

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SAN JOAQUÍN - CHILE



“CARGA AUTOMÁTICA DE DATOS PARA
ANÁLISIS Y PREVENCIÓN DE FRAUDE: CASO
PRÁCTICO APLICADO A LA DETECCIÓN DE
FRAUDE EN COMERCIOS”

JUAN CARLOS ARRIAGADA NAVARRO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO DE EJECUCIÓN INFORMÁTICA

Profesor Guía: José Luis Martí Lara
Profesor Correferente: Mauricio Olivares Faúndez

Septiembre - 2023

DEDICATORIA

A mi madre, que en paz descansa, quien pese a su enfermedad, trabajó arduamente para poder dar a mis hermanos y a mi acceso a la educación.

A mi padre y hermanos, por su apoyo y confianza incondicional, y en especial a mi hermano Andrés, quien siempre ha sido y será un ejemplo y un modelo a seguir.

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a todas las personas que han sido parte fundamental de mi viaje académico y que han contribuido de manera significativa a la culminación de este importante logro.

En primer lugar, quiero extender un agradecimiento especial a mi hermano Andrés, por su apoyo constante e incondicional, y por siempre estar allí para recordarme la importancia de concluir este ciclo, de mantener la perseverancia y la determinación.

A mis amigos y amigas, en especial a aquellos que conocí durante mi formación universitaria, y con quienes aún mantengo fuertes lazos, que respecto a este hito, que parecía tan lejano, siempre me motivaron e impulsaron a retomarlo y darle buen término.

No puedo pasar por alto el agradecimiento a mi profesor guía, José Luis Martí, quien a pesar de mis prolongadas ausencias y demoras, siempre estuvo presente, dispuesto a brindarme orientación y consejo. Su compromiso con mi crecimiento académico y paciencia fueron invaluable para mí.

A todos los que, de una u otra manera, formaron parte de mi travesía universitaria, gracias.

RESUMEN

Resumen— La prevención del fraude en el sector financiero y bancario es una tarea en constante evolución, que requiere ser rediseñada y mejorada constantemente para cubrir posibles nuevas vulnerabilidades a ser explotadas por los defraudadores, por lo que para abordar efectivamente este problema, es clave contar con una amplia cantidad de información para analizar comportamientos y patrones fraudulentos. El presente trabajo se enfoca en la carga automática de datos de diferentes orígenes en una base centralizada, de fácil y rápido acceso para que el equipo de prevención de fraude pueda realizar los análisis pertinentes para mejorar la estrategia de prevención de fraude.

Palabras Clave— Código abierto, UTFSM, plataforma web, desarrollo web, Inteligencia de Negocio

ABSTRACT

Abstract— Fraud prevention in the financial and banking sector is an ever-evolving task that needs to be constantly redesigned and improved to cover potential new vulnerabilities that fraudsters may exploit. Therefore, to effectively address this issue, having a wealth of information to analyze fraudulent behaviors and patterns is crucial. This work focuses on the automatic loading of data from various sources into a centralized database, which is easily and quickly accessible for the fraud prevention team to conduct relevant analyses and enhance the fraud prevention strategy.

Keywords— Open Source, UTFSM, ETL, Business Intelligence, Dashboard

GLOSARIO

ABR: Autorización basada en riesgo.

AGPL: GNU Affero General Public License.

BI: *Business Intelligence*.

ETL: *Extract, transform, and load*

KPI: *Key Performance Indicator*.

NRT: *Near Real Time*.

POS: *Point of Sale*.

ROI: *Return On Investment*.

RT: *Real Time*.

Time Out: Transacciones cuya respuesta tardo mas del tiempo máximo esperado, por lo que la respuesta no es considerada al momento de autorizar o denegar la transacción.

UTFSM: Universidad Técnica Federico Santa María.

XML: *Extensive Markup Language*.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Fraude bancario	2
1.1.1 Tipos de fraude financiero externo	2
1.1.2 Análisis y prevención de fraude	3
1.1.3 Principales KPI asociados a la gestión del fraude bancario	4
1.2 Objetivos	5
1.2.1 Objetivo principal	5
1.2.2 Objetivos específicos	5
1.3 Alcance del proyecto	5
CAPÍTULO 2: ESTADO DEL ARTE	6
2.1 METODOLOGÍAS ÁGILES	6
2.1.1 Extreme Programming	6
2.1.2 SCRUM	8
2.1.3 Feature-Driven Development	15
2.2 Inteligencia de negocios	17
2.2.1 Definición de la inteligencia de negocios	17
2.2.2 Herramientas de la inteligencia de negocios	20
2.3 Visualización de datos	24
2.3.1 Atributos visuales y tipos de datos	24
CAPÍTULO 3: DISEÑO DE LA SOLUCIÓN	26
3.1 Modelo de solución	26
3.1.1 Entradas	26
3.1.2 Almacenaje y procesamiento	27
3.1.3 Presentación	27
3.2 Metodología de desarrollo	27
3.3 Herramientas seleccionadas	29
3.4 Diseño de reportes y tableros	29
3.5 Plan de trabajo	30
3.5.1 Transacciones financieras	30

3.5.2	Transacciones no financieras	31
3.5.3	Información demográfica de clientes	33
3.5.4	Información complementaria	34
3.5.5	Plantilla base para carga de archivos	35
3.5.6	Otros <i>jobs</i> reutilizables	36
3.5.7	Reporte resumen comercios	38
3.5.8	Monitoreo calidad transaccional	39
3.5.9	Monitoreo cargas de datos	39
CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN		41
4.1	Etapa 1: <i>Jobs</i> de Carga de datos	41
4.1.1	Carga transacciones financieras	41
4.1.2	Carga Información no financiera	45
4.1.3	Carga Información demográfica clientes	50
4.1.4	Carga Información complementaria	51
4.2	Etapa 2: Reportes y tableros de control	52
4.2.1	Reporte resumen comercios	52
4.2.2	Reporte calidad transaccional	52
4.2.3	Reporte monitoreo de cargas	53
5 CONCLUSIONES		56
REFERENCIAS BIBLIOGRÁFICAS		60

ÍNDICE DE FIGURAS

1	Diagrama ilustrativo de la metodología SCRUM	9
2	Ejemplo de Product Backlog	10
3	Ejemplo de Sprint Backlog	11
4	Ejemplo de Sprint Burndown Chart	12
5	Ejemplo de Release Backlog	13
6	Ejemplo de Release BurnDown Chart	14
7	Ciclo de vida de la metodología FDD	15
8	Entendimiento básico de la inteligencia de negocios	18
9	Proceso de la inteligencia de negocios	19
10	Los 3 niveles de la inteligencia de negocios	20
11	Cuadrante Mágico de Gartner para plataformas de analítica e inteligencia de negocios	21
12	Onda Forrester para plataformas de inteligencia de negocios	22
13	Atributos visuales típicos en visualización de datos	25
14	Modelo simplificado de solución	26
15	Plantilla base para carga de archivos, creada en Talend Open Studio.	35
16	<i>Job</i> reutilizable que mueve archivos desde una ruta a otra	37
17	<i>Job</i> reutilizable que escribe en una tabla el registro de ejecución	37
18	<i>Job</i> orquestador de ejecución de flujo carga ATM-POS RT/NRT	41
19	<i>Job</i> que carga transacciones ATM-POS RT/NRT a tabla temporal	42
20	<i>Job</i> que carga transacciones ATM-POS RT/NRT en la tabla definitiva	42
21	<i>Job</i> orquestador de ejecución para la carga de transferencias RT/NRT/ABR	43
22	<i>Job</i> que clasifica los archivos de transferencia según RT, NRT o ABR	43
23	<i>Job</i> que carga transferencias ABR	44
24	<i>Job</i> que carga transferencias RT	44
25	<i>Job</i> que carga transferencias NRT	44
26	<i>Job</i> orquestador que carga notificaciones VU	45
27	<i>Job</i> que carga notificaciones VU a tablas definitivas	46
28	<i>Job</i> que orquesta carga notificaciones ABR	46
29	<i>Job</i> que carga notificaciones ABR a tabla definitiva	47
30	<i>Job</i> que orquesta carga de las acciones automáticas	47
31	<i>Job</i> que carga las acciones automáticas por hora en las tablas definitivas	48
32	<i>Job</i> que orquesta carga de las transacciones marcadas como fraude	48
33	<i>Job</i> que carga el fraude marcado en una tabla temporal de transición	49
34	<i>Job</i> que realiza la carga en las tablas definitivas	49
35	<i>Job</i> que carga la información personal de los clientes	50
36	<i>Job</i> que carga la información resumen de los productos de los clientes	50
37	<i>job</i> que carga la información resumen de los productos de los clientes	51
38	<i>Job</i> que carga la información resumen de los productos de los clientes	51
39	Vista reporte resumen comercios	52
40	Vista calidad transaccional ATM POS	53
41	Vista transacciones financieras del reporte de monitoreo de cargas	54

42	Vista transacciones no financieras del reporte de monitoreo de cargas .	55
43	Vista acciones automáticas del reporte de monitoreo de cargas	55

ÍNDICE DE TABLAS

1	Compración de planes Microsoft Power BI	23
2	Comparación de planes Tableau Software	23
3	Resumen comparativo entre metodologías ágiles XP, SCRUM y FDD .	28
4	Historias de usuario para transacciones financieras	31
5	Pruebas unitarias y de aceptación para transacciones financieras	31
6	Historias de usuario para transacciones no financieras	32
7	Pruebas unitarias y de aceptación para transacciones no financieras . .	33
8	Historias de usuario para información demográfica de clientes	33
9	Pruebas unitarias y de aceptación para información demográfica de clientes	34
10	Historias de usuario para información complementaria	34
11	Pruebas unitarias y de aceptación para información complementaria . .	35
12	Historia de usuario para confección del reporte resumen de comercios .	38
13	Pruebas unitarias y de aceptación reporte resumen comercios	38
14	Historia de usuario para confección de reporte de calidad transaccional	39
15	Pruebas automáticas y de aceptación reporte calidad transaccional . . .	39
16	Historia de usuario para confección de reporte de cargas de datos . . .	40
17	Pruebas automáticas y de aceptación reporte de carga de datos	40

INTRODUCCIÓN

La prevención de fraude es un proceso altamente dinámico, en constante cambio, debido a la aparición de vulnerabilidades no detectadas en el acelerado proceso de transformación digital en el que esta inmerso el rubro financiero y bancario, como también debido a las diversas estrategias de ingeniería social mediante el cual los defraudadores engañan a los usuarios de productos bancarios (ej: *Phishing*), por lo que para ser más efectivos en la prevención del fraude, es crucial contar con la mayor cantidad de información para poder realizar análisis de conductas fraudulentas y definir la estrategia de prevención más adecuada para mantener controlado los niveles de fraude.

El presente trabajo está dividido en cuatro capítulos principales: El primer capítulo corresponde a la definición del problema, donde se explicará en que consiste el fraude bancario, los tipos de fraude existentes, qué es la gestión y prevención de fraude, cuáles son los principales indicadores asociados a la gestión del fraude bancario y se establecerán los objetivos y el alcance del presente proyecto. El segundo capítulo corresponde al estado del arte, donde se abordarán algunas metodologías de trabajo ágil, se profundizará acerca de qué es la inteligencia de negocios, revisando algunas opciones de herramientas de inteligencia de negocios disponibles en el mercado, y se estudiará la visualización de datos, sus ventajas y desventajas. El tercer capítulo corresponde al diseño de la solución, donde se presentará el modelo de solución a implementar para abordar la problemática planteada, se realizará la elección de la metodología ágil más adecuada, se elegirán las herramientas a utilizar, y se detallará el diseño y plan de trabajo para el desarrollo de la solución propuesta. El cuarto capítulo corresponde al desarrollo de la solución, donde se detallará el proceso de desarrollo y los resultados obtenidos, basado en el plan de trabajo presentado en el capítulo anterior.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

Para la prevención del fraude bancario, especialmente en lo que se refiere al fraude externo con tarjeta, es sumamente importante contar con la información integral del cliente en el uso de sus productos financieros, por lo que contar con acceso a los datos de forma oportuna, actualizada, rápida y fácil es indispensable para definir y mejorar las estrategias que permitan controlar el fraude. Dado que el fraude bancario es altamente dinámico, puesto que los defraudadores están constantemente poniendo a prueba las plataformas financieras y los medios de pago, los que a su vez, día a día están incorporando nuevas funcionalidades en las plataformas, aplicaciones y/o medios de pago, las que, en ocasiones, pueden presentar vulnerabilidades no detectadas oportunamente que pueden ser explotadas por los defraudadores.

1.1. Fraude bancario

El fraude bancario puede ser definido como “el uso indebido y no autorizado de los productos y sistemas financieros por parte de terceros, para obtener beneficio personal”. Es una de las principales preocupaciones de la industria, puesto que si no se gestiona adecuadamente, puede significar enormes pérdidas financieras y reputacionales para el negocio. En términos simples, se puede clasificar el fraude bancario en dos grandes categorías: El fraude bancario interno, que corresponde al fraude realizado por parte del personal de la entidad bancaria y/o terceros que presten servicios a la entidad financiera, y el fraude bancario externo, que corresponde a transacciones financieras realizadas por terceros, mediante el acceso no autorizado por el cliente a sus productos y/o cuentas.

1.1.1. Tipos de fraude financiero externo

Los tipos de fraude externo más comunes son:

- **Phishing:** Según VISA (2022), corresponde al “método que los ciberdelincuentes utilizan para engañar al cliente y conseguir que revele información personal, como contraseñas o datos de tarjetas de crédito y números de cuentas bancarias”. Los medios más típicos mediante los que se realiza *phishing* son las llamadas telefónicas, mensajes de texto, sitios web falsos y redes sociales.

- **Skimming:** Corresponde al método utilizado para obtener la información de tarjetas de crédito/débito al momento de realizar una transacción presencial para hacer uso fraudulento posterior, generalmente mediante la alteración del terminal de pago o cajero automático.
- **SIM Swapping:** Según la Subtel (2022), “el *SIM Swapping* consiste en la suplantación de identidad de un usuario de telefonía móvil, donde los estafadores realizan un trabajo de arqueología social”. Es una técnica de suplantación en alza dado que el número de teléfono se usa para validar la identidad en la mayoría de los sistemas bancarios, mediante envío de *SMS* con clave temporal de único uso (OTP por sus siglas en inglés).
- **Card Testing:** También llamado explotación de tarjetas o prueba de tarjetas, corresponde a la prueba de números de tarjetas de crédito/débito que fueron robadas, compradas o generadas aleatoriamente a partir de otras tarjetas válidas, las que son probadas en comercios web, para identificar cuales son válidas para luego materializar el fraude.

1.1.2. Análisis y prevención de fraude

El análisis y prevención del fraude consiste en el uso de técnicas de análisis de grandes volúmenes de datos con el fin de detectar patrones y cambios de conducta transaccionales de los clientes (velocidad de transacciones, denegaciones, errores de validación de claves, recurrencia en el rubro o comercio, etc.), y transacciones potencialmente fraudulentas, con el fin de definir estrategias adecuadas para su mitigación. La estrategia de prevención debe ser certera (detener la mayor cantidad de fraude), evitando generar mayor fricción a los clientes al momento de usar sus productos financieros. Dado que por lo general los ataques de fraude suelen afectar a alguna vulnerabilidad específica o a algún tipo de transacción en particular, resulta recomendable realizar el análisis clasificando el fraude. Una segmentación que resulta efectiva para detectar anomalías es clasificar las transacciones fraudulentas según su origen y forma, las que se detallan a continuación.

- **Origen:** Desde que lugar geográfico se realiza la transacción, el más típico es separarlo por “nacional” e “internacional”, aunque una categorización por país o agrupaciones de países pueden resultar útiles.
- **Forma:** Se refiere a la forma en que fue realizada la transacción, si esta fue “presencial” (ej: Compra en un comercio con la tarjeta física a través de algún terminal de punto de venta POS, por sus siglas en inglés) o “ausente” (ej: Compras por internet mediante alguna pasarela de pago web).

Se recomienda considerar complementar las clasificaciones combinándolas entre si y con otras segmentaciones, como pueden ser el rubro, país, comercio, como también crear

otras basadas en los perfiles de uso de los clientes, dado que pueden aportar mayores antecedentes acerca de las conductas de fraude y así enfocar la estrategia de prevención a dicho segmento, y no impactar otras conductas que no están relacionadas con el fraude.

1.1.3. Principales KPI asociados a la gestión del fraude bancario

Teniendo en cuenta que el objetivo principal de la gestión del fraude es minimizar la pérdida por transacciones fraudulentas, generando la menor fricción posible al cliente, maximizando las ganancias, se espera que los indicadores de la gestión del fraude reflejen que el monto de fraude aprobado se mantiene dentro de la tolerancia del negocio. Las reglas del modelo de prevención de fraude deben ser certeras, con la menor fricción al cliente en el uso de sus productos. Por lo anterior, los principales indicadores que permiten evaluar el estado de la gestión del fraude se detallan a continuación.

- **Puntos base:** Es un indicador del monto o cantidad de transacciones fraudulentas aprobadas, contrastado contra el monto o cantidad de transacciones de la venta total aprobada, para un mismo periodo de tiempo. Su fórmula es:

$$\frac{[Monto\ Fraude]}{[monto\ venta\ total]} * 10,000$$

El valor esperado de este indicador depende del apetito del negocio, y mientras más cercano a 0, mejor. Se puede considerar bueno cuando el valor de este indicador esta entre 0 y 5.

- **Falso positivo:** Corresponde a la medida de efectividad en la detección del fraude de cada regla implementada que forma parte de la estrategia de prevención de fraude. Su fórmula es:

$$\frac{[Transacciones\ alertadas]}{[Transacciones\ alertadas\ con\ fraude\ confirmado]}$$

Se espera que el indicador sea el menor posible, siendo 1 el resultado ideal, que significa que todas las transacciones alertadas resultaron ser fraudulentas. En la práctica, una buena regla debería tener un falso positivo de entre 1 y 5 (máximo que cada 5 transacciones alertadas, 1 es fraude).

- **Tasa de rechazos de fraude:** Corresponde a las denegaciones surgidas a partir de las reglas implementadas en el motor de fraude, contra el total de transacciones aprobadas. Es un indicador relevante puesto que permite identificar si el motor de prevención de fraudes está denegando muchas transacciones. Su fórmula es:

$$\frac{[Transacciones\ denegadas\ sospecha\ de\ fraude]}{[Transacciones\ totales\ aprobadas]+[Transacciones\ denegadas\ sospecha\ de\ fraude]} * 100\%$$

Se espera que el valor de este indicador sea inferior al 3%.

1.2. Objetivos

1.2.1. Objetivo principal

Mejorar la estrategia de prevención y gestión del fraude externo no presencial de una importante entidad bancaria, mediante la centralización de la información financiera, no financiera y demográfica de los clientes en una única fuente de datos, de manera adecuada y oportuna según la necesidad del negocio.

1.2.2. Objetivos específicos

- Controlar que las cargas de actualización se efectúen correctamente y lleven a una base de datos actualizada.
- Mejorar el indicador de puntos base para comercios fraudulentos tras el uso periódico de los reportes para la detección de nuevos comercios fraudulentos.
- Contar con un procedimiento claro y conciso para la integración de nuevas fuentes de datos basado en el repositorio definido, para facilitar las cargas futuras de información

1.3. Alcance del proyecto

El presente proyecto contempla el diseño e implementación de procesos ETL, para la carga de la información transaccional financiera y no financiera, acciones automáticas del motor de reglas e información demográfica de los clientes para apoyar el análisis y la gestión del fraude, y la implementación posterior de paneles que estarán enfocados al apoyo de la gestión de potenciales comercios nuevos fraudulentos, el monitoreo de las cargas de datos y de la calidad transaccional.

CAPÍTULO 2

ESTADO DEL ARTE

Para resolver exitosamente el problema expuesto, se estudiarán en detalle algunas metodologías ágiles, la inteligencia de negocios y la visualización de datos.

