia la versión SSH, el tipo de cifrado (RSA/ DSA), etc. Esto por medio del servidor y el cliente.
- c) La clave pública del servidor viaja al cliente.
- d) La lista de claves del cliente es comparada con la clave pública. Para la primera vez el usuario dará el visto bueno de validez.
- e) Creación de clave de sesión por medio del cliente, luego la clave se dirige al servidor por medio de la clave tipo pública y el cifrado con un algoritmo seleccionado.
- f) Finalmente, el algoritmo que se escogió para la encriptación se usará en la comunicación.

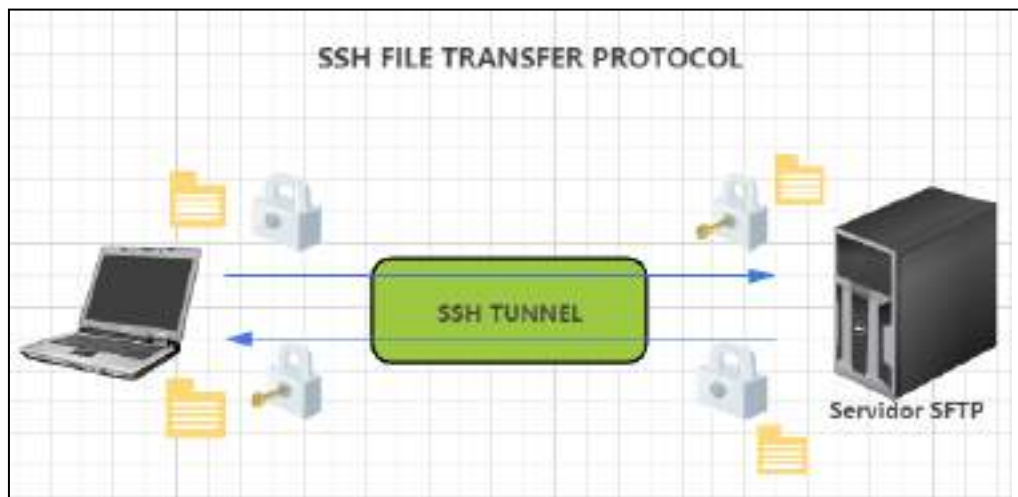
Por otra parte, los algoritmos RSA y DSA son importante en el proceso de cifrado. En cuanto a estos algoritmos Gisbert (2010), define al RSA como un sistema criptográfico de clave publica y es útil, eficiente para el cifrado y firmado digital; y DSA es un algoritmo de firma digital de tipo simétrico y un algoritmo para intercambio de claves.

Servidor SFTP

Servidor que posee el acceso SSH, como se muestra en la Figura 2. Con este protocolo los datos de acceso están disponibles para el usuario SFTP, estos datos comprenden la dirección del servidor, nombre del usuario y la contraseña. Por consiguiente, estos datos se registran en el programa SFTP del cliente, luego de esto se establece la conexión y se guarda la información si se desea; bajo este detalle, el cliente se autentica con el servidor en cada conexión.

Figura 2

Proceso de comunicación SFTP



Nota. Elaboración propia.

Base de Datos Relacional Oracle

El sistema gestor de base de datos (SGBD) relacional Oracle, almacena la data en tablas para luego ser utilizada de una manera eficiente, como es el caso de los backups o mantenimiento de tablas.

JDBC

JDBC es un API (Interfaz de Programación de aplicaciones). Como se visualiza en la Figura 3, grupo de interfaces y clases usados para la conexión al repositorio de datos y realizarlas operaciones en la misma desde el lenguaje de programación Java, estas operaciones consisten en consultas y recuperación de data. De manera similar Moreno (2011), menciona que el API JDBC brinda un acceso a base de datos relacionales desde Java, se pueden realizar consultas SQL, recuperar los datos del repositorio y realizar cambios en la misma a través del datasource u origen de datos.

Deléglise (2013), nos dice que la manipulación de datos en java mediante JDBC se realiza de forma global con los siguientes pasos:

- a) Se crea la conexión al repositorio de datos.
- b) Creación de una petición.
- c) Obtención y uso del resultado, en el caso de la sentencia SELECT.
- d) Cierre de la conexión.

Framework MYBATIS

Framework de persistencia que aguanta lenguaje de consulta estructurado (SQL) y procedimientos almacenados, el cual se puede configurar con archivos de tipo XML (Lenguaje Marcado Extensible) o anotaciones y permite mapear clases java con registros de base de datos. Así mismo, es parte de la implementación de la solución; con el uso de este framework se han implementado archivos XML que contienen las consultas queries tales como update, select e insert.

XML (Lenguaje de Marcado Extensible)

López, Soltero, Sánchez, Moreno, Bolatti y Vara (2016), definen XML como un metalenguaje de marcado, como un lenguaje que permite describir a otro lenguaje a través del uso de etiquetas, estas etiquetas se usan como definidoras de la información y también se usan para distinguirlas del contenido que se describe.

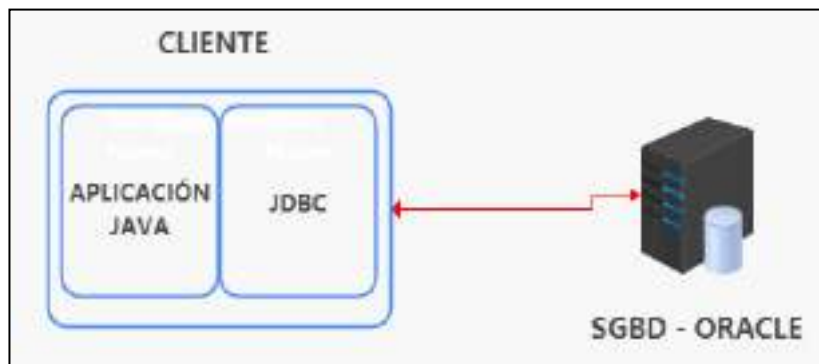
Por otro lado, Sánchez (2012), explica que el lenguaje de marcado extensible o XML es una potente herramienta para definir estructuras de datos fáciles de ser procesadas por una gran cantidad de aplicaciones para realizar un eficiente intercambio de información.

Así mismo Boulanger (2015), dice que XML es una alternativa poderosa frente al HTML ⁶para la adaptación a nuevos entornos más difíciles como comercio electrónico o mensajería, el lenguaje XML tienen como objetivo disociar los datos de su presentación; esta disocia de mejor manera los datos y su presentación; además, su propósito es almacenar información en lugar de mostrarla y posee varios archivos con una función correspondiente. En cuanto a la sintaxis de este lenguaje extensible, su archivo, es compuesto por un tipo de texto especial que se basa en etiquetas en forma de árbol.

⁶ Lenguaje de marcado para elaboración de páginas web.

Figura 3

Proceso de comunicación JDBC



Nota. Elaboración propia.

Cortafuegos

Lederkremer (2020), define el firewall como un cortafuegos que tiene origen en la seguridad industrial, el cual allí se utilizaban gruesas paredes con cámara de aire en los sitios con peligro de incendio, es así como de manera rápida se podía aislar el lugar de y las consecuencias del daño serán mínimas. Además, estas paredes de seguridad pueden ser con fines internos o externos.

Bajo estos detalles del autor, se puede decir que el objetivo de los muros de seguridad es contener el peligro que puedo tener un área con información valiosa, es decir si hubiera un área donde hubiera mucha documentación física y se estuviera incendiando; lo que se trataría de hacer es contener el fuego para que no llego a las otras áreas u oficinas. Por consiguiente, al implementar un cortafuegos estamos protegiendo nuestra red de posibles infiltraciones de atacantes o hackers y de daños a la red producidos por virus informáticos.

Por otro parte esta herramienta se puede implementar por hardware o por software. Es así como Aguilera (2010), nos dice que tanto por el lado software y hardware es aceptable la implementación de un firewall y que ambas tienen posibles políticas a la hora de aplicarse:

Política restrictiva: Esta política impide todo el tráfico con excepción del autorizado realizado en la configuración.

Política permisiva: Esta permite el paso de todo el tráfico o comunicación a excepción de las prohibidas.

DMZ (Zona desmilitarizada)

El uso de firewall puede darse entre varios servidores tales como SFTP, FTP u otros en una misma empresa, de esta manera surge el modelo de seguridad informática llamada DMZ o zona desmilitarizada.

Para Casla (2012), define la DMZ como la zona de la empresa que se va a ubicar entre la red interna y generalmente internet, esta estructura tiene como fin permitir la conectividad entre la red interna e internet con la DMZ, el cual desde la zona desmilitarizada solo se permitirá acceder a internet; y la red interna quedara protegida de los posibles ataques desde internet porque no podrán pasar de la DMZ.

