



UNIVERSIDAD NACIONAL DE LA PLATA
Facultad de Informática

**Minería de Procesos en la
Ingeniería de Software**

Análisis del proceso de distribución de piezas postales

Autor: Lic. Víctor Martínez

Director: Dra. Laura Lanzarini

Director: Dr. Franco Ronchetti

**Trabajo Final presentado para obtener el grado de
Magister en Ingeniería de Software**

Junio, 2023

*A Lauti y Santi
El motivo de todo.*

Resumen

En los últimos años la Minería de Procesos tuvo un gran crecimiento debido a la gran disponibilidad de datos y los avances en algoritmos y herramientas. También se incrementó su utilidad en la Ingeniería de Software, ya que proporciona información valiosa sobre la ejecución real de los procesos. Esto permite a las organizaciones analizar y generar productos de software orientados a mejorar el servicio al cliente. Con la Minería de Procesos, es posible identificar cuellos de botella, ineficiencias y desvíos de los comportamientos esperados; lo que lleva a una optimización de los procesos y a una mejor asignación de recursos. Al utilizar las técnicas de Minería de Procesos, los equipos de Ingeniería de Software pueden comprender a fondo los procesos, mejorar la toma de decisiones y entregar productos de software de mayor calidad.

En esta tesis de maestría se aplicó la Minería de Procesos a la distribución postal en la República Argentina con el objetivo de identificar oportunidades de mejora en el proceso, contribuyendo así a la reducción de conflictos operativos y a la mejora en la calidad del servicio.

El estudio se inició con una revisión bibliográfica del material disponible, seguido de la aplicación de técnicas de Minería de Procesos a un extenso conjunto de datos relacionados con la distribución postal. Esto permitió obtener información valiosa sobre el rendimiento y la eficiencia del sistema de distribución.

Los resultados de estas investigaciones fueron difundidos a través de los siguientes artículos:

- Martinez, V., Lanzarini, L., Ronchetti, F. (2021). Process Mining Applied to Postal Distribution. Pertenece al libro: Memorias del XXVII Congreso Argentino de Ciencias de la Computación. RedUNCI. ISBN: 978-987-633-574-4. Págs: 271-280.
- Martinez, V., Lanzarini, L., Ronchetti, F. (2022). Distribution Analysis of Postal Mail in Argentina Using Process Mining. In: Pesado, P., Gil, G. (eds) Computer Science – CACIC 2021. CACIC 2021. Communications in Computer and Information Science, vol 1584. Springer, Cham.

El nuevo conocimiento adquirido a partir del modelo generado fue utilizado para desarrollar un prototipo de software que automatiza la aplicación de técnicas de Minería de Procesos a la distribución postal y genera alertas ante posibles desvíos operativos. De esta forma, las incidencias descubiertas pueden gestionarse de manera rápida y eficiente con una interfaz moderna y amigable para el usuario.

Los resultados obtenidos demuestran claramente que la aplicación de la Minería de Procesos en la distribución postal brindan un aporte significativo a la hora de modelizar el proceso subyacente. En el caso de estudio propuesto en esta tesis, han permitido la detección y gestión temprana de incidencias operativas en el proceso de distribución de piezas postales, proporcionando herramientas novedosas que permiten tomar acciones correctivas y/o predictivas mejorando el servicio a los clientes de manera efectiva y oportuna.

Palabras Clave: Minería de Procesos, Minería de Datos, Distribución Postal, Proceso Postal, Gestión de Procesos de Negocio.

Abstract

In recent years, Process Mining has experienced tremendous growth due to the great availability of data and advances in algorithms and tools. Its usefulness in Software Engineering has also increased, since it provides valuable information on the actual execution of processes. This allows organizations to analyze and generate software products aimed at improving customer service. With Process Mining, it is possible to identify bottlenecks, inefficiencies, and deviations from expected behaviors; which leads to an optimization of processes and a better estimation of resources. By using Process Mining techniques, Software Engineering teams can gain a deeper understanding of processes, improve decision making, and deliver higher-quality software products.

In this master's thesis, Process Mining was applied to postal distribution in the Republic of Argentina with the objective of identifying opportunities for process improvement, thereby contributing to the reduction of operational conflicts and enhancing service quality.

The study commenced with an extensive literature review, followed by the application of Process Mining techniques to a large dataset pertaining to postal distribution. This facilitated the extraction of valuable insights into the performance and efficiency of the distribution system.

The results of these investigations were disseminated through the following articles:

- Martinez, V., Lanzarini, L., Ronchetti, F. (2021). Process Mining Applied to Postal Distribution. From the book: Memorias del XXVII Congreso Argentino de Ciencias de la Computación. RedUNCI. ISBN: 978-987-633-574-4. Pages: 271-280.
- Martinez, V., Lanzarini, L., Ronchetti, F. (2022). Distribution Analysis of Postal Mail in Argentina Using Process Mining. In: Pesado, P., Gil, G. (eds) Computer Science – CACIC 2021. CACIC 2021. Communications in Computer and Information Science, vol 1584. Springer, Cham.

The new knowledge acquired from the generated model was used to develop a software prototype that automates the application of Process Mining techniques to the distribution postal service and generates alerts in the event of possible operational deviations. In this way, the incidents discovered can be managed quickly and efficiently with a modern and user-friendly interface.

The results obtained clearly demonstrate that the application of Process Mining in postal distribution provide a significant contribution when it comes to modeling the underlying process. In the case study proposed in this thesis, they have allowed the detection and management of operational incidents in the distribution process of metal parts, providing innovative tools that allow corrective and/or predictive actions to be taken, improving the serve customers in an effective and timely manner.

Keywords: Process Mining, Data mining, Postal Distribution, Postal Processes, Business process management.

Agradecimientos

A mi familia, por su esfuerzo y apoyo incondicional, pilar donde se sustentan todos los resultados.

Acrónimos

BI:	Business Intelligence - Inteligencia Empresarial
BPM:	Business Process Management - Gestión de Procesos de Negocio
BPMN:	Business Process Model and Notation - Modelado y Notación de Procesos de Negocio
CRM:	Customer Relationship Management - Gestión de Relaciones con los Clientes
ERP:	Enterprise Resource Planning - Planificación de Recursos Empresariales
GIT:	Sistema de control de versiones distribuido
IDE:	Integrated Development Environment - Entorno de Desarrollo Integrado
IEEE:	Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y electrónicos
TFPM:	Task Force Process Mining - Grupo de Trabajo de Minería de Procesos
TMS:	Transportation Management System - Sistema de Gestión de Transporte
WMS:	Warehouse Management System - Sistema de Gestión de Almacenes
XES:	eXtensible Event Stream - Flujo de Eventos Extensible
XML:	eXtensible Markup Language - Lenguaje de Marcado Extensible

Glosario

Business Intelligence:	Inteligencia Empresarial. Es la transformación de datos en información y conocimiento para optimizar la toma de decisiones en procesos de negocio, generalmente se muestran los resultados en tiempo real.
Business Process Management	Gestión de Procesos de Negocio. Es una metodología para diseñar, ejecutar, analizar y mejorar procesos de negocio. Permite definir claramente todos los procesos de distintos modelos de negocio.
Business Process Model and Notation	Modelado y Notación de Procesos de Negocio. Es una metodología estándar para documentar procesos de negocio de forma gráfica. Actualmente se encuentra la especificación 2.0
Customer Relationship Management	Gestión de Relaciones con los Clientes. Es un conjunto de herramientas para administrar las relaciones con los clientes (registro, ventas, marketing, reclamos y todos los puntos de contacto), generalmente se agrupan bajo un conjunto de aplicaciones denominados CRM.
Data Analysis:	Ánalisis de Datos. Es la ciencia de analizar datos con el objetivo de sacar conclusiones que permitan tomar decisiones o expandir el conocimiento.
Data Mining:	Minería de Datos. Técnicas para descubrir y analizar información en grandes volúmenes de datos.
Enterprise Resource Planning	Planificación de Recursos Empresariales. Es un conjunto de herramientas que centraliza todos los procesos de negocio necesarios para operar una empresa (finanzas, RRHH, facturación, manufactura, cadena de suministros, compras y otros). Generalmente se integran en una única aplicación.
Data Science:	Ciencia de Datos. Disciplina que transforma los datos en conocimiento, analizando grandes volúmenes de información.
Institute of Electrical and Electronics Engineers	Instituto de Ingenieros Eléctricos y electrónicos. Es una asociación profesional internacional sin fines de lucro dedicada a generar estándares, entre otras cosas, para las nuevas tecnologías.
Integrated Development Environment	Entorno de Desarrollo Integrado. Es una herramienta que permite desarrollar y probar aplicaciones de software.

Log:	Registro o historial de log. Es una grabación en forma secuencial en un archivo de texto o base de datos que registra los eventos o acciones de un proceso en particular.
Machine Learning:	Aprendizaje Automático. Es una técnica de análisis de datos que enseña a las computadoras a aprender de la experiencia, de forma análoga a como aprenden las personas o los animales.
Process Mining:	Minería de Procesos. Técnicas para descubrir y analizar un modelo de negocio a partir de logs de eventos.
Transportation Management System:	Sistema de Gestión de Transporte. Es un sistema de gestión de la cadena de suministro y entrega de una empresa. Generalmente abarca todos los aspectos relevantes de la logística.
Task Force Process Mining:	Es un grupo de trabajo promovido por la IEEE con el objetivo de promover, investigar y desarrollar la Minería de Procesos.
Warehouse Management System:	Sistema de Gestión de Almacenes. Es un sistema que brinda soporte a las operaciones diarias de un almacén y brinda una visibilidad amplia de los recursos que dispone la empresa.
eXtensible Event Stream:	Es un formato de archivos de texto para registrar eventos, está basado en XML. Actualmente se considera el formato estándar para Minería de Procesos.
eXtensible Markup Language:	Es un formato de archivo de texto que utiliza una serie de etiquetas personalizadas para describir tanto la estructura, datos y otras características del documento.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
Acrónimos	xI
Glosario	xIII
Índice de figuras	xvII
Índice de tablas	xix
Índice de listados	xxI
1. Introducción	1
1.1. Motivación	4
1.2. Objetivos	5
1.3. Trabajos Publicados	6
1.4. Estructura del documento	6
2. Contexto General	9
2.1. Proceso de Distribución Postal	9
2.2. Business Process Management (BPM)	13
2.3. BPMN	16
2.4. Minería de Procesos	19
2.5. Trabajos Relacionados	24
3. Arquitectura de la propuesta	27
3.1. Introducción	27
3.2. Producto elegido	28
3.3. Definición de la prueba	28
3.4. Software utilizado	29
4. Minería de Procesos en la distribución postal	31
4.1. Introducción	31
4.2. Extracción de los datos	32
4.3. Conversión de datos	33
4.4. Filtrado	35
4.5. Descubrimiento del proceso	37

4.6. Verificación de Conformidad	40
4.7. Problemas encontrados	43
4.8. Resultados obtenidos	50
5. Solución Propuesta	51
5.1. Introducción	51
5.2. Metodología aplicada	51
5.3. Ciclo de vida utilizado	52
5.4. Arquitectura de la solución	53
5.5. Análisis de la necesidad	56
5.6. Diseño del prototipo	59
5.7. Desarrollo	61
5.8. Testeo	76
5.9. Casos de prueba	76
5.10. Implementación	79
6. Conclusiones y futuras líneas de investigación	81
6.1. Conclusiones	81
6.2. Futuras líneas de investigación	82
Bibliografía	83
A. Publicaciones	89
B. Código Fuente	113
B.1. Módulo Extractor de Datos	113
B.2. Base de Datos	117
B.3. API de gestión de Incidencias	118
B.4. Web de administración	121
B.5. Contenedores	125
C. Herramientas utilizadas	127

Índice de figuras

2.1. Registro de entrega de OCA, Andreani y Correo Argentino	11
2.2. Proceso de Distribución Postal para el caso de estudio	13
2.3. Ciclo BPM	15
2.4. Principales componentes de BPMN	18
2.5. Ejemplo de registro de eventos en un archivo log	21
2.6. Play-In, Play-Out, Replay	22
2.7. Diferentes tipos de Minería de Procesos	24
 3.1. Minería de Procesos aplicada a la distribución postal	27
4.1. Minería de Procesos en la distribución postal	31
4.2. Datos Originales	33
4.3. Datos en formato XES	35
4.4. Filtrado de archvos XES en ProM	36
4.5. Proceso de Distribución Postal modelado en BPMN	37
4.6. Proceso descubierto con Alpha+ que deberán cumplir las trazas	39
4.7. Recorrido de las trazas: eventos más comunes en oscuro y recorridos más frecuentes en línea más gruesa	40
4.8. Visualización de ProM de trazas con exceso de movimientos y ajuste muy bajo . . .	41
4.9. Visualización de trazas según análisis de Performance	42
4.10. Trazas de dos movimientos con inactividad	43
4.11. Trazas con desvíos leves	45
4.12. Trazas con desvíos graves	46
4.13. Trazas exceso de movimientos	47
4.14. Trazas con excesivo tiempo de finalización	49
4.15. Trazas con tiempo extremadamente alto de finalización	50
 5.1. Arquitectura del prototipo desarrollado	55
5.2. Tablero Kanban con funcionalidades a desarrollar	58
5.3. Épica API de Gestión de Incidencias	59
5.4. DER del Prototipo de Minería de Procesos aplicado a la distribución postal	61
5.5. Prueba de concepto visualización del entorno de desarrollo Spider	62
5.6. Tabla donde se almacenan las trazas encontradas con ajuste bajo	65
5.7. Fragmento de capas controlador, servicio y repositorio en Spring	67
5.8. Desarrollo de la API de gestión de incidencias en IntelliJ IDEA	68
5.9. Documentación de la API con Swagger y Open API	69
5.10. Desarrollo de la web de administración en Visual Studio Code	70
5.11. Web de gestión de incidencias	71
5.12. Listado de incidencias	72
5.13. Web de gestión de incidencias - detalle de incidencia	73
5.14. Tablero de Grafana con las principales incidencias	74

5.15. Pantalla de Docker Desktop con los servicios iniciados en contenedores	75
5.16. Caso de Prueba Ingreso al Sistema	78
5.17. Caso de Prueba Detalle de incidencia	79

Índice de tablas

4.1. Detalle de trazas y eventos utilizados	37
4.2. Detalle de irregularidades encontradas	44
4.3. Irregularidades encontradas	44
5.1. Casos de Prueba realizados	77

Índice de listados

5.1.	Proceso de extracción de datos de Oracle	63
B.1.	Comando para programar ejecución de extracción de datos	113
B.2.	Código fuente del proceso de extracción de datos de Oracle	113
B.3.	Script de ejecución del módulo de Minería de Procesos	114
B.4.	Comando para programar ejecución del módulo de Minería de Procesos	114
B.5.	Código en Python del componente de Minería de Procesos	114
B.6.	Código SQL tabla trazasajustebajo	117
B.7.	Código SQL tabla trazas	118
B.8.	Código del Controlador de Incidencias	118
B.9.	Código del servicio de Incidencias	120
B.10.	Código del Repositorio de Incidencias	121
B.11.	Componente de inicio en Web de Administración	121
B.12.	Componente que consume datos de la API Java	123
B.13.	Archvo docker-compose.yml para levantar los servicios en contenedores	125

CAPÍTULO 1

Introducción

En todo proceso de negocio hay una búsqueda constante de reducir costos y mejorar el servicio a los clientes. Estas acciones contribuyen a hacer más competitivas las empresas y posicionarse mejor en el mercado. Para ello es indispensable conocer en detalle el funcionamiento del proceso de negocio, encontrar sus problemas, debilidades, cuellos de botella, posibles mejoras o nuevas oportunidades que posibiliten una mejora de servicio. Con el advenimiento de las nuevas tecnologías y la *era de la información* fueron apareciendo infinidad de nuevas técnicas, herramientas y metodologías que brindan un gran espectro de posibilidades para mejorar procesos de negocio y experiencia de los clientes.

Actualmente existe una amplia variedad de sistemas de información que dan soporte a las diferentes áreas de cada empresa ya sean propias del negocio o administrativas. Entre estas herramientas pueden encontrarse desarrollos ad hoc para cada compañía o herramientas de gestión como CRM, ERP, WMS, TMS, BI, etc.

Dada la facilidad y el bajo costo para almacenar datos en estos días, la mayoría de estos sistemas deja registro de las actividades realizadas con información para auditoría o resolución de errores en archivos de texto o bases de datos. Entre otras cosas, se registra qué tarea se realizó, cuándo se realizó, quién la hizo, si hubo algún error. Analizando estos datos puede obtenerse información acerca de errores o problemas ocurridos, cual es el proceso de negocio que se realiza, como proponer mejoras o encontrar soluciones a diversos inconvenientes operativos, cuál es el comportamiento de los clientes y sus preferencias.

Esta gran disponibilidad de información contribuye al desarrollo de nuevas metodologías que permiten analizar datos en sus diferentes formas y proporcionar valor agregado en casi cualquier área de negocio.

El análisis y estudio de grandes volúmenes de información es conocido como *Data Science* o *Ciencia de Datos*.

En [31] se define como "La Ciencia de Datos es un campo interdisciplinario que tiene como objetivo convertir datos en información con valor real. Los datos pueden ser estructurados o no estructurados, grandes o pequeños, estáticos o dinámicos. El valor obtenido puede ser en forma de predicciones, decisiones automatizadas, modelos aprendidos de los datos o cualquier forma de visualización que nos

brinde información". También se expresa que "La ciencia de datos incluye la extracción, preparación, exploración y transformación de datos. Su almacenamiento y recuperación, infraestructuras, diversos tipos de minería y aprendizaje, presentación de explicaciones y predicciones. Además en la explotación de resultados se debe tener en cuenta la ética y los aspectos sociales, legales y comerciales"^[31].

La información obtenida permite tomar decisiones o incrementar el conocimiento de algún aspecto en particular. Para ello se utilizan funciones matemáticas y estadísticas que permitan manipular y evaluar los datos.

Si bien el análisis de datos existe desde hace mucho tiempo, la denominación *Ciencia de Datos* tiene unos pocos años, en los cuales algunas metodologías han crecido mucho impulsadas por el avance tecnológico.

En general el análisis de datos se compone de varias etapas entre las que se encuentra la recopilación de requisitos, aquí es donde se define que es lo que se quiere obtener, la recopilación de datos de una o varias fuentes, la limpieza o pre procesamiento donde se sacan los datos incompletos o duplicados, y finalmente el análisis y presentación de los resultados.

Dentro del análisis de datos podemos encontrar la **Minería de Datos (Data Mining)** y la **Minería de Procesos (Process Mining)** entre otras disciplinas.

La Minería de Datos es un proceso que combina estadística, técnicas de aprendizaje automático y tecnología para producir modelos que permitan analizar grandes volúmenes de datos.

En [8] se define la Minería de Datos como el "*Proceso de descubrimiento de patrones interesantes y conocimiento a partir de grandes volúmenes de datos*". También se expresa que la Minería de Datos satisface la necesidad actual de datos efectivos, flexibles y escalables.

El ciclo de vida de un proceso de Data Mining se define en [35] como una secuencia de tareas que pueden realizarse en una o varias iteraciones y que se compone de las siguientes etapas:

1. **Entendimiento del negocio:** En esta etapa se definen los objetivos de negocio y los requerimientos. Se decide qué tipo de análisis se puede aplicar para cumplirlos y se determina qué tipo de información se recolectará.
2. **Entendimiento de los datos:** En esta etapa se define y se estudia un set inicial de datos para evaluar si es posible realizar el análisis. Si la calidad de los datos es baja puede ser necesario recolectar nuevos datos con un criterio más riguroso. Con la comprensión de los datos es probable que haya que reconsiderar los objetivos de negocio y replantearse los requerimientos.
3. **Preparación de los datos:** Involucra el pre procesado de los datos iniciales para que los algoritmos puedan generar el modelo.
4. **Modelado:** Es donde se realiza el análisis. Está fuertemente relacionado con la preparación de los datos, generalmente los resultados obtenidos por el modelado brindan nuevas perspectivas que se relacionan con la preparación de los datos generando una iteración desde esa etapa con nuevas técnicas de pre procesamiento.

5. **Evaluación:** Analiza si las descripciones inferidas de los datos tienen algún valor predictivo o simplemente reflejan irregularidades espurias. Si la etapa de evaluación muestra que el modelo es pobre puede ser probable reconsiderar el proyecto entero y volver a la etapa de entendimiento del negocio e identificar nuevos objetivos. Por otro lado si la precisión del modelo es lo suficientemente alta se procede con el despliegue.
6. **Despliegue:** Generalmente se integra el modelo en un entorno productivo en grandes sistemas de información por lo tanto el proyecto debe trasladarse a los ingenieros de software para su análisis y desarrollo (es posible que se deba realizar en un nuevo lenguaje de programación)

La Minería de Procesos es una disciplina que permite descubrir, analizar y mejorar procesos de negocio a través de los archivos de registro dejados por los sistemas de información. Para ello, utiliza técnicas capaces de modelizar dichos procesos para luego extraer información útil relacionada con su funcionamiento que ayude a la toma de decisiones. La modelización del proceso no sólo permite comprender el funcionamiento real (basado en las acciones que se ejecutan día a día), sino que facilita la detección de desvíos operativos y cuellos de botella; aspectos que redundan en una mejora del proceso agilizando su funcionamiento y reduciendo las congestiones.

En el *Manifiesto de la Minería de Procesos*[30] se define la Minería de Procesos como una disciplina de investigación relativamente joven que se ubica entre Machine Learning y el Data Mining, por una parte, y la modelización y análisis de procesos, por otra. El registro generado por los sistemas de información puede utilizarse para obtener una mejor perspectiva de los procesos actuales, por ejemplo los desvíos pueden analizarse en detalle lo que lleva a una mejora del modelo.

La Minería de Procesos provee un puente importante entre la Minería de Datos y el modelado y análisis de procesos de negocio[30]

Al trabajar directamente con datos productivos se obtiene el comportamiento real que se realiza para el negocio y el proceso completo que se está realizando, el cual en algunos casos puede diferir del que se diseñó originalmente debido a desvíos operativos que no están registrados.

La Minería de Procesos puede responder interrogantes de tipo: ¿qué ocurrió realmente?, ¿por qué ocurrió?, ¿qué podría suceder en el futuro?, ¿cómo se puede mejorar el control del proceso?, ¿cómo se puede mejorar el proceso para incrementar la performance? [31].

En [30] se definen tres tipos de Minería de Procesos a partir de los archivos de eventos, estos son:

1. **Descubrimiento:** Se trata de técnicas capaces de modelizar el proceso de negocio que se está realizando únicamente a partir de la secuencia de logs correspondientes. La calidad del modelo generado está relacionada en forma directamente proporcional a la calidad de los datos de entrada. Esta es una característica común a cualquier técnica de extracción de conocimiento inductiva.
2. **Conformidad:** Se compara un modelo de proceso existente con un registro de eventos del mismo proceso. La verificación de conformidad puede ser usada para verificar si la realidad, tal como está almacenada en el registro de eventos, es equivalente al modelo y viceversa.

3. **Mejoramiento:** la idea es extender o mejorar un modelo de proceso existente usando la información acerca del proceso real almacenada en algún registro de eventos. Mientras la verificación de conformidad mide el alineamiento entre el modelo y la realidad, este tercer tipo de Minería de Procesos busca cambiar o extender el modelo[30].

En este trabajo se aplicó *Ciencia de Datos* a la distribución de productos postales en la República Argentina. Puntualmente se aplicaron técnicas de *Minería de Procesos* para analizar los archivos de registro generados por los sistemas de información postales con el objetivo de analizar y proponer mejoras al proceso de distribución.

En el proceso de distribución postal se lleva un registro de todas las actividades, desde el ingreso del producto hasta su entrega al cliente, estas deben realizarse en un orden y en un intervalo de tiempo específicos.

Se trata de un proceso en el que se registran distintos movimientos conformados por eventos tales como la recepción del producto en sucursal, el ingreso a un centro de distribución, los movimientos internos y los diversos intentos de entrega con sus resultados (pudiendo ser una entrega efectiva o no). Todas las tareas están claramente definidas y deben cumplirse para asegurar la calidad del servicio a los clientes. Ocasionalmente se producen desvíos, estos pueden por ser redundancia o inconsistencia de tareas (se repiten o no se realizan en el orden correspondiente), excesivo tiempo para su finalización u otros motivos.

Este trabajo continúa la investigación publicada en [20] donde se aplican de técnicas de Minería de Procesos a la distribución Postal en la República Argentina y en [21] donde se incorpora un análisis de performance por medio del cual es posible identificar un nuevo tipo desvíos operativos. En ambos casos se pudieron detectar diferentes tipos de desvíos operativos con gran precisión.

Los datos analizados corresponden a la distribución postal efectuada en la República Argentina en el período comprendido entre los años 2017 y 2020.

A la fecha de generación de este documento los autores no conocen la existencia de trabajos con estas características para el negocio postal en la República Argentina.

1.1. MOTIVACIÓN

Durante mi desempeño de más de 12 años en desarrollo de Sistemas de Información para el sector postal tuve la oportunidad de conocer detalladamente el proceso de entrega de varios productos con diferentes características (entrega bajo puerta, con firma, producto de alta seguridad, etc)¹ En los diversos proyectos que participé tuve la oportunidad de trabajar en conjunto con integrantes de otras áreas de negocio como Marketing, Ingeniería de procesos y Operaciones encontrando situaciones o inconvenientes operativos transversales a los diversos sectores y que afectan la calidad de servicio. Al no encontrar una herramienta capaz de solucionar estos inconvenientes o al menos disminuir su impacto decidí realizar un análisis detallado utilizando Ingeniería de Software y Ciencia de Datos.

¹Distintas modalidades de entrega en el negocio postal. Bajo Puerta: se deja en domicilio sin verificar si hay alguien. Con firma: la persona que recibe debe firmar una planilla y acreditar identidad. Producto de alta seguridad: se realiza un seguimiento detallado de cada movimiento sumado a los requisitos de entrega con firma.

Según la definición de la IEEE[25], la *Ingeniería de Software* es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software.

La ingeniería de software se ha desarrollado desde hace aproximadamente 50 años buscando diseñar, desarrollar y mantener herramientas de software que se ajusten a las necesidades de las personas y de las organizaciones.

Basado en mi experiencia profesional en el desarrollo de software, mi conocimiento de la *Ingeniería de Software* y el proceso postal decidí utilizar técnicas de Ciencia de Datos para analizar el proceso de distribución postal en la República Argentina.

1.2. OBJETIVOS

1.2.1. Objetivo principal

El objetivo central de esta tesis es aplicar las técnicas de Minería de Procesos en un contexto específico como es la distribución postal en la República Argentina utilizando una herramienta desarrollada con este fin que permita encontrar desvíos operativos y cuellos de botella.

1.2.2. Objetivos específicos

Este documento detallan las distintas etapas desarrolladas para cumplir con el objetivo general propuesto. Las tareas realizadas han sido las siguientes:

- Relevamiento de las distintas técnicas de Minería de Procesos y Minería de Datos aplicables a la distribución Postal y selección de las técnicas más adecuadas.
- En base a las técnicas seleccionadas se definió una estrategia para identificar automáticamente los desvíos operacionales.
- Con el objetivo de desarrollar un prototipo que permitiera incorporar la estrategia propuesta al sistema de trazabilidad de las piezas postales, se analizaron detenidamente las características y los componentes necesarios de la solución requerida para cubrir la problemática.
- Una vez definida la estrategia para el desarrollo del software, se generaron entregables finales con facilidad de uso para los usuarios y visualizaciones gráficas que permiten una rápida interpretación para personas no técnicas.
- Durante todo el proceso, se documentaron los pasos realizados en el análisis y el diseño de la solución y las problemáticas encontradas.
- Como aporte concreto del prototipo desarrollado, se incorporaron alarmas que se envían a los usuarios luego de la evaluación de los envíos lo que permite automatizar la detección de incidencias.
- Se generó una herramienta que automatiza y facilita el análisis de los envíos que están en proceso permitiendo una rápida detección de incidencias operativas y facilitando su resolución.

1.3. TRABAJOS PUBLICADOS

Los resultados de las investigaciones realizadas en el marco de esta tesis fueron publicados a través de los siguientes artículos:

- Process Mining Applied to Postal Distribution:

Presentado en el *XXVII Congreso Argentino de Ciencias de la Computación (CACIC 2021)* [20]. En este trabajo se realizó el descubrimiento del proceso postal a partir de una muestra del log de distribución utilizando técnicas de Minería de Procesos. Luego se comparó el proceso descubierto contra otro archivo de log más extenso (aproximadamente 16.000 trazas con alrededor de 44.000 eventos de distribución) para verificar el cumplimiento del mismo. Se comprobó que la gran mayoría de los envíos son entregados y los que no se pueden entregar se distribuyen uniformemente entre los diversos motivos de no entrega. El proceso se cumple en tiempo y forma para el 85 % de las trazas. También se encontraron desvíos operativos y se realizó un análisis sobre los mismos. Estos desvíos estaban dados en dos grandes grupos, por un lado los que tenían eventos repetidos, es decir el mismo evento cargado en más de una oportunidad generalmente en días diferentes y por otro lado inconsistencias donde los eventos secuenciales no se ejecutaban en el orden correcto incluso se repetían en sucesivos días. Ambos grupos de desvíos estaban en el orden del 10 % del total. El 5 % restante eran trazas cuyos desvíos eran menores para tener en cuenta en el análisis.

- Distribution Analysis of Postal Mail in Argentina Using Process Mining: Publicado en *Communications in Computer and Information Science. Springer* [21]. En este trabajo se continúa con el análisis sobre la muestra de 16.000 trazas que se realizó en [20] adicionando un control de performance que permite identificar aquellos envíos que cumplen correctamente el proceso pero en un intervalo de tiempo muy alto, ocasionando demoras en la entrega. Con este nuevo análisis se encontró que alrededor de un 4 % de las trazas tenían este problema. En ambos trabajos publicados se comprobó que la utilización de técnicas de Minería de Procesos ayuda a mejorar un proceso de negocio permitiendo identificar rápidamente desvíos operativos.

1.4. ESTRUCTURA DEL DOCUMENTO

El presente documento está estructurado de la siguiente forma:

1. **Introducción:** Descripción general del trabajo a realizar.
2. **Contexto General:** Descripción de las diferentes metodologías de Ciencia de Datos que se utilizarán en la investigación y el material adicional.
3. **Arquitectura propuesta:** Detalle de las técnicas que se aplicarán para el análisis.
4. **Minería de Procesos en la distribución postal:** Aplicación de Minería de Procesos a la distribución postal con datos de la República Argentina entre los años 2017 al 2020.
5. **Solución Propuesta:** Diseño y desarrollo de un prototipo completamente funcional que aplique las técnicas de Minería de Procesos a la distribución postal y alertar de forma temprana

posibles desvíos.

6. **Conclusiones:** Se expresan en este capítulo los resultados obtenidos y futuras líneas de investigación.

CAPÍTULO 2

Contexto General

2.1. PROCESO DE DISTRIBUCIÓN POSTAL

La distribución postal consiste en llevar un producto desde un lugar de origen a un destino. En la República Argentina esta distribución o entrega puede ser nacional donde el origen y el destino se encuentran dentro del país y el envío se puede llevar de cualquier punto a cualquier otro punto dentro del país, o internacional donde o la persona que envía o la persona que recibe están fuera del país. En este último caso puede ser internacional saliente o internacional entrante dependiendo del origen del envío.

El cliente que solicita realizar el envío de un producto se conoce como *Remitente*, la persona que recibe el envío se conoce como *Destinatario*.

El envío puede ser de varios productos con distintas características, entre ellos se encuentran los diferentes tipos de cartas como certificada, expreso, o simple, cartas documento o telegramas, estos son los productos *Tradicionales* del negocio postal. También se incluyen productos de *Paquetería* que pueden ser encomiendas o entregas de tipo *Puerta a Puerta* o productos de *Alta Seguridad* como tarjetas de crédito, débito o medicamentos.

Cada producto cuenta con características diferentes en cuanto a estándares de entrega, forma de recepción, cantidad de visitas, registro de entrega, seguimiento o incluso adicionales como *Aviso de recibo*, *Valor declarado* o *telegrama colacionado* que cambian las condiciones de servicio.

La descripción de cada producto y su implementación dependen de cada empresa, por ejemplo en el caso particular del Correo Oficial de la República Argentina estos se detallan en el *Manual de Productos y Servicios (PS-MM-001)* un documento interno que especifica cada producto que brinda la empresa a sus clientes.

Los productos postales se ofrecen en tres segmentos Corporativo, PyME y minorista, con condiciones diferentes en cada uno. Esta diversidad de productos, segmentos y tipos de envío genera una gran variedad de formas y procesos de entrega. Esto genera una actividad de alta complejidad donde intervienen varias personas de diferentes lugares en distintas etapas de cada proceso.

En la distribución postal se lleva un registro de todas las actividades, desde el ingreso del producto hasta su entrega al cliente. Entre estas actividades se encuentra la recepción del envío, ingreso a un

almacén, trasladados internos o intentos de entrega, entre otras.

Todas las tareas deben realizarse en un orden y en un intervalo de tiempo específicos. Dada la complejidad del proceso es frecuente que se produzcan errores o desvíos, estos pueden ser redundancia o inconsistencia de tareas donde las mismas se repiten o no se realizan en el orden correspondiente, excesivo tiempo para su finalización u otras. Bajo estas circunstancias es de vital importancia monitorear y evaluar el o los procesos de distribución para proporcionar un servicio de calidad a los clientes.

Las empresas de correo proporcionan información a los clientes para realizar el seguimiento del envío y observar los principales movimientos, esta información es parte del producto y permite brindar transparencia y una mejor calidad de servicio a los clientes. En general se dispone de un portal donde se pueden visualizar los movimientos previamente registrados del envío.

En la figura 2.1 se muestra el resultado de entrega y la historia del envío en los sitios web de Correo Argentino¹, Ocasa², Andreani³ y de OCA⁴.

¹<https://www.correoargentino.com.ar/formularios/ondnc>

²<https://www.ocasa.com/>

³<https://www.andreani.com/personas>

⁴<https://www.oca.com.ar/Busquedas/Envios>

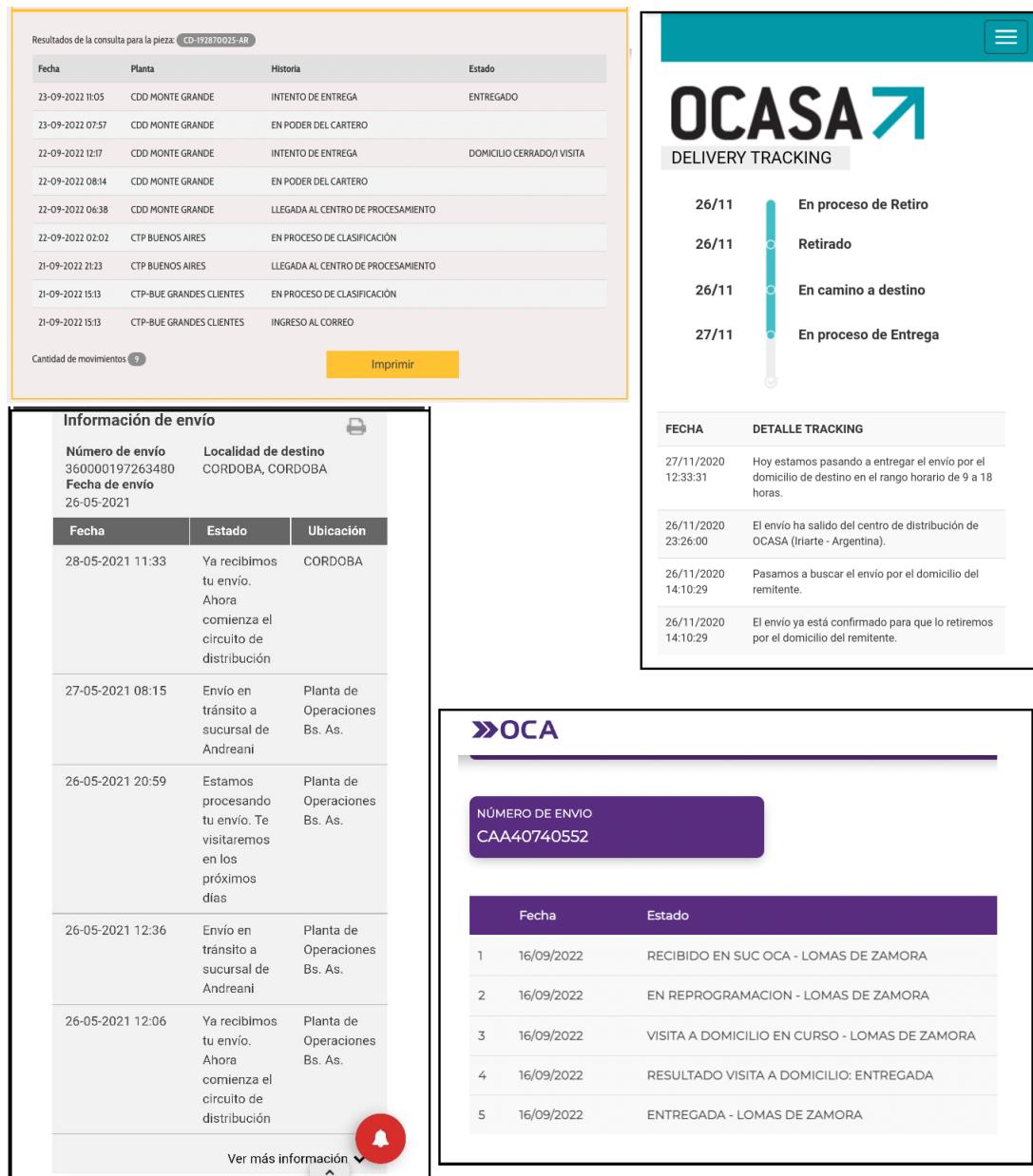


Figura 2.1: Registro de entrega de OCA, Andreani y Correo Argentino

[19]

En todos los casos, el proceso postal consta al menos de las siguientes etapas:

1. **Ingreso a Correo:** El producto a entregar ingresa a través de una sucursal, un punto de retiro o un sistema informático según sea el caso dependiendo de las características del producto y del cliente.
2. **Asignación de número de seguimiento:** A cada producto se le asigna un identificador único sobre el cual se registrarán todos los eventos relacionados con la distribución.
3. **Envío Sede Central Operativa:** Los productos se envían a un centro de clasificación y distribución donde agrupan de forma manual o automatizada según lugar de destino (por localidad y código postal) y características.

4. **Traslado a Centro de Distribución:** Los productos se envían al centro de distribución más cercano al destino para la etapa final de entrega.
5. **Entrega en domicilio:** Los distribuidores, comúnmente conocidos como carteros realizan el reparto de los productos, dependiendo de las características se puede intentar la entrega en más de una oportunidad.
6. **Rendición:** Se brinda al remitente un informe con el resultado de la entrega y los diferentes estados del proceso.

En la República Argentina el servicio postal está regulado por el Ente Nacional de Comunicaciones (ENACOM)⁵ y la principal empresa del rubro es el *Correo Oficial de la República Argentina S.A.* con más de 1800 sucursales y llegada a todos los puntos del país.

Para la realización de este trabajo se eligió el producto *Carta Documento* ya que es un producto de alta seguridad, con registro y seguimiento de todas las etapas del proceso, entrega bajo firma del destinatario, con dos intentos de entrega dejando aviso de visita en ambos casos. Si el envío no se pudo entregar se espera en sucursal un período de tiempo a que el destinatario lo venga a retirar y una eventual devolución al cliente en caso de no entrega.

El proceso del producto elegido se detalla en la figura 2.2 y se describe de la siguiente manera:

El producto ingresa a una sucursal de correo donde se recepciona por un operador quien verifica que el envío contenga todos los datos necesarios para su correcta distribución. En este caso se verifica que los datos para la entrega estén correctamente cargados, tanto del remitente como del destinatario y además que el producto cumpla con las características requeridas de tamaño y peso. Si todo es correcto se procede con la recepción, en caso contrario se rechaza el envío.

Al ingresar el producto se le asigna un número de seguimiento único y diferente al de cualquier otro envío, este número se utilizará para registrar todos los eventos asociados al envío, generando una historia completa del proceso, comenzando por su recepción en sucursal.