2.1. METODOLOGÍAS ÁGILES

La rapidez con la que el fraude va evolucionando, adaptándose a las estrategias de fraude y buscando constantemente nuevas vulnerabilidades a ser explotadas, hacen que sea imperante abordar el diseño y desarrollo de la solución al problema planteado mediante la aplicación de metodologías ágiles, por lo que el estudio y selección de este tipo de metodologías es fundamental para el éxito de la implementación, ya que permiten idear soluciones flexibles, adaptativas y tolerantes a requerimientos cambiantes. Por lo anterior, es necesario evaluar algunas alternativas de metodologías ágiles, y seleccionar la que mejor se adapte a la propuesta de solución y los recursos disponibles. Las metodologías a revisar son:

- Extreme Programming
- Scrum
- Feature-Driven Development

2.1.1. Extreme Programming

Según Maurer y Martel (2002), la metodología ágil Extreme Programming -en adelante XP- funciona mejor con pequeños equipos de 5 a 15 desarrolladores, dejando de lado la captura de requerimientos en papel y el diseño de la documentación, centrándose en producir código ejecutable y la automatización de las pruebas (test drivers).

Sus principales características son:

- Ideada para equipos pequeños (entre 5 a 15 desarrolladores).
- Enfocada en el diseño simple y la refactorización.
- Reemplaza la documentación basada en papel por la comunicación cara a cara.
- Aplica pruebas de regresión fuerte.

- Cuenta con la inspección continua del código.

Para llevar a cabo XP, es necesario adoptar las siguientes prácticas:

- **Cliente en sitio:** Se requiere contar con un representante del cliente en el sitio de trabajo, para que el equipo pueda aclarar requerimientos y definir prioridades rápidamente, documentando los requerimientos a través de historias de usuario (descripciones textuales de casos de uso).
- **Liberaciones pequeñas:** Se debe mantener ciclos cortos de liberación, donde cada versión genere una pieza de software útil con valor comercial para el cliente.
- **Metáfora:** Representa “lo que estamos tratando de hacer”. Por lo general se representa como una sola historia de usuario que proporciona todos los conceptos básicos del sistema. Sirve como arquitectura de software de alto nivel.
- **Testing:** El cliente define las pruebas funcionales (aceptación) que los desarrolladores implementan. Los desarrolladores escriben las pruebas unitarias antes de escribir el código actual.
- **Diseño simple:** El diseño se centra en resolver problemas actuales en lugar de diseñar para futuros cambios. El desarrollo no debe tener redundancias de código, debe tener la menor cantidad de clases y métodos, y ser fácil de entender.
- **Refactorización:** El código fuente debe seguir siendo simple y fácil de entender, después de cada cambio. Consiste en cambiar la estructura del código para mejorar su comprensibilidad y mantenimiento, sin cambiar su funcionalidad.
- **Programación de a pares:** Todo el código está escrito por dos personas que trabajan en una máquina. Una persona controla el mouse y teclado, y valida si la lógica implementada funcionará y si es posible simplificarla. La otra persona piensa en forma más estratégica y decide si esta es la mejor manera de implementar la funcionalidad. Ambas personas cambian de rol frecuentemente durante el desarrollo.
- **El juego de planificación:** Consiste en identificar quien tiene el requisito de conocimiento para decidir los siguientes puntos:
 - a) Alcance y prioridad: Consiste en identificar las características requeridas, cuales son las más importantes, prioritarias y cuáles pueden postergarse.
 - b) Fecha de lanzamiento: Identificar cuando debe estar disponible la próxima versión.
 - c) Estimaciones: El equipo de desarrollo debe identificar cuanto esfuerzo implica implementar una nueva función o corregir un error, y el trabajo que se puede asignar a la próxima versión.

- d) Consecuencias: Se debe identificar el impacto de las decisiones comerciales al proceso y al esfuerzo de desarrollo.
- e) Procesos: Se debe definir como trabajará y se organizará el equipo.

El objetivo del juego de planificación es equilibrar los intereses del cliente con la experiencia del equipo de desarrollo. El equipo estima el esfuerzo de la tarea, mientras que el cliente elige las tareas para la próxima iteración.

- **Desarrollo sostenible:** Nadie puede trabajar 45 horas semanales consecutivamente, sin afectar la calidad del producto. La administración o los clientes pueden presionar para que el equipo trabaje largas horas extra, pero el resultado suele ser una disminución de la calidad, desarrolladores agotados y una alta tasa de rotación.
- **Propiedad colectiva:** Todos los miembros del equipo poseen el código base. Cualquier persona que pueda agregar valor a cualquier porción del código en cualquier momento debe hacerlo.
- **Estándar de codificación:** Debido a que los desarrolladores programan diferentes partes del sistema con varios miembros del equipo, los estándares de codificación son obligatorios. El estándar debe ser fácil de seguir y adoptar de forma voluntaria.
- **Integración continua:** Los desarrolladores deben integrar el código tan seguido como sea posible, y al menos una vez al día. Esto garantiza que siempre haya disponible una versión del sistema ejecutable que contenga todas las características nuevas y pueda servir como la línea de base para todo el trabajo.
- **Contras:** Si el equipo de desarrollo crece, aumenta el tiempo invertido en intercambio de conocimiento y capacitación de nuevas personas, lo que hace que XP no sea adecuado en equipos de tamaño variable y con alta rotación.

2.1.2. SCRUM

Según Sutherland y Schwaber (2012), SCRUM es un marco iterativo e incremental para el desarrollo de productos y proyectos, que estructura el desarrollo en ciclos de trabajo llamados Sprints . Al comienzo de cada Sprint, el equipo selecciona requisitos del cliente (en adelante, items) de una lista priorizada, y se compromete a completar los items seleccionados para el final del Sprint, los que no cambian durante el Sprint. Todos los días, el equipo se reúne brevemente para revisar el progreso y definir los próximos pasos para completar el trabajo restante. Al final del Sprint, el equipo revisa el estado de este con las partes interesadas (*Stakeholders*), obteniendo comentarios y observaciones que pueden ser incorporados en el próximo Sprint. La figura 1 ilustra los elementos clave de SCRUM.

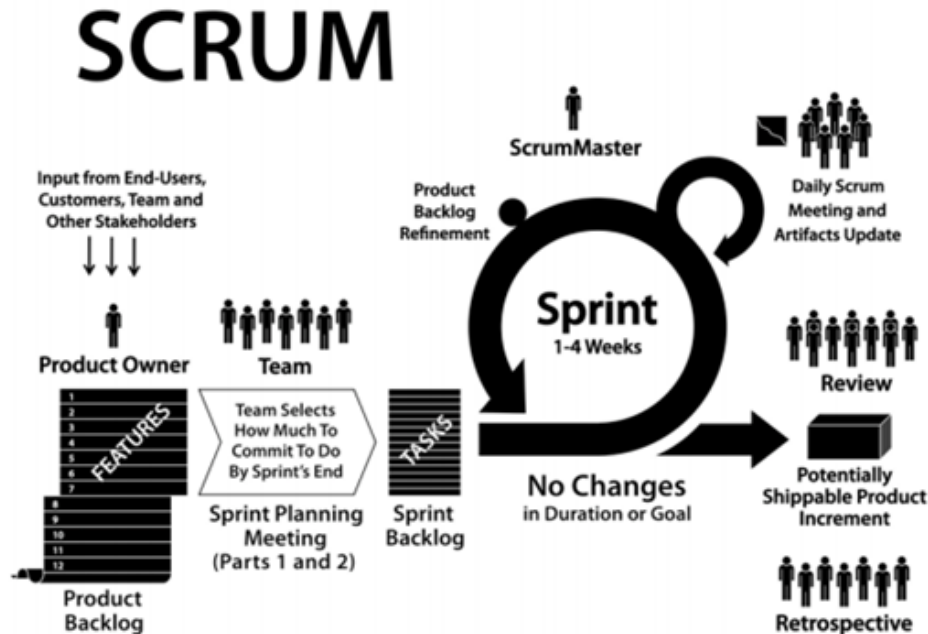


Figura 1: Diagrama ilustrativo de la metodología SCRUM
Fuente: Sutherland y Schwaber (2012).

Los roles en SCRUM son el ProductOwner, el Team y el ScrumMaster, los que se detallan a continuación.

- **ProductOwner:** Es el responsable de maximizar el retorno de la inversión (ROI por sus siglas en ingles), mediante la identificación de las características del producto y su priorización. Es quien define el orden de prioridad de las características para los Sprint.
- **Team:** Es el equipo que crea el producto que el ProductOwner solicita. Según Sutherland y Schwaber (2012), debe ser un equipo “Cross-Functional”¹, autogestionado y poseer un alto grado de autonomía y responsabilidad. En SCRUM, el equipo decide a qué comprometerse y como cumplirá dicho compromiso. El tamaño del equipo en SCRUM es de siete personas (más o menos dos), y podría incluir personas con habilidades en análisis, desarrollo, pruebas, diseño de interfaz, diseño de bases de datos, arquitectura, documentación, entre otros.
- **ScrumMaster:** Sutherland y Schwaber (2012) señala que el ScrumMaster es “quien ayuda al equipo y al ProductOwner a aprender y aplicar SCRUM para lograr un valor comercial”. El ScrumMaster debe hacer todo lo que está a su alcance

¹Cuenta con toda la experiencia necesaria para entregar el producto potencialmente distribuible en cada Sprint

para ayudar al equipo y al ProductOwner a tener éxito. El rol del ScrumMaster es asegurar que todos comprendan y sigan las prácticas de SCRUM. Es importante que el ScrumMaster y el ProductOwner **no sean la misma persona**, pues sirve como control por si el ProductOwner intenta hacer cosas que se salen de las practicas de SCRUM.

El primer paso en SCRUM es que el ProductOwner entregue su visión del producto, lo que se transforma en la lista de características priorizadas llamada **Product Backlog**.

Nuevas estimaciones de esfuerzo Restante a partir del Sprint...										
Item	Detalles (wiki URL)	Prioridad	Estimación de valor	Estimación inicial de esfuerzo	1	2	3	4	5	6
Como comprador, quiero colocar un libro en un carrito de compras (consulte los bocetos de la interfaz de usuario en la página wiki).	...	1	7	5						
Como comprador, quiero eliminar un libro de un carrito de compras.	...	2	6	2						
Mejorar el rendimiento del procesamiento de transacciones (consulte las métricas de rendimiento objetivo en wiki).	...	3	6	13						
Investigar soluciones para acelerar la validación de tarjetas de crédito (consulte las métricas de rendimiento objetivo en la wiki).	...	4	6	20						
Actualizar todos los servidores a Apache 2.2.3.	...	5	5	13						
Diagnosticar y corregir los errores de los scripts de procesamiento de pedidos (bugzilla ID 14823).	...	6	2	3						
Como comprador, quiero crear y guardar una lista de deseos.	...	7	7	40						
Como comprador, quiero agregar o eliminar artículos en mi lista de deseos.	...	8	4	20						

Figura 2: Ejemplo de Product Backlog
Fuente: Sutherland y Schwaber (2012).

Como se aprecia en la figura 2, el Product Backlog consiste en una variedad de items, principalmente nuevas características del cliente, y pueden presentarse de cualquier manera que sea clara, siendo generalmente definidos como casos de uso o historias de usuario. El equipo proporciona al ProductOwner estimaciones del esfuerzo requerido para cada item en el Product Backlog, y el ProductOwner le asigna a cada item una estimación del valor comercial, con el fin de que este último pueda priorizar el Release Backlog, maximizando el ROI y/o reduciendo algunos riesgos importantes. El Release Backlog corresponde a los items que serán abordados y liberados en esta versión del producto.

Sprint Planning

Al comienzo de cada Sprint se realiza la Sprint Planning, que consiste en una reunión que se divide en dos partes. En la primera parte, el ProductOwner y el equipo revisan los items del Product Backlog de alta prioridad que el ProductOwner está interesado en implementar en el Sprint actual, y revisan la “Definición de hecho” de todos los items. La segunda parte se centra en la planificación detallada de tareas, sobre cómo implementar los items que el equipo decide asumir. El equipo selecciona los items del Product Backlog que se comprometen a completar al final del Sprint, respetando

el orden de prioridad. Comenzando por el primer elemento en el Product Backlog y trabajando juntos, el equipo desglosa cada item en tareas individuales, que se registran en un documento llamado Sprint Backlog, tal como se ilustra en la Figura 3.

				Nuevas estimaciones de esfuerzo					
				Restante al día...					
Item Product Backlog	Sprint Task	Voluntario	Estimación inicial de esfuerzo	1	2	3	4	5	6
Como comprador, quiero colocar un libro en un carrito de compras	Modificar base de datos		5						
	Crear pagina web (UI)		8						
	Crear pagina web (Logica Javascript)		13						
	Escribir pruebas de aceptacion automaticas		13						
	...		3						
Mejorar el rendimiento del procesamiento de transacciones	Unir codigo DCP y completar pruebas layer-level		5						
	...		8						

Figura 3: Ejemplo de Sprint Backlog
Fuente: Sutherland y Schwaber (2012).

Sutherland y Schwaber (2012) también menciona que "si aparece una circunstancia externa que cambia significativamente las prioridades, y significa que el equipo estaría perdiendo su tiempo si continuara trabajando, el ProductOwner o el equipo pueden finalizar el Sprint", si esto ocurre, el equipo se detiene e inicia una nueva reunión de Sprint Planning para iniciar un nuevo Sprint.

Daily Scrum

Corresponde a una reunión corta de no más de 15 minutos, y se realiza todos los días hábiles a la hora acordada, en la cual cada miembro del Equipo informa a los otros miembros del equipo las siguientes 3 preguntas: (1) ¿que pudieron hacer desde la última reunión?; (2) ¿Que planean terminar para la próxima reunión?; y (3) ¿Se identifico algún bloqueo o impedimento para el desarrollo de algún item?.

Actualizando el Sprint Backlog y Sprint Burndown Chart

Todos los días, los miembros del equipo actualizan su estimación de la cantidad de tiempo restante para completar su tarea actual en el Spring Backlog. Después de esta actualización, alguien suma las horas restantes para el equipo en su conjunto y lo traza en el Sprint Burndown Chart (Figura 4). Este gráfico muestra, cada día, una nueva estimación de cuánto trabajo queda hasta que finalicen las tareas del equipo, y sirve para ver rápidamente si hay desviaciones respecto a la planificación inicial.

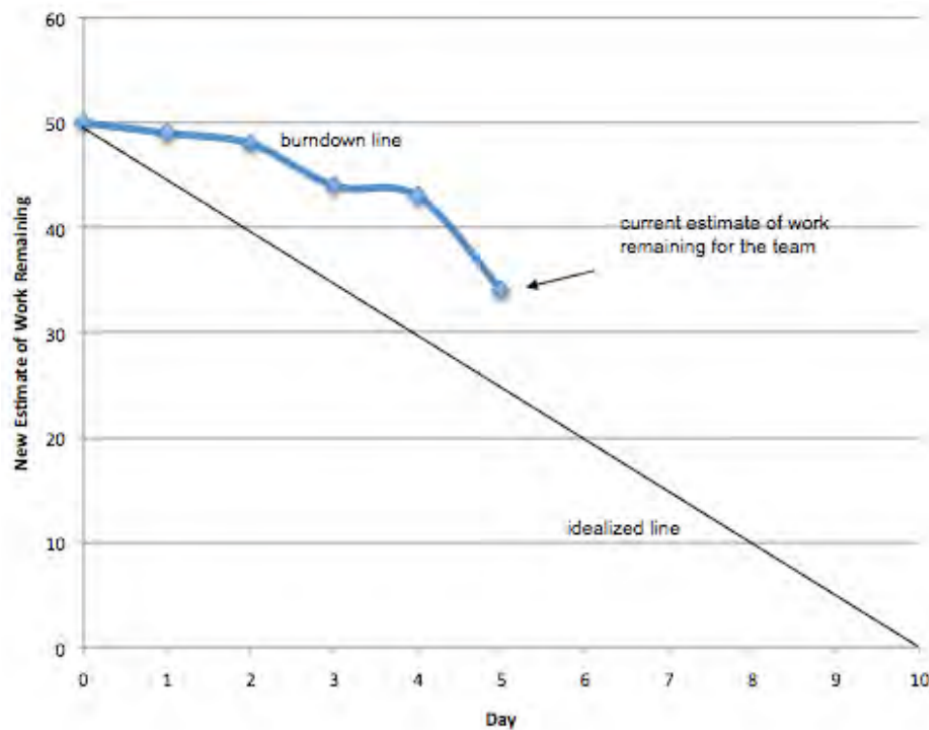


Figura 4: Ejemplo de Sprint Burndown Chart
Fuente: Sutherland y Schwaber (2012).

Refinando el Product Backlog

El equipo debe dedicar cinco o diez por ciento de cada Sprint a refinar el Product Backlog. Esto incluye análisis de requisitos detallados, división de items grandes en items más pequeños, estimación de items nuevos y reestimación de items existentes. Esta actividad es para items futuros, no para el Sprint actual.

Finalizando el Sprint

El Sprint finaliza en la fecha asignada, aun si el equipo no ha completado el trabajo al que se comprometió. En ocasiones, el equipo finaliza antes de tiempo, en cuyo caso, puede pedirle al ProductOwner más items del Product Backlog para trabajar.

Sprint Review

Una vez finalizado el Sprint se realiza la Sprint Review, que consiste en la revisión del Sprint como una actividad de inspección y adaptación para el producto, que incluye una demostración de lo que el equipo construyó durante el Sprint, y participan el ProductOwner, los miembros del equipo y ScrumMaster, además de clientes y otras partes interesadas.

Sprint Retrospective

Luego de la Sprint Review tiene lugar la Sprint Retrospective, que consiste en inspeccionar y adaptar el proceso. Es una oportunidad para que el Equipo discuta lo que funciona y lo que no funciona, y acuerde los cambios para intentar. El equipo y Scrum-Master asistirán, el ProductOwner no está obligado a hacerlo.

Actualizando el Release Backlog & Burndown Chart

El ProductOwner debe garantizar que los cambios producto del termino del Sprint se reflejen en el Release Backlog (Figura 5).

Nuevas estimaciones de esfuerzo Restante a partir del Sprint...										
Item	Detalles (wiki URL)	Prioridad	Estimación de valor	Estimación inicial de esfuerzo	1	2	3	4	5	6
Como comprador, quiero colocar un libro en un carrito de compras (consulte los bocetos de la interfaz de usuario en la página wiki).	...	1	7	5	0	0	0			
Como comprador, quiero eliminar un libro de un carrito de compras.	...	2	6	2	0	0	0			
Mejorar el rendimiento del procesamiento de transacciones (consulte las métricas de rendimiento objetivo en wiki).	...	3	6	13	13	0	0			
Investigar soluciones para acelerar la validación de tarjetas de crédito (consulte las métricas de rendimiento objetivo en la wiki).	...	4	6	20	20	20	0			
Actualizar todos los servidores a Apache 2.2.3.	...	5	5	13	13	13	13			
Diagnosticar y corregir los errores de los scripts de procesamiento de pedidos (bugzilla ID 14823).	...	6	2	3	3	3	3			
Como comprador, quiero crear y guardar una lista de deseos.	...	7	7	40	40	40	40			
Como comprador, quiero agregar o eliminar artículos en mi lista de deseos.	...	8	4	20	20	20	20			
Total				537	580	570	500			

Figura 5: Ejemplo de Release Backlog
Fuente: Sutherland y Schwaber (2012).

En esta etapa también se incluye un Release BurnDown Chart (Figura 6), que muestra el progreso hacia la fecha de lanzamiento. Es análogo Sprint Burndown Chart, pero está en el nivel más alto de items en lugar de tareas específicas.