Por consiguiente, Costas (2011), nos menciona que, al realizarse un diseño de red, es importante determinar que equipos brindarán servicios de tipo público y por lo tanto serán accedidos desde el exterior a la red organizativa y que equipos deben ser intocables desde el exterior para mantener un buen nivel de seguridad en las comunicaciones internas; surge así la implementación de una DMZ o zona perimetral.

JAVA

Lenguaje de programación escogido para la solución, este lenguaje fue elaborado para el desarrollo de aplicaciones y se basa en el paradigma de programación Orientada a objetos o POO, para este lenguaje los objetos de la vida real y sus características son a como una clase y sus métodos.

Así mismo, para Torres (2013) el lenguaje ha revolucionado la programación y ha promovido un código tomando como derivada de C y C++; además, en corto tiempo ha sido exitoso en las tecnologías de información por su filosofía y su forma de operar de sus aplicaciones cliente – servidor; tanto en la plataforma como a nivel web.

Ceballos (2010), nos dice que si queremos mencionar un lenguaje de programación de alto nivel ese es Java, con el cual se pueden elaborar programas tradicionales y también los que son para la Internet; de las ventajas más importantes del lenguaje comparado con otros lenguajes es la característica de ser indiferente a la plataforma, tanto del código fuente como en binario; esto quiere decir que el compilador java genera código, esta luego puede instaurarse en la plataforma que más se crea conveniente con solo tener instalada una máquina virtual java y posteriormente ejecutarse.

Por otro lado, Wanumen, Rivas y Mosquera (2017), consideran que existe un ambiente de desarrollo y un compilador, el primero es una herramienta visual para que el desarrollador introduzca su código fuente y el compilador es la herramienta que vuelve ejecutable el código y crea un archivo binario que es ejecutable en distintas plataformas, este compilador para Java es llamado JDK (Kit de desarrollo Java).

Spring Framework

Walls (2015), lo describe como un marco de trabajo de código abierto que fue creado por Rod Johnson, este se creó para para hacer frente a la complejidad de los aplicativos empresariales, haciendo uso de JavaBeans ⁷ para conseguir objetivos que anteriormente no eran posible con JavaBeans Empresarial; así mismo, cualquier aplicación puede beneficiarse de Spring, gracias a la capacidad de prueba, acoplamiento débil y su simplicidad.

Por consiguiente, Spring se divide en diversos módulos o capas como en la Figura 4. Así mismo Pérez (2015), nos dice que Spring tiene diversos módulos que brindan un amplio rango de servicios:

Programación Orientada a aspectos: Apoya con la implementación de código sesgado.

Contenedor de inversión de control: Este módulo autoriza la composición de elementos del aplicativo y la gestión de instancias de clases Java, el cual se realiza a través de la famosa inyección de dependencias.

Modelo Vista Controlador: MVC es un marco de trabajo basado en controladores, vistas y el modelo.

Ingreso a base de datos: Proporciona la interacción con distintos sistemas gestores de base de datos con distintos paradigmas tales como relacionales o NoSQL⁸.

⁷ Modelo de componente para el desarrollo de aplicaciones Java.

⁸ Base de datos no relacionales.

Acceso Remoto: A partir de tecnologías como RMI ⁹ y protocolos referenciados por HTTP ¹⁰ adjuntando Servicios web SOAP brinda facilidades de importación y exportación al modo RPC de objetos java.

Dirección de transacciones: Centraliza variadas APIs de dirección y gestiona las transacciones para los objetos Java.

Convención sobre estructuración: Se incluyó el patrón Spring Roo para agilizar el inicio de proyectos en Spring en base a un Shell potente, simple sin dejar de lado la flexibilidad.

Mensajes: A través de JMS¹¹, matricula configurable de objetos receptores de mensajes.

Procesamiento por Lotes: El módulo incluye un framework de procesamiento de tareas, que tiene funcionalidades reutilizables como gestión de transacciones, inicio de tareas, etc.

Testing: Este módulo es el apoyo de componentes que facilitan el modelado de unidades de integración y prueba.

Autenticación y Autorización: Incluye herramientas netamente para la seguridad del aplicativo, se basan en estándares, protocolos, instrumentos de seguridad como autenticación por medio del subproyecto Spring Seguridad.

Administración remota: A través de JMX (Extensiones de administración Java) y la configuración de visibilidad.

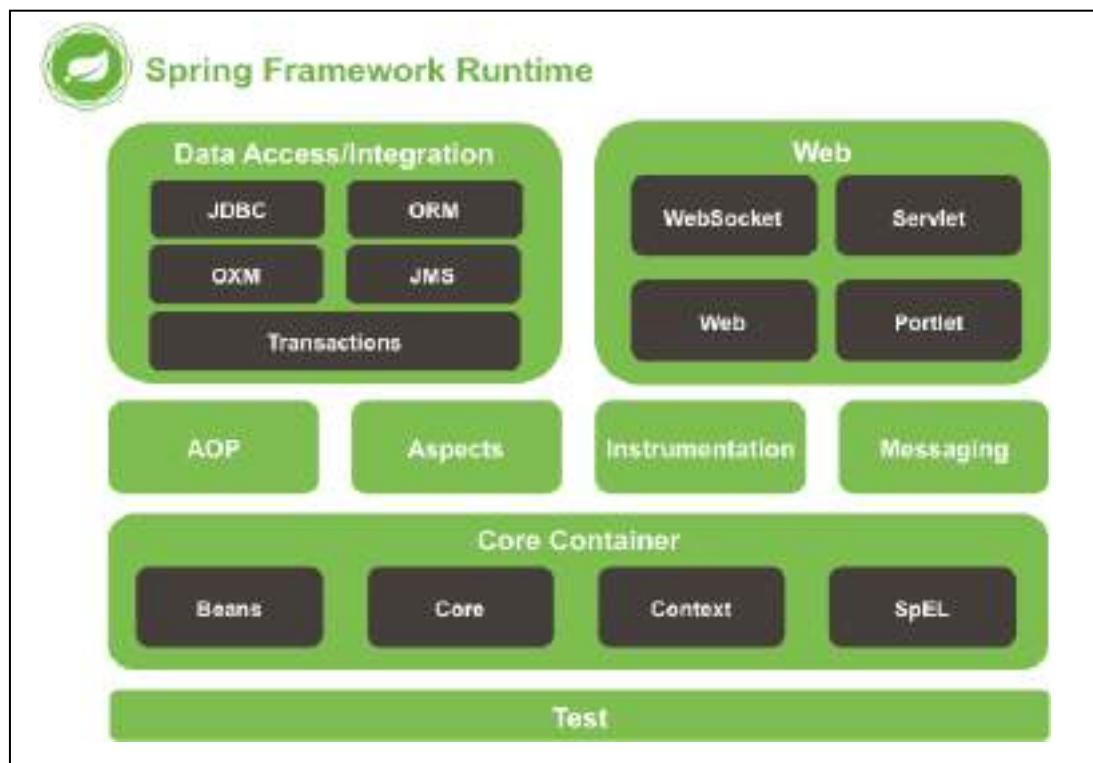
Figura 4

⁹ Mecanismo de Java para invocar un método de forma remota.

¹⁰ Protocolo de transferencia de información a través de archivos.

¹¹ Estándar de mensajería basado en Java.

Módulos Runtime Spring Framework



Spring Boot

Es un proyecto de Spring que permite desarrollar aplicaciones de manera simplificada y más rápida, mediante el uso del asistente de Spring Boot llamada Spring Initializer ya no se pierde tiempo agregando dependencias y con esta tecnología ya no requiere instalar un servidor en el entorno de desarrollo, sino que Spring Boot posee un servidor Tomcat embebido o integrado listo para crear el aplicativo.

Por otro lado, de las facilidades ya mencionadas que brinda Boot, esta tecnología posee características que potencian el desarrollo de las aplicaciones. Es así como Wells (2015), define cuatro características de Spring Boot:

Iniciadores de Spring boot: Aquí se añaden agrupaciones de dependencias comunes a una única dependencia que permite agregar a la generación de un proyecto que usa Maven o Gradle como gestores de repositorios.

La configuración automática: Esta funcionalidad recurre a la compatibilidad de Spring con la configuración condicional para establecer los Bean o clases Java que la aplicación necesita y configurarlos de manera autónoma.

Agente: la funcionalidad Actuador Spring boot agrega funcionalidades de gestión a una aplicación del Boot.

Finalmente, CLI o interfaz de línea de comandos se apoya del lenguaje de programación Groovy y de la mano de la configuración automática, se reduce todavía más el desarrollo de aplicaciones Spring.

Maven

Maven es una tecnología que sirve para gestionar y construir proyectos Java, la misma que se usara para la construcción del aplicativo batch. Maven utiliza un archivo XML llamado POM (Modelo de Objeto de proyecto) que describe el proyecto construido, además, muestra las dependencias de otros módulos y componentes externos.