Luego se trasladan los envíos al centro de distribución donde se registra su ingreso y se realiza la clasificación a destino. A continuación se traslada el producto a sucursal de destino registrando su ingreso en la misma. Luego se procede con los intentos de entrega, si el envío se entrega correctamente se registra el evento y finaliza el proceso. Si el envío no puede entregarse debido a un motivo definitivo por ejemplo problemas con la dirección, no existe el número, dirección incompleta, se mudó o falleció el destinatario, se registra el motivo y finaliza el proceso.

Si el motivo de no entrega no es definitivo, por ejemplo el destinatario no se encuentra en el domicilio al momento de la entrega, se deja un aviso de visita y se procede con un segundo intento de entrega al próximo día hábil. En caso de entrega en el segundo intento finaliza el proceso. Si no se pudo entregar nuevamente se deja un aviso de visita y se guarda el envío en sucursal durante un período de tiempo, generalmente una semana, esperando que el destinatario haga el retiro. En caso de que el destinatario no pase a retirar el envío luego del tiempo especificado se procede a la devolución del mismo al remitente.

⁵<https://www.enacom.gob.ar/>

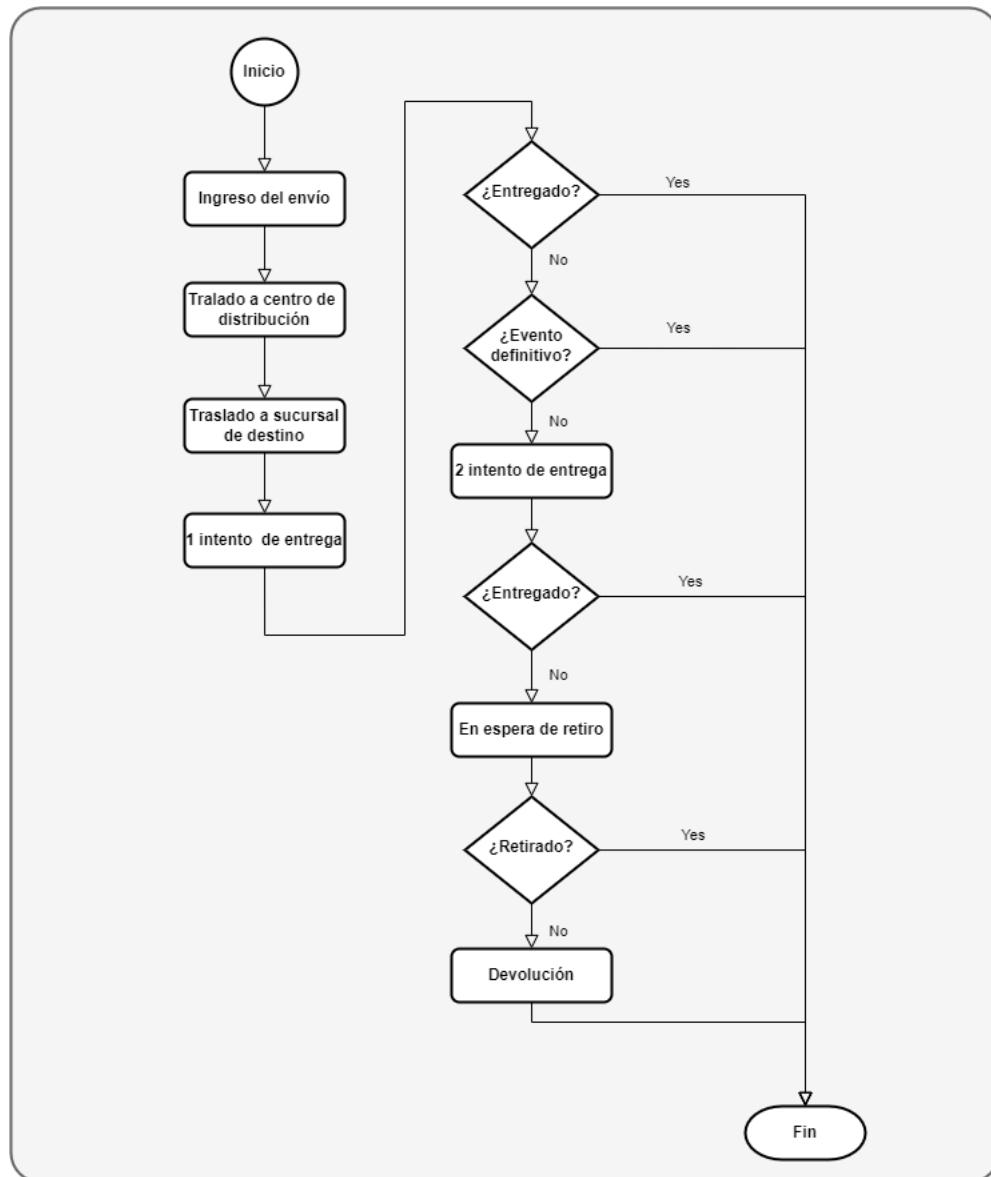


Figura 2.2: Proceso de Distribución Postal para el caso de estudio

[19]

Dada la complejidad del proceso y la cantidad de actores intervenientes se producen habitualmente demoras y problemas con la entrega. Entre los más comunes están omitir algún paso en el proceso, lo que lleva a no cumplir con los estándares de entrega.

2.2. BUSINESS PROCESS MANAGEMENT (BPM)

La Administración de Procesos de Negocio (BPM por sus siglas en inglés) es un enfoque disciplinado para identificar, diseñar, ejecutar y monitorear procesos de negocio. Estos pueden ser automatizados o no automatizados.

BPM se enfoca en encontrar las causas principales de los problemas de performance y su mejora. Busca descubrir las debilidades del proceso y no de la gente que lo realiza, esto puede ser extremadamente complejo debido a la cantidad de procesos que puede tener una organización, incluso los procesos

pueden funcionar correctamente y aún no alcanzar los niveles de performance requeridos por la organización o los clientes.

Utilizando BPM se busca obtener resultados que estén alineados con los objetivos de la organización. En [33] se define el proceso de BPM como un ciclo que comienza con la creación de un proceso formal claramente definido y documentado. Este primer paso es muy importante ya que muchas veces las organizaciones carecen de un proceso definido de extremo a extremo (por ejemplo desde la adquisición de materia prima hasta la entrega al cliente), en ocasiones se resuelven temas con improvisación y sin que quede registro de ello. Luego de que el proceso está en marcha debe monitorearse de forma continua, y se debe evaluar su rendimiento con métricas que se relacionan con las necesidades del cliente y de la empresa. Estas métricas se comparan con los objetivos del proceso; los mismos pueden basarse en las expectativas de los clientes, los puntos de referencia de la competencia, las necesidades de la empresa y otras fuentes. Si el rendimiento no cumple con los objetivos, se debe determinar el motivo y rediseñar el proceso.

En general los procesos no cumplen con los requisitos de rendimiento, ya sea por un diseño defectuoso o una ejecución parcial o deficiente. Si la falla está en la ejecución, se debe determinar la causa raíz como entrenamiento inadecuado, o recursos insuficientes, o equipo defectuoso, u otras. Hacerlo es una tarea difícil, debido al gran número de posibles causas fundamentales, sin embargo en general, una vez que se ha encontrado la causa raíz, es fácil de solucionar. Por el contrario si el problema está en el diseño del proceso, en general es fácil de encontrar (debido al rendimiento inadecuado) pero mucho más difícil de solucionar ya que puede requerir un rediseño completo del proceso. Una vez que se ha elegido e implementado la modificación adecuada, se evalúan los resultados y todo el ciclo comienza de nuevo.

En la figura 2.3 se muestra el ciclo de BPM propuesto en [33].

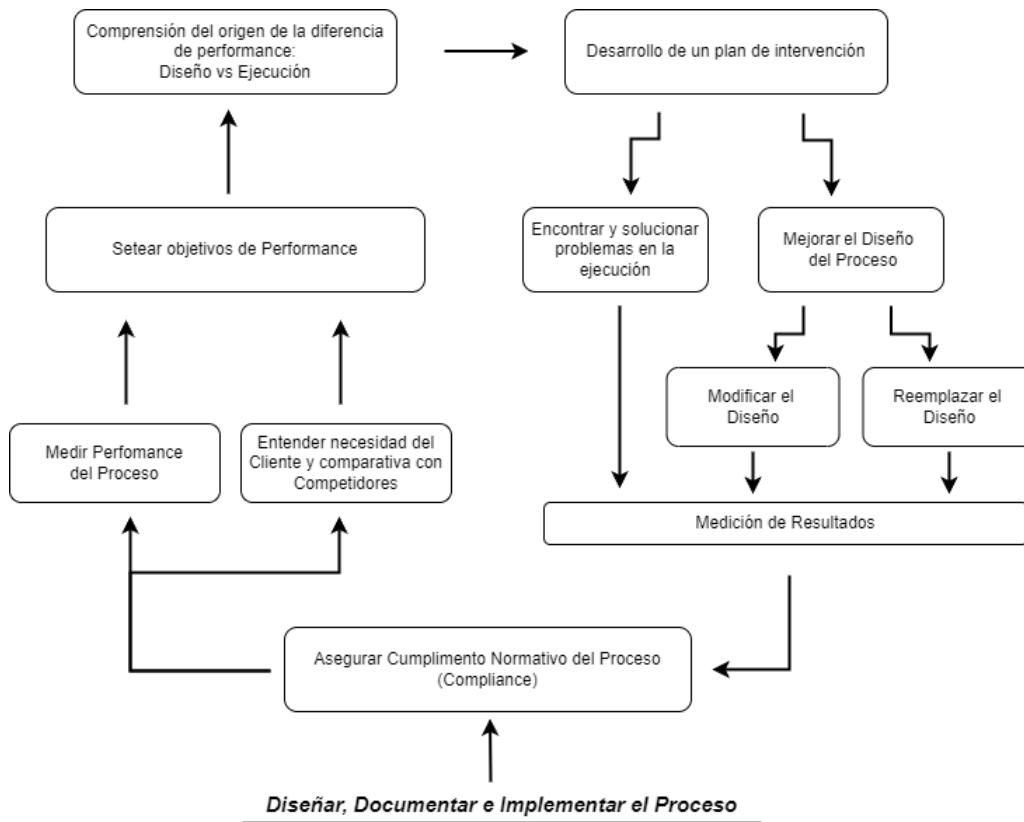


Figura 2.3: Ciclo BPM

[33]

BPM se basa en que el rendimiento de una organización no se enfoca en prueba y error, o generando una mayor motivación en la gente con estímulos financieros, sino a través de la gestión deliberada de los procesos comerciales de extremo a extremo a través de los cuales se crea todo el valor para el cliente. De hecho, BPM es un enfoque centrado en el cliente para la gestión organizativa [33]. Los clientes evalúan los resultados. Tales resultados provienen de los procesos comerciales, de las secuencias de actividades que trabajan juntas. Optimizando los procesos se obtienen mejores resultados que derivan en satisfacción de los clientes. De todas maneras, tener un buen diseño para un proceso no garantiza buenos resultados debido a la complejidad del mundo real; las máquinas se rompen, las personas se van, los datos se corrompen, ocasionando que un proceso bien diseñado no logre el rendimiento que puede dar.

Las empresas que utilizan la gestión de procesos para supervisar y verificar el rendimiento de cada proceso pueden reconocer y corregir problemas de forma mucho más dinámica. También descubren más fácilmente las oportunidades de mejorar el diseño del proceso.

BPM establece cinco “Facilitadores” para diseñar un proceso de alto rendimiento [4], estos son:

1. **Diseño del Proceso:** Es el aspecto más importante del proceso. Aquí se detallan las tareas que se deben realizar, por quién, en qué momento, en qué lugar y bajo qué circunstancias, el grado de precisión, con qué información entre otros. El diseño es la especificación del proceso; sin un diseño, solo hay una actividad individual descoordinada y un caos organizativo.

2. **Métricas del Proceso:** Los procesos necesitan métricas de extremo a extremo que se derivan de las necesidades del cliente y los objetivos de la empresa. Es necesario establecer objetivos en términos de estas métricas y el rendimiento supervisado en relación con ellas. Se debe implementar un conjunto equilibrado de métricas de proceso, tales como costo, velocidad y calidad, para que las mejoras en un área no oculten las caídas en otra.
3. **Ejecutantes del Proceso:** Las personas que trabajan en procesos necesitan un conjunto diferente de habilidades y comportamientos de los que trabajan en funciones y departamentos convencionales. Necesitan una comprensión del proceso general y sus objetivos, la capacidad de trabajar en equipo y la capacidad de gestionarse a sí mismos. Sin estas características, no podrán realizar el potencial del trabajo de extremo a extremo.
4. **Infraestructura del Proceso:** Los ejecutantes deben contar con el apoyo de los sistemas de información y recursos humanos para cumplir con las responsabilidades del proceso. Los sistemas de información funcionalmente fragmentados no admiten procesos integrados, y los sistemas convencionales de recursos humanos (formación, compensación y carrera, etc.) refuerzan las perspectivas de trabajo fragmentadas. Se necesitan sistemas integrados, como sistemas ERP y sistemas de compensación basados en resultados, para los procesos integrados.
5. **Dueño del Proceso:** En una organización convencional, nadie es responsable de un proceso de extremo a extremo, por lo que nadie estará en condiciones de gestionarlo de extremo a extremo, es decir, llevar a cabo el ciclo de gestión de procesos. Una organización que se toma en serio sus procesos debe tener propietarios de procesos: altos directivos con autoridad y responsabilidad por un proceso en toda la organización en su conjunto. Ellos son los que realizan el trabajo ilustrado en la figura 2.3.

2.3. BPMN

Business Process Model and Notation (BPMN) es una notación gráfica estándar diseñada para representar y documentar las actividades que conforman los procesos de negocio de una organización y los mensajes que se intercambian entre los participantes de las diferentes actividades. BPMN es mantenido por el Object Management Group (OMG)⁶ una organización internacional sin fines de lucro con el objetivo de generar estándares tecnológicos que provean valor agregado cuantificable para la industria.

Actualmente hay una gran variedad de herramientas de gestión de procesos empresariales (BPM), varias basadas en XML proporcionando mecanismos formales para la definición de procesos de negocio. En la mayoría de los casos estas herramientas están diseñadas para interactuar con sistemas de BPM y están orientados a usuarios avanzados o con grandes conocimientos técnicos o matemáticos. Por sus características un proceso de negocio complejo puede organizarse en un formato complejo y poco intuitivo, que se ajusta muy bien a las herramientas de software o a un programador pero de difícil comprensión para analistas o gerentes de una organización que son los que generalmente se encargan de desarrollar, administrar y monitorear el proceso.

⁶<https://www.omg.org/>

BPMN pretende proporcionar una notación fácilmente legible y entendible por todos los actores que participan en el proceso. En general, estos actores (gerentes, analistas) se sienten muy cómodos visualizando los procesos de negocio en un formato de diagrama de flujo.

En [3] se expresa que hay una gran cantidad de analistas de negocio con diagramas de flujo simples. Esto crea una brecha técnica entre el formato en que los procesos de negocio son diseñados y el formato de los lenguajes, que ejecutarán estos procesos de negocio. Esta brecha debe cerrarse con un mecanismo formal que mapee la visualización adecuada de los procesos de negocio (una notación) al formato de ejecución apropiado (un lenguaje de ejecución BPM) para estos procesos de negocio. Esta brecha se puede resolver con la estandarización del Modelo y Notación de Procesos de Negocio (BPMN).

BPMN proporciona múltiples diagramas, que están diseñados para su uso por las personas que diseñan y gestionan los procesos de negocio, además proporciona un mapeo a un lenguaje de ejecución. Por lo tanto, BPMN permite tener un mecanismo de visualización estándar para los procesos de negocio definidos en un lenguaje de procesos de negocio optimizado para la ejecución.

BPMN proporciona a las empresas la capacidad de comprender sus procedimientos comerciales internos en una notación gráfica y brinda a las organizaciones la capacidad de comunicar estos procedimientos de manera estándar. Actualmente, hay decenas de herramientas y metodologías de modelado de procesos. Dado que las personas se moverán de una empresa a otra y que las empresas se fusionarán y divergirán, es probable que los analistas de negocios necesiten comprender múltiples representaciones de los procesos de negocio, representaciones diferentes del mismo proceso a medida que se mueve a través de su ciclo de vida de desarrollo, implementación, ejecución, monitoreo y análisis. Por lo tanto, una notación gráfica estándar facilita la comprensión de las colaboraciones de rendimiento y las transacciones comerciales dentro de las organizaciones y entre ellas[3].

La versión actual es BPMN 2.0. Esta versión permite el modelado de procesos de negocio extremo a extremo. En [3] se define que hay tres tipos básicos de submodelos dentro de un modelo BPMN de extremo a extremo, estos pueden ser:

1. **Procesos:** los procesos pueden ser de dos tipos:

- a) Procesos privados: Son los internos de una organización específica. Se conocen como flujos de trabajo y pueden ser de dos tipos : ejecutables y no ejecutables. Un proceso ejecutable se modela para realizar una acción específica, un proceso no ejecutable se realiza para documentar un proceso existente en forma detallada, es decir tiene toda la información necesaria para su ejecución.
- b) Procesos públicos: Un proceso público representa la interacción de un proceso privado y algún otro participante externo a la organización.

2. **Coreografías:** Una coreografía es una definición del comportamiento esperado, básicamente un contrato de procedimiento, entre los participantes que interactúan. Mientras que existe un proceso normal dentro de un grupo, existe una coreografía entre grupos o participantes. La coreografía se parece a un proceso de negocio privado, sin embargo, una coreografía es

diferente en el hecho de que las actividades son interacciones que representan un conjunto (1 o más) de intercambios de mensajes, que involucra a dos o más participantes. Además, a diferencia de un Proceso normal, no hay un controlador central, una entidad responsable o un observador del Proceso.

3. **Colaboraciones:** Una colaboración describe las interacciones entre dos o más entidades comerciales. También se pueden mostrar los mensajes asociados con los flujos de mensajes. La colaboración se puede mostrar como dos o más procesos públicos que se comunican entre sí.

Todos los procesos tienen un evento de inicio y un evento de fin. BPMN proporciona una gran cantidad de componentes para documentar cada etapa del proceso, los principales se muestran en la figura 2.4.

Elemento	Descripción	Notación
Evento	Un evento es algo que "sucede" durante el curso de un proceso. Generalmente tiene una causa (disparación) o un impacto (resultado)	○
Actividad	Una actividad es un término genérico para el trabajo que la empresa realiza. Una actividad puede ser atómica o no atómica (compuesta). Los tipos de actividades que forman parte de un modelo de proceso son: Subproceso y Tarea.	□
Gateway	Una puerta de enlace se utiliza para controlar la divergencia y convergencia de los flujos de secuencia en un proceso y en una coreografía. Por lo tanto, determinará la ramificación, la bifurcación, la fusión y la unión de caminos.	◇
Flujo de secuencia	Un flujo de secuencia se utiliza para mostrar el orden en que las actividades se realizarán en un proceso y en una coreografía.	→
Flujo de mensajes	Un flujo de mensajes se utiliza para mostrar el flujo de mensajes entre dos participantes que están preparados para enviarlos y recibirlos.	↔
Asociación	Una asociación se utiliza para vincular información y artefactos con elementos gráficos de BPMN. Una punta de flecha en la Asociación indica una dirección de flujo (por ejemplo, datos), cuando corresponda.→
Grupo	Un grupo es la representación gráfica de un participante en una colaboración. Es un contenedor gráfico para particionar un conjunto de actividades de otros grupos. Puede no tener detalles internos, es decir, puede ser una "caja negra".	Box Name
Carril	Un carril es una subpartición dentro de un proceso, a veces dentro de un grupo, y se extenderá toda la longitud del proceso, ya sea vertical u horizontalmente. Los carriles se utilizan para organizar y categorizar las actividades.	Box Name Name
Datos	Proporcionan información sobre qué actividades deben realizarse y/o qué producen. Pueden representar un objeto singular o una colección de objetos. La entrada y salida de datos proporcionan la misma información para los procesos.	File
Mensaje	Un mensaje se utiliza para describir el contenido de una comunicación entre dos participantes o una entidad asociada empresarial.	✉
Inicio	Indica el comienzo de un proceso.	○
Fin	Indica el fin de un proceso.	●

Figura 2.4: Principales componentes de BPMN

[19]

En [3] también se definen 5 categorías donde se agrupan los componentes de BPMN, estas son:

1. **Objetos de flujo:** son los principales elementos gráficos para definir el comportamiento de un proceso de negocio. Hay tres objetos de flujo: eventos, actividades y puertas de enlace.
2. **Información:** La información se representa con cuatro elementos: entradas de datos, salidas de datos, objetos de datos y almacenes de datos.
3. **Conexión de objetos:** La conexión puede ser de los objetos de flujo entre sí o con otra información y puede ser a través de: flujos de secuencia, flujos de mensaje y asociaciones.
4. **Grupos:** permiten agrupar los elementos, puede ser con grupos o carriles.
5. **Artefactos:** se utilizan para proporcionar información adicional sobre el proceso. Hay dos artefactos estandarizados agrupación y anotación de texto.

En el presente trabajo se utilizará BPMN para modelar el proceso de distribución postal.

2.4. MINERÍA DE PROCESOS

Como se mencionó anteriormente, la Minería de Procesos es parte de la *Ciencia de Datos*.

En la Ciencia de Datos hay algunas preguntas genéricas que se pueden realizar en cualquier situación. Tal es el caso de la pregunta ¿Qué pasó?. En el caso de estudio aquí planteado, se busca determinar si hay desvíos o un cuello de botella. De ser así, si se registran datos sobre esos eventos, es posible ver qué cosas pasaron.

A continuación es posible preguntar ¿Por qué pasó?, es decir ¿por qué hubo tal demora o desvío?. De la misma manera si se registran datos de los eventos estos se pueden analizar y obtener esta información. Estas preguntas permiten saber qué ocurrió en el pasado, pero con la *Ciencia de Datos* se pueden extraer y responder preguntas sobre el futuro. Por ejemplo ¿Qué pasará ante determinada situación?, ¿Qué se puede aprender de la información histórica para hacer predicciones sobre lo que sucederá?, o ¿Cómo se puede mejorar esta demora o desvío?, o ¿Qué acciones se pueden tomar para que no vuelvan a ocurrir, o al menos mitigar el riesgo de que ocurran?

Por lo tanto con la *Ciencia de Datos* se puede usar analítica para mejorar un comportamiento, para ello hay que tener en cuenta que se debe trabajar con grandes volúmenes de datos, que probablemente se obtendrán de una gran variedad de fuentes y que están en constante crecimiento. Además se deben considerar factores éticos y sociales al trabajar con estos datos y extraer conocimiento de ellos.

El desafío actual no es guardar los datos, ya que en los últimos tiempos se incrementó notablemente el volumen de datos que se registra. El desafío es verificar la veracidad de los datos almacenados, su correcta interpretación y utilización.

Se puede usar *Estadística y Minería de Datos* para obtener información muy valiosa, pero estas técnicas están principalmente enfocadas en los datos y no en los procesos que registran esos datos.

La Minería de Procesos pretende responder estas preguntas y obtener esta información con el foco en análisis de procesos basados en datos.

No se desea mejorar los datos, se desea mejorar el proceso que genera esos datos. Se busca una mejora en el proceso de extremo a extremo. Este enfoque permite generar nuevas preguntas como ¿Cuál es el proceso que la gente realmente sigue? ¿Qué hace realmente?, ¿Cuáles son los cuellos de botella?, ¿Qué los está causando?, ¿Dónde y por qué la gente o las máquinas se desvían de un proceso esperado?

La Minería de Procesos busca aprender los procesos de negocio y mejorarlo [14].

En [31] se pone a la Minería de Procesos como una metodología que cierra la brecha entre el modelado clásico de procesos (BPM) donde no se tienen en cuenta los datos a la hora de diseñar el proceso y la *Minería de Datos* o la *Estadística* que se centran en los datos pero no analizan el proceso que los genera.

Se realiza Minería de Procesos para responder preguntas relacionadas con el rendimiento y con el cumplimiento del proceso para obtener mejores resultados.

El punto de partida para la Minería de Procesos son los datos de eventos almacenados por los sistemas de información en archivos de log o en bases de datos en varios formatos. Un evento es algo que ocurrió, generalmente se registra una descripción del evento, quién fue el actor que lo realizó y la fecha en ocurrió. Adicionalmente se registra un identificador del evento y uno del caso o traza que permitirá agrupar todos los eventos que le pertenezcan. Los formatos de los archivos de eventos pueden diferir dependiendo de los sistemas de información que los generan o del propósito para el que fueron generados, pero casi siempre contienen al menos los datos anteriormente mencionados.

El punto más importante en los archivos de registro es la veracidad de la información, es decir la calidad del archivo. Debido a que son el punto de entrada de la Minería de Procesos se debe contar con archivos de gran calidad ya que los resultados obtenidos dependen en gran medida de ellos, por eso al momento de diseñar los datos que se guardarán se deberá evaluar cuales son los más descriptivos de las tareas que se están registrando.

El formato elegido por la Task Force on Process Mining⁷ de IEEE para los archivos de log es el xes⁸ el cual es soportado por las principales herramientas de *Process mining* como ProM Tools⁹ y Disco¹⁰

En la figura 2.5 se muestra un ejemplo de registro de eventos en un archivo de log con el formato necesario para aplicar Minería de Procesos.

⁷<https://www.tf-pm.org/>

⁸<https://xes-standard.org/>

⁹<https://www.promtools.org/doku.php>

¹⁰<https://fluxicon.com/disco/>

Identificador de traza	Identificador de evento		Descripción del evento		Fecha evento
	123 trazalD	123 EveID	abc eveDescrip	eveFecha	
31	481,053	0	INGRESADO	2017-08-16 10:45:22	
32	481,053	2	1 INTENTO DE ENTREGA	2017-08-18 11:15:00	
33	481,053	9	DEVOLUCION	2017-08-18 13:00:00	
34	481,054	0	INGRESADO	2017-08-16 10:45:28	
35	481,054	2	1 INTENTO DE ENTREGA	2017-08-22 12:05:00	
36	481,054	9	DEVOLUCION	2017-08-22 17:25:00	
37	481,055	0	INGRESADO	2017-08-16 10:45:27	
38	481,055	1	ENTREGADO	2017-08-22 15:13:00	

Figura 2.5: Ejemplo de registro de eventos en un archivo log

[19]

Por ejemplo en el caso de una biblioteca se puede tener el caso de un préstamo de un libro donde posiblemente se registre la reserva del libro por el socio, su retiro y devolución en tres eventos distintos. Estos tres eventos componen la historia de ese préstamo y deben ejecutarse en ese orden. No sería lógico tener una devolución del préstamo antes que la entrega, aunque estas situaciones pueden darse por errores en el registro de los datos o en el proceso.

Como se mencionó, la Minería de Procesos opera sobre los archivos de registro de uno o varios sistemas de información. Se comienza con una extracción de datos que contenga todos los eventos de una determinada actividad, luego de filtrar estos datos se puede realizar un análisis automático donde se determina cuál es el proceso de negocio, qué actividades lo componen y qué secuencia se debe cumplir. De esta forma es posible hacer un análisis y responder las preguntas antes mencionadas como por ejemplo ¿Qué fue lo que pasó?, ¿Por qué pasó?, ¿Qué es lo que podría pasar en el futuro?, ¿Cómo se puede mejorar el control?, ¿Cómo se puede mejorar la performance? [31].

Con la información obtenida luego de la interpretación del modelo se podrán realizar diferentes tipos de análisis como por ejemplo: quién realiza la actividad, si la actividad está completa o si se realiza en un intervalo de tiempo coherente.

Poniendo el foco en la relación entre los modelos de proceso y los datos de eventos se encuentra que en la Minería de Procesos hay tres tipos de relaciones y dependiendo de en qué sentido se relacionen son llamadas: Play-out, Play-in y Replay.

- *Play-out*: Se comienza desde el diseño del modelo de forma manual y a partir de ahí se genera comportamiento, es decir se generan registros con información que verifica el proceso. Esta información cumple con el proceso *ideal* y generalmente se usa para verificar manualmente el proceso generado.

Play-out ejemplifica el uso clásico de BPM para generar un modelo sin interactuar con datos de eventos.

- *Play-in*: Por el contrario, aquí se parte de los registros de eventos y se genera el modelo correspondiente. En este caso el proceso es *descubierto* desde los archivos de log. La idea básica es mirar el número de comportamientos que se encuentran registrados y automáticamente deducir un modelo. No se genera ningún modelo a mano. Se infieren automáticamente modelos

de proceso a partir de datos de eventos sin procesar.

Play-in se relaciona con el descubrimiento, aprender automáticamente un modelo de negocio sin ninguna información adicional que los registros de datos.

- *Replay*: Se toman datos reales y se verifican contra el proceso generado. Se realiza una conformidad de los datos sobre el modelo, y se evalúa que tan cercano es este a la realidad. Replay permite además analizar el rendimiento del modelo. Si se cuenta con el instante en que se generó el evento se puede analizar cuánto tiempo toman las actividades, cuáles son sus retrasos, con qué frecuencia se ejecuta una actividad o se sigue un camino en particular. Replay permite verificar conformidad, investigar problemas de rendimiento y proponer mejoras al proceso.

La figura 2.6 extraída de [31] muestra las diferentes relaciones de los modelos y los archivos de datos.

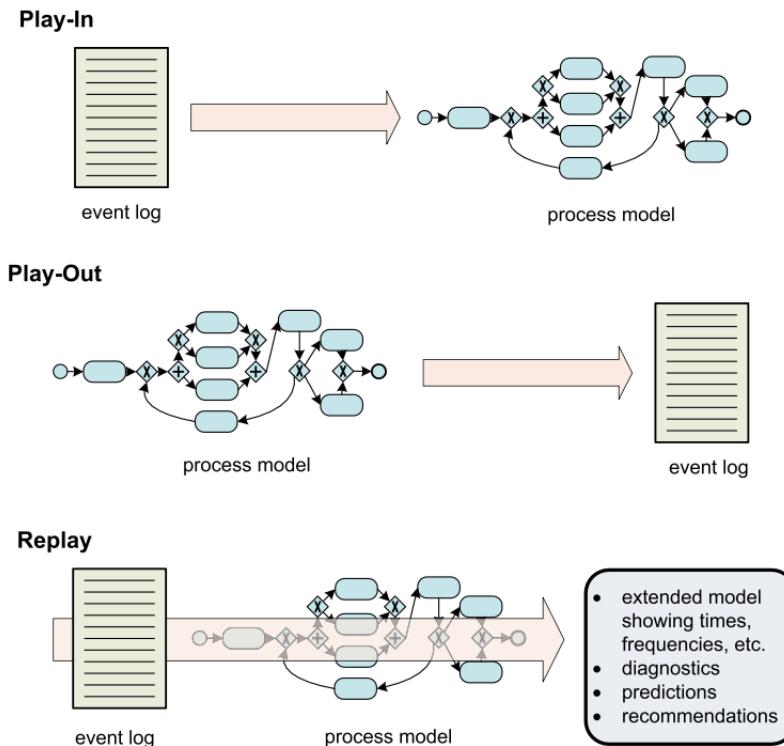


Figura 2.6: Play-In, Play-Out, Replay

[31]

La Minería de Procesos puede clasificarse en tres tipos [30]:

- *Descubrimiento*: Se trata de técnicas capaces de modelar un proceso de negocio que se está ejecutando únicamente a partir de la secuencia de archivos de log correspondientes; estos logs pueden extraerse de archivos texto, bases de datos o algún otro medio. Para ello se utilizan algoritmos que de forma automatizada analizan los archivos y establecen el orden en que se ejecutan las tareas. Como resultado se obtiene un proceso de negocio que representa la realidad de lo que se está realizando en la operación, la cual puede diferir del proceso diseñado originalmente. Esta información es de vital importancia para los analistas de proceso y proporciona un activo que se puede utilizar para mejoras al proceso.

Posiblemente el algoritmo más popular para realizar esta tarea es el algoritmo *Alpha* [28] que genera un red de Petri con el proceso descubierto. Este algoritmo escanea los eventos en busca de patrones y con ellos se arma el proceso. Es un algoritmo simple y eficiente pero no se ajusta a todos los tipos de proceso, además es muy sensible al ruido (comportamiento anormal) en los datos.

Otros algoritmos son *Euristics Mining*[34], *Fuzzy Mining*[6] e *Inductive Mining*[15]. Algunos son muy buenos para manejar ruido y trazas incompletas. En general se basan en la frecuencia de actividades y la cantidad de veces que una actividad es seguida de otra para generar el proceso de salida.

La calidad del modelo generado está relacionada en forma directamente proporcional a la calidad de los datos de entrada. Esta es una característica común a cualquier técnica de extracción de conocimiento inductiva. Es fundamental que las secuencias de eventos relevadas representen completamente al proceso a modelizar. De lo contrario, quedarán aspectos sin descubrir.

- *Verificación de Conformidad*: Permite contrastar una secuencia de eventos contra un modelo de proceso ya sea el descubierto anteriormente u otro, para determinar cuánto se cumple y cuáles son las ocurrencias que se desvían. Las técnicas de verificación de conformidad consideran cada uno de los eventos del archivo registro como actividades del modelo. De esta manera, los comportamientos observados en el registro de eventos se pueden comparar con el comportamiento que hay en el modelo. Hay varias aplicaciones de verificación de conformidad. Por ejemplo, identificar desvío, evaluar la calidad de un modelo de proceso descubierto o mejoras al proceso y auditoría.

Pueden utilizarse aplicaciones que representan gráficamente los resultados y realizan animaciones para una mejor visualización. El análisis resultante de esta operatoria puede ser de una gran precisión ya que se trabaja con datos reales, no es una simulación.

- *Mejoramiento*: Se trata de mejorar el proceso existente a partir del análisis realizado. En el mejoramiento del proceso se toma como entrada un archivo de eventos y el modelo que se desea mejorar. Es decir se puede mejorar el proceso analizando lo que ocurrió en el pasado. Permite analizar tiempos medios de ejecución de trazas, cuáles son los desvíos y los cuellos de botella.

Se diferencia de la Verificación de Conformidad porque aquí el foco está puesto en el proceso y no en la ejecución de las trazas.

La figura 2.7 extraída de [31] muestra los distintos tipos de Minería de Procesos y su interacción con los procesos reales y sus datos.

Existen adicionalmente cuatro aspectos a tener en cuenta para obtener resultados de calidad en la Minería de Procesos. Estos son: precisión, ajuste, generalización y simplicidad [31]. Debe mantenerse una relación balanceada entre estas cuatro fuerzas para que el modelo descubierto sea representativo del proceso y fácil de comprender.

Por último, pero no menos importante, cabe destacar que generalmente se encuentra una gran cantidad de ruido en los datos de entrada que puede deberse a duplicidad de datos o trazas incompletas que

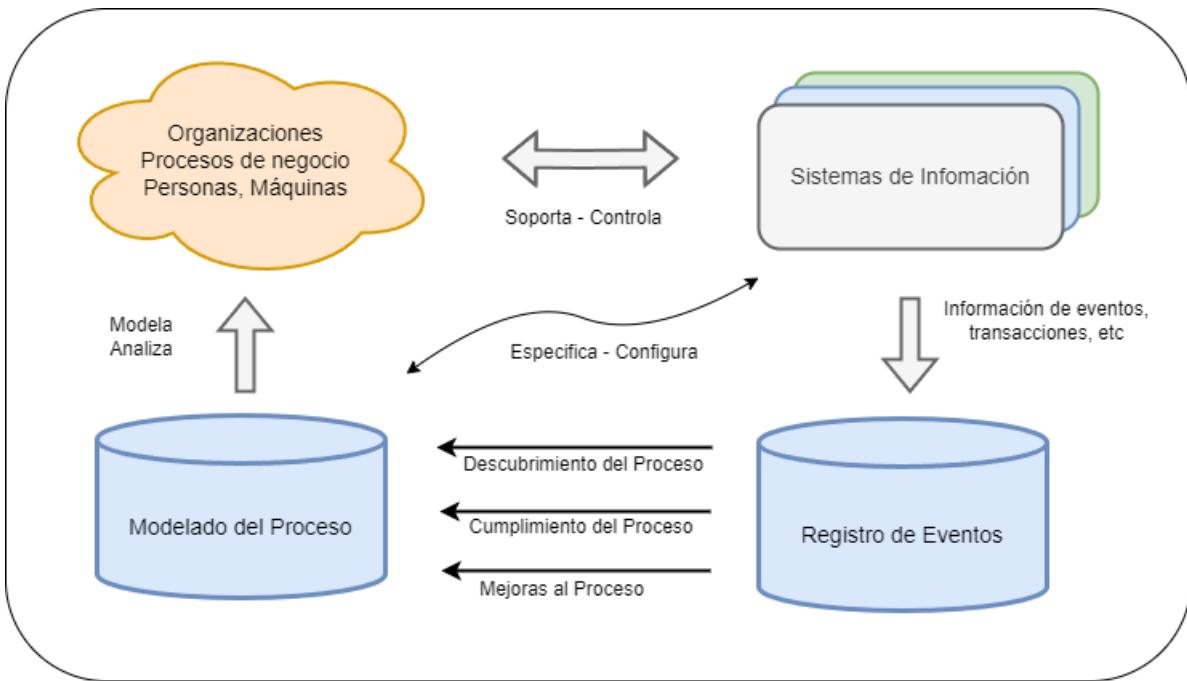


Figura 2.7: Diferentes tipos de Minería de Procesos

[31]

pueden distorsionar el resultado del análisis [7]. Como en otras técnicas de minería se suele realizar un procesamiento previo para mejorar la calidad de los datos de entrada con el fin de eliminar datos incompletos o corruptos que puedan llevar a una modelización errónea.

2.5. TRABAJOS RELACIONADOS

Existen algunas publicaciones que relacionan el negocio postal, la logística, la inteligencia de datos y la Minería de Procesos. Tal es el caso de [9] que utiliza *Data Mining* en un entorno de Big Data para el correo de China. Debido a la problemática del negocio postal, en dicho artículo se utilizaron técnicas de Clustering para agrupar a los clientes según el comportamiento, los hábitos de consumo y el foco de interés logrando generar una estrategia de Marketing postal más precisa y efectiva con resultados muy satisfactorios.

Otro caso es [1] donde se aplica Minería de Procesos en logística buscando similitudes y diferencias entre distintos procesos de entrega en un contexto cambiante de manufactura y logística. En dicho trabajo se comparan diferentes procesos utilizando técnicas de agrupamiento en busca de lograr una documentación automatizada de procesos en un contexto cambiante.

Por otro lado, en [12] se analizaron los registros del sistema no cumplen los requisitos que los algoritmos de descubrimiento de procesos exigen a los datos de registro. Para estos casos, se utilizó la reconstrucción de trazas para extraer trazas de los datos de registro, para luego utilizarlas para el descubrimiento de procesos en un paso posterior. En este trabajo se propuso combinar diferentes métodos para agrupar mensajes de registro. Si bien sólo presenta un breve experimento, es una muestra de que la reconstrucción de trazas a partir de datos de registro simples podría ser posible.

En [23] se vincula la Ingeniería de Software con la Minería de Procesos para mostrar la utilidad de

esta última en el proceso de desarrollo, presentando varios casos de estudio y mostrando los desafíos que enfrentan ambas disciplinas al trabajar en conjunto.

Más recientemente se encuentra [24] donde se propone emplear la Minería de Procesos para modelar, analizar y mejorar la cadena de suministro de productos utilizando, por ejemplo, con la identificación de procesos, la mejora del rendimiento y la detección de problemas. Además el artículo analiza cómo la Minería de Procesos puede integrarse con otras técnicas analíticas, como el aprendizaje automático, para mejorar aún más la gestión de la logística lo que puede conducir a beneficios para la empresa.

Una visión interesante se plantea en [16] donde se pone foco en el grado de abstracción que se debe tomar para proyectos de Minería de Procesos en logística. El nivel de abstracción se refiere a la granularidad de los datos que se utilizan para modelar y analizar los procesos. En el artículo se describe un enfoque para seleccionar el nivel de abstracción adecuado. Este enfoque se basa en la identificación de los objetivos de análisis y la selección de los datos que sean relevantes para esos objetivos. También incluye un caso de estudio en el que se aplicó este enfoque a los datos de seguimiento de los envíos en una empresa de logística. Ahí se demostró que la selección del nivel adecuado de abstracción es crítica para obtener resultados precisos y útiles de la Minería de Procesos.