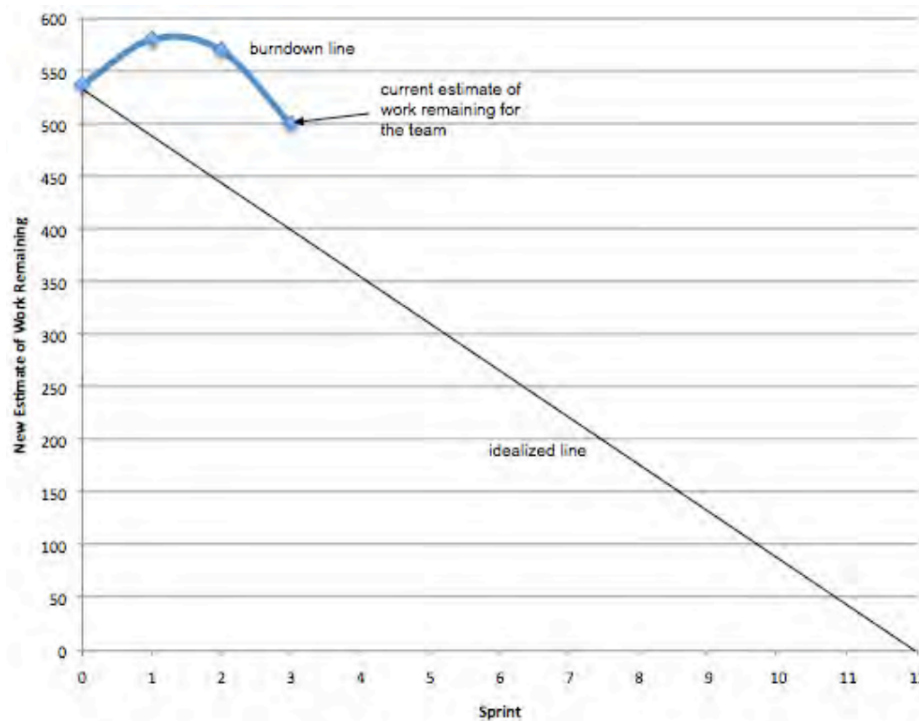


Figura 6: Ejemplo de Release BurnDown Chart

Fuente: Sutherland y Schwaber (2012).

Después de la Sprint Review, el ProductOwner puede actualizar el Product Backlog con cualquier información nueva. En este punto, el ProductOwner y el equipo están listos para comenzar otro ciclo de Sprint.

Release Sprint

En SCRUM todo está completamente terminado al final de cada Sprint, esto significa que cada incremento es una porción completa del producto final y brinda total transparencia al ProductOwner y a las partes interesadas.

Release Planning & Refinamiento del Product Backlog inicial

Durante un taller inicial de refinamiento del Product Backlog y durante el refinamiento continuo de este, en cada Sprint, el equipo y el ProductOwner realizarán la release Planning, en la cual irán refinando las estimaciones, prioridades y el contenido a medida que aprenden.

2.1.3. Feature-Driven Development

Según Anwer *et al.* (2017), el modelo ágil Feature-Driven Development - en adelante FDD -, usa iteraciones cortas para desarrollar software funcional, enfocado en la calidad durante todas las fases de desarrollo. Está centrado en las características, las que son “cualquier función valiosa que el usuario desee en el software”. Sus principales fases son las de diseño y construcción, y su objetivo es el desarrollo frecuente de resultados tangibles con información continua del progreso sobre el estado del proyecto.

Ciclo de vida de FDD

Consiste en cinco procesos secuenciales que se realizan iterativamente para construir software en incrementos (ver Figura 7).

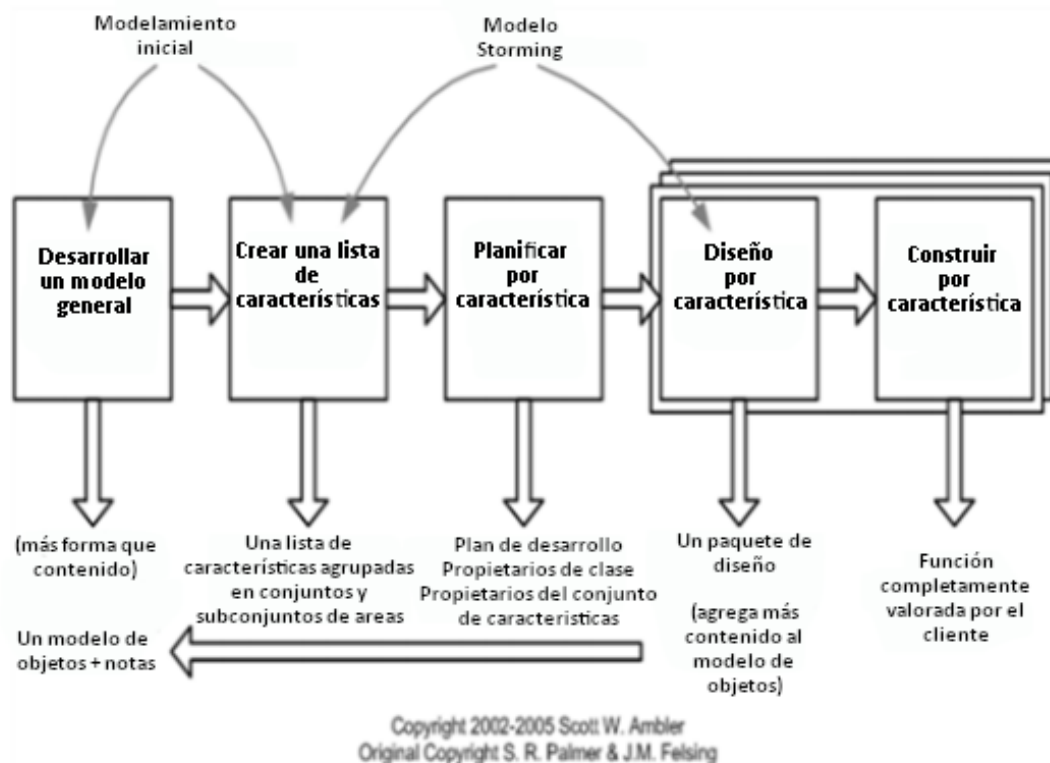


Figura 7: Ciclo de vida de la metodología FDD

Fuente: International journal of multidisciplinary sciences and engineering, Vol. 8.

- **Desarrollar un modelo general:** Consiste en definir el contexto y el alcance del proyecto a desarrollar, mediante una reunión paso a paso de alto nivel, seguida de un recorrido detallado de cada área de dominio.

- **Crear una lista de características:** Al ser un modelo impulsado por características, el proceso de desarrollo de FDD se centra en las características. Una característica es una función valiosa que tiene algún valor comercial en el software. Después de crear el modelo de objetos, es más fácil para el equipo definir una lista completa de características para desarrollar. Las características en la lista deben implementarse en un máximo de dos semanas. Cualquier característica que requiera más que esto se puede dividir en tareas más pequeñas. La lista de características finalmente es aprobada por el cliente.
- **Planificar por característica:** En esta fase, las prioridades se asignan a las características al tener en cuenta las características de las dependencias, los riesgos involucrados, la complejidad y la carga de trabajo del equipo. Luego, el jefe de programadores asigna cada clase de entidad a un desarrollador específico llamado propietario de la clase.
- **Diseño por característica:** Esta es una actividad iterativa que puede durar de unos días a dos semanas como máximo. En esta fase los paquetes de diseño se producen para cada clase por el jefe de programadores y los propietarios de la clase. Los diagramas de secuencia son desarrollados por el jefe de programadores y el propietario de la clase en conjunto; sin embargo, los diagramas de clase y de objetos son desarrollados por los respectivos propietarios de clase. Finalmente, los paquetes de diseño se revisan e inspeccionan para su aprobación.
- **Función de compilación:** Esta es una fase iterativa en la que los equipos de funciones múltiples realmente implementan las clases de diseño y los módulos. Después de la codificación, se realiza la inspección del código, la prueba unitaria y la prueba de integración. Las clases se crean en la secuencia que se definen en la fase de planificación por característica. Después de completar una iteración con éxito, las características desarrolladas se incluyen en la compilación principal y otra iteración con nuevas características se inicia nuevamente. Cada característica tiene que lograr seis hitos que son la inspección de dominio, el diseño y la inspección de diseño desde la fase de diseño y el código, la inspección del código y la promoción hasta la construcción principal, desde la fase de construcción. Cuando se completan todos los hitos, el gráfico de seguimiento por característica y el gráfico de agotamiento se actualizan para reflejar el progreso. Todas estas actividades deben completarse en un máximo de dos semanas.

Roles de FDD

En FDD hay tres tipos de roles: clave, de apoyo y adicionales. A continuación se detallan los roles clave de FDD.

- **Gerente de proyecto:** Es el líder que es responsable de proporcionar orientación administrativa durante todo el proyecto. Dirige el personal y proporciona el

mejor entorno de trabajo disponible protegiéndolos de interferencias externas. El gerente del proyecto también decide sobre el alcance, el cronograma del proyecto desarrollado. Informa el progreso del proyecto y maneja los problemas financieros y presupuestarios.

- **Arquitecto jefe:** Es responsable del diseño general del sistema. Su decisión se considera final sobre cuestiones relacionadas con el diseño. También guía y proporciona la sesión de entrenamiento necesaria para el equipo.
- **Gerente de desarrollo:** Las actividades diarias de desarrollo son supervisadas por el gerente de desarrollo. Debe tener una buena habilidad técnica para resolver los problemas que no maneja el programador jefe. También es responsable de resolver el conflicto de recursos entre los miembros del equipo.
- **Programador jefe:** Es quien dirige las actividades generales de desarrollo. Dirige los equipos pequeños durante el análisis, el diseño y la codificación de diferentes conjuntos de características. Debe tener una buena experiencia en desarrollo para dirigir el equipo. El programador jefe selecciona las funciones establecidas y el propietario de la clase respectivo para cada iteración y resuelve cualquier problema al implementar estas clases.
- **Propietario de clase:** estos son responsables de diseñar, codificar y probar conjuntos de características asignados por el programador jefe. Trabajan bajo la supervisión del programador jefe.
- **Experto de dominio:** Son las personas que tienen una comprensión clara de la empresa para la que se está desarrollando el software. Estos pueden ser clientes, patrocinadores, usuarios o analistas de negocios. Dan instrucciones para que el sistema se desarrolle proporcionando conocimiento sólido. Deben tener buenas habilidades de comunicación para transmitir los requisitos al equipo de desarrollo. Su participación y su buen conocimiento del dominio pueden mejorar las posibilidades de éxito del proyecto.
- **Equipo de características:** Es un grupo temporal de desarrolladores que trabajan en algunos conjuntos de características durante una iteración. Este equipo se disuelve cuando un conjunto de características fue implementada con éxito y se promocionó a la compilación principal.

2.2. Inteligencia de negocios

2.2.1. Definición de la inteligencia de negocios

Según Azeroual y Theel (2019), el término inteligencia de negocios (BI por sus siglas en inglés) fue introducido por el analista Howard Dresner (Gartner Group) a mediados de

los 90, y fue definido como un término colectivo para conceptos y métodos que apoyan la toma de decisiones a través del análisis, entrega y procesamiento de la información. Stackowiak *et al.* (2007) define la inteligencia de negocios como “proceso de obtener grandes cantidades de datos, analizarlos y presentar un conjunto de informes de alto nivel que condensan la esencia de esos datos en la base de las acciones comerciales, permitiendo a la administración tomar decisiones fundamentales de negocios diarias”. La inteligencia de negocios también puede ser definida, según Ranjan (2005), como “la aplicación de un conjunto de metodologías y tecnologías para mejorar la efectividad de la operación empresarial, apoyar la gestión y toma de decisiones para lograr ventajas competitiva”.

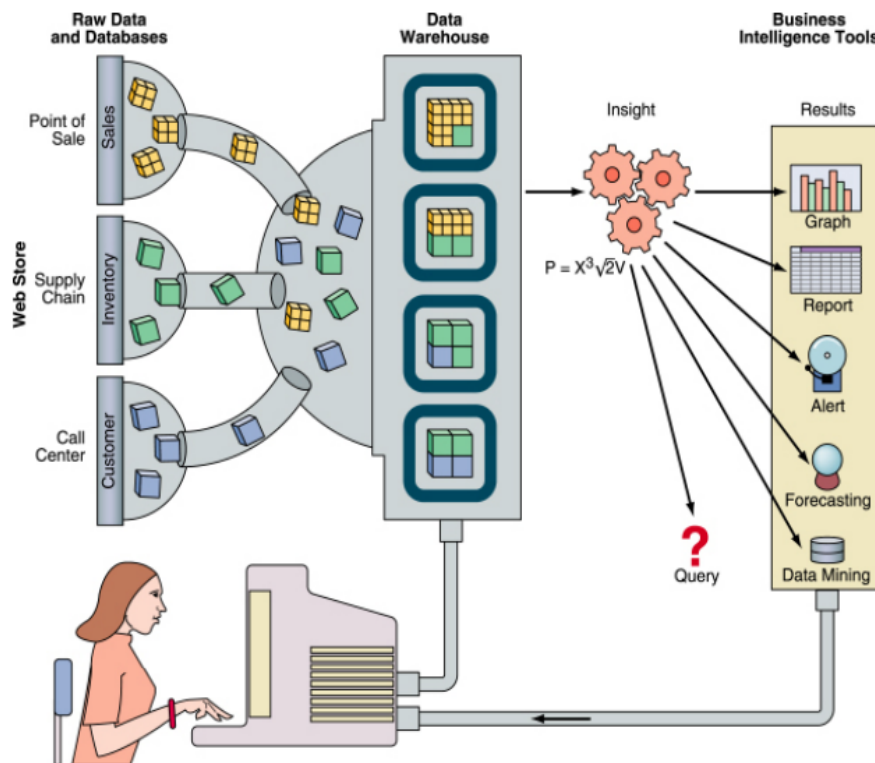


Figura 8: Entendimiento básico de la inteligencia de negocios
Fuente: Ranjan (2005), Institute of Management Technology.

Tal como se puede apreciar en la figura 8, los datos son la piedra angular en la arquitectura BI, por lo que dicha arquitectura debe estar definida basada en ellos. En Azeroual y Theel (2019), se distingue en la arquitectura BI los siguientes elementos:

- **Colección de datos:** Corresponde a la identificación de los datos que son requeridos por los sistemas BI y los sistemas operacionales que los proveen, tanto internos como externos.

- **Integración de datos:** Consiste en la forma de extraer los datos desde los sistemas operacionales u otros orígenes, realizando las transformaciones necesarias para garantizar que los datos puedan ser almacenados pesistentemente en el almacén de datos o mantenidos. También se le conoce como el proceso de extracción, transformación y carga o ETL (por sus siglas en ingles).]
- **Almacenamiento de datos:** La definición de la arquitectura a utilizar para el almacenamiento de datos (Data Warehouse, Data Marts, etc).
- **Procesamiento de datos:** También conocido como análisis de datos, consiste en todos los conceptos y herramientas que se utilizaran para analizar los datos, como por ejemplo el análisis de datos en linea (OLAP por sus siglas en ingles), o componentes de minería de datos para detectar patrones, etc.
- **Presentación de datos:** Definición de como se prepararan y presentaran los resultados de los análisis de datos (Dashobard, Cockpikt, reportes, etc).

La Figura 9 muestra una vista de los elementos antes mencionados y su rol en BI.

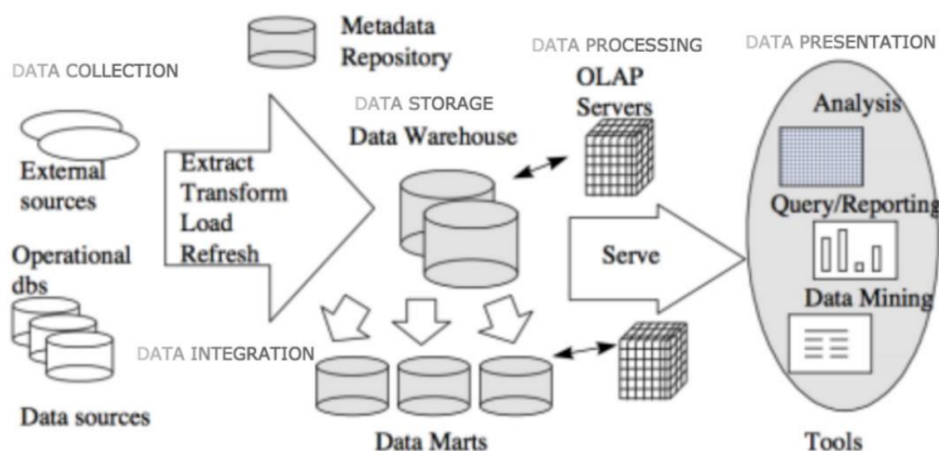


Figura 9: Proceso de la inteligencia de negocios

Fuente: Azeroual y Theel (2019), Azeroual, Otmane and Theel, Horst.

Según Quinn (2009), BI puede ser usado en 3 formas distintas: Estratégica, operacional y analítica. Estos niveles del BI, aunque son únicos en su propia forma, no son excluyentes entre si; deberían estar directamente conectados unos con otros, pues el análisis estratégico conduce al BI analítico, mientras que el BI analítico se enfoca en las iniciativas operacionales, las que impactan en la agilidad, productividad, costo-eficiencia y rentabilidad.

La Figura 10 ilustra como estas 3 formas del BI se relacionan entre si, para generar un ciclo, que, en la practica, nunca termina.



Figura 10: Los 3 niveles de la inteligencia de negocios

Fuente: Azeroual y Theel (2019), Azeroual, Otmane and Theel, Horst.

2.2.2. Herramientas de la inteligencia de negocios

En el mercado existen una gran cantidad de herramientas y soluciones de inteligencia de negocios orientadas a apoyar el monitoreo, control y toma de decisiones del negocio, por lo cual se vuelve necesario conocer cuales son las opciones que entrega el mercado, y como estas se posicionan en el mercado. Por lo anterior, la elección de una herramienta adecuada puede ser un proceso complicado; sin embargo existen organizaciones que se encargan de facilitar esta tarea a través del estudios de distintos tipos de soluciones y herramientas, como Gartner Inc. o Forrester Research.

El cuadrante mágico de Gartner (Garner Inc.)

El cuadrante mágico de Gartner es una metodología para para determinar la posición de una herramienta en el mercado y comparar herramientas y plataformas basadas en dos principales factores, que son la “habilidad de ejecutar” y la “completitud de la visión”. La “habilidad de ejecutar”, según Gartner, corresponde a la capacidad de cada proveedor para hacer de su visión una realidad de mercado que los clientes consideran diferenciada y que están preparados para comprar. La capacidad de un proveedor para ofrecer una experiencia positiva al cliente, que abarca experiencia de ventas, soporte, calidad del producto, habilitación del usuario, disponibilidad de habilidades y facilidad

de actualización y migración son factores que también influyen en su posición en el eje de la capacidad para ejecutar. La “completitud de la visión” corresponde a la capacidad de los proveedores para comprender cómo se pueden explotar las fuerzas del mercado para crear valor para los clientes y oportunidades para ellos mismos.

En la Figura 11 se ilustra cuales son las empresas que figuran en el cuadrante mágico de Gartner, a enero 2023.



Figura 11: Cuadrante Mágico de Gartner para plataformas de analítica e inteligencia de negocios

Fuente: Gartner (Enero 2023).

La Onda Forrester (Forrester Research)

La Forrester Wave o Onda Forrester es una forma gráfica de identificar a las distintas opciones tecnológicas en el mercado para problemáticas específicas. Funciona como una guía para los compradores considerando las opciones disponibles en el mercado tecnológico. Este gráfico aplica diferentes criterios para evaluar las opciones del mercado,

y así distinguir entre Retadores, Contendientes, Ejecutores fuertes y Líderes, además de mostrar visual y rápidamente la presencia de la herramienta en el mercado. Las tres categorías de los criterios de evaluación de la Onda Forrester son la oferta actual (*current offering*), la estrategia (*strategy*), y la presencia en el mercado (*market presence*).

En la Figura 12 se aprecia la Onda Forrester para plataformas empresariales de inteligencia de negocios, emitido durante el 2023.