Para Pérez (2015), Maven es algo más que una herramienta para automatizar la compilación del código se considera como herramienta gestora de proyectos. En un primer vistazo nos encontramos con un fichero XML y un comando que pone todo en marcha (Ant), esta herramienta partiendo de una interfaz única, tiene la capacidad de resolver las dependencias de los componentes del proyecto y reutiliza múltiples Plugin para solucionar cualquier tarea.

QUARTZ como Metodología de Solución

Cuando implementamos aplicaciones es algo recurrente que tengamos la necesidad de programar tareas en el aplicativo que puedan ejecutarse cada cierto tiempo o de mejor manera configurar los tiempos de ejecución; un ejemplo claro es cuando el cliente nos pida que de lunes a viernes se ejecute tal servicio Rest y los sábados a las 10:00 am se ejecutó otro servicio del aplicativo. Estas funcionalidades pueden no acoplarse en los aplicativos, debido a esto existen herramientas llamadas “scheduling” o en español “planificación” que pueden acoplarse a los aplicativos.

En cuanto al caso de las aplicaciones empresariales de java, el EJB provee funcionalidades scheduling, pero no es tan óptima y completa, también, el mismo Spring posee el API Java Tiempo que tiene funcionalidades básicas para el manejo de tiempo y no es una herramienta completa de calendarización.

Por consiguiente, el framework Quartz cumple con las funcionalidades óptimas para la programación o calendarización de tareas. Este marco Open Source es una biblioteca de programación de Jobs (trabajos) que se adapta a cualquier tipo de aplicación ya sea la más pequeña hasta la más grande. Quartz tiene la capacidad de ejecutar gran cantidad de Jobs que son los encargados de ejecutar lo que se quiera desarrollar.

En cuanto a lo que ofrece la API Quartz, se tiene el scheduler o programador, esta es primordial ya que se encarga de gestionar la programación o desprogramación (scheduling o unscheduling) de los Jobs y de la activación de los Triggers. Los Jobs son clases Java, el cual contienen la tarea que se desea ejecutar; estos Jobs pueden estar relacionado con uno o más Triggers, luego este Trigger al ser activado ejecutara los Jobs.

En cuanto a la creación de un Job, se debe crear una clase y luego se realiza la implementación de la interfaz “org.quartz.Job” que brinda un único método, como la muestra la figura 5:

Figura 5

Método único de la interfaz Job

```
public class YatacoJob implements Job{  
  
    public void execute(JobExecutionContext context) throws JobExecutionException{  
        .....  
    }  
}
```

Nota. Elaboración propia.

En primer lugar, el programador invocará el método de ejecución (execute) cuando se active o dispare el Trigger relacionado al Job creado. JobExecutionContext es un objeto que se pasa como argumento y tiene todos los datos de contexto para el Job, tales como JobDataMap. Este último es parecido a un mapa de Java, pero se diferencia por brindar métodos PUT (establecer) y obtener (GET); además, se establece en el archivo JobDetail antes de programar un Job y se puede recuperar más adelante cuando se esté ejecutando el Job mediante el método getJobDetail().getJobDataMap() de JobExecutionContext.

Por otro lado, JobDetail es un objeto creado por el mismo Quartz al momento de agregar el Job al Scheduler, además, este objeto posee distintas configuraciones de propiedades para el Job, como los ya mencionados JobDataMap, que se usan para almacenar información de estado para una instancia determinada de la clase Job.

En cuanto a los Triggers o disparadores, estos objetos se utilizan para realizar la ejecución de los Jobs, durante la programación de un Job se crea una instancia del disparador y se puede

configurar en sus propiedades la programación que se necesita. Además, los Triggers tienen la capacidad de tener un JobDataMap relacionado con la finalidad de pasar parámetros a un Job que son específicos de los disparos que haga el Trigger; también, Quartz trabaja con distintos tipos de disparadores, pero los más usados son SimpleTrigger y CronTrigger.

Por el lado de los SimpleTrigger, se utiliza cuando se requiere una única ejecución del Job en un momento dado matriculado en la configuración o también si se quiere ejecutar un Job en un momento dado y hacer que se ejecute la cantidad de veces que se desee. Un ejemplo sería la ejecución del Job en un tiempo específico, como el viernes quince a las diez de la mañana.

Por otro lado, los CronTrigger son utilizados cuando se requiere que la activación se base en horarios como son los calendarios, un ejemplo sería matricular la ejecución de un Job los viernes a las diez de la mañana de cada mes.

Por consiguiente, en la Tabla 1 se detallan las propiedades de los tipos de Triggers mencionados:

Tabla 1

Propiedades del Tipo de Trigger

Tipo	Propiedad	Descripción
SimpleTrigger	Posee hora de inicio.	El tiempo de disparo del Trigger.
	Posee hora de finalización.	Hora de finalización del Trigger, esta se puede configurar para eliminar el recuento de repeticiones.
	Puede repetir recuento.	Es la cantidad de veces que se disparará el Trigger, puede ser cero, un positivo de tipo Integer o SimpleTrigger.REPEAT_INDEFINITELY.
	Posee intervalo de repetición.	El intervalo de tiempo entre repeticiones del Job. Este intervalo puede ser cero si se requiere que el conteo de repeticiones sea en paralelo, sino puede ser un número entero positivo.
	Posee hora	El tiempo de

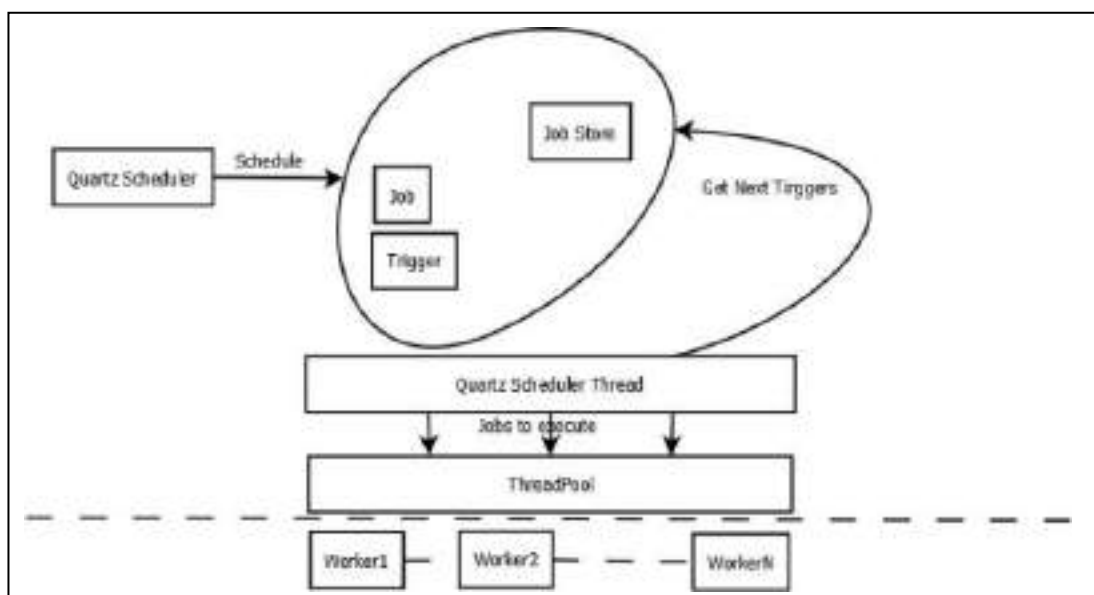
	de inicio.	disparo del Trigger.
CronTrigger	Posee hora de finalización opcional.	Hora de finalización del Trigger, esta es opcional.
	Conformada por 7 partes.	Cada parte representa un componente de tiempo: 1 representa segundos, 2 minutos, 3 horas, 4 días del mes, 5 mes, 6 día de la semana y 7 año, siendo esta última opcional.

Otro componente primordial, es el programador de fabrica (Scheduler Factory), este es el responsable de construir el modelo del Scheduler y conectar todos los componentes dependientes según la configuración del fichero de propiedades de Quartz.

En cuanto a componentes secundarios que están en la Figura 6, se incluye el JobStore, el cual es una interfaz que son implementadas por las clases que brindan un mecanismo de almacenamiento para el Job y el Trigger.

Por otro lado, otro componente se llama ThreadPool, antes de la ejecución del Job, esta pasa al grupo de subprocesos representados por ThreadPool.

En cuanto al Scheduler Thread, es el subproceso responsable de realizar la operación de disparar el Trigger.

Figura 6*Arquitectura Quartz Scheduler***3.4. Recursos requeridos***Equipos de Proyecto*

El área de apoyo para realizar el testing del aplicativo batch, es el área QA (Seguro de Calidad), esta realiza las pruebas de software que permiten verificar y revelar la calidad del software. Este equipo de proyecto se encargará de encontrar los errores en el código, teniendo conocimiento del flujo a pagos de proveedores y el producto del banco para facilitar las pruebas.