En [32] se presenta una metodología que sirve de guía para la ejecución de proyectos de Process Mining y que describe las diferentes etapas. Además se muestra como caso de estudio su aplicación práctica en el proceso de compras de IBM.

Otro aspecto interesante se muestra en [13] donde se hace foco en los cambios en la composición de los hogares y el entorno residencial de los últimos años en la República de Corea que ocasionaron un gran impacto en las regiones urbanas, lo que produjo un aumento considerable de la carga de trabajo de los servicios de entrega postal. En el documento se realizan diferentes tipos de análisis complejos para áreas de entrega postal utilizando aprendizaje no supervisado. Los enfoques de aprendizaje no supervisados son útiles para agrupar con éxito las zonas de servicio postal.

Además de los trabajos anteriormente mencionados las herramientas de software que permiten implementar estas técnicas avanzan en conjunto con la tecnología y los lenguajes de programación. En [2] se presenta la biblioteca PM4Py, que es una herramienta de código abierto desarrollada en Python para el análisis de procesos. El artículo aborda la brecha entre la ciencia de procesos y la ciencia de datos, proporcionando una solución para integrar ambas disciplinas.

PM4Py ofrece una amplia gama de funcionalidades para realizar tareas de Minería de Procesos, como descubrimiento de procesos, verificación de conformidad, mejora de procesos y visualización de procesos [2]. La biblioteca se basa en una estructura modular que permite a los usuarios seleccionar y combinar diferentes técnicas y algoritmos según sus necesidades específicas.

La biblioteca PM4Py se utilizó en esta tesis como parte de los componentes necesarios para implementar las técnicas de Minería de Procesos en un prototipo de software.

CAPÍTULO 3

Arquitectura de la propuesta

3.1. INTRODUCCIÓN

En esta tesis de maestría se trabajó sobre un proceso productivo existente para la distribución postal de productos trazables. Se tomaron algunos archivos productivos de distribución y se realizó el descubrimiento con las técnicas de Minería de Procesos anteriormente mencionadas.

Luego se realizó la verificación de conformidad sobre el modelo descubierto con un conjunto mucho más grande de datos productivos para verificar consistencia. A continuación se procedió con el análisis de los resultados y se evaluó la posibilidad de realizar mejoras en el proceso productivo de distribución postal para el producto elegido. Finalmente se desarrolló un prototipo completamente funcional para detectar en forma temprana desvíos en el proceso operativo.

El detalle del flujo de trabajo realizado se muestra en la figura 3.1 y se compone de la siguiente manera:

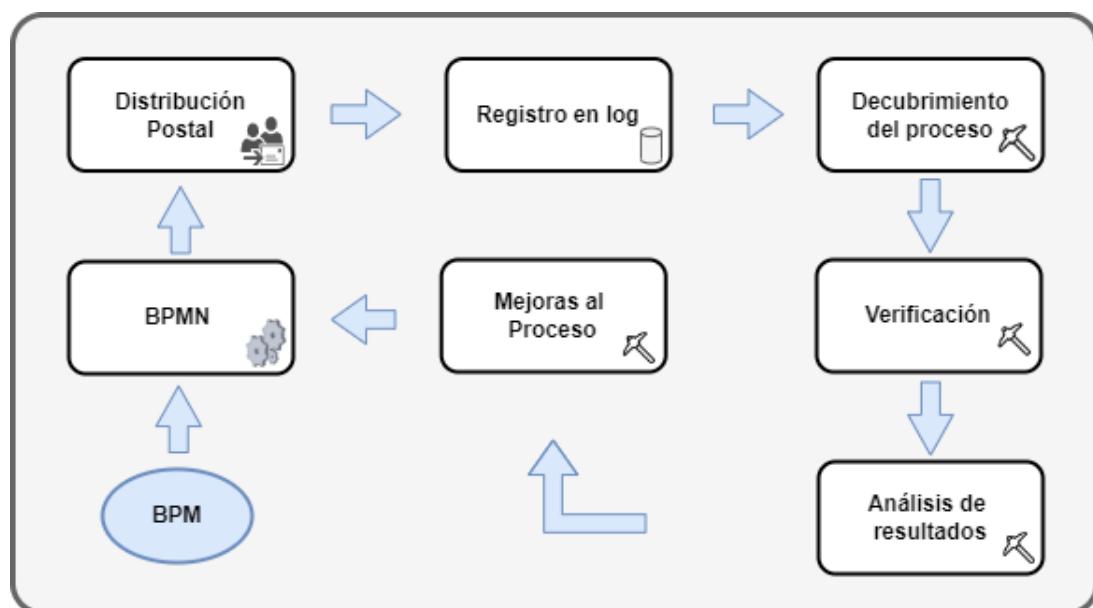


Figura 3.1: Minería de Procesos aplicada a la distribución postal

Inicia con la definición del proceso de distribución postal con BPM y su modelado en BPMN. Luego de la definición del proceso se lanza el producto al mercado y comienza su comercialización; los eventos relacionados con la distribución se registran y se almacenan en uno o más sistemas de información. El diseño original del proceso, su lanzamiento y registro de eventos se encuentran en funcionamiento desde hace algunos años lo que permite obtener información suficiente para aplicar las técnicas de Minería de Procesos.

A continuación se procede a realizar el descubrimiento del proceso *real*, es decir lo que sucede en la operación diaria y en el cual se visualizarán las diferencias con el diseñado originalmente. En esta etapa pueden utilizarse algoritmos de descubrimiento automático de procesos como el algoritmo *Alpha*[28] o el algoritmo *Inductive Visual Miner*[15].

Luego del descubrimiento se realiza una verificación del proceso contrastando una muestra de datos contra el proceso descubierto. La muestra de datos a utilizar para la verificación debe ser diferente a la empleada para descubrir el proceso.

Finalmente se realiza un análisis de los resultados obtenidos en la verificación. Este análisis puede proporcionar mejoras al proceso las cuales se reflejarán en la documentación BPMN y se incluirán en la distribución iniciando nuevamente el proceso. Adicionalmente se podrá ejecutar la verificación del proceso con los envíos que se encuentran *activos*, es decir los que aún se encuentran en distribución, con el objetivo de encontrar desvíos operativos rápidamente y proceder a su ajuste mejorando la calidad del servicio brindado a los clientes.

3.2. PRODUCTO ELEGIDO

Para el presente trabajo se eligió trabajar con el producto Carta Documento ya que es uno de los productos postales más representativos en Argentina y posee un proceso complejo para su distribución.

3.3. DEFINICIÓN DE LA PRUEBA

El trabajo estará compuesto de las siguientes etapas:

1. Recopilación de datos: Este componente implica la recopilación de datos relevantes del proceso postal, como registros de eventos, logs de procesos, información de seguimiento de envíos, tiempo y fecha, volumen de procesamiento, y otros datos pertinentes al proceso postal.
2. Preparación de datos: En esta etapa, los datos recopilados se limpian, transforman y preparan para su posterior análisis. Esto puede incluir la eliminación de datos duplicados, la normalización de los datos, la corrección de errores, la agregación de datos, y otras tareas de preparación de datos necesarias.
3. Análisis de procesos: En esta etapa, se aplican técnicas de Minería de Procesos para analizar los datos recopilados y descubrir patrones, tendencias, desviaciones y oportunidades de mejora en el proceso postal. Esto puede incluir la identificación de actividades, flujos de trabajo, roles de los participantes, reglas de negocio, tiempos de ejecución, y otras características del proceso

postal.

4. Verificación de conformidad: Como se mencionó anteriormente, la verificación de conformidad es una parte importante de la Minería de Procesos. En esta etapa, se comparan los datos reales del proceso con el modelo de proceso teórico o de referencia para identificar desviaciones y discrepancias.
5. Análisis de rendimiento: En esta etapa, se evalúa el rendimiento del proceso postal en términos de eficiencia, tiempos de ejecución, cuellos de botella, recursos utilizados, y otros indicadores de rendimiento relevantes. Esto puede ayudar a identificar oportunidades de mejora y optimización del proceso.
6. Visualización y presentación de resultados: Los resultados del análisis de procesos y del análisis de rendimiento se visualizan y presentan de forma clara y comprensible para los stakeholders relevantes en el proyecto de Minería de Procesos. Esto puede incluir informes, gráficos, dashboards u otras representaciones visuales de los hallazgos del proyecto.
7. Mejora y optimización del proceso: Basado en los hallazgos del análisis de procesos y del análisis de rendimiento, se pueden identificar oportunidades de mejora y optimización del proceso postal. Estas oportunidades pueden ser implementadas en el proceso real para obtener beneficios y mejoras en la eficiencia y calidad del proceso.

3.4. SOFTWARE UTILIZADO

A continuación se listan las principales herramientas de software utilizadas para la elaboración de esta tesis de maestría. El listado completo se detalla en el Anexo B *Herramientas Utilizadas*

- ProM Tools: Conjunto de herramientas de software de código abierto para el análisis y descubrimiento de procesos en el campo de la Minería de Procesos. Puede realizar análisis y validaciones sobre grandes volúmenes de datos.¹
- RapidMiner: Plataforma de análisis y Minería de Datos de código abierto que permite a los usuarios realizar tareas de análisis de datos, Minería de Datos y Minería de Procesos de manera visual y fácil de usar.²
- Java: Lenguaje de programación orientado a objetos que permite desarrollar aplicaciones independientes de la plataforma, utilizado en la actualidad para desarrollo de aplicaciones de back end.
- IntelliJ Idea: Entorno de desarrollo integrado para lenguaje java y kotlin que permite desarrollar aplicaciones rápidamente y soporta integración con varias herramientas.
- Python: Lenguaje de programación de alto nivel, interpretado, orientado a objetos y multiplataforma cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo.

¹<https://promtools.org/>

²<https://rapidminer.com/>

- PM4PY: Plataforma de código abierto que implementa Minería de Procesos en Python.³
- PostgreSQL: Motor de gestión de bases de datos relacionales orientado a objetos y de código abierto.⁴
- DBeaver Community: Herramienta de Base de Datos universal y gratuita para conectarse a diversos motores de Base de Datos⁵
- Repositorio GIT: se genera un repositorio GIT⁶ en GitHub⁷ para administrar el versionado de los fuentes, gráficos y documentación realizados. El repositorio donde se encuentra todo el material de esta tesis es <https://github.com/vimartinez/UNLP-Tesis>

³<https://pm4py.fit.fraunhofer.de/>

⁴<https://www.postgresql.org/>

⁵<https://dbeaver.io/>

⁶<https://git-scm.com/>

⁷<https://github.com/>

CAPÍTULO 4

Minería de Procesos en la distribución postal

4.1. INTRODUCCIÓN

En el capítulo anterior se definieron las técnicas y metodologías que se utilizaron en este documento para aplicar Minería de Procesos a la Distribución Postal. En este capítulo se explica cómo se aplicaron estas técnicas y se detallan los desafíos encontrados durante el proceso.

Se obtuvieron los datos de entrada de archivos de texto exportados de una base de datos Oracle, se utilizó RapidMiner para realizar una visualización de los datos y verificar que no haya inconsistencias. Se empleó ProM Tools para eliminar trazas incompletas y aplicar las técnicas de Minería de Procesos requeridas según se definió anteriormente.

Los pasos realizados se muestran en la Figura 4.1.

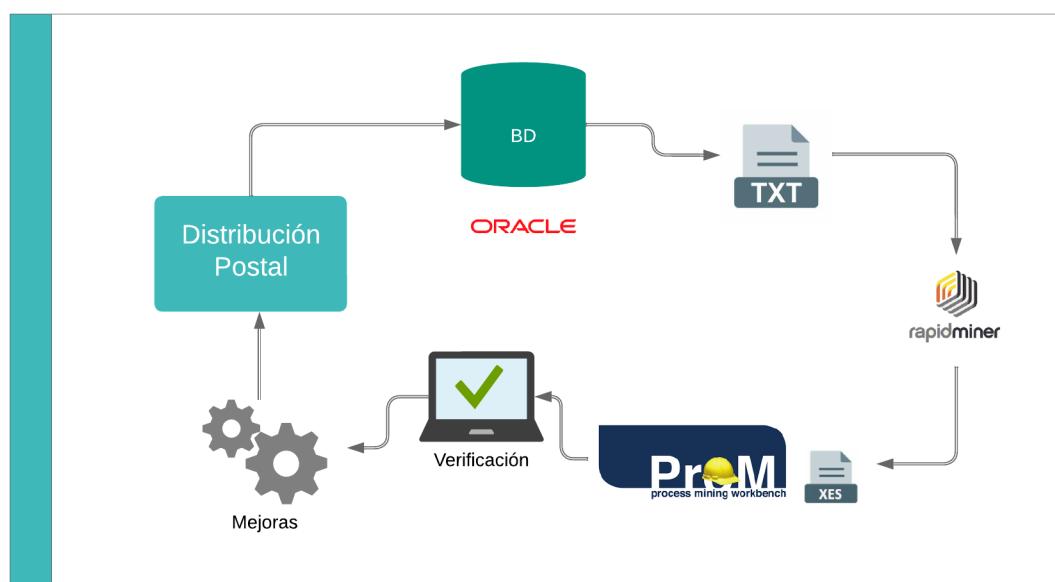


Figura 4.1: Minería de Procesos en la distribución postal

El análisis realizado consta de las siguientes partes:

1. Extracción de los datos: se extrajo la muestra de los archivos de origen. Para este caso de estudio se utilizaron datos productivos de distribución de los años 2017 al 2020.
2. Conversión: Se transformaron los datos a un formato de fácil interpretación para las herramientas de Minería de Procesos.
3. Filtrado: se realizó un procesamiento previo de los datos donde se hizo un muestreo aleatorio y una revisión visual para verificar consistencia. Luego se eliminaron datos inconsistentes y se dividió la muestra en dos tandas, una para descubrimiento y otra para la verificación.
4. Descubrimiento: se aplicaron los algoritmos de Minería de Procesos y se realizó el descubrimiento del modelo actual utilizando una parte de los datos.
5. Verificación: Se realizó la verificación de conformidad del modelo descubierto utilizando la otra parte del set de datos.
6. Análisis: Se realizó un análisis de los desvíos y problemas encontrados en la distribución.
7. Presentación: Se realizaron tableros web que muestran los resultados obtenidos de forma clara y amena para personal no técnico.
8. Mejoras: Se analizaron posibles mejoras al proceso las cuales sirvieron como insumo para el desarrollo de un prototipo de software en el que se incorporaron las técnicas analizadas y utilizadas en este trabajo dentro del proceso productivo.

4.2. EXTRACCIÓN DE LOS DATOS

Los datos de distribución postal se encuentran almacenados en una base de datos Oracle. Para obtener los datos a procesar se realizó una exportación del período entre enero de 2017 hasta diciembre de 2020 ordenados por identificador de envío. Como resultado se obtuvieron más de **9.000.000** de registros que corresponden aproximadamente a **3.750.000** piezas postales distribuidas. Cada uno de esos registros representa un evento de un envío.

Se considera cada envío como una traza y cada movimiento como un evento. La traza se considerará completa cuando el envío tenga registrado el ingreso, y el fin, pudiendo ser la entrega exitosa o no. Se exporta para cada traza los siguientes datos: identificador de traza e identificador de evento, descripción del evento, fecha del evento. El resto de los campos disponibles no aportan valor para el procesamiento que se va a realizar así que no se tienen en cuenta. Para considerar una traza como válida cada traza debe tener al menos dos eventos, uno de inicio y otro de fin. Se descartaron los registros que no cumplían con estas condiciones en un procesamiento posterior.

Originalmente los datos se almacenan según se muestra en la figura [4.2](#).

123 LOGTTID	123 NOTTID	123 PIED	LOGTTFECHA	LOGTTOBSEVAC
99,889,052	9	41,230,484	2018-05-07 12:00:00	PLAZO VENCIDO NO RECL.
99,889,053	1	41,230,487	2018-05-07 13:20:00	ENTREGADO
99,889,054	2	41,230,292	2018-05-04 12:36:00	CERRADO/AUSENTE/1-VISIT
99,889,057	9	41,230,275	2018-05-07 11:50:00	PLAZO VENCIDO NO RECL.
99,889,058	2	41,230,260	2018-05-07 16:00:00	CERRADO/AUSENTE/1-VISIT
99,889,060	2	41,230,025	2018-05-07 11:30:00	CERRADO/AUSENTE/1-VISIT
99,889,061	6	41,230,026	2018-05-07 13:30:00	SE MUDO
99,889,062	9	41,230,490	2018-05-07 10:00:00	PLAZO VENCIDO NO RECL.
99,889,064	1	41,218,737	2018-05-04 10:05:00	ENTREGADO
99,889,065	9	41,218,677	2018-05-07 08:30:00	PLAZO VENCIDO NO RECL.
99,889,066	9	41,218,676	2018-05-07 08:30:00	PLAZO VENCIDO NO RECL.
99,889,067	1	41,219,809	2018-05-03 12:10:00	ENTREGADO
99,889,069	2	41,229,739	2018-05-07 12:30:00	CERRADO/AUSENTE/1-VISIT
99,889,070	1	41,229,736	2018-05-07 10:45:00	ENTREGADO
99,889,071	9	41,229,718	2018-05-07 12:00:00	PLAZO VENCIDO NO RECL.
99,889,072	9	41,229,713	2018-05-07 11:00:00	PLAZO VENCIDO NO RECL.
99,889,073	4	41,229,711	2018-05-07 12:48:00	CERRADO/AUSENTE/2-VISIT
99,889,075	9	41,229,702	2018-05-07 08:53:00	PLAZO VENCIDO NO RECL.
99,889,076	9	41,229,700	2018-05-07 08:53:00	PLAZO VENCIDO NO RECL.
99,889,077	2	41,229,699	2018-05-07 13:33:00	CERRADO/AUSENTE/1-VISIT
99,889,079	1	41,229,698	2018-05-07 13:00:00	ENTREGADO
99,889,081	9	41,222,293	2018-05-07 09:10:00	PLAZO VENCIDO NO RECL.
99,889,082	1	41,222,301	2018-05-04 12:11:00	ENTREGADO
99,889,083	2	41,222,567	2018-05-02 10:00:00	CERRADO/AUSENTE/1-VISIT
99,889,084	4	41,222,317	2018-05-03 14:22:00	CERRADO/AUSENTE/2-VISIT
99,889,085	2	41,222,324	2018-05-04 11:35:00	CERRADO/AUSENTE/1-VISIT

Figura 4.2: Datos Originales

[19]

Los datos originales se exportan de la base de datos como archivos de texto con formato csv para su posterior utilización, dado el gran volumen de datos a procesar se pone un límite de 150 MB para cada archivo ya que se comprobó con operaciones previas que archivos de un tamaño mayor se tornan muy difíciles de trabajar en las etapas posteriores de la Minería de Procesos con los recursos de hardware disponibles para el presente trabajo.

Se importan uno a uno los archivos en RapidMiner¹ para realizar una comprobación visual, no se detecta nada extraño y se verifica que la exportación fue exitosa.

Finalmente los datos originales quedan en 9 archivos de 150 MB cada uno, que totalizan 3.753.495 envíos con 9.105.396 eventos.

4.3. CONVERSIÓN DE DATOS

Una vez generados los archivos como se indicó en la sección anterior, se los procesa uno a uno, utilizando Prom Tools² y se los convierte a formato XES [10].

Prom Tools es un conjunto de herramientas de código abierto para el análisis y la Minería de Procesos. Está diseñado para procesar y analizar datos de eventos registrados en sistemas de información y descubrir información valiosa sobre los procesos comerciales y las actividades dentro de una organización. ProM Tools ofrece una amplia gama de técnicas y algoritmos de Minería de Procesos que permiten descubrir, modelar, analizar y mejorar los procesos comerciales. Algunas de las

¹<https://rapidminer.com/>

²<https://promtools.org/>

funcionalidades incluidas en ProM son el descubrimiento de modelos de procesos, el análisis de conformidad, el cálculo de métricas de rendimiento, la simulación de procesos, la visualización de procesos y el análisis de redes de procesos.

XES (eXtensible Event Stream) es un formato de archivo basado en etiquetas, similar a XML³, que se utiliza para almacenar y compartir datos de eventos. Es el principal formato de archivo utilizado en el ámbito de la Minería de Procesos y la gestión de procesos empresariales. Algunas de las ventajas de usar el formato XES son:

1. Flexibilidad: El formato XES es muy flexible y permite a los usuarios definir sus propios esquemas de datos, lo que significa que puede adaptarse a diferentes tipos de procesos y flujos de trabajo.
2. Interoperabilidad: El formato XES es compatible con una amplia gama de herramientas de software de Minería de Procesos, lo que facilita la interoperabilidad entre diferentes sistemas y la colaboración entre diferentes equipos y organizaciones.
3. Capacidad de registro detallado: El formato XES puede registrar una gran cantidad de información detallada sobre cada evento, lo que puede ayudar a los usuarios a identificar patrones y tendencias en los procesos y a tomar decisiones informadas basadas en los datos.
4. Facilidad de uso: El formato XES es fácil de usar y leer, lo que facilita la comprensión de los datos por parte de los usuarios, incluso si no tienen experiencia en Minería de Procesos.
5. Aplicabilidad: El formato XES se puede utilizar en una amplia gama de aplicaciones, como la gestión de procesos empresariales, la monitorización de sistemas y la gestión de proyectos, lo que lo convierte en una herramienta valiosa para muchas organizaciones.

Durante la conversión, los datos se organizan por traza y con sus eventos en orden cronológico para facilitar su procesamiento en etapas posteriores.

Luego de la conversión se exportan nuevamente los 9 archivos originales. Dichos archivos resultantes de la conversión tienen un tamaño aproximado de 260 MB cada uno debido al agregado de las etiquetas propias del formato XES pero están optimizados para su utilización con las herramientas de Minería de Procesos.

El formato del archivo XES se muestra en la figura 4.3 donde se visualiza que las trazas están claramente delimitadas incluyendo cada uno de los eventos que la componen. Además cada traza y cada uno de sus eventos están anotados como tales.

³<https://www.w3.org/XML/>

```

        </event>
    </trace>
    <trace>
        <string key="concept:name" value="493094"/>
        <event>
            <string key="concept:name" value="INGRESADO"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="EveID" value="0"/>
            <date key="time:timestamp" value="2017-08-17T11:12:55.000-03:00"/>
        </event>
        <event>
            <string key="concept:name" value="1 INTENTO DE ENTREGA"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="EveID" value="2"/>
            <date key="time:timestamp" value="2017-08-23T17:00:00.000-03:00"/>
        </event>
        <event>
            <string key="concept:name" value="1 INTENTO DE ENTREGA"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="EveID" value="2"/>
            <date key="time:timestamp" value="2017-08-28T15:00:00.000-03:00"/>
        </event>
        <event>
            <string key="concept:name" value="1 INTENTO DE ENTREGA"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="EveID" value="2"/>
            <date key="time:timestamp" value="2017-08-29T14:22:00.000-03:00"/>
        </event>
        <event>
            <string key="concept:name" value="DEVOLUCION"/>
            <string key="lifecycle:transition" value="complete"/>
            <string key="EveID" value="9"/>
            <date key="time:timestamp" value="2017-09-13T10:28:00.000-03:00"/>
        </event>
    </trace>
    <trace>
        <string key="concept:name" value="493152"/>
        <event>
            <string key="concept:name" value="INGRESADO"/>

```

Figura 4.3: Datos en formato XES

[19]

Luego de la conversión los datos originales quedan en 9 archivos de aproximadamente 260 MB cada uno, manteniendo los totales de 3.753.495 envíos con 9.105.396 eventos.

4.4. FILTRADO

Para tener una mayor consistencia en los datos y facilitar el análisis se eliminaron todas las trazas incompletas, ya sea porque no tienen un estado inicial (ingreso) o un estado final (entregado, devuelto, no existe dirección, falleció, o se mudó). Esto puede ocurrir por algún error en la carga de los datos o porque son trazas que todavía no completaron el proceso.

Esta tarea se realiza en ProM importando cada uno de los archivos de datos en formato XES que se obtuvieron en la etapa anterior y aplicando un filtro de reglas heurísticas simples para quitar las trazas incompletas. Para ello el algoritmo escanea el archivo y permite seleccionar cuales son los eventos iniciales válidos de todos los eventos iniciales que se detectaron en las trazas. Luego debemos

realizar la misma selección pero con los eventos finales. Basándose en estos datos de entrada se procede a descartar todas las trazas que no cumplan estas condiciones, obteniendo como resultado un nuevo archivo en formato XES que contiene únicamente las trazas que se utilizarán en este caso de estudio reduciendo considerablemente el ruido en los datos de entrada.

La utilización de este filtro se visualiza en la figura 4.4 donde se muestra seleccionado INGRESADO como único estado inicial válido en las trazas y los estados definitivos previamente mencionados como finales.

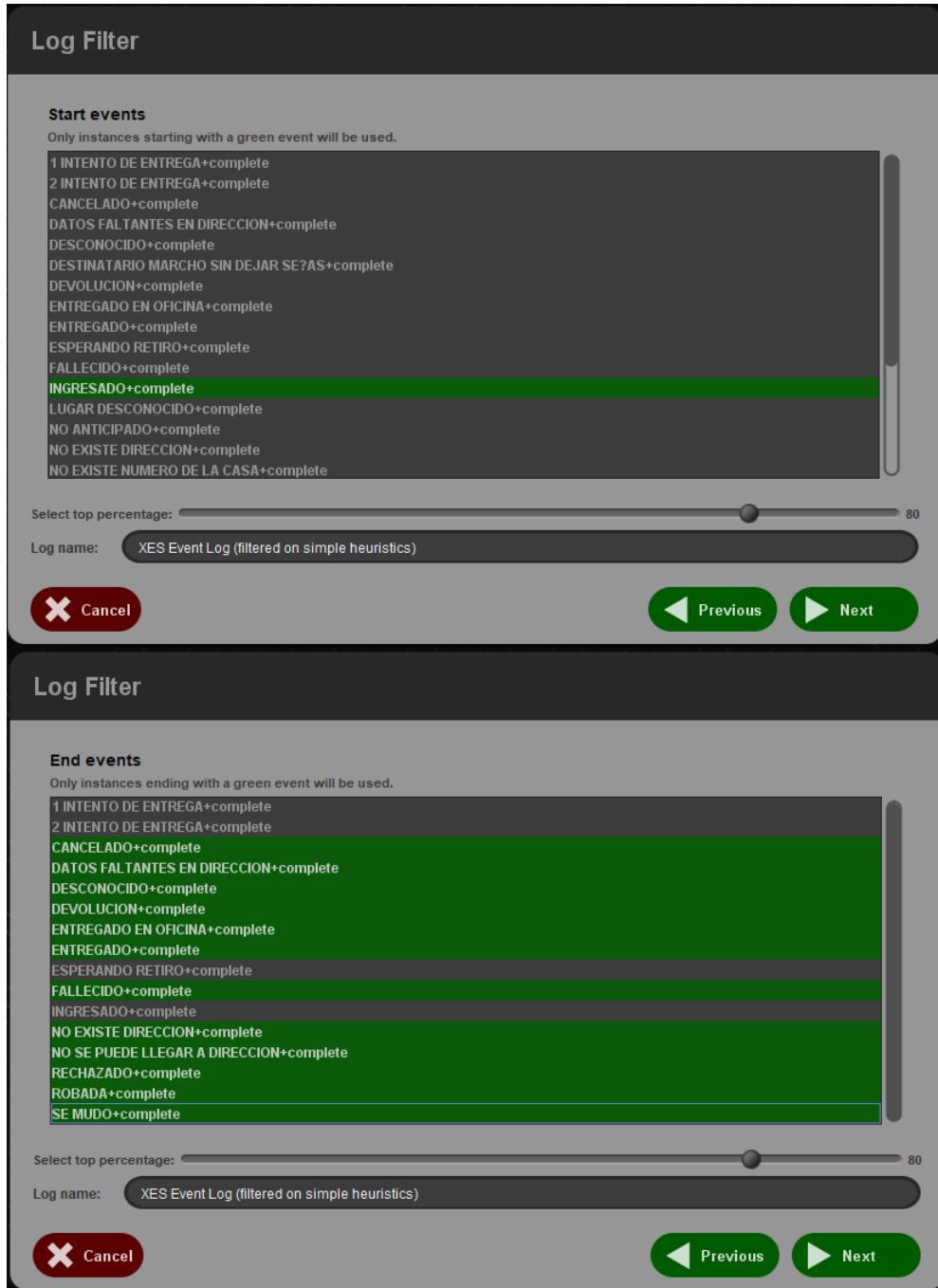


Figura 4.4: Filtrado de archivos XES en ProM

El proceso de filtrado se realiza con cada uno de los archivos obtenidos en el paso anterior descartando trazas incompletas y dejando los datos de entrada listos para utilizar las técnicas de Minería de Procesos.

El detalle de cada archivo se visualiza en la tabla 4.1

Tabla 4.1: Detalle de trazas y eventos utilizados

Archivo	Trazas Originales	Eventos Originales	Trazas Filtradas	Eventos Filtrados
1	469.288	1.115.026	433.256	1.041.854
2	466.271	1.096.119	427.240	1.018.169
3	455.551	1.089.551	419.241	1.015.692
4	455.931	1.088.746	430.914	1.025.856
5	432.879	1.078.439	405.369	981.950
6	400.576	1.040.031	354.174	908.407
7	403.035	1.041.523	371.039	919.938
8	398.503	1.044.665	357.374	913.666
9	271.461	511.296	258.325	452.680
TOTAL	3.753.495	9.105.396	3.456.932	8.278.212

[19]

Luego de este filtrado los archivos resultantes tienen aproximadamente 230 Mb cada uno y totalizan 3.456.932 trazas con un total de 8.278.212 eventos.

4.5. DESCUBRIMIENTO DEL PROCESO

El proceso de distribución postal que deben cumplir los envíos que se utilizaron en este trabajo se describió anteriormente y se modeló según BPMN como se muestra en la figura 4.5

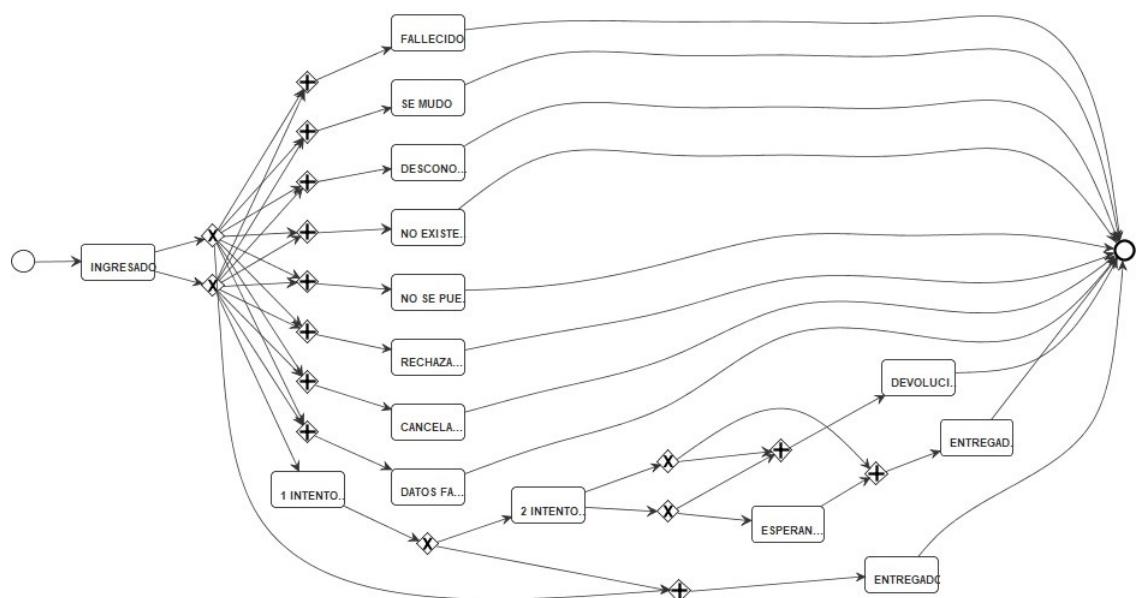


Figura 4.5: Proceso de Distribución Postal modelado en BPMN

[19]

Este proceso fue validado por personal de *Ingeniería de Procesos* y se verificó que cumple con las características del producto utilizado en este trabajo.

Para realizar el descubrimiento del proceso que cumplen las piezas se utilizó el algoritmo Alpha+ dado que evidenció un correcto funcionamiento en todas las pruebas realizadas, ajustándose adecuadamente a este tipo de datos.

Tanto el algoritmo *Alpha* como *Alpha+*[28] son ampliamente usados en Minería de Procesos para descubrir procesos de negocio a partir de registros de eventos.

El algoritmo Alpha busca descubrir modelos de proceso basados en la estructura de flujo de eventos. El proceso de descubrimiento del modelo comienza con la creación de un grafo dirigido que representa la estructura de los eventos compuesto de lugares y transiciones, los lugares son estados o condiciones del sistema con comportamiento similar y las transiciones son eventos o acciones que pueden ocurrir. Seguido a esto se eliminan las transiciones redundantes, y se agrupan las transiciones similares para formar nuevos lugares en el modelo. Finalmente, se fusionan lugares similares para obtener un modelo de proceso simplificado. Los pasos son los siguientes:

1. Creación del grafo de eventos: En este paso se crea un grafo dirigido que representa la estructura de flujo de eventos según lo registrado en el archivo de log. Los nodos del grafo representan las actividades del proceso y los arcos representan la relación de orden entre las actividades.
2. Eliminación de transiciones redundantes: En este paso, se eliminan las transiciones redundantes del grafo. Una transición se considera redundante si su eliminación no afecta la estructura del grafo. Esto se hace para simplificar el modelo y reducir la complejidad.
3. Agrupación de transiciones similares: En este paso, se agrupan las transiciones que tienen un comportamiento similar para formar lugares en el modelo. Un lugar representa un conjunto de actividades que tienen un comportamiento similar. Esto se hace para simplificar aún más el modelo.
4. Fusión de lugares similares: En este paso, se fusionan los lugares que tienen un comportamiento similar para obtener un modelo de proceso simplificado. La fusión de lugares se basa en la similitud de las actividades y los arcos que los conectan.

Por otro lado, el algoritmo *Alpha+* mejora el algoritmo *Alpha* al tener en cuenta los tiempos de los eventos. Además de la estructura de flujo de eventos, también considera los tiempos de los eventos para descubrir un modelo de proceso que refleje las relaciones de orden entre los eventos [29].

El descubrimiento es el primer tipo de Minería de Procesos de los que se describieron anteriormente.

El proceso fue encontrado aplicando el algoritmo Alpha+ a la muestra de datos. Este algoritmo representa el proceso descubierto como una *Red de Petri*.

Una red de Petri es un modelo matemático y gráfico que permite representar sistemas distribuidos y concurrentes. Se compone de dos elementos principales: lugares y transiciones. Un lugar representa una condición o estado del sistema. Los lugares se representan como círculos en el diagrama y pueden contener tokens, que representan la existencia o disponibilidad de un recurso o condición,

por ejemplo stock de un producto. Una transición representa un evento o acción que cambia el estado del sistema. Las transiciones se representan como barras en el diagrama y se activan cuando hay suficientes tokens en los lugares de entrada correspondientes. Al activarse, una transición consume los tokens de los lugares de entrada y produce tokens en los lugares de salida correspondientes.

En la Figura 4.6 se muestra la red de Petri descubierta que representa el proceso que deberán cumplir las trazas.

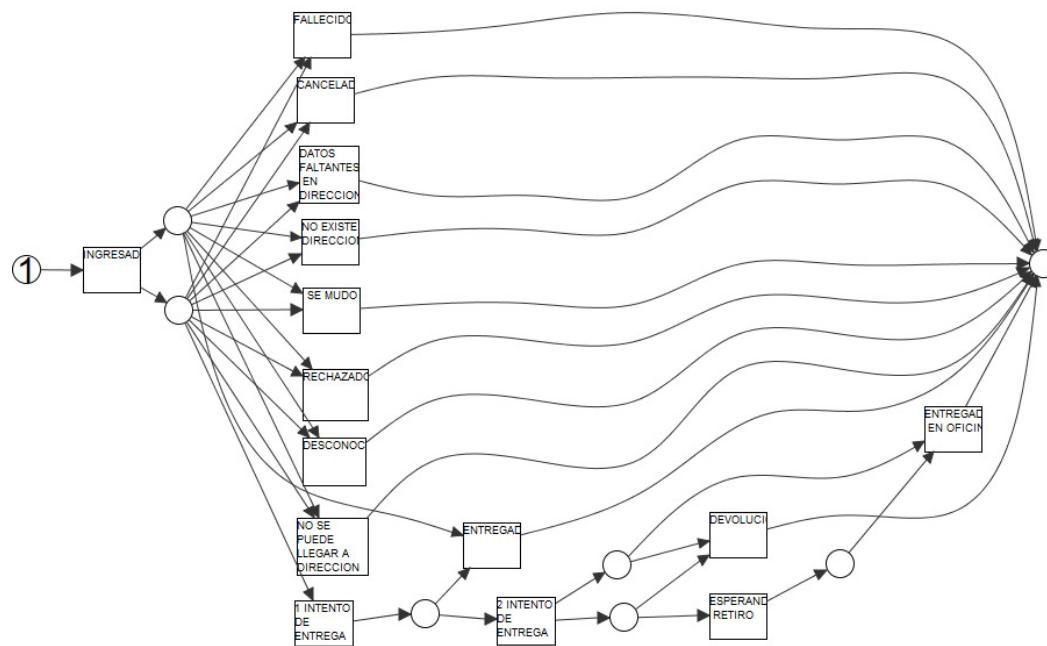


Figura 4.6: Proceso descubierto con Alpha+ que deberán cumplir las trazas

[19]

El proceso descubierto se conforma de la siguiente manera:

- Siempre hay un evento de ingreso que deben cumplir todas las trazas luego del cual se realiza la distribución.
- Si se puede entregar, se registra la entrega y finaliza la historia de la traza.
- Si no se puede entregar por un evento final, por ejemplo persona fallecida, no existe la dirección o no hay suficientes datos para encontrar la dirección, se registra la no entrega con el motivo y finaliza el proceso.
- Si el motivo de no entrega no es un evento final, por ejemplo no había nadie para entregar, se registra un primer intento de entrega y se intenta nuevamente al día siguiente.
- Si al día siguiente tampoco se puede entregar, se mantiene el envío en la oficina durante un tiempo esperando que el destinatario se presente a retirar dejando un aviso luego de cada visita.
- Si el destinatario pasa a retirar el envío por la sucursal, se registra la entrega y finaliza el proceso.

- Si el destinatario no pasa a retirar el envío luego de un tiempo especificado, usualmente dos semanas, se devuelve el envío al remitente y finaliza el proceso.

Una vez finalizada la generación del proceso, se lo exporta desde ProM para su uso en etapas posteriores.

4.6. VERIFICACIÓN DE CONFORMIDAD

La verificación de conformidad consiste en contrastar cada uno de los archivos de datos filtrados contra el modelo descubierto previamente.

El objetivo es determinar cuánto se ajustan las trazas al proceso, es decir qué tanto cumplen las etapas requeridas. Cuanto más se acerque la traza al proceso, en cantidad de pasos y orden, mayor será su ajuste o fitness con respecto al mismo. Por el contrario cuanto menos se acerque al proceso, ya sea porque hay pasos sobrantes u omitidos o porque no se realizan en el orden correcto, la traza tendrá un ajuste menor. Por lo tanto para este análisis se tuvo en cuenta qué pasos se cumplieron, cuáles no, si hubo repetidos o adicionales que no estuvieran reflejados en el proceso descubierto y su correcta sucesión.

A continuación se reprodujeron los archivos de datos sobre el proceso descubierto anteriormente por medio del comando “*Replay a Log on Petri Net for Conformance Analysis*”. La Figura 4.7 ejemplifica una de estas reproducciones. Allí se resaltan los eventos más comunes en color oscuro y los recorridos más frecuentes con una línea más gruesa.

