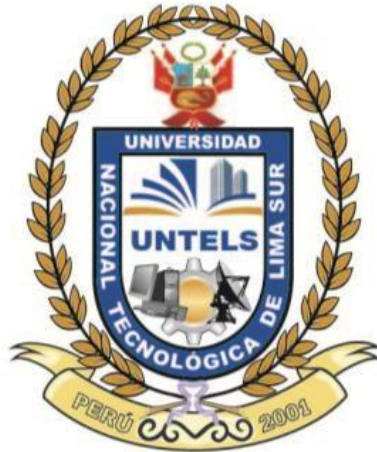


UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FACULTAD DE INGENIERÍA Y GESTIÓN

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“USO DE UN VALIDADOR DE NOMENCLATURAS PARA LA MEJORA
DE LAS BUENAS PRACTICAS DE DATASTAGE EN EL ÁREA DE
CENTRO DE INFORMACIÓN EN UNA ENTIDAD BANCARIA”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

CARI TUEROS, KATHERINE ALEXIS

Villa El Salvador

2017

DEDICATORIA

A Dios, por haberme ayudado a lo largo de toda mi vida personal, académica, profesional y laboral. Por ser luz en mi camino, por ser mi TODO.

A mi mamita, Teresa Amelia, mi motor y fuerza para seguir adelante, porque su sonrisa alegra mi vida, porque su simple existencia me llena de felicidad.

AGRADECIMIENTO

A Dios, porque cada logro mío es tuyo. Gracias Señor!

A mi mamá, mis hermanos y amigos que confiaron en mí. Gracias por todos los ánimos. Gracias a ti, cariño, por todo tu tiempo y apoyo.

A mi alma máter, Universidad Nacional Tecnológica de Lima Sur, por todos los conocimientos adquiridos.

A mi asesor Frank Escobedo, por su paciencia y por todos los conocimientos compartidos.

INDICE

INTRODUCCIÓN.....	1
CAPÍTULO I	3
PLANTEAMIENTO DEL PROBLEMA	3
1.1. Descripción de la Realidad Problemática.....	3
1.2. Justificación del Problema	8
1.3. Delimitación del Proyecto.....	9
1.3.1 Delimitación temporal.	9
1.3.2 Delimitación espacial.....	9
1.4 Formulación del Problema	9
1.4. Objetivos.....	9
1.4.1. Objetivo General.	9
1.4.2. Objetivos Específicos.....	9
CAPÍTULO II:	11
MARCO DE REFERENCIA TEÓRICO Y CONCEPTUAL.....	11
2.1 ANTECEDENTES DE LA INVESTIGACIÓN.....	11
2.1.1. Antecedentes Internacionales	11
2.1.2 Antecedentes Nacionales.....	12
2.2 BASES TEORICAS.....	16
2.2.1 Inteligencia de Negocios	16
2.2.2 Metodología SCRUM.....	17
2.2.3 Control de la Evolución del Proyecto.....	22
2.2.4 Info Sphere DataStage (DS).....	23
2.3 MARCO CONCEPTUAL	26

CAPÍTULO III	30
DISEÑO Y DESCRIPCIÓN DE LA HERRAMIENTA.....	30
3.1 ANÁLISIS DE LA HERRAMIENTA.....	30
3.2 DISEÑO O SIMULACIÓN DE LA HERRAMIENTA	56
3.2.1 Consideraciones para el uso del Validador	56
3.2.2 Carga de Archivos	57
3.2.3 Verificación de Errores – Observaciones	60
3.2.4 Guardar Reporte.....	69
3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS	74
3.3.1 Resultados de la Investigación	74
3.2.2 Resultados de los Objetivos	79
CONCLUSIONES	88
RECOMENDACIONES.....	89
BIBLIOGRAFÍA.....	90

LISTADO DE FIGURAS

Figura 1. Ciclo del desarrollo ágil	18
Figura 2. Ventana de Inicio de Sesión DataStage 8.7	24
Figura 3. Validador SES - Interfaz Principal.....	56
Figura 4. Archivos dsx y env	57
Figura 5. Seleccionando dsx	57
Figura 6. Seleccionar dsx y env	58
Figura 7. Ubicación de archivo dsx	58
Figura 8. Requiere ingresar env	59
Figura 9. Seleccionar archivo env	59
Figura 10. Procesar validación	60
Figura 11. Mensaje de Error.....	60
Figura 12. Reportes.....	61
Figura 13. Validador SES – Rutinas.....	62
Figura 14. Validador SES – Listado de Jobs.....	63
Figura 15. Validador SES – Listado Detalle	65
Figura 16. Validador SES – Error Documentación.....	65
Figura 17. Validador SES – Error Documentación Jobs	66
Figura 18. Validador SES – Argumentos	67
Figura 19. Validador SES – Argumentos Error	69
Figura 20. Validador SES – Resultados.....	70
Figura 21. Validador SES – Pruebas	70
Figura 22. Validador SES – Salida	72
Figura 23. Porcentaje Tiempo Estimado de Revisiones	76
Figura 24. Resultado Encuesta – Pregunta 1	80
Figura 25. Resultado Encuesta – Pregunta 2	81
Figura 26. Resultado Encuesta – Pregunta 2A.....	83
Figura 27. Resultado Encuesta – Pregunta 3	85
Figura 28. Resultado Encuesta – Pregunta 3A.....	87

LISTADO DE TABLAS

Tabla 1. Asignación de Roles	30
Tabla 2. Detalle de Software.....	32
Tabla 3. Tabla de prioridades	32
Tabla 4. CheckList Revisión de Pases	33
Tabla 5. Requerimientos del Sistema (Product BackLog).....	38
Tabla 6. Requerimientos del Sistema Primera Iteración	43
Tabla 7. Tareas para la primera iteración.....	46
Tabla 8. Datos Generales para el primer Sprint	49
Tabla 9. Requerimientos para la Segunda Iteración	51
Tabla 10. Tareas a realizarse para la Segunda Iteración.....	53
Tabla 11. Datos Generales de Segunda Iteración	55
Tabla 12. Comparación de Tiempos.....	75
Tabla 13. Tiempo y Porcentaje Revisiones	75
Tabla 14. Comparación de Costos	78
Tabla 15. Comparación de Costo Acumulado.....	78
Tabla 16. Encuesta de Percepción – Pregunta 1	79
Tabla 17. Encuesta de Percepción – Pregunta 2	81
Tabla 18. Encuesta de Percepción – Pregunta 2A.....	83
Tabla 19. Encuesta de Percepción – Pregunta 3	84
Tabla 20. Encuesta de Percepción – Pregunta 3A.....	86

INTRODUCCIÓN

El presente proyecto tiene por finalidad efectuar el Uso de un validador de Nomenclaturas para la mejora de las Buenas Prácticas de DataStage en el área de Centro de Información en Scotiabank. Este validador será desarrollado por una consultora externa, que brinda servicios a Scotiabank, llamada Software Enterprise Services (SES). Inicialmente SES desarrolló un prototipo de validador para mejorar su proceso de revisión de objetos de DataStage. Lo que se quiere es adaptar este validador para Scotiabank, teniendo en cuenta sus estándares y nomenclaturas en sus revisiones. Para la aplicación del validador, nos basaremos en la metodología de desarrollo ágil SCRUM.

En el capítulo I, se describe la Realidad Problemática presente en Scotiabank. Así mismo, se plantea la justificación del Problema, la delimitación del Proyecto, se realiza la formulación del Problema y se definen los objetivos del Proyecto.

En el capítulo II, se muestran antecedentes de la investigación, tanto nacionales como internacionales. Se definen las bases teóricas de la investigación y se realiza el marco conceptual. En este capítulo definimos el método por el que se desarrollará el validador.

Finalmente se presentan las conclusiones y recomendaciones a las que se llegaron después de realizar la presente investigación.

CAPÍTULO I:

PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la Realidad Problemática

Actualmente, las entidades bancarias están manejando sus procedimientos bancarios en base a Business Intelligence (BI), con la ayuda de la herramienta DataStage versión 8.7 y 11.3, así mismo la herramienta AS400. Claro ejemplo de ello, es Scotiabank. S.A.A.

El área de BI en Scotiabank se creó hace algunos años para ser de apoyo en los procesos que realiza Scotiabank. Los analistas pertenecientes al área de BI, trabajan creando proyectos en DataStage, así mismo generando objetos en cada Proyecto. Estos objetos son creados inicialmente en un ambiente de Desarrollo, luego pasan a un ambiente de prueba IST. Posteriormente se crean las solicitudes para impactar los objetos en el ambiente de Producción.

El área de Centro de Información se dedica a la atención de las solicitudes (Pases, Soporte e Incidencia). Los pases solicitan impactos a Producción. Puede ser de tipo "DataStage o DB2. En los pases DataStage, los analistas adjuntan un archivo dsx, donde se encuentran los objetos. En los pases DB2, se envían Scripts y volumetrías. Se crean tablas en producción.

Los Soportes solicitan ejecuciones de Jobs, creación de tablas en IST y accesos a los ambientes en desarrollo. Estas solicitudes no producen impacto en Producción.

Las Incidencias son solicitudes que necesitan ser atendidas en un mínimo tiempo, ya que van a tener un efecto sobre los procesos referidos a los negocios de Scotiabank. Todas las incidencias se impactan en el ambiente de producción. Una incidencia puede incluir archivos dsx, Script y ejecución de Jobs.

Todas las solicitudes tienen estado inicial de "Registrado", aquellas que requieren impacto o Ejecución en producción deben tener estado "Aprobado" para iniciar su revisión y posterior impacto. Los Pases y las Incidencias son las únicas solicitudes que contienen adjuntos los archivos dsx. Estos archivos contienen objetos DataStage que serán revisados antes de impactarlos, verificando el cumplimiento de los Estándares definidos por el Banco.

Para las revisiones de los objetos se utilizan documentos propios del Banco y del área de Centro de Información, básicamente tres documentos, el

primero contiene los estándares de Nomenclatura DataStage, considera las buenas prácticas y reglas que se deben cumplir en el Desarrollo de Jobs en DataStage.

También se tiene el documento de Arquitectura y Buenas prácticas en DataStage, cuyo objetivo es definir los mismos, con los que se trabajará los proyectos de Inteligencia de Negocios del Banco. Este documento contiene además, los estándares de construcción de Jobs y la configuración de DataStage.

El tercer documento conforma una lista de puntos que el analista debe considerar al registrar un Pase. Este documento viene a ser el CheckList, en el cual se verifica, según la observación encontrada, si el Pase es “Observado y Reprogramado” o “Rechazado”.

Cuando un pase es “Observado”; es decir, se encontraron errores de nomenclatura o buenas prácticas, se le envía un correo al analista indicando que el Pase contiene observaciones que debe corregir. El analista debe responder el correo, adjuntando los objetos con las correcciones. Estos deben ser revisados nuevamente. Si se verifica la corrección, el pase procede a ser impactado en el ambiente de producción en el proyecto que el analista haya indicado. El estado final de la solicitud finaliza como “Cerrado”.

En caso que se hubiere encontrado observaciones graves conforme al CheckList, el Pase se cierra como “Rechazado” y no continúa la atención, ni

impacto. Se le envía un correo al analista informando el motivo del rechazo de solicitud. El analista debe registrar un nuevo ticket con los objetos corregidos.

Las revisiones se realizan manualmente, visualizando detenidamente la nomenclatura de cada Stage en cada Job, así mismo revisando que los objetos (Jobs, parámetros, rutinas) cumplan con la documentación de Buenas prácticas. Cada revisor se toma un tiempo estimado de 10 Jobs por hora.

Cada día se registran de 18 a 25 solicitudes. Si la cantidad de objetos es mayor, el tiempo que se tome cada revisor será mayor. La carga es mayor cuando se registran Incidencias, las cuales son de Prioridad, estas solicitudes no pasan por Revisión de nomenclatura, por ser una solicitud de Prioridad, pero sí se revisan las buenas prácticas. Cuando una Incidencia es asignada, debe atenderse lo más pronto posible. El revisor debe postergar su revisión actual y dedicarse a revisar la Incidencia.

Es así como se origina una cola de solicitudes, tanto la revisión como ejecución de impacto. Todo esto genera un retraso en la atención y cierre de pases, soportes e incidencias. Las cuales, depende del tipo de solicitud, tiene un tiempo estimado de atención y cierre. Esta deficiencia trae molestia e incomodidad en los analistas que necesitan el impacto urgente para llevar a ejecución su proyecto.

Otro problema que sucede frecuentemente se debe a que los revisores pueden obviar o no percatarse de ciertas observaciones, con lo cual esa observación no reportada persiste en el impacto al ambiente de producción.

1.2. Justificación del Problema

Debido a que el área de BI genera proyectos que serán usados por clientes de Scotiabank, se ve la necesidad de atender las solicitudes en menor tiempo.

Para esto es necesario que el impacto de los objetos contenidos en las solicitudes, sea lo más minucioso posible, ya que una observación no reportada podría afectar gravemente el proyecto, así mismo traer consecuencias en el ambiente de Producción.

El área de Centro de Información se ha visto afectado por las quejas en analistas, cuyas solicitudes no fueran atendidas a tiempo, y considerando los problemas indicados anteriormente, se ha determinado el uso de la herramienta Validador de Nomenclaturas y Buenas Prácticas en DataStage.

Este validador ayudará considerablemente en el tiempo de Revisión, así mismo se puede determinar si la solicitud es Rechazada. Con esto permitirá una toma de decisiones mucho más eficiente y oportuna. Así mismo los analistas se sentirán satisfechos con el servicio que se les brinda.

Para cumplir con las necesidades mencionadas, se plantea el uso del Método Ágil de Desarrollo "SCRUM", ya que Scotiabank está iniciando la metodología SCRUM en la mayoría de sus procesos de Inteligencia de Negocios.

1.3. Delimitación del Proyecto

1.3.1 Delimitación temporal.

El periodo bajo estudio comprende desde Agosto 2016 hasta Marzo 2017.

1.3.2 Delimitación espacial

El problema en el cual se trabajará se encuentra en la provincia de Lima, departamento de Lima, Perú, en la empresa Scotiabank S.A.A.

1.4 Formulación del Problema

¿De qué manera el uso de un validador de nomenclatura permitirá la mejora de las buenas prácticas de DataStage en el área de Centro de Información en una entidad bancaria?

1.4.1 Problemas específicos

- ¿Cómo encontrar todos los errores de nomenclatura en los objetos a impactar?
- ¿Cómo adaptar los estándares cambiantes del Banco?
- ¿Cómo atender las solicitudes en menor tiempo?

1.4. Objetivos

1.4.1. Objetivo General.

Efectuar el uso de un validador de nomenclatura para la mejora de las buenas prácticas de DataStage en el área de Centro de Información en una entidad bancaria.

1.4.2. Objetivos Específicos.

- Identificar las razones por la que un objeto es observado.

- Aplicar una metodología de desarrollo de software que permita adaptarse a los cambios de estándares del Banco.
- Determinar la influencia del Validador en la mejora de las revisiones.

CAPÍTULO II:

MARCO DE REFERENCIA TEÓRICO Y CONCEPTUAL

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Antecedentes Internacionales

En Venezuela, año 2011, Hung Campos, Marco Luis presentó la tesis de maestría de título Diseño de una metodología de desarrollo de software basada en la metodología ágil SCRUM y las prácticas de la gerencia de proyectos, en la Universidad Católica Andrés Bello. La investigación abarca la aplicación de la metodología SCRUM y las mejores prácticas de PMI (Project Management Institute) como Metodología de Desarrollo para la Gerencia de Desarrollo de Farmacias FARMAHORRO. El análisis de la investigación muestra cómo es posible determinar requerimientos, fortalezas y debilidades de la Gerencia y con esta información definir la metodología más adecuada para el desarrollo de software. De la investigación cómo es que de un proyecto piloto se puede aplicar la integración de la metodología ágil SCRUM y las mejores prácticas del PMI(Project Management Institute).