Por consiguiente, el analista tester se encargará de realizar un plan de pruebas, la creación de la data de prueba, la ejecución de los casos que se probaron y el reporte utilizando documentación

validadas por el banco; finalmente dará el visto bueno para continuar con el proceso de despliegue, esta vez a producción.

Por otro lado, otro equipo de apoyo para el proyecto es el SS-400, encargada de gestionar la data del banco, esta área posee un gran servidor y su propio entorno de trabajo. Este entorno o core bancario internamente utiliza un lenguaje de programación tradicional y posee su propio gestor de datos. Por lo cual, para este caso esta área brindará los ficheros de respuesta del estado de pago de cada lote.

En cuanto a otras funciones de esta área, SS-400 también trabaja con la plataforma B2B, generando los ficheros que validan la afiliación del cliente. La comunicación del servidor Business to Business hacia el SS400 se da por contenedores de mensaje llamada MQ Colas.

Software

Se utilizará un servidor SFTP existente para las pruebas de pre-producción en conjunto con el área de QA, este servidor tiene como Sistema operativo Linux con una capacidad de memoria suficiente para alojar gran cantidad de ficheros, este servidor alojara el aplicativo y las carpetas que alojaran los ficheros. Se solicitará al banco, credenciales de acceso a su red privada; estas credenciales únicas servirán para las pruebas del software, primero accediendo a la VPN y luego dejar los ficheros de prueba.

Por consiguiente, se va a configurar el muro de seguridad o Firewall y será brindado por el área respectiva del banco para la infraestructura planteada que dará alta seguridad a los servidores, evitando que algún atacante quiera obtener la información.

Por otro lado, se utilizará la base datos del entorno B2B, esta base de datos ya funciona para distintos proyectos elaborados para el banco y la comunicación será JDBC. Para el caso de las pruebas se solicitará al banco las credenciales para el acceso al repositorio de datos.

Por consiguiente, para realizar las pruebas del aplicativo se solicitará al banco el acceso a su Red Privada Virtual (VPN) mediante un usuario y contraseña. El login será a través de una herramienta que mediante un código se conectará a los servidores del banco. Además, ya conociéndose las credenciales y la ruta del servidor SFTP de pruebas, se accederá al servidor a través de una herramienta cliente SFTP gráfica para Windows y se alojaran los ficheros de prueba. Otro software requerido es el compilador de Java o JDK, sabemos que el lenguaje de desarrollo escogido es Java, pero es necesario contar con un compilador para que el código se ejecute, la versión del compilador a instalar será la versión 1.8.

En cuanto al desarrollo, se requiere instalar el entorno de desarrollo Eclipse versión Oxygen de 64 bits, este entorno posee la interfaz ideal para el desarrollo del aplicativo con los frameworks o marcos de trabajo a implementar, cabe mencionar que otra alternativa de entorno es STS (Conjunto de herramientas Spring) o Net Beans, pero no son tan ligeras como Eclipse.

Conocimientos

Antes de comenzar con el desarrollo de un proyecto informático, debemos tener muy claro el flujo de negocio, los procesos y productos del banco. Por consiguiente, el producto con el cual trabaja el cliente para este caso es el Confirming Cerrado. Así mismo el Confirming es el servicio de pago a proveedores que la mayoría de las entidades financieras brinda como parte de su catálogo de productos. Esta consiste en realizar pagos de facturas de manera anticipada o con fecha específica de una Empresa 1 hacia sus proveedores mediante el pacto que realiza la empresa 1 con

el banco, pero para este caso el banco puede cobrar una comisión o intereses según sea la política de la entidad financiera.

Este producto Confirming la adquiere la empresa 1 dirigiéndose al banco y obteniendo una línea de crédito Contra Confirming, desde ese momento ya se realiza un pacto con el banco y posteriormente la empresa 1 informara a sus proveedores sobre la afiliación para que las facturas de pagos sean aprobadas y el banco pueda realizar los pagos. En cuanto a los pagos, existen modalidades distintas de Confirming en las empresas financieras, pero las más conocidas son el Confirming abierto y Confirming cerrado.

IV. ANALISIS Y PRESENTACION DE RESULTADO

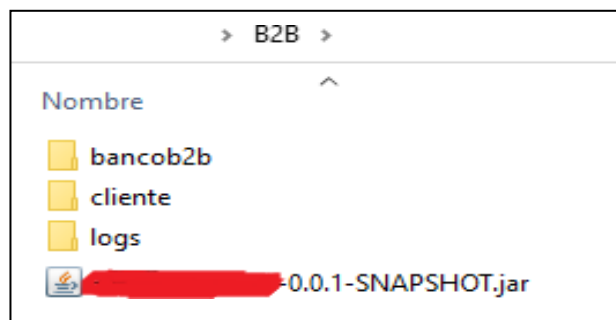
En cuanto al objetivo general planteado, la implementación del aplicativo logró integrar el sistema externo con el Business to Business de manera exitosa; así mismo, Dominguez (2017) tenía como objetivo de investigación desarrollar un aplicativo informático con el apoyo de recursos del ERP EasyWin y del lenguaje de programación Visual Basic; pudo concluir que la configuración del nuevo sistema utilizando la herramienta EasyPro del ERP mencionado realiza una integración rápida y sencilla, el cual evita errores de ejecución de la aplicación informática por su compatibilidad. Para el flujo de pruebas se utilizaron las credenciales VPN solicitadas para ingresar a la red del banco, luego se usó una herramienta cliente SFTP para conectarnos a los servidores de prueba ya autorizados, se utilizó como “muestra A” los ficheros con registros de lotes de pago de hasta un número de 9999, que según lo pactado era el máximo de registros en un lote que podía generar el sistema externo del cliente. Por consiguiente, se consideró a los ficheros de la plataforma bancaria como la “muestra B”.

Por otro lado, la figura 7 y 8 muestran que en el servidor SFTP de pruebas se creó la estructura de directorios para que los Jobs del aplicativo puedan identificarlos y se puedan ejecutar y se creó una carpeta para almacenar los registros de ejecución del aplicativo o mejor llamados logs, luego se instaló el archivo comprimido del aplicativo bachero (.jar), y posteriormente se configuró mediante comandos Linux a través de la consola Putty del cliente SFTP su ejecución de manera autónoma.

Un detalle que mencionar es que el servidor Tomcat embebido que nos brindaba el proyecto Spring Boot a través de una dependencia en Maven, fue removida y se generó un nuevo archivo .jar para las pruebas de preproducción; luego se desplegó un servidor Tomcat en el servidor de pruebas por medio de comandos Linux.

Figura 7

Carpetas principales y aplicativo batch – Servidor SFTP pruebas



Nota. Elaboración propia.

Figura 8

Estructura de carpetas cliente y bancob2b

Nombre	Fecha de modificación	Tipo	Tamaño
entrada	2/05/2021 21:33	Carpeta de archivos	
entradaTemp	2/05/2021 21:33	Carpeta de archivos	
salida	2/05/2021 21:34	Carpeta de archivos	
salidaTemp	2/05/2021 21:34	Carpeta de archivos	

Nota. Elaboración propia.

Por consiguiente, se elaboró un documento con los distintos escenarios de prueba que se podían presentar al ejecutar el jar del aplicativo, tales como los flujos no satisfactorios y satisfactorios; posteriormente los casos de prueba fueron testeados en las pruebas de integración con el área QA y que finalmente el usuario final o usuario del banco lo pudo aprobar.

En cuanto a otros recursos requeridos, se implementó el firewall que se le fue asignada al área de infraestructura de seguridad ubicada en una de las sedes internacionales del banco por políticas de la misma, el cual tenía como objetivo brindar la seguridad correspondiente a la información transmitida del aplicativo a la plataforma B2B, se instaló el entorno de desarrollo Eclipse y el JDK versión 8 para el desarrollo con el lenguaje de programación Java y se pudo realizar el acceso a la Base de datos Oracle que poseía el banco. Por consiguiente, se crearon las tablas para el registro de información del aplicativo, para los mantenimientos futuros y también la generación de backups.

Prueba de Plataforma B2B

Según la problemática planteada, la estructura de los ficheros usados por el sistema externo del cliente (Muestra A) no se adapta a la estructura de ficheros que maneja la plataforma B2B; es así como, los archivos no se han podido procesar o leer por la plataforma y como consecuencia

eran almacenadas en carpetas de historial o carpetas temporales. En la figura 9 se comparte la estructura del fichero del “cliente” que en resumen es la “muestra A” de la investigación:

Figura 9

Muestra A

```

C201090444776042020100109999
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000001000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000002000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000003000000
  
```

Nota. Elaboración propia.