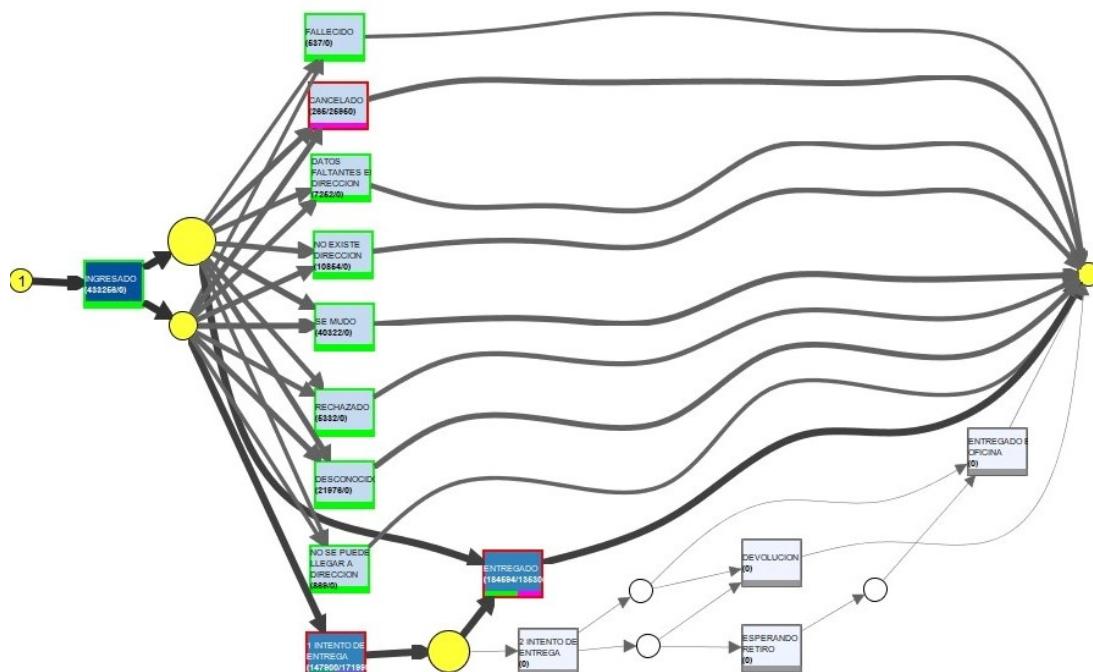


Figura 4.7: Recorrido de las trazas: eventos más comunes en oscuro y recorridos más frecuentes en línea más gruesa

[19]

Luego de procesar todos los archivos por la Red de Petri se analizaron los resultados verificando las

estadísticas de los datos.

Inicialmente se observó que en un **73 %** de los casos el proceso finaliza con la pieza entregada, ya sea en el primer o segundo intento y que el resto se distribuye uniformemente entre los diversos motivos de no entrega. Esto no implica un mal funcionamiento ya que la no entrega es parte del proceso; sólo muestra una característica del producto Carta Documento.

Se observó además que en todo el conjunto de datos las trazas tienen una media de **2,37** eventos en cada traza y un ajuste con respecto al proceso de **0,72**. Estos valores son similares a los encontrados en los trabajos publicados [20] y [21] que dan soporte a esta tesis.

Se observó también que muchas trazas no verificaban el proceso exactamente ya sea por no cumplir con los pasos en el orden correcto o por repetir algunos. Un ejemplo de esto se muestra en la figura 4.8. Utilizando el visualizador de trazas de ProM, allí se pueden ver trazas con muchos movimientos lo que implica un ajuste muy bajo (entre 0,23 y 0,33) con respecto al proceso. En base a estos resultados, se consideró razonable establecer un valor umbral para el ajuste que deben cumplir las trazas con respecto al proceso para no ser consideradas casos sospechosos. Dicho valor fue fijado en 0,5.



Figura 4.8: Visualización de ProM de trazas con exceso de movimientos y ajuste muy bajo

[19]

Con el objetivo de analizar con mayor detalle las trazas que tienen un ajuste más bajo del umbral establecido, se exportaron de cada archivo aquellas trazas con un ajuste menor al **50 %** por considerarlas un gran desvío operativo y se consolidan en un único archivo para un análisis posterior. Como resultado, se encontró un total de **374.876** trazas en estas condiciones.

Como se mencionó anteriormente, la cantidad de movimientos por traza sigue una distribución normal con una media **2,37** y un desvío estándar de **0,95**. Por lo tanto se decidió tomar un valor superior a la *media mas tres desvíos estándar* como un valor representativo de una cantidad excesiva de movimientos; esto implicó tener en cuenta las trazas con más de **5,22** movimientos. En base a esto, se procedió a filtrar y exportar las trazas con más de **6** movimientos para realizar un análisis posterior encontrando un total de **1.724** trazas en estas condiciones.

Adicionalmente realizó un análisis de tiempos de finalización sobre las trazas del modelo en busca de aquellas que, si bien pudieron haber completado el proceso correctamente, el mismo se realizó en un intervalo de tiempo extremadamente superior a la media de resolución de cada traza. Para ello se efectuó nuevamente la reproducción de cada uno de los archivos de datos sobre el proceso descubierto anteriormente en ProM pero en este caso con un algoritmo orientado al análisis de performance llamado *“Replay a Log on Petri Net for Performance Analysis”*, que se enfoca más en los tiempos de resolución y no tanto en ver si las trazas se ajustan o no al proceso.

La figura 4.9 muestra una visualización obtenida luego de ejecutar esta herramienta representando en tonos más oscuros las tareas que demoran más tiempo en realizarse y en tonos más claro las que finalizan más rápidamente.

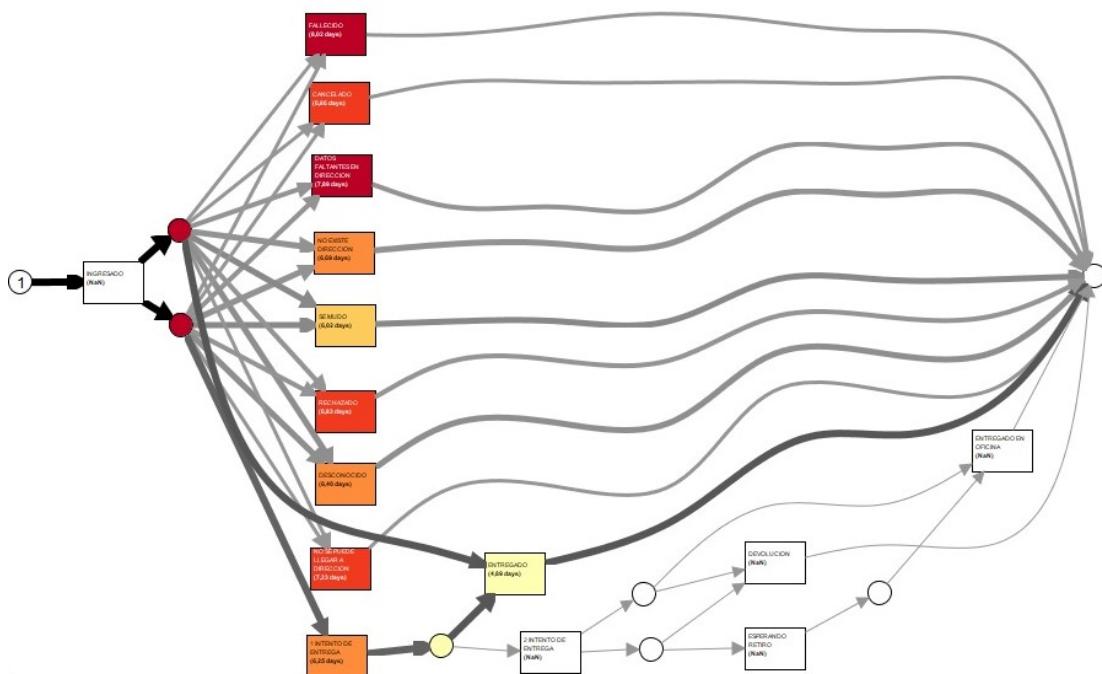


Figura 4.9: Visualización de trazas según análisis de Performance

[19]

Analizando los resultados se encontraron trazas que se ajustan bien al proceso pero con tiempos extremadamente altos. En algunos casos las trazas tienen más de tres movimientos y un intervalo de hasta 3 días entre cada uno de ellos lo que justifica un tiempo total tan elevado.

Adicionalmente se encontraron muchos casos que contaban solo con dos movimientos, donde el envío ingresó, estuvo sin actividad un tiempo prolongado y luego cumplió correctamente con una entrega o devolución. Estos casos poseen un buen ajuste al proceso descubierto pero representan un desvío operativo importante ya que impactan directamente en la calidad del servicio a los clientes incumpliendo los estándares de entrega del producto y son extremadamente difíciles de detectar debido a que la operación finaliza correctamente.

En la figura 4.10 se observan tres casos que cumplen el proceso correctamente con solo dos movimientos pero con una diferencia de 10 días o más entre uno y otro. El primero ingresó el 11/12/2017

y se entregó el 21/12/2017, el segundo ingresó el 19/07/2019 y se entregó el 29/07/2019 y el tercero ingresó el 15/08/2019 y se entregó el 26/08/2019. Con el análisis de performance se descubrieron nuevos desvíos operativos que no aparecían en el análisis anterior ya que se trata de trazas que se ajustan perfectamente al modelo pero que poseen tiempos muy por encima de la media.

2 events	
INGRESADO	#1 complete
	11.12.2017 08:43:35.000
ENTREGADO	#2 complete
	21.12.2017 19:14:00.000
3306727	2 events
INGRESADO	#1 complete
	19.07.2019 19:15:45.000
ENTREGADO	#2 complete
	29.07.2019 12:20:00.000
3451408	2 events
INGRESADO	#1 complete
	15.08.2019 08:34:55.000
ENTREGADO	#2 complete
	26.08.2019 10:23:00.000

Figura 4.10: Trazas de dos movimientos con inactividad

[19]

En base a las observaciones realizadas, se determinó un tiempo medio de resolución para cada traza de **5,79** días, con un desvío de **4,96** días. Utilizando un criterio similar al usado en la cantidad de movimientos por traza se decidió enfocar el análisis en aquellos casos con un tiempo de finalización mayor a la media más un desvío por considerar que un tiempo de resolución mayor a este valor se trata de un desvío operativo. En particular, para el caso de estudio considerado en esta tesis, se ajustó ese valor a 10,75 días. Luego de realizar la exportación de las trazas que cumplen con ese criterio y se obtiene un total de **595.880** casos.

El resultado de la verificación del proceso mostró una gran cantidad de posibles desvíos operativos a verificar. El detalle de las irregularidades encontradas en cada archivo se visualiza en la tabla 4.2

A continuación se analizarán en detalle los diferentes desvíos encontrados.

4.7. PROBLEMAS ENCONTRADOS

Los resultados de la verificación de conformidad realizada anteriormente proporcionaron una evaluación objetiva de cómo el modelo de proceso descubierto se ajusta a los datos reales, es decir muestran que está pasando en la operación y permiten identificar posibles mejoras en el proceso real.

Tabla 4.2: Detalle de irregularidades encontradas

Archivo	Trazas	Ajuste menor a 0,5	Más de 6 movimientos	Más de 10 días
1	433.256	35.729	91	64.397
2	427.240	38.395	51	107.722
3	419.241	46.642	88	76.782
4	430.914	41.215	76	76.627
5	405.369	60.049	324	101.26
6	354.174	57.295	338	50.271
7	371.039	55.949	257	52.728
8	357.374	20.063	347	33.359
9	258.325	19.539	152	32.727
TOTAL	3.456.932	374.876 (10,84 %)	1.724 (0,05 %)	595.880 (17,24 %)

[19]

En este caso particular se encontró que para los envíos realizados entre enero de 2017 y Diciembre de 2020 el **28,13 %** son posibles desvíos operativos, lo que representa una gran cantidad en términos de volumen y porcentaje, abriendo una amplia gama de posibilidades para realizar algunos ajustes que permitan mejorar la calidad del servicio a los clientes.

Estos inconvenientes pueden deberse a desviaciones menores donde algunas trazas puntuales no cumplen con exactitud el proceso o también a desviaciones significativas de las trazas lo que implica incumplimientos graves en el proceso que requieren de inmediata atención y medidas predictivas para evitar que se repitan en el futuro.

Por lo tanto, se decidió analizar en detalle estos casos realizando un muestreo de las trazas exportadas en cada caso para verificar si realmente son problemas en la entrega del producto y de ser así conocer el motivo. Se analizaron tres tipos de inconvenientes, trazas que se ajustan muy poco al proceso, trazas que tienen un ajuste mejor pero que tienen gran cantidad de movimientos y trazas que ajustan bien pero que finalizan el proceso en un intervalo de tiempo muy elevado.

El resumen de las irregularidades encontradas sobre el total de **3.456.932** trazas procesadas se observa en la tabla **4.3**

Tabla 4.3: Irregularidades encontradas

Irregularidad	Cantidad	Porcentaje
Trazas con ajuste menor a 0,5	374.876	10,84 %
Trazas con más de 6 movimientos	1.724	0,05 %
Trazas que demoraron más de 10 días	595.880	17,24 %
TOTAL	972.480	28,13 %

[19]

4.7.1. Trazas con ajuste menor a 0.5

Como se mencionó anteriormente el ajuste dentro de la Minería de Procesos se refiere a una medida que cuantifica la calidad o adecuación de los datos reales al modelo de proceso descubierto. Es una métrica que indica qué tan bien se ajustan entre sí los datos observados y el modelo descubierto. Cuanto mayor sea el ajuste, mayor será la conformidad entre el modelo y los datos reales, lo que

indica que el modelo es una representación precisa del proceso observado.

Se realizó un muestreo estratificado de 300 elementos entre las 374.876 trazas que tienen un ajuste menor a 0,5 respecto del modelo, dividiendo las mismas en tres estratos y analizando puntualmente los casos seleccionados.

Los estratos se formaron de la siguiente manera:

- Estrato 1: Ajuste hasta 0,2.
- Estrato 2: Ajuste entre 0,2 y 0,4.
- Estrato 3: Ajuste mayor a 0,4.

Se encontraron algunos casos que representan desvíos menores pero el proceso logró continuar y finalizar correctamente. Un ejemplo de esto es la repetición de un evento que, si bien es incorrecto, no provoca una distorsión excesiva en la secuencia de la trama. La figura 4.11 ejemplifica esta situación al mostrar tres trazas que repiten el evento *1 intento de entrega* pero el proceso continúa correctamente. Posiblemente esto se deba a un error de carga, donde se cargó nuevamente el primer intento en lugar del segundo, ya que las fechas de los eventos duplicados son posteriores al original. De todas maneras el producto se entrega o se devuelve respetando la esencia del proceso postal.



Figura 4.11: Trazas con desvíos leves

[19]

También se encontraron varios casos con desvíos graves y varias inconsistencias en el proceso; estos casos requieren una mayor atención ya que no se está cumpliendo correctamente el servicio.

La Figura 4.12 ilustra estas situaciones en las que se observan casos que hay eventos que no pueden estar en la misma traza.

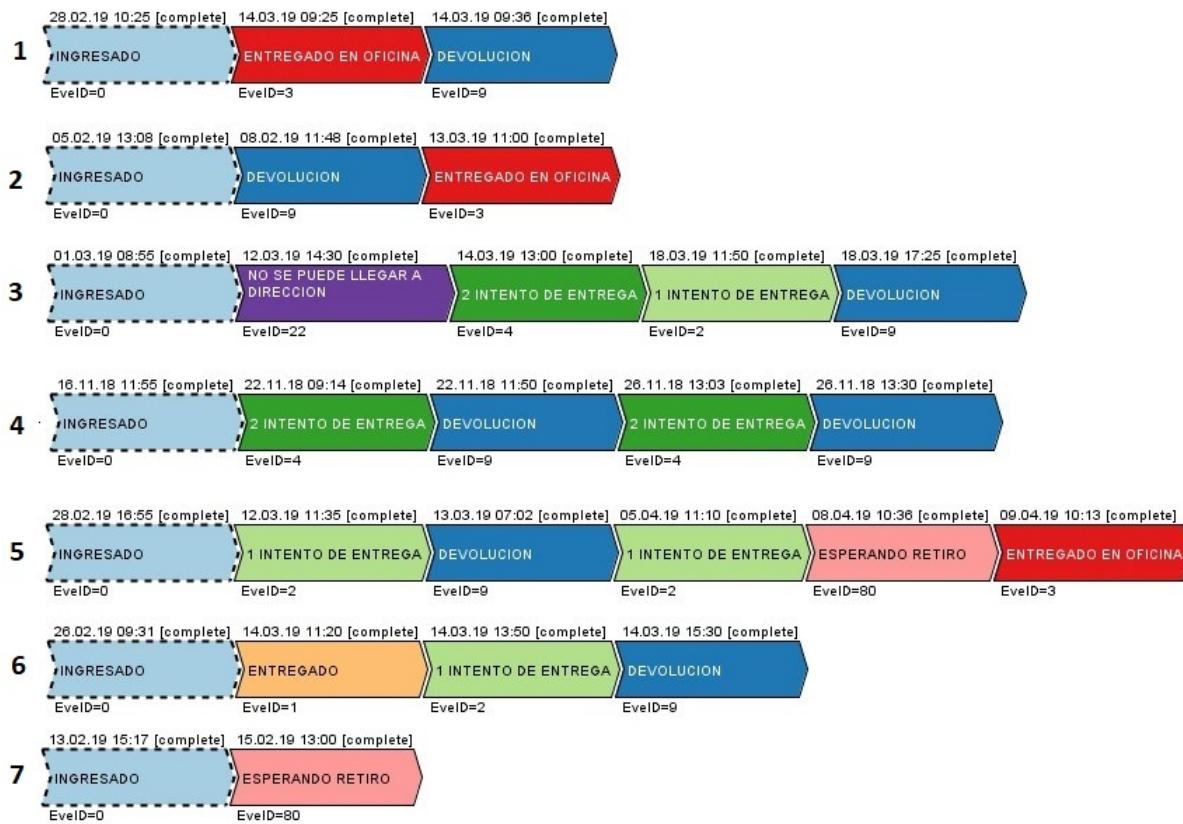


Figura 4.12: Trazas con desvíos graves

[19]

Se detallan a continuación los casos que se muestran:

1. Ingresado, entregado en oficina y devuelto.
2. Ingresado, devuelto y entregado en oficina.
3. Ingresado, no se puede llegar a la dirección, segundo intento de entrega, primer intento de entrega y devolución.
4. Ingresado, segundo intento de entrega, devolución, otro segundo intento de entrega y una segunda devolución.
5. Ingresado, primer intento de entrega, devolución, otro primer intento de entrega, esperando retiro y entregado.
6. Ingresado, entregado, primer intento de entrega, devolución.
7. Ingresado, esperando retiro y no hay nuevos movimientos.

Claramente son problemas graves ya que por ejemplo entregado y devuelto es imposible que se cumplan para un envío porque no se puede entregar el producto al destinatario y devolverlo al remitente.

Todas estas inconsistencias graves del proceso deben ser revisadas en tiempo y forma para tomar

acciones correctivas a fin de evitar que se omitan eventos realizados o que se carguen eventos inexistentes.

4.7.2. Trazas con exceso de movimientos

Para realizar el análisis sobre los casos que tenían muchos movimientos se realizó un muestreo aleatorio simple debido a la baja cantidad de casos con estas características. A pesar de ser solo el 0,05 % se encontraron algunos casos con inconsistencias graves y movimientos contradictorios por ejemplo sucesivas entregas y devoluciones del mismo envío en diferentes fechas.

Algunos de los casos más representativos se muestran en la figura 4.13 donde se observan diferentes irregularidades.

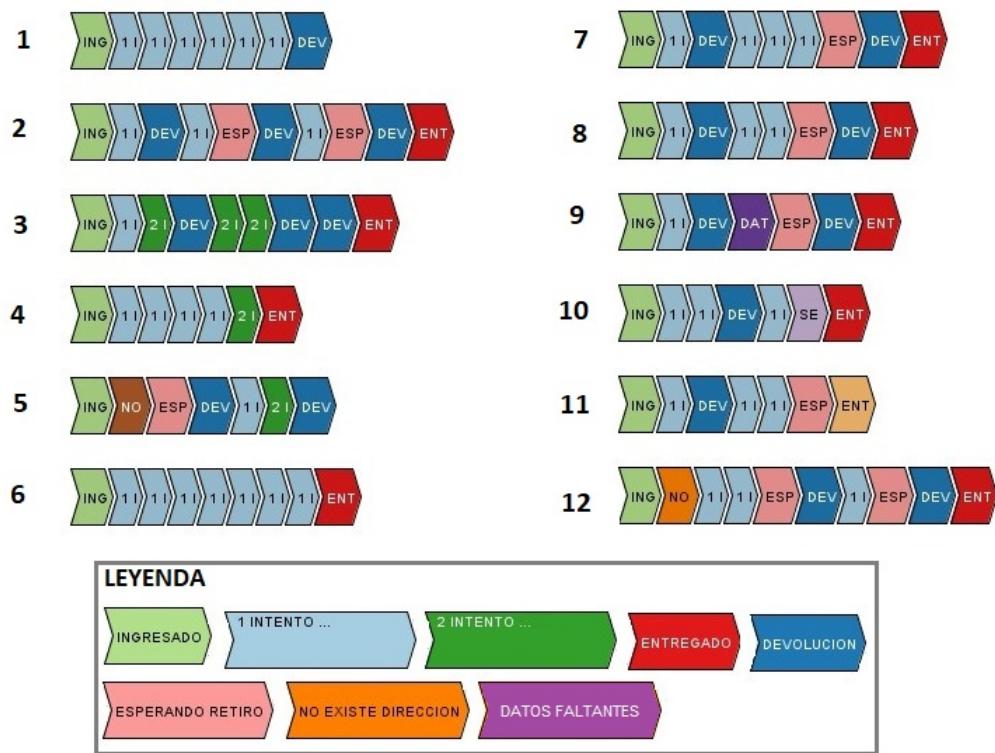


Figura 4.13: Trazas exceso de movimientos

[19]

A continuación se explican en detalle los eventos de los casos mostrados en la figura 4.13:

1. Ingreso, seis intentos de primer entrega y devolución.
2. Ingreso, primer intento de entrega, devolución, primer intento de entrega, espera en sucursal, otra devolución, otro primer intento de entrega, espera en sucursal, otra devolución, y finalmente entregado. Es imposible devolver más de una vez un envío y luego entregarlo.
3. Ingreso, primer intento de entrega, segundo intento de entrega, devolución, luego dos veces segundo intento de entrega, dos devoluciones y finalmente entregado.

4. Ingreso, cuatro primer intento de entrega, un segundo intento de entrega y finalmente entregado.
5. Ingreso, no se pudo llegar a domicilio, esperando retiro, devolución, luego un primer y un segundo intento de entrega y finalmente otra devolución.
6. Ingreso, siete primer intento de entrega y luego entregado.
7. Ingreso, primer intento de entrega, devolución, luego tres primer intento de entrega, espera en sucursal, otra devolución y finalmente entregado.
8. Ingreso, primer intento de entrega, devolución, nuevamente primer intento de entrega dos veces, espera en sucursal, devolución y entregado
9. Ingreso, primer intento, devolución, datos faltantes en dirección, esperando retiro, otra devolución y entregado.
10. Ingreso, dos veces primer intento de entrega, devolución, otro primer intento de entrega, un segundo intento y luego entregado.
11. Ingreso, primer intento de entrega, devolución, nuevamente dos veces primer intento de entrega, esperando en sucursal y entregado.
12. Ingreso, no existe dirección, dos veces primer intento de entrega, esperando en sucursal, devolución, nuevamente un primer intento de entrega, otra vez esperando en sucursal, una segunda devolución y finalmente entregado.

En estos ejemplos se muestran graves inconsistencias en los datos que reflejan el incumplimiento del proceso y peor aún dejan en evidencia que no se cuenta con información fidedigna de lo que pasó con el envío. En más de un ejemplo se ven sucesivas devoluciones y entregas de una misma pieza postal en sucesivos días, movimientos que no solo no representan el proceso sino que son contradictorios y no podrían ocurrir con el mismo envío, por ejemplo una carta no puede ser entregada y devuelta a la vez, o devuelta varias veces en días sucesivos.

En el análisis de trazas con exceso de movimientos se encontraron graves incumplimientos del proceso.

4.7.3. Trazas que demoraron mucho tiempo

Finalmente se realizó el análisis en las trazas que tuvieron un excesivo tiempo para su finalización. Para ello se realizó también un muestreo estratificado entre las 595.880 trazas que tardaron más de 10 días en completar el proceso y dividiendo las mismas en tres estratos de la siguiente forma:

- Estrato 1: Hasta 20 días de duración (493.241 trazas)
- Estrato 2: Entre 20 y 40 días de duración (91.687 trazas)
- Estrato 3: Más de 40 días de duración (10.952 trazas)

La figura 4.14 muestra algunos ejemplos de los estratos 1 y 2, donde se observa que el proceso se cumple correctamente pero el envío permanece hasta más de 10 días sin actividad. Esto se repite en la mayoría de los casos analizados, aunque también se encontraron trazas con exceso de movimientos, verificando que varias de las que se analizaron en la sección 4.7.2 también estaban en este conjunto de datos.

2885916 2 events	2885943 2 events	2885934 2 events
INGRESADO #1 complete 01.03.2019 16:32:39.000	INGRESADO #1 complete 01.03.2019 16:25:17.000	INGRESADO #1 complete 01.03.2019 16:27:55.000
ENTREGADO #2 complete 21.03.2019 09:57:00.000	ENTREGADO #2 complete 18.03.2019 14:20:00.000	DATOS FALTANTES EN DIRECCION #2 complete 18.03.2019 14:00:00.000
2885918 2 events	3196771 4 events	3031122 4 events
INGRESADO #1 complete 01.03.2019 16:29:59.000	INGRESADO #1 complete 01.07.2019 13:52:08.000	INGRESADO #1 complete 09.05.2019 09:44:20.000
ENTREGADO #2 complete 14.03.2019 12:52:00.000	1 INTENTO DE ENTREGA #2 complete 12.07.2019 13:37:00.000	1 INTENTO DE ENTREGA #2 complete 27.05.2019 11:00:00.000
2885923 2 events	ESPERANDO RETIRO #3 complete 15.07.2019 08:03:31.000	ESPERANDO RETIRO #3 complete 30.05.2019 07:43:54.000
INGRESADO #1 complete 01.03.2019 16:29:54.000	DEVOLUCION #4 complete 30.07.2019 14:09:04.000	ENTREGADO EN OFICINA #4 complete 31.05.2019 13:15:01.000
SE MUDO #2 complete 14.03.2019 11:02:00.000		

Figura 4.14: Trazas con excesivo tiempo de finalización

[19]

La figura 4.15 muestra algunos ejemplos del estrato 3 donde se encontraron casos con tiempos extremadamente altos de hasta 8 meses de duración y que cumplen el proceso. Por ejemplo está el caso 3059379 que ingresa el 09/05/2019 y permanece sin movimientos hasta el 24/06/2019 donde sale por primera vez a distribución y no se pudo entregar registrando el primer intento de entrega. Luego no registra actividad hasta el 2/10/2019 donde se ingresa un segundo intento de entrega y una devolución al remitente el mismo día. Este caso particular registra una demora de casi 5 meses. Otro ejemplo es el 3742021 que ingresa el 12/12/2019 y se ajusta correctamente al proceso registrando un primer intento de entrega, un segundo intento de entrega y la devolución pero finalizando el 06/05/2020.

3991130 3 events	3742021 4 events	4066783 5 events
INGRESADO #1 complete 30.03.2020 14:19:10.000	INGRESADO #1 complete 12.12.2019 18:56:33.000	INGRESADO #1 complete 24.04.2020 16:38:13.000
1 INTENTO DE ENTREGA #2 complete 02.06.2020 12:02:00.000	1 INTENTO DE ENTREGA #2 complete 07.02.2020 09:00:00.000	1 INTENTO DE ENTREGA #2 complete 06.05.2020 14:50:00.000
DESCONOCIDO #3 complete 30.07.2020 12:00:00.000	2 INTENTO DE ENTREGA #3 complete 18.02.2020 09:45:00.000	2 INTENTO DE ENTREGA #3 complete 07.05.2020 13:00:00.000
3059379 4 events	DEVOLUCION #4 complete 06.05.2020 11:00:00.000	DEVOLUCION #4 complete 08.07.2020 12:00:00.000
INGRESADO #1 complete 09.05.2019 18:09:28.000	3656878 3 events	DEVOLUCION #5 complete 08.07.2020 13:00:00.000
1 INTENTO DE ENTREGA #2 complete 24.06.2019 18:40:00.000	INGRESADO #1 complete 26.11.2019 18:43:55.000	
2 INTENTO DE ENTREGA #3 complete 02.10.2019 11:10:00.000	1 INTENTO DE ENTREGA #2 complete 19.12.2019 11:15:00.000	
DEVOLUCION #4 complete 02.10.2019 11:15:00.000	DEVOLUCION #3 complete 06.05.2020 09:40:00.000	

Figura 4.15: Trazas con tiempo extremadamente alto de finalización

[19]

4.8. RESULTADOS OBTENIDOS

Con el aporte realizado en los trabajos previos [20] y [21] y el presente análisis aplicando Minería de Procesos a la distribución postal se verifica que aplicando estas técnicas se puede visualizar el cumplimiento del proceso en detalle y observar inconvenientes de forma más precisa.

Entre otras cosas se encontraron desvíos considerables al proceso operativo, inconsistencias en la información cargada y posteriormente entregada a los clientes y casos en los que a pesar de que el proceso se cumplió correctamente los envíos se demoraron de forma excesiva sin ningún motivo aparente.

La posibilidad de detectar desvíos operativos hasta el momento ignorados verifica que es posible aplicar las técnicas de minería de procesos a la distribución postal en la República Argentina y que su implementación puede brindar información que podría utilizarse para mejorar la calidad del servicio.

Adicionalmente se puede generar una herramienta para automatizar el análisis sobre los envíos que se encuentran en proceso de distribución y presentar los resultados de forma clara a un operador para que verifique estos posibles desvíos y de ser necesario tome las acciones necesarias.

En el próximo capítulo se diseñará y se desarrollará un prototipo que permita incorporar estas herramientas en la operación diaria alertando al personal responsable de posibles desvíos operativos en tiempo y forma para que puedan tomar medidas predictivas o correctivas.

CAPÍTULO 5

Solución Propuesta

5.1. INTRODUCCIÓN

En este capítulo se explica el diseño y desarrollo del prototipo completamente funcional que se realizó aplicando las técnicas de Minería de Procesos explicadas anteriormente dentro del circuito postal con el objetivo de proporcionar una nueva herramienta que permita evaluar posibles desvíos operativos en un intervalo de tiempo cercano a la operatoria real para poder corregirlos rápidamente mitigando así el impacto de los mismos en el servicio al cliente.

Luego del análisis realizado en los capítulos anteriores y en base a los resultados obtenidos en los trabajos previamente publicados [20, 21], se decide tomar solo dos de los tres tipos de incidencias analizadas. Anteriormente se consideraron como posibles desvíos las trazas con un ajuste menor a 0,5 con respecto al modelo, las trazas que tengan más de 6 movimientos y las trazas que demoraron más de 10 días. En la aplicación desarrollada sólo se han considerado aquellas trazas con un ajuste menor a 0,5 y las que tardaron más de 10 días dado que las trazas que tenían más de 6 movimientos representaban un porcentaje muy bajo con respecto al total y se encontraron algunas trazas con estas características en los otros conjuntos exportados.

5.2. METODOLOGÍA APLICADA

La metodología aplicada en el presente trabajo es cuantitativa donde la investigación se basa en mediciones objetivas y datos cuantificables, con un diseño experimental. Dado que se están evaluando mejoras en la distribución postal, se elige el tiempo medio de entrega y tasa de errores en la distribución para determinar los resultados y tomar decisiones basadas en datos, ya que impactan directamente en el grado de satisfacción del cliente.

El alcance de este trabajo se basó en el conocimiento del negocio postal del autor y los desafíos para aplicar la Minería de Procesos dentro del marco de la ingeniería de software. Por lo tanto se aplicaron en primera instancia las técnicas de minería de procesos con las herramientas estándar sobre un conjunto de datos reducido para verificar su viabilidad sobre el negocio postal. Posteriormente se aplicaron las técnicas sobre un grupo de datos mucho mayor obteniendo resultados similares, como está descrito en los capítulos anteriores.

Como solución se desarrolló un prototipo de una aplicación de gestión de incidencias productivas que identifica de forma automática posibles desvíos utilizando técnicas de minería de procesos y las disponibiliza para la verificación de un operador. El análisis se realizó sobre envíos que se encuentran en proceso posibilitando la rápida resolución de las incidencias detectadas.

Para validar la solución propuesta se contrastaron los resultados obtenidos de forma manual con las herramientas de minería de procesos estándar contra los obtenidos de forma automática con el prototipo. Adicionalmente se ejecutaron casos de prueba automatizados y manuales para validar las funcionalidades desarrolladas.

En la fase final del presente trabajo se elaboraron las conclusiones y se determinó la validez y fiabilidad de la solución.

Según lo expresado anteriormente para el desarrollo de este trabajo se realizaron una serie de pasos que permitieron analizar la problemática y generar una solución al caso de estudio:

1. Relevamiento de la Minería de Procesos: Se investigó la minería de procesos, sus aplicaciones, ventajas y desventajas con el objetivo de evaluar su utilización en el negocio postal. Realizando dos capacitaciones sobre el tema y varias prácticas con datos de diferentes dominios de negocio.
2. Revisión documental: Se realizó una investigación documental consultando libros, artículos, papers y sitios web reconocidos por la comunidad científica acerca de minería de procesos y su aplicación con el objetivo de identificar la situación actual y antecedentes.
3. Aplicación de Minería de Procesos al negocio postal: Como se expresó anteriormente se realizaron dos papers [20, 21] aplicando las técnicas de minería de procesos en el negocio postal, posteriormente se aplicaron estas técnicas sobre un gran volumen de datos obteniendo resultados similares.
4. Desarrollo del caso de estudio: a partir del contexto obtenido se realizó la propuesta y el desarrollo de una solución tecnológica que permite aplicar las técnicas descriptas anteriormente de forma automatizada.
5. Validación del modelo: Se contrastaron los resultados obtenidos entre la operación manual y la herramienta propuesta.
6. Presentación de resultados: Se presentaron las conclusiones y los resultados obtenidos y algunas posibles futuras líneas de investigación relacionadas.

5.3. CICLO DE VIDA UTILIZADO

El ciclo de vida clásico o cascada incluye al menos las siguientes etapas: recopilación de los requerimientos, análisis, diseño, desarrollo, testeo e implementación. En esta metodología, no se puede comenzar una fase hasta que no se haya completado la actual y luego, no se puede volver a ella, lo que significa que cualquier cambio en los requerimientos tiene un impacto muy grande y debe abordarse en una fase posterior a la finalización del desarrollo. Una de las principales ventajas de esta metodología es que es fácil de entender y aplicar, lo que la hace adecuada para proyectos bien

definidos y estables. Sin embargo, su rigidez y falta de flexibilidad pueden llevar a problemas si los requisitos cambian durante el proceso de desarrollo, lo que puede resultar en retrasos y costos adicionales. Además el cliente no puede tener contacto con el producto hasta etapas avanzadas del desarrollo lo que puede ocasionar el descubrimiento de algún error de negocio en etapas muy avanzadas.

En contraste las metodologías ágiles poseen un enfoque iterativo e incremental para la gestión y desarrollo de software lo que permite adaptarse rápidamente al cambio y entregar un producto de mayor calidad. Según se expresa en [26], Scrum ha surgido para realizar un proceso iterativo, incremental en un marco refinado, bien estructurado y sencillo para el desarrollo de productos complejos.

Entre sus características están:

- Mayor flexibilidad y capacidad de respuesta al cambio: al enfocarse en pequeñas iteraciones, se adapta fácilmente a los cambios en los requisitos y necesidades del cliente.
- Mayor eficiencia en el desarrollo: Al enfocarse en entregas frecuentes, la metodología ágil permite identificar y corregir problemas de manera temprana, lo que aumenta la eficiencia del proceso de desarrollo.
- Mejora de la calidad del software: como la metodología ágil se enfoca en la entrega de software funcional en cada iteración, tiene una mayor eficacia en la identificación y corrección de problemas, reduciendo el tiempo y los costos asociados con la corrección de errores.

Si se verifica que la implementación del prototipo contribuye a la mejora del proceso y la calidad del servicio se propondrá el desarrollo de una solución definitiva.

El prototipo de la aplicación se desarrolló utilizando metodologías ágiles ya que presentan ventajas frente al ciclo de vida clásico; están diseñadas para ser flexibles, adaptarse a las necesidades de cada equipo o proyecto y son las más comunes en la actualidad. Implementar metodologías ágiles se trata de adaptarse al cambio, no de planificar todo por adelantado, los requerimientos cambian constantemente por lo tanto se espera y acepta el cambio.[22].

Se siguieron los lineamientos de Scrum¹ utilizando un Backlog para registrar el desarrollo junto con un tablero Kanban² para documentar los requisitos y tareas que se llevaron a cabo durante este prototipo y reflejar el seguimiento de su avance realizado.

5.4. ARQUITECTURA DE LA SOLUCIÓN

Para el desarrollo del prototipo y posteriormente la aplicación definitiva se evaluaron diferentes alternativas de arquitectura de software pensando en el mantenimiento posterior de la aplicación debido a que como se expresa en [18] una buena arquitectura de software hace el sistema fácil de entender, fácil de desarrollar, fácil de mantener y fácil de desplegar.

¹<https://www.scrum.org/>

²<https://kanban.university/>

Se decidió utilizar una arquitectura orientada a microservicios, la cual utiliza una estrategia de diseño que busca dividir una aplicación en pequeños componentes independientes y autónomos que se comunican entre sí mediante APIs. Esta arquitectura permite una mayor flexibilidad, escalabilidad y facilidad de mantenimiento, ya que cada microservicio se puede desarrollar, probar y desplegar de forma independiente. Los componentes se estructuraron en contenedores independientes usando Docker³ para una rápida integración con la nube y futura orquestación con Kubernetes⁴.

Se desarrollaron de manera independiente los siguientes componentes:

1. Proceso de exportación de archivos: proceso automático que realiza una exportación de la base de datos Oracle productiva y toma las trazas que hayan ingresado desde la última ejecución. Este proceso debe ser programado para ejecutarse automáticamente en un intervalo de tiempo especificado, por defecto una vez por semana. Como resultado dejará las trazas en un archivo csv.
2. Módulo de Minería de Procesos: aplicación que lee el archivo exportado en el paso anterior y aplica las técnicas de Minería de Procesos en la distribución postal como se describieron previamente en este trabajo y luego del análisis, exporta los resultados a una base de datos PostgreSQL.
3. API de gestión: es una API que permite gestionar las incidencias encontradas por el módulo de Minería de Procesos.
4. Web de Administración: Se desarrolló una web que interactúa con la API de gestión de incidencias y permite registrar reclamos y cerrar las incidencias de una forma ágil.
5. Base de Datos: Se utilizó una base de datos relacional para almacenar las incidencias encontradas y registrar su resolución.
6. Observabilidad: Se utilizó Prometheus para verificar el funcionamiento de los servicios y Grafana para mostrar estadísticas.