Se ha considerado también una tesis realizada en Ecuador, por Guanotasig Cambisaca, José Ricardo y Siza Ronquillo, Edison Xavier, el año 2012, cuyo título Diseño e implementación de un sistema web utilizando java y empleando la metodología SCRUM para la gestión administrativa de la junta parroquial de Guaytacama perteneciente al cantón Latacunga de la Provincia de Cotopaxi, se presentó como tesis de pre-grado en la Universidad Técnica de Cotopaxi. Esta investigación aborda el desarrollo de una plataforma web que automatice los procesos administrativos de una empresa. Toda esta automatización y desarrollo se realizará mediante la metodología SCRUM.

Se tiene que, en Sangolquí (Ecuador), enero de 2015 se presentó en la Universidad de las Fuerzas Armadas, la tesis de grado Maestría en Gerencia de Sistemas, de título Desarrollo de un caso de estudio aplicando el Modelo de Gestión de Calidad de datos basado en la Metodología IBM Data Quality para el portafolio de Business Intelligence de la Empresa Dwconsultware. Los autores Andrade Tirira, Christian Andrés y Madrid Ruiz, David Alejandro.

2.1.2 Antecedentes Nacionales

Se conoce que este validador inicialmente se creó y aplicó en una consultora que brinda servicios a Scotiabank. La consultora tiene por nombre Software Enterprise Services, a la cual, abreviadamente llamaremos SES. Esta consultora brinda recursos de apoyo (desarrolladores de DataStage) a los analistas de Scotiabank. Es por

eso que se ve afectado si una solicitud es Rechazada u Observada, ya que son los desarrolladores quienes están encargados de la creación o modificación de objetos que se envían en los Pases o Incidencias.

Es así como SES decide crear un prototipo del Validador, para la verificación de la correcta nomenclatura de los objetos de DataStage. Este validador es lo más cercano que se tiene como antecedente para este proyecto. La herramienta será adaptada y se regirá bajo los estándares propios de Scotiabank.

Así mismo, se realizó una investigación para determinar la existencia de casos similares al proyecto, y aunque no se encontró algún proyecto que aplique un validador de nomenclatura de buenas prácticas DataStage, se tomó en cuenta los proyectos detallados a continuación.

En primer lugar se tiene que, en julio del 2013 fue presentado en la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, el trabajo de grado Desarrollo de una infraestructura de Software para realizar pruebas automatizadas de Sistemas de Información desarrollados en lenguaje Cobol en el Contexto Bancario por Tasato Cánepa, Kenjy, como requisito para optar el título de Ingeniero Informático. Con este proyecto se logró desarrollar una infraestructura de pruebas de software usando como metodología NTP-ISO/IEC. También se logró probar el funcionamiento correcto de la infraestructura de pruebas en un ambiente simulado. Si bien este

proyecto no aplica la metodología SCRUM en el desarrollo de software, se tomó en consideración la investigación porque se está creando una herramienta de validación que servirá de apoyo a una entidad bancaria.

En el año 2011, en Chiclayo, Perú fue presentada la tesis de pre grado que lleva por título Sistema de Información ejecutivo para la toma de decisiones en el proceso de evaluación a través de la construcción de escenarios virtuales en tópicos de álgebra en el 3er grado de educación secundaria, en la Universidad Católica Santo Toribio de Mogrovejo, por Moreno Descalzi, Julio César. Se tienen dos procesos, uno de enseñanza y otro de evaluación, que serán automatizados en esta investigación, para esto se implementó un sistema transaccional para el proceso de enseñanza y otro a nivel ejecutivo para el proceso de Evaluación. Para la implementación de estos sistemas se usa la metodología SCRUM; de esta manera se muestra la evolución de una planificación inicial hasta el logro de resultados finales del proceso.

Así mismo en febrero del 2012 se presentó en la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú la tesis de título Intérprete para un Lenguaje de Programación orientado a objetos, con mecanismos de optimización y modificación dinámica de código, como requisito para optar el Título de Ingeniero Informático, tesis desarrollada por los bachilleres Gómez Díaz, Renzo Gonzalo y Salamanca Guillén, Juan Jesús. Se realizó la implementación de un intérprete con tres características, como son, el lenguaje orientado a

objetos, mecanismos de modificación dinámica de código y opciones de optimización.

Se tiene la presentación de la tesis en Huancayo, Perú, año 2015, de título Implementación de un Sistema de Información para la gestión académica del colegio particular Zárate mediante la Metodología SCRUM dentro de la Facultad de Ingeniería de Sistemas de la Universidad Nacional del Centro del Perú. Esta tesis fue desarrollada por García Huacachi, Henry Paúl, para optar el título profesional de Ingeniero de Sistemas.

2.2 BASES TEORICAS

2.2.1 Inteligencia de Negocios

La información es uno de los activos más valiosos para cualquier entidad bancaria. Su valor depende de cómo es gestionada, y en qué medida es utilizada eficientemente. Las organizaciones saben muy bien que la información es vital para la toma de decisiones.

La Inteligencia de Negocios (Business Intelligence) es el conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización. (Caralt J., 2010).

La Inteligencia de Negocios es un conjunto de herramientas y aplicaciones para la ayuda a la toma de decisiones que posibilitan acceso interactivo, análisis y manipulación de información corporativa de misión crítica. Estas aplicaciones contribuyen a un conocimiento valioso sobre la información operativa identificando problemas y oportunidades de negocio. Con estas, los usuarios son capaces de tener acceso a grandes volúmenes de información para establecer y analizar relaciones y comprender tendencias que posteriormente soportarán decisiones de negocios. (Mendez, 2006).

La inteligencia de negocios comprende las mejores prácticas, aplicaciones informáticas, información y herramientas para que la empresa pueda entender su negocio, de esta forma mejorando la toma de decisiones. (Carrera A., 2015).

La inteligencia de negocios es necesaria y puede ser aplicada a cualquier tipo de organización. Hoy en día las entidades bancarias recurren a su utilización para mejorar las operaciones con respecto a sus clientes, ya que es necesaria toda información de cada operación que realice el cliente.

2.2.2 Metodología SCRUM

SCRUM es una metodología de desarrollo ágil para gestionar el desarrollo de software, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua de las circunstancias de la evolución del proyecto. (Palacio J., 2006)

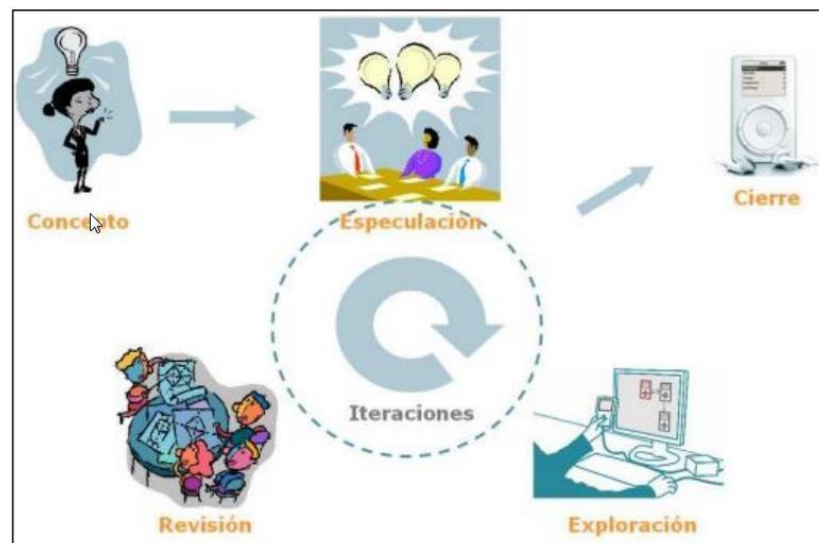
Según Palacio J. (2014), ahora se construye el producto mientras se modifican y aparecen nuevos requisitos. El cliente parte de una visión medianamente clara, pero el nivel de innovación que requiere, y la velocidad a la que se mueve su sector de negocio, no le permiten predecir con detalle cómo será el resultado final.

SCRUM permite además seguir de forma clara el avance de las tareas a realizar, de forma que los "jefes" puedan ver día a día cómo progresa el trabajo (CHR, 2007).

Esta metodología garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo (SCRUM, 2016).

El objetivo de SCRUM es maximizar el retorno de la inversión para la organización, así mismo permite realinear el software con los objetivos del negocio.

Figura1: Ciclo de desarrollo ágil



Fuente: Palacio, 2006

El marco de trabajo SCRUM consiste en los roles, eventos, artefactos y reglas, a su vez emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. (SCRUM Organization, 2013).

2.2.2.1 Reuniones

En la Planificación de Sprint se tomará un tiempo dentro de la jornada laboral previo inicio de cada Sprint en la que se determina cuál va a ser el trabajo y los objetivos en esa iteración.

Las Reuniones son diarias en el cual se realiza una breve revisión del equipo de trabajo realizado hasta la fecha y la previsión del siguiente día. En la Revisión de Sprint se realiza un análisis y verificación de Incremento de grado.

2.2.2.2 Elementos

Pila del Producto (Product Backlog), es la lista de requerimientos y funcionalidades requerida por el proyecto, cuyo origen es la visión inicial del producto. Esta lista se incrementa a lo largo del desarrollo. Es uno de los elementos más importantes, por ser una lista dinámica (Schwaber & Sutherland, 2011).

Pila del Sprint (Sprint BackLog), es la lista de las tareas a realizar y quién las desempeña. Se asigna las horas de trabajo y costo que va a suponer realizar cada tarea.

2.2.2.1 Roles

(SCRUM, 2016) clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: Propietario del producto (Product Owner), líder de SCRUM (Scrum Master), equipo

de desarrollo (Scrum Team). Todas las responsabilidades del proyecto se reparten en tres roles detallados a continuación.

Product Owner, es el propietario del producto y quién determinará los requerimientos. Representa la voz del cliente y el resto de interesados que no están implicados directamente en el Proyecto. Conoce a fondo las necesidades y proporciona la información necesaria en el momento preciso. Tiene la responsabilidad de obtener el resultado de mayor valor posible para el cliente. Así mismo, es el responsable de la financiación necesaria para el proyecto.

Scrum Master, es el líder del proyecto. Es quien asegura que el proyecto se lleve a cabo de acuerdo con las prácticas y reglas de SCRUM. Según la Guía SBOK (SCRUM, 2016), el Scrum Master es una facilitador que asegura que el Equipo Scrum esté dotado de un ambiente propicio para completar con éxito el desarrollo del producto. Cabe resaltar que Scrum Master llega a ser, más que un rol, una responsabilidad frente al funcionamiento de modelo; por ello es recomendable asignar este puesto a personas más adecuadas según la organización.

Equipo de Desarrollo, aquel que está encargado de desarrollar y entregar el producto. Su trabajo es imprescindible por eso deben ser capaces de saber todo acerca del Proyecto. También

sugiere cambios durante los procesos de Crear Entregables.

(SCRUM, 2016).

2.2.2.1 Valores

Según Palacio J. (2014), se tienen los siguientes valores en la metodología Scrum:

Delegación de atribuciones (empowerment) al equipo, para que éste pueda auto organizarse y tomar las decisiones sobre el desarrollo.

Respeto entre los miembros del equipo, debe haber confianza entre ellos, respetando la opinión, conocimiento y capacidad de cada uno.

Responsabilidad y autodisciplina. Que sean capaces de ser disciplinados sin que se les pida.

Trabajo centrado en el valor para el cliente y desarrollo de lo comprometido.

2.2.3 Control de la Evolución del Proyecto

El marco de trabajo SCRUM consiste en los roles, eventos, artefactos y reglas, a su vez emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. (SCRUM Organization, 2013). SCRUM maneja las siguientes tácticas:

2.2.3.1 Revisión de las Iteraciones

Cada iteración dura aproximadamente 30 días; y al finalizarse, se realiza una revisión con todas las personas implicadas en el proyecto.

Según Palacio J. (2014), cada ciclo de desarrollo o iteración (sprint) finaliza con la entrega de una parte operativa del producto (incremento). La duración de cada sprint puede ser desde una, hasta seis semanas, aunque se recomienda que no excedan de un mes. Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Es por tanto la duración del sprint, el período de tiempo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto.

2.2.3.2 Desarrollo Incremental

El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede evaluar.

Scrum resulta adecuado en proyectos con requisitos inciertos y, o inestables. (Palacio J, 2014).

2.2.3.3 Auto-organización

En SCRUM los equipos son auto organizados, con margen de decisión suficiente para tomar las decisiones que consideran oportunas. (Palacio J, 2014).

2.2.3.4 Colaboración

Es necesario que cada miembro del equipo colabore de forma abierta con los demás, según sus capacidades y no según su rol o su puesto. (Palacio J, 2014). Esto permitirá la auto organización en el equipo de desarrollo.

2.2.4 Info Sphere DataStage (DS)

DataStage es una herramienta ETL que permite crear y mantener fácil y rápidamente almacenes de datos procedentes de sistemas de Aplicaciones Empresariales incluyendo SAP, Siebel, Oracle y PeopleSoft CRM y de otros sistemas relacionados al negocio.

Datastage soporta la extracción, integración y transformación de altos volúmenes de datos desde estructuras simples hasta muy complejas. Entre funciones de DataStage, la más destacada es el diseño de tareas que extraen, integran, agregan, cargan y transforman los datos para el Data Warehouse o Data Mart. Las tareas son compiladas para crear ejecutables que son calendarizados por el Director y ejecutadas por el Servidor.

Figura 2: Ventana de Inicio de Sesión de DataStage 8.7

The image shows a Windows-style dialog box titled "Conectar con proyecto" with a close button (X) in the top right corner. The background is a blue and white wave pattern. In the top right, the IBM logo is displayed. The main text reads "IBM® InfoSphere™ DataStage® and QualityStage" followed by "Versión 8.7". Below this, there are four input fields: "Nombre de host de la capa de servicios" (a dropdown menu), "Nombre de usuario" (a text box), "Contraseña" (a text box with asterisks), and "Proyecto" (a dropdown menu). At the bottom, there are two buttons: "Iniciar sesión" and "Cancelar".

Fuente: Elaboración Propia

2.2.4.1 Componentes de DataStage

Administrator, es la Interfaz de usuario usada para configurar proyectos de Datastage y usuarios. Los trabajos están organizados en proyectos, junto a elementos de diseño asociados. Distintos usuarios pueden disponer de acceso a distintos proyectos. (InfoSphere DataStage, 2011)

Designer, es el Diseñador DataStage, usada para crear, diseñar y compilar tareas Datastage, también permite testear y ejecutar. El Cliente del Diseñador viene a ser un entorno de trabajo o un lienzo en blanco con una paleta que contiene las herramientas que conforman los bloques básicos para la creación: etapas, enlaces y anotaciones (InfoSphere DataStage, 2011).

Director DataStage para validar, calendarizar, testear, ejecutar y monitorizar jobs Datastage. A través del Director podemos apreciar el historial de las ejecuciones, Así mismo la duración de cada Job.

Manager, es una interfaz de usuario usada para visualizar y editar el contenido del repositorio.

2.2.4.2 Beneficios Info Sphere DataStage

Lo beneficioso de DS es que posee una plataforma ETL potente que admite la recopilación, integración y transformación de grandes volúmenes de datos.

DS permite la integración de datos prácticamente en tiempo real, facilitando así la conectividad entre aplicaciones y orígenes de datos.

Otro beneficio importante consta de la facilidad de su uso, poseedor de una amplia velocidad, flexibilidad y efectividad para crear, desplegar, actualizar y gestionar la infraestructura de datos.

2.3 MARCO CONCEPTUAL

Codificación: Expresión escrita en un lenguaje de ordenador.