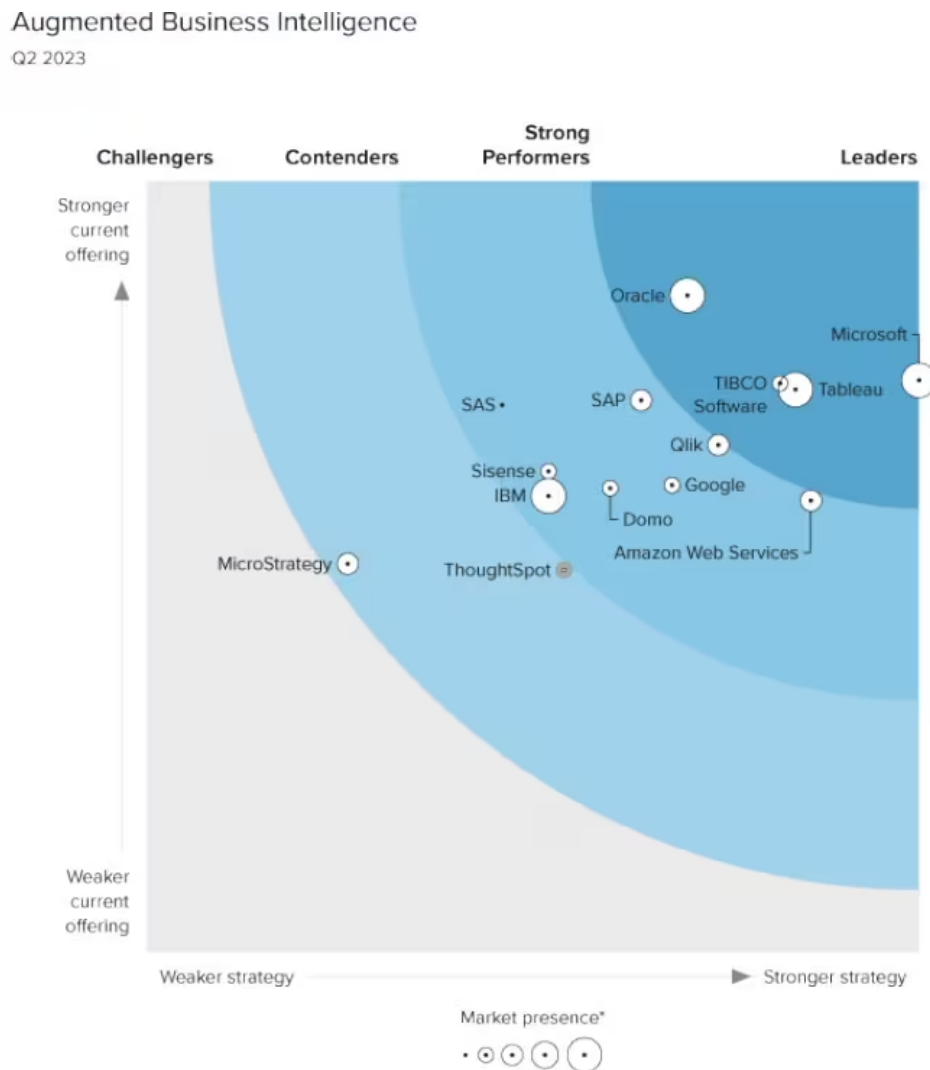


Figura 12: Onda Forrester para plataformas de inteligencia de negocios
Fuente: Forrester Research (2023).

Tanto Gartner Inc como Forrester Research resultan ser una excelente guía para determinar las mejores soluciones disponibles en el mercado. Se usarán ambos para determinar quienes son los líderes en lo que respecta a plataformas de inteligencia de negocios, y se evaluará los costos asociados a las soluciones ofrecidas.

Microsoft y Power BI

Es la mejor evaluada por en ambos estudios, y según la sección Precios del sitio oficial (Microsoft (2019)), este presenta las siguientes diferentes tipos de suscripción, dependiendo de las necesidades del negocio, que se detallan en la Tabla 1.

Tipo	Descripción General	Precio (USD)
Power BI Pro	Inteligencia empresarial moderna con características de autoservicio en la nube. Colaboración, publicación, uso compartido y análisis Ad Hoc. Totalmente administrado por Microsoft.	\$9,99 mensual por usuario.
Power BI Premium	Inteligencia empresarial, análisis de macrodatos, e informes en la nube y en el entorno local. Controles de administración e implementación avanzados. Recursos de almacenamiento y proceso en la nube dedicados. Consumo de contenido de Power BI permitido para cualquier usuario.	\$4.995 mensual por recurso de almacenamiento y proceso en la nube dedicado.

Tabla 1: Comparación de planes Microsoft Power BI

Tableau Software

Otra empresa líder en ambos estudios es Tableau Software, con su herramienta BI del mismo nombre, cuyos precios, según su sitio oficial (Software (2019)), son los que se detallan en la Tabla 2.

Tipo	Descripción General	Precio (USD)
Individual (Tableau Creator)	Licencia de desarrollo.	\$70 mensual por usuario.
Implementación con Tableau Server	Implementación en las instalaciones físicas o en la nube publica.	\$70 mensual por usuario (Tableau Creator). \$35 mensual por usuario, mínimo 5 usuarios (Tableau Explorer) \$12 mensual por usuario, mínimo 100 usuarios (Tableau Viewer)
Implementación con Tableau Online	Solución completamente hospedada en servidores de Tableau	\$70 mensual por usuario (Tableau Creator). \$42 mensual por usuario, mínimo 5 usuarios (Tableau Explorer) \$15 mensual por usuario, mínimo 100 usuarios (Tableau Viewer)

Tabla 2: Comparación de planes Tableau Software

2.3. Visualización de datos

Según Tableau (2022), se puede definir visualización de datos como “la representación gráfica de información y datos”. Mediante el uso adecuado de diversos elementos visuales como tablas, gráficos, mapas, entre otros, es posible entregar rápida y efectivamente información de valor a empleados, dueños de negocio y cualquier otra audiencia no técnica, con el fin de poder apoyar la toma de decisiones. La visualización de datos es, por tanto, parte importante de este trabajo, por lo que es necesario conocer cuáles son sus principales ventajas, desventajas, buenas practicas y cosas se deben evitar.

Ventajas

- **Accesibilidad:** No se requiere de alto conocimiento técnico para acceder y analizar los datos.
- **Rápida interpretación:** Ayudan a los usuarios a interpretar la información de forma rápida, permitiendo identificar patrones y tendencias.
- **Apoyo a la toma de decisiones:** Permiten identificar tendencias o patrones de forma visual.

Desventajas

- **Suposiciones inexactas/Pobre calidad de datos:** Puede llevar a suposiciones inexactas si la visualización esta mal diseñada y/o los datos utilizados son poco confiables o imprecisos.
- **Costos:** Algunas herramientas BI pueden ser costosas, o tener características limitadas, lo que puede suponer una barrera de entrada para su uso domestico o pequeñas empresas.

2.3.1. Atributos visuales y tipos de datos

Según Wilke (2019), los atributos visuales describen cada aspecto de un determinado elemento gráfico, como lo son la posición, formas, tamaños, colores, como también el ancho de las líneas, tipos de líneas, fuentes, entre otros, los que se observan en la Figura 13.

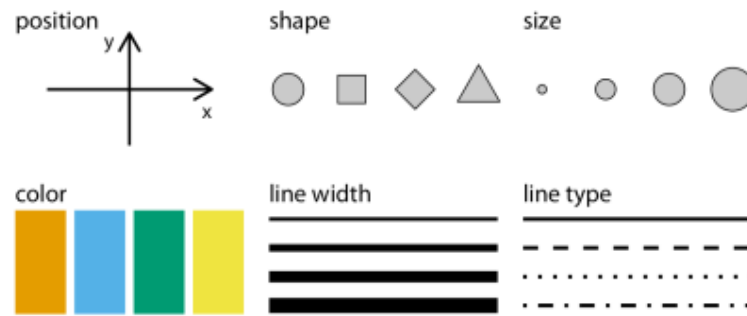


Figura 13: Atributos visuales típicos en visualización de datos

Fuente: Fundamentos de la visualización de datos, Claus O. Wilke (Marzo 2019).

Los atributos mencionados se pueden clasificar según el tipo de dato que pueden representar, que pueden ser datos continuos y/o datos discretos. Los datos continuos son aquellos en los que siempre es posible encontrar un valor intermedio entre dos medidas (ej: Tiempo), en cambio, los datos discretos son valores en los cuales lo anterior no es posible (ej: Cantidad de personas en una habitación).

Wilke (2019) indica que con la visualización de datos lo que se busca es “presentar a la audiencia una historia emocionante y significativa”. Para esto, hay ciertos puntos clave a considerar en la elección de las visualizaciones y su forma de presentación:

- **Haz una figura para los generales:** Nunca se debe asumir que la audiencia puede procesar rápidamente visualizaciones complejas; se debe simplificar las figuras tanto como sea posible, solo los puntos principales deberían permanecer.
- **Construye hacia figuras complejas:** No se debe partir directamente con figuras complejas, primero se debe mostrar alguna versión simplificada de la figura a mostrar.
- **Has tus figuras memorables:** Si bien las visualizaciones simples tienen varias ventajas (evitan la distracción, son fáciles de leer, y permiten a la audiencia concentrarse en los puntos importantes), el uso excesivo de estas puede provocar el efecto contrario, por lo que agregar elementos distintivos como figuras que representen visualmente los datos pueden ayudar a que estas gráficas se vuelvan memorables.
- **Ser consistente sin ser repetitivo:** Las gráficas usadas tienen que demostrar que son todas parte de una misma historia, pero evitando repetir usar la misma visualización, no usar siempre la misma gráficas, evitar, por ejemplo, para un mismo tema, mostrar dos o más gráficas de barra seguidas.

CAPÍTULO 3

DISEÑO DE LA SOLUCIÓN

3.1. Modelo de solución

Basado en la problemática a resolver, se puede dividir el modelo de solución en tres partes, como se muestra en la figura 14.

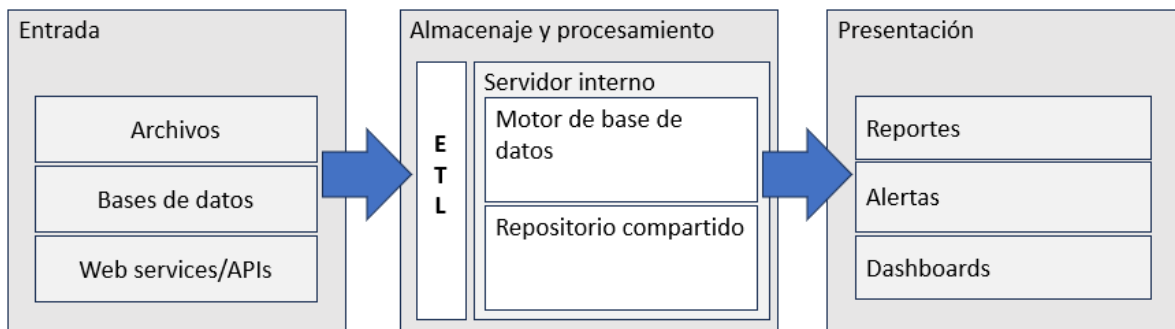


Figura 14: Modelo simplificado de solución

Fuente: Elaboración propia.

3.1.1. Entradas

La primera parte del modelo de solución propuesto, corresponde a todas las fuentes de datos del proyecto, que en este trabajo corresponde a las bases de datos internas con la información demográfica y financiera de los clientes, los archivos con información complementaria, y otros orígenes internos con acceso restringido. Para estos últimos, la forma de entrega actualmente disponible es mediante archivos de texto plano, depositados por el área de operaciones en una ruta compartida interna de la organización. Las entradas a considerar en el proyecto se detallan a continuación.

- **Información demográfica y financiera:** Son los datos propios del cliente como también de los distintos productos que posee en la organización. Esta información, que forma parte de las bases de la organización, tiene un acceso restrictivo y poco práctico para realizar análisis y cruces de información por parte del equipo de prevención de fraude, por lo que su integración con las bases de datos del área es parte crucial del proyecto.
- **Información transaccional financiera:** Es la información transaccional de compras, transferencias, créditos, devoluciones, pagos, entre otros movimientos que afecten los productos y cuentas de los clientes. Esta información recibida por

el motor de prevención de fraude, sin embargo, no esta disponible de forma practica para el equipo, debido a que al formar parte de la trama de autorización, esta sujeta a los estándares internacionales normativos que rigen sobre el tratamiento de datos de esta índole, según PCISSC (2023) en norma PCI DSS de seguridad de datos para la industria de tarjetas de pago.

- **Información transaccional no financiera:** Es la información respecto a la interacción necesaria para poder realizar transacciones no presenciales. Algunos ejemplos son los ingresos autenticados a las aplicaciones móviles o vía navegador web del cliente, el uso de claves de seguridad, cambios de datos demográficos del cliente, dispositivos utilizados, entre otros.
- **Información complementaria histórica:** Es toda información que no necesariamente forma parte de la organización, si no que provienen de otras fuentes externas, como proveedores externos por ejemplo. Por políticas de seguridad y confidencialidad de la estrategia de prevención de fraude, estos no serán detallados en el proyecto.

3.1.2. Almacenaje y procesamiento

La segunda parte del modelo corresponde a donde serán almacenados y procesados los datos cargados. Estos deben estar disponibles en una base centralizada, para que el equipo pueda hacer los análisis y cruces de información requeridos. El negocio dispone de un servidor en la red interna de la organización para este fin, con un motor de base de datos Microsoft SQL Server 2011, y espacio de almacenamiento adecuado para la recepción de las bases en archivo de texto plano mencionados en la primera parte del modelo.

3.1.3. Presentación

La tercera parte del modelo corresponde a la forma en que se presentará la información procesada anteriormente, para dejarla disponible a los principales interesados dentro del negocio. Además de permitir el acceso de consulta a las tablas cargadas en la base de datos, la información será usada para monitoreo transaccional, control de cargas, generación de alertas, mediante el desarrollo de reportes y tableros.

3.2. Metodología de desarrollo

La elección de la metodología ágil se realizó mediante la comparación de las metodologías expuestas en el la sección 2.1, donde se presentó XP, SCRUM y FDD. En base a

lo expuesto, se elaboró la Tabla 3:

Modelo	Tamaño del equipo	Tamaño de la iteración	Tamaño del proyecto	Características clave
XP	5 a 15	1 a 2 semanas	Proyectos pequeños	-Enfocado en el diseño simple y refactorización. -Reemplaza la documentación basada en papel por la comunicación cara a cara. -Generalmente se limita a problemas existentes. -Aplica pruebas de regresión fuerte. -Inspección continua de código a través del <i>par programming</i> .
SCRUM	2 a 6	1 a 4 semanas	Mediano a largos	-Fomenta la colaboración entre el cliente y el equipo. -Permite la auto gestión del equipo. -Deja de lado la documentación extensiva. -Es flexible y adaptativo según las prioridades del negocio.
FDD	4 a 20	1 a 4 semanas	Largos	-Escalable a grandes equipos. -Prácticas de desarrollo altamente especificadas. -Cinco subprocesos, cada uno definidos con criterios de entrada y salida. -Desarrollo arquitectónico, con modelos UML de objeto y secuencia.

Tabla 3: Resumen comparativo entre metodologías ágiles XP, SCRUM y FDD

Considerando lo ilustrado en la Tabla 3, para determinar la metodología adecuada, es necesario conocer las características y recursos disponibles para el proyecto en cuestión, las que se listan a continuación:

- **Tamaño del equipo:** Se dispondrá de uno a dos desarrolladores a lo mucho.
- **Tamaño de la iteración:** Requerirá entregas rápidas, dentro de la misma semana, basado en las necesidades del área.
- **Tamaño del proyecto:** El tamaño corresponde al de un proyecto pequeño a mediano.

- **Tipo de proyecto:** Proyecto interno, donde el ROI esta enfocado en tener disponible la información lo antes posible para análisis y respuesta oportuna a posibles conductas o eventos de fraude.

Dadas las características antes mencionadas, y contrastándolas con el resumen de la Tabla 3, se optará por la aplicación de la metodología XP para este proyecto.

3.3. Herramientas seleccionadas

Respecto a las herramientas a utilizar, estas están limitadas a las disponibles en el negocio, por lo que tras la revisión y evaluación de las alternativas disponibles, se definió que se utilizaran las siguientes:

- **Integración de datos:** Talend Open Studio.
- **Motor de base de datos:** Microsoft SQL Server.
- **Reportabilidad:** Microsoft Power BI.

3.4. Diseño de reportes y tableros

El proyecto contemplará la elaboración de los siguientes reportes:

- **Reporte resumen comercios:** Reporte para monitorear la actividad de los distintos comercios en los que operan las tarjetas de los clientes, con un enfoque especial en los comercios nuevos, que no presentaban transacciones y que tienen un aumento considerable de compras o intentos de compra. El panel debe considerar la segmentación por cuadrantes (nacional, internacional, presente, ausente y sus respectivas combinaciones), y cada comercio debe tener indicadores clave (puntos base, cantidad de tarjetas involucradas, tasa de rechazo). El objetivo de este reporte es permitir identificar oportunamente desviaciones de los indicadores clave por comercio, para realizar las gestiones pertinentes, de tal forma de mantener los niveles de fraude y aprobación dentro de los parámetros requeridos por el negocio.
- **Reporte monitoreo calidad transaccional:** Reporte para monitorear que la calidad transaccional de los diferentes tipos de transacción cumpla con los umbrales de aceptación que defina el negocio. Los indicadores que deben ser monitoreados son *Time Out* (TO), transacciones *Real Time* (RT) sin *Near Real Time* (NRT), o *Near Real Time* sin *Real Time*. Estos escenarios son importantes para el negocio ya que su aumento de ocurrencia se traduce a aprobación de transacciones que en circunstancias normales habrían sido rechazadas por la estrategia vigente implementada el motor de prevención de fraude.

- **Reporte monitoreo cargas de bases:** Reporte para monitorear que todas las cargas fundamentales realizadas como parte de este proyecto se hayan efectuado, y así, en caso de detectar anomalías en el flujo, poder actuar oportunamente para corregir la situación y mantener las bases actualizadas.

3.5. Plan de trabajo

Se requerirá realizar en primera instancia los procesos de carga de información, mencionados anteriormente en la sección 3.1.1, los que se definirán a continuación, de forma general, por temas de confidencialidad de la organización y de la estrategia de prevención de fraude. Adicionalmente, y según la metodología XP, para cada requerimiento se definirán las historias de usuario correspondiente, las pruebas unitarias y de aceptación.

3.5.1. Transacciones financieras

Las transacciones financieras, se pueden clasificar en dos grandes grupos, que se detallan a continuación:

- **Compra:** Corresponde a todas las transacciones en comercios presenciales y no presenciales (internet), realizadas usando la tarjeta de crédito o débito del cliente. Se puede clasificar estas transacciones en dos grandes subgrupos relacionados, que son las *Real Time* (en adelante RT) y *Near Real Time* (en adelante NRT), que vendrían siendo la transacción que llega para ser evaluada por el motor de prevención de fraude, y la transacción de confirmación de respuesta final de cara al cliente, respectivamente.
- **Transferencia:** Corresponde a todas las transacciones de transferencia realizada por los clientes, son movimientos de dinero disponible en la cuenta corriente/vista/ahorro del cliente a otra cuenta corriente/vista/ahorro, que puede ser de la misma organización u otra entidad bancaria nacional. Al igual que las transacciones de compra, se pueden sub-clasificar en RT y NRT, y en este caso, adicionalmente en ABR, que vendría siendo una pre-autorización.

Las historias de usuario para estas bases son las detalladas en la Tabla 4.

Tipo	sub-tipo	Historia de usuario
Compra	RT y NRT	Se quiere contar con las transacciones de compras que llegan al motor de prevención de fraude, para poder realizar análisis de patrones transaccionales de fraude para compras ausentes, presentes, nacionales o internacionales, y sus diferentes combinaciones.
Transferencia	RT, NRT y ABR	Se quiere contar con las transacciones de transferencias que llegan al motor de prevención de fraude, para poder realizar análisis de patrones transaccionales de fraude para destinatarios recurrentes, nuevos, cuentas propias mismo banco u otros bancos.