La arquitectura de la solución propuesta se muestra en la figura 5.1

³<https://www.docker.com/>

⁴<https://kubernetes.io/es/>

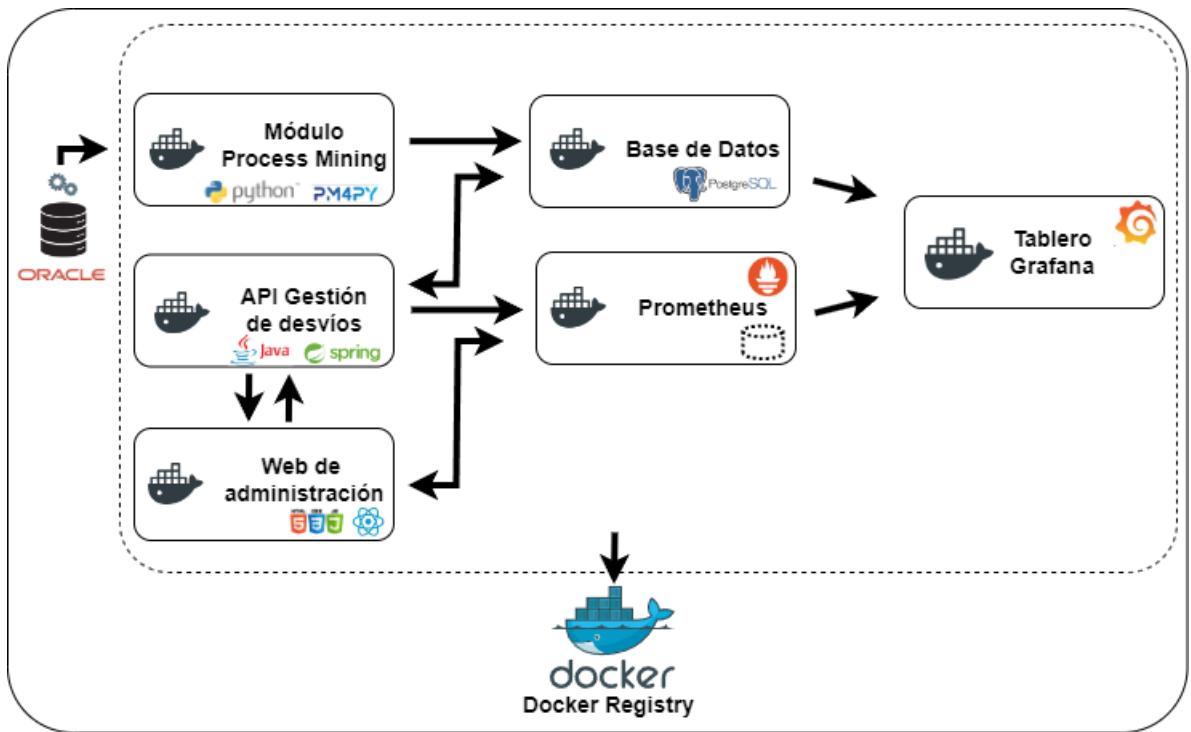


Figura 5.1: Arquitectura del prototipo desarrollado

[19]

Para el desarrollo de este prototipo se siguieron las buenas prácticas de desarrollo de software actuales entre las que se encuentran los casos de prueba automatizados, el uso de estrategias de desarrollo colaborativo y estrategias enfocadas al desarrollo incremental como GitFlow⁵, el código limpio (clean code)[17] generando un código fácil de entender y mantener, la refactorización constante[5] que permite mejorar continuamente el diseño y la estructura del código y el uso de herramientas de evaluación de código como SonarQube⁶.

La implementación de estas prácticas permite integrar la aplicación rápidamente en un ambiente de integración continua y despliegue continuo (CI/CD) donde se automatiza la construcción, pruebas y despliegue de los microservicios de manera automática mediante la utilización de pipelines generados con herramientas de DevOps como Jenkins, GitLab CI/CD o GitHub Actions.

La API de gestión de incidencias se desarrolló con una arquitectura de tres capas como se estila actualmente en este tipo de aplicaciones desarrolladas con Spring o Spring Boot implementando las capas de controlador, servicio y repositorio; estos tres componentes juegan un papel crucial en la construcción de una API robusta y escalable.

El controlador es el punto de entrada para la API. Recibe las solicitudes del cliente y las asigna a los métodos apropiados en la capa de servicio. El controlador es responsable de manejar todas las solicitudes y respuestas HTTP y delega la lógica comercial a la capa de servicio.

El Servicio es una capa que contiene toda la lógica de negocio de la aplicación. Maneja el acceso, procesamiento y transformación de datos. Esta capa es responsable de ejecutar todas las operaciones

⁵<https://aulasoftwarelibre.github.io/taller-de-git/gitflow/>

⁶<https://docs.sonarqube.org/latest/>

necesarias requeridas por la API e interactuar con la capa de datos.

El repositorio es una capa que se ocupa del acceso a los datos. Es responsable de persistir las entidades en una base de datos o cualquier otro medio de almacenamiento.

Adicionalmente se generó la documentación contenida en la aplicación junto con un mecanismo que permite verificar su estado, es decir, si el servicio está activo o tiene algún inconveniente (health check).

5.5. ANÁLISIS DE LA NECESIDAD

5.5.1. Requerimientos funcionales

Para el desarrollo del prototipo para aplicar Minería de Procesos a la distribución postal se recopilaron los siguientes requisitos:

- Generación de Prueba de Concepto: Se desarrolló una herramienta mínima para verificar la viabilidad de utilizar las técnicas de Minería de Procesos desde el código, la misma incluye:
 - Importación de un archivo de trazas en formato .xes.
 - Importación de un proceso modelado como una Red de Petri.
 - Realizar la verificación de conformidad del archivo de trazas contra la Red de Petri.
 - Filtrar las trazas que no cumplan con los criterios establecidos (ajuste o duración)
 - Exportar las trazas resultantes en un archivo de formato .csv.
- Verificación de Prueba de Concepto: Se contrastaron los resultados obtenidos por la prueba de concepto contra los que brinda una herramienta de Minería de Procesos como por ejemplo ProM tools⁷.
- Generación de Proceso de exportación de datos: se generó un proceso batch que se puede ejecutar de manera automática en los servidores de la empresa y realiza una extracción de la base de datos productiva con todos los envíos que están pendientes de análisis. En la aplicación final este proceso estará programado para ejecutarse en el servidor una vez a la semana.
- Componente de Minería de Procesos: Este componente se encarga de procesar los archivos que se exporten semanalmente y realiza el análisis completo de las trazas a través de los siguientes pasos:
 - Importación del archivo resultante del proceso de exportación.
 - Conversión de los datos a formato .xes.
 - Filtrado de trazas inconsistentes.
 - Importación del proceso a verificar.

⁷<https://promtools.org/>

- Verificación de conformidad del archivo log.
 - Filtrado de trazas con ajuste menor a 0,5
 - Filtrado de trazas que tengan más de diez días de duración.
 - Ingreso de datos resultantes del análisis a la base de datos de gestión de incidencias.
 - Envío de alerta por mail con resumen de posibles desvíos encontrados.
- API de Gestión de incidencias: Permite listar las incidencias encontradas y brinda información adicional de las mismas para tomar acciones predictivas o correctivas con el desvío que representan. Contiene las siguientes funcionalidades:
 - Listado de posibles desvíos pendientes de tratamiento por ajuste bajo.
 - Listado de posibles desvíos pendientes de tratamiento por excesivo tiempo de ejecución.
 - Obtención de datos adicionales del desvío como responsable, sucursal y datos adicionales.
 - Registro de resultado de tratamiento por ejemplo resuelto, informado, o no es desvío.
 - Web de administración: Permite visualizar las incidencias encontradas y gestionarlas interactuando con la API de gestión de incidencias. Contiene las siguientes funcionalidades:
 - Resumen de incidencias encontradas.
 - Visualización de posibles desvíos pendientes de tratamiento (por ajuste bajo).
 - Visualización de posibles desvíos pendientes de tratamiento (por excesivo tiempo de ejecución).
 - Visualización de datos adicionales del desvío.
 - Registro de resultado de tratamiento, por ejemplo resuelto, informado o no es un desvío.
 - Base de Datos: Se generó una base de datos relacional para almacenar la información de obtenida del análisis realizado.

5.5.2. Requerimientos no funcionales

- Observabilidad de la aplicación: Se implementaron herramientas que permiten visualizar el estado de los servicios en tableros dinámicos, así como monitorear su uso y su estado a través de un componente que permite realizar el health check de la aplicación.
- Rápida visualización de incidencias: Se implementaron de tableros que permiten visualizar rápidamente las incidencias encontradas y su estado sin necesidad de ingresar en la web de gestión de incidencias.
- Contenerización: Los servicios se disponibilizaron en contenedores para un despliegue productivo rápido, de fácil mantenimiento y escalamiento vertical y horizontal adecuado.

Las funcionalidades se mapearon en un tablero Kanban siguiendo los lineamientos de Scrum organizando las principales tareas en épicas que se componen de tareas más pequeñas con el objetivo de atomizar al máximo las tareas para organizar y distribuir mejor la carga de trabajo. Se generó un ticket por cada tarea, el tablero se muestra en la figura 5.2

Proyectos / UNLP-Tesis

Tablero UNLP-Tesis

POR HACER 6 INCIDENCIAS	EN CURSO 3 INCIDENCIAS	REVISIÓN 2 INCIDENCIAS	LISTO 2 INCIDENCIAS
Importar resultados VISUALIZACIONES <input checked="" type="checkbox"/> UT-11	Proceso de extracción de datos de Oracle GENERAR APLICACIÓN PROCESS ... <input checked="" type="checkbox"/> UT-12	Verificar funcionamiento POC POC PROCESS MINING POSTAL <input checked="" type="checkbox"/> UT-2	Generar POC Process Mining POC PROCESS MINING POSTAL <input checked="" type="checkbox"/> UT-1
Generar Visualizaciones Grafana VISUALIZACIONES <input checked="" type="checkbox"/> UT-6	Generar Aplicación Process Mining GENERAR APLICACIÓN PROCESS ... <input checked="" type="checkbox"/> UT-4	Generar base de API de gestión de desvíos, login, persistencia, monitoreo API GESTIÓN DE DESVIOS <input checked="" type="checkbox"/> UT-5	Generar BD PostgreSQL API GESTIÓN DE DESVIOS <input checked="" type="checkbox"/> UT-9
Generar endpoint para visualización de incidencias con excesivo tiempo de ejecución sin procesar API GESTIÓN DE DESVIOS <input checked="" type="checkbox"/> UT-14	Generar endpoint para visualización de incidencias con ajuste bajo sin procesar API GESTIÓN DE DESVIOS <input checked="" type="checkbox"/> UT-13		
Generar endpoint de información adicional de incidencia API GESTIÓN DE DESVIOS <input checked="" type="checkbox"/> UT-15			

Figura 5.2: Tablero Kanban con funcionalidades a desarrollar

[19]

En la figura 5.3 se muestra el detalle de la épica *API Gestión de Desvíos* con la lista de tareas necesarias para completar el desarrollo y el estado actual de cada una.

The screenshot shows a software interface for managing a backlog of tasks. At the top, there's a header with a lock icon, a user profile, and various navigation buttons. Below the header, the title "API Gestión de desvíos" is displayed, along with buttons for "Adjuntar", "Añadir una incidencia secundaria", "Vincular incidencia", and "...".

The main area is titled "Incidencias secundarias" and contains a table with the following data:

ID	Descripción	Status
UT-9	Generar BD PostgreSQL	FINALIZADA
UT-5	Generar base de API de gestión de desvíos, login, persistencia, monitoreo	REVISIÓN
UT-13	Generar endpoint para visualización de incidencias con ajuste bajo sin procesar	EN CURSO
UT-14	Generar endpoint para visualización de incidencias con excesivo tiempo de ejecución sin ...	TAREAS POR HACER
UT-15	Generar endpoint de información adicional de incidencia	TAREAS POR HACER
UT-16	Generar endpoint de registro de incidencia procesada	TAREAS POR HACER
UT-17	Generar casos de prueba	TAREAS POR HACER

At the bottom of the main area, there's a button to "Añadir un comentario..." and a note: "Consejo de expertos: pulsa M para comentar".

The right sidebar provides detailed information for the selected task (UT-9):

- Tareas por hacer**: Includes a note: "Haz clic en el >, situado junto a la etiqueta de un campo para anclarlo."
- Acciones**: Includes a note: "Haz clic en el >, situado junto a la etiqueta de un campo para anclarlo."
- Campos anclados**: No items listed.
- Detalles**:
 - Responsable**: Sin asignar
 - Asignarme a mí**
 - Etiquetas**: Ninguno
 - Start date**: Ninguno
 - Fecha de vencimiento**: Ninguno
 - Informador**: Victor M
- Configurar**: Buttons for "Creado hace 3 días" and "Actualizado hace 3 días".

Figura 5.3: Épica API de Gestión de Incidencias

[19]

5.6. DISEÑO DEL PROTOTIPO

5.6.1. Consideraciones generales

El prototipo desarrollado está compuesto por tres partes principales, el componente que analiza los datos de distribución postal y aplica las técnicas de Minería de Procesos, el componente que permite gestionar las incidencias encontradas y registrarlas en la base de datos donde se persiste la información y la web de visualización donde se administran las incidencias encontradas. Adicionalmente se implementaron herramientas actuales de monitoreo y visualización de datos.

El proceso que toma los datos originales de la Base de Datos Oracle es un shell script que ejecuta una consulta a la base de datos y guarda los resultados en un archivo csv.

Para desarrollar el módulo de Minería de Procesos se utilizó Python como lenguaje de programación y la librería PM4PY⁸ para implementar las técnicas de Minería de Procesos seleccionadas.

Para desarrollar la API de gestión de incidencias se utilizó java con Spring boot y JPA. Spring Boot es un framework de desarrollo de aplicaciones Java que simplifica la configuración y el desarrollo, lo que permite crear rápidamente aplicaciones robustas y escalables. Proporciona características como la configuración automática, el inicio rápido, la gestión de dependencias y la integración con otros módulos de Spring. JPA es una especificación de Java que proporciona una capa de abstracción para el acceso y la manipulación de datos en una base de datos. Con JPA, es posible mapear objetos Java a entidades de base de datos y realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera

⁸<https://pm4py.fit.fraunhofer.de/>

sencilla.

Se documentó con Open API⁹ a través de Swagger, una herramienta para diseñar, construir, documentar y consumir servicios web RESTful. Proporciona una interfaz visual interactiva, conocida como Swagger UI, que permite explorar y probar los endpoints de la API de forma fácil y comprensible. Además, Swagger genera automáticamente documentación precisa y detallada de la API, lo que facilita su comprensión y uso por parte de otros desarrolladores.

Se utilizó el componente Actuator¹⁰ de Spring¹¹ para exponer las estadísticas de uso, realizar el monitoreo y la comprobación del estado de la aplicación (health check).

La web de gestión fue desarrollada utilizando o React, HTML5, CSS y Bootstrap. React es una biblioteca JavaScript de código abierto que permite crear interfaces de usuario interactivas y dinámicas. Sus principales ventajas incluyen la reutilización de componentes, rendimiento optimizado, gestión eficiente del estado y una amplia comunidad de desarrollo. HTML5 es la última versión del lenguaje de marcado estándar para la creación de páginas web. Proporciona nuevas etiquetas y atributos que permiten una estructura más clara y semántica del contenido, así como soporte para multimedia y aplicaciones web avanzadas. CSS (Cascading Style Sheets) se utiliza para definir el diseño, el aspecto visual y la presentación de una página web. Bootstrap es un framework CSS popular que proporciona una serie de componentes y estilos predefinidos. Permite crear interfaces de usuario responsivas y con un aspecto profesional de manera rápida y sencilla.

Para la persistencia de los datos se consideró necesario elegir una herramienta de almacenamiento de datos confiable y escalable, por lo que se optó por un motor de base de datos relacional. Para ello, se seleccionó PostgreSQL, una herramienta open source ampliamente reconocida en el mercado por su robustez y estabilidad. La elección de PostgreSQL como herramienta de almacenamiento se debió a su capacidad para manejar grandes volúmenes de datos, su arquitectura orientada a objetos y su compatibilidad con diversos sistemas operativos. Además, permitió utilizar un dblink para la integración de datos provenientes de distintas fuentes en un solo repositorio.

Para realizar el monitoreo de los servicios se utilizó Prometheus ya que permite recopilar las estadísticas de uso y funcionamiento. Adicionalmente se usó Grafana para su visualización. En Grafana también se generaron tableros que permiten observar y diagnosticar rápidamente las incidencias encontradas.

5.6.2. Diagrama de entidad relación

La base de datos de este prototipo contiene las tablas necesarias para almacenar la información de las incidencias encontradas y su gestión. Adicionalmente puede conectarse a la base de datos Oracle productiva a través de un dblink para traer información correspondiente a sucursales y envíos necesaria para la gestión.

La figura 5.4 muestra el Diagrama de Entidad Relación con las tablas necesarias para el prototipo

⁹<https://www.openapis.org/>

¹⁰<https://spring.io/guides/gs/actuator-service/>

¹¹<https://spring.io/>

y las principales tablas que se consultan de las Base de datos Externa. En el caso puntual de este prototipo las tablas externas se replicaron en la base de datos PostgreSQL ya que no se tiene acceso a los servidores productivos.

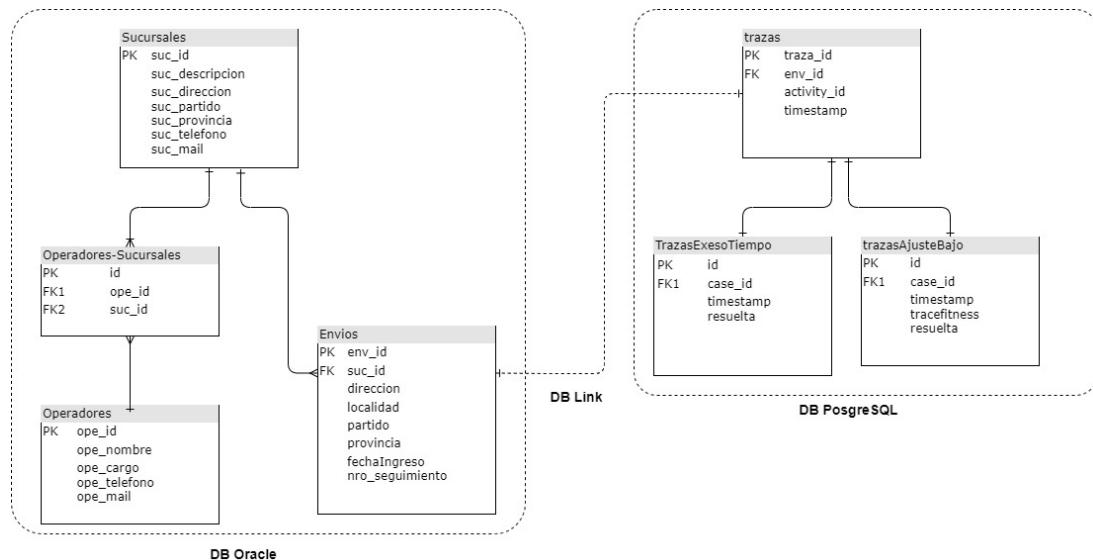


Figura 5.4: DER del Prototipo de Minería de Procesos aplicado a la distribución postal

[19]

5.7. DESARROLLO

5.7.1. Prueba de Concepto

Como se mencionó anteriormente, la prueba de concepto se desarrolló con el lenguaje de programación Python, utilizando la librería PM4PY para aplicar las técnicas de Minería de Procesos y verificar viabilidad del desarrollo. Se eligió Anaconda Navigator¹² una interfaz gráfica que permite administrar extensiones y lanzar paquetes de Anaconda y entornos de desarrollo. Puntualmente se utilizó el IDE Spider¹³ para hacer el desarrollo por ser uno de los más utilizados en Python para hacer desarrollos e investigación científica.

La figura 5.5 ilustra el desarrollo de la prueba de concepto en el IDE Spider.

¹²<https://docs.anaconda.com/free/navigator/>

¹³<https://www.spyder-ide.org/>

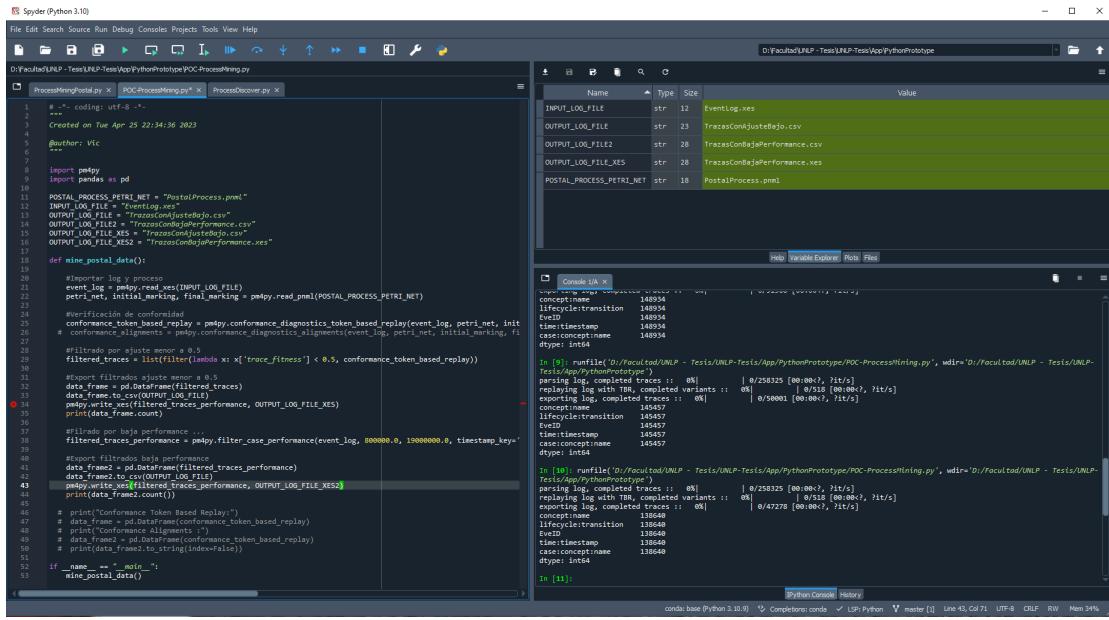


Figura 5.5: Prueba de concepto visualización del entorno de desarrollo Spyder

[19]

El prototipo desarrollado importa un archivo de trazas en formato .XES y el proceso postal descubierto anteriormente en formato de Red de Petri. Luego, reproduce una a una las trazas sobre el modelo del proceso Postal y filtra las trazas que tengan un ajuste menor a 0,5 o más de 10 días de duración, siguiendo los lineamientos establecidos en este trabajo. Finalmente exporta los resultados en archivos .XES y .csv para su posterior utilización.

5.7.2. Verificación de la Prueba de Concepto

Para comprobar el funcionamiento de la prueba de concepto se procesaron nuevamente los archivos de datos que se utilizaron en este trabajo y luego se realizó una comprobación visual de cada uno de los archivos de datos resultantes en ProM Tools, la misma herramienta que se utilizó originalmente. Se comparó la cantidad de trazas con ajuste menor a 0,5 y con más de 10 días de duración, encontrando resultados similares.

Con estos resultados se dio por válida la implementación de Minería de Procesos en la prueba de concepto desarrollada en Python y se continuó con el desarrollo de la aplicación.

5.7.3. Proceso de exportación de Datos desde Oracle

Para el proceso de extracción de datos de la base de datos Oracle, se decidió utilizar un shell script que estará programado para ser ejecutado periódicamente en el servidor de Base de Datos. El Listado B.1 del Anexo B ejemplifica la manera de programar la ejecución del proceso todos los domingos a las 13:00 hs. La tarea principal de este script es ejecutar una consulta SQL para obtener todos los envíos que hayan sido ingresados desde la última ejecución del script.

Una vez obtenidos los resultados de la consulta, el script los exporta a un archivo .csv. Este archivo se genera en una ubicación específica del servidor, y contendrá los datos de los envíos ordenados por

traza. Este ordenamiento se ha realizado con el fin de optimizar el procesamiento posterior de los datos.

Luego de la correcta generación del archivo .csv, se procede a marcar los envíos como *exportados*. Esta tarea se realiza para asegurar que no se vuelvan a exportar los mismos envíos en la próxima ejecución del script. El script se ejecutará una vez a la semana, siguiendo un cronograma específico en el servidor Unix. Esto permitirá una actualización regular de los datos y un flujo de información constante y confiable.

Siguiendo los pasos hasta aquí descriptos, se logró automatizar el proceso de extracción de datos de la base de datos Oracle, reduciendo el tiempo y la complejidad de esta tarea. Además, gracias a la marcación de envíos exportados, se evitó la duplicidad de datos y se aseguró la integridad y calidad de la información.

Una sugerencia será programar la ejecución de este script los domingos por la mañana cuando se registra muy poco movimiento en la base de datos productiva.

El código fuente del script de extracción de datos de Oracle se muestra en 5.1

Listado 5.1: Proceso de extracción de datos de Oracle

```
1  #!/bin/bash
2
3  # Configuración de la conexión a la base de datos Oracle
4  ORACLE_USER="userName"
5  ORACLE_PASSWORD="userPassword"
6  ORACLE_HOST="192.168.0.54"
7  ORACLE_PORT="5001"
8  ORACLE_SID="bdProd"
9
10 # Consulta SQL a ejecutar
11 SQL_QUERY="SELECT env_id AS case_id , eve_descrip AS activity , ←
12   ↪ eve_fecha AS timestamp , eve_id AS activityid
13   FROM envios
14 WHERE informado IS NULL
15 ORDER BY env_id"
16
17 # Archivo de salida csv
18 OUTPUT_FILE="TrazasEntrada.csv"
19
20 # Conectarse a la base de datos Oracle y ejecutar la consulta
21 sqlplus -s ${ORACLE_USER}/${ORACLE_PASSWORD}@${ORACLE_HOST}: ←
22   ↪ ${ORACLE_PORT}/${ORACLE_SID} << EOF
23 set pagesize 0
24 set colsep ,
25 set feedback off
26 set echo off
27 set heading off
28 spool ${OUTPUT_FILE}
29 ${SQL_QUERY};
30 spool off
31 exit
```

30 EOF

5.7.4. Base de Datos

La base de datos utilizada en este prototipo cuenta con las tablas necesarias para almacenar la información de las incidencias encontradas y su respectiva gestión. Para el ambiente productivo debe establecerse una conexión entre la base de datos PostgreSQL y la base de datos Oracle mediante el uso de dblink, con el objetivo de acceder a la información correspondiente a sucursales y envíos necesarios para la gestión. En el ámbito de este prototipo se replicaron los datos provenientes de Oracle dado que no se cuenta con acceso a los servidores productivos.

Las tablas principales, generadas según lo establecido en el diseño de la Base de Datos, son:

- trazas: Contiene el listado de trazas exportado desde la base de datos Oracle. Guarda un registro por cada movimiento de la traza para poder reconstruir su historia.
- trazasajustebajo: Contiene las trazas que tengan un ajuste por debajo del umbral establecido como posible desvío, el cual actualmente fue fijado en 0,5.
- trazasexcesotiempo: Contiene las trazas que tengan una duración mayor a la establecida como umbral de posibles desvío. Dicha duración, para este caso fue fijada en 10 días corridos.

La figura 5.6 muestra cómo se almacenan las trazas que tienen un ajuste bajo para su posterior tratamiento. El componente que aplica las técnicas de Minería de Procesos sobre los archivos de entrada es quien inserta los datos en la BD PostgreSQL.

Grid	case_id	timestamp	@@index	@@case_index	trace_fitness	resuelta
1	3726984	2019-12-06 16:02:48.000 -0300	44	14	0.4	[]
2	3727024	2019-12-09 17:09:13.000 -0300	129	45	0.4722222222	[]
3	3727337	2019-12-12 06:41:20.000 -0300	671	232	0.4285714286	[]
4	3727449	2019-12-12 06:29:12.000 -0300	936	339	0.3666666667	[]
5	3727479	2019-12-12 06:27:36.000 -0300	1,009	368	0.4642857143	[]
6	3727486	2019-12-12 06:28:09.000 -0300	1,028	375	0.3666666667	[]
7	3727518	2019-12-12 07:12:14.000 -0300	1,100	403	0.4	[]
8	3727519	2019-12-12 07:11:39.000 -0300	1,102	404	0.4	[]
9	3727720	2019-12-12 07:17:17.000 -0300	1,570	592	0.4285714286	[]
10	3727815	2019-12-12 07:16:18.000 -0300	1,773	672	0.4285714286	[]
11	3728024	2019-12-12 07:24:59.000 -0300	2,253	868	0.375	[]
12	3728374	2019-12-12 06:17:19.000 -0300	3,071	1,194	0.4285714286	[]
13	3728473	2019-12-09 16:55:14.000 -0300	3,241	1,262	0.4285714286	[]
14	3728569	2019-12-09 16:50:01.000 -0300	3,457	1,358	0.4285714286	[]
15	3728656	2019-12-09 16:53:08.000 -0300	3,658	1,445	0.4017857143	[]
16	3728659	2019-12-09 16:54:04.000 -0300	3,665	1,448	0.4017857143	[]
17	3728724	2019-12-09 16:48:55.000 -0300	3,832	1,513	0.4017857143	[]
18	3728866	2019-12-09 17:19:25.000 -0300	4,195	1,655	0.4	[]
19	3728884	2019-12-09 17:19:48.000 -0300	4,237	1,673	0.4017857143	[]
20	3728910	2019-12-09 17:20:33.000 -0300	4,301	1,699	0.4285714286	[]
21	3728985	2019-12-09 17:16:50.000 -0300	4,526	1,774	0.4285714286	[]
22	3729079	2019-12-09 15:56:33.000 -0300	4,783	1,868	0.4285714286	[]
23	3729372	2019-12-09 15:58:33.000 -0300	5,526	2,161	0.4285714286	[]
24	3729393	2019-12-09 15:58:49.000 -0300	5,576	2,182	0.4285714286	[]
25	3729728	2019-12-09 16:59:40.000 -0300	6,398	2,511	0.4285714286	[]

Figura 5.6: Tabla donde se almacenan las trazas encontradas con ajuste bajo

[19]

5.7.5. Componente de Minería de Procesos

Luego de la verificación de la prueba de concepto se desarrolló el componente que aplica las técnicas de Minería de Procesos sobre los archivos resultantes del proceso de exportación de datos. Este componente está diseñado para procesar y analizar trazas de los envíos postales realizando una serie de tareas predefinidas en función de los parámetros de entrada establecidos, las cuales se detallan a continuación:

En primer lugar, se realiza una verificación de la existencia del archivo csv que contiene las trazas a procesar. Si este archivo existe, se importa para poder procesar los datos contenidos en él. A continuación, se importa el modelo de Red de Petri en el que se define el proceso postal y se realiza la verificación de conformidad de las trazas contra este modelo.

Una vez verificada la conformidad, se remueven las columnas adicionales que se hubieran generado durante el análisis y se filtran las trazas en función de los resultados obtenidos. Luego se separan aquellas que tengan un ajuste menor a 0,5 y las que tardan más de diez días en finalizar. Finalmente, se insertan los datos resultantes en la base de datos PostgreSQL en las tablas diseñadas para tal fin: trazas, trazasajustebajo y trazasexcesotiempo, agregando los nuevos valores a los existentes en cada tabla.

Después de realizada la inserción en la base de datos, se envía una notificación por correo electrónico a la dirección de correo electrónico establecida en uno de los parámetros de entrada.

Finalmente, se renombra el archivo csv original concatenando la fecha actual al nombre y se mueve a la carpeta *procesados* para guardarla como copia de seguridad de las trazas procesadas.

De esta manera este componente es idempotente y altamente configurable por lo tanto puede ser adaptado a distintas necesidades específicas. El pseudo código del componente de Minería de Procesos se muestra en el algoritmo 5.1

Algoritmo 5.1: Módulo de Minería de Procesos

```

Datos de entrada: Archivo de trazas
Resultado : Incidencias encontradas exportadas a Base de Datos
1 inicialización;
2 Leer archivo de entrada;
3 if El archivo contiene registros then
4   Importar proceso en red de Petri;
5   Ejecutar verificación de conformidad;
6   Filtrar trazas con ajuste menor a 0,5;
7   Filtrar trazas que tardaron más de 10 días;
8   Insertar Trazas filtradas en Base de datos;
9   Enviar mail indicando posibles desvíos;
10 else
11   Enviar mail Indicando que no se encontraron desvíos;
12 end
13 Mover el archivo a Procesados;
```

La ejecución del componente de Minería de Procesos está programada para ejecutarse luego de la extracción de datos de Oracle. Una opción es configurar la programación para ejecutarse todos los domingos a las 16:00 hs, de esta manera cuando el operador ingrese el lunes a trabajar tendrá el aviso de incidencias encontradas en su correo electrónico.

5.7.6. API de administración de desvíos

La API de administración de desvíos se desarrolló utilizando una arquitectura de tres capas que separa claramente las responsabilidades y permite tener una aplicación modular y fácil de mantener.

La figura 5.7 muestra fragmentos de código de los componentes controlador, servicio y repositorio de trazas.

The screenshot shows three code snippets from an IntelliJ IDEA project:

- TrazasController.java:**

```

@.RestController
@RequestMapping(value="/trazas")
@CrossOrigin
public class TrazasController {
    @Autowired
    TrazasService trazasService;
    @PostMapping(value="/", produces= "application/json")
    @Operation(summary="Dar de alta una nueva traza", description="Permite agregar un traza manualmente", tags = {"Trazas"})
    public @ResponseBody void addTrazas(Traza traza){
        trazasService.addTrazas(traza);
    }
}

```
- TrazasService.java:**

```

@Service
public class TrazasService {
    @Autowired
    TrazasRepository trazasRepository;
    public Traza addTrazas(Traza traza) { return trazasRepository.save(traza); }
    public List<Traza> getTrazasById(Long case_id) {
        return trazasRepository.getAllTrazasByCase_id(String.valueOf(case_id));
    }
}

```
- TrazasRepository.java:**

```

public interface TrazasRepository extends JpaRepository<Traza, Long> {
    @Query(nativeQuery=true, value = "SELECT id, case_id, activity, timestamp, activityid FROM trazas WHERE case_id= :case_id")
    List<Traza> getAllTrazasByCase_id(String case_id);

    Page<Traza> findAll(Pageable pageable);

    @Query(nativeQuery=true, value = "SELECT id, activity, timestamp, activityid, FROM trazas WHERE timestamp= :date")
    List<Traza> getAllTrazasByDate(Integer date);
}

```

Figura 5.7: Fragmento de capas controlador, servicio y repositorio en Spring

[19]

El lenguaje de programación utilizado fue Java con el IDE IntelliJ IDEA¹⁴ utilizando el framework Spring¹⁵. La persistencia de datos se realizó utilizando JPA y repositorios para una gestión eficiente de la información. Para documentar las funcionalidades, se utilizó la especificación Open API¹⁶ y Swagger, lo que facilita la interacción con la API posibilitando pruebas directamente a través de un navegador web y permitiendo que la documentación esté insertada dentro del código.

El componente Actuator de Spring¹⁷ se utilizó para monitorear y garantizar la observabilidad de la API, lo que permite una rápida identificación de problemas y una solución efectiva de errores en tiempo real. El desarrollo de la API se realizó siguiendo las buenas prácticas actuales.

¹⁴<https://www.jetbrains.com/idea/>

¹⁵<https://spring.io/>

¹⁶<https://www.openapis.org/>

¹⁷<https://spring.io/guides/gs/actuator-service/>

La figura 5.8 muestra una captura de pantalla del IDE IntelliJ IDEA donde se desarrolló la API de gestión de incidencias.

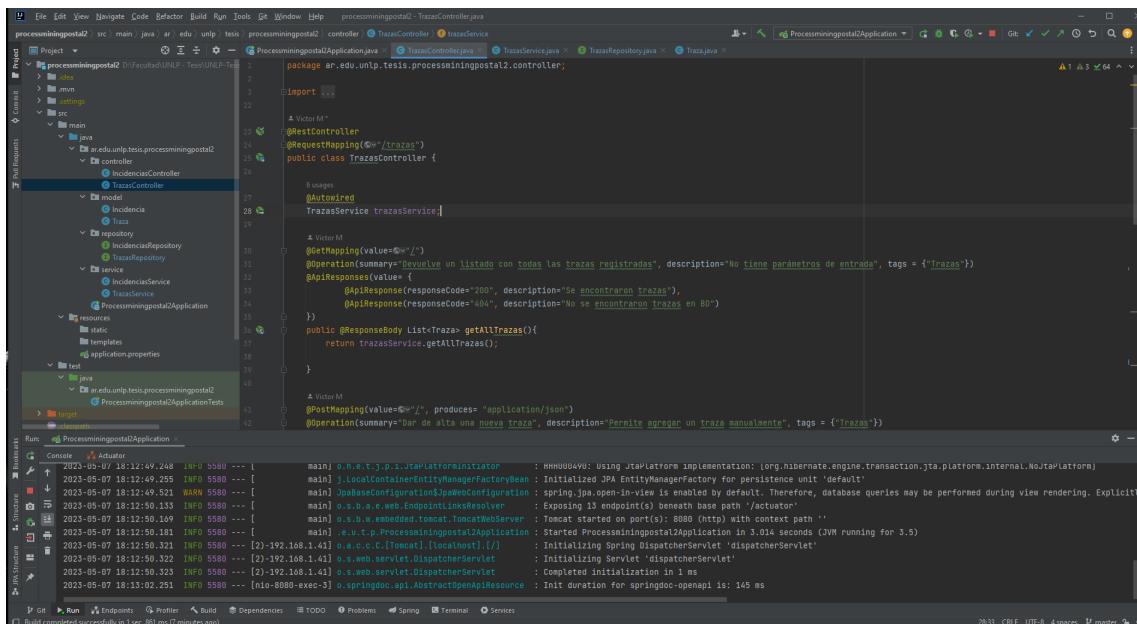


Figura 5.8: Desarrollo de la API de gestión de incidencias en IntelliJ IDEA

[19]

De esta manera se logra que la API de administración de desvíos sea una solución robusta, escalable y bien estructurada, que cumple con los estándares actuales de desarrollo de software y ofrece una experiencia de usuario fluida y segura.

La figura 5.9 muestra como se implementó la documentación de la API utilizando SWagger y Open API.

The screenshot shows the Swagger UI interface for the UNLP Thesis API. At the top, it displays the title "UNLP Tesis - Minería de Procesos Aplicada a la Distribución Postal" and the version "V1 OAS3". Below this, there are links for "Universidad Nacinal de La Plata - Facultad de Informática" and "Esta API es parte del trabajo final para la Maestría en Ingeniería de Software - Victor Martinez". There are also links for "Terms of service" and "Apache 2.0". A dropdown menu under "Servers" shows "http://localhost:8080 - Generated server url".

Trazas

- GET /trazas/** Devuelve un listado con todas las trazas registradas
- PUT /trazas/** Modificar una traza una traza de la base
- POST /trazas/** Dar de alta una nueva traza
- GET /trazas/{idTraza}** Obtener una indicencia de la base
- GET /trazas/year/{year}** Obtener un listado de trazas según el año de estreno
- DELETE /trazas/{id}** Borrar una traza de la base

Incidencias

- GET /incidencias/** Devuelve un listado con todas las incidencias registradas
- PUT /incidencias/** Modificar una incidencia una incidencia de la base
- POST /incidencias/** Dar de alta una nueva incidencia

Figura 5.9: Documentación de la API con Swagger y Open API

[19]

5.7.7. Web de Administración

Para la gestión de incidencias se desarrolló un Front End Web en React¹⁸ que interactúa con la API de gestión de incidencias.

Se eligió React debido a facilidad para el desarrollo y velocidad de renderizado, utilizando HTML 5, CSS y Bootstrap¹⁹ para generar una aplicación flexible y crear una interfaz de usuario interactiva, personalizable y compatible con dispositivos móviles, lo que mejora la experiencia del usuario y facilita el desarrollo del software, obteniendo las siguientes ventajas:

1. Desarrollo eficiente: React permite dividir la interfaz de usuario en componentes reutilizables, lo que facilita el desarrollo y el mantenimiento del código.
2. Interfaz de usuario interactiva: React actualiza y renderiza de forma eficiente solo los componentes que han cambiado, lo que proporciona una experiencia de usuario fluida y rápida.
3. Diseño y apariencia personalizables: Con CSS y Bootstrap, se puede personalizar fácilmente el diseño y la apariencia de la aplicación para que se ajuste a las necesidades y estilo de marca requeridos.
4. Responsividad y compatibilidad con dispositivos móviles: Bootstrap ofrece una estructura de

¹⁸<https://es.react.dev/>

¹⁹<https://getbootstrap.com/>

rejilla adaptable que permite que la aplicación se adapte a diferentes tamaños de pantalla y dispositivos.

5. Compatibilidad con estándares modernos: HTML5 proporciona características avanzadas y compatibilidad con multimedia, formularios interactivos y otras funcionalidades modernas de la web.

Este componente se desarrolló con el IDE Vistual Studio Code²⁰. La figura 5.10 muestra una captura de pantalla del mismo.