Diseño de Job: Metadatos que definen los orígenes y destinos que se utilizan dentro de un trabajo y la lógica que se utiliza para operar con los datos asociados. Diseño de trabajo formado por etapas y los enlaces entre esas etapas. Los desarrolladores utilizan herramientas visuales para crear diseños de trabajo. El diseño de trabajo se almacena en el repositorio de metadatos, separado del ejecutable del trabajo. (InfoSphere DataStage, 2011)

Dsx: Es un archivo comprimido, el cual contiene objetos propios de DataStage. Este archivo se obtiene al exportar objetos desde el Diseñador DataStage.

Ejecutable de Job: El conjunto de objetos binarios, scripts generados y archivos asociados que se utilizan al ejecutar un trabajo. El trabajo ejecutable se almacena en el directorio de proyectos en la capa de motor. (InfoSphere DataStage, 2011)

ETL: Proceso mediante el cual se extrae la data de la base de datos transaccional, se transforma mediante operaciones, y se carga los datos ya transformados en el Data Warehouse.

Job: Es el objeto principal de DataStage, Existen tres tipos de Job (Paralelo, Secuencial y Server). Cada Job contiene un conjunto de stages que interactúan entre sí para mostrar un resultado definido conforme sea desarrollado.

Información: Definimos la información como “un conjunto de datos cerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen un significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo”. Su propósito puede ser el de reducir la incertidumbre o aumentar el conocimiento acerca de algo. Es decir, reducir la falta de conocimiento seguro y claro de algo; lo cual, genera mayor seguridad en la persona y permite tomar decisiones con mayor probabilidad de acierto. (Thompson, 2008)

Parameter Set: Objeto DataStage, que se encuentra en el Diseñador DataStage. Los Parameter sets se utilizan para almacenar parámetros que servirán para conexiones a fuentes de datos.

Rol: Función que una persona desempeña en un lugar o en una situación.

Rutina: Objeto dsx, que contiene un código de programación. Las rutinas son utilizadas en los Jobs Secuenciales.

Script: Contiene un código de sentencias SQL el cual puede contener creación de tablas, alteración de tablas, inserción, eliminación y modificación de data.

Sprint: Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (incremento) (Palacio J, 2014). El Sprint es el periodo en el que se realizan todas las acciones pactadas en el Sprint Backlog. Las entregas se realizan parcialmente para testear el producto hasta el final.

Stage: Los stages son elementos que serán usados en los Jobs. Sólo podrán ser utilizados en el diseñador. Todo el flujo de datos de un proceso en un Job, pasa a través de cada stage.

Volumetría: Es un documento que contiene datos de tablas a crear, indicando los campos que conformarán cada tabla, su llave primaria, entre otros..

CAPÍTULO III:

DISEÑO Y DESCRIPCIÓN DE LA HERRAMIENTA

3.1 ANÁLISIS DE LA HERRAMIENTA

El contenido de este capítulo abarcará la presentación del desarrollo del Validador, bajo la metodología seleccionada que es SCRUM, detallando cada actividad que se realizará y los cambios que surgieron a lo largo del proyecto. Como se mencionó, el Validador será desarrollado por SES, una consultora que brinda servicios a Scotiabank, por lo que, integrantes de esa consultora, formarán parte del equipo.

3.1.1. Recursos Humanos:

Se lista la asignación de roles conforme a la metodología SCRUM.

Tabla 1: Asignación de Roles

Rol	Responsable
Scrum Owner	Jefe de Centro de Información
Scrum Master	Líder de Proyecto - SES
Equipo de Desarrollo	Programadores

Fuente: Elaboración Propia

Producto Owner: Para este caso, el Jefe del área de Centro de Información de Sctiabank, representa este rol. Conoce todos los errores por la que un objeto DataStage es observado. Brinda toda la información necesaria para el desarrollo del proyecto.

Scrum Master: Este rol será asumido por el Líder del Proyectos de SES. Ya que tiene completa cercanía con el equipo de desarrollo y constante comunicación con el Jefe del área de Centro de Información.

Equipo de Desarrollo: Conformado por dos programadores, a cargo del líder de Proyecto, quienes mantendrán reuniones continuas con el Scrum Master, con el fin de asegurar que cada progreso del proyecto cumpla con los requisitos exigidos por el jefe de del área de Centro de Información.

3.1.2 Software

Se listan a continuación las herramientas que se utilizarán, tanto en el desarrollo, como en la documentación, a lo largo del proyecto:

Tabla 2: Detalle de Software

Tipo	Nombre
Desarrollo	Microsoft Visual Studio 2008
Edición Documentaria	Microsoft Office 2013
Sistema Operativo	Microsoft Windows 7

Fuente: Elaboración Propia

3.1.3 Requerimientos (Product BackLog)

Se realizó el siguiente cuadro de prioridades, dándole un nivel de complejidad para cada uno de ellas.

Cada requerimiento tendrá una prioridad baja, media, alta o muy alta, depende de lo indicado por el Product Owner, se definirá cuáles son los más importantes y necesarios

Tabla 3: Tabla de Prioridades

Número	Prioridad	Complejidad
1	Baja	Fácil
2	Media	Moderada
3	Alta	Complejo
4	Muy alta	Muy Complejo

Fuente: Elaboración Propia

A continuación se listan las razones por las que una solicitud de Pase o Incidencia (que contenga dsx), debe observada, según el CheckList de Revisiones. La lista siguiente sólo contempla errores en objetos dsx, ya que como mencionamos anteriormente, los Pases pueden ser Dsx o Db2.

Tabla 4: CheckList Revisión de Pases

N°	Descripción Breve	Resultado
1	Se debe registrar o seleccionar en el pase de tipo DataStage, objetos por proyecto indicando correctamente el nombre del mismo (ejem:prodProject). No se debe incluir en un mismo pase, objetos de dos proyectos.	Observado y Reprogramado
2	La lista de Objetos en la documentación Word Checklist debe coincidir con la lista de objetos en el DSX adjunto.	Observado y Reprogramado
3	Las evidencias de pruebas en IST, adjuntas como imágenes en el documento Word Checklist, deben evidenciar los Jobs con estado “Finalizados”, así mismo los campos Fecha, Hora de Inicio y Tiempo transcurrido.	Observado y Reprogramado
4	Las pruebas en IST deben ser realizadas con más de 0 registros, incluir imagen del Supervisor del Director mostrando todos los flujos y casos deberán ser testeados, excepto para casos de Duplicados donde solo se verifica la igualdad de PKs de las tablas de origen con el destino.	Rechazado

5	Las pruebas en IST deben ser apuntando al ambiente IST y no ha DESA. Referido tanto a la BD como a las rutas respectivas /ds/work/ist/istProjectName.	Rechazado
6	Objetos del pase deben tener una documentación completa y correctamente ingresada, Nombre del job, Breve descripción, Autor, Fecha, Nombre del desarrollador y Analista responsable, en caso de modificación se debe ingresar una breve descripción del cambio. (Incluyen todo tipo de objetos Datastage ya sean Secuenciales, Paralelos, Rutinas, ParameterSets, etc.)	Observado y Reprogramado
7	Eliminación de DataSets innecesarios al finalizar la ejecución del proceso. Se debe utilizar un stage tipo rutina donde se invoca el comando orchadmin, este stage debe incluir su control de errores utilizando el stage Termination Activity. Los datasets que tienen el prefijo OLD_ no deben ser eliminados.	Rechazado

8	<p>La utilización de archivos de trabajo (.txt, .csv, etc. que no sean datasets) deben ser eliminados en caso no sean requeridos posteriormente por algún proceso, esta eliminación debe incluirse al finalizar el flujo mismo, invocando a la rutina de eliminación de archivo a través de un stage tipo Routine con su respectivo Control de Errores.</p>	Rechazado
9	<p>Todos los stages Job Activity de los Jobs Secuenciales deben contemplar su control de errores utilizando el stage Termination Activity. Considerar que todo “error”, “warning” debe ser detectado por el job como error y detener la ejecución para evidenciar así la cancelación y mostrar el estado “Terminado anormalmente”.</p> <p>No debe utilizarse el stage Exception Handler para realizar el control de errores.</p>	Rechazado
10	<p>En los stages Termination Activity, si se personaliza el mensaje de las propiedades “Texto de registro” y “Texto del mensaje final” este debe ser acorde con el job que genera el error, para evitar dar falsa información en el log del job ante</p>	Rechazado

	una cancelación.	
11	Todo job principal debe ser reinicializable, para ello este debe tener habilitada la propiedad "Añadir puntos de interrupción para que la secuencia se pueda reiniciar en caso de anomalía" y adicionalmente todos los stages Job Activity de los Jobs Secuenciales internos deben tener activa la propiedad de "Ejecutar y restablecer automáticamente".	Rechazado
12	El envío de correo debe obedecer, en el Asunto, el siguiente formato: "Ejecución Correcta Nombre_aplicativo_o_malla" o "Ejecución Anómala Nombre_aplicativo_o_malla"	Rechazado
13	La nomenclatura de objetos debe obedecer a los Estándares definidos, incluyendo nombre de Datasets.	Rechazado
14	Los valores en el proyecto como Nombre de BD, Usuario de conexión, Password, Rutas y Esquema deben ser parametrizables, a través de la utilización de parameter sets y variables de entorno a nivel de proyecto. No deben encontrarse	Rechazado

	durezas de código.	
15	En caso de modificación de parameterset, se debe incluir en el pase los Jobs que lo utilizan, ya que requieren recompilación, así como la evidencia en IST respectivamente.	Rechazado
16	En caso de modificación de rutinas, se debe incluir en el pase la evidencia en IST de todos los Jobs que la utilizan, para validar no sufra impacto alguno sobre otros procesos.	Rechazado
17	Los stages de conexión a orígenes o destinos, deben indicar una BD (DSN) y tabla existentes en el ambiente de Producción.	Rechazado
18	En los stages de Base de Datos no se debe ejecutar sentencias GRANT ni sentencias DDL, sólo se puede ejecutar sentencias DML.	Rechazado
19	Toda sentencia SQL ejecutada en un stage de Base de Datos, no debe tener comentarios.	Rechazado
20	En casos donde se utilice el stage tipo Agregator verificar la correcta configuración del modo, para	Rechazado

ordenamientos grandes utilizar SORT ya que el HASH tiene un límite y este causa problemas en ejecución.

Fuente: Elaboración Propia

En la siguiente tabla se listan los requerimientos referentes al Sistema.

Tabla 5: Requerimientos del Sistema (Product Backlog)

ID Requisito	Nombre Requisito	Nro. De Prioridad	Descripción	Complejidad
REQ01	Validar correcta Nomenclatura de Stages.	4	La aplicación validará la nomenclatura de los Stage conforme a los documentos básicos de Revisiones Nomenclaturas y Estándares DataStage.	4
REQ02	Validar correcta Nomenclatura de Job.	2	La aplicación validará si el nombre del Job se encuentra bien escrita según los estándares.	3
REQ03	Validar correcta Documentación del Job.	3	La aplicación validará el nombre del archivo del dataSet o sequential file (fuente o destino) interno, debe ser igual al nombre de Stage.	3
REQ04	Validar	2	La aplicación validará	3

	parametrización de los dataSet.		que la ruta del archivo este parametrizado.	
REQ05	Validar comentarios en el query del db2 Connector.	3	La aplicación validará si en el Stage db2 Connector se encuentra línea de comentarios el cual no está permitido.	2
REQ06	Validar puntos de comprobación en los Jobs secuenciales.	4	La aplicación validará si las casillas del Job secuencial se encuentran activadas cuando este es malla padre y dependa otro Job de este, así como también las otras casillas de acuerdo a los Estándares Scotiabank.	3
REQ07	Validar Query en el db2 Connector sin sentencia SELECT * FROM	4	La aplicación validará si en el Stage db2 Connector existen consultas con SELECT * FROM ya que estas no están permitidas.	2
REQ08	Validar nombre interno del Job Activity.	2	La aplicación validará si el nombre interno del Stage Job Activity concuerda con el nombre del Job Paralelo a que se hace referencia.	4
REQ09	Validar nomenclatura de variables	2	La aplicación validará si las variables creadas dentro del Stage	4

	del Transformer.		Transformer comienzan con el prefijo correcto "sv".	
REQ10	Validar de desencadenante en un Job secuencial.	2	La aplicación validará si la configuración de los desencadenantes del Stage son los correctos, ejemplo:"Condición contraria" su Stage destino es "Terminator Activity".	4
REQ11	Validar correcta descripción del Terminator Activity.	3	La aplicación validará si existen mensaje dentro del Stage Terminator Activity, de ser así debe coincidir con el que hace referencia.	3
REQ12	Validar correcta nomenclatura de rutinas.	2	La aplicación validará si la nomenclatura de la rutina está bien definida con su prefijo "rts".	3
REQ13	Validar funciones SQL no permitidas.	3	La aplicación validará el uso de cast, trim, smallint, to char, substring, coalesce, detectándolo como queries costoso.	3
REQ14	Validar uso de la sentencia Delete.	3	La aplicación validará si se hace uso de la sentencia Delete en cualquier punto del Stage db2 Connector como	3

			configuración o aplicadas en consultas.	
REQ15	Validar la propiedad Keep Conductor Alive.	4	La aplicación validará si la opción de la propiedad Keep Conductor Connection Alive esta seleccionado "SI".	3
REQ16	Validar el uso de Before/After SQL en el db2 Connector.	4	La aplicación validará si en la configuración de before/after SQL esta seleccionada la opción "Sí".	3
REQ17	Validar la propiedad Acción de Ejecución del Job activity.	4	La aplicación validará si el JobActivity tiene la configuración correcta "Restablecer si es necesario, y a continuación, ejecutar".	3

Fuente: Elaboración Propia

3.1.4 Determinación e Prioridades y complejidad por Requerimiento

3.1.4.1 Prioridades

Se ha determinado la prioridad para cada requerimiento, según lo establecido en dos reuniones, considerando también la opinión de los revisores de las solicitudes, a voz del Product Owner. Los revisores son los que pueden dar fe de las complicaciones que surgen cada vez que se les asigna un pase o incidencia. De esta manera, con lo acordado por los miembros del Equipo, se asignó un nivel de prioridad a cada requerimiento.

3.1.4.2 Complejidades

Para asignar la complejidad a cada requerimiento, se realizaron dos reuniones, una reunión para cada Sprint. Se definieron tareas por requerimiento y éstas a su vez fueron desglosadas en sub tareas de la siguiente manera:

Para tareas involucradas en prototipado de interfaces se tiene:

- Diseñar Interfaz
- Establecer controles de Interfaz

Para tareas involucradas en codificación se tiene:

- Establecer el lenguaje de programación.
- Establecer la plataforma para el desarrollo.

En las reuniones, cada miembro del equipo determinó valores para cada complejidad por subtarea, según su experiencia. La complejidad de cada requerimiento fue estimada por el responsable del Proyecto.

3.1.5 Análisis y diseño (Sprint BackLog)

Para esta etapa se estimaron tiempos, seleccionaron los requerimientos y se prepararon 2 Sprints con sus respectivas iteraciones.

3.1.5.1 Primera iteración

Para el primer Sprint se han considerado los requerimientos 01, 03, 05, 06, 07, 11, 13, 15, 16 y 17, ya que estos tienen la más alta

prioridad, La duración del primer Sprint fue de 60 días. Se inició en Octubre del 2016 y duró hasta fines de Noviembre del 2016. La presentación de los avances se realizó durante las reuniones con el usuario cada semana. Las reuniones con el equipo de desarrollo se realizaron diariamente, con un tiempo aproximado de 20 a 25 minutos, con la finalidad de conocer los avances de las tareas asignadas referentes a este Sprint.