A continuación, en la figura 10 se visualiza la estructura del fichero que la plataforma Business procesa y que será la muestra B de la investigación:

Figura 10

Muestra B

```

CARECERA [REDACTED] (03-04-2021) 00:00:00 | [REDACTED] | COMFIBAC(0) |
DETALLE 1 (5) 20140569745 (BB) 20140569745 (01) 51444.94 (03-04-2020) | 04 | [REDACTED] (J) | [REDACTED] (000001) |
DETALLE 2 (5) 20789854124 (BB) 20789854124 (01) 53333.94 (03-04-2020) | 04 | [REDACTED] (J) | [REDACTED] (000012) |
DETALLE 3 (5) 11478546314 (BB) 11478546314 (01) 5888.94 (03-04-2020) | 02 (0) (J) | [REDACTED] (000003) |
  
```

Nota. Elaboración propia.

Proceso de Pruebas con Aplicativo de Lotes

Inicialmente se realizaron distintos escenarios que fueron planteados en el documento de casos de prueba, para cada caso se contó con el apoyo del equipo QA o Testing, además se obtuvo la autorización por medio de credenciales únicas de poder utilizar el entorno del core bancario del ambiente de preproducción para validar si la información de los ficheros procesados por el Job número 3 se registraban; esta información se generó de manera satisfactoria con lo cual se pudo

realizar las pruebas de los ficheros generados por el Job número 4. Cabe mencionar que solo para preproducción estos últimos ficheros se obtuvieron de una carpeta que está ubicada en el servidor que autorizó el área de SS-400; luego estos ficheros fueron trasladados de forma manual a la carpeta del servidor SFTP de pruebas para posteriormente ser consumido por el Job 4, así mismo, este proceso se detallará más adelante en los casos de prueba.

Un detalle importante para las pruebas es que la estructura del fichero (Muestra A) esta dividido en dos partes, solo la primera fila corresponde a la cabecera que posee la información de la empresa principal del cliente y un indicador alfabético. A partir de la segunda fila en adelante es el detalle de pagos, conformada por la información de los proveedores con una longitud fijada y pactada por el cliente; esta estructura es generada por el sistema externo del cliente. Por otro lado, toda la información registrada en los lotes MA¹² y MB¹³ serán registrados en la base de datos Oracle para los futuros mantenimientos del aplicativo.

Casos de prueba

Para todos los casos de prueba se puso en funcionamiento el aplicativo que incluye a los Jobs y estaba en funcionamiento la plataforma Business to Business. Por consiguiente, se mostrarán 4 escenarios de los cuales, los dos primeros son escenarios de error de parte de la estructura del fichero dejado por el cliente MA y los últimos dos nos mostrará los ficheros con la estructura correcta que demuestran la integración del sistema externo con la plataforma Business to Business o B2B.

Escenario 1: Ficheros con estructura de cabecera incorrecta

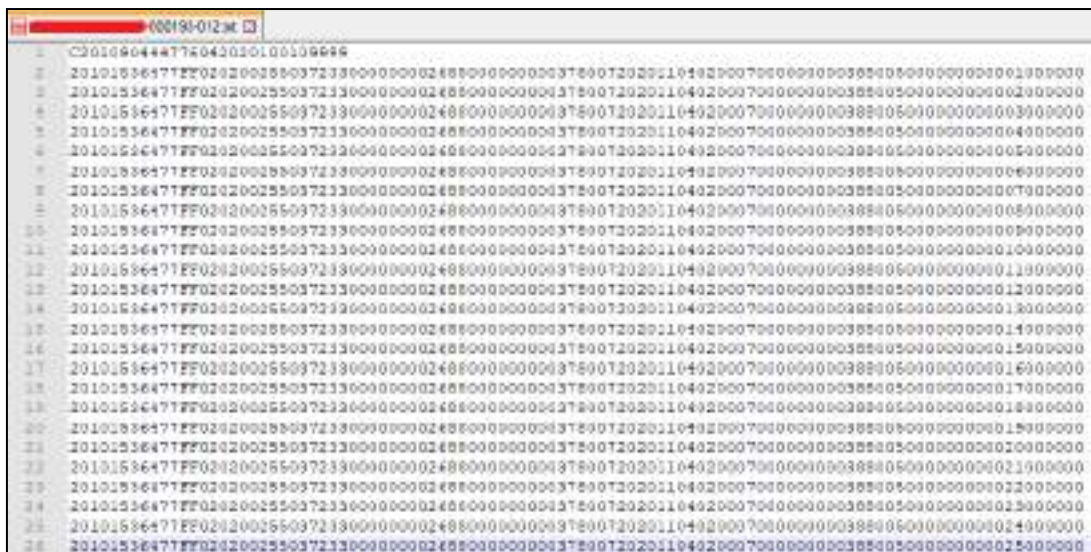
¹² Muestra A.

¹³ Muestra B.

La figura 11 muestra el fichero que se dejó en la carpeta cliente/entrada con la estructura de cabecera incorrecta.

Figura 11

Fichero con cabecera incorrecta

The image shows a screenshot of a text editor window titled '00191-012.txt'. The window contains a list of 24 lines, each representing a data record. Each record begins with a unique 20-character header string: 'C201090448775043010100109999'. This is followed by 19 columns of data. The data fields are separated by spaces and contain a mix of numbers and some null-like characters (e.g., '0000'). The records are numbered 1 through 24 on the left side of the window. The overall structure suggests a header row followed by 23 data rows, but the format of the header and the data fields appears inconsistent or 'incorrect' as per the caption.

Nota. Elaboración propia.

Resultado:

Este fichero fue consumido y traslado a la carpeta *cliente/entradaTemp* por el Job 1 del aplicativo, pero no pudo ser consumido por los otros 3 Jobs ya que contenía errores, como indicador del error se devuelve el fichero con un código y mensaje error en la carpeta *cliente/salida*. En la figura 12 se muestra el fichero de resultado que es entregado al cliente:

Figura 12

Fichero de respuesta para cabecera incorrecta

En referencia a la figura 14, el fichero fue consumido y traslado a la carpeta *cliente/entradaTemp* por el Job 1 del aplicativo, pero no pudo ser consumido por los otros 3 Jobs ya que contenía errores de longitud tanto para la información de la empresa y para sus proveedores, como indicador de los errores se devuelve el fichero con los códigos y mensajes de error correspondientes en la carpeta *cliente/salida*. A continuación, se muestra el fichero de resultado que obtiene el cliente:

Figura 14

Fichero de respuesta para longitud incorrecta

```
01001099999002003Longitud de cabecera diferente al Permitido
02550372330000000026880000000003780072020110402000700000000038800500000000001000000
02550372330000000026880000000003780072020110402000700000000038800500099999002004Longitud de detalle diferente
02550372330000000026880000000003780072020110402000700000000038800500000000003000000
02550372330000000026880000000003780072020110402000700000000038800500000000004000000
02550372330000000026880000000003780072020110402000700000000038800500000000005000000
02550372330000000026880000000003780072020110402000700000000038800500000000006000000
```

Nota. Elaboración propia.

Escenario 3: Fichero con estructura correcta para generar fichero R-1

En referencia a la figura 15, se aloja un fichero (MA) con formato correcto en la carpeta *cliente/entrada* para tener como resultado un fichero (Muestra B) que haya pasado por la plataforma B2B y haya sido ejecutada por 3 Jobs implementados en el aplicativo con el modelo Quartz Scheduler ya detallado anteriormente. El objetivo de esta prueba es que el fichero pueda pasar el flujo de integración completo y que genere el fichero R-1 que necesita el cliente para sus reportes.

Figura 15

Fichero con estructura correcta

```

C201090444776042020100109999
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000001000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000002000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000003000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000004000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000005000000
20101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000006000000
00101536477FF0202002550372330000000026880000000003780072020110402000700000000038800500000000007000000

```

Nota. Elaboración propia.

Resultado:

En primero lugar, el fichero fue consumido y traslado a la carpeta *cliente/entradaTemp* por el Job 1 del aplicativo, luego fue trasladado a la carpeta *bancob2b/entrada* donde el Job 2 se encargó de trasladar el archivo a la carpeta input del “servidor SFTP interno” que es consumido por la plataforma Business to Business, luego esta aplica una serie de validaciones tanto de seguridad como de estructura para que finalmente la plataforma entregue un archivo validado en su carpeta output; luego de este proceso el Job 2 se ejecuta nuevamente y se encarga de trasladar el archivo a la carpeta *bancob2b/salida*. Por consiguiente, el Job 3 empieza a actuar identificando la nomenclatura o el nombre del fichero dejado en la última ruta de carpeta por el Job 2, si posee el nombre correcto esta es procesada y enviada a la carpeta *cliente/salida*, sino no se procesa el fichero informándose con un mensaje en la consola y por ende en los logs. A continuación, en la figura 16 se muestra el formato del fichero R-1 que obtiene el cliente:

Figura 16

Fichero R-1

```

99000000000000Registrado correctamente
330000000026880000000003780072020110402000700000000038800500000000001000000999999100120Fecha de Pago debe ser mayor
33000000002688000000000378007202011040200070000000003880050000000000200000000000000000000Registrado correctamente
330000000026880000000003780072020110402000700000000038800500000000003000000999999100120Fecha de Pago debe ser mayor

```

Nota. Elaboración propia.