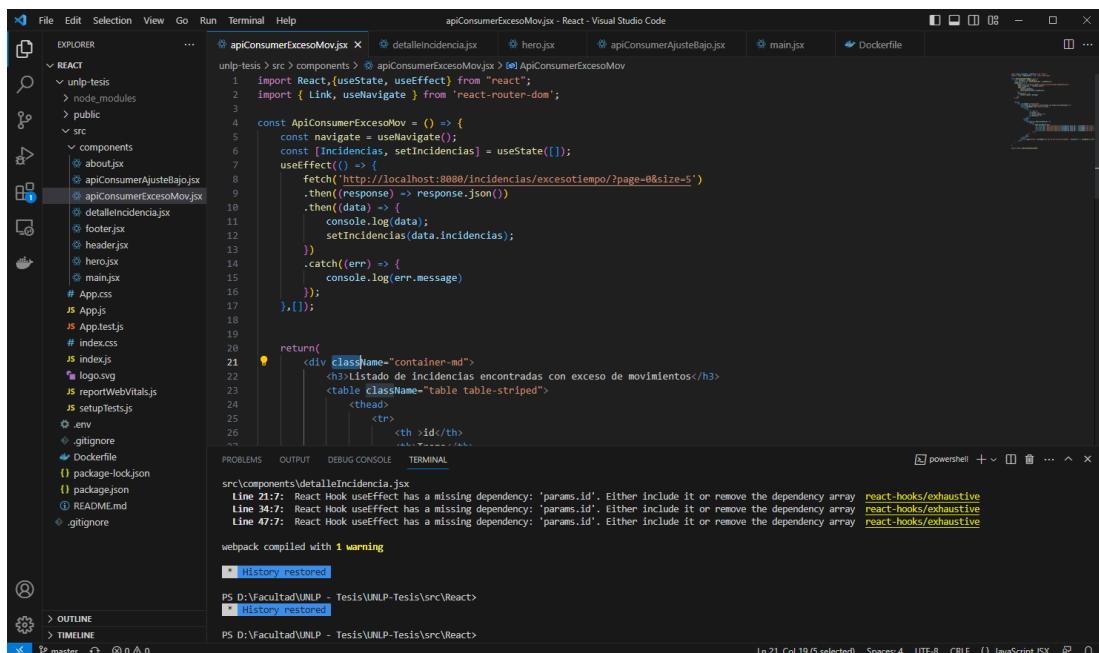


Figura 5.10: Desarrollo de la web de administración en Visual Studio Code

[19]

La figura 5.11 muestra la pantalla de inicio de la web para la gestión de las incidencias, en la que se puede ver el detalle de cada caso y el registro un nuevo reclamo.

²⁰<https://code.visualstudio.com/>



Figura 5.11: Web de gestión de incidencias

[19]

La figura 5.12 muestra el listado de incidencias encontradas con ajuste bajo. En este punto, el prototipo desarrollado ofrece la posibilidad de seleccionar una incidencia del listado para ver sus detalles.

INCIDENCIAS CON AJUSTE BAJO

Id	Traza	Ajuste	Fecha Ingreso	Resuelta
1	3726984	0.4	2019-12-06	No
2	3727024	0.4722222	2019-12-09	No
3	3727337	0.42857143	2019-12-12	No
4	3727449	0.36666667	2019-12-12	No
5	3727479	0.4642857	2019-12-12	No

Anterior 1 2 3 4 5 Siguiente

Volver

UNLP - Maestría en Ingeniería de Software - Victor Martinez

Figura 5.12: Listado de incidencias

[19]

La figura 5.13 muestra el detalle de una incidencia donde se observan todos los movimientos que tuvo, los principales datos del envío como dirección de entrega, partido o localidad, provincia y los datos de la sucursal a la que se encuentra asignado el envío para poder comunicarse. Adicionalmente se visualizan los reclamos generados de ese envío (si hubiera) y una opción para registrar un nuevo reclamo de ser necesario. También se puede cerrar esta incidencia con lo cual dejaría de mostrarse en el aplicativo.

The screenshot displays a web-based incident management system. At the top, the URL is shown as `localhost:8001/detalleincidencia/3726970`. The header includes the institution name "UNLP - Tesis" and navigation links for "Inicio", "Ajuste bajo", "Exceso de Movimientos", and "Acerca de...". A search bar with the placeholder "Incidencia" and a green "Buscar" button are also present.

The main content area is titled "Detalle incidencia 3726970". It is divided into four sections:

- Datos del Envío** (Shipping Data):

Dirección de Entrega:	Matheu 148
Localidad:	Banfield
Partido:	Lomas de Zamora
Provincia:	Buenos Aires
Nro de Seguimiento:	CD123456123AR
- Datos de Sucursal** (Branch Data):

Nombre:	Banfield
Dirección:	Maipú 125
Partido:	Lomas de Zamora
Provincia:	Buenos Aires
Teléfono:	4248-2254
Mail:	sucbanfield@sucursalsales.com.ar
- Movimientos** (Movements):

id	Evento	Fecha Evento
1	INGRESADO	2019-12-06
2	1 INTENTO DE ENTREGA	2019-12-12
3	2 INTENTO DE ENTREGA	2019-12-16
4	ESPERANDO RETIRO	2019-12-17
5	DEVOLUCION	2019-12-26
- Reclamos del envío** (Delivery Complaints):

id	Fecha Reclamo
2	2023-05-10
4	2023-05-14

At the bottom of the page, there are three buttons: "Registrar Reclamo" (Register Complaint), "Cerrar Incidencia" (Close Incident), and "Volver" (Return).

Figura 5.13: Web de gestión de incidencias - detalle de incidencia

[19]

5.7.8. Visualizaciones

Se generaron visualizaciones de los servicios y de las incidencia en un tablero de Grafana lo que permite realizar de forma intuitiva y efectiva el monitoreo, análisis y evaluación de los eventos en tiempo real y en el pasado. Con el objetivo de facilitar la toma de decisiones y la optimización de los procesos.

La figura 5.14 muestra el tablero de Grafana de incidencias configurado para una rápida visualización.



Figura 5.14: Tablero de Grafana con las principales incidencias

[19]

El uso de este tipo de tableros proporciona las siguientes ventajas:

1. Monitoreo en tiempo real: Grafana permite visualizar los eventos en tiempo real, lo que permite monitorear el estado actual del sistema o los procesos en ejecución. Esto es especialmente útil para identificar problemas y tomar medidas rápidas.
2. Acceso a datos históricos: Grafana puede conectarse a diversas fuentes de datos y proporcionar visualizaciones de eventos históricos. Esto permite realizar análisis retrospectivos y extraer información valiosa sobre patrones, tendencias y comportamientos pasados.
3. Visualización intuitiva: Grafana ofrece una amplia gama de opciones para visualizar eventos, como gráficos de líneas, barras, tablas y paneles de control personalizados. Estas visualizaciones son altamente personalizables y se pueden adaptar a las necesidades específicas del negocio o del sistema en cuestión.
4. Alertas y notificaciones: Grafana permite configurar alertas y notificaciones basadas en eventos específicos. Esto facilita la detección temprana de problemas o situaciones anormales y permite a los responsables tomar medidas de manera proactiva.
5. Centralización de datos: Grafana puede integrarse con diversas fuentes de datos, como bases de datos, sistemas de registro y herramientas de monitoreo. Esto permite centralizar los datos en un solo lugar y tener una vista holística de los eventos en tiempo real.
6. Colaboración y toma de decisiones: Los tableros de Grafana se pueden compartir y colaborar con otros usuarios, lo que facilita la colaboración en equipos y la toma de decisiones basada en datos. Esto mejora la comunicación y la transparencia en la gestión de eventos.
7. Escalabilidad y rendimiento: Grafana está diseñado para manejar grandes volúmenes de datos y ofrece un rendimiento escalable. Esto permite visualizar eventos en tiempo real, incluso en sistemas de alta carga.

5.7.9. Contenedores

El prototipo se desarrolló basado en la arquitectura de microservicios y contenedores. La idea es dividir la aplicación en pequeños servicios independientes, que se comunican entre si a través de una red para realizar una tarea específica, donde cada uno de ellos se ejecuta dentro de un contenedor Docker independiente.

Una de las principales ventajas de esta arquitectura es la capacidad de escalar y desplegar los servicios de forma independiente, lo que permite un mayor control y flexibilidad en el proceso de implementación. Además, los contenedores ofrecen una mayor portabilidad, ya que se pueden ejecutar en cualquier sistema operativo o entorno de nube, sin preocuparse por las dependencias del sistema.

Otra ventaja de los microservicios es la facilidad de mantenimiento y actualización, ya que cada servicio puede actualizarse y desplegarse sin afectar a los demás servicios, lo que reduce el riesgo de errores y problemas de compatibilidad. Adicionalmente permite una rápida integración con ambientes de Integración y Entrega Continua (CI/CD).

En resumen, el uso de microservicios y contenedores permite una mayor agilidad en el desarrollo, el despliegue y el mantenimiento del software, lo que mejora la eficiencia y la calidad de la aplicación.

La figura 5.15 muestra la aplicación Docker Desktop²¹ con los contenedores iniciados para cada servicio.

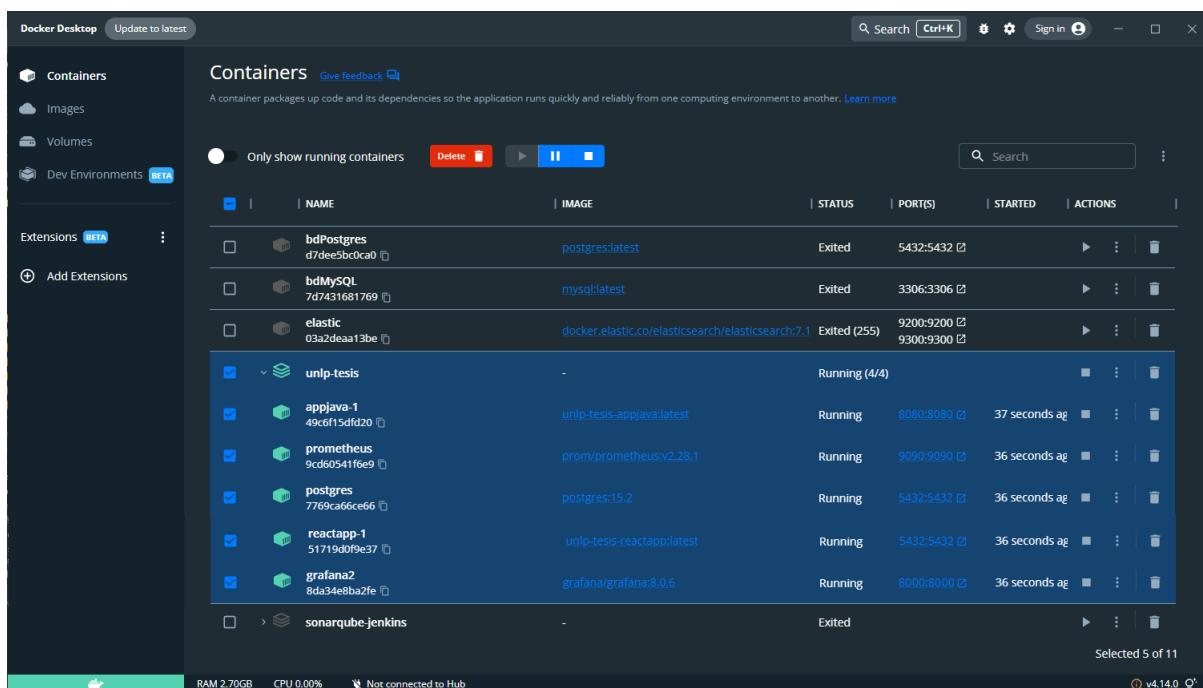


Figura 5.15: Pantalla de Docker Desktop con los servicios iniciados en contenedores

[19]

²¹<https://www.docker.com/products/docker-desktop/>

5.8. TESTEO

El proceso de prueba del software para este prototipo consta de varias etapas para garantizar la calidad y la funcionalidad del producto final. La primera etapa son las pruebas unitarias del módulo de Minería de Procesos y de la API de gestión de incidencias en la cual se comprueba que cada módulo cumpla con los requisitos de entrada y salida esperados.

La siguiente etapa es la prueba de integración de la API de gestión de incidencias y la base de datos. Se verifica que los datos se ingresen y actualicen correctamente y que la comunicación entre los componentes sea fluida.

La prueba de integración del front-end web se realiza para asegurar que las vistas y funcionalidades de la interfaz gráfica sean coherentes con las especificaciones de diseño y se comuniquen adecuadamente con la API de gestión de incidencias.

Finalmente, se realiza una prueba de integración del conjunto de microservicios que conforman el software completo, asegurando que la comunicación entre los componentes sea fluida y que los datos se envíen y procesen correctamente a través de la plataforma.

Además, se coordinarán pruebas de aceptación del usuario para verificar que el software cumpla con los requisitos de negocio y satisfaga las necesidades de los usuarios finales.

Una vez completadas todas las pruebas, se implementa el tablero de Grafana para visualizar las incidencias procesadas y realizar un seguimiento de los indicadores de desempeño clave (KPI) definidos. El tablero permite una visualización en tiempo real de la información y permite a los usuarios identificar y solucionar rápidamente las incidencias.

El proceso de prueba del software se llevó a cabo de manera exhaustiva y siguiendo las buenas prácticas actuales para garantizar la calidad y la funcionalidad del producto final.

5.9. CASOS DE PRUEBA

Luego de finalizar el desarrollo del prototipo en su totalidad se realizaron las pruebas técnicas necesarias para garantizar su correcto funcionamiento antes de la puesta en producción.

Se ejecutaron los casos de testeо listados en la tabla 5.1, basados en los requerimientos del sistema.

Las pruebas se realizaron en el ambiente de Desarrollo, se considera que el producto está listo para pasar al ambiente de Testeo ya que se obtuvo el resultado esperado en cada uno de los casos de prueba.

Si se decide implementar el prototipo en producción el usuario final realizará las pruebas correspondientes en el ambiente de Testeo, se considerará que el producto está listo para pasar a producción luego de que el testeо de usuario sea exitoso.

Tabla 5.1: Casos de Prueba realizados

Componente	Caso de Prueba	Resultado
Web Administración	CP001 - Ingreso al Sistema	OK
Web Administración	CP002 - Total de incidencias detectadas	OK
Web Administración	CP003 - Listado de incidencias con ajuste bajo	OK
Web Administración	CP004 - Listado de incidencias con exceso de tiempo	OK
Web Administración	CP005 - Detalle de incidencia	OK
Web Administración	CP006 - Registro de reclamo sobre incidencia	OK
Web Administración	CP007 - Cierre de incidencia	OK
API	CP008 - Total de incidencias con ajuste bajo	OK
API	CP009 - Total de incidencias con exceso de tiempo	OK
API	CP010 - Listado de incidencias con ajuste bajo	OK
API	CP011 - Listado de incidencias con exceso de tiempo	OK
API	CP012 - Detalle de Incidencia	OK
API	CP013 - Registro de Reclamo	OK
API	CP014 - Cierre de incidencia	OK
Exportación	CP015 - Exportación de datos a csv	No ejecutado

[19]

Los casos de prueba realizados sobre la Web de Administración se ejecutaron mediante la interacción del usuario con el sistema.

Los casos de prueba realizados sobre la API se ejecutaron utilizando Swagger y Postman interactuando directamente con el back end sin el front end (la Web de Administración) verificando manualmente la correcta generación de datos en BD.

La comprobación de los datos en Base de datos se realizó manualmente con consultas SQL.

Para la prueba de concepto y el componente de minería de procesos se procesaron los mismos archivos que se utilizaron en el análisis con las herramientas de minería de procesos estándar y se verificaron cantidades similares de incidencias encontradas en ambos casos.

Los casos de prueba del proceso de exportación de datos de Oracle no pudieron realizarse ya que para la realización del presente trabajo se realizó una exportación de datos previa, quedará pendiente para ambiente de Testeo.

Se incluyen a continuación dos ejemplos de los casos de prueba realizados

La figura 5.16 muestra el caso de prueba CP001 Ingreso al Sistema.

CP_001	Ingreso al Sistema	
Funcionalidad	Ingreso al Sistema	
Ambiente	Desarrollo	
Actor	Operador	
Descripción	Permite al operador ingresar al sistema.	
Precondición	1. El usuario debe estar registrado en el sistema. 2. El usuario debe estar habilitado.	
Secuencia normal	Paso	Acción
	1	El operador ingresa su nombre de usuario y contraseña.
	2	Se ingresará al sistema con la opción “Ingresar”.
	3	El sistema redireccionará a la pantalla de inicio
Postcondición	El usuario queda <u>logueado</u> en el sistema.	
Resultado esperado	El ingreso se realiza de forma satisfactoria.	
Resultado obtenido	Se obtuvo el resultado esperado.	
Comentarios		
Importancia	Alta	

Figura 5.16: Caso de Prueba Ingreso al Sistema

[19]

La figura 5.17 muestra el caso de prueba CP005 Detalle de incidencia.

CT_005	Detalle de incidencia	
Funcionalidad	Detalle de incidencia	
Ambiente	Desarrollo	
Actor	operador	
Descripción	Muestra el los datos registrados de una posible incidencia registrada en el sistema.	
Precondición	1. El usuario debe estar logueado en el sistema. 2. El usuario tener un perfil que permita realizar la acción. 3. Debe existir la incidencia en el sistema	
Secuencia normal	Paso	Acción
	1	El operador seleccionará la opción “Listado de incidencias”.
	2	Se mostrará el listado de incidencias registradas en el sistema.
	3	El operador seleccionará una incidencia del listado
	4	Se mostrará el detalle del de la incidencia seleccionada incluyendo datos del envío, datos de la sucursal y los movimientos registrados.
Postcondición	NA	
Resultado esperado	Se muestran los datos la incidencia seleccionada.	
Resultado obtenido	Se obtuvo el resultado esperado.	
Comentarios		
Importancia	Alta	

Figura 5.17: Caso de Prueba Detalle de incidencia

[19]

5.10. IMPLEMENTACIÓN

La implementación en el ambiente productivo se realizará luego del desarrollo del aplicativo final.

Este trabajo ha culminado con la creación de un prototipo completamente funcional que se enfoca en la aplicación de la Minería de Procesos en la distribución postal. Este prototipo se presenta como una sólida base sobre la cual se podrá desarrollar la aplicación final para su implementación en el ambiente productivo. El desarrollo del aplicativo final se llevará a cabo siguiendo estrictas buenas prácticas de programación, tales como el uso de Clean Code y la metodología Git Flow, que han sido implementadas y seguidas durante la creación del prototipo.

Asimismo, el uso de contenedores ha sido considerado como una herramienta clave en la integración de los diferentes componentes del sistema, permitiendo su fácil despliegue en un ambiente de Integración y Despliegue Continuo (CI/CD) productivo.

El presente prototipo ha sido desarrollado con el objetivo de poner en práctica los conocimientos y herramientas de la Minería de Procesos, para optimizar los procesos de distribución postal. Una vez finalizado el aplicativo final, se espera que este pueda ser integrado en el ambiente productivo, brindando una mejora significativa en la eficiencia y eficacia en el proceso de distribución postal.

CAPÍTULO 6

Conclusiones y futuras líneas de investigación

Se describen a continuación las conclusiones de este trabajo y algunas posibles futuras líneas de investigación para continuar con las mejoras al proceso de distribución Postal en la República Argentina.

6.1. CONCLUSIONES

El presente trabajo tuvo como objetivo principal aplicar la Minería de Procesos en el contexto de la distribución postal, con el fin de identificar mejoras que contribuyan a prevenir desvíos operativos y cuellos de botella en dicho proceso. Se comenzó realizando un relevamiento bibliográfico del material existente el cual permitió conocer el estado del arte. A través del análisis detallado de los registros de eventos y el descubrimiento de patrones en los datos, se logró obtener información valiosa sobre el desempeño y la eficiencia del sistema de distribución.

Luego del análisis realizado sobre el total de trazas procesadas se observa que la tasa de errores tiene un valor del 10,89 % y el tiempo medio de entrega es de 5,79 días. Estos valores brindaron información hasta el momento desconocida que permitió encontrar posibles desvíos operativos, por ejemplo los envíos que cumplen el proceso pero en un tiempo de entrega muy alto, que hasta el momento no se podían detectar, salvo ante un reclamo del cliente.

Los resultados obtenidos demostraron es posible aplicar las técnicas de Minería de Procesos en un contexto específico como es la distribución postal en la República Argentina de forma manual o a través de una herramienta informática que permita incorporar este análisis al proceso operativo.

Esto proporcionó una visión clara de las áreas de mejora necesarias para optimizar el rendimiento del sistema y reducir los tiempos de entrega.

Estos resultados de estas investigaciones dieron lugar a las publicaciones [20] y [21] incluidas en el Anexo A.

Una vez modelizado el proceso e identificados los puntos críticos, como parte de esta investigación, se desarrolló un prototipo de software completamente funcional que permite la aplicación de la

Minería de Procesos en el circuito productivo actual. Este prototipo ofrece a los actores involucrados en la distribución postal una herramienta poderosa para analizar y monitorear el rendimiento del proceso, identificar oportunidades de mejora y tomar decisiones basadas en datos con el objetivo de maximizar la eficiencia operativa.

El aporte y la contribución de esta tesis radica en la aplicación exitosa de la Minería de Procesos en un contexto específico, como lo es la distribución postal. Los hallazgos obtenidos brindan una base sólida para implementar mejoras significativas en el sistema actual, lo que puede resultar en una mayor eficiencia, reducción de costos y un mejor servicio a los clientes.

6.2. FUTURAS LÍNEAS DE INVESTIGACIÓN

La Minería de Procesos es un tema de sumo interés en la actualidad [36, 11, 27]. En el caso particular de esta tesis, se sugieren como líneas de trabajo futuras las siguientes líneas de investigación:

- Implementación de técnicas de aprendizaje automático (machine learning) en conjunto con la Minería de Procesos en el contexto de la distribución postal. Esto permitiría aprovechar el potencial de los algoritmos de aprendizaje automático para predecir desvíos operativos de forma automatizada, identificar patrones ocultos y ofrecer recomendaciones para optimizar el proceso de distribución.
- Incorporación de tecnologías emergentes como Internet de las cosas (IoT) y la inteligencia artificial (IA) en la distribución postal. Por ejemplo, el uso de sensores en los vehículos de entrega para recopilar datos en tiempo real y la aplicación de algoritmos de IA para la toma de decisiones automatizada en situaciones complejas.
- Otra línea de investigación interesante sería explorar enfoques de optimización y simulación en la distribución postal. Estos enfoques permitirían modelar y simular diferentes escenarios operativos, teniendo en cuenta variables como la carga de trabajo, la disponibilidad de recursos y las restricciones logísticas. Esto ayudaría a identificar configuraciones óptimas y estrategias de planificación que maximicen la eficiencia y la satisfacción del cliente.

Finalmente, se considera que se ha cumplido don los objetivos de este trabajo y que la Minería de Procesos aplicada en la distribución postal ha demostrado ser una herramienta valiosa para identificar desvíos operativos y cuellos de botella, permitiendo tomar medidas correctivas y optimizar el rendimiento del sistema. El desarrollo del prototipo de software proporciona una solución práctica y funcional para aplicar la Minería de Procesos en el entorno operativo actual. Estos resultados resaltan la importancia y el impacto significativo que esta investigación puede tener en la mejora continua de los procesos de distribución postal y su eficiencia operativa.

Bibliografía

- [1] BECKER, T., AND INTOYOAD, W. Context aware process mining in logistics. *Procedia CIRP* 63 (2017), 557–562. Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems.
- [2] BERTI, A., VAN ZELST, S. J., AND VAN DER AALST, W. M. P. Process mining for python (pm4py) : Bridging the gap between process- and data science. In *ICPMD 2019, ICPM Demo Track 2019 : proceedings of the ICPM Demo Track 2019, co-located with 1st International Conference on Process Mining (ICPM)* (Aachen, Germany, Jun 2019), A. Burattin, A. Polyvyanyy, and S. J. van Zelst, Eds., vol. 2374 of *CEUR workshop proceedings*, RWTH Aachen, pp. 13–16.
- [3] BRIOL, P. *Bpmn, the Business Process Modeling Notation*. Ingénierie des Processus net, 2010.
- [4] DUMAS, M., LA ROSA, M., MENDLING, J., AND REIJERS, H. A. *Fundamentals of Business Process Management*. Springer Berlin, Heidelberg, 2019.
- [5] FOWLER, M., AND BECK, K. *Refactoring: Improving the Design of Existing Code*, 1st ed. Addison Wesley, 2018.
- [6] GÜNTHER, C. W., AND VAN DER AALST, W. M. P. Fuzzy mining: Adaptive process simplification based on multi-perspective metrics. In *Proceedings of the 5th International Conference on Business Process Management* (Berlin, Heidelberg, 2007), BPM’07, Springer-Verlag, p. 328–343.
- [7] GÜNTHER, C. W. *Process mining in flexible environments*. PhD thesis, Technische Universiteit Eindhoven, 2009.
- [8] HAN, J., PEI, J., AND TONG, H. *Data Mining Concepts and Techniques*, 4th ed. Morgan Kaufmann, 2022.
- [9] HU, X., JIN, Y., AND WANG, F. Research of postal data mining system based on big data. *3rd International Conference on Mechatronics, Robotics and Automation* (2015).
- [10] IEEE. Ieee standard for extensible event stream (xes) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016* (2016), 1–50.
- [11] IMRAN, M., ISMAIL, M. A., HAMID, S., AND NASIR, M. H. N. M. Complex process modeling in process mining: A systematic review. *IEEE Access* 10 (2022), 101515–101536.
- [12] JÜRGENSEN, J. P. Trace reconstruction in system logs for processing with process mining. *Procedia Computer Science* 180 (2021), 352–357. Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020).
- [13] KEEJUN, H., YEONGWOONG, Y., DONG-GIL, N., HOON, J., YOUNGGYO, H., HYEONCHEOL, J., SUNGUK, Y., AND JUNGEUN, K. Understanding postal delivery areas in the republic of korea using multiple unsupervised learning approaches. *Etri Journal* (2022).

- [14] KEITH, B., AND VEGA, V. Process mining applications in software engineering. In *Trends and Applications in Software Engineering* (Cham, 2017), J. Mejia, M. Muñoz, Á. Rocha, T. San Feliu, and A. Peña, Eds., Springer International Publishing, pp. 47–56.
- [15] LEEMANS, S. J. J. *Enhancement & Inductive visual Miner*. Springer International Publishing, Cham, 2022, pp. 423–447.
- [16] MANESCHIJN, D. Finding the appropriate level of abstraction for process mining in logistics. *University of Twente* (June 2021).
- [17] MARTIN, R. C. *Clean Code: A Handbook of Agile Software Craftsmanship*, 1st ed. Pearson Education, 2008.
- [18] MARTIN, R. C. *Clean Architecture*. Prentice Hall, 2018.
- [19] MARTINEZ, V. Creada por el autor de esta tesis. In *Figura generada por el autor de esta tesis* (2023).
- [20] MARTÍNEZ, V., LANZARINI, L., AND RONCHETTI, F. Process mining applied to postal distribution. In *XXVII Congreso Argentino de Ciencias de la Computación (CACIC)* (2021), RedUNCI, pp. 271–280.
- [21] MARTÍNEZ, V., LANZARINI, L., AND RONCHETTI, F. Distribution analysis of postal mail in argentina using process mining. In *Communications in Computer and Information Science. Springer* (Cham, 2022), vol. 1584, Springer International Publishing, pp. 159–169.
- [22] MCKENNA, D. *The Art of Scrum: How Scrum Masters Bind Dev Teams and Unleash Agility*. CA Technologies, 2016.
- [23] NORAMBUENA, B. K., AND VEGA ZEPEDA, V. Process mining applications in software engineering. *RISTI - Revista Ibérica de Sistemas y Tecnologías de Información* 23 (2017), 51–66.
- [24] REIL, T., GROHER, E., AND SIEGFRIED, P. Process mining in supply chain management. *Supply Chain Management Journal* (2022).
- [25] SOCIETY, I. C. *Guide to the Software Engineering*, v3 ed. IEEE, 2014.
- [26] SUTHERLAND, J. *Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo*, bilingual ed. Oceano De Mexico, 2021.
- [27] URREA-CONTRERAS, S., FLORES-RIOS, B. L., ASTORGA-VARGAS, M. A., AND IBARRA-ESQUER, J. E. Process mining perspectives in software engineering: A systematic literature review. In *2021 Mexican International Conference on Computer Science (ENC)* (2021), pp. 1–8.
- [28] VAN DER AALST, W., WEIJTERS, T., AND MARUSTER, L. Workflow mining: discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16, 9 (2004), 1128–1142.
- [29] VAN DER AALST, W. M. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st ed. Springer-Verlag Berlin Heidelberg, 2011.
- [30] VAN DER AALST, W. M. Process mining manifesto. In *Business Process Management Workshops* (Berlin, Heidelberg, 2012), Springer Berlin Heidelberg, pp. 169–194.
- [31] VAN DER AALST, W. M. *Process Mining: Data Science in Action*, 2nd. ed. Springer Berlin Heidelberg, 2016.
- [32] VAN ECK, M. L., LU, X., LEEMANS, S. J., AND VAN DER AALST, W. M. Pm2: a process mining project

- methodology. *Eindhoven University of Technology, The Netherlands* (2017).
- [33] VOM BROCKE, J., AND ROSEMANN, M. *Handbook on Business Process Management 1*, 1st ed. Springer, 2015.
- [34] WEIJTERS, A. J. M. M., AND VAN DER AALST, W. M. Process mining with the heuristicsminer algorithm. In *Information Systems IE&IS* (2006), Technische Universiteit Eindhoven.
- [35] WITTEN, IAN H. AND FRANK, E., HALL, M. A., AND PAL, C. J. *Data Mining Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann, 2017.
- [36] ZERBINO, P., STEFANINI, A., AND ALOINI, D. Process science in action: A literature review on process mining in business management. *Technological Forecasting and Social Change* 172 (2021), 121021.

ANEXOS

ANEXO A

Publicaciones

Los resultados de estas investigaciones realizadas en el marco de esta tesis fueron difundidos a través de los siguientes artículos:

- Martinez, V., Lanzarini, L., Ronchetti, F. (2021). **Process Mining Applied to Postal Distribution.** Pertenece al libro: Memorias del XXVII Congreso Argentino de Ciencias de la Computación. RedUNCI. ISBN: 978-987-633-574-4. Págs: 271-280.
- Martinez, V., Lanzarini, L., Ronchetti, F. (2022). **Distribution Analysis of Postal Mail in Argentina Using Process Mining.** In: Pesado, P., Gil, G. (eds) Computer Science – CACIC 2021. CACIC 2021. Communications in Computer and Information Science, vol 1584. Springer, Cham.

Minería de Procesos aplicada a la distribución postal

Victor Martinez¹, Laura Lanzarini^{1,2}, and Franco Ronchetti^{1,2,3}

¹ Facultad de Informática, Universidad Nacional de La Plata

² Instituto de Investigación en Informática LIDI (UNLP-CIC)

³ Comisión de Investigaciones Científicas de la provincia de Buenos Aires
(CIC-PBA)

martinezvictor@hotmail.com, {laural, fronchetti}@lidi.info.unlp.edu.ar

Resumen La minería de procesos es una técnica que permite analizar procesos de negocio a través de registros de eventos. En este artículo se utilizan distintas técnicas de Minería de Procesos para analizar datos basados en la distribución postal de productos en la República Argentina durante los años 2017 a 2019. Los resultados obtenidos permiten afirmar que el 85 % de los envíos realizados se ajustan exactamente al modelo. El análisis de las situaciones con bajo nivel de ajuste al proceso descubierto constituyó una herramienta de identificación rápida de algunos problemas recurrentes en la distribución facilitando el análisis de los desvíos ocurridos. A futuro se espera poder incorporar estas técnicas en la construcción de alertas tempranas que adviertan de la existencia de desvíos excesivos del proceso.

Keywords: Process Mining, Data Mining, Postal Distribution, Postal Processes, Business Process Management

1. Introduction

En la actualidad la mayoría de los sistemas de información dejan registro de las actividades realizadas, ya sean sistemas de gestión empresarial (ERP, CRM, WMS, BI, etc) como también desarrollos propios de cada empresa. Por lo general se registra qué se realizó, quién lo realizó y cuándo se realizó, entre otras cosas. A partir de estos datos almacenados puede obtenerse información que describa el proceso y buscar mejoras o solucionar problemas.

Las técnicas de Minería de Procesos se utilizan para analizar estos datos y encontrar patrones de comportamiento. Se puede encontrar con que tareas comienza un proceso, cual es la secuencia que se cumple, y con que tareas termina. De esta forma se puede “descubrir” el proceso que se está realizando para una determinada actividad.

Con la minería de procesos se pueden responder preguntas como: ¿qué fue lo que pasó realmente?, ¿por qué pasó?, ¿qué podría pasar en el futuro?, ¿cómo se puede mejorar el control del proceso?, ¿cómo se puede rediseñar el proceso para mejorar la performance? [1].

Una de sus principales ventajas es que permite trabajar directamente sobre datos reales y obtener el verdadero comportamiento del proceso, el cual, en algunos casos, no es el diseñado originalmente.

En este trabajo se aplicarán las técnicas de minería de procesos a la distribución postal en la República Argentina para analizar su funcionamiento con el objetivo de encontrar desvíos operativos, cuellos de botella y otros problemas que impactan negativamente en la calidad del servicio.

En la distribución postal se lleva un registro de todas las actividades, desde el ingreso del producto hasta su entrega al cliente. Entre estas actividades se encuentra la recepción del envío, ingreso a un almacén, traslados internos o intentos de entrega, entre otras. Todas deben realizarse en un orden y en un intervalo de tiempo específicos. Ocasionalmente se producen desvíos, estos pueden ser redundancia o inconsistencia de tareas (se repiten o no se realizan en el orden correspondiente), excesivo tiempo para su finalización u otras.

En la literatura se pueden encontrar trabajos que relacionan el negocio postal, la inteligencia de datos y la minería de procesos. Tal es el caso de [2] que utiliza data mining en un entorno de Big Data en el correo de China. Debido a la problemática del negocio postal, en dicho artículo se utilizaron técnicas de clustering para agrupar a los clientes según el comportamiento, los hábitos de consumo y el foco de interés logrando generar una estrategia de Marketing postal más precisa y efectiva con resultados muy satisfactorios. Otro caso es [3] donde se aplica process mining en logística buscando similitudes y diferencias entre distintos procesos de entrega en un contexto cambiante de manufactura y logística. En dicho trabajo se comparan diferentes procesos utilizando técnicas de agrupamiento en busca de lograr una documentación automatizada de procesos en un contexto cambiante. En [4] se presenta una metodología que sirve de guía para la ejecución de proyectos de process mining que describe las diferentes etapas. Además se muestra como caso de estudio su aplicación práctica en el proceso de compras de IBM.

En este artículo se utilizarán distintas técnicas de Minería de Procesos para analizar datos basados en la distribución postal de productos en la República Argentina durante los años 2017 a 2019. Los autores de este trabajo desean manifestar que, a la fecha de generación del presente documento, no tienen conocimiento de la existencia de trabajos similares que implementen la minería de procesos en la distribución postal en la República Argentina.

2. Minería de procesos (Process Mining)

El punto de partida de la Minería de Procesos es el registro de eventos. Se asume que el proceso a analizar requiere del registro de una serie de eventos secuenciales pertenecientes a la actividad y que se relacionan con un caso particular. Si bien se pueden almacenar datos adicionales de cada evento deben registrarse obligatoriamente la fecha del evento (día y hora), el identificador del caso y el tipo de evento.

Los tres tipos básicos de minería de procesos [5] son:

- *Descubrimiento:* las técnicas de descubrimiento toman como punto de entrada un log de procesos y generan un modelo sin ninguna información adicional. Un ejemplo es el algoritmo Alpha [6] que toma los datos del log y genera una red de Petri que explica el comportamiento reflejado en el log. Como toda técnica que extrae conocimiento a partir de los datos, la calidad del proceso descubierto dependerá del grado de representación de los eventos relevados con respecto al funcionamiento de dicho proceso. No se pueden descubrirse partes del proceso que no se encuentren representadas por los eventos.
- *Verificación de Conformidad:* Consiste en realizar la comparación de un modelo existente (puede ser el descubierto anteriormente u otro) con la secuencia de eventos real para encontrar desvíos y verificar el funcionamiento. Suelen utilizarse aplicaciones capaces de generar representaciones gráficas y animaciones que posibilitan observar el comportamiento real y ver cuánto se ajusta la realidad al proceso definido originalmente. Su principal ventaja es que muestra la realidad, no es una simulación, por lo que permite realizar un análisis mucho más preciso.
- *Mejoramiento:* Se busca extender o mejorar el proceso existente a través de la información subyacente en la secuencia de eventos. A diferencia de la Verificación de Conformidad, donde se comparan los datos con el modelo, aquí se busca modificar el proceso

Se debe regular el grado de abstracción a utilizar durante el análisis considerando los siguientes aspectos: ajuste, precisión, generalización y simplicidad [1]. Existe una relación de compromiso entre ellos que debe ser tenida en cuenta si se espera obtener buenos resultados. El ajuste se refiere a la habilidad del modelo para explicar el comportamiento observado. La precisión se refiere a la exactitud con la que se ejecuta el proceso; aquí es importante no sobreajustar el modelo a los datos de entrada porque se perderá generalización impidiendo lograr el nivel de abstracción deseado. La simplicidad también se ve comprometida por el sobreajuste ya que se logra agregando mayor detalle a la descripción del proceso.

Finalmente es importante destacar que el resultado obtenido del análisis con minería de procesos está altamente vinculado a la calidad de los datos de entrada. De hecho se sabe que siempre hay una cantidad de ruido en los datos el cual puede deberse a trazas incompletas, intervalos que no se hayan registrado correctamente o duplicidad de datos. Esta información puede distorsionar o falsificar el resultado del análisis [7].

Generalmente es necesario verificar los datos de entrada y realizar un procesamiento previo para eliminar la mayor cantidad de ruido posible.

3. Descubrimiento del proceso de Distribución Postal

En esta sección se llevará a cabo el descubrimiento del proceso a partir de datos basados en la distribución postal de productos en la República Argentina durante los años 2017 a 2019.

El proceso de distribución postal abarca diferentes tipos de productos; en todos los casos el proceso consiste de una recepción del producto del remitente (ya sea por un medio físico o un canal digital) y una entrega al destinatario, pudiendo esta ser exitosa o no. La no entrega del producto no implica un error de proceso ya que puede haber motivos que lo justifiquen.

El proceso registra al menos las siguientes actividades: recepción e identificación, envío para su distribución, uno o más intentos de entrega, espera en centro de distribución y devolución (si no fue posible la entrega). En todos los casos se debe dejar registro de cada uno de los pasos realizados asociados a un identificador único de envío, lo que permite conocer el estado actual del mismo e informarlo al cliente.

Extracción de los datos

El primer paso en el descubrimiento del modelo consiste en la recolección y preprocesamiento de los datos a utilizar. En particular, para este caso de estudio se utilizaron envíos de productos con dos intentos de entrega en el domicilio. El procedimiento actual establece que aquellos productos que no pueden ser entregados se guardan un tiempo a la espera de que el destinatario venga a retirarlo; transcurrido ese tiempo se realiza la devolución al remitente.

Se define una traza como un envío. Cada movimiento que se registre de ese envío es un evento. Se considera una traza completa cuando el envío tiene registrado su ingreso y su fin, con entrega exitosa o no.

Como resultado de la recolección de datos se generó una tabla o muestra de alrededor de 33.000 trazas y un total de 78.000 eventos. Para cada evento se registraron los campos mínimos necesarios para el análisis; estos son: identificador de traza, identificador de evento, fecha de evento, descripción del evento (ver Figura 1). Cada traza puede tener uno o más eventos asociados, se necesita identificador de traza e identificador de cada uno de los eventos para armar la historia de la traza.

De esta forma se dispone de los distintos pasos o eventos que se produjeron a lo largo del envío de cada producto.

Luego se transformaron los datos utilizando el formato XES [8]. XES es una gramática para un lenguaje basado en etiquetas cuyo objetivo es proporcionar a los diseñadores de sistemas de información una metodología unificada y extensible para capturar los comportamientos de los sistemas mediante registros y flujos de eventos [8]. De esta forma se agiliza el manejo de los datos pudiendo ser procesados por distintas herramientas de manera más eficiente.

A fin de facilitar el descubrimiento del proceso correcto, se eliminaron todas las trazas incompletas entendiendo por tales a aquellas trazas que no tengan o bien un evento de inicio (Recepción) por algún error de carga o bien un evento final (entrega, devolución, destinatario inexistente, fallecido, rechaza el envío o se mudó). Esto último podría ocurrir por error de carga o porque todavía no transcurrió el tiempo estipulado para finalizar el proceso.