3.1.5.1.1 Selección de los Requerimientos para la Primera Iteración.

A continuación se muestra la lista de requerimientos elegidos para el primer Sprint.

Tabla 6: Requerimientos del Sistema Primera Iteración

ID Requisito	Nombre Requisito	Nro. De Prioridad	Descripción	Complejidad
REQ01	Validar correcta Nomenclatura de Stages.	4	La aplicación validará la nomenclatura de los Stage conforme a los documentos básicos de Revisiones Nomenclaturas y Estándares DataStage.	4
REQ03	Validar correcta Documentación del Job.	3	La aplicación validará el nombre del archivo del	3

			dataSet o sequential file (fuente o destino) interno, debe ser igual al nombre de Stage.	
REQ05	Validar comentarios en el query del db2 Connector.	3	La aplicación validará si en el Stage db2 Connector se encuentra línea de comentarios el cual no está permitido.	2
REQ06	Validar puntos de comprobación en los Jobs secuenciales.	4	La aplicación validará si las casillas del Job secuencial se encuentran activadas cuando este es malla padre y dependa otro Job de este, así como también las otras casillas de acuerdo a los Estándares Scotiabank.	3
REQ07	Validar Query en el db2 Connector sin sentencia SELECT * FROM	4	La aplicación validará si en el Stage db2 Connector existen consultas con SELECT * FROM ya que estas no están permitidas.	2

REQ11	Validar correcta descripción del Terminator Activity.	3	La aplicación validará si existen mensaje dentro del Stage Terminator Activity, de ser así debe coincidir con el que hace referencia.	3
REQ13	Validar funciones SQL no permitidas.	3	La aplicación validará el uso de cast, trim, smallint, to char, substring, coalesce, detectándolo como queries costoso.	3
REQ15	Validar la propiedad Keep Conductor Alive.	4	La aplicación validará si la opción de la propiedad Keep Conductor Connection Alive esta seleccionado "SI".	3
REQ16	Validar el uso de Before/After SQL en el db2 Connector.	4	La aplicación validará si en la configuración de before/after SQL esta seleccionada la opción "Sí".	3
REQ17	Validar la propiedad Acción de Ejecución del Job activity.	4	La aplicación validará si el JobActivity tiene la configuración correcta "Restablecer	3

		si es necesario, y a continuación, ejecutar".	
--	--	---	--

Fuente: Elaboración Propia

En la siguiente tabla se listan las tareas realizadas en tiempo estimado definido por el equipo de desarrollo.

Tabla 7: Tareas para la primera iteración

ID Tarea	Tarea	Responsable	Tiempo Estimado
TR001	Diseño de la Interfaz del Validador.	Desarrolladores	5 horas
TR002	Generación de código para identificar la correcta nomenclatura de cada Stage.	Desarrolladores	36 horas
TR003	Generación de código para identificar documentación del Job.	Desarrolladores	4 horas
TR004	Generación de código para reconocer configuración del db2 Connector.	Desarrolladores	4 horas
TR005	Generación de código para identificar comentarios en Query de db2 Connector.	Desarrolladores	2 horas
TR006	Generación de código para identificar Jobs secuenciales.	Desarrolladores	3 horas
TR007	Generación de código para identificar descripciones en el Stage Terminator Activity.	Desarrolladores	2 horas

TR008	Generación de código para identificar funciones en el query del db2 Connector.	Desarrolladores	2 horas
TR009	Generación de código para identificar Job Activity dentro de un Job secuencial.	Desarrolladores	2horas
TR010	Revisión de los prototipos realizados.	Líder de Proyectos - SES	4 horas
TR011	Pruebas de Primer Sprint.	Jefe de Centro de Información Scotiabank	3 horas
TOTAL			71 horas

Fuente: Elaboración Propia.

3.1.5.1.2 Criterios de determinación de prioridad y complejidad por Requerimiento

Para la prioridad, se tomó en cuenta la opinión de cada miembro del equipo de desarrollo por requerimiento, bajo la decisión del Jefe de Proyecto.

Para la complejidad, los integrantes del equipo de desarrollo llegaron a un acuerdo para establecerla.

3.1.5.1.3 Generación y Seguimiento del Sprint BackLog

Se realizó el listado de las tareas que se deben cumplir en la primera iteración. Así mismo se creó una tabla que muestra los datos generales para este primer Sprint.

La Tabla 7, está dividida en dos partes. La primera, consta de lo siguiente:

- Nombre del Proyecto.
- Número del Sprint.
- Fecha de Inicio del desarrollo del Sprint.
- Tiempo en días de desarrollo del Sprint.
- Tiempo en horas que demorará el desarrollo del Proyecto.

La segunda parte de la Tabla 7, consta de:

- Tipo de tarea.
- Análisis.
- Codificación.
- Prototipado o Pruebas.
- Reunión.
- Estado del desarrollo (Pendiente, En curso, Terminada, Eliminada)
- Equipo de desarrollo.

- Festivos, se menciona los días en los que el equipo de desarrollo no labora.

Tabla 8: Datos Generales para el primer Sprint

Proyecto			
Validador de Nomenclaturas			
N° de Sprint	Inicio	Días	Jornada(Horas)
1	03/10/2016	40	9
Tareas		Equipo	Festivos
Tipos	Estados		
*Análisis	*Pendiente	*Jefe de Centro de Información	1 de Noviembre
*Codificación	*En Curso	*Líder de Proyectos SES	
*Prototipado	*Terminada	*Desarrolladores	
*Pruebas	*Eliminada		
*Reunión			

Fuente: Elaboración propia.

En la primera iteración se realizó un seguimiento del cumplimiento de cada una de las tareas mencionadas en la Tabla 7.

Mediante la aplicación de la metodología SCRUM referente a sus reuniones diarias, se permitió conocer los avances de cada día realizados por el equipo, así se verificó el estado en el que se encontraba el desarrollo de tareas.

Si bien las tareas fueron presentadas en la pila, pero no se completaron algunas, se procede a finalizar las tareas.

Una vez culminada la iteración; en la tabla 7 se listaran las tareas y su progreso durante el desarrollo de la primera iteración.

3.1.5.2 Segunda Iteración

Para la segunda iteración, se ha considerado los mismos puntos que en la primera iteración. La finalidad de esta segunda iteración fue tener la segunda versión del Validador para el día 25 de Noviembre del 2016.

Así mismo las reuniones se realizaron diariamente para verificar el avance del desarrollo de las tareas.

3.1.5.2.1 Selección de los Requerimientos para la Segunda Iteración.

A continuación se listarán los requerimientos para la segunda iteración. Cabe mencionar que estos requerimientos forman parte de la lista principal, y no fueron considerados en la primera iteración.

Tabla 9: Requerimientos para la Segunda Iteración

ID Requisito	Nombre Requisito	Nro. De Prioridad	Descripción	Complejidad
REQ02	Validar correcta Nomenclatura de Job.	2	La aplicación validará si el nombre del Job se encuentra bien escrita según los estándares.	3
REQ04	Validar parametrización de los dataSet.	2	La aplicación validará que la ruta del archivo este parametrizado.	3
REQ08	Validar nombre interno del Job Activity.	2	La aplicación validará si el nombre interno del Stage Job Activity concuerda con el nombre del Job Paralelo a que se hace referencia.	4
REQ09	Validar nomenclatura de variables del Transformer.	2	La aplicación validará si las variables creadas dentro del Stage Transformer comienzan con el prefijo correcto "sv".	4
REQ10	Validar de desencadenante en un Job secuencial.	2	La aplicación validará si la configuración de los desencadenantes del Stage son los correctos, ejemplo:"Condición	4

			contraria” su Stage destino es “Terminator Activity”.	
REQ12	Validar correcta nomenclatura de rutinas.	2	La aplicación validará si la nomenclatura de la rutina está bien definida con su prefijo “rts”.	3
REQ14	Validar uso de la sentencia Delete.	3	La aplicación validará si se hace uso de la sentencia Delete en cualquier punto del Stage db2 Connector como configuración o aplicadas en consultas.	3

Fuente: Elaboración Propia

Los requerimientos listados serán desglosados en tareas, las cuales serán asignadas a los responsable; tendrán un tiempo estimado para realizar cada tarea. La siguiente tabla muestra la lista de tareas antes mencionada.

Tabla 10: Tareas a realizarse para la Segunda Iteración

ID Tarea	Tarea	Responsable	Tiempo Estimado
TR001	Generación de código para reconocimiento de periodicidad en el nombre del Job.	Desarrolladores	4 horas
TR002	Crear código que lea parámetros comparando con variables de entorno.	Desarrolladores	5 horas
TR003	Generación de código para identificar nombre del Stage Job Activity.	Desarrolladores	4 horas
TR004	Generación de código para reconocer variables dentro del Stage Transformer.	Desarrolladores	2 horas
TR005	Generación de código para identificar desencadenantes en un Job Secuencial.	Desarrolladores	2 horas
TR006	Generación de código para identificar rutinas	Desarrolladores	3 horas
TR007	Generación de código para verificar documentación de rutinas.	Desarrolladores	3 horas
TR008	Generación de código para identificar sentencia Delete en la configuración del db2 Connector.	Desarrolladores	2 horas

TR010	Revisión de los prototipos realizados	Líder de Proyectos - SES	3 horas
TR011	Pruebas de Primer Sprint	Jefe - Centro de Información Scotiabank	3 horas
TOTAL			31 horas

Fuente: Elaboración Propia

3.1.5.2.2 Criterios para la determinación de prioridad y complejidad por requerimiento

Para considerar una prioridad a cada requerimiento se solicitó la opinión del equipo de desarrollo, las decisiones fueron tomadas por el Jefe del Proyecto.

Para asignar la complejidad a cada requerimiento se tomó en cuenta la opinión del equipo de desarrollo como punto inicial. Se llegó a un acuerdo general entre los integrantes del equipo de desarrollo.

3.1.5.2.3 Generación y seguimiento de Sprint BackLog de la Segunda Iteración

Una vez obtenida la lista de tareas de la Segunda Iteración, se genera el inicio del Sprint con fecha 12 de

Diciembre del 2016, con un tiempo diario de 9 horas por día.

Tabla 11: Datos Generales de Segunda Iteración.

Proyecto			
Validador de Nomenclaturas			
N° de Sprint	Inicio	Días	Jornada(Horas)
2	12/12/2016	30	9

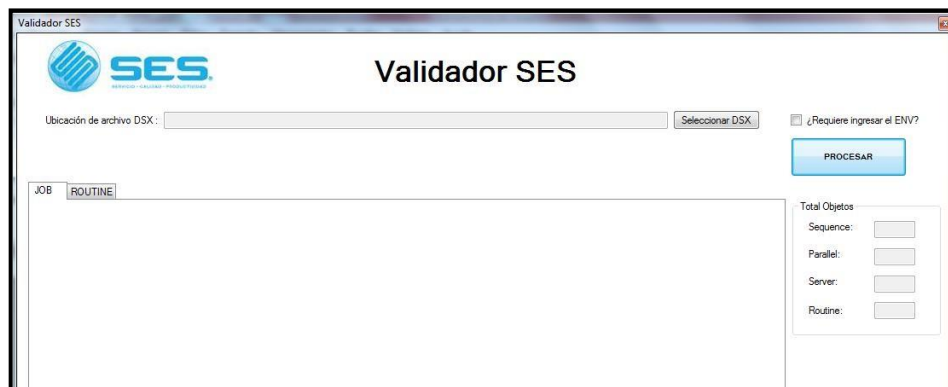
Fuente: Elaboración Propia

En esta Segunda iteración, al igual que la primera, se realizó un seguimiento del avance de tareas que fueron repartidas al equipo de desarrollo.

3.2 DISEÑO O SIMULACIÓN DE LA HERRAMIENTA

En la siguiente Figura 3 se observa la Interfaz Principal del Validador. Este validador será mejorado conforme cambien los requerimientos de Scotiabank con respecto a las consideraciones en las Revisiones.

Figura 3: Validador SES – Interfaz Principal





Fuente: Elaboración Propia

3.2.1 Consideraciones para el uso del Validador

Para el correcto funcionamiento del validador se debe contar con los siguientes archivos:

- Archivo dsx, el cual contiene los Jobs. Este archivo es exportado desde el Diseñador DataStage, es obligatorio.
- Archivo env, que contiene las variables del proyecto. Este archivo es exportado desde el Administrador DataStage, es opcional, mayormente se envía cuando el analista envía dsx con Jobs que son nuevos en el proyecto a impactar.

Figura 4: Archivos dsx y env

Nombre	Fecha de modifica...	Tipo	Tamaño
 TEST_DB2_FROM.dsx	02/11/2016 04:05 ...	Archivo DSX	25 KB
 TEST_variablesEntorno.env	02/11/2016 04:01 ...	Archivo ENV	7 KB

Fuente: Elaboración Propia

3.2.2 Carga de Archivos

Para revisar los objetos, se debe realizar la carga de estos. Para esto hacer click en el botón “Seleccionar DSX”. Se debe tener sólo archivos dsx correctamente exportados para su importación exitosa.

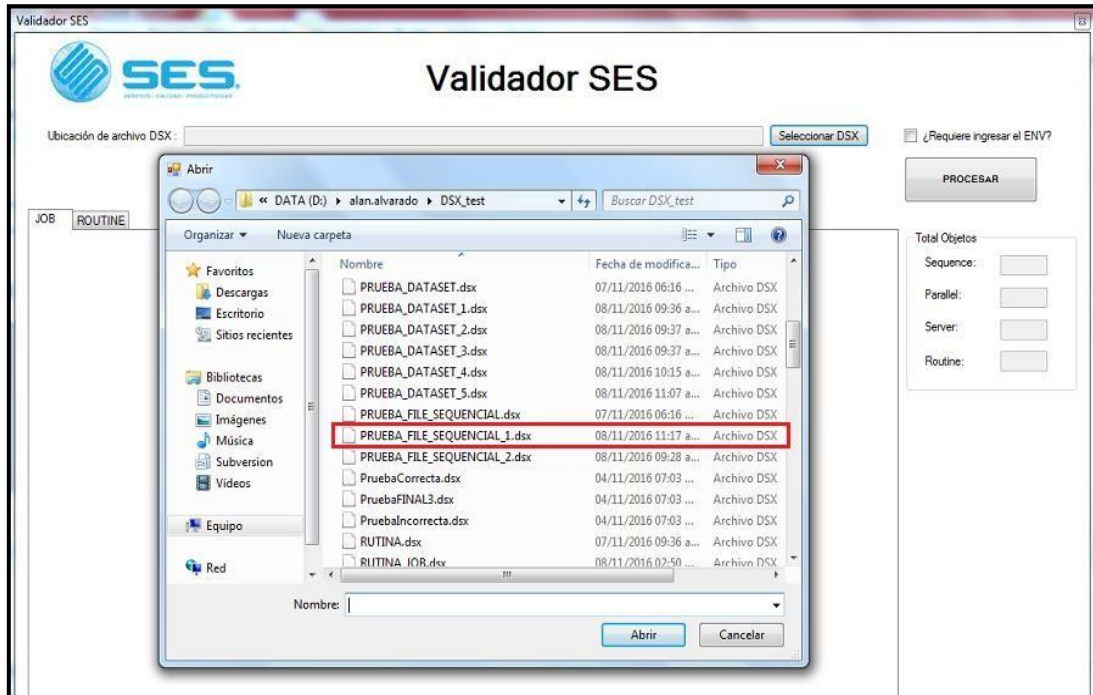
Figura 5: Seleccionando dsx



Fuente: Elaboración Propia

Ubicar la ruta y el archivo dsx, y seleccionar con el botón “Abrir” como lo muestra la siguiente imagen.