Tabla 4: Historias de usuario para transacciones financieras

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 5.:

Tipo	sub-tipo	Pruebas unitarias	Pruebas aceptación
Compra	RT y NRT	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Transferencia	RT, NRT y ABR	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.

Tabla 5: Pruebas unitarias y de aceptación para transacciones financieras

3.5.2. Transacciones no financieras

Las transacciones no financieras corresponden a transacciones de interacción del cliente con su sitio web privado o aplicación móvil de la organización. También se incluyen en esta categoría las acciones automáticas gatilladas por el motor de prevención fraude, y las transacciones marcadas como fraude por el equipo o por procesos masivos.

A continuación se detallan estos tipos de transacciones.

- **Notificación VU:** Corresponden a los registros de interacción del cliente en su sitio privado o aplicación móvil de la organización, incluye login correctos y/o erróneos, cambios de datos, instalación de app, entre otros.

- **Notificacion ABR:** Corresponde a los registros con la información del dispositivo desde el cual se realizó un login exitoso. Dentro de la información rescatada se tiene si es web, web mobile, app, dirección IP, geolocalización, información del dispositivo como marca y modelo, entre otros datos.
- **Acciones automáticas:** Corresponde a los registros de las acciones automáticas gatilladas sobre las transacciones por parte del motor de prevención de fraude, identificando la información de la regla que gatilló la acción. Esto aplica tanto para transacciones financieras como no financieras. Algunos ejemplos de estas acciones son el envío de email, envío de sms, generación de alertas, aprobar o denegar la transacción, entre otros.
- **Fraude marcado:** Corresponde a todas las transacciones que fueron marcadas como fraude por algún analista o de forma masiva por algún proceso, con la información complementaria del evento, por ejemplo usuario que marco la transacción, fecha de detección, tipo de fraude, entre otros.

Las historias de usuario para estas bases son las detalladas en la Tabla 6.

Tipo	Historia de usuario
Notificación VU	Se quiere contar con las notificaciones de interacción del cliente con su cuenta bancaria a través de la app y/o web, para poder analizar patrones de conducta de alto y bajo riesgo y utilizar esta información para definir nuevas estrategias y mejoras en la prevención de fraude.
Notificación ABR	Se quiere contar con las interacciones desde los distintos dispositivos a través de los cuales el cliente interactúa con su cuenta, para determinar si esta utilizando un dispositivo habitual o nuevo, para poder analizar patrones de conducta de alto y bajo riesgo y utilizar esta información para definir nuevas estrategias y mejoras en la prevención de fraude.
Acciones automáticas	Se quiere contar con el detalle de las acciones automáticas gatilladas por el motor de prevención de fraude sobre las transacciones financieras y no financieras que son analizadas en RT y NRT, para poder evaluar el falso positivo de las reglas implementadas y que tan efectivas esta siendo la estrategia de prevención, y detectar oportunidades de mejora.
Fraude marcado	Se quiere tener identificadas todas las transacciones con fraude marcado gestionado por los analistas de prevención de fraude, para poder hacer análisis y mejora de las reglas de prevención de fraude respecto al fraude confirmado.

Tabla 6: Historias de usuario para transacciones no financieras

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 7.

Tipo	Pruebas unitarias	Pruebas aceptación
Notificación VU	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Notificación ABR	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados por día, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Acciones automáticas	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Fraude marcado	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados por día, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.

Tabla 7: Pruebas unitarias y de aceptación para transacciones no financieras

3.5.3. Información demográfica de clientes

Corresponde a la información de productos y cuentas de los clientes, que incluye su información personal registrada con sus productos vigentes. Son múltiples orígenes los utilizados, los que no se detallarán, pero que de igual forma serán abordados en el proceso de carga.

Las historias de usuario para estas bases son las detalladas en la Tabla 8.

Tipo	Historia de usuario
Información personal	Se quiere contar con la información personal actualizada de números de teléfono, direcciones, emails, etc, asociados a los clientes para poder enriquecer los análisis de prevención de fraude y detectar actividades anómalas para la mejora de la estrategia de prevención de fraude.
Información de productos	Se quiere contar con las información de los productos que los clientes mantienen vigentes en la organización, para complementar los análisis y utilizar esta información para definir nuevas estrategias, mejoras en la prevención de fraude y apoyar la gestión de casos/reclamos.

Tabla 8: Historias de usuario para información demográfica de clientes

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 9.

Tipo	Pruebas unitarias	Pruebas aceptación
Información personal	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Información de productos	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.

Tabla 9: Pruebas unitarias y de aceptación para información demográfica de clientes

3.5.4. Información complementaria

La organización cuenta con bases de información complementaria para poder perfilar mejor a los clientes y su información histórica, ya sea a través de bases propias o de terceros, por lo que su carga y centralización para análisis y detección de posibles fraudes es parte importante del proceso. Estos no serán detallados, pero también serán abordados y contemplados en el proceso de carga.

Las historias de usuario para estas bases son las detalladas en la Tabla 10.

Tipo	Historia de usuario
Base de envíos SMS	Se quiere contar con la base de SMS enviados desde el motor de prevención de fraude, con su detalle de si fue realmente entregado, para poder consultar rápidamente que fue lo que se le envió al cliente, y verificar rápidamente si este mensaje fue efectivamente entregado.
Respuestas SMS	Se quiere contar con la base de respuestas a los SMS enviados desde el motor de prevención de fraude, para medir y mejorar la efectividad de la estrategia de prevención de fraude basada en el envío de SMS, y detectar con ello oportunidades de mejora.

Tabla 10: Historias de usuario para información complementaria

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 11.

Tipo	Pruebas unitarias	Pruebas aceptación
Base de envíos SMS	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados, por día y hora, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.
Respuestas SMS	-Ejecución flujo principal OK -Ejecución flujo error (archivo(s) no encontrado(s))	Conteo de registros cargados por día, para los últimos 5 días, manteniendo tendencias, sin anomalías y ventanas faltantes.

Tabla 11: Pruebas unitarias y de aceptación para información complementaria

3.5.5. Plantilla base para carga de archivos

Dado que, en su mayoría, todos los procesos de carga presentan similitudes, y con el fin de facilitar la implementación de estos, se creó una plantilla base para carga desde archivos, usando Talend Open Studio, tal como se ve en la figura 15.

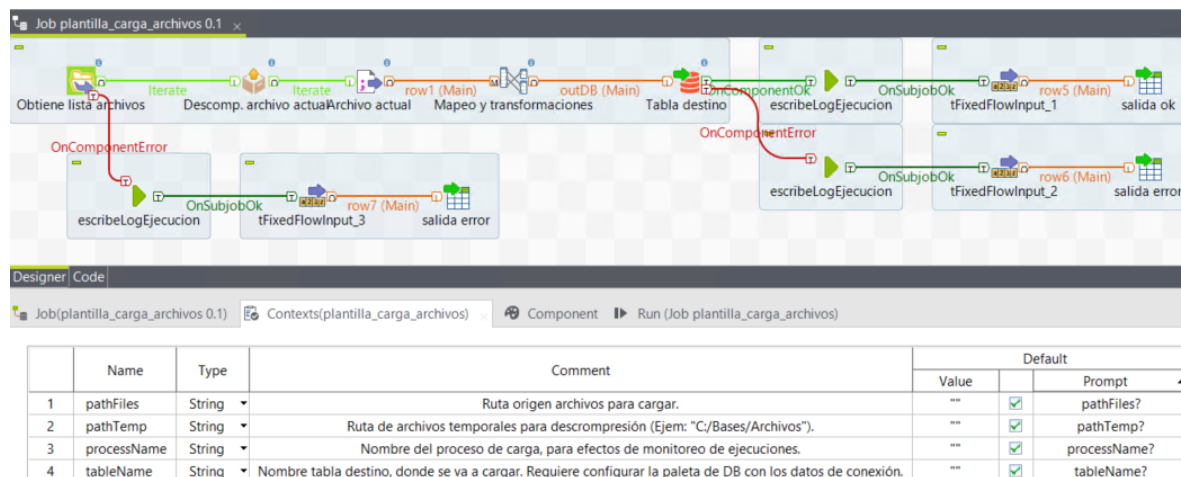


Figura 15: Plantilla base para carga de archivos, creada en Talend Open Studio.

Todo *job* de carga que sea creado en base a la plantilla de la Figura 15, requiere la configuración de los siguientes elementos:

- **Variables de contexto:** Para cada *job* que se creado a partir de la plantilla base, se debe configurar las variables de contexto predefinidas (pestaña *Context* en la figura 15), Estas variables son “pathFiles”, “pathTemp”, “processName” y “table-Name”, y deben completarse siguiendo las instrucciones indicadas en el campo *Comment* de la misma variable.

- **Paleta “Obtiene lista archivos”:** Paleta utilizada para iterar sobre los archivos en el directorio indicado en la variable de contexto “pathFiles”, se puede configurar para que solo acepte algunos tipos de archivos o ciertas estructuras de nombre usando expresiones regulares. Aquí lo importante es que se debe configurar el esquema del archivo a cargar (campos, delimitador, tipo de dato, etc.).
- **Paleta “Descomp. archivo actual”:** Paleta utilizada para descomprimir el archivo actual e iterar sobre cada uno de los archivos dentro del archivo comprimido. Para efectos del proyecto, todas las entregas vienen comprimidas, pero esta paleta puede ser omitida, pero se tendría que configurar la paleta “archivo actual” para que trabaje con la iteración de la paleta “Obtiene lista archivos”.
- **Paleta “Archivo actual”:** Paleta utilizada para leer el archivo de la iteración actual de la paleta “Descomp. archivo actual”. En esta paleta se debe configurar el esquema del archivo a cargar y los delimitadores de filas, columnas y texto.
- **Paleta “Mapeo y transformaciones”:** Paleta para realizar el mapeo y transformación, si es necesario, de los campos que vienen desde la paleta “Archivo actual”. Su configuración depende del esquema configurado en “Archivo actual” y el esquema de “Tabla destino”.
- **Paleta “Tabla destino”:** Paleta usada para la escritura en la tabla de base de datos, configurada en la variable de contexto “tableName”. En este caso, es necesario configurar los datos conexión de base de datos; además, considerar que esta paleta está pensada para operar con Microsoft SQL Server 11, pero puede ser perfectamente reemplazada por cualquier otra paleta de escritura en base de datos.
- **Paleta “escribeLoEjecución”:** *Job* reutilizable creado para este proyecto, el que permite registrar en una tabla de base de datos previamente configurada, el registro de la ejecución del *job* donde es invocado. Requiere que se completen las variables de contexto.

El resto de las paletas no requiere cambios, pues están previamente configuradas para que el *job* retorne códigos de error o ejecución ok al *job* padre que las invoque, si es que aplica.

3.5.6. Otros *jobs* reutilizables

Para facilitar la implementación de los *jobs* de carga, se crearon algunos *jobs* simples, reutilizables, que simplifican algunas de las tareas repetitivas simples, y son descritos a continuación.

- ***Job* “Mueve Archivos Directorio”:** *Job* reutilizable para mover archivos desde el directorio origen configurado en la variable de contexto “pathFile”, al directorio

destino configurado en la variable de contexto “destinationPathFile”, tal como se muestra en la figura 16.

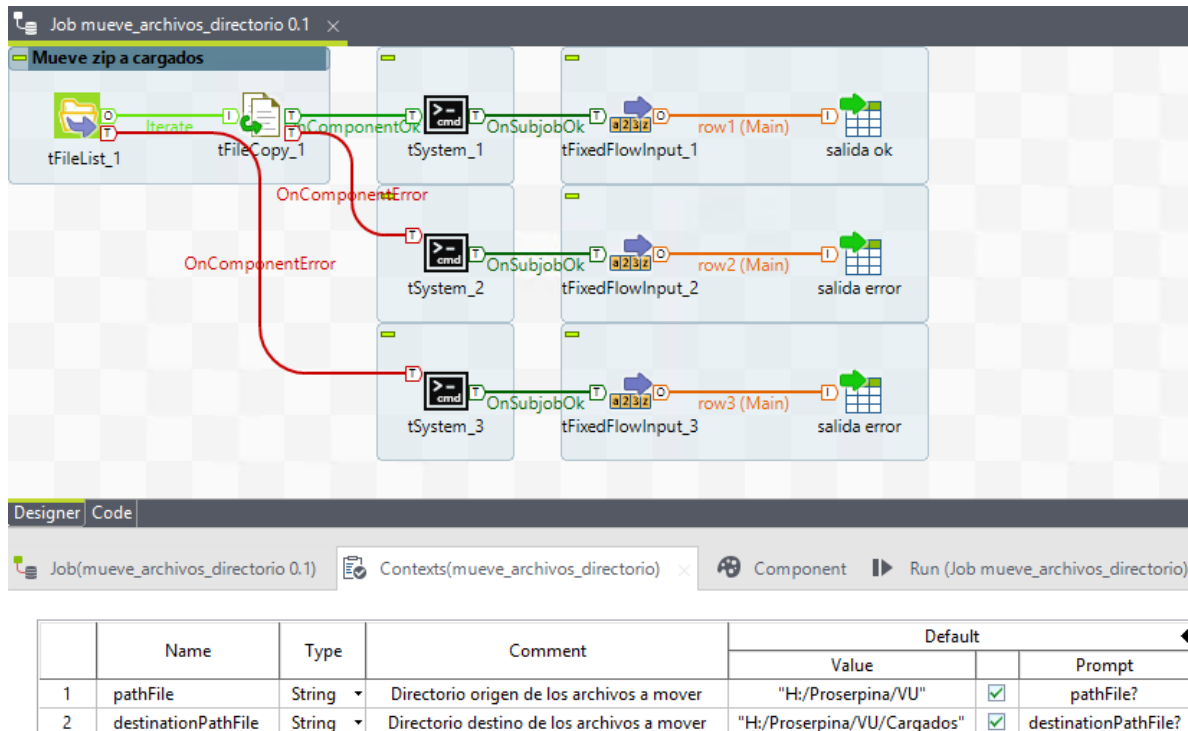


Figura 16: *Job* reutilizable que mueve archivos desde una ruta a otra

- ***Job* “Escribe Log Ejecución”:** *Job* reutilizable para escribir en una tabla de registro de ejecución de *Job*, los datos recibidos mediante la configuración de las variables de contexto “Process”, “State” y “Log”, tal como se muestra en la Figura 16. Requiere que se configure los datos de conexión en la paleta “escribe log ejecución”.

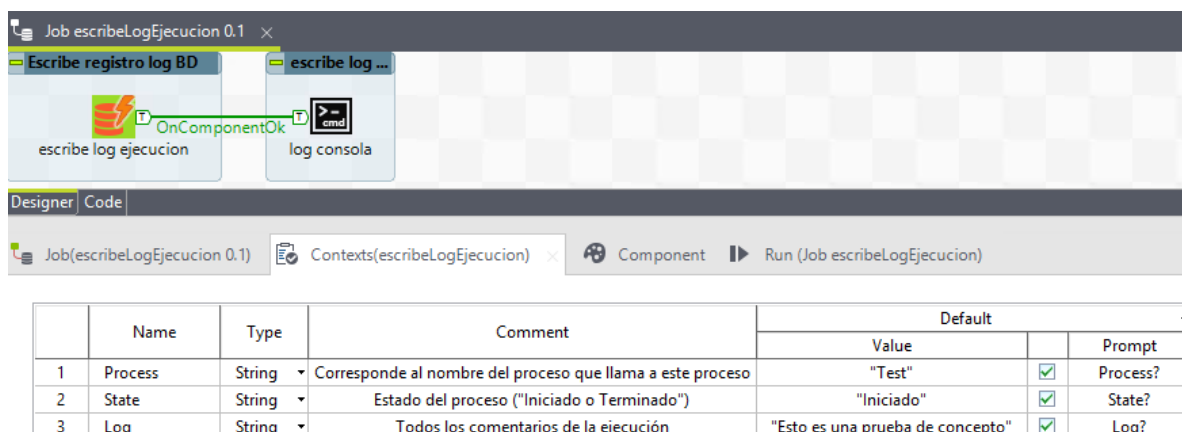


Figura 17: *Job* reutilizable que escribe en una tabla el registro de ejecución

3.5.7. Reporte resumen comercios

Contando ya con la información transaccional cargada, es posible utilizarla para la identificación de posibles comercios de prueba, de alto riesgo o poco confiables, es por eso que un panel para realizar este tipo de seguimiento es fundamental.

La historia de usuario de este reporte se detalla en la Tabla 12.

Reporte	Historia de usuario
Resumen comercios	Se quiere contar con un reporte de monitoreo de comercios, con enfoque en los comercios que empiezan a presentar transacciones con las tarjetas de los clientes, y que permita detectar visualmente anomalías o desviaciones significativas, para el apoyo en la identificación oportuna de posibles pruebas masivas de tarjetas o comercios de alto riesgo.

Tabla 12: Historia de usuario para confección del reporte resumen de comercios

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 13.

Reporte	Pruebas unitarias	Pruebas aceptación
Resumen comercio	-Elementos visuales sin error tras actualizar. -Indicadores consistentes con la información de los gráficos. -Publicación sin errores del reporte.	Revisión post carga y publicación del reporte para verificar visualmente que el reporte muestra la información actualizada.

Tabla 13: Pruebas unitarias y de aceptación reporte resumen comercios

3.5.8. Monitoreo calidad transaccional

Para que el motor de prevención de fraudes opere correctamente, es necesario que las transacciones lleguen a tiempo y en la forma esperada para ser evaluada por el motor, y que la respuesta del motor de prevención también sea entregada en los tiempos establecidos, dado que si esto no ocurre, no se estarían evaluando correctamente los patrones de fraude, y por consiguiente, podrían autorizarse transacciones de alto riesgo, que en condiciones normales, habrían sido denegadas. Para efectos de este panel, el objetivo es identificar oportunamente la ocurrencia de este tipo de errores, para que se apliquen las contingencias correspondientes.

La historia de usuario de este reporte se detalla en la Tabla 14.

Reporte	Historia de usuario
Calidad transaccional	Se quiere contar con un reporte de monitoreo de calidad transaccional, enfocado en las transacciones ATM-POS, que permita detectar visualmente anomalías o desviaciones significativas, para los escenarios de <i>Time Out</i> , RT sin NRT, NRT sin RT y Desfase de transacciones, para el apoyo en la identificación oportuna de posible errores que afecten la calidad transaccional, y poder implementar rápidamente las medidas de contingencia correspondientes.

Tabla 14: Historia de usuario para confección de reporte de calidad transaccional

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 15.

Reporte	Pruebas unitarias	Pruebas aceptación
Calidad transaccional	-Elementos visuales sin error tras actualizar. -Indicadores consistentes con la información de los gráficos. -Publicación sin errores del reporte.	Revisión post carga y publicación del reporte para verificar visualmente que el reporte muestra la información actualizada.

Tabla 15: Pruebas automáticas y de aceptación reporte calidad transaccional

3.5.9. Monitoreo cargas de datos

Como parte crucial de la solución planteada, el monitoreo de las cargas diaria y por hora es crucial, es por esta razón que el monitoreo de las cargas es necesario, tanto

diaria como por hora, en los casos que aplique.

La historia de usuario de este reporte se detalla en la Tabla 16.

Reporte	Historia de usuario
Cargas de datos	Se quiere contar con un reporte de monitoreo de cargas transaccionales, con enfoque en las transacciones ATM-POS, TEF, notificaciones ABR y VU, Acciones automáticas y marcado de transacciones fraude, que permita detectar visualmente anomalías, desviaciones significativas o bloques faltantes. para realizar las acciones pertinentes, como solicitar el reenvío de la información faltante o contrastar si ocurre algo desde el origen.

Tabla 16: Historia de usuario para confección de reporte de cargas de datos

Las pruebas unitarias y de aceptación a realizar son las detalladas en la Tabla 17.

Reporte	Pruebas unitarias	Pruebas aceptación
Carga de datos	-Elementos visuales sin error tras actualizar. -Indicadores consistentes con la información de los gráficos. -Publicación sin errores del reporte.	Revisión post carga y publicación del reporte para verificar visualmente que el reporte muestra la información actualizada.

Tabla 17: Pruebas automáticas y de aceptación reporte de carga de datos

La propuesta de diseño de solución descrita en este capítulo esta enfocada en proporcionar acceso rápido y actualizado a los datos del cliente, y permitir mediante el desarrollo de reportes el monitoreo el seguimiento y control del fraude, aplicado a la detección de fraude en comercios como caso práctico, permitiendo así una toma de decisiones más precisa y una respuesta efectiva ante las amenazas de fraude. En el próximo capítulo, se profundizará en el desarrollo de la solución, detallando cómo se convertirá esta propuesta en una realidad efectiva.

CAPÍTULO 4

DESARROLLO DE LA SOLUCIÓN

Para el desarrollo de la solución, basado en el diseño del capítulo anterior, se optó por dividirlo en dos grandes etapas: Los *jobs* de carga de datos y los reportes/tableros de control. Ambas etapas son detalladas a continuación.

4.1. Etapa 1: *Jobs* de Carga de datos

En esta etapa, y en base al plan de trabajo, se elaboraron *jobs* para todos los flujos de carga de datos, usando como base la plantilla de carga de archivos creada para estos efectos.

4.1.1. Carga transacciones financieras

Para todos los *jobs* de carga de esta sección, se optó por crear una réplica de las tablas tal como fueron entregadas por el área de operaciones, procurando que los campos clave mantuvieran los formatos adecuados a nivel de base de datos (campos fecha, por ejemplo). Por lo mismo, el esquema de los archivos de entrada y los de salida son, en su mayoría, iguales, salvo algunas transformaciones.

- **Carga transacciones ATM POS RT/NRT:** El archivo recibido en este caso particular, contiene las transacciones ATM POS, tanto RT como NRT, las que por decisión del negocio, fueron almacenadas en la misma tabla a nivel de base de datos. Para su carga, se crearon tres *jobs*, el primero de ellos corresponde al *jobs* orquestador, que hace el llamado de los *jobs* auxiliares y fundamentales para la carga, como se observa en la Figura 18.

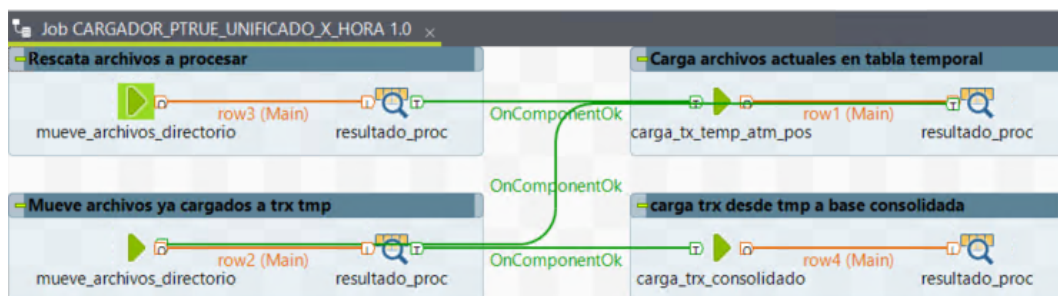


Figura 18: *Job* orquestador de ejecución de flujo carga ATM-POS RT/NRT

Además del llamado al *job* auxiliar que mueve archivos, se tiene el llamado al *job* que carga los archivos que se encuentran en la ruta actual a la base de datos, en la tabla temporal. Este *job* en cuestión, fue creado a partir de la plantilla base definida en el diseño de la solución, dando como resultado el *job* de la Figura 19

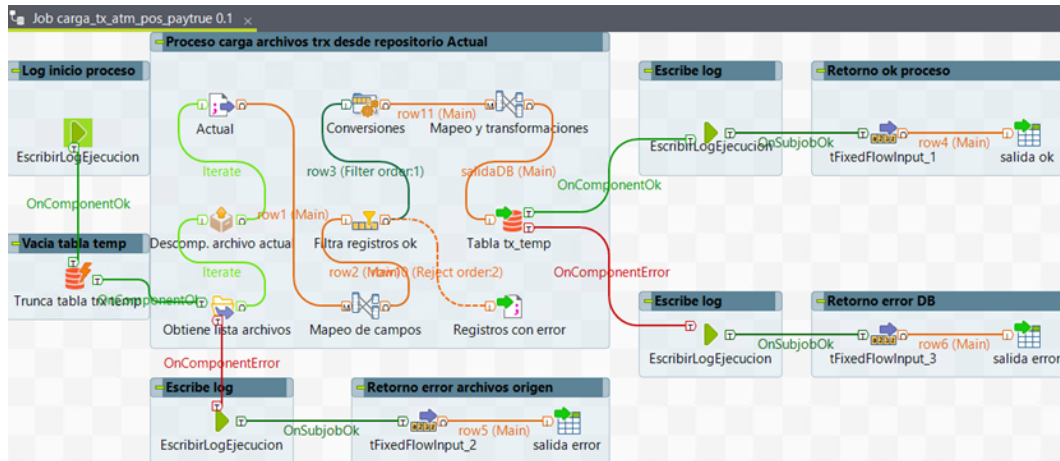


Figura 19: *Job* que carga transacciones ATM-POS RT/NRT a tabla temporal

El *job* de la 19 mantiene la lógica base de la plantilla base, pero con algunos componentes adicionales basado en la lógica particular requerida para esta carga, dentro de las que destacan los registros con error desde el origen, que se almacenan en un archivo temporal, como también la lógica que vacía previamente la tabla temporal al inicio de la ejecución del *job*. Una vez concluida su ejecución correctamente, el orquestador llama al *job* que carga las transacciones desde la tabla temporal a la tabla definitiva, que se encarga de realizar la carga en la tabla definitiva a partir de la tabla temporal, como se ilustra en la figura 20

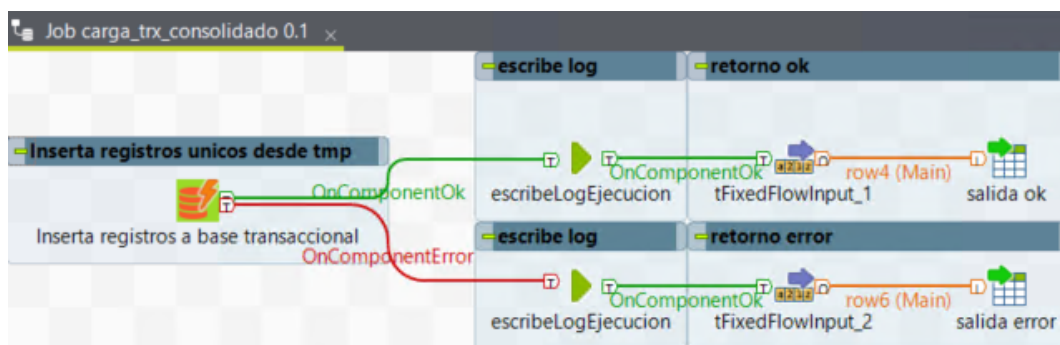


Figura 20: *Job* que carga transacciones ATM-POS RT/NRT en la tabla definitiva

Luego de cargar los registros, se captura el resultado de la ejecución, devolviendo al *job* orquestador su estado actual.

Respecto a las pruebas unitarias definidas, se validó la correcta ejecución del *job*

orquestrador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

- **Carga Transferencias RT/NRT/ABR:** Para los *jobs* de carga de transferencias, a diferencia del *job* de carga ATM-POS, donde los registros RT y NRT venían en un mismo archivo, estos fueron entregados por operaciones en archivos separados, y por decisión del negocio, se optó por mantenerlo en diferentes tablas a nivel de base de datos, creando las tablas según la estructura de los archivos origen recibidos, procurando que los campos clave mantuvieran los formatos adecuados a nivel de base de datos. A continuación se detallan los *jobs* que permiten la carga de los archivos de transferencias.

El *job* orquestador de transferencias de la Figura 21, se encarga de realizar los llamados para los *jobs* que realizan la carga de transferencias RT/NRT/ABR.

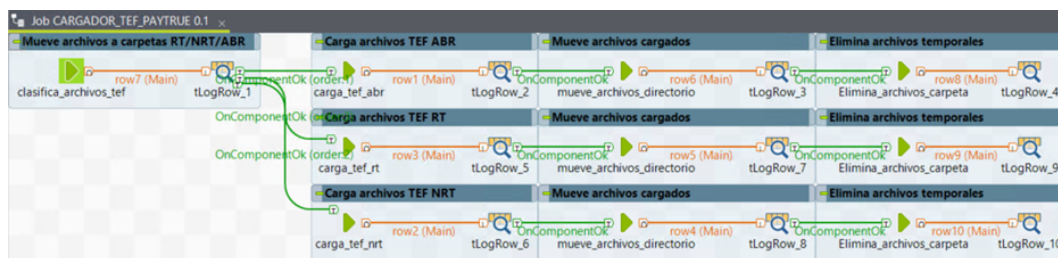


Figura 21: *Job* orquestador de ejecución para la carga de transferencias RT/NRT/ABR

Se comienza con la clasificación de los archivos entrantes, para identificar a qué tipo de registro de transferencia corresponde, según los nombres de los archivos recibidos, lo que se realiza utilizando el *job* auxiliar de la figura 22.

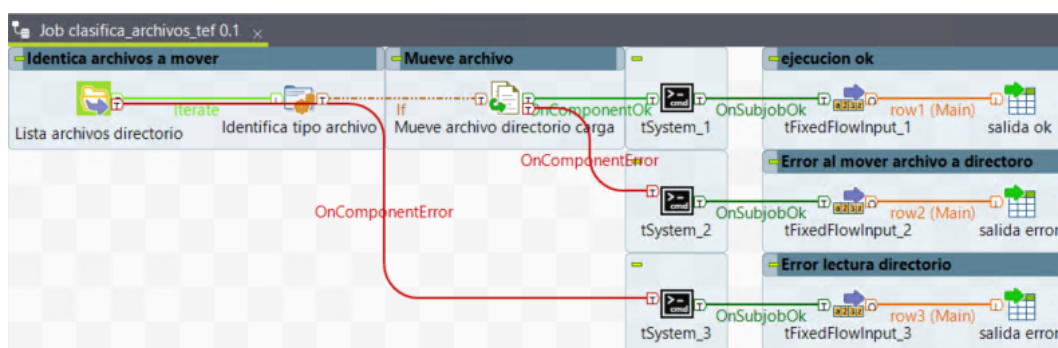


Figura 22: *Job* que clasifica los archivos de transferencia según RT, NRT o ABR

Luego de clasificados, se ejecuta primero el *job* que carga las transferencias ABR, como se ilustra en la Figura 23.

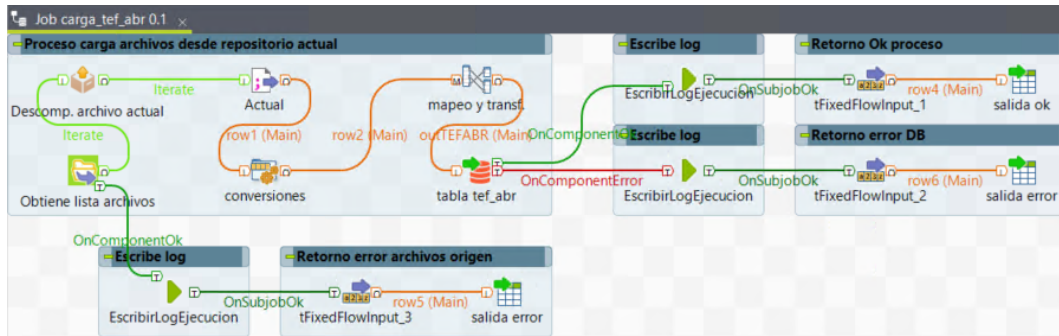


Figura 23: *Job* que carga transferencias ABR

Tras concluir la carga de TEF ABR, el flujo procede con la carga de las transferencias RT.

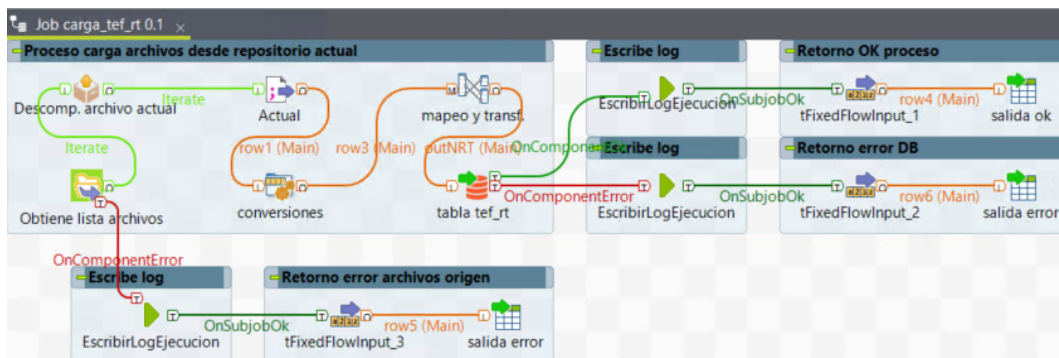


Figura 24: *Job* que carga transferencias RT

Y finalmente, para concluir, se ejecuta el *job* que carga de transferencias NRT, ilustrado en la Figura 25.

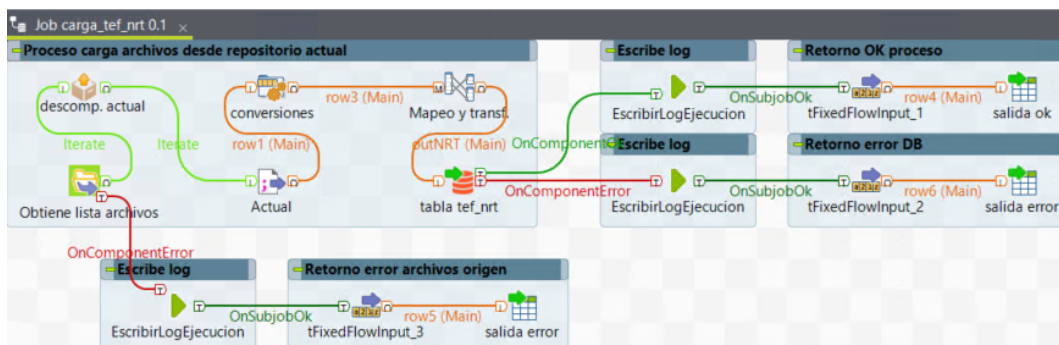


Figura 25: *Job* que carga transferencias NRT

Cabe destacar que los tres *jobs* que ejecuta el orquestador, fueron creados siguiendo la plantilla base.

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job* orquestador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

4.1.2. Carga Información no financiera

Al igual que para los *jobs* de carga de transacciones financieras, se optó por crear una réplica de las tablas tal cual fueron entregadas por operaciones, procurando que los campos clave mantuvieran los formatos adecuados a nivel de base de datos. Por lo anterior, el esquema de entrada y salida de los archivos son, en su mayoría, iguales, salvo algunas transformaciones.

- **Cargador VU:** El archivo recibido para este caso, contiene las notificaciones VU que llegan al motor de prevención de fraude. Para su carga, se crearon dos *jobs*, uno de ellos corresponde al *job* orquestador, ilustrado en la Figura 26, que hace el llamado de los *job* auxiliares y fundamentales para la carga.



Figura 26: *Job* orquestador que carga notificaciones VU

El *job* orquestador llama al *job* que carga la notificación VU, como se ilustra en la Figura 27.

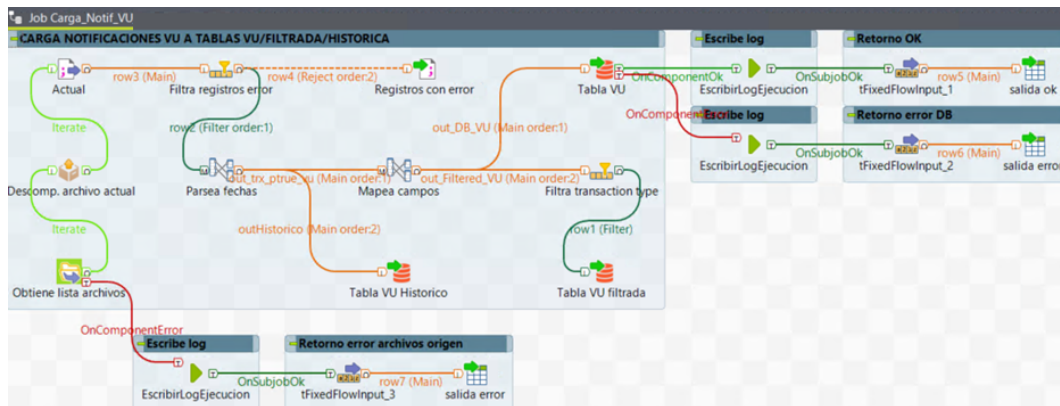


Figura 27: Job que carga notificaciones VU a tablas definitivas

Una vez concluida la carga, el *job* orquestador mueve los registros cargados, dejando el directorio listo en espera de nuevos registros.

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job* orquestador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

- **Cargador ABR:** El archivo recibido para este caso, contiene las notificaciones ABR que llegan al motor de prevención de fraude. Para su carga, se crearon dos *jobs*, uno de ellos corresponde al *job* orquestador, ilustrado en la Figura 28, que hace el llamado de los *jobs* auxiliares y fundamentales para la carga.

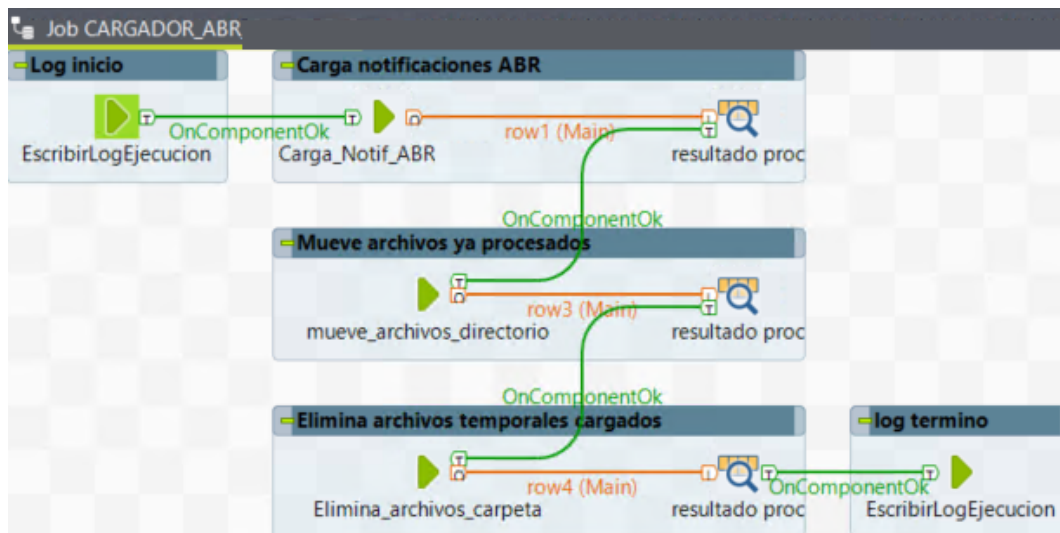


Figura 28: Job que orquesta carga notificaciones ABR

El *job* orquestador llama al *job* que carga la notificación ABR, como se ilustra en la Figura 29.

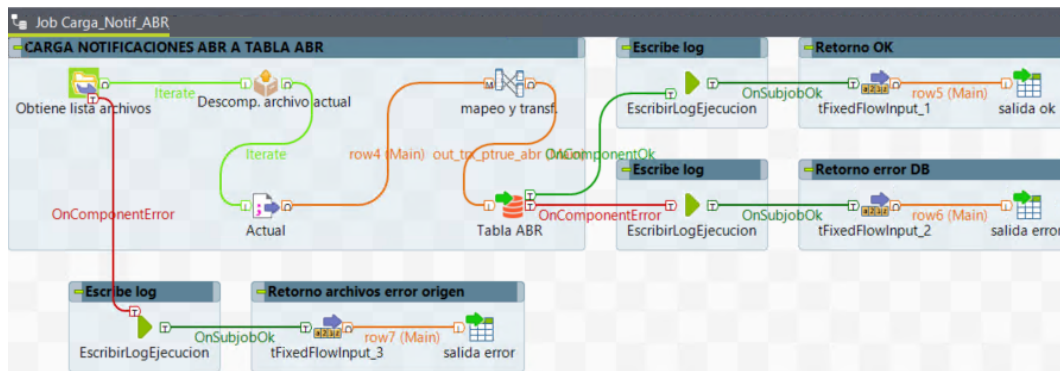


Figura 29: Job que carga notificaciones ABR a tabla definitiva

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job* orquestador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

- **Acciones automáticas:** Para la carga de las acciones automáticas se crearon dos *jobs*, uno de ellos corresponde al *job* orquestador, ilustrado en la Figura 30 , que hace el llamado de los *jobs* auxiliares y fundamentales para la carga.



Figura 30: Job que orquesta carga de las acciones automáticas

El *job* orquestador llama al *job* que carga las acciones automáticas, como se ilustra en la Figura 31.

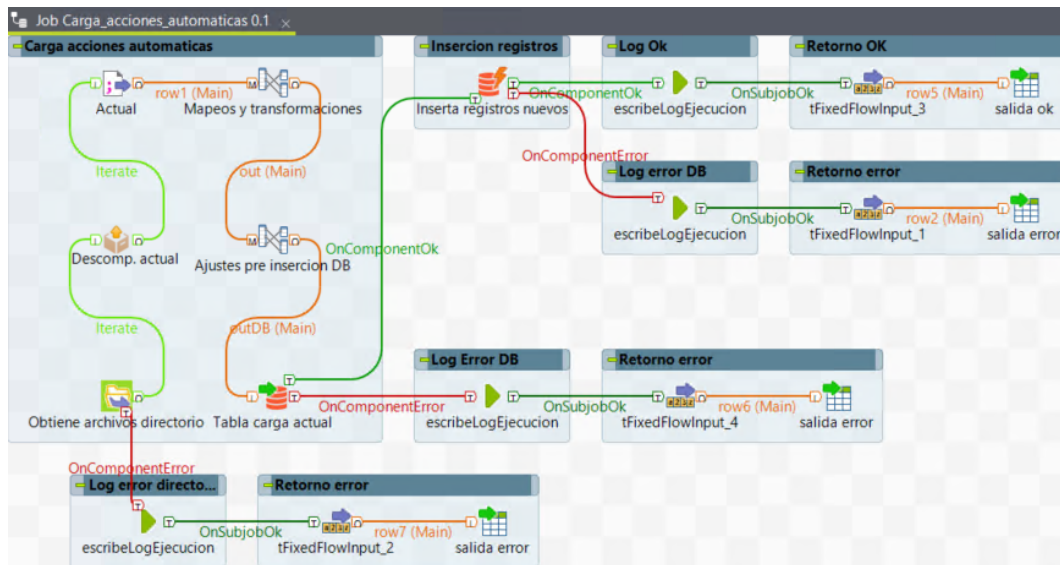


Figura 31: *Job* que carga las acciones automáticas por hora en las tablas definitivas

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job* orquestador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

- **Fraude marcado** Para la carga del fraude marcado, se crearon tres *jobs*, uno de ellos corresponde al *job* orquestador, que hace el llamado de los *jobs* auxiliares y fundamentales para la carga.

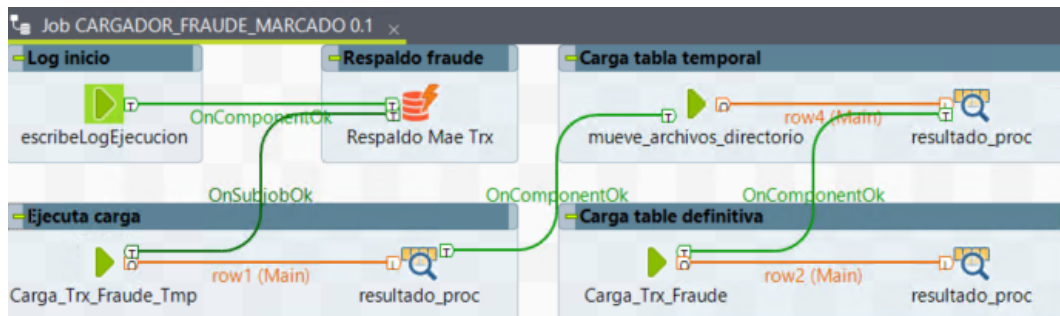


Figura 32: *Job* que orquesta carga de las transacciones marcadas como fraude

El *job* orquestador llama a un segundo *job*, ilustrado en la Figura 33, que carga en una tabla temporal los registros, para luego ser procesados y cargados en las tablas definitivas.

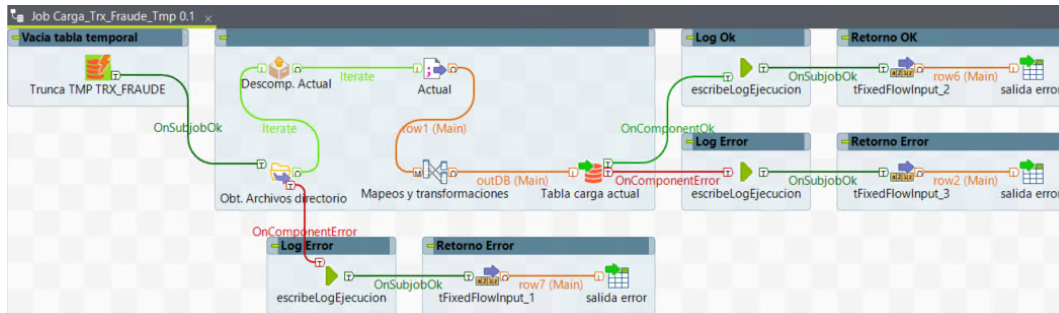


Figura 33: *Job* que carga el fraude marcado en una tabla temporal de transición

Luego de cargado en la tabla temporal, se procede a cargar las tablas definitivas con la información nueva, lo que se realiza con el *job* ilustrado en la Figura 34.

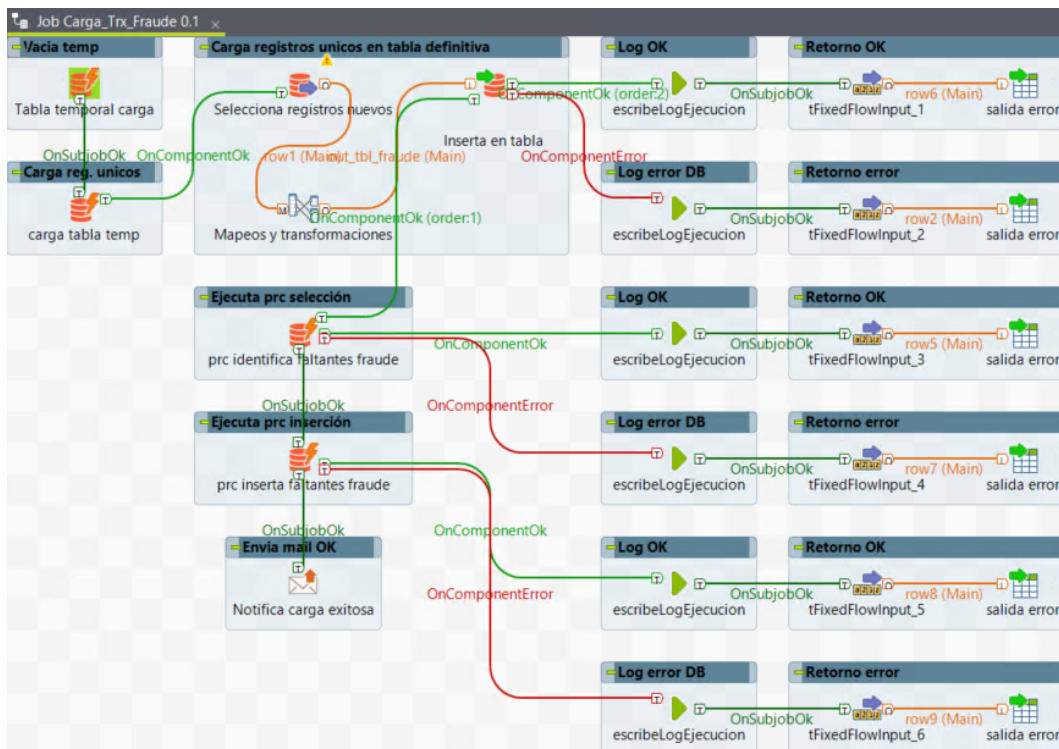


Figura 34: *Job* que realiza la carga en las tablas definitivas

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job* orquestador, como también la validación del flujo con error para la ruta origen, ejecutando el *job* sin archivos, obteniéndose el error controlado esperado.

4.1.3. Carga Información demográfica clientes

- **Información personal:** La información personal de los clientes se obtiene a partir de otra base de datos interna, con accesos restringidos, y a fin de facilitar los accesos y cruces del equipo de prevención de fraude, se crea una réplica de algunos datos en la base interna de fraude mediante el siguiente *job*.

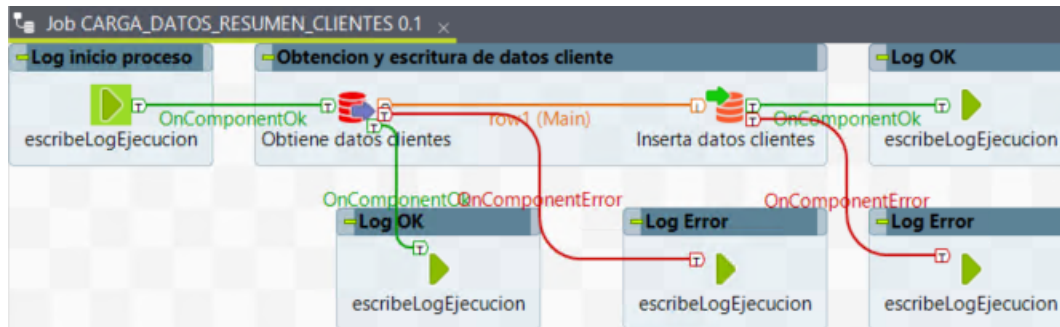


Figura 35: *Job* que carga la información personal de los clientes

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job*, como también la validación del flujo con error para la base de datos origen, ejecutando el *job* forzando un error a nivel de conexión, obteniéndose el error controlado esperado.

- **Información de productos:** La información de los productos contratados por los clientes se obtiene a partir de otra base de datos interna, con accesos restringidos, y a fin de facilitar los accesos y cruces del equipo de prevención de fraude, se crea una réplica de algunos datos en la base interna de fraude mediante el siguiente *job*.



Figura 36: *Job* que carga la información resumen de los productos de los clientes

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job*, como también la validación del flujo con error para la base de datos origen, ejecutando el *job* forzando un error a nivel de conexión, obteniéndose el error controlado esperado.

4.1.4. Carga Información complementaria

- **Base de envíos SMS** La información de los SMS enviados y su estatus respecto al envío se obtiene a partir de la información entregada por el proveedor externo, quien entrega diariamente un reporte con esta información, dicho reporte es depositado automáticamente en una ruta del servidor interno del equipo de prevención de fraude, y es cargado automáticamente en la base de datos alojada en el mismo servidor diariamente mediante el *job* ilustrado en la Figura 37.

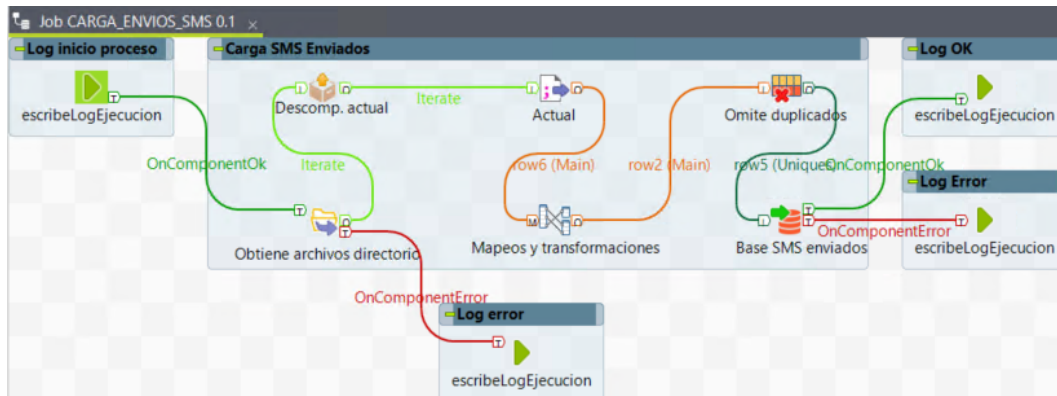


Figura 37: *job* que carga la información resumen de los productos de los clientes

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job*, como también la validación del flujo con error para la base de datos origen, ejecutando el *job* forzando un error respecto a los archivos depositados en la ruta compartida, obteniéndose el error controlado esperado.

- **Base respuestas SMS:** La información de las respuestas SMS recibidas de parte de los clientes se obtiene a partir de una API Rest proporcionada por el proveedor externo, la que es consumida en múltiples veces durante el día mediante el *job* ilustrado en la Figura 38.

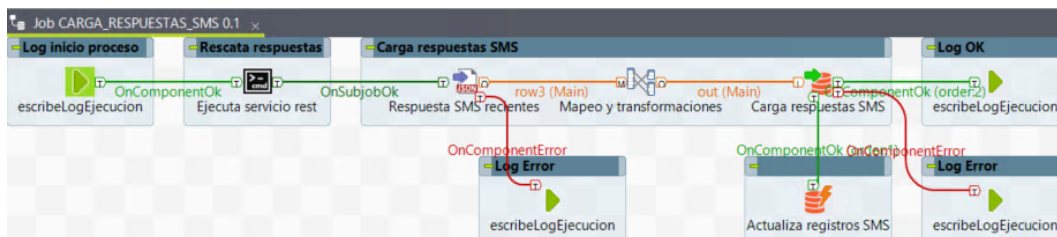


Figura 38: *Job* que carga la información resumen de los productos de los clientes

Respecto a las pruebas unitarias, se validó la correcta ejecución del *job*, como también la validación del flujo con error para la base de datos origen, ejecutando

el *job* forzando un error respecto a la lectura del archivo JSON de respuesta de la API, obteniéndose el error controlado esperado.

4.2. Etapa 2: Reportes y tableros de control

4.2.1. Reporte resumen comercios

Para apoyar el monitoreo de la gestión oportuna de posibles comercios de riesgo, se creó la vista resumen comercios, que resume cada uno de los comercios en los cuales se realizaron transacciones o intentos de compras. Los factores más relevantes para este análisis son la antigüedad del comercio, la cantidad de tarjetas que operan en este, la tasa de aprobación por cliente/monto, si tiene fraude confirmado, y sus respectivos puntos base, además de otros factores útiles para este análisis.

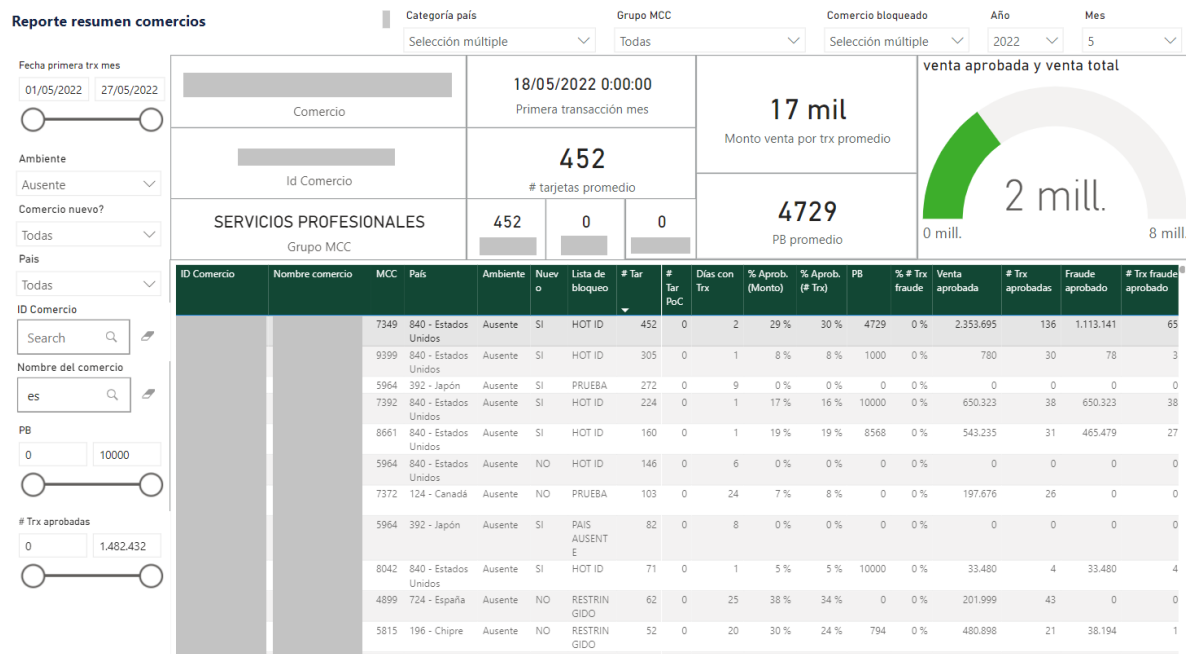


Figura 39: Vista reporte resumen comercios

Respecto a las pruebas unitarias, las visualizaciones no presentan errores, los indicadores son consistentes en volumen y forma, y se publicó correctamente.

4.2.2. Reporte calidad transaccional

Para identificar oportunamente mermas en la calidad transaccional, se creó la vista de calidad transaccional, que entrega un resumen de las últimas horas de las transacciones

que llegan al motor de prevención de fraude, lo que permite identificar desviaciones y por consiguiente, activar las contingencias y medidas de respuesta pertinentes.

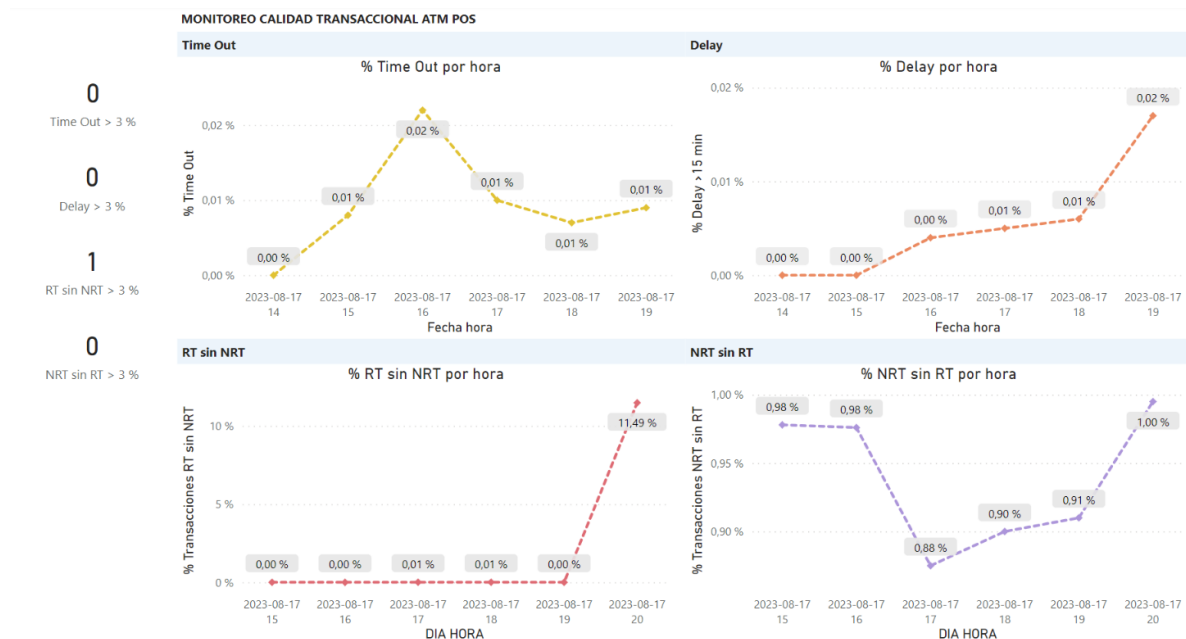


Figura 40: Vista calidad transaccional ATM POS

Respecto a las pruebas unitarias, las visualizaciones no presentan errores, los indicadores son consistentes en volumen y forma, y se publico correctamente.