Escenario 4: Fichero con estructura correcta para generar fichero R-2

Para este escenario, se requiere haber realizado el escenario 3, en otras palabras, esta es una prueba dependiente porque se requiere de la ejecución del Job número 3, ya que de manera automática el fichero R-2 es generado por el core bancario luego de haberse entregado al cliente el fichero R-1; por consiguiente, el Job 4 entrará en ejecución para dejar el fichero en la carpeta del cliente. El objetivo de esta prueba es que se pueda comprobar la correcta integración del core al flujo de pagos, digo ficheros porque se generan ficheros del mismo lote cada vez que el estado de pago cambie hasta que el pago se haya efectuado, el cual se valida internamente por el código de estado dentro del fichero.

Figura 17: Fichero generado por el core bancario de manera automática con sus respectivos estados:

Figura 17

Fichero de respuesta B2B

```

0070|000000000000|00000000|000001|EN PROCESO
5600.04|03-04-0001|103|0112470000000000561244|0|||INTER-SORVEN S.A.S.|000001|000000|Abonado
5600.04|03-04-0001|103|0050000000007000000000|7|||GENERALIZA CORE S.A.S|000000|000000|Abonado

```

Nota. Elaboración propia.

Resultado:

Primeramente, solo para este caso este fichero es alojado en la carpeta output del “SFTP interno” de manera manual desde un servidor del SS-400, luego el Job 2 se ejecuta y

posteriormente deja el archivo en la ruta de carpeta *bancob2b/salida*, luego el Job 4 entra en acción que cumple la función de entregarle al cliente el fichero con su formato brindado inicialmente, pero adicionándole los estados de pago a cada detalle o registro del lote y finalmente este fichero es dejado por el Job 4 en la carpeta *cliente/salida* como se muestra en la figura 18.

Figura 18

Fichero de respuesta R-2

99000000000000000000EN PROCESO
33000000000268800000000037800720201104020007000000000388005000000000010000000000000000000Abonado
33000000000268800000000037800720201104020007000000000388005000000000020000000000000000000Abonado

Nota. Elaboración propia.

Bajo el resultado R-1 y R-2 de las pruebas que favorecen de manera exitosa al aplicativo batch, podemos mencionar al modelo Quartz Sheduler como la arquitectura que dio solución para la integración de las partes detalladas anteriormente. Por consiguiente, Flores (2019) en su estudio tenía como finalidad mejorar la calidad del artículo software en base a la lógica Ágil Iconix aplicadas en las fases de desarrollo de software. El autor concluye, que mediante la inserción de las técnicas de elaboración de programas que se aplican en la norma antes mencionada, mejora el carácter informático; por lo tanto, el autor favorece la implementación de Quartz como arquitectura de desarrollo software.

En primer lugar, se requería procesar múltiples archivos por parte del cliente; además, se tenía conocimiento de la cantidad de data que conllevaba el producto Confirming Cerrado y lo más importante era que el procesado de pagos requería ser automático y programado para que el cliente realice sus reportes respectivos en un determinado día.

Siendo así, se procedió a utilizar Quartz configurando los disparadores para que sean ejecutados en el tiempo pactado con el cliente, luego se utilizó el componente Job para establecer los procesos batch para el consumo de ficheros según se requieran, teniendo como base que este esquema soporta múltiples Jobs y por ende es escalable para cuando se necesite ejecutar múltiples tareas.

Así mismo, este modelo demuestra ser exitoso, porque cumple con lo pactado con el cliente, teniendo los componentes con las funcionalidades correctas para la integración, ya sea desde la calendarización de las tareas para el proceso de pagos que pidió el banco y la correcta orquestación de los Jobs implementados por medio del componente Scheduler Factory.

Por otro lado, el modelo Quartz nos da la opción de realizar el mantenimiento correspondiente de los procesos, ya que, si se requiere crear una nueva tarea, de manera sencilla se matricula en un archivo de propiedades el nombre del trigger correspondiente, luego se crea el cuerpo del servicio y repositorio que se requiera y se ejecuta la tarea a través del Job referenciado por el nombre del trigger en una anotación Java. Esta configuración nos permite solucionar futuros requerimientos del cliente y brinda la seguridad correspondiente a los ficheros que se consumen al realizar el apagado de alguna tarea.

Por otro lado, se implementaron los procesos batch con resultados favorables, como hemos visto en cada caso de prueba se han utilizado 4 Jobs para poder convertir, reconvertir y trasladar ficheros de carpeta en carpeta en el servidor SFTP de pruebas y en el servidor SFTP interno dentro de la red del banco. Por otro lado, Vilchez (2019) tenía como objetivo de estudio determinar el impacto de la unión de un sistema externo con el SAP en el contenido de los compradores, el cual concluyó que esta integración de sistemas con componentes diferentes impacta en la dimensión de

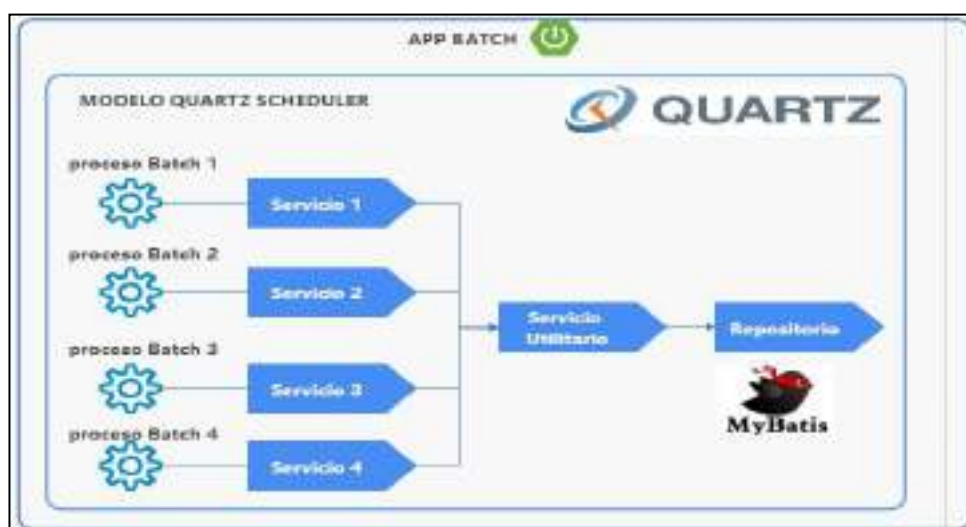
fiabilidad, comunicación y capacidad de respuesta del cliente. Este estudio favorece el impacto de fiabilidad y comunicación de los procesos batch del aplicativo.

Como en la figura 19, cada Job ha sido necesaria la construcción de servicios batch para cada proceso antes mencionado, estos procesos son los engranajes que hacen que nuestro aplicativo funcione en base al modelo Quartz.

Cada proceso incluye un servicio, que mediante un método realiza una serie de validaciones y usa un servicio reutilizable o clase utilitaria que por dentro se inyecta una interfaz que consume la base de datos Oracle.

Figura 19

Arquitectura de aplicativo



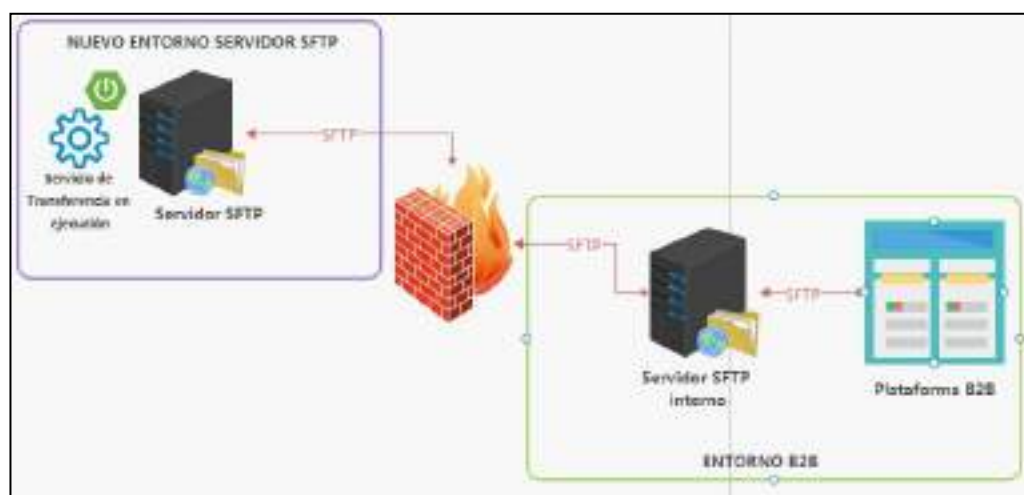
Nota. Elaboración propia.