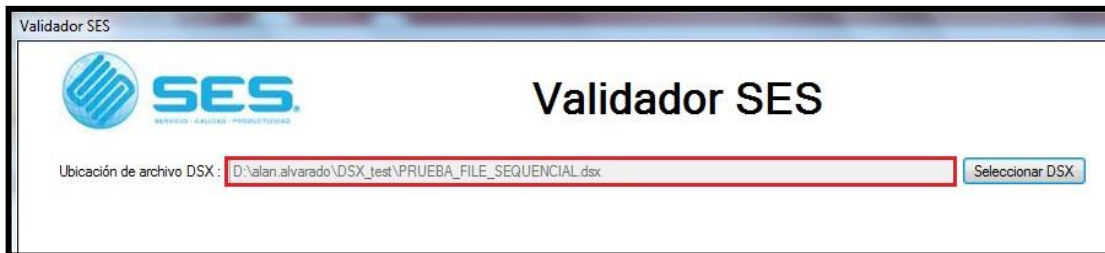
Figura 6: Seleccionar dsx y env



Fuente: Elaboración Propia.

En la siguiente imagen se aprecia la ruta seleccionada. Verificar que se la correcta, es decir. Donde se encuentra su archivo dsx.

Figura 7: Ubicación de archivo dsx

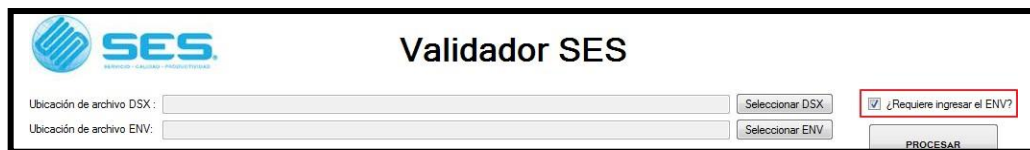


Fuente: Elaboración Propia.

Si se requiere validar los parámetros del Job secuencial y paralelo se debe seleccionar el check que se ubica en la parte superior

derecha de la pantalla, esto habilitará el botón para seleccionar un archivo “ENV” que contiene las variables de entorno del proyecto que contiene el archivo DSX.

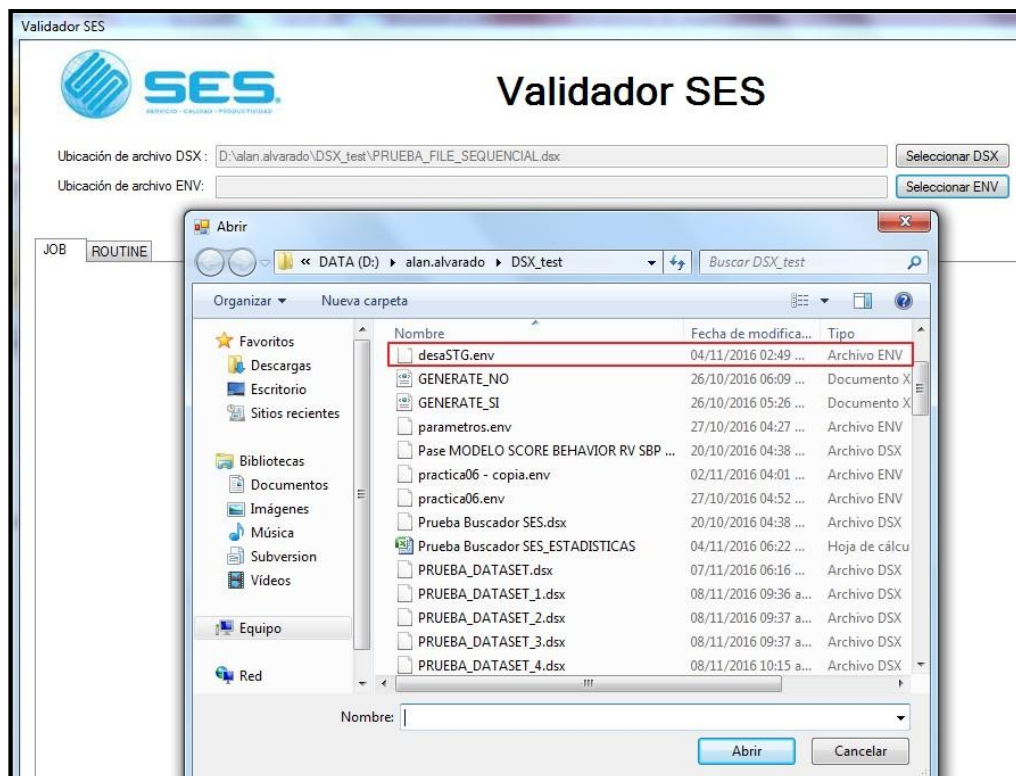
Figura 8: Requiere ingresar env



Fuente: Elaboración Propia.

En la siguiente figura se puede ubicar la ruta y el archivo .env y seleccionar con el botón “Abrir”.

Figura 9: Seleccionar archivo env

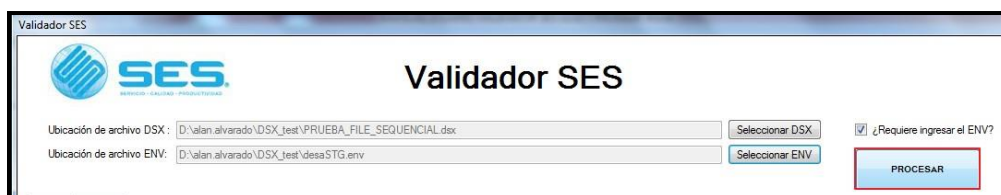


Fuente: Elaboración Propia

3.2.3 Verificación de Errores – Observaciones

Una vez cargados los archivos DSX y ENV, hacer click en el botón “PROCESAR”. El validador se tomará unos segundos para leer todos los objetos.

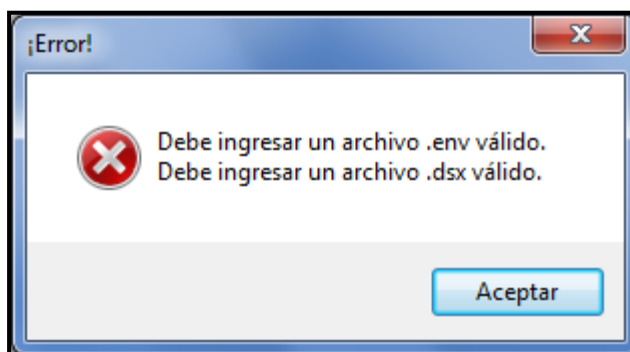
Figura 10: Procesar validación



Fuente: Elaboración Propia

En caso de haber seleccionado un archivo no válido o no haber cargado ningún archivo se mostrará un mensaje de error.

Figura 11: Mensaje de Error



Fuente: Elaboración Propia

3.2.3.1 Listado de Jobs

En la primera pestaña se mostrará en el cuadro de detalle la lista de Jobs que contiene el Dsx, el cual detalla nombre de job, tipo de Job (Sequence, parallel o server), número de stages que contiene el Job,

estado (OK, NOT OK) que muestra el estado general del job para la revisión.

Para este ejemplo, el archivo dsx sólo contiene un Job Sequence, cuarenta y cuatro stages y una rutina.

Además, se muestra un botón, el cual limpia todos los campos y permite una nueva carga de archivos a revisar. También, un cuadro estadístico el cual indica la cantidad de Jobs por tipo y las rutinas contenidas.

Figura 12: Reportes

The screenshot displays the 'Validador SES' application window. At the top, the title bar reads 'Validador SES'. Below the title bar, the 'Ubicación de archivo DSX' field contains the path 'C:\Users\o95181\Desktop\ROUTINAS\SEQ dsx'. To the right of this field is a 'Seleccionar DSX' button. Below the path field, there is a checkbox labeled '¿Requiere ingresar el ENV?' and a 'PROCESAR' button with a magnifying glass icon. A red box highlights the 'Contiene lista de job' label above the table.

Abbr	Nombre Job	Tipo de Job	Nro Stages	Estado	Fecha Exportación	Fecha Últ. mod.
>	SEQ_PC_HISTORICOS_FSH611	Sequence	44	NOTOK	2016-11-08 18:30:32	2016-11-07 11:37:56

Below the table, there are two buttons: 'ESTADÍSTICAS' and 'RESULTADOS DE VALIDACIÓN'. To the right of the table, a 'Total Objetos' section contains four rows of statistics, each with a red box around the value and an arrow pointing to a label:

- Sequence: 1 → Cantidad de job secuenciales
- Paralelo: 0 → Cantidad de job paralelo
- Server: 0 → Cantida de job server
- Routine: 1 → Cantidad de rutinas

Fuente: Elaboración Propia

El botón “RESULTADOS DE VALIDACIÓN” se utiliza para guardar reportes (Detalle de la grilla) en un archivo Excel.

3.2.3.2 Listado de Rutinas

En la segunda pestaña se mostrará en el cuadro de detalle la lista de rutinas que contiene el Dsx, el cual detalla nombre de rutina, tipo de rutina (Parallel o server), número de argumento que contiene la rutina, estado general de la rutina conforme la revisión (OK, NOT OK). Así mismo se tiene el detalle de la fecha de creación y modificación del objeto.

Figura 13: Validador SES – Rutinas

The screenshot displays the 'Validador SES' application window. At the top left is the SES logo with the tagline 'SERVICIO - CALIDAD - PRODUCTIVIDAD'. The main title is 'Validador SES'. Below the title, there is a text field for 'Ubicación de archivo DSX:' containing the path 'C:\Users\ic95181\Desktop\RUTINASYSEQ.dsx' and a 'Seleccionar DSX' button. To the right, there is a checkbox labeled '¿Requiere ingresar el ENV?' and a 'PROCESAR' button with a yellow lightning bolt icon. Below this, there are two tabs: 'JOB' and 'ROUTINE', with 'ROUTINE' selected. A red box highlights the 'ROUTINE' tab and a label 'Lista de rutinas' with an arrow pointing to it. The main area contains a table with the following data:

Abbr	Nombre Rutina	Tipo de Rutina	Nro Arg.	Estado	Fecha Exportación	Fecha últ. mod.
>	rtsDeleteAllDataSet.Arg	Server	2	OK	2016-11-08 18.30.32	2016-09-30 09.23.33

On the right side, there is a 'Total Objetos' section with four input fields: 'Sequence:' (value 1), 'Parallel:' (value 0), 'Server:' (value 0), and 'Routine:' (value 1). Below this are two buttons: 'ESTADÍSTICAS' and 'RESULTADOS DE VALIDACIÓN'.

Fuente: Elaboración Propia

3.2.3.3 Listado de Stages

Seleccionar un botón de la primera columna del job, cuyos stages desee ver.

Figura 14: Validador SES – Listado Jobs

The screenshot shows the 'Validador SES' application window. At the top, there is a logo for 'SES' and the title 'Validador SES'. Below the logo, there is a text field for 'Ubicación de archivo DSX:' containing the path 'C:\Users\lo95181\Desktop\RUTINASYSEQ.dsx' and a 'Seleccionar DSX' button. To the right, there is a checkbox labeled '¿Requiere ingresar el ENV?' and a 'PROCESAR' button with a yellow bell icon. Below this, there is a 'JOB' tab and a 'ROUTINE' tab. The main area contains a table with the following data:

Abir	Nombre Job	Tipo de Job	Nro Stages	Estado	Fecha Exportación	Fecha últ. mod.
>	SEQ_PC_HISTORICOS_FSH611	Sequence	44	NOTOK	2016-11-08 18.30.32	2016-11-07 11.37.56

Below the table, there is a 'Total Objetos' section with four input fields: 'Sequence:' (value 1), 'Paralel:' (value 0), 'Server:' (value 0), and 'Routine:' (value 1). At the bottom right, there are two buttons: 'ESTADÍSTICAS' and 'RESULTADOS DE VALIDACIÓN'.

Fuente: Elaboración Propia

Se mostrará la ventana “Detalle TIPO DE JOB – NOMBRE DEL JOB”, con el listado de Stages contenidos en el Job seleccionado, la cual detalla:

- Nombre del Job
- Ruta del Job
- Validaciones del Job.
- Botón para visualizar todas las validaciones del Job.

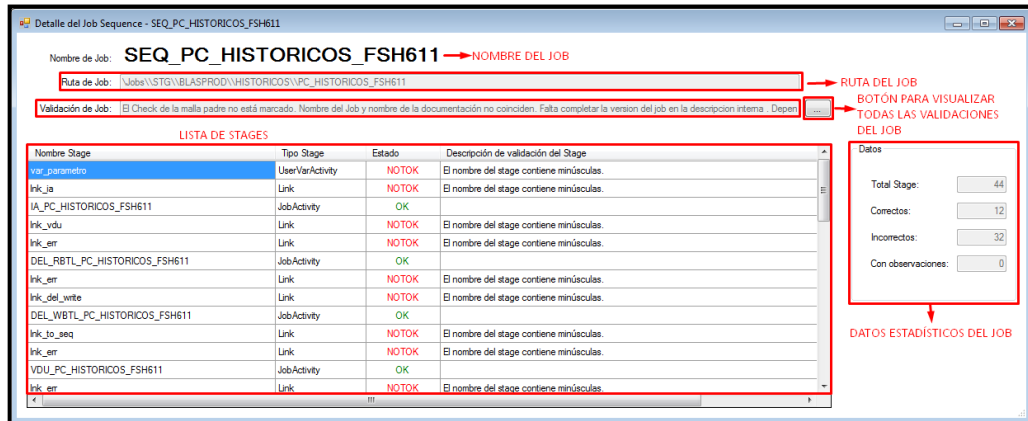
Dentro de la grilla se detalla la lista de los stages contenidos en el Job.

- Nombre del stage.
- Tipo de stage.
- Estado de stage (el estado del stage es incorrecto, observado o correcto)
- Descripción de la validación del stage. Detallará observaciones y errores que tiene el stage.

En el lado derecho de la ventana se muestran datos estadísticos del job.

- Total de stages dentro del Job.
- Cantidad de stages correctos,
- Cantidad de stages incorrectos,
- Cantidad de stages con observaciones.

Figura 15: Validador SES – Listado Detalle

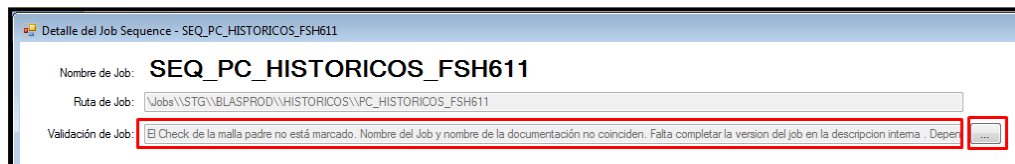


Fuente: Elaboración Propia.

En caso que la documentación del Job sea correcta, el campo de “Validación de Job” estará vacío, caso contrario, se llenará automáticamente.