4.2.3. Reporte monitoreo de cargas

Para poder validar que los procesos de carga se ejecutan correctamente, se crearon varias vistas de monitoreo de cargas, que nos permiten identificar si hay periodos faltantes y si el flujo se mantiene en los umbrales esperados. Es importante mencionar que el flujo transaccional depende de varios factores, pero hay tendencias claras de peaks y umbrales a considerar (hay menos transacciones en horario nocturno, como también hay un alza de transacciones los últimos días de cada mes, etc.), por lo que en las vistas se consideró dejar ciertos umbrales de flujo esperado mínimo en ciertos horarios.

Vista flujo transaccional no financiero

Corresponde a la vista del flujo financiero, con sus respectivos umbrales por hora. Dado que solo se cuenta con las transacciones ATM POS por hora, solo ese tipo de transacción incluye una gráfica de las últimas horas, mientras que las demás gráficas son para los últimos días, tal como se observa en la Figura 41.

CARGA AUTOMÁTICA DE DATOS PARA ANÁLISIS Y PREVENCIÓN DE FRAUDE: CASO PRÁCTICO APLICADO A LA DETECCIÓN DE FRAUDE EN COMERCIOS

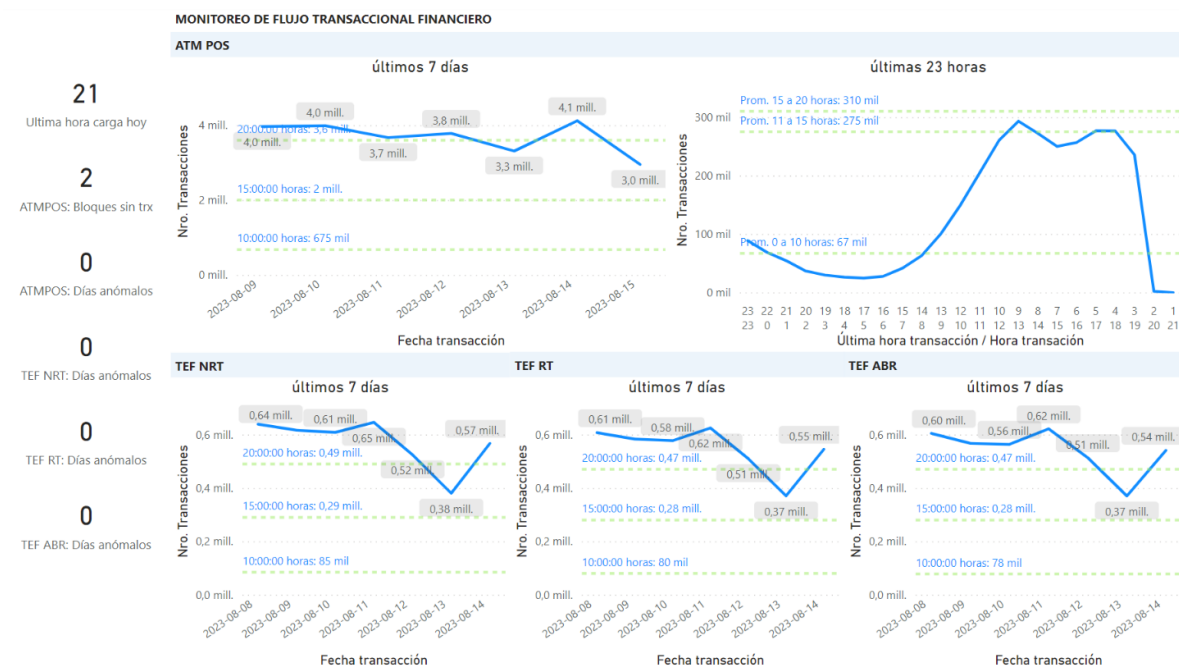


Figura 41: Vista transacciones financieras del reporte de monitoreo de cargas

Vista flujo transaccional no financiero

Corresponde a la vista del flujo no financiero, con sus respectivos umbrales por hora. Dado que solo se cuenta con las notificaciones VU por hora, solo ese tipo de transacción incluye una gráfica de las últimas horas, mientras que las demás gráficas son para los últimos días, tal como se observa en la Figura 42.

CARGA AUTOMÁTICA DE DATOS PARA ANÁLISIS Y PREVENCIÓN DE FRAUDE: CASO PRÁCTICO APLICADO A LA DETECCIÓN DE FRAUDE EN COMERCIOS

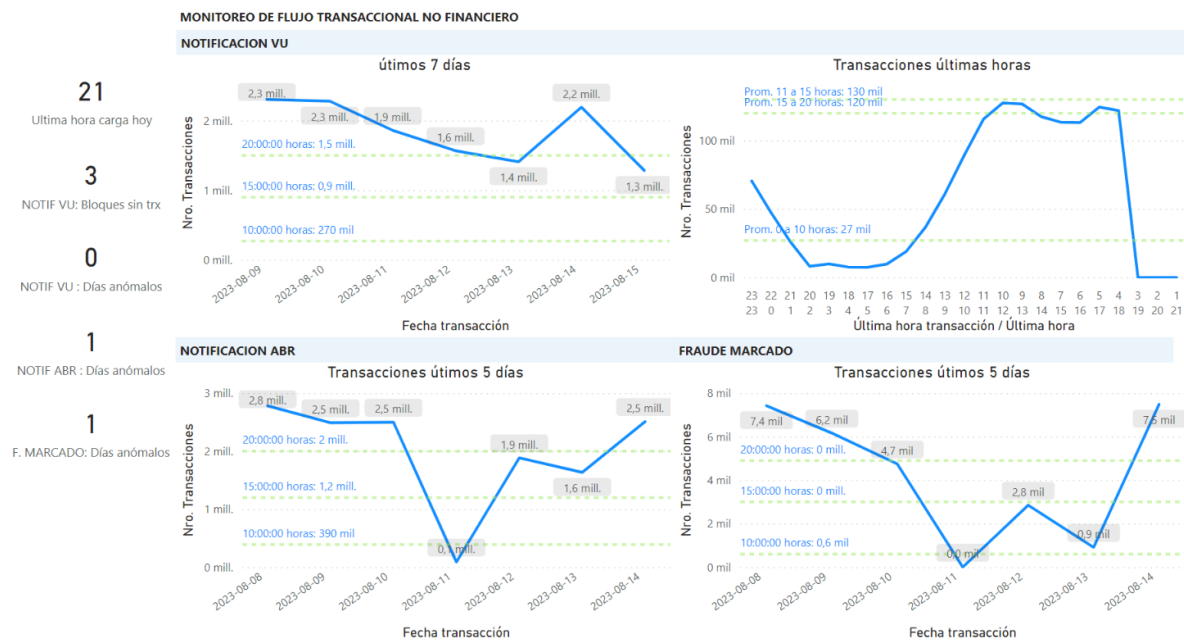


Figura 42: Vista transacciones no financieras del reporte de monitoreo de cargas

Vista flujo transaccional acciones automáticas

Corresponde a la vista del flujo de acciones automaticas del motor de prevencion de fraude, con sus respectivos umbrales por hora, tal como se observa en la Figura 43.

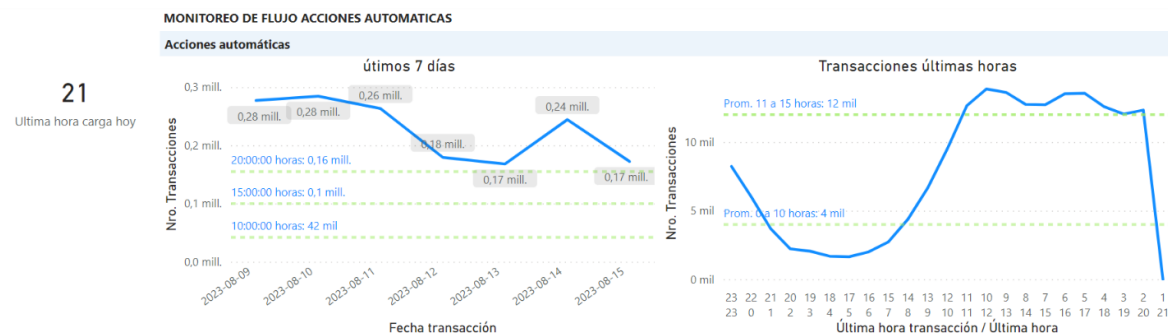


Figura 43: Vista acciones automáticas del reporte de monitoreo de cargas

Respecto a las pruebas unitarias, las visualizaciones no presentan errores, los indicadores son consistentes en volumen y forma, y fueron publicados correctamente.

CONCLUSIONES

Sobre el trabajo hecho

Todo el desarrollo del proyecto fue realizado enfocado en el objetivo principal, donde la carga y acceso a los datos fue el eje fundamental, al ser la materia prima para obtener información útil para la definición de estrategias efectivas de prevención de fraude. En ese sentido, en la etapa 1 del desarrollo, correspondiente a los procesos de carga, fue el trabajo principal para lograr el objetivo propuesto, sin embargo, durante el desarrollo de las cargas de datos, surgieron algunas problemáticas que no fueron contempladas inicialmente y que deben ser consideradas en desarrollos futuros, estas son:

- **Volumen de carga:** Se debe considerar la cantidad de registros a cargar en cada ejecución, dado que afecta los tiempos de escritura, por lo que el tamaño de los archivos en cuanto al número de registros y estimaciones de crecimiento futuro son cosas a considerar.
- **Capacidad de almacenamiento:** Se debe considerar el espacio de almacenamiento futuro. En algún momento se tuvieron problemas de almacenamiento, por lo que se tuvo que solicitar un aumento de la capacidad de almacenamiento en el servidor donde esta alojada la base de datos, y se tuvieron que realizar algunas purgas de información antigua.
- **Capacidad de procesamiento:** Se debe evaluar la capacidad de procesamiento del servidor, y si comparte recursos y uso con otros procesos. Como el volumen de datos ha ido en aumento, y el servidor no es de uso exclusivo del motor de base de datos, fue necesario solicitar un aumento en la capacidad de procesamiento del servidor.
- **Tiempos de acceso:** Se debe considerar el tiempo que tarda en ejecutarse las consultas SQL en la base de datos. Como en algunos casos los cruces de información y análisis presentaban tiempos muy altos (en ocasiones de más de 30 minutos), se optó por la creación de índices sobre campos de consulta habitual, sin abusar de estos mismos puesto que si bien los índices mejoran los tiempos de acceso a datos, también impactan en el espacio de almacenamiento y en los tiempos de escritura de nuevos registros.

Pese a las dificultades expuestas, se pudo concluir exitosamente las cargas de datos, y por consiguiente el objetivo principal del proyecto, sorteando adecuadamente las complicaciones presentadas como resultado de aplicar principios fundamentales de la metodología ágil de XP, como son el diseño simple y la refactorización.

Acerca de los objetivos específicos

Como parte del proyecto, además de contar con procesos de carga de datos, se busco cumplir objetivos específicos siguientes:

1. Contar con un procedimiento claro y conciso para la integración de nuevas fuentes de datos basado en el repositorio definido, para facilitar las cargas futuras de información.
2. Controlar que las cargas de actualización se efectúen correctamente y lleven a una base de datos actualizada.
3. Mejorar el indicador de puntos base para comercios fraudulentos tras el uso periódico de los reportes para la detección de nuevos comercios fraudulentos.

Cada uno de los objetivos antes descritos fue abordado en el diseño y solución del proyecto, como se explica a continuación:

Para cumplir el objetivo 1, se elaboró la plantilla base de cargas, cuyo uso esta descrito en la sección 3.5.5. Esta fue utilizada para la implementación de todos los *jobs* de cargas, obteniéndose con su uso la carga exitosa de los datos, por lo que del resultado de su aplicación en la construcción de los *jobs*, se concluye que se logró contar con un procedimiento claro y conciso para la integración de nuevas fuentes de datos.

Para lograr el objetivo 2, se desarrollo el reporte de monitoreo de cargas, descrito en la sección 4.2.3, el que mediante su revisión periódica, permitió identificar anomalías en las bases y con ello, actuar oportunamente ante errores en el proceso de carga, como también en los casos en que los datos no hayan sido depositados en las rutas respectivas por algún error por el lado de operaciones. Con su uso diario, se logró implementar los controles mínimos requeridos en lo que respecta a la carga efectiva de datos, cumpliéndose el objetivo propuesto.

Finalmente, para lograr el objetivo 3, se desarrollo el reporte resumen comercios, descrito en la sección 4.2.1. Este reporte fue creado mediante iteraciones continuas de mejora con el cliente interno de la organización, contemplando vistas por segmentaciones clave, y un resumen transaccional para cada comercio que opera con las tarjetas de los clientes de la organización. En esta linea, es importante mencionar que, de su revisión diaria, se logró identificar rangos de tarjetas que eran objeto de *Card Testing*, y adelantarse por tanto, al bloqueo del comercio y de las tarjetas expuestas, como también, en los casos en que se presentará fraude aprobado, realizar la gestión anticipada para mejorar la experiencia del cliente, realizando la devolución de los fondos defraudados, el bloqueo del producto y la notificación al cliente, evitando que se siga materializando fraude con dicha tarjeta. Todas las acciones antes descritas, producto del uso del reporte, inciden directamente en el indicador de puntos base, por lo que su uso recurrente ha permitido una mejora del indicador para el negocio.

Impacto en la organización

El equipo de prevención de fraude paso de tener la información para análisis distribuida entre distintas fuentes (bases de datos, archivos, apis), a contar con una base de datos centralizada de información, lo que impactó directamente en los análisis de datos, la definición de la estrategia de prevención, la identificación de patrones de alto riesgo y la mejora de los indicadores clave del área. El impacto de esto se traduce en una estrategia de prevención más efectiva (mejora en falso positivo), lo que significa una mejor tasa de rechazos, y por consiguiente, una mejora en los puntos base de la organización.

Trabajo futuro

Durante el desarrollo del proyecto, se pudieron detectar varias oportunidades de mejora, y por consiguiente, traducirse en mejoras en la estrategia de prevención de fraude y los indicadores clave del área, las que se describen a continuación.

Extensión 1 : Incorporación de alertas mail/sms

Una de las falencias de los controles de carga y calidad transaccional actualmente implementados, es que debe haber alguien que los revise diariamente, por lo que un sistema de alertas y notificaciones vía mail o sms, por ejemplo, permitirá actual con mayor premura, y elimina el riesgo de que se pase alguna anomalía porque no se reviso el reporte.

Extensión 2 : Incorporación de bases por hora para todos los tipos de transacciones

Contar con información actualizada lo antes posible es crucial en en análisis y prevención de fraude, por lo que no tener todas las transacciones por hora, puede traducirse a no detectar oportunamente alguna casuística de fraude, y por tanto, materializarse como perdida para el negocio, por lo que se debe apuntar, en primera instancia, en contar con cargas por hora de todos los reportes transaccionales (financieros y no financieras), y a futuro, ver la opción de reducir el periodo de actualización, aspirando a contar con la información casi en linea.

Extensión 3 : Migración a tecnologías *cloud*

Siguiendo la línea de las tendencias de transformación digital de la industria, y dado que los análisis de fraude requieren procesar grandes volúmenes de información, contemplar la migración de la información a tecnologías cloud, como Google Cloud Platform, significaría una mejora significativa en los tiempos de análisis y obtención de resultados, eliminando los problemas de mantención de servidores y base de datos, al externalizar este tipo de servicio.

Contenidos fundamentales de lo aprendido en la universidad

Para el desarrollo del presente trabajo se requirió de una serie de competencias que permitieron abordar de manera efectiva los desafíos técnicos y de planificación, las que fueron adquiridas en diferentes ramos realizados en la UTFSM, dentro de los cuales destacan los siguientes:

- **Base de datos:** Por su aporte en la comprensión de base de datos y su manipulación mediante el uso de SQL, dado que tanto en los procesos de carga como en los reportes implementados, el diseño de las consultas y su optimización fue crucial para obtener resultados favorables.
- **Lenguajes de programación:** Como Talend es una herramienta basada en el IDE Eclipse, en el que los proyectos son implementados usando el lenguaje JAVA, la comprensión del lenguaje y el desarrollo orientado a objetos fue fundamental para la implementación de los procesos de cargas de datos.
- **Taller de gestión de proyectos informáticos:** Por proporcionar una base sólida respecto a la gestión de proyectos informáticos, ayudando a definir un marco de trabajo y establecer metas realistas para el proyecto.
- **Ingeniería de Software:** Al proporcionar un marco de trabajo y la primera experiencia de desarrollo de proyectos informáticos utilizando metodologías de desarrollo de software.

REFERENCIAS BIBLIOGRÁFICAS

- Anwer, F., Aftab, S., Waheed, U., y Muhammad, S. S. (2017). Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2):1–10.
- Azeroual, O. y Theel, H. (2019). The effects of using business intelligence systems on an excellence management and decision-making process by start-up companies: A case study. *arXiv preprint arXiv:1901.10555*.
- Maurer, F. y Martel, S. (2002). Extreme programming. rapid development for web-based applications. *IEEE Internet computing*, 6(1):86–90.
- Microsoft (2019 (visitada el 17 de noviembre, 2019)). Precios de power bi. <https://powerbi.microsoft.com/es-es/pricing/>.
- PCISSC (2023 (visitada el 21 de agosto, 2023)). Pci security standards council. <https://www.pcisecuritystandards.org/>.
- Quinn, K. (2009). How business intelligence should work. *Information Builders White Papers*.
- Ranjan, J. (2005). Business intelligence: concepts, components, techniques and benefits, <raspoloživo na: [http://www.jatit.org/volumes/research-papers.Vol9No1/9Vol9No1.pdf](http://www.jatit.org/volumes/research-papers/Vol9No1/9Vol9No1.pdf)>, [pristupljeno: 18.8. 2017].
- Software, T. (2019 (visitada el 17 de noviembre, 2019)). Precios de tableau. <https://www.tableau.com/es-es/pricing/>.
- Stackowiak, R., Rayman, J., y Greenwald, R. (2007). *Oracle data warehousing & business intelligence SO*. John Wiley & Sons.
- Subtel (2020 (visitada el 18 de noviembre, 2022)). Subtel en alerta por aumento de estafas de suplantacion de tarjeta sim. <https://www.subtel.gob.cl/subtel-en-alerta-por-aumento-de-estafas-de-suplantacion-de-tarjeta-sim/>.
- Sutherland, J. y Schwaber, K. (2012). The scrum papers: Nut, bolts, and origins of an agile framework. 224p. 2011.
- Tableau, S. (2022 (visitada el 21 de noviembre, 2022)). What is data visualization? definition, examples, and learning resources. <https://www.tableau.com/learn/articles/data-visualization>.
- VISA (2022 (visitada el 17 de octubre, 2022)). Prevención de fraude digital. <https://www.visa.cl/soporte/consumidores/seguridad/prevencion-fraude-digital.html>.
- Wilke, C. O. (2019). Fundamentals of data visualization.