Identificador de traza	Identificador de evento		Descripción del evento		Fecha evento
	123 trazalD	123 EveID	abc eveDescrip	eveFecha	
31	481,053	0	INGRESADO	2017-08-16 10:45:22	
32	481,053	2	1 INTENTO DE ENTREGA	2017-08-18 11:15:00	
33	481,053	9	DEVOLUCION	2017-08-18 13:00:00	
34	481,054	0	INGRESADO	2017-08-16 10:45:28	
35	481,054	2	1 INTENTO DE ENTREGA	2017-08-22 12:05:00	
36	481,054	9	DEVOLUCION	2017-08-22 17:25:00	
37	481,055	0	INGRESADO	2017-08-16 10:45:27	
38	481,055	1	ENTREGADO	2017-08-22 15:13:00	

Figura 1. Ejemplo de eventos extraídos para el análisis

Utilizando un filtro de reglas heurísticas simples se identificaron las trazas completas dejando sólo aquellas que tienen su estado inicial y los estados finales válidos.

Como resultado de este proceso de filtrado se obtuvieron aproximadamente 16.000 trazas con 43.000 eventos.

3.1. Modelo del proceso

Para generar el modelo del proceso se utilizó un algoritmo clásico de minería de procesos, el algoritmo Alpha, propuesto por primera vez por van der Aalst, Weijters y Mărușter[6]. El objetivo de este algoritmo es reconstruir la causalidad a partir de un conjunto de secuencias de eventos. Construye Redes de Petri con propiedades especiales (redes de flujo de trabajo) a partir de registros de eventos (como los que podría recoger un sistema ERP). Cada transición en la red corresponde a una tarea observada.

En lo que se refiere al caso de estudio de la distribución postal, si bien es posible construir el modelo directamente a partir de las 16000 trazas mencionadas previamente, se seleccionaron manualmente algunos casos buscando los más representativos para encontrar el proceso ideal que deben cumplir las trazas, simplificando de esta manera esta etapa de descubrimiento del proceso.

La Figura 2 ilustra la red de petri correspondiente al proceso descubierto que deberán cumplir las trazas.

El proceso encontrado es el siguiente: Todas las trazas deben comenzar con un evento de ingreso y luego sale a distribución. Si se puede entregar, se registra el evento y finaliza el proceso. Si no se puede entregar, en caso de ser un evento final (fallecido, datos faltantes en dirección, no existe dirección, desconocido, etc.), se registra el motivo y finaliza el proceso. Si no es un evento final (por ejemplo se pudo llegar a la dirección pero no había nadie) se registra un primer intento de entrega. Posteriormente se realiza una nueva visita. Si se pudo entregar, se registra el evento y finaliza el proceso. Si en la segunda visita no

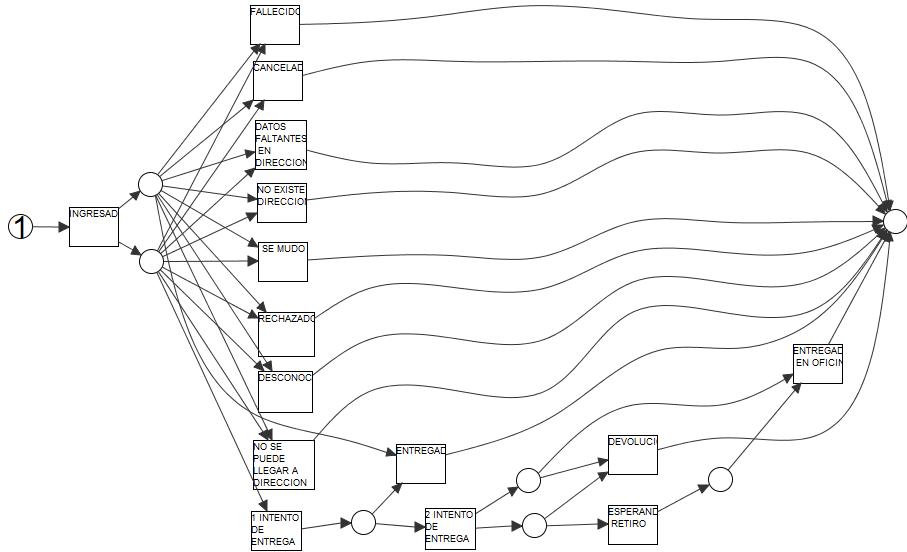


Figura 2. Proceso descubierto que deberán cumplir las trazas

se puede entregar, se guarda el envío en oficina por un tiempo esperando que el destinatario venga a retirarlo y luego se registra la entrega en oficina. Transcurrido el tiempo de espera, si el destinatario no vino a retirar el envío, se registra la devolución. Esos son los posibles escenarios que contempla el proceso.

3.2. Verificación del modelo

El modelo de proceso descubierto en la etapa anterior fue generado a partir de un subconjunto de trazas seleccionadas previamente. En esta sección se llevará a cabo la verificación de conformidad. Tal como se explicó previamente, se buscará determinar cuánto se ajustan las trazas de la muestra al proceso. Se tendrán en cuenta qué pasos se cumplen, cuáles no y si hay algunos adicionales que no están reflejados en el proceso descubierto. Para ello, se comparará cada una de las trazas obtenidas contra el proceso encontrado en el punto anterior. Se utilizará la muestra de datos obtenida en la sección anterior, la cual está conformada por 16.811 trazas con un total de 43.888 eventos.

El resultado se muestra en la Figura 3. Allí se resaltan los eventos más comunes en color oscuro y los recorridos más frecuentes con una línea más gruesa. Se observa que la mayoría de las trazas siguen el camino de la entrega, ya sea en el primer intento o en el segundo.

Las estadísticas de la muestra indican que en un 80 % de los casos la pieza es entregada, ya sea en el primer o segundo intento y que el resto se distribuye uniformemente. En base a estas observaciones se puede afirmar que el 85 % de las trazas se ajusta perfectamente al proceso descubierto con una cantidad promedio de 2.6 eventos en cada traza.

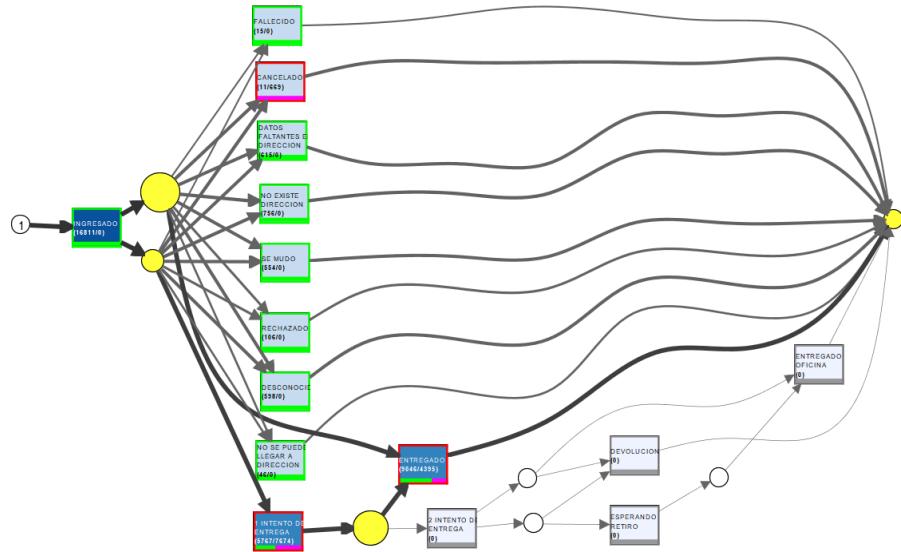


Figura 3. Recorrido de las trazas: eventos más comunes en oscuro y recorridos más frecuentes en línea más gruesa

Sin embargo, algunas trazas no verifican exactamente el proceso descubierto ya sea porque se saltean pasos, o no los hacen en el orden correspondiente o hacen pasos por fuera del modelo. Estas trazas poseerán un valor de ajuste que será más bajo cuanto más se diferencien del modelo. Aquellas que tengan un ajuste menor al 50 % son el objeto de análisis de esta sección ya que representan un gran desvío operativo.

A partir de estos resultados se enfoca el análisis en dos caminos, por un lado las trazas que tienen una excesiva cantidad de movimientos y por otro aquellas que no se ajustan al modelo, ya sea porque hay eventos faltantes o porque no cumplen la secuencia correspondiente.

En el primer caso, dado que la cantidad de movimientos por traza sigue una distribución normal con media 2,6 y desvío 0,95 se consideró adecuado tomar un valor superior a la media mas tres desvíos estándar como un valor representativo de una cantidad excesiva de movimientos.

En base a esto, se procedió a filtrar y posteriormente a analizar las trazas con más de 6 movimientos encontrando 108 casos.

El segundo caso analizado fue el de las trazas que no se ajustaban exactamente al proceso descubierto. En esta oportunidad se fijó el valor del umbral de ajuste en 0.5 por considerar que las trazas que contienen al menos el 50 % de sus movimientos desalineados (no se corresponden) con respecto a lo indicado por el modelo representan una fuerte distorsión del procedimiento y deben ser inspeccionadas. Como se dijo previamente, esto puede ocurrir o bien porque hay

3760173.0 8 events	3780088.0 6 events
INGRESADO #1 02.01.2020 16:29:45.000	INGRESADO #1 20.01.2020 21:00:14.000
1 INTENTO DE ENTREGA #2 13.01.2020 12:25:00.000	1 INTENTO DE ENTREGA #2 21.01.2020 11:12:00.000
2 INTENTO DE ENTREGA #3 14.01.2020 11:20:00.000	1 INTENTO DE ENTREGA #3 22.01.2020 11:16:00.000
2 INTENTO DE ENTREGA #4 15.01.2020 11:20:00.000	ESPERANDO RETIRO #4 22.01.2020 16:04:31.000
1 INTENTO DE ENTREGA #5 16.01.2020 11:20:00.000	DEVOLUCION #5 29.01.2020 15:02:07.000
2 INTENTO DE ENTREGA #6 17.01.2020 09:39:00.000	ENTREGADO #6 31.01.2020 11:15:00.000
2 INTENTO DE ENTREGA #7 20.01.2020 10:10:00.000	
DEVOLUCION #8 24.01.2020 10:10:00.000	

Figura 4. Eventos repetidos e inconsistencias

eventos en la trazas que no aparecen en el modelo o porque los eventos de la traza no cumplen con la secuencia correspondiente.

Como resultado de esto se observó que se trataban de repeticiones de eventos en diferentes días o de registros inconsistentes. La Figura 4 ilustra ambas situaciones. En el cuadro de la izquierda se puede observar un caso en el que aparece registrada una primera visita luego de haber ocurrido previamente una segunda; esta situación sólo puede deberse a un error en el registro. En el cuadro de la derecha de la misma figura se observa una inconsistencia ya que no es posible entregar un producto que previamente fue devuelto.

También es posible utilizar herramientas visuales para generar animaciones que faciliten la comprensión de estas situaciones. En este caso particular, con el conjunto de trazas que tienen un ajuste inferior al 50 % se utilizó Inductive Visual Miner[9] para generar una animación que permite ver cómo se suceden los eventos de cada traza en una línea de tiempo. La Figura 5 ilustra esta animación. En dicha figura cada traza está simbolizada con un token (o círculo) que va recorriendo las diferentes etapas del proceso. Esta representación visual permite observar que unas trazas demoran más que otras y que en ocasiones se presentan algunos comportamientos erróneos. Por ejemplo en la figura 5 se observan movimientos de retroceso en las trazas; en particular, se han marcado con círculos que 25 trazas vuelven del segundo intento al primero y 20 vuelven sobre sí mismas.



Figura 5. Trazas que no cumplen con el modelo, en círculo la cantidad que retroceden en de un estado en lugar de avanzar (ver sentido de la flecha)

La exportación de las trazas que no se ajustan al modelo permite un análisis detallado de los motivos que ocasionan los desvíos en busca de mejoras al proceso de distribución.

4. Conclusión y futuras líneas de investigación

En este artículo se han utilizado distintas técnicas de Minería de Procesos para analizar datos reales de distribución postal de productos en la República Argentina durante los años 2017 a 2019. No se encontraron a la fecha de generación del presente documento trabajos similares que implementen la minería de procesos en la distribución postal en la República Argentina.

Por medio del modelo generado a partir de trazas seleccionadas se descubrió el proceso que realmente se lleva a cabo. Es importante remarcar el impacto que el criterio de selección de dichas trazas tiene en el modelo obtenido. Los primeros modelos, generados a partir de la totalidad de las trazas, tenían un detalle excesivo de la situación lo cual dificultaba su representación e interpretación posteriores. Actualmente se está trabajando para poder identificar las trazas más frecuentes a partir del modelo inicial para luego filtrar en forma automática las más representativas.

Por su parte, la verificación de conformidad ha permitido determinar situaciones anómalas de interés. Se identificaron tanto los casos que no cumplieron con el modelo como también aquellos que lo hicieron pero fuera de los tiempos esperados o con redundancia de tareas. Se deberá analizar si estos casos representan errores manuales en la carga y buscar una solución.

Se utilizó un umbral de ajuste para establecer el grado de distorsión mínimo que una traza debe cumplir para no afectar el desarrollo del proceso. Este factor debe ser analizado con más detalle para determinar su valor en base al caso de estudio del cual se trate.

Como línea de trabajo futura se propone continuar trabajando con las técnicas de minería de procesos no sólo para modelar situaciones que ya han ocurrido

sino para poder insertar en el sistema alertas tempranas que adviertan de la existencia de desvíos excesivos del modelo.

Referencias

1. Process Mining: Data Science in Action, Wil van der Aalst,978-3-662-49850-7,2016,Springer
2. Research of Postal Data mining system based on big data, Xia Hu1; Yanfeng Jin1; Fan Wang, 3rd International Conference on Mechatronics, Robotics and Automation, 2015, 10.2991/icmra-15.2015.124, https://www.researchgate.net/publication/300483008_Research_of_Postal_Data_mining_system_based_on_big_data
3. Context Aware Process Mining in Logistics, Mitchell M. Tseng; Hung-Yin Tsai; Yue Wang, 2017,The 50th CIRP Conference on Manufacturing Systems, <https://www.sciencedirect.com/science/article/pii/S2212827117303311>
4. PM2: a Process Mining Project Methodology,Maikel L. van Eck; Xixi Lu; Sander J.J. Leemans; Wil M.P. van der Aalst,Eindhoven University of Technology, The Netherlands, http://www.processmining.org/_media/blogs/pub2015/pm2_processminingprojectmethodology.pdf
5. Wil van der Aalst, The Process Mining Manifesto by the IEEE Task Force, 2012, <https://www.tf-pm.org/resources/manifesto>
6. Workflow mining: discovering process models from event logs, W. van der Aalst; T. Weijters; L. Maruster, 1041-4347, 2004, IEEE, <https://ieeexplore.ieee.org/document/1316839>
7. Process mining in flexible environments, Christian Walter Gunther,978-90-386-1964-4,2009,Technische Universiteit Eindhoven, <https://research.tue.nl/en/publications/process-mining-in-flexible-environments>
8. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams, IEEE Std 1849-2016, 2016, DOI 10.1109/IEEEESTD.2016.7740858
9. inductive visual miner, Sander J.J. Leemans, 2017, <http://leemans.ch/leemansCH/publications/ivm.pdf>
10. Reinventing the Postal Sector in an Electronic Age, Michael A. Crew; Paul R. Kleindorfer,978-1849803601,2011,Edward Elgar Publishing

Análisis de la distribución postal en Argentina utilizando Minería de Procesos

Victor Martinez¹[0000-0003-3581-5704], Laura Lanzarini²[0000-0001-7027-7564],
and Franco Ronchetti^{2,3}[0000-0003-3173-1327]

¹ Facultad de Informática, Universidad Nacional de La Plata

² Instituto de Investigación en Informática LIDI (UNLP-CIC)

³ Comisión de Investigaciones Científicas de la provincia de Buenos Aires (CIC-PBA)

Resumen La minería de procesos combina una serie de técnicas que permiten analizar procesos de negocio únicamente con registros de eventos. Este artículo continúa la investigación realizada en [1] para analizar datos basados en la distribución postal de productos en la República Argentina entre los años 2017 a 2020. Los resultados obtenidos inicialmente dejaron en evidencia que el 85 % de los envíos realizados cumplen el proceso correctamente. También se identificaron rápidamente los casos que no cumplían el modelo y se encontraron problemas recurrentes que facilitan el análisis para una mejora del proceso. Los problemas más comunes fueron trazas que no respetan el orden de tareas, exceso de movimientos o movimientos faltantes y trazas que cumplen el proceso pero demoran demasiado tiempo. En este artículo se incorporó un análisis de performance por medio del cual se descubrieron trazas que, si bien se ajustan correctamente al proceso, tienen desvíos operativos debido al excesivo tiempo de finalización. Se espera poder incorporar estas técnicas en el proceso mediante la construcción de alertas tempranas que adviertan de la existencia de este tipo de situaciones para mejorar la calidad del servicio.

Keywords: Process Mining · Data Mining · Postal Distribution · Postal Processes · Business Process Management

1. Introduction

Actualmente existe una amplia variedad de sistemas de información que dan soporte a las diferentes áreas de cada empresa ya sean propias del negocio o administrativas. Entre estas herramientas pueden encontrarse desarrollos ad hoc para cada compañía o herramientas de gestión como CRM, ERP, WMS, TMS, BI, etc.

Dada la facilidad y el bajo costo para almacenar datos en estos días, gran parte de estos sistemas dejan información para auditoría o resolución de errores en archivos de texto o bases de datos. Entre otras cosas, se registra qué se realizó, cuándo se realizó, quién lo hizo, si hubo algún error, etc. Analizando estos datos puede obtenerse información acerca de errores o problemas ocurridos, cual es el

proceso de negocio que se realiza y como proponer mejoras o encontrar soluciones a diversos inconvenientes operativos.

La minería de procesos proporciona técnicas que permiten analizar estos registros y obtener información.

La minería de procesos puede responder interrogantes cómo: ¿cómo se puede mejorar el control del proceso?, ¿qué ocurrió realmente?, ¿por qué ocurrió?, ¿qué podría suceder en el futuro?, ¿cómo se puede mejorar el proceso para incrementar la performance? [2].

Al trabajar directamente con datos productivos se obtiene el comportamiento real que se realiza para el negocio y el proceso completo que se está realizando, el cual en algunos casos puede diferir del que se diseñó originalmente.

El caso de estudio en este trabajo es la distribución de productos postales. Se trata de un proceso en el que se registran distintos movimientos conformados por eventos tales como la recepción del producto en sucursal, el ingreso a un centro de distribución, los movimientos internos y los diversos intentos de entrega. Todas las tareas están claramente definidas y deben realizarse en un orden e intervalo de tiempo determinado. En la operación diaria se producen desvíos por retrasos en la ejecución de tareas o inconsistencia (tareas repetidas o realizadas en un orden incorrecto)

Existen algunos trabajos que vinculan el negocio postal, la minería de procesos y la inteligencia de datos. Por ejemplo [3] donde se usa data mining para el correo de China en un entorno de Big Data. Dada la complejidad del negocio postal se utilizaron técnicas de clustering que agrupaban a los clientes según los hábitos de consumo, los principales intereses y el comportamiento para lograr una estrategia de Marketing más efectiva y precisa. Los resultados fueron muy satisfactorios. Otro caso es [4] aquí se aplica Process Mining en una cadena de logística y manufactura para buscar similitudes y diferencias de los diferentes procesos de entrega en un entorno cambiante. Para ello se comparan diferentes procesos utilizando técnicas de agrupamiento para automatizar la documentación de procesos. Finalmente en [5] se muestra una metodología que se puede utilizar como guía en proyectos de process mining y como ejemplo se da el caso de estudio de su aplicación en IBM.

Este trabajo continúa la investigación publicada en [1] referido a la aplicación de técnicas de Minería de Procesos a la distribución Postal en la República Argentina con el objetivo de analizar su funcionamiento, encontrar desvíos o inconvenientes en la distribución y proponer mejoras que mejoren la calidad del servicio. A lo allí reportado, en el presente trabajo se ha incorporado un análisis de performance por medio del cual es posible identificar un nuevo tipo desvíos operativos. Los datos analizados corresponden a la distribución postal efectuada en la República Argentina en el período comprendido entre 2017 y 2020. Cabe destacar que, a la fecha de publicación de este artículo los autores no conocen la existencia de trabajos con estas características para el negocio postal.

2. Minería de procesos (Process Mining)

La Minería de Procesos opera sobre los archivos de log del sistema de información a analizar. Comienza con la extracción de la información de todos los eventos de una determinada actividad y luego, mediante un análisis automático, se determina cual es el proceso de negocio, que actividades lo componen y que secuencia se debe cumplir. De esta forma es posible hacer un análisis y responder preguntas como ¿Qué fue lo que pasó?, ¿Por qué pasó?, ¿Qué es lo que podría pasar en el futuro?, ¿Cómo se puede mejorar el control?, ¿Cómo se puede mejorar la performance? [2] Con la información obtenida luego de la interpretación del modelo se podrán realizar diferentes tipos de análisis como por ejemplo: quién realiza la actividad, si la actividad está completa y si se realiza en un intervalo de tiempo coherente.

La minería de procesos puede clasificarse en tres tipos [6]:

- *Descubrimiento:* Se trata de técnicas capaces de modelizar el proceso de negocio que se está realizando únicamente a partir de la secuencia de logs correspondientes, estos logs pueden extraerse de archivos, bases de datos o algún otro medio. Posiblemente el algoritmo más popular para realizar esta tarea es el algoritmo Alpha [7] que genera un red de Petri con el proceso descubierto.

La calidad del modelo generado está relacionada en forma directamente proporcional a la calidad de los datos de entrada. Esta es una característica común a cualquier técnica de extracción de conocimiento inductiva. Es fundamental que las secuencias de eventos relevadas representen completamente al proceso a modelizar. De lo contrario, quedarán aspectos sin descubrir.

- *Verificación de Conformidad:* Permite contrastar una secuencia de eventos real contra un modelo de proceso (puede ser el descubierto anteriormente u otro) para determinar cuanto se cumple y cuáles son las ocurrencias que se desvían.

Pueden utilizarse aplicaciones que representan gráficamente los resultados y realizan animaciones para amenizar la presentación de los resultados. El análisis resultante de esta operatoria puede ser de una gran precisión ya que se trabaja con datos reales, no es una simulación.

- *Mejoramiento:* Se trata de mejorar el proceso existente a partir del análisis realizado. Se diferencia de la Verificación de Conformidad porque aquí el foco está puesto en el proceso y no en la ejecución de las trazas.

Existen adicionalmente cuatro aspectos a tener en cuenta para obtener resultados de calidad en la Minería de Procesos. Estos son: precisión, ajuste, generalización y simplicidad [2]. Debe mantenerse una relación balanceada entre estas cuatro fuerzas para que el modelo descubierto sea representativo del proceso y fácil de comprender.

Por último pero, no menos importante, cabe destacar que generalmente se encuentra una gran cantidad de ruido en los datos de entrada que puede deberse a duplicidad de datos o trazas incompletas que pueden distorsionar el resultado del análisis [8]. Como en otras técnicas de minería se suele realizar un procesamiento

previo para mejorar la calidad de los datos de entrada con el fin de eliminar datos incompletos o corruptos que puedan llevar a una modelización errónea.

3. Minería de Procesos en la Distribución Postal

Se comenzará con el descubrimiento del proceso a partir de datos de distribución postal en la República Argentina entre los años 2017 a 2020.

La distribución postal puede aplicarse a varios productos tales como cartas, telegramas o encomiendas o algo menos "tradicional" por ejemplo e-commerce.

En todos los casos durante el proceso se cumplen al menos los siguientes pasos: registro y recepción del producto a distribuir, encaminamiento interno por una o más plantas o centros de distribución, uno o más intentos de entrega, entrega efectiva o devolución si no se pudo entregar. Siempre queda un registro de cada paso realizado, dejando constancia de quién lo realizó, que realizó y cuando lo hizo. Este registro está asociado a un identificador único de envío que permite informar estos eventos al cliente.

3.1. Extracción de los datos

Para descubrir el modelo y luego analizar los eventos se realizará una extracción y procesamiento previo de los datos a utilizar. Para este caso de estudio en particular se utilizaron envíos de productos que requirieron dos intentos de entrega. El procedimiento en este caso es el siguiente: el producto sale a distribución, si no se pudo entregar por algún motivo que no sea definitivo se intenta una nueva entrega al día siguiente. Si tampoco se pudo entregar se aguarda en oficina un tiempo a que el destinatario venga a retirar (se deja un aviso en cada visita). Transcurrido el tiempo establecido si el cliente no retiró se devuelve el envío al remitente.

Se considera cada envío como una traza y cada movimiento como un evento. La traza se considerará completa cuando el envío tiene registrado el ingreso, y el fin, pudiendo ser la entrega exitosa o no.

Como resultado de la recolección de datos se generó una muestra que contiene alrededor de 33.000 trazas con más de 77.000 eventos (ver Figura 1). Cada traza tiene al menos dos eventos asociados: identificador de traza e identificador de cada evento. En particular, en todos los eventos del caso de estudio se registraron los campos necesarios para el análisis: identificador de traza, identificador de evento dentro de la traza, descripción, fecha de evento

Con esta información es posible reconstruir cada traza y armar el modelo correspondiente.

Luego, los datos se convierten a formato XES [9]. XES Propone un lenguaje basado en etiquetas que proporciona una metodología unificada y extensible para registrar comportamiento en los sistemas de información [9]. Los datos estructurados de esta forma se pueden procesar en una amplia variedad de herramientas de una forma eficiente.

Identificador de traza	Identificador de evento		Descripción del evento		Fecha evento
	123 trazalD	123 EveID	abc eveDescrip	eveFecha	
31	481,053	0	INGRESADO	2017-08-16 10:45:22	
32	481,053	2	1 INTENTO DE ENTREGA	2017-08-18 11:15:00	
33	481,053	9	DEVOLUCION	2017-08-18 13:00:00	
34	481,054	0	INGRESADO	2017-08-16 10:45:28	
35	481,054	2	1 INTENTO DE ENTREGA	2017-08-22 12:05:00	
36	481,054	9	DEVOLUCION	2017-08-22 17:25:00	
37	481,055	0	INGRESADO	2017-08-16 10:45:27	
38	481,055	1	ENTREGADO	2017-08-22 15:13:00	

Figura 1. Ejemplo de eventos extraídos para el análisis

Para tener una mayor consistencia en los datos y facilitar el análisis se eliminaron todas las trazas incompletas, ya sea porque no tienen un estado inicial (ingreso) o un estado final (entregado, devuelto, no existe dirección, falleció, o se mudó). Esto puede ocurrir por algún error en la carga de los datos o porque son trazas que todavía no completaron el proceso.

Para quitar las trazas incompletas se utilizó un filtro de reglas heurísticas simples donde se especificó cual es el estado inicial y cuales son los estados finales válidos. El filtro descarta todas las trazas que no cumplan con estos requisitos. Luego de este filtrado se obtiene una muestra de aproximadamente 16.000 trazas con 43.000 eventos.

3.2. Descubrimiento del proceso

Se eligió el algoritmo Alpha para realizar el descubrimiento del proceso. El algoritmo Alpha fue propuesto por van der Aalst, Weijters y Mărușter[7] y es ampliamente utilizado en Minería de Procesos. El algoritmo reconstruye causalidad a partir de una secuencia de eventos y como resultado devuelve una Red de Petri donde se refleja el proceso de negocio utilizado. Cada transición en la red representa una tarea.

Para este caso de estudio en particular se extrae una pequeña muestra representativa que se utilizará para descubrir el proceso, luego se contrastará este proceso descubierto con el resto de las trazas para observar el cumplimiento.

En la Figura 2 se muestra la red de petri descubierta que representa el proceso que deberán cumplir las trazas.

El proceso descubierto se conforma de la siguiente manera: siempre hay un evento de ingreso que deben cumplir todas las trazas luego del cual se realiza la distribución. Si se puede entregar se registra la entrega y finaliza la historia de la traza. Si no se puede entregar por un evento final (persona fallecida, no existe la dirección, no hay suficientes datos para encontrar la dirección, etc) se registra la no entrega con el motivo y finaliza el proceso. Si el motivo de no entrega no es

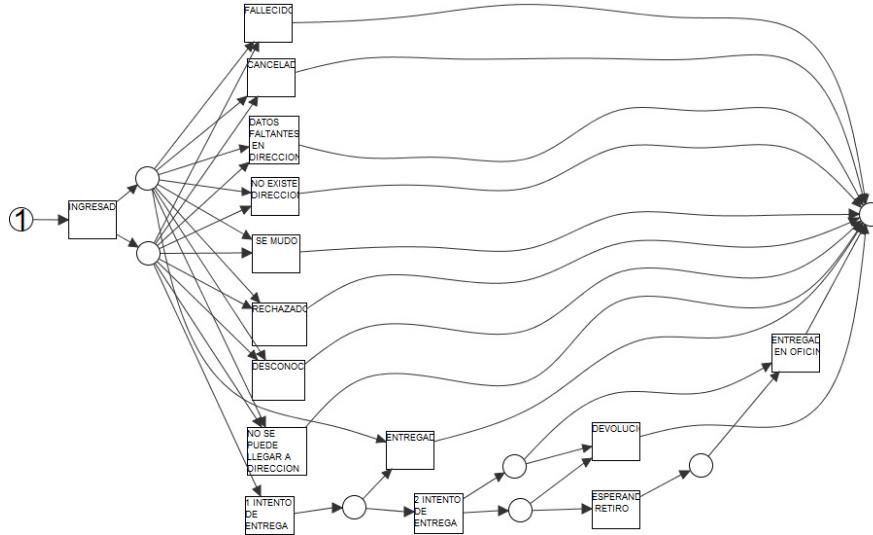


Figura 2. Proceso descubierto que deberán cumplir las trazas

un evento final (por ejemplo no había nadie para entregar) se registra un primer intento de entrega y se intenta nuevamente al día siguiente. Si al día siguiente tampoco se puede entregar se mantiene el envío en la oficina durante un tiempo esperando que el destinatario se presente a retirar (se deja un aviso en cada visita). Finalmente se registra el evento de entrega en la oficina o se devuelve el envío al remitente. La Figura 2 ilustra los escenarios que componen el proceso descubierto.

3.3. Verificación del modelo

Para llevar a cabo la verificación de conformidad se toman todas las trazas de la muestra (en total 16.811 trazas con 43.888 eventos) y se contrastan con el modelo descubierto.

Se busca determinar cuánto se ajustan las trazas al modelo y analizar aquellas que más se desvían teniendo en cuenta qué pasos se cumplen, cuáles no y si hay pasos que no están reflejados en el proceso descubierto. El resultado se muestra en la Figura 3. Allí se muestran los eventos más comunes en color oscuro y los caminos más frecuentes con una línea más gruesa. Se puede ver que la mayoría de las trazas finalizan con una entrega, ya sea en el primer intento o en el segundo.

Analizando las estadísticas de la muestra se observa que en un 80 % de los casos la pieza es entregada, ya sea en el primer o segundo intento y que el resto se distribuye uniformemente. Según estas observaciones se puede afirmar que el 85 % de las trazas cumple por completo el proceso descubierto (ya sea entregado o no) con una media de 2.6 eventos en cada traza.

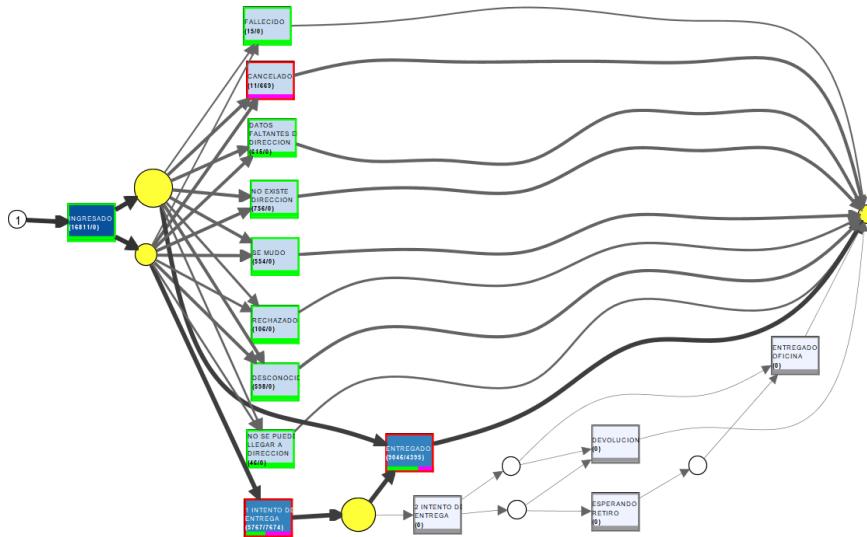


Figura 3. Recorrido de las trazas: eventos más comunes en oscuro y recorridos más frecuentes en línea más gruesa

Se observa también que muchas trazas no verifican el proceso exactamente ya sea porque no cumplen los pasos en el orden correcto o porque se saltean alguno. Estas trazas tienen un ajuste más bajo cuanto más se diferencien del modelo por lo tanto, se analizarán aquellas que tengan un ajuste menor al 50% ya que representan un gran desvío operativo.

Con estos resultados se abren dos caminos diferentes para analizar, por un lado las trazas que tienen demasiados movimientos y por otro aquellas que no realizan las tareas en el orden correspondiente o tienen eventos faltantes.

Adicionalmente se analizarán aquellas trazas que cumplen el proceso correctamente pero en un intervalo de tiempo excesivamente alto.

Se verifica que la cantidad de movimientos por traza sigue una distribución normal con media 2,6 y desvío 0,95, por lo tanto se decide tomar un valor superior a la media mas tres desvíos estándar como un valor representativo de una cantidad excesiva de movimientos.

En base a esto, para el primer caso se procedió a filtrar y exportar las trazas con más de 6 movimientos para realizar un análisis posterior. Se encontraron 108 casos con estas características.

En el segundo caso se analizaron las trazas que no se ajustan exactamente al proceso descubierto. Para ello se estableció el valor de ajuste en 0.5 al considerar que las trazas tienen al menos el 50% de sus movimientos fuera del modelo representando un gran desvío al procedimiento. Como se mencionó previamente esto puede ocurrir porque hay eventos de más en las trazas (no existen en el modelo) o no cumplen correctamente la secuencia.

<p>3760173.0 8 events</p> <p>INGRESADO #1 02.01.2020 16:29:45.000</p> <p>1 INTENTO DE ENTREGA #2 13.01.2020 12:25:00.000</p> <p>2 INTENTO DE ENTREGA #3 14.01.2020 11:20:00.000</p> <p>2 INTENTO DE ENTREGA #4 15.01.2020 11:20:00.000</p> <p>1 INTENTO DE ENTREGA #5 16.01.2020 11:20:00.000</p> <p>2 INTENTO DE ENTREGA #6 17.01.2020 09:39:00.000</p> <p>2 INTENTO DE ENTREGA #7 20.01.2020 10:10:00.000</p> <p>DEVOLUCION #8 21.01.2020 10:10:00.000</p>	<p>3780088.0 6 events</p> <p>INGRESADO #1 20.01.2020 21:00:14.000</p> <p>1 INTENTO DE ENTREGA #2 21.01.2020 11:12:00.000</p> <p>1 INTENTO DE ENTREGA #3 22.01.2020 11:16:00.000</p> <p>ESPERANDO RETIRO #4 22.01.2020 16:04:31.000</p> <p>DEVOLUCION #5 29.01.2020 15:02:07.000</p> <p>ENTREGADO #6 31.01.2020 11:15:00.000</p>
---	--

Figura 4. Eventos repetidos e inconsistencias

En este caso se observó que el mismo evento se repite en sucesivos días o existen registros inconsistentes La Figura 4 muestra estas situaciones. En la parte izquierda se observa un caso que aparece una primera visita luego de la segunda; esto no puede ocurrir y seguramente es debido a un error en la carga (ejemplo se cargó mal la fecha en uno de los movimientos). Del lado derecho, en la misma figura, se muestra una inconsistencia ya que aparece registrado un movimiento de entrega luego de que el producto fue devuelto.

Adicionalmente se utilizó la herramienta Inductive Visual Miner para generar una animación visual y representar de forma más amena estos casos. Para ello se utilizó el conjunto de trazas que tiene un ajuste menor al 50 % obteniendo la visualización de las trazas que más se desvían. La Figura 5 ilustra esta animación. En dicha figura cada traza se representa con un círculo que recorre las diferentes etapas del proceso. Esta representación visual muestra que unas trazas tardan mucho más que otras. También muestra que en ocasiones se presentan algunos comportamientos erróneos. Por ejemplo en la figura 5 se observan retrocesos en las trazas, es decir luego de una tarea se ejecuta otra que debería ser anterior; se resaltan con círculos que 25 trazas van del segundo intento al primero y 20 vuelven sobre sí mismas.

Esta exportación de trazas que no se ajustan al modelo muestran los motivos que ocasionan los desvíos y permiten hacer un análisis más detallado con el fin de encontrar mejoras al proceso.



Figura 5. Trazas que no cumplen con el modelo, en círculo la cantidad que retroceden en de un estado en lugar de avanzar (ver sentido de la flecha)

3.4. Análisis de Performance

Finalmente se procede a realizar un análisis de performance sobre las trazas del modelo en busca de aquellas que, si bien pueden haber completado el proceso correctamente, el mismo se realizó en un intervalo de tiempo extremadamente superior a la media de resolución de cada traza. Para ello se vuelve a reproducir la muestra de datos sobre la red de Petri pero en este caso con una herramienta orientada al análisis de performance que se enfoca en los tiempos de resolución y no tanto en ver si las trazas se ajustan o no al proceso.

Como resultado se obtiene una visualización que representa en tonos más oscuros las tareas que demoran más tiempo en realizarse.

Se observa un tiempo medio de resolución para cada traza de 3,75 días, con un desvío de 2,81 días. Utilizando el mismo criterio que para la cantidad de movimientos por traza se decide enfocar el análisis en aquellos casos que tengan un tiempo de finalización mayor a la media más dos desvíos (9,37 días) por considerar que un tiempo de resolución mayor a este valor se trata de un desvío operativo.

Se realiza la exportación de las trazas que cumplen con ese criterio y se obtiene un total de 687 casos.

Observando los casos filtrados se encuentran trazas que se ajustan perfectamente al proceso pero con tiempos extremadamente altos. En algunos casos las trazas tienen más de tres movimientos y un intervalo de hasta 3 días entre cada uno de ellos lo que justifica un tiempo total tan elevado. Si bien 3 días entre movimientos es un tiempo mayor a la media, podría justificarse con algunas situaciones que pueden ocurrir en el día a día, como por ejemplo la falta de personal debido a enfermedad o vacaciones. Por otro lado, la mayoría de los casos que se obtuvieron en esta muestra son de solamente dos movimientos, donde el envío ingresó y estuvo detenido, en algunos casos hasta 14 días antes de tener un evento de salida a distribución. Luego el proceso se cumple correctamente con una entrega o devolución. Estos casos representan un desvío operativo importante ya que impactan directamente en la calidad del servicio a los clientes incumpliendo los estándares de entrega del producto.

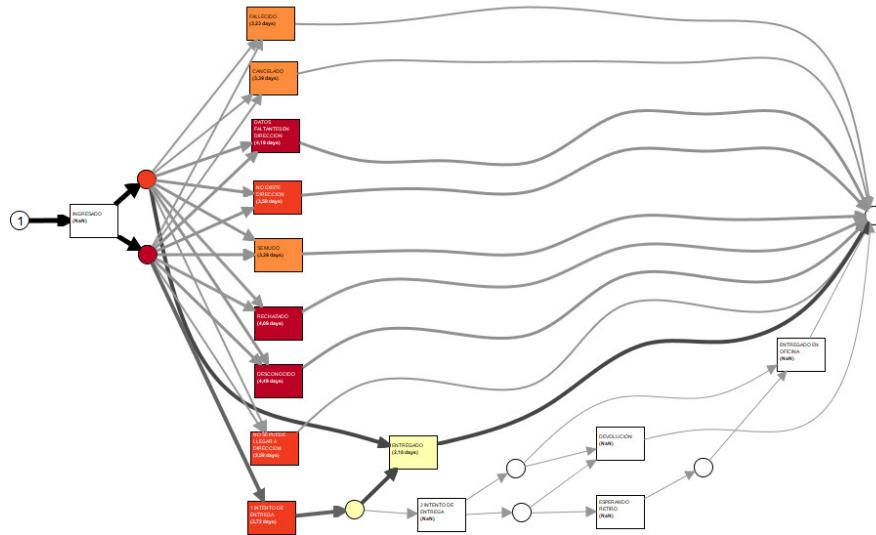


Figura 6. Análisis de Performance de las trazas de la muestra

En la figura 7 se observan trazas que cumplen el proceso con solo dos movimientos pero con una diferencia de más de 10 días entre uno y otro.

Con el análisis de performance se descubrieron nuevos desvíos operativos que no aparecían en el análisis anterior ya que se trata de trazas que se ajustan perfectamente al modelo pero que están con los tiempos muy por encima de la media, llegando en casos extremos a 24 días para una entrega. En base a este estudio se decidió tomar una muestra de los 10 casos más extensos y consultar a los responsables de cada sector / sucursal al que pertenezca para obtener información acerca del motivo de este retraso.

Como resultado del análisis de performance se encuentra la necesidad de generar una alarma que permita identificar estos casos luego de un intervalo de tiempo determinado para mejorar la calidad del servicio a los clientes.

4. Conclusión y futuras líneas de investigación

En este trabajo se utilizaron diferentes técnicas de Minería de Procesos para analizar datos reales de distribución postal en la República Argentina durante los años 2017 a 2020. No se encontraron a la fecha de generación de este documento investigaciones similares que analicen este mismo caso de estudio.

Luego de generar el modelo a partir de una muestra representativa de trazas se descubrió el proceso que realmente se lleva a cabo. Posteriormente se contrastó este modelo con la totalidad de trazas. Por su parte, la verificación de conformidad permitió identificar situaciones fuera de lo común que se traduje-

3765390.0 2 events	3769366.0 2 events	3763891.0 2 events	3763937.0 2 events
INGRESADO #1 07.01.2020 10:58:28.000	INGRESADO #1 10.01.2020 16:30:40.000	INGRESADO #1 03.01.2020 20:23:55.000	INGRESADO #1 03.01.2020 20:40:43.000
ENTREGADO #2 22.01.2020 09:55:00.000	ENTREGADO #2 28.01.2020 10:32:00.000	ENTREGADO #2 17.01.2020 13:00:00.000	DESCONOCIDO #2 16.01.2020 16:37:00.000
3764216.0 2 events	3769606.0 2 events	3767176.0 2 events	3775013.0 2 events
INGRESADO #1 03.01.2020 20:45:15.000	INGRESADO #1 06.01.2020 20:10:58.000	INGRESADO #1 07.01.2020 15:56:15.000	INGRESADO #1 15.01.2020 19:53:23.000
ENTREGADO #2 23.01.2020 13:20:00.000	ENTREGADO #2 21.01.2020 10:31:00.000	ENTREGADO #2 23.01.2020 12:01:00.000	SE MUDO #2 27.01.2020 12:30:00.000

Figura 7. Trazas que cumplen el proceso pero con tiempos muy altos

ron en errores operativos. Se identificaron tanto los casos que no cumplieron con el modelo como también aquellos que lo hicieron con redundancia de tareas.

Con el análisis de performance se descubrieron trazas que tienen desvíos operativos debido al excesivo tiempo de finalización pero que se ajustan correctamente al proceso. Estos casos tienen un gran impacto en la calidad del servicio.

Como futura línea de trabajo se propone continuar con las técnicas de minería de procesos y encontrar la forma de insertar en el sistema alertas tempranas que eviten los desvíos y cuellos de botella operativos.

Referencias

1. Process Mining Applied to Postal Distribution, Victor Martinez; Laura Lanzarini; Franco Ronchetti, CACIC 2021, 2021, <http://sedici.unlp.edu.ar/handle/10915/130342>
2. Process Mining: Data Science in Action, Wil van der Aalst, 978-3-662-49850-7, 2016, Springer
3. Research of Postal Data mining system based on big data, Xia Hu1; Yanfeng Jin1; Fan Wang, 3rd International Conference on Mechatronics, Robotics and Automation, 2015, 10.2991/icmra-15.2015.124, https://www.researchgate.net/publication/300483008_Research_of_Postal_Data_mining_system_based_on_big_data
4. Context Aware Process Mining in Logistics, Mitchell M. Tseng; Hung-Yin Tsai; Yue Wang, 2017, The 50th CIRP Conference on Manufacturing Systems, <https://www.sciencedirect.com/science/article/pii/S2212827117303311>
5. PM2: a Process Mining Project Methodology, Maikel L. van Eck; Xixi Lu; Sander J.J. Leemans; Wil M.P. van der Aalst, Eindhoven University of Technology, The Netherlands, http://www.processmining.org/_media/blogs/pub2015/pm2_processminingprojectmethodology.pdf
6. Wil van der Aalst, The Process Mining Manifesto by the IEEE Task Force, 2012, <https://www.tf-pm.org/resources/manifesto>

7. Workflow mining; discovering process models from event logs, W. van der Aalst; T. Weijters; L. Maruster, 1041-4347, 2004, IEEE, <https://ieeexplore.ieee.org/document/1316839>
8. Process mining in flexible environments, Christian Walter Gunther,978-90-386-1964-4,2009,Technische Universiteit Eindhoven, <https://research.tue.nl/en/publications/process-mining-in-flexible-environments>
9. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams, IEEE Std 1849-2016, 2016, DOI 10.1109/IEEEESTD.2016.7740858
10. inductive visual miner, Sander J.J. Leemans, 2017, <http://leemans.ch/leemansCH/publications/ivm.pdf>
11. Reinventing the Postal Sector in an Electronic Age, Michael A. Crew; Paul R. Kleindorfer,978-1849803601,2011,Edward Elgar Publishing

ANEXO B

Código Fuente

Este anexo recopila algunas partes del código fuente de las principales funcionalidades de cada uno de los módulos desarrollados.

El código fuente de todos los módulos realizados para el presente trabajo se encuentra en **GitHub** en el repositorio: <https://github.com/vimartinez/UNLP-Tesis>

B.1. MÓDULO EXTRACTOR DE DATOS

El módulo extractor de datos se conecta a la base de datos Oracle para extraer los datos de entrada. La ejecución del extractor de datos debe programarse en el servidor para ejecutarse en un momento específico. El Listado B.1 ejemplifica cómo programar la ejecución del extractor para que se ejecute todos los domingos a las 13:00 hs.

Listado B.1: Programar extracción de datos de Oracle

```
1 #Comando para programar ejecución
2 crontab -e
3
4 # Proceso de esxtracción de datos de Oracle: Todos los domingos a las 13:00 hs
5 0 13 * * 0 /usr/local/bin/extractorDatosOracle.sh
```

La conexión a la base, para efectuar la extracción de datos, se realizó con iSQLPlus por línea de comandos. iSQLPlus es una herramienta que viene con la instalación de Oracle. El código correspondiente a este proceso puede verse en el Listado B.2.

Listado B.2: Código fuente del proceso de extracción de datos de Oracle

```
1#!/bin/bash
2
3# Configuración de la conexión a la base de datos Oracle
4ORACLE_USER="userName"
5ORACLE_PASSWORD="userPassword"
6ORACLE_HOST="192.168.0.54"
7ORACLE_PORT="5001"
8ORACLE_SID="bdProd"
9
10# Consulta SQL a ejecutar
11SQL_QUERY="SELECT env_id AS case_id ,eve_descrip AS activity ,←
12    ↪ eve_fecha AS timestamp ,eve_id AS activityid
13    FROM envios
14    WHERE informado IS NULL
15    ORDER BY env_id"
16
17# Archivo de salida CSV
18OUTPUT_FILE="TrazasEntrada.csv"
```

```

18
19 # Conectarse a la base de datos Oracle y ejecutar la consulta
20 sqlplus -s ${ORACLE_USER}/${ORACLE_PASSWORD}@${ORACLE_HOST}: ↵
    ↪ ${ORACLE_PORT}/${ORACLE_SID} << EOF
21 set pagesize 0
22 set colsep ,
23 set feedback off
24 set echo off
25 set heading off
26 spool ${OUTPUT_FILE}
27 ${SQL_QUERY};
28 spool off
29 exit
30 EOF

```

B.1.1. Módulo de Minería de Procesos

La aplicación de Minería de Procesos se ejecutará de forma automática luego del proceso de extracción de datos. El comando para programar la ejecución del módulo de Minería de Procesos se muestra en el listado B.3

Listado B.3: Script de ejecución del módulo de Minería de Procesos

```

1#!/bin/bash
2
3# Módulo de Minería de Procesos
4PYTHON_SCRIPT="/usr/local/bin/ProcessMiningPostal.py"
5
6# Comando para ejecutar el programa de Python
7python3 "$PYTHON_SCRIPT"

```

La ejecución del componente de Minería de Procesos estará programada para ejecutarse luego de la extracción de datos de Oracle. El listado B.4 muestra un ejemplo de programación para ejecutarse todos los domingos a las 16:00 hs, de esta manera cuando el operador ingrese el lunes a trabajar tendrá el aviso de incidencias encontradas en su correo electrónico.

Listado B.4: Programar ejecución del módulo de Minería de Procesos

```

1 #Comando para programar ejecución
2 crontab -e
3
4 # Proceso de esxtracción de datos de Oracle: Todos los domingos a las 16:00 hs
5 0 16 * * 0 /usr/local/bin/jobMineriaProcesos.sh

```

El código del componente de Minería de Procesos se muestra en el listado B.5

Listado B.5: Componente de Minería de Procesos en distribución Postal

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Apr 26 22:38:59 2023
4
5 @author: Vic
6 """
7
8 import pm4py
9 import pandas as pd
10 import sqlalchemy
11 import smtplib
12 import shutil
13 import os

```

```
14 import shutil
15 from datetime import datetime
16
17 POSTAL_PROCESS_PETRI_NET = "PostalProcessAlpha.pnml"
18 INPUT_LOG_FILE = "TrazasEntrada-01.csv"
19 DB_HOST = "localhost"
20 DB_NAME = "UNLP-Tesis"
21 DB_USER = "postgres"
22 DB_PASSWORD = "1234"
23 EXPORT_FILES = False
24
25 def mine_postal_data():
26
27     if os.path.exists(INPUT_LOG_FILE):
28
29         #importar csv
30         event_log = pd.read_csv(INPUT_LOG_FILE, sep=',')
31         event_log = event_log.applymap(str)
32         event_log = pm4py.format_dataframe(event_log, ↵
33                                         ↵ case_id='case_id', activity_key='activity', ↵
34                                         ↵ timestamp_key='timestamp')
35
36         #importar proceso
37         petri_net, initial_marking, final_marking = ↵
38             ↵ pm4py.read_pnml(POSTAL_PROCESS_PETRI_NET)
39
40         #verificación de conformidad
41         conformance_token_based_replay = ↵
42             ↵ pm4py.conformance_diagnostics_token_based_replay ↵
43             ↵ (event_log, petri_net, initial_marking, ↵
44                 ↵ final_marking, activity_key='activity', ↵
45                 ↵ case_id_key='case_id', ↵
46                 ↵ timestamp_key='time:timestamp')
47         data_frame = pd.DataFrame(conformance_token_based_replay)
48         event_log_traces = ↵
49             ↵ event_log.drop_duplicates(subset=['case_id'])
50         event_log_traces.drop('activityid', inplace=True, axis=1)
51         event_log_traces.drop('activity', inplace=True, axis=1)
52         merged_df = pd.merge(event_log_traces, data_frame, ↵
53             ↵ left_index=True, right_index=True, how='inner')
54
55         #Remover columnas adicionales
56         merged_df.drop('case:concept:name', inplace=True, axis=1)
57         merged_df.drop('concept:name', inplace=True, axis=1)
58         merged_df.drop('time:timestamp', inplace=True, axis=1)
59         merged_df.drop('activated_transitions', inplace=True, ↵
60                         ↵ axis=1)
61         merged_df.drop('reached_marking', inplace=True, axis=1)
62         merged_df.drop('enabled_transitions_in_marking', ↵
63                         ↵ inplace=True, axis=1)
64         merged_df.drop('transitions_with_problems', ↵
65                         ↵ inplace=True, axis=1)
66         merged_df.drop('missing_tokens', inplace=True, axis=1)
67         merged_df.drop('consumed_tokens', inplace=True, axis=1)
68         merged_df.drop('remaining_tokens', inplace=True, axis=1)
69         merged_df.drop('produced_tokens', inplace=True, axis=1)
70         merged_df.drop('trace_is_fit', inplace=True, axis=1)
71
72         #Filtrado por ajuste menor a 0.5
73         traces_with_low_alignemnt = ↵
74             ↵ merged_df.query("trace_fitness < 0.5")
75
76         #Filtrado por tiempo excesivo de resolución
```

```

63     filtered_traces_performance = ↪
64         pm4py.filter_case_performance(event_log, ↪
65             800000.0, 19000000.0, ↪
66             timestamp_key='time:timestamp', ↪
67             case_id_key='case:concept:name')
68
68 #Remover columnas adicionales
69 traces_with_excessive_time_detail = ↪
70     pd.DataFrame(filtered_traces_performance)
71 traces_with_excessive_time_detail.drop('case:concept:name', ↪
72     inplace=True, axis=1)
73 traces_with_excessive_time_detail.drop('concept:name', ↪
74     inplace=True, axis=1)
75 traces_with_excessive_time_detail.drop('time:timestamp', ↪
76     inplace=True, axis=1)
77 traces_with_excessive_time = ↪
78     traces_with_excessive_time_detail.drop_duplicates(↪
79         subset=['case_id'])
80 traces_with_excessive_time.drop('activityid', ↪
81     inplace=True, axis=1)
82 traces_with_excessive_time.drop('activity', ↪
83     inplace=True, axis=1)
84
85 event_log.drop('case:concept:name', inplace=True, axis=1)
86 event_log.drop('concept:name', inplace=True, axis=1)
87 event_log.drop('time:timestamp', inplace=True, axis=1)
88
89 #Export a BD Postgre
90 engine = ↪
91     sqlalchemy.create_engine("postgresql://" + DB_USER + ":" ↪
92         + DB_PASSWORD + "@" + DB_HOST + "/" + DB_NAME)
93 event_log.to_sql('trazas', engine, if_exists='append', ↪
94     index=False)
95 traces_with_low_alignemnt.to_sql('trazasajustebajo', ↪
96     engine, if_exists='append', index=False)
97 traces_with_excessive_time.to_sql('trazasexcesotiempo', ↪
98     engine, if_exists='append', index=False)
99 send_notification()
100 move_file_proceced()
101
102 if EXPORT_FILES == True:
103
104     OUTPUT_LOG_FILE = "TrazasConAjusteBajoAPP.csv"
105     OUTPUT_LOG_FILE2 = "TrazasConBajaPerformanceAPP2.csv"
106     OUTPUT_LOG_FILE_XES = "TrazasConAjusteBajoApp.xes"
107     OUTPUT_LOG_FILE_XES2 = ↪
108         "TrazasConBajaPerformanceAPP.xes"
109
110     #Expot a csv las trazas con ajuste bajo
111     data_frame = pd.DataFrame(traces_with_low_alignemnt)
112     data_frame.to_csv(OUTPUT_LOG_FILE, index=False, ↪
113         header=True)
114     #Export a csv las trazas con más de diez días
115     traces_with_excessive_time_detail.to_csv(↪
116         (OUTPUT_LOG_FILE2, index=False, header=True))
117     #Export a xes las trazas con ajuste bajo
118     pm4py.write_xes(data_frame, OUTPUT_LOG_FILE_XES)
119     #Export a xes las trazas con más de diez días
120     pm4py.write_xes(filtered_traces_performance, ↪
121         OUTPUT_LOG_FILE_XES2)
122
123 else:
124     print("No hay archivos de trazas para procesar")

```

```

105
106
107 def send_notification():
108     remitente = "Notificaciones<notificaciones@mail.com.ar>"
109     destinatario = "Responsable de envíos<
110         ↪ <responsable@mail.com.ar>""
111     asunto = "Posibles desvíos operativos en envíos de trazas"
112     mensaje = """Atención:<br/> <br/>
113     Se encontraron posibles desvíos operativos en el ↪
114         ↪ procesamiento de varios envíos. <br>
115     Ingresar en la aplicación de seguimiento para su gestión. ↪
116         ↪ <br><br>
117     Este mensaje se envió de forma automática, no responder al ↪
118         ↪ mismo.
119 """
120
121
122     email = """From: %s
123     To: %s
124     MIME-Version: 1.0
125     Content-type: text/html
126     Subject: %s
127
128     %s
129 """ % (remitente, destinatario, asunto, mensaje)
130     try:
131         smtp = smtplib.SMTP('localhost')
132         smtp.sendmail(remitente, destinatario, email)
133         print ("Correo enviado")
134     except:
135         print ("Error: no se pudo enviar el mensaje.")
136
137 def move_file_proceded():
138     fecha_actual = datetime.now().strftime("%Y-%m-%d")
139     nuevo_nombre = ↪
140         ↪ f"{INPUT_LOG_FILE.split('.')[0]}_{fecha_actual}.csv"
141     ruta_procesados = "procesados"
142     os.rename(INPUT_LOG_FILE, nuevo_nombre)
143     shutil.move(nuevo_nombre, os.path.join(ruta_procesados, ↪
144         ↪ nuevo_nombre))
145
146 if __name__ == "__main__":
147     mine_postal_data()

```

B.2. BASE DE DATOS

El listado B.6 muestra el código en SQL utilizado para generar la tabla *trazasajustebajo* donde se almacenan las trazas que tengan un ajuste menor a 0,5 con respecto al modelo.

Listado B.6: Código en SQL par generar la tabla *trazasajustebajo*

```

1 CREATE TABLE public.trazasajustebajo (
2     case_id text NOT NULL,
3     "timestamp" timestamptz NULL,
4     "@@index" int8 NULL,
5     "@@case_index" int8 NULL,
6     trace_fitness float8 NULL,
7     resuelta bool NULL DEFAULT false,
8     CONSTRAINT trazasajustebajo_pk PRIMARY KEY (case_id),
9     CONSTRAINT trazasajustebajo_fk FOREIGN KEY (case_id) ↪
10        REFERENCES public.trazas(case_id)
11 );

```

El listado B.7 muestra el código en SQL utilizado para generar la tabla *trazas* donde se almacenan el listado de trazas a analizar.

Listado B.7: Código en SQL par generar la tabla trazas

```

1  -- public.trazas definition
2
3  -- Drop table
4
5  -- DROP TABLE public.trazas;
6
7  CREATE TABLE public.trazas (
8      case_id text NOT NULL,
9      activity text NULL,
10     "timestamp" timestamptz NULL,
11     activityid text NULL,
12     "@@index" int8 NULL,
13     "@@case_index" int8 NULL,
14     id int4 NOT NULL GENERATED ALWAYS AS IDENTITY
15 );

```

B.3. API DE GESTIÓN DE INCIDENCIAS

La API de gestión de incidencias se desarrolló en Java con Spring utilizando una arquitectura de tres capas. A continuación se muestra el código de un Controlador, Servicio y Repositorio.

El listado B.8 muestra el código del controlador IncidenciasABController.

Listado B.8: Código del Controlador de Incidencias

```

1  package ar.edu.unlp.tesis.processminingpostal2.controller;
2
3  import java.util.*;
4
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.data.domain.Page;
7  import org.springframework.data.domain.PageRequest;
8  import org.springframework.data.domain.Pageable;
9  import org.springframework.web.bind.annotation.*;
10
11 import ←
12   ↪ ar.edu.unlp.tesis.processminingpostal2.model.TrazaAjusteBajo;
13 import ar.edu.unlp.tesis.processminingpostal2.service.
14 IncidenciasABService;
15 import io.swagger.v3.oas.annotations.Operation;
16 import io.swagger.v3.oas.annotations.Parameter;
17 import io.swagger.v3.oas.annotations.responses.ApiResponse;
18 import io.swagger.v3.oas.annotations.responses.ApiResponses;
19
20 @RestController
21 @RequestMapping("/incidencias")
22 @CrossOrigin
23 public class IncidenciasABController {
24
25     @Autowired
26     IncidenciasABService incidenciasABService;
27
28     @GetMapping(value="/ajustebajo/")
29     @Operation(summary="Devuelve un listado con todas las ←
30       ↪ incidencias registradas con ajuste bajo", ←
31       ↪ description="No tiene parámetros de entrada, se puede ←
32       ↪ enviar una página y registros por página", tags = ←
33       ↪ {"Incidencias Ajuste Bajo"})

```

```

29     @ApiResponses(value= {
30         @ApiResponse(responseCode="200", description="Se ↵
31             ↵ encontraron ↵
32             ↵ incidencias"),
33         @ApiResponse(responseCode="404", description="No ↵
34             ↵ se ↵
35             ↵ encontraron ↵
36             ↵ en la BD")
37     })
38     public @ResponseBody Map<String, Object> ↵
39         ↵ getAllIncidenciasAjusteBajo( ↵
40             ↵ @RequestParam(defaultValue = "0") int page, ↵
41             ↵ @RequestParam(defaultValue = "5") int size) {
42
43         List<TrazaAjusteBajo> trazaAjusteBajos = new ↵
44             ↵ ArrayList<TrazaAjusteBajo>();
45         Pageable paging = PageRequest.of(page, size);
46         Page<TrazaAjusteBajo> pageIncidencia;
47         pageIncidencia = ↵
48             ↵ incidenciasABService.getAllIncidenciasAjusteBajo ↵
49             ↵ (paging);
50         trazaAjusteBajos = pageIncidencia.getContent();
51         Map<String, Object> response = new HashMap<>();
52         response.put("incidencias", trazaAjusteBajos);
53         response.put("currentPage", pageIncidencia.getNumber());
54         response.put("totalItems", ↵
55             ↵ pageIncidencia.getTotalElements());
56         response.put("totalPages", pageIncidencia.getTotalPages());
57
58         return response;
59     }
60
61     @PostMapping(value="/ajustebajo/", produces= ↵
62         ↵ "application/json")
63     @Operation(summary="Dar de alta una nueva incidencia", ↵
64         ↵ description="Permite agregar una incidencia ↵
65             ↵ manualmente", tags = {"IncidenciasAjusteBajo"})
66     public @ResponseBody void ↵
67         ↵ addIncidenciaAjusteBajo(TrazaAjusteBajo trazaAjusteBajo){
68         incidenciasABService.addIncidenciaAjusteBajo(trazaAjusteBajo);
69     }
70
71     @GetMapping(value="/ajustebajo/{idIncidencia}")
72     @Operation(summary="Obtener una incidencia de la base", ↵
73         ↵ description="Se debe enviar el id de la incidencia", ↵
74             ↵ tags = {"IncidenciasAjusteBajo"})
75     public @ResponseBody Optional<TrazaAjusteBajo> ↵
76         ↵ getIncidenciaAjusteBajoById(@Parameter(description="id de la incidencia") @PathVariable("idIncidencia") Long id){
77         return ↵
78             ↵ incidenciasABService.getIncidenciaAjusteBajoById(id);
79     }
80
81     @DeleteMapping(path="/ajustebajo/{id}", ↵
82         ↵ produces="application/json")
83     @Operation(summary="Borrar una incidencia de la base", ↵
84         ↵ description="Se debe enviar el id de la incidencia", ↵
85             ↵ tags = {"IncidenciasAjusteBajo"})
86     public @ResponseBody void ↵
87         ↵ delIncidenciaAjusteBajo(@Parameter(description="id de la incidencia") Long id) {
88         incidenciasABService.delIncidenciaAjusteBajo(id);
89     }

```

```

68
69     @PutMapping(path="/ajustebajo/", produces="application/json")
70     @Operation(summary="Modificar una incidencia una incidencia ←
71         → de la base", description="Se debe enviar el id de la ←
72         → incidencia", tags = {"IncidenciasAjusteBajo"})
73     public @ResponseBody void ←
74         → UpdIncidenciaAjusteBajo(TrazaAjusteBajo ←
75             → trazaAjusteBajo) {
76         incidenciasABService.updIncidenciaAjusteBajo(trazaAjusteBajo);
77     }
78
79
80 }
81 }
```

El listado B.9 muestra el código del servicio IncidenciasABService

Listado B.9: Código del Servicio de Incidencias

```

1 package ar.edu.unlp.tesis.processminingpostal2.service;
2
3 import java.util.Optional;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.data.domain.Page;
7 import org.springframework.data.domain.Pageable;
8 import org.springframework.stereotype.Service;
9
10 import ←
11     → ar.edu.unlp.tesis.processminingpostal2.model.TrazaAjusteBajo;
12 import ar.edu.unlp.tesis.processminingpostal2.repository.
13 IncidenciasABRepository;
14
15 @Service
16 public class IncidenciasABService {
17
18     @Autowired
19     IncidenciasABRepository incidenciasABRepository;
20
21     public TrazaAjusteBajo ←
22         → addIncidenciaAjusteBajo(TrazaAjusteBajo ←
23             → trazaAjusteBajo) {
24         return incidenciasABRepository.save(trazaAjusteBajo);
25     }
26
27
28     public Optional<TrazaAjusteBajo> ←
29         → getIncidenciaAjusteBajoById(Long id) {
30         return incidenciasABRepository.findById(id);
31     }
32
33     public void delIncidenciaAjusteBajo(Long id) {
34         incidenciasABRepository.deleteById(id);
35     }
36
37     public void updIncidenciaAjusteBajo(TrazaAjusteBajo ←
38         → trazaAjusteBajo) {
39         incidenciasABRepository.save(trazaAjusteBajo);
40     }
41 }
```

```

35
36     public Page<TrazaAjusteBajo> ↵
37         → getAllIncidenciasAjusteBajo(Pageable paging) {
38             return incidenciasABRepository.findAll(paging);
39         }
40
41     public Long count() {
42         return ↵
43             → incidenciasABRepository.countIncidenciasSinResolver();
44     }

```

El listado B.10 muestra el código del repositorio IncidenciasABRepository

Listado B.10: Código del Repositorio de Incidencias

```

1 package ar.edu.unlp.tesis.processminingpostal2.repository;
2
3 import org.springframework.data.domain.Page;
4 import org.springframework.data.domain.Pageable;
5 import org.springframework.data.jpa.repository.JpaRepository;
6
7 import ↵
8     → ar.edu.unlp.tesis.processminingpostal2.model.TrazaAjusteBajo;
9 import org.springframework.data.jpa.repository.Query;
10
11
12 public interface IncidenciasABRepository extends ↵
13     → JpaRepository<TrazaAjusteBajo, Long>{
14
15     Page<TrazaAjusteBajo> findAll(Pageable pageable);
16
17     @Query(nativeQuery=true, value = "SELECT COUNT(*) FROM ↵
18         → trazasajustebajo WHERE resuelta=false")
19     Long countIncidenciasSinResolver();
20 }

```

B.4. WEB DE ADMINISTRACIÓN

La web de administración se desarrolló en React. A continuación se muestra el código fuente del componente de inicio y de un componente que consume el listado de incidencias.

Componente de pantalla de inicio de web de administración B.12

Listado B.11: Componente de inicio en Web de Administración

```

1 import React,{useState, useEffect} from "react";
2 import { Link } from 'react-router-dom';
3
4 const Hero = (data) => {
5
6     const [IncidenciasExcesoTiempo, setIncidencias] = useState([]);
7     useEffect(() => {
8         fetch('http://localhost:8080/incidencias/excesotiempo/count')
9             .then((response) => response.json())
10            .then((data) => {
11                console.log(data);
12                setIncidencias(data);
13            })
14            .catch((err) => {

```

```

15         console.log(err.message)
16     });
17 }, []);
18
19 const [IncidenciasAjusteBajo, setIncidencias2] = useState([]);
20 useEffect(() => {
21     fetch('http://localhost:8080/incidencias/ajustebajo/count')
22     .then((response) => response.json())
23     .then((data) => {
24         console.log(data);
25         setIncidencias2(data);
26     })
27     .catch((err) => {
28         console.log(err.message)
29     });
30 }, []);
31
32 return (
33 <div>
34     <div className="container-my-1">
35         <div className="p-5 text-center bg-body-tertiary rounded-3">
36             <h1 className="text-body-emphasis">{data.name}</h1>
37             <p className="lead">
38                 Aplicación para gestión de las incidencias encontradas ←
39                     ← por el módulo de Minería de Procesos.
40             </p>
41             <div className="row row-cols-1 row-cols-sm-2" ←
42                 ← row-cols-md-2 ug-3">
43                 <div className="col">
44                     <div className="card shadow-sm">
45                         <svg className="bd-placeholder-img card-img-top" ←
46                             ← width="100%" height="225" ←
47                             ← xmlns="http://www.w3.org/2000/svg" role="img" ←
48                             ← aria-label="Placeholder:Thumbnail" ←
49                             ← preserveAspectRatio="xMidYMid slice" ←
50                             ← focusable="false">
51                             <title>Placeholder</title><rect width="100%" ←
52                                 ← height="100%" fill="#6ea8fe"></rect>
53                             <text x="45%" y="50%" font-size="55" ←
54                                 ← fill="#eceeef" dy=".3em"> ←
55                                 {IncidenciasAjusteBajo}</text>
56                         </svg>
57                         <div className="card-body">
58                             <p className="card-text">Cantidad de ←
59                                 ← incidencias con ajuste bajo sin resolver.</p>
60                             <div className="d-flex justify-content-between" ←
61                                 ← align-items-center">
62                                 <div className="btn-group">
63                                     <button type="button" className="btn btn-sm" ←
64                                         ← btn-secondary"><Link ←
65                                         ← to="/excesomovimientos" ←
66                                         ← className="nav-link">Ver ←
67                                         ← Detalle</Link></button>
68                                 </div>
69                                 <small className="text-body-secondary" > ←
70                                     ← Incidencias </small>
71                             </div>
72                         </div>
73                     </div>
74                 </div>
75             <div className="col">
76                 <div className="card shadow-sm">
77                     <svg className="bd-placeholder-img card-img-top" ←

```

```

    ↵ width="100%" height="225" ↵
    ↵ xmlns="http://www.w3.org/2000/svg" role="img" ↵
    ↵ aria-label="Placeholder:Thumbnail" ↵
    ↵ preserveAspectRatio="xMidYMidslice" ↵
    ↵ focusable="false">
61      <title>Placeholder</title>
62      <rect width="100%" height="100%" ↵
    ↵ fill="#6ea8fe"></rect>
63      <text x="35%" y="50%" font-size="55" ↵
    ↵ fill="#eceeef" ↵
    ↵ dy=".3em">{IncidenciasExcesoTiempo}</text>
64    </svg>
65    <div className="card-body">
66      <p className="card-text">Cantidad de ↵
    ↵ incidencias con exceso de movimientos sin ↵
    ↵ resolver.</p>
67      <div className="d-flex justify-content-between" ↵
    ↵ align-items-center">
68        <div className="btn-group">
69          <button type="button" className="btn btn-sm" ↵
    ↵ btn-secondary"><Link to="/ajustebajo" ↵
    ↵ className="nav-link">Ver ↵
    ↵ Detalle</Link></button>
70        </div>
71        <small className="text-body-secondary"> ↵
    ↵ Incidencias </small>
72      </div>
73    </div>
74  </div>
75  </div>
76  </div>
77  </div>
78 </div>
79 </div>
80 )}

81 export default Hero;

```

Componente que realiza la búsqueda de Incidencias de la API Java B.12

Listado B.12: Componente que consume datos de la API Java

```

1 import React,{useState, useEffect} from "react";
2 import { Link, useNavigate } from 'react-router-dom';
3 import './estilos.css'

4
5 const ApiConsumerAjusteBajo = () => {
6   const navigate = useNavigate();
7   const [Incidencias, setIncidencias] = useState([]);
8   useEffect(() => {
9     fetch('http://localhost:8080/incidencias/
10                           ajustebajo/?page=0&size=5')
11       .then((response) => response.json())
12       .then((data) => {
13         console.log(data);
14         setIncidencias(data.incidencias);
15       })
16       .catch((err) => {
17         console.log(err.message)
18       });
19   }, []);
20
21   return (
22     <div className="container-md">

```

```

23     <h3>INCIDENCIAS CON AJUSTE BAJO</h3>
24     <table className="table table-striped" >
25       <thead>
26         <tr>
27           <th>id</th>
28           <th>Trazo</th>
29           <th>Ajuste</th>
30           <th>Fecha Ingreso</th>
31           <th>Resuelta</th>
32         </tr>
33       </thead>
34       <tbody>
35         {Incidencias.map((Incidencia) =>{
36           return(
37             <tr key={Incidencia.id} >
38               <td><Link ↵
39                 ↵ to={`/detalleincidencia/${Incidencia.case_id}`}
40                 className="nav-link">{Incidencia.id}</Link></td>
41               <td><Link ↵
42                 ↵ to={`/detalleincidencia/${Incidencia.case_id}`}
43                 className="nav-link">{Incidencia.case_id}</Link></td>
44               <td><Link ↵
45                 ↵ to={`/detalleincidencia/${Incidencia.case_id}`}
46                 className="nav-link">{Incidencia.ajuste}</Link></td>
47               <td><Link ↵
48                 ↵ to={`/detalleincidencia/${Incidencia.case_id}`}
49                 className="nav-link">{Incidencia.fechaIngreso}</Link>
50               <td>
51                 <td><Link ↵
52                   ↵ to={`/detalleincidencia/${Incidencia.case_id}`}
53                   className="nav-link">{Incidencia.resuelta ? 'Si' : ↵
54                     ↵ 'No'}

```

```
70
71     export default ApiConsumerAjusteBajo;
```

B.5. CONTENEDORES

Cada uno de los servicios se generó en un contenedor de Docker independiente.

El listado B.13 muestra el código fuente del archivo docker-compose.yml que permite levantar la imagen de cada servicio en un contenedor diferente.

Listado B.13: Archvo docker-compose.yml para levantar los servicios en contenedores

```
1  version: "3.7"
2  name: UNLP-Tesis
3
4  services:
5    grafana:
6      image: grafana/grafana:8.0.6
7      container_name: grafana2
8      restart: unless-stopped
9      volumes:
10        - D:/Practicas/Docker/Graphana:/usr/share/grafana/data
11      ports:
12        - 3000:3000
13      environment:
14        - GF_PLUGIN_ALLOW_LOCAL_MODE=true
15
16    postgres:
17      image: postgres:15.2
18      container_name: postgres
19      ports:
20        - 5432:5432
21      environment:
22        - POSTGRES_PASSWORD=1234
23
24    javaapp:
25      build: ./Java/processminingpostal2
26      ports:
27        - "8080:8080"
28
29    reactapp:
30      build: ./React/unlp-tesis
31      environment:
32        - NODE_ENV=development
33      volumes:
34        - ./:/usr/src/app
35      command: sh -c 'yarn && yarn start'
36      ports:
37        - "8001:8001"
38      tty: true
39
40    prometheus:
41      image: prom/prometheus:v2.28.1
42      container_name: prometheus
43      restart: unless-stopped
44      ports:
45        - 9090:9090
```

ANEXO C

Herramientas utilizadas

Se listan a continuación las técnicas y herramientas de software utilizadas para la investigación y el desarrollo de esta tesis:

ProM Tools: Conjunto de herramientas open source que permite aplicar técnicas de Minería de Procesos en forma de plugins que se pueden concatenar uno detrás de otro. ProM es un set extensible, independiente de la plataforma, desarrollado en Java con una gran comunidad que lo mantiene en constante crecimiento.

<http://promtools.org/doku.php>

RapidMiner: Conjunto de herramientas para análisis de datos que contempla operadores de entrada - salida, pre procesamiento y visualización de resultados

<https://rapidminer.com/>

Disco: (Demo) Software de Minería de Procesos que genera rápidamente mapas visuales del proceso de negocio y permite optimizar la performance, controlar desvíos o explorar variaciones.

<https://fluxicon.com/disco/>

Signavio: (Demo) Herramienta para el análisis y el diseño de procesos de negocio. Permite diseñar, modelar, documentar y simular procesos de negocio.

<https://www.signavio.com/>

Eclipse: Entorno de desarrollo integrado (IDE) que proporciona un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar java y otros lenguajes de programación.

<https://www.eclipse.org/ide/>

IntelliJ Idea: Entorno de desarrollo integrado para lenguaje java y kotlin que permite desarrollar aplicaciones rápidamente y soporta integración con varias herramientas.

<https://www.jetbrains.com/idea/>

Java: Lenguaje de programación orientado a objetos que permite desarrollar aplicaciones independientes de la plataforma, utilizado en la actualidad para desarrollo de aplicaciones de back end.

<https://www.java.com/es/>

NotePad++: Editor gratuito avanzado de archivos de texto y código fuente con soporte a una gran variedad de lenguajes de programación.

<https://notepad-plus-plus.org/>

GitHub: Repositorio Git en la nube que permite almacenar gratuitamente proyectos de desarrollo de software en una gran cantidad de lenguajes de programación, facilitando el versionado y la distribución de los fuentes.

<https://github.com/>

Dbeaver: software cliente de SQL y herramienta de administración de bases de datos. Para las bases de datos relacionales, utiliza la interfaz de programación de aplicaciones JDBC para interactuar con las bases de datos a través de un controlador JDBC.

<https://dbeaver.io/>

Postman: Herramienta de código abierto que permite probar APIs rápidamente y mantener las colecciones de las solicitudes actualizadas, ahorrando los tiempos de respuesta al momento de realizar los test o las llamadas a los servicios.

<https://www.postman.com/>

Draw.io: Herramienta online gratuita que permite realizar todo tipo de diagramas de software y empresariales.

<https://app.diagrams.net/>

React: Biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre.

<https://es.react.dev/>

JavaScript: JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

<https://developer.mozilla.org/es/docs/Web/JavaScript>

HTML: Siglas en inglés de HyperText Markup Language, es un lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido en los sitios web.

<https://www.w3.org/html/>

CSS: Hojas de estilo en cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de etiquetas como HTML

<https://www.w3.org/TR/CSS/>

Bootstrap: Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

<https://getbootstrap.com/>

Anaconda: Distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático. Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos.

<https://www.anaconda.com/download>

Spider: Entorno de desarrollo integrado multiplataforma de código abierto para la programación científica en el lenguaje Python.

<https://www.spyder-ide.org/>

Python: Lenguaje de programación de alto nivel, interpretado, orientado a objetos y multiplataforma cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo.

<https://www.python.org/>

PM4Py: Plataforma de código abierto que implementa Minería de Procesos en Python.

<https://pm4py.fit.fraunhofer.de/>

Jira Software: Software propietario para la gestión de proyectos, seguimiento de errores e incidencias. La herramienta fue desarrollada por la empresa australiana Atlassian.

<https://www.atlassian.com/es/software/jira>

Spring: framework de Java que permite crear aplicaciones web y de microservicios de manera rápida y sencilla, con una configuración mínima.

<https://spring.io/>

JPA (Java Persistence API): API de Java que permite gestionar el acceso a bases de datos de manera sencilla y estandarizada.

<https://spring.io/guides/gs/accessing-data-jpa/>

Swagger: Swagger es un conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful.

<https://swagger.io/tools/swagger-ui/>

Open API: Originalmente conocida como la especificación Swagger, es una especificación para documentar APIs y generar una interfaz legible por máquina para describir, producir, consumir y visualizar servicios web RESTful.

<https://www.openapis.org/>

Postgre SQL: base de datos relacional open-source, escalable y altamente disponible, utilizada ampliamente en la actualidad para proyectos de microservicios.

<https://www.postgresql.org/>

Docker: plataforma de contenedores que permite empaquetar, distribuir y ejecutar aplicaciones de

manera sencilla, reproducible y escalable.

<https://www.docker.com/>

Prometheus: Aplicación de software gratuita que se utiliza para monitorear y alertar eventos. Registra las métricas en una base de datos de series temporales creada con un modelo de extracción HTTP, con consultas flexibles y alertas en tiempo real.

<https://prometheus.io/>

Grafana: herramienta de código abierto que permite realizar monitoreo, visualización de datos y métricas de una aplicación de manera sencilla y personalizada.

<https://grafana.com/>

GIT: Control de versiones distribuido de código abierto diseñado para administrar proyectos de cualquier tamaño

<https://git-scm.com/>

GitHub: Plataforma para almacenar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de software

<https://github.com/>

GitFlow: Flujo de trabajo basado branchs o ramas en Git, en el que se usan las ramas principales y la feature (nuevas características). De modo que la rama feature la crean los desarrolladores para fusionarla con la rama principal, únicamente cuando cumpla con sus labores.

<https://aulasoftwarelibre.github.io/taller-de-git/gitflow/>