Hacer doble click en el botón con texto “...” para ver el detalle, como lo muestra la siguiente imagen:

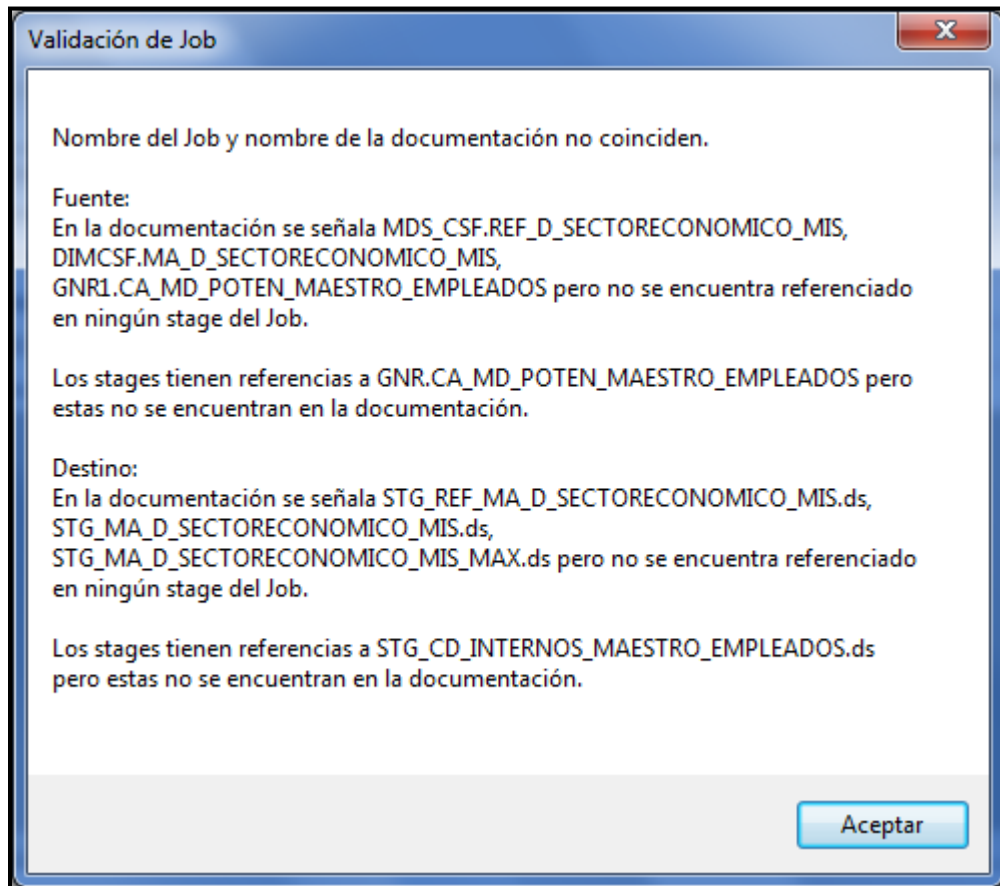
Figura 16: Validador SES – Error Documentación



Fuente: Elaboración Propia

Se mostrará la siguiente ventana “Validación de Job” que describe las observaciones y errores del Job.

Figura 17: Validador SES – Error Documentación Job



Fuente: Elaboración Propia

Una vez verificado los errores de documentación, hacer click en “Aceptar” para regresar a la ventana Detalle de Job Parallel, para revisar los errores del siguiente stage si se tuviera.

3.2.3.4 Listado de Argumentos

Seleccionar un botón de la primera columna de la rutina, la cual se desea ver el detalle de sus argumentos.

Figura 18: Validador SES - Argumentos

The screenshot shows the 'Validador SES' application window. At the top, there is a logo for 'SES' (SERVICIOS - CALIDAD - PRODUCTIVIDAD) and the title 'Validador SES'. Below the logo, there is a text field for 'Ubicación de archivo DSX' with the path 'C:\Users\o95181\Desktop\VARIOSROUTINES.dsx' and a 'Seleccionar DSX' button. To the right, there is a checkbox for '¿Requiere ingresar el ENV?' and a 'PROCESAR' button with a yellow lightning bolt icon.

The main area is divided into two sections. On the left, there is a 'JOB' dropdown menu with 'ROUTINE' selected. Below it is a table with the following columns: 'Abrir', 'Nombre Routine', 'Tipo de Routine', 'Nro Arg.', 'Estado', 'Fecha Exportación', and 'Fecha últ. mod.'. The table contains five rows of data:

Abrir	Nombre Routine	Tipo de Routine	Nro Arg.	Estado	Fecha Exportación	Fecha últ. mod.
>	rstDeleteAllDataSet	Server	1	NOTOK	2016-11-07 12.24.04	2016-09-30 09.23.12
>	rstDeleteAllDataSetArg	Server	2	NOTOK	2016-11-07 12.24.04	2016-09-30 09.23.13
>	rstDeleteAllDataSetV2	Server	1	NOTOK	2016-11-07 12.24.04	2016-09-30 09.23.15
>	rstPipExisteFile	Server	6	NOTOK	2016-11-07 12.24.04	2016-09-30 09.23.16
>	rteCopiaDataSet	Server	2	OK	2016-11-07 12.24.04	2016-09-30 09.23.23

On the right side, there is a 'Total Objetos' section with four input fields: 'Sequence' (0), 'Paralel' (0), 'Server' (0), and 'Routine' (5). Below this are two buttons: 'ESTADÍSTICAS' and 'RESULTADOS DE VALIDACIÓN'.

Fuente: Elaboración Propia

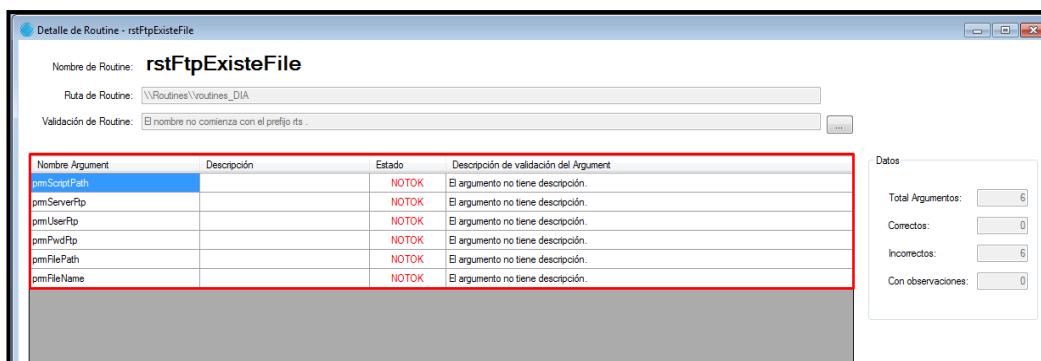
Se mostrará la ventana “DETALLE DE ROUTINE – NOMBRE DE LA RUTINA”, con el listado de argumentos contenido en la rutina seleccionada, la cual detalla:

- Nombre de la rutina.
- Ruta de la Rutina.
- Validaciones de la rutina.
- Botón para visualizar todas las validaciones de la rutina.
- Dentro de la grilla se detalla la lista de los argumentos contenidos en la rutina.
- Estado del argumento (El estado del argumento es correcta, observado e incorrecta, respectivamente).
- Descripción de la validación del argumento. Detallará observaciones y errores que tiene el argumento.

En el lado derecho de la ventana se muestra datos estadísticos de la rutina.

- Total de argumentos dentro de la rutina.
- Cantidad de argumentos correctos.
- Cantidad de argumentos incorrectos.
- Cantidad de argumentos con observaciones.

Figura 19: Validador SES – Argumentos Error



Fuente: Elaboración Propia

3.2.4 Guardar Reporte

Para guardar los Reportes de la validación, ubicarse en la ventana principal “Validador SES”

Hacer click sobre el botón. El botón “RESULTADOS DE VALIDACIÓN” permite exportar el listado de los Jobs, Stages, Rutinas y Argumentos validados en un archivo Excel.

Se mostrará la siguiente ventana, por defecto el nombre del archivo excel sera el de DSX concantenado con el nombre de la opción seleccionada como se visualiza en el ejemplo, luego se escoge una ruta y guardar el archivo.

Los datos en el Excel se mostrarán de la siguiente manera.

Reporte de validaciones.

En el archivo excel de reporte de validaciones se visualizará dos libros por defecto “Estadísticas de JOB”, “Detalle de JOB”, “Estadísticas de ROUTINE” y “Detalle de ROUTINE”.

En el libro de Estadísticas de JOB se visualizará de la siguiente manera.

- Nombre del Job.
- Tipo de Job.
- Número de Stages.
- Estado.
- Fecha de exportación.
- Fecha de última modificación.
- Números correctos de Stage.
- Números incorrectos de Stage.
- Número de observaciones.
- Descripción de validación del Job.
- Ruta del Job.

En el libro de Detalle de Job se visualizará de la siguiente manera.

- Nombre del Job.
- Tipo de Job.
- Nombre de Stage.

- Tipo de Stage.
- Estado.
- Descripción de validación del Stage.

Figura 22: Validador SES – Salida

A	B	C	D	E	
Nombre Job	Tipo de Job	Nombre Stage	Tipo Stage	Estado	Descripción de validación del Stage
SEQ_PC_HISTORICOS_FSH611	Sequence	var_parametro	UserVarActivity	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_ia	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	IA_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_vdu	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	DEL_RBTL_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_del_write	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	DEL_WBTL_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_to_seq	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	VDU_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_to_var	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	DED_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_to_seq	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	EST_PC_HISTORICOS_FSH611	JobActivity	OK	
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_err	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_to_var	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	var_duplicados	UserVarActivity	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_to_nec	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	nec_condicional	NestedCondition	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_no_dup	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	lnk_si_dup	Link	NOTOK	El nombre del stage contiene minúsculas.
SEQ_PC_HISTORICOS_FSH611	Sequence	seq_all	Sequencer	NOTOK	El nombre del stage contiene minúsculas.

Fuente: Elaboración Propia

En el libro de Estadísticas de ROUTINE se visualizará de la siguiente manera.

- Nombre de la rutina.
- Tipo de rutina.
- Número de argumento.
- Estado.
- Fecha de exportación.
- Fecha de última modificación.
- Número correctos de argumento.
- Número incorrectos de argumento.
- Número de observaciones.
- Descripción de validación de rutina.
- Ruta de la rutina.

En el libro de Detalle de ROUTINE se visualizará de la siguiente manera.

- Nombre del routine.
- Tipo de routine.
- Nombre de argumento.
- Descripción.
- Estado.
- Descripción de validación del argumento.

3.3 REVISIÓN Y CONSOLIDACIÓN DE RESULTADOS

En este capítulo se demuestra que el uso del Validador de Nomenclaturas para las Buenas prácticas de DataStage, aplicado a Scotiabank, mediante la metodología SCRUM, brindó los resultados esperados en la minimización del tiempo en revisiones de Pases e Incidencias que contengan DSX.

3.3.1 Resultados de la Investigación

3.3.1.1 El tiempo de Revisión

Se redujo el tiempo que se toma una persona al revisar los objetos DataStage. Normalmente, cada revisor se tomaba el tiempo de 10Jobs por hora; ahora considerablemente, el validador toma 1 minuto para mostrar las observaciones que se encontraron en cada uno de los Jobs del archivo DSX.

Tomamos por ejemplo, un caso en el que se envía un dsx con 10 objetos que contienen un promedio de 220 stages para su Revisión.

Se estiman los tiempos para el caso que se realice la Revisión manualmente (Antes) comparándolo con la Revisión hecha por el Validador.

Tabla 12: Comparación de tiempos

	Tiempo de Revisión (min)	
	Antes	Después
Revisión de Nomenclatura	40	5
Revisión de Documentación	15	3
Otros	15	15
Realización de Documento de observaciones	25	25
Total tiempo estimado	95	48

Fuente: Elaboración Propia

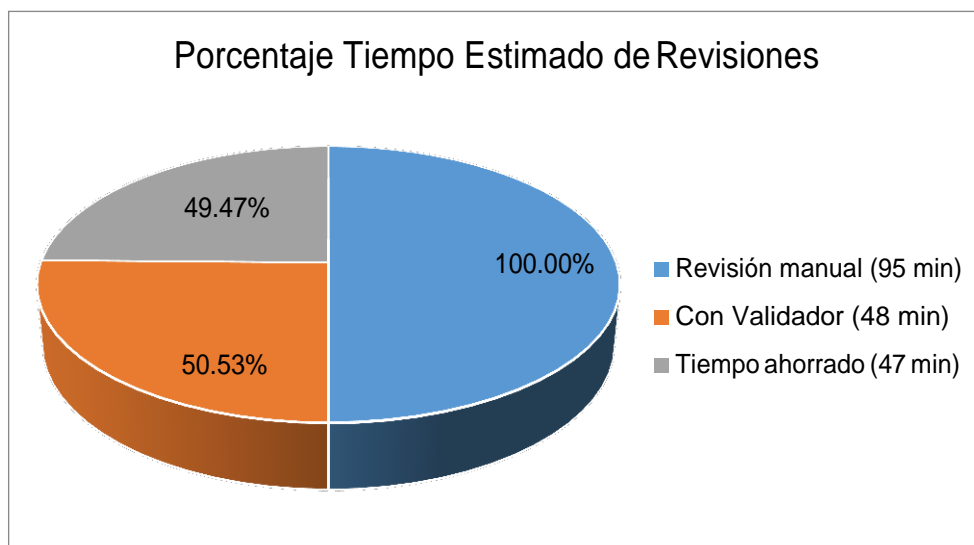
Se aprecia que el tiempo en la entrega del documento de Revisiones ha disminuido considerablemente de los 95 minutos a los 48 minutos.

Tabla 13: Tiempo y Porcentaje Revisiones

	Tiempo (min)	Porcentaje
Sin el validador	95	100.00%
Con el validador	48	50.53%
	Tiempo ahorrado	49.47%

Fuente: Elaboración Propia

Figura 23: Porcentaje Tiempo Estimado de Revisiones



Fuente: Elaboración Propia

En el pastel se aprecia la comparación de tiempos, cuando la Revisión se realizaba manualmente, y cuando se emplea el Validador de Nomenclaturas.

En lo que se refiere a costos, la empresa otorga un sueldo mensual de X soles a un revisor en el área de Centro de Información. Por tanto, lo que un revisor recibe al día equivale a $X/30$ soles, y $X/270$ soles es lo que recibe cada hora, puesto que trabaja 9 horas por día.

Por lo general la revisión de 10 jobs se toma el tiempo de 1 hora, tomando en cuenta que la cantidad de stages no supere las 200 unidades. Según esto, se tiene la siguiente ecuación:

Tiempo – Revisión – Costo

1h – 10 Jobs – $(X/30)/9 = X/270 \rightarrow X/270$ soles por 1 hora

Con el uso del validador, el revisor se tomará 30 minutos en promedio para realizar la revisión de los 10 jobs. Es decir en una hora podrá revisar 20 jobs.

1h – 20 jobs – $2(X/270) = X/135 \rightarrow X/135$ soles por 1 hora

Con la ayuda del validador, el revisor estaría realizando la labor de dos horas en una sola, ya que la revisión de 20 jobs en una hora equivalen a dos revisores. Por la hora de revisión, la empresa paga $X/270$ soles, por lo que estaría ahorrando otros $X/270$ soles al día gracias al uso del validador.

Al mes se tiene un ahorro de X soles, lo que equivale al sueldo mensual de un revisor, añadido a esto, se tiene un mejor tiempo de respuesta en la revisión de pases que se impactarán en el ambiente de producción.

Con lo antes mencionado, se deduce que el costo de dos revisores equivale al costo de 1 revisor con el uso de validador, dado que pueden revisar la misma cantidad de pases.

Se estima el sueldo del revisor en 1500 soles mensual y el costo del validador en 5000 soles. A continuación se detalla la comparación de costos a lo largo de un año.

Tabla 14: Comparación de Costos

Mes	2 revisores S/.	1 revisor + validador S/.
1	3000	1500 + 5000
2	3000	1500
3	3000	1500
...
12	3000	
Total	36000	23000

Fuente: Elaboración Propia

El primer mes se pagará el validador y los demás meses se considerará su uso.

En el cuadro anterior se observa que en un año, contratando dos revisores, se necesitan 36000; mientras que, si reemplazamos 1 revisor por el validador, sólo necesitamos 23000.

Con esto, concluimos que el uso del validador permite tener a Scotiabank un ahorro de 13000 soles al año.

Se tiene el siguiente cuadro el cual muestra el costo acumulado durante los primeros meses.

Tabla 15: Comparación de Costos Acumulado

Mes	Acumulado (2 revisores)	Acumulado (1 revisor + validador)
1	3000	6500
2	6000	8000
3	9000	9500
4	12000	11000

Fuente: Elaboración Propia

Se puede observar que a partir del cuarto mes Scotiabank estará gastando menos con el uso del validador, puesto que ya se habrá recuperado lo invertido tanto en costo como en tiempo.

3.2.2 Resultados de los Objetivos

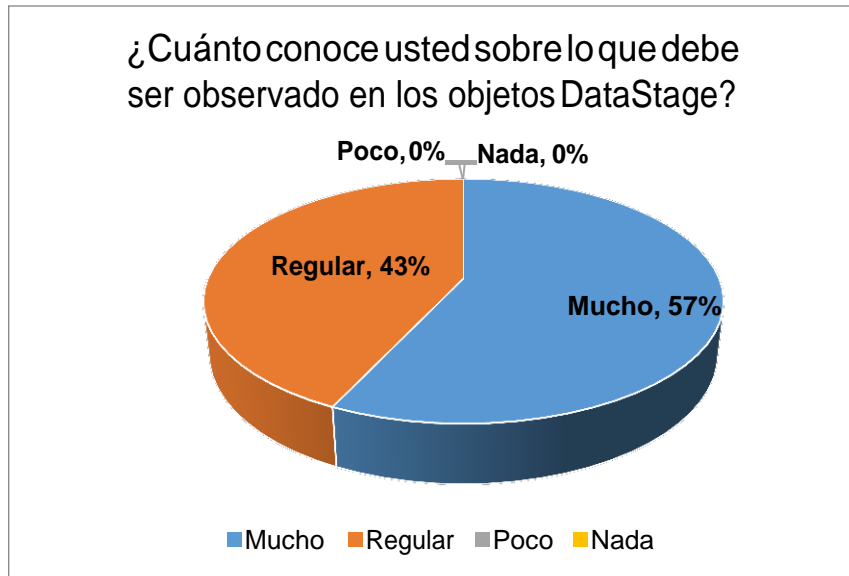
A continuación se describen los resultados de una breve encuesta para valorar los resultados con respecto a los objetivos específicos.

Tabla 16: Encuesta de Percepción – Pregunta 1

OBJETIVO ESPECIFICO 1		
PREGUNTA 1		
Fecha: 09/01/2017		
¿Cuánto conoce usted sobre lo que debe ser observado en los objetos DataStage?		
ENCUESTADOS		
Roles:	Cantidad	
1. Jefes de Centro de Información	2	
2. Equipo de Desarrollo	2	
2. Revisores	3	
CUADRO DE FRECUENCIAS		
Categoría	Frecuencia	Porcentaje
Mucho	4	57%
Regular	3	43%
Poco	0	0%
Nada	0	0%

Fuente: Elaboración Propia

Figura 24: Resultado Encuesta – Pregunta 1



Fuente: Elaboración Propia

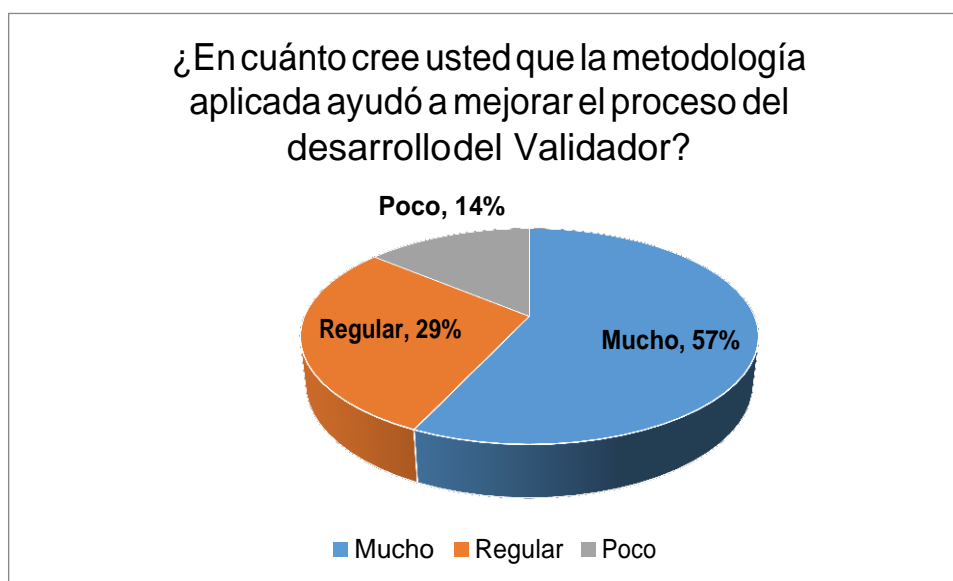
Para la realización del Validador y su adaptación a los estándares del ScotiaBank, es necesario conocer los documentos básicos propios del banco. Este resultado de la encuesta demuestra que se llevó un conocimiento adecuado para el desarrollo de la Herramienta. Esto debido a las reuniones diarias que hicieron posible la distribución de la información con respecto a documentos de Estándares y Buenas Prácticas de DataStage.

Tabla 17: Encuesta de Percepción – Pregunta 2

OBJETIVO ESPECIFICO 2		
PREGUNTA 2		
Fecha: 09/01/2017		
¿En cuánto cree usted que la metodología aplicada ayudó a mejorar el proceso del desarrollo del Validador?		
ENCUESTADOS		
Roles:	Cantidad	
1. Jefes de Centro de Información	2	
2. Equipo de Desarrollo	2	
3. Revisores	3	
CUADRO DE FRECUENCIAS		
Categoría	Frecuencia	Porcentaje
Mucho	4	57%
Regular	2	29%
Poco	1	14%

Fuente: Elaboración Propia

Figura 25: Resultado Encuesta – Pregunta 2



Fuente: Elaboración Propia

Se puede apreciar la aceptación que tiene la metodología, ya que para cualquier inconveniente, ya se agregará un requerimiento, se puede realizar sin problemas por tratarse de una metodología ágil.

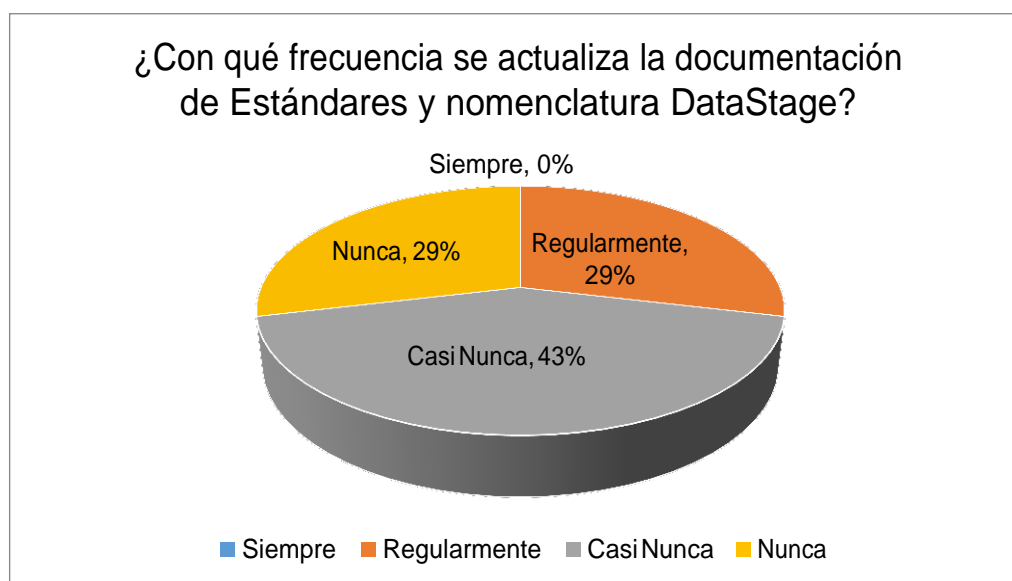
Hay casos en los que no se ha detectado caídas en las ejecuciones de Jobs, esto debido a impactos de solicitudes que han contenido Jobs aparentemente sin observaciones, ya que se rigieron a los estándares definidos en la documentación. Sin embargo, existen otras observaciones a considerar que no se encuentran en la documentación. Para estos casos, la aplicación de SCRUM como metodología de desarrollo se considera adecuada.

Tabla 18: Encuesta de Percepción – Pregunta 2A

OBJETIVO ESPECIFICO 2		
PREGUNTA 2A		
Fecha: 09/01/2017		
¿Con qué frecuencia se actualiza la documentación de Estándares y nomenclatura DataStage?		
ENCUESTADOS		
Roles:	Cantidad	
1. Jefes de Centro de Información	2	
2. Jefes Área de BI	2	
3. Revisores	3	
CUADRO DE FRECUENCIAS		
Categoría	Frecuencia	Porcentaje
Siempre	0	0%
Regularmente	2	29%
Casi Nunca	3	43%
Nunca	2	29%

Fuente: Elaboración Propia

Figura 26: Resultado Encuesta – Pregunta 2A



Fuente: Elaboración Propia

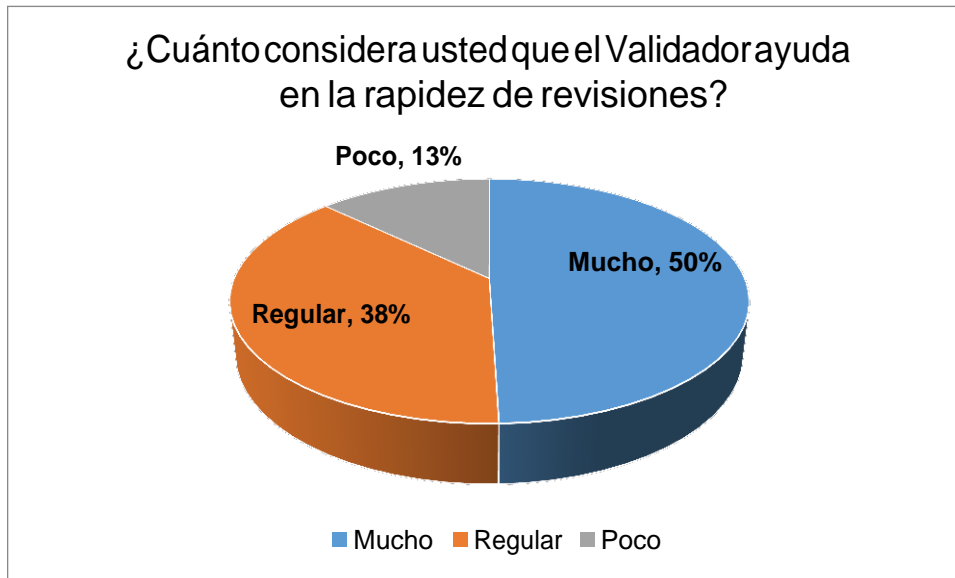
Tal como le describe el resultado de la encuesta, la documentación se actualiza cada cierto tiempo, pero no es frecuente; pero considerando la explicación anterior, se observará los objetos por alguna anomalía que presente, así éstos objetos cumplan con los estándares de la documentación.

Tabla 19: Encuesta de Percepción – Pregunta 3

OBJETIVO ESPECIFICO 3		
PREGUNTA 3		
Fecha: 09/01/2017		
¿Cuánto considera usted que el Validador ayuda en la rapidez de revisiones?		
ENCUESTADOS		
Roles:		Cantidad
1. Jefes de Centro de Información		2
2. Scrum Master		1
3. Equipo de Desarrollo		2
4. Revisores		3
CUADRO DE FRECUENCIAS		
Categoría	Frecuencia	Porcentaje
Mucho	4	50%
Regular	3	38%
Poco	1	13%

Fuente: Elaboración Propia

Figura 27: Resultado Encuesta – Pregunta 3



Fuente: Elaboración Propia

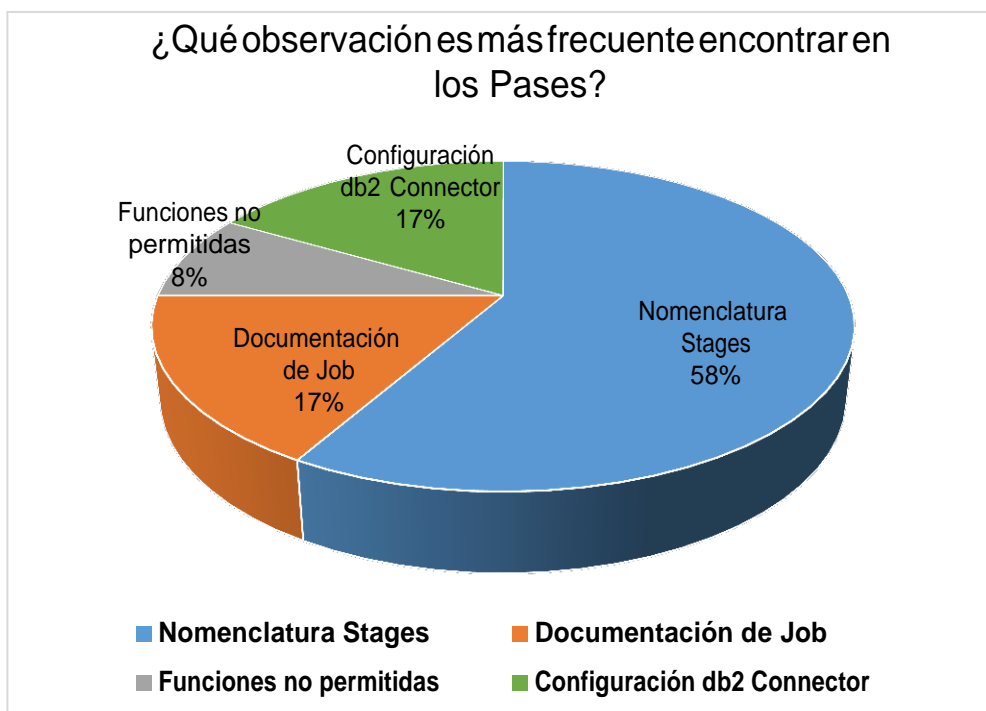
Se tienen los resultados y se llega a la conclusión que el Validador sirve de mucho apoyo en el tiempo que se toma revisando objetos DataStage. Esto aceptado al 100% por los revisores, ya que ellos son los que usarán propiamente el validador y conocen el flujo diario.

Tabla 20: Encuesta de Percepción – Pregunta 3A

OBJETIVO ESPECIFICO 3		
PREGUNTA 3A		
Fecha: 09/01/2017		
¿Qué observación es más frecuente encontrar en los Pases?		
ENCUESTADOS		
Roles:		Cantidad
1. Jefes de Centro de Información		2
2. Jefes Área de BI		2
3. Analistas		5
4. Revisores		3
CUADRO DE FRECUENCIAS		
Categoría	Frecuencia	Porcentaje
Nomenclatura de Stages	7	58%
Documentación de Job	2	17%
Funciones no permitidas	1	8%
Configuración db2 Connector	2	17%

Fuente: Elaboración Propia

Figura 28: Resultado Encuesta – Pregunta 3A



Fuente: Elaboración Propia

De este resultado, se tiene que la observación comúnmente encontrada en un Pase son errores de Nomenclatura y como se tiene en el CheckList inicialmente presentado, esto es motivo de Rechazo a la solicitud.

Teniendo estos resultados no cabe duda que el Validador aportará no sólo en la mejora de procesos de atención a las solicitudes, con respecto al tiempo, sino que permitirá la aplicación y mejora de las buenas prácticas de DataStage, al no permitir que un Pase, que afecte negativamente a algún proyecto, sea impactado.

CONCLUSIONES

El uso del validador y su desarrollo detecta con facilidad las razones por la que un pase es observado o rechazado conforme a los documentos definidos en Scotiabank.

La metodología Scrum en el desarrollo del validador permite modificarlo según cambien los requerimientos, es decir, conforme cambien los estándares establecidos en los documentos.

Se comprobó que el uso del validador disminuye la revisión de pases manuales, con un tiempo de ahorro de 49.47%, gracias a ello mejoraría el tiempo de respuesta de una revisión de solicitud con dsx. Así mismo, se demostró que el uso del validador permite a Scotiabank ahorrar aproximadamente 13000 soles al año, además de ello garantiza la limpieza de objetos en los impactos a producción.

El uso del validador en cada Revisión ha traído como consecuencia, una atención mucho más fluida. Los Pases o Incidencias pueden ser atendidos mucho más rápido, y no hay quejas por parte de los analistas. Se tienen impactos más limpios, libres de observaciones. Gracias al uso del validador, ahora se pueden encontrar la mayor cantidad de observaciones en un Pase o Incidencia.

RECOMENDACIONES

Se recomienda que al personal nuevo que utilizará esta herramienta, capacitarlo en el uso del Validador y de todas las funcionalidades.

El validador no contempla encontrar todos los errores en un objeto al 100%, sólo asegura completamente verificar la correcta nomenclatura de los Stages, que es una de las mayores dificultades, así que es recomendable que los puntos no tomados en cuenta por el validador, se sigan realizando manualmente.

Se sugiere seguir con la metodología SCRUM en el proceso de desarrollo del Validador, ya que los requerimientos para sus funciones pueden cambiar en cualquier momento, y no se debería ver afectado los avances del desarrollo.

Como recomendación, se tiene de uno de los desarrolladores del Validador, Alan Alvarado, sugirió que si el validador se implementara como una herramienta Web, podría trabajarse en lenguaje JAVA de manera más eficiente.

BIBLIOGRAFÍA

- Caralt, J. (2010). INTRODUCCIÓN AL BUSINESS INTELLIGENCE.
- Méndez , L. (2006). MÁS ALLÁ DEL BUSINESS INTELLIGENCE.
Barcelona: GESTIÓN 2000.
- Carrera, A. (2015). Implementación de una suite de inteligencia de negocios para el “Sistema de información para la gestión del patrimonio cultural- Abaco”.
- (CHR, 2007) Chuidiang Roxas, SCRUM, 2007, Web.
Obtenido de: <http://www.chuidiang.com/ood/metodologia/SCRUM.php>
- SCRUM Organization. (2013). Resources.
- Obtenido de:
<https://www.SCRUM.org/Portals/0/Documents/SCRUM%20Guides/2013/Scrum-Guide-ES.pdf>
- SCRUM Alliance. (2013). Why SCRUM?
Obtenido de SCRUM Alliance: <http://www.SCRUMalliance.org/why-SCRUM>

- SCRUM Organization. (2013).
Obtenido de:
<https://www.SCRUM.org/Portals/0/Documents/SCRUM%20Guides/2013/Scrum-Guide-ES.pdf>

- Palacio, J. (2006). El modelo SCRUM.
Obtenido de El modelo SCRUM: http://www.navegapolis.net/files/s/NST-010_01.pdf

- Thompson, I. (2008). PromoNegocios.net.
Obtenido de <http://www.promonegocios.net/mercadotecnia/que-es-informacion.html>

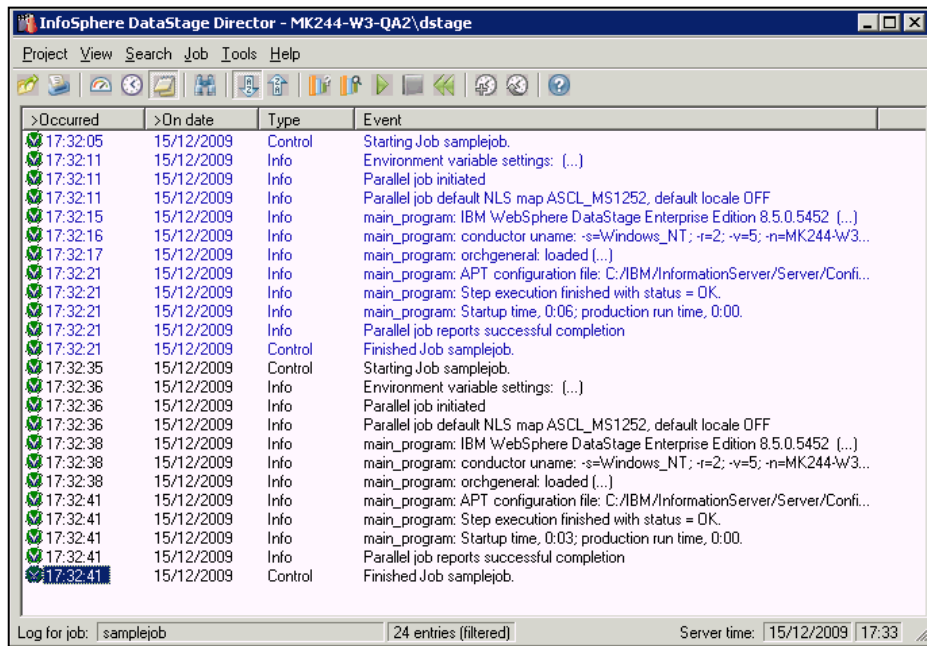
- Scrum (2016) Una guía para el conocimiento de Scrum
Obtenido de: <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016-spanish.pdf>

- Juan Palacio (2014) Scrum Manager – Gestión de proyectos Scrum Manager v2.5

- InfoSphere DataStage (2011)
Obtenido de: <https://www.ibm.com/support/knowledgecenter/en/SSZJPZ>

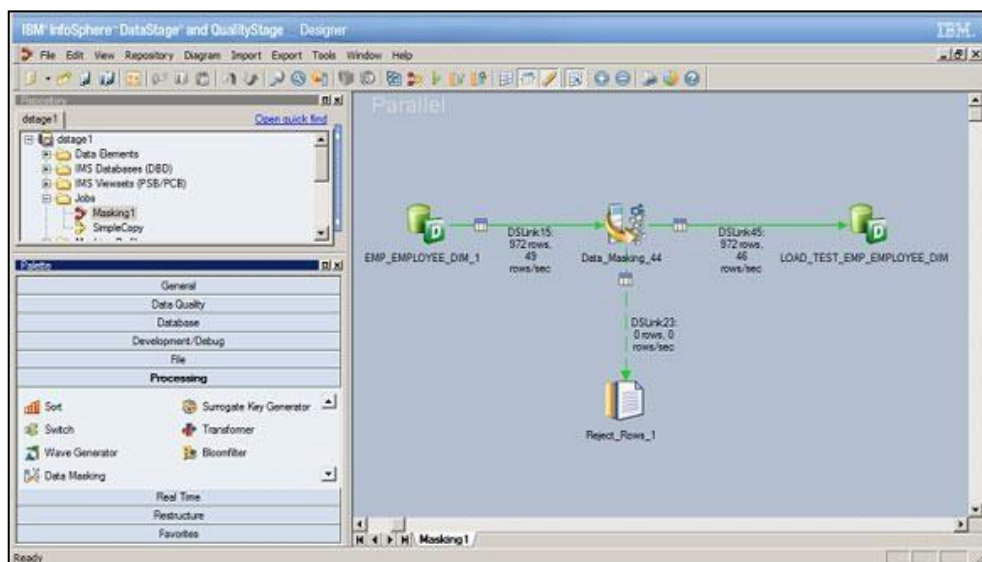
ANEXOS

A1. Ventana Cliente del Director – DataStage.



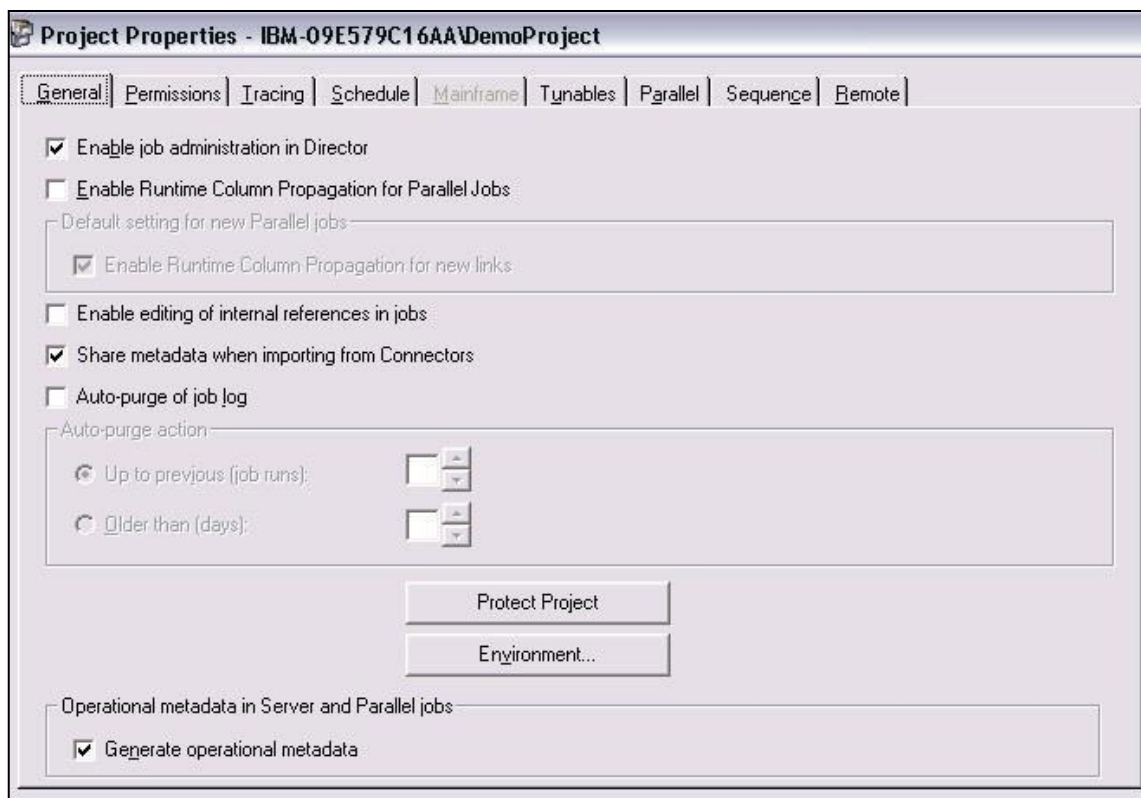
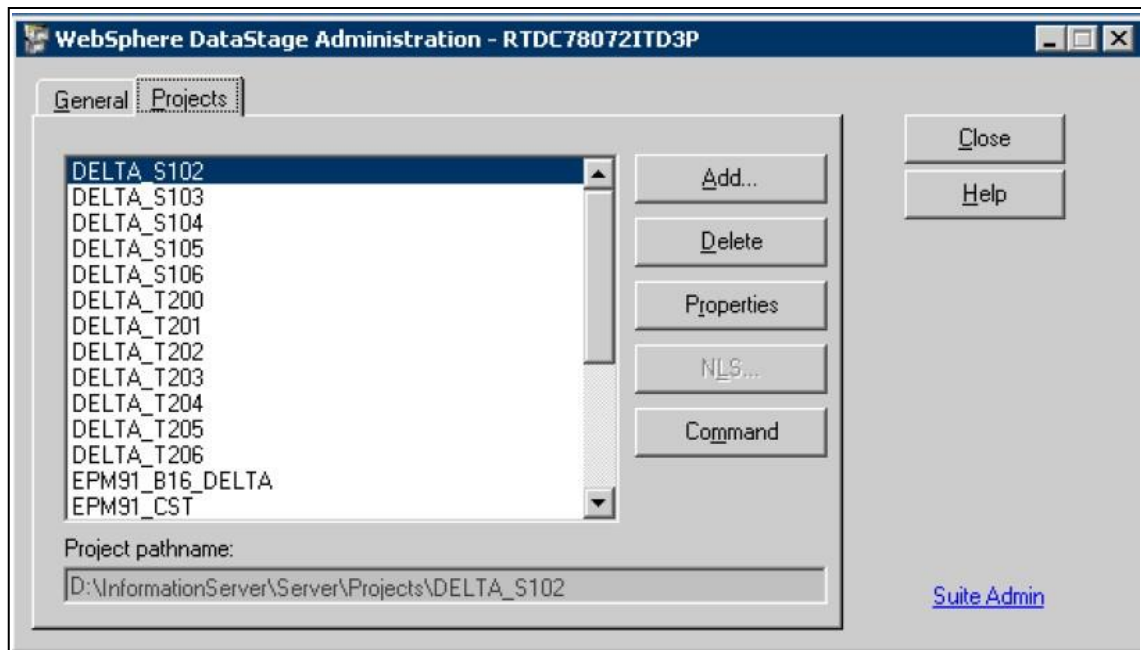
En el Cliente del Director, se puede apreciar el log de las ejecuciones de todos los Jobs existentes en el Proyecto.

A2. Ventana Cliente del Diseñador – DataStage.



En el Diseñador se crean los Jobs, aquí podemos utilizar todos los Stages conforme a la lógica que se use.

A3. Ventana Cliente del Administrador – DataStage.










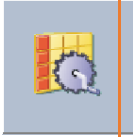






En el Administrador se encuentran el listado de todos los proyectos existentes.









Aquí también se configuran las variables de entorno y se otorgan accesos.




A4. Nomenclatura de Stages

Se tiene el listado de la nomenclatura considerada para cada Stage.

Stage	Gráfico	Mnemónico	Descripción
General			
Links		lnk_NOMBRE	Donde STAGE corresponde a los nombres definidos en el documento.
Shared container		shrd_NOMBRE	Stage que contiene un componente reutilizable compuesto de lógica.
Archivos			
Dataset		ds_NOMBRE_DATASET	Stage que manipula datasets
Sequential file		seqf_NOMBR E_ARCHIVO	Stage que manipula archivos secuenciales.
Base de datos			
DB2 conector		db2_ESQUEM A_TABLA	Stage de conexión a una base de datos db2
ODBC Connector		odbc_ESQUE MA_TABLA	Stage de conexión a diversas bases de datos (ejemplo: bd sql)
Oracle conector		ora_ESQUEM A_TABLA	Stage de conexión a Oracle
Desarrollo / Depuracion			

Column generator		col_NOMBRE	Stage que genera columnas de acuerdo a las condiciones configuradas
Peek		pek_NOMBRE	Stage que permite escribir la salida de un proceso directamente en el log.
Row generator		rog_NOMBRE	Stage que genera registros de acuerdo a las condiciones configuradas
Proceso			
Agregator		agg_NOMBRE	Stage que permite realizar operaciones de agregación, tales como, sumatorias, promedios, etc.
Bloom filter		bof_NOMBRE	Stage que realiza búsquedas de forma más eficiente.
Change Apply		cha_NOMBRE	Stage que permite comparar registros de dos orígenes (ejemplos nuevos, eliminados y actualizados).
Change capture		chg_NOMBRE	Stage que permite comparar registros de dos orígenes (ejemplos

			nuevos, eliminados y actualizados)
Copy		cpy_NOMBRE	Stage que realiza la copia de un input
Filter		fil_NOMBRE	Stage que permite realizar filtros de registros a partir de una condición dada.
FTP Enterprise		ftp_NOMBRE	Stage para conexión ftp entre servidores.
Funnel		fun_NOMBRE	Stage que realiza la consolidación de n inputs.
Join		jon_NOMBRE	Stage que permite realizar el producto cartesiano de dos orígenes de datos.
Lookup		lok_NOMBRE	Stage de búsqueda.
Merge		mrg_NOMBRE	Stage que permite realizar la unión de dos orígenes de datos.
Modify		mdf_NOMBRE	Stage que permite realizar mapeos y transformación no complejas como en el Transformer.

Remove duplicates		rmv_NOMBRE Stage que realiza la eliminación de registros a partir de una clave determinada.
Sort		srt_NOMBRE Stage que permite ordenar una entrada de datos.
Transformer		tra_NOMBRE Stage que permite mapeos y transformaciones de datos.