En referencia a la figura 20, la implementación de un servicio de transferencia de archivos (servicio 2) fue vital en la integración, ya que sin este servicio no habríamos podido dejar o trasladar los ficheros que están alojados en las carpetas que el cliente deja y que la plataforma B2B

deja como respuesta. Este proceso se realiza con la comunicación vía SFTP entre el servidor SFTP de pruebas y el servidor SFTP interno del entorno Business to Business en la red del banco. Además, este proceso se da en ida y vuelta, trasladando los ficheros para seguir siendo ejecutados por los otros procesos batch e internamente este servicio identifica las carpetas correspondientes de entrada y salida que hemos establecido en el servidor SFTP de pruebas.

Figura 20

Comunicación SFTP usando Servicio de Transferencia



Nota. Elaboración propia.

Finalmente, la actuación del framework MyBatis que se planteó al inicio del proyecto tubo resultados positivos, este esquema utilizado para el consumo de base de datos se acomodó perfectamente al modelo Quartz utilizado en el aplicativo bachero.

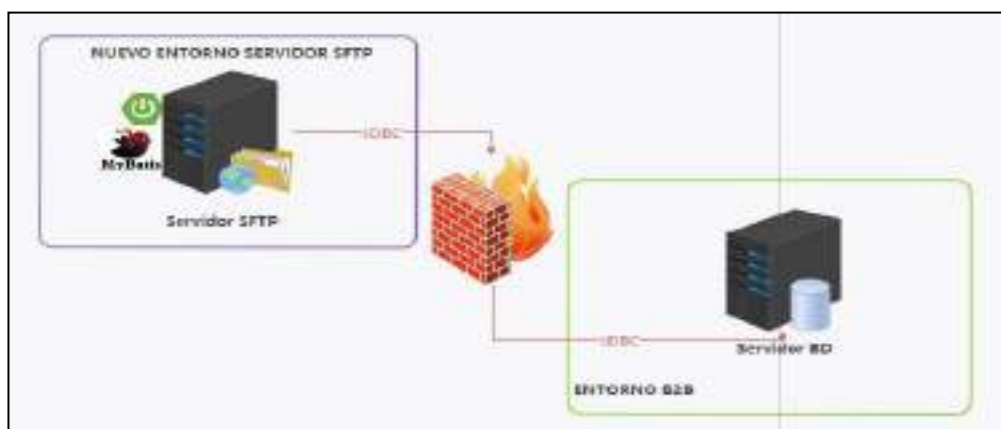
En primer lugar, utilizamos los archivos de tipo XML para poder configurar nuestras queries que nos permitió mapear las distintas clases Java que creamos en la estructura del proyecto con registros de la base de datos Oracle. En segundo lugar, creamos interfaces con sus respectivos métodos y con sus clases Java como parámetros para consumir nuestros archivos con estructura

XML, estas interfaces tienen el mismo nombre que nuestras tablas creadas en la base de datos y además se crearon otras interfaces que consumían tablas existentes para obtener datos que se necesitaban del cliente y sus proveedores. Esta data se utilizó para la integración de la data entre la estructura del fichero del cliente y la plataforma Business to business.

Por último, la data obtenida y registrada en la base de datos se dio primero desde los repositorios e interfaces alojadas dentro del aplicativo que luego vía JDBC pasa a través del muro de seguridad (Firewall) como lo muestra la infraestructura y se conecta al servidor de base de datos del entorno B2B, luego de llegar al servidor esta registra la información o para el caso de pedido de información, esta devuelve la información pasando nuevamente el Firewall y llegando a retornar la información al aplicativo.

Figura 21

Comunicación JDBC usando MyBatis



Nota. Elaboración propia.

Como en la figura 21, el marco de trabajo MyBatis permitió la integración de la plataforma y brindó facilidades en el proceso de data de manera rápida, simple y segura.

V. CONCLUSIONES

En el presente proyecto se determinó como la implementación del aplicativo Batch mejoró el Business to Business en una entidad bancaria, Lima -2021, y se concluyó que la plataforma del banco ahora es adaptable al sistema externo (Ver figura 16 y 18) y que, si se requiere alguna nueva validación de la estructura o campo que contiene el lote, la implementación sería rápidamente adecuada en el código validado por un nuevo método; además, que no impactaría en la arquitectura del marco de trabajo Quartz del aplicativo y finalmente para el caso de las pruebas, se tomaría como base los escenarios o casos de prueba ya elaborados.

En el presente proyecto se determinó como la implementación del modelo Quartz Scheduler integra el Business to Business en una entidad bancaria, Lima -2021, esto realizado con los componentes Jobs que a través de la ejecución de cada tarea implementada lograron sus objetivos ya sea desde transformar y retornar el fichero como lo requería el cliente. Además, el modelo Quartz logró calendarizar las tareas del proceso de pagos de manera automática, rápida y sencilla (Ver figura 19); así mismo, logró optimizar la plataforma permitiéndose ahora poder crear otros procesos si fueran necesarios tales como, un nuevo Job o un disparador o Trigger (sea Simple o Cron).

En el presente proyecto se determinó como la implementación de procesos Batch integra el Business to Business en una entidad bancaria, Lima -2021, pues se concluyó que cada proceso tiene la capacidad de adaptarse a la tarea de procesar un nuevo tipo de pago a proveedores (Ver figura 19), es así que se puede crear un nuevo servicio batch; luego, este internamente llama al método que procesará su lote de pago, posteriormente se procesará el lote y finalmente consultará el repositorio usando MyBatis para guardar la información en la base de datos.

En el presente proyecto se determinó como la implementación de un servicio de transferencia de ficheros integra el Business to Business en una entidad bancaria, Lima -2021, se concluyó que este servicio brinda soporte a nivel de seguridad de cada lote en el proceso de transferencia de archivos (Ver figura 20). Por un lado, cumple con transferir los archivos desde un servidor hacia otro de manera segura y, por otro lado, en caso se tuviera alguna amenaza de seguridad de los ficheros que están siendo transferidos, ahora se tiene la opción de poder bajar o apagar el servicio de manera que se cortaría el proceso de transferencia de archivos entre los servidores.

En el presente proyecto se determinó como la implementación del framework MyBatis integra el Business to Business en una entidad bancaria, Lima -2021, y se concluyó que las propiedades de los archivos XML que usa el framework para el intercambio de información, la hacen un esquema potente, estable y sencillo de usar por su facilidad de poder hacer consultas de gran cantidad de información a través de la etiqueta de listas en las queries o también en el caso que se requirió obtener un único dato, solo se agregó el tipo de dato que se quería obtener en la etiqueta XML.

VI. RECOMENDACIONES

Se recomienda que la arquitectura del aplicativo sea reutilizada para los futuros requerimientos y también para futuros sistemas externos que no puedan adaptarse a la plataforma Business to Business, esto por el ahorro de tiempo y costes para el área de soporte de los clientes; que a través de sus componentes ya comentados se hace automatizable y menos laborioso para dar soporte.

Se recomienda que para la configuración de las tareas a través de los Triggers estas deben ser de tipo Cron Trigger debido a la precisión de tiempos que pueden tener las tareas requeridas y el nombre de los Jobs debe ser simplificado y entendibles para los futuros desarrolladores cuando realicen un mantenimiento de este tipo de modelos de arquitectura como es Quartz.

Se recomienda que los procesos batch no abarquen muchas funcionalidades y que de preferencia sea un componente por funcionalidad; ya que de esta manera se puede realizar un mantenimiento más preciso y se hace menos dependiente a otros procesos. Con funcionalidades me refiero a cada acción que requiera los lotes de pagos, ya sea transformación o Re-transformación de estructura y tener en cuenta que cada función debe tener un código entendible sean o no muchas líneas.

Se recomienda que el servicio de transferencias de archivos pueda estar siempre en ejecución para que los ficheros puedan conectarse al entorno Business to Business a través del Servidor SFTP interno del banco.

Se recomienda utilizar queries sencillas para que los archivos XML no estén muy saturados de código, sencillas me refiero a que en las queries sea fácil de identificar las clases y campos que se están consumiendo y/o insertando, con ello los actuales y futuros mantenimientos de tablas en la base de datos sean rápidas y más precisas.

VII. REFERENCIAS BIBLIOGRÁFICAS

Andreu J. (2010). *Servicios en Red*.

Editex.<https://books.google.com.pe/books?id=vhit3ZmGQPsC&printsec=frontcover&dq>

=Servicios+en+Red&hl=es&sa=X&redir_esc=y#v=onepage&q=Servicios%20en%20Red&f=false

Aguilar P. (2010). *Seguridad Informática*. España, Madrid: Editorial Editex.

Boulanger T. (2015). *XML práctico: Bases esenciales, conceptos y casos prácticos*. Ediciones ENI.https://books.google.com.pe/books?id=x6QToh9wHMYC&printsec=frontcover&dq=XML+pr%C3%A1ctico:+Bases+esenciales,+conceptos+y+casos+pr%C3%A1cticos&hl=es&sa=X&redir_esc=y#v=onepage&q=XML%20pr%C3%A1ctico%3A%20Bases%20esenciales%2C%20conceptos%20y%20casos%20pr%C3%A1cticos&f=false

Casla P. (2012). *Administración de Servicios de mensajería electrónica*. Editorial RA-

MA.https://books.google.com.pe/books?id=0o6fDwAAQBAJ&printsec=frontcover&dq=Administraci%C3%B3n+de+Servicios+de+mensajer%C3%ADa+electr%C3%B3nica&hl=es&sa=X&redir_esc=y#v=onepage&q=Administraci%C3%B3n%20de%20Servicios%20de%20mensajer%C3%ADa%20electr%C3%B3nica&f=false

Costas J. (2011). *Seguridad y Alta disponibilidad (Grado Superior)*. Editorial RA-

MA.[https://books.google.com.pe/books?id=QIfDwAAQBAJ&printsec=frontcover&dq=Seguridad+y+Alta+disponibilidad+\(Grado+Superior\).&hl=es&sa=X&redir_esc=y#v=onepage&q=Seguridad%20y%20Alta%20disponibilidad%20\(Grado%20Superior\).&f=false](https://books.google.com.pe/books?id=QIfDwAAQBAJ&printsec=frontcover&dq=Seguridad+y+Alta+disponibilidad+(Grado+Superior).&hl=es&sa=X&redir_esc=y#v=onepage&q=Seguridad%20y%20Alta%20disponibilidad%20(Grado%20Superior).&f=false)

Calero C., Moraga A. & Piattini M. (2010). *Calidad del producto y proceso software*. Editorial

RA-

MA.<https://books.google.es/books?hl=es&lr=&id=M4h1WAvbgqQC&oi=fnd&pg=PA19&dq=Calidad+del+producto+y+proceso+software&ots=S7AdaRPawV&sig=-GzYnJ4nuOK8h3tj0WdAlYZczqc#v=onepage&q&f=false>

- Caicedo R. (2019). *Desarrollo de un aplicativo web para el control de las cotizaciones y las postventas de los proyectos inmobiliarios en la empresa Zoom Constructora S.A.* Universidad Cooperativa de Colombia, Bogotá.
- Corrales A. L., Beltrán M. & Guzmán A. (2006). *Diseño e Implementación de arquitecturas informáticas seguras: Una aproximación práctica.* Madrid, España: Editorial Dykinson.
- Ceballos J. (2010). *Java 2 Curso de Programación.* Editorial RA-
MA.https://books.google.com.pe/books?id=pq4-DwAAQBAJ&printsec=frontcover&dq=Java+2+Curso+de+Programaci%C3%B3n&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false
- Deléglise D. (2013). *MySQL 5 (versiones 5.1. a 5.6) Guía de referencia del desarrollador.* España, Barcelona: Editorial ENI.
- Dominguez W. (2017). *Implementación de un Sistema de Información para Controlar Solicitudes de Servicios Administrativos para una Entidad Financiera en la Ciudad de Lima - 2017.* Universidad Tecnológica del Perú, Lima.
- Flores E. (2019). *Metodología ágil Iconix en la calidad del producto software.* Universidad Nacional Federico Villarreal, Lima.
- Fortes J. (2016). *API de servicios web REST extensible dinámicamente basada en una arquitectura Plug-in para el procesamiento de imágenes recibidas desde dispositivos móviles.* Universidad de la Laguna, Tenerife.
- Ferrer J. (2012). *Implantación de Aplicaciones Web.* Editorial RA-
MA.https://books.google.com.pe/books?id=eofDwAAQBAJ&printsec=frontcover&dq=Implantaci%C3%B3n+de+Aplicaciones+Web&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false

- García I. & Munilla E. (2005). *E-Business Colaborativo*. Madrid, España: Editorial: Fundación CONFEMETAL.
- Gisbert B. (2015). *Administración y auditoría de los Servicios Web*. Madrid, España: Editorial Ra-Ma.
- Kendall K. E. & Kendall J. (2005). *Análisis y Diseño de Sistemas*. Juárez, México: Cámara Nacional de la Industria Editorial Mexicana.
- López M., Soltero F., Sánchez D., Moreno Á., Bolatti V. & Vara J. (2016). *Programación web en el Entorno Servidor*. Editorial RA-
 MA.https://books.google.com.pe/books?id=7I2fDwAAQBAJ&printsec=frontcover&dq=Programaci%C3%B3n+web+en+el+Entorno+Servidor&hl=es&sa=X&redir_esc=y#v=onepage&q=Programaci%C3%B3n%20web%20en%20el%20Entorno%20Servidor&f=false
 e
- Lederkremer M. (2020). *Redes Informáticas Avanzado*. Argentina, Buenos Aires: Editorial Six Ediciones.
- Molina J. & Baena L. M. (2007). *Implantación de aplicaciones Informáticas de gestión*. Madrid, España: Editorial Vision Net.
- Moreno J. C. (2012). *Sistemas Informáticos y redes locales*. Editorial RA-
 MA.https://books.google.com.pe/books?id=cI-fDwAAQBAJ&printsec=frontcover&dq=Sistemas+Inform%C3%A1ticos+y+redes+locales&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false
- Morales M. (2010). *Analítica Web para empresas: Arte, ingenio y anticipación*. Barcelona, España: Editorial UOC.
- Moreno J. C. (2011). *Programación (Grado superior)*. España, Madrid: Editorial Ra-Ma.

- Ñaupas, Mejía, Novoa & Villagómez (2014). *Metodología de la investigación cuantitativa-cualitativa y redacción de la tesis*. Bogotá, Colombia: Ediciones de la U.
- Piattini M. & Garzás J. (2010). *Fábricas de Software: Experiencias, tecnologías y Organización*. Editorial RA-
 MA. https://books.google.com.pe/books?id=WI6fDwAAQBAJ&printsec=frontcover&dq=F%C3%A1bricas+de+Software:+Experiencias,+tecnolog%C3%ADas+y+Organizaci%C3%B3n.&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false
- Pérez E. (2015). *Desarrollo de aplicaciones mediante el Framework de Spring*. España, Madrid: Editorial Ra-Ma.
- Roldán D., Valderas P. J. & Torres V. (2018). *Microservicios Un enfoque integrado*. Editorial RA-
 MA. https://books.google.com.pe/books?id=2o6fDwAAQBAJ&printsec=frontcover&dq=Microservicios+Un+enfoque+integrado&hl=es&sa=X&redir_esc=y#v=onepage&q=Microservicios%20Un%20enfoque%20integrado&f=false
- Raya J. L., Raya L. & Zurdo J. S. (2014). *Sistemas Informáticos*. Madrid, España: Editorial Ra-Ma.
- Sánchez A. & Montes J. (2013). *Programación de Servicios y Procesos*. Editorial RA-
 MA. https://books.google.com.pe/books?id=Uo2fDwAAQBAJ&printsec=frontcover&dq=Programaci%C3%B3n+de+Servicios+y+Procesos&hl=es&sa=X&redir_esc=y#v=onepage&q=Programaci%C3%B3n%20de%20Servicios%20y%20Procesos&f=false
- Sommerville Ian (2005). *Ingeniería del Software*. Madrid, España: Editorial Pearson Educación.
- Sánchez M. (2012). *Manual de Desarrollo Web basado en ejercicios y supuestos prácticos*. España, Andalucía: Editorial Ichtón.

Talledo J. V. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet*.

Madrid, España: Editorial Paraninfo.

Talledo J. V. (2017). *Instalación y configuración de aplicaciones Informáticas*. Madrid, España:

Editorial Paraninfo.

Torres M. (2013). *Desarrollo de aplicaciones con Java*. Perú, Lima: Editora Macro EIRL.

Vilchez J. (2019). *Integración de sistemas con ERP SAP y su impacto en la satisfacción de los clientes de Decor Center, 2017*. Universidad Privada del Norte, Lima.

Villegas G. (2018). *Desarrollo de un aplicativo web con diseño adaptable para la automatización del historial clínico, impresión de permisos médicos, recetas y órdenes de exámenes médicos para el control de los pacientes del área de consulta general de la clínica rivera ubicada en la ciudad de guayaquil*. Universidad de Guayaquil, Guayaquil.

Wanumen L. F., Rivas E. & Mosquera D. J. (2017). Colombia, Bogotá: Editorial ECOE.

Walls C. (2015). *Spring Cuarta edición*. España, Madrid: Ediciones ANAYA Multimedia.

ANEXOS

App Cliente SFTP - WINSCP

Evidencia de ingreso al Servidor de pruebas SFTP a través del cliente SFTP WinSCP. A continuación, se muestra el aplicativo WinSCP con el respectivo usuario y contraseña para el ingreso al servidor SFTP donde se alojarán los ficheros con la información de cada lote que será procesado:

