

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Instituto de Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



**Implementación de perspectivas de Minería de
Procesos en Proyectos de Desarrollo de Software**

Tesis que para obtener el grado de
DOCTOR EN CIENCIAS

Presenta

Silvia Jaqueline Urrea Contreras

Directora de tesis

Dra. Brenda Leticia Flores Rios

Co-Directora de tesis

Dra. María Angélica Astorga Vargas

Mexicali, Baja California, enero de 2024.

RESUMEN de la Tesis de **SILVIA JAQUELINE URREA CONTRERAS**, presentada como requisito para la obtención del **GRADO DE DOCTOR EN CIENCIAS**. Mexicali, Baja California, México. Diciembre del 2023.

IMPLEMENTACIÓN DE PERSPECTIVAS DE MINERÍA DE PROCESOS EN PROYECTOS DE DESARROLLO DE SOFTWARE

Se presenta la **IMPLEMENTACIÓN DE PERSPECTIVAS DE MINERÍA DE PROCESOS EN PROYECTOS DE DESARROLLO DE SOFTWARE**, como la del flujo de control, caso, organizacional y del tiempo, para obtener una visión completa de los procesos e identificar áreas de mejora orientados a la industria de software. Para la implementación de la Minería de Procesos se adaptó el modelo de ciclo de vida L* desde un enfoque de Ingeniería de Software por capas: 1) Pre-procesamiento, 2) Minería de Procesos y 3) Perspectivas de Minería. A través de esta adaptación, se identificaron inconsistencias en las actividades en dos proyectos reales de desarrollo de software.

Los resultados obtenidos permitieron a la alta dirección y al equipo de mejora de una MiPyME (Micro, pequeña y mediana empresa) implementar un plan de acción para optimizar los procesos y aumentar la eficiencia y efectividad. Además, se identificaron retos y limitaciones en el proceso de extracción de datos, la comunicación y la documentación de los procesos, lo que sugiere áreas de mejora futura.

La aplicación de la metodología adaptada en capas propuesta demostró ser una herramienta valiosa para comprender y mejorar los procesos de las MiPyMEs de desarrollo de software. La integración de perspectivas permitió obtener una visión integral de los procesos, identificar oportunidades de mejora y tomar decisiones basadas en datos. Este enfoque basado en la Minería de Procesos ayudará a la industria de software a optimizar sus operaciones, mejorar flujos de trabajo y lograr resultados exitosos en sus proyectos de desarrollo de software.

TÉRMINOS CLAVE: Minería de Procesos, Procesos de Software, MiPyME, Perspectivas de Minería de Procesos, Ingeniería de Software.

Dedicatoria

A mi madre Silvia Contreras, mi hija Sofía Isabella y mi esposo Nahir Sánchez.

Reconocimientos

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por la financiación proporcionada para llevar a cabo mis estudios de doctorado y, de esta manera, tener la oportunidad de llevar a cabo la investigación presentada en esta tesis.

Al Instituto de Ingeniería agradezco la oportunidad que me han brindado de formar parte del programa de Maestría y Doctorado en Ciencias e Ingeniería (MyDCI) de la Universidad Autónoma de Baja California (UABC).

Agradezco a los miembros del comité de tesis, Dr. Jorge Eduardo Ibarra Esquer y Dr. Félix Fernando González Navarro, por su contribución a este trabajo de investigación. Así mismo al Dr. Iván García y la Dra. Carla Pacheco de la Universidad Tecnológica de la Mixteca por su orientación en el desarrollo de artículos relacionados con la investigación.

A las doctoras Dra. Brenda Leticia Flores Ríos y Dra. María Angélica Astorga Vargas por el tiempo brindado en la tutoría de cada logro académico alcanzado, y por las valiosas contribuciones que proporcionaron, las cuales han sido un factor determinante en mi avance constante.

Índice

Capítulo 1. Introducción	1
1.1 Conceptualización de la Minería de Procesos.....	2
1.1.1 Registro de Eventos	3
1.1.2 Tipos de Minería de Procesos	5
1.1.3 Perspectivas de Minería de Procesos	7
1.2 Planteamiento del Problema.....	10
1.3 Justificación	12
1.4 Preguntas de Investigación.....	14
1.5 Objetivo General y Específicos.....	15
1.6 Metodología	15
1.7 Estructura del documento.....	16
Capítulo 2. Estado del Arte.....	17
2.1 Planificación.....	19
2.1.1 Preguntas de Investigación	19
2.1.2 Selección de fuentes	19
2.1.3 Cadena de búsqueda	20
2.1.4 Criterios de Inclusión y Exclusión.....	20
2.2 Ejecución.....	21
2.2.1 Identificación y Selección de Estudios Primarios	21
2.2.2 Criterios de calidad	22
2.2.3 Extracción de información.....	24
2.3 Análisis de resultados.....	25

2.3.1	Identificación y definición de perspectivas de Minería de Procesos	26
2.3.2	Implementación de perspectivas en estudios de caso	28
2.3.3	Perspectivas de Minería de Procesos en la Ingeniería de Software	31
Capítulo 3. Marco Teórico		34
3.1	Metodologías de Minería de Procesos	34
3.1.1	Modelo de ciclo de vida L*	34
3.1.2	Metodología PM2	35
3.1.3	Metodología PCM	36
3.1.4	Metodología PDM	37
3.1.5	Comparativa de las metodologías	38
3.2	Notación	38
3.2.1	Definición de Evento	39
3.2.2	Definición de Caso	40
3.2.3	Definición de Traza	41
3.2.4	Definición de Registro de eventos	41
3.2.5	Definición de Petri net	42
3.3	Aplicabilidad de las definiciones formales	43
Capítulo 4. Una propuesta de aplicación de la Minería de Procesos en proyectos de software.....		47
4.1	Descripción de la metodología L* en capas.....	47
4.1.1	Actividades ETL: Desarrollo de un API.....	49
4.2	Propuestas de definiciones matemáticas.....	51
4.3	Análisis de Escenarios	54

Capítulo 5. Resultados del Estudio de caso	58
5.1 Caracterización de la empresa de desarrollo de software	58
5.1.1 Proceso de desarrollo de la MiPyME de software.....	59
5.1.2 Repositorios	60
5.1.3 Actividades del proceso de software	62
5.2 Análisis del Proyecto 1	63
5.2.1 Obtención del Registro de Eventos.....	64
5.2.2 Creación y descripción del modelo.....	66
5.2.3 Verificación de eventos y actividades	69
5.2.4 Extensión del Modelo de Procesos con las perspectivas	72
5.3 Análisis del Proyecto 2	79
5.3.1 Obtención del Registro de Eventos.....	79
5.3.2 Creación y descripción del modelo.....	80
5.3.3 Verificación de eventos y actividades	81
5.3.4 Extensión del Modelo de Procesos con las perspectivas	82
5.4 Comparación de proyectos.....	84
5.4.1 Modelo Integrado.....	84
Capítulo 6. Conclusiones y Trabajo Futuro	88
6.1 Conclusiones	88
6.2 Trabajo Futuro.....	89
Referencias	91
Anexo 1. Productos académicos.....	97

Publicaciones arbitradas en revistas y congresos	97
Registro de Modelo de Utilidad	98
Ponente por invitación a eventos académicos	98
Anexo 2. Desarrollo para las Actividades ETL.....	100
Anexo 3. Ejemplo de conjunto de trazas totales del estudio de caso.....	102

Lista de Tablas

Tabla 1. Fragmento de un registro de eventos.....	5
Tabla 2. Resultados del proceso de selección.....	22
Tabla 3. Criterios de calidad por cada Estudio Primario	22
Tabla 4. Resultados del proceso de selección.....	24
Tabla 5. Listado de Estudios Primarios	24
Tabla 6. Publicaciones por año	25
Tabla 7. Definición de perspectivas de Minería de Procesos	26
Tabla 8. Caracterización de las perspectivas en casos prácticos	30
Tabla 9. Áreas de aplicación de perspectivas de MP.....	31
Tabla 10. Aportación de las perspectivas en la Ingeniería de Software	32
Tabla 11. Comparativa de las definiciones y notaciones de evento	39
Tabla 12. Comparativa de las definiciones y notaciones de Caso	40
Tabla 13. Comparativa de las definiciones y notaciones de Traza.....	41
Tabla 14. Comparativa de las definiciones y notaciones de Registro de eventos	42
Tabla 15. Comparativa de las definiciones y notaciones de Petri net	42
Tabla 16. Definición y representación formal de perspectiva	45
Tabla 17. Relación de atributos para el minado de perspectivas.....	54
Tabla 18. Áreas y temas de conocimiento para la identificación del Escenario 1.....	55
Tabla 19. Áreas y temas de conocimiento para la identificación del Escenario 2.....	56
Tabla 20. Áreas y temas de conocimiento para la identificación del Escenario 3.....	56
Tabla 21. Áreas y temas de conocimiento para la identificación del Escenario 4.....	57
Tabla 22. Descripción de los estados de Jira utilizados por la MiPyME.	62
Tabla 23. Roles del equipo del proyecto.	63
Tabla 24. Actividades en el registro de eventos del Proyecto 1.	65
Tabla 25. Conjunto de trazas totales presentes en el registro de eventos Proyecto 1.....	69
Tabla 26. <i>Fitness</i> y alineaciones del conjunto de trazas en el registro de eventos.	71
Tabla 27. Porcentaje de la verificación de la conformidad Proyecto 1.	72

Tabla 28. Asociación del flujo de desarrollo con las etapas del ciclo de vida del desarrollo de software.....	72
Tabla 29. Especificación del Tiempo por actividad Proyecto 1.	77
Tabla 30. Actividades en el registro de eventos del Proyecto 2.	79
Tabla 31. Conjunto de trazas totales presentes en el registro de eventos Proyecto 2.....	81
Tabla 32. Porcentaje de la verificación de la conformidad Proyecto 2.	82
Tabla 33. Frecuencia y porcentaje de participación Proyecto 1.	83
Tabla 34. Especificación del Tiempo por actividad Proyecto 2.	83
Tabla 35. Comparativa entre el Proyecto 1 y el Proyecto 2.	86

Lista de Figuras

Figura 1. Minería de Procesos como parte de la Ciencia de datos y de procesos.	3
Figura 2. Meta modelo de Minería de Procesos (Poncin et al., 2011).....	4
Figura 3. Tipos básicos de Minería de Procesos.....	6
Figura 4. Enfoque para la obtención de un modelo integrado (Van der Aalst, 2016). ...	10
Figura 5. Incorporación de la Minería de Procesos al desarrollo de software.....	13
Figura 6. Diagrama del protocolo de la Revisión Sistemática de la Literatura.	18
Figura 7. Proceso de selección de Estudios Primarios	21
Figura 8. Representación de los elementos de Minería de Procesos.	39
Figura 9. Red Petri de los estados de transición de un proceso de software.	46
Figura 10. Aplicación de la Metodología L* para proyecto desarrollo de software	48
Figura 11. Query utilizado para obtener el listado de <i>issues</i> del proyecto.	50
Figura 12. Filtrado y vinculación de los atributos obtenidos de Jira.	50
Figura 13. Query para la obtención de los <i>commits</i> por <i>issue</i>	51
Figura 14. Diagrama del flujo del proceso de desarrollo de la empresa de software.	60
Figura 15. Fragmento del registro de eventos del proceso de desarrollo de software....	65
Figura 16. Modelo de procesos real Proyecto 1	67
Figura 17. Histograma de eventos por trazas.....	68
Figura 18. Secuencia de actividades de la traza con mayor Proyecto 1.	68
Figura 19. Resultados de la verificación de la Conformidad.....	71
Figura 20. Modelo de flujo de control del ciclo de vida de desarrollo de software.	73
Figura 21. Modelo de flujo de control de las actividades (estados del <i>issue</i> de Jira)....	73
Figura 22. Modelo de flujo de control (Conformidad al 100%).....	74
Figura 23. Secuencia de actividades de la segunda traza Proyecto 1 (<i>Feature</i>).....	75
Figura 24. Frecuencia y porcentaje de participación Proyecto 1.....	76
Figura 25. Modelo de procesos con tiempo promedio entre actividades.	78
Figura 26. Modelo de procesos real Proyecto 2.	80
Figura 27. Secuencia de actividades de la traza con mayor presencia Proyecto 2.	81
Figura 28. Secuencia de actividades de la segunda traza Proyecto 2 (<i>Feature</i>).....	82

Figura 29. Modelo integrado del Proyecto 1.	84
Figura 30. Modelo integrado del Proyecto 2.	85
Figura 31. Codificación para las actividades ETL (parte 1).	99
Figura 32. Codificación para las actividades ETL (parte 2).	100
Figura 33. Codificación para las actividades ETL (parte 3).	101

Capítulo 1. Introducción

El desarrollo que las tecnologías de la información han experimentado en las últimas dos décadas ha cambiado de manera sustancial, entre otras cosas, la forma en que las organizaciones operan y compiten. Generando, además, múltiples perspectivas y oportunidades adicionales (Pablos Heredero et al., 2019). Este avance ha llevado al surgimiento de un universo digital, es decir, la totalidad de la información almacenada e intercambiada electrónicamente en los sistemas de información, el cual está alineado con los procesos de las organizaciones que hacen posible registrar y analizar eventos (TFPM – IEEE, 2012).

Con la gran cantidad de datos que el universo digital genera y almacena, emergen nuevos desafíos y objetivos a los cuales se les tiene que hacer frente; en estos días no se trata de acumular la mayor cantidad de datos posible, sino de manejarlos de manera eficiente, analizarlos y obtener resultados y conocimientos valiosos para las organizaciones en un plazo razonable (Štolfa, 2018). Se trata de explotar los datos de eventos de una manera significativa, por ejemplo, con el propósito de detectar obstáculos, prever posibles inconvenientes, registrar incumplimientos de políticas, sugerir medidas correctivas y optimizar la agilidad de los procesos, pero sobre todo proporcionar información valiosa (TFPM – IEEE, 2012).

En este sentido, la información se encuentra almacenada en repositorios generada por la ejecución de las actividades propias de la administración de las organizaciones; por ello, resulta conveniente diseñar sistemas para producirla y gestionarla, con el objetivo de asegurar que la información sea fiable y esté disponible en el momento de tomar una decisión (de Pablos Heredero et al., 2019). Por esta razón, los sistemas de gestión y de información comenzaron a orientarse cada vez más a los procesos, es decir, orientados a las acciones y pasos a seguir, que directamente a los datos que trabajan (Štolfa, 2018).

Los sistemas de información conscientes del proceso se definen como software que administra y ejecuta procesos operativos involucrando a individuos, aplicaciones y/o fuentes de datos basados en modelos de procesos (Dumas y otros, 2018). Estos sistemas proporcionan registros de eventos

directos. No obstante, la mayoría de estos sistemas almacenan dichos datos en una forma no estructurada, como por ejemplo, los datos de eventos que están dispersos en múltiples tablas o deben ser extraídos de subsistemas para la comunicación. En estos escenarios, los datos de eventos existen, pero se requiere un esfuerzo adicional para su extracción (Van der Aalst, 2016).

La Minería de Procesos es un campo en desarrollo que considera la obtención de datos como un componente fundamental de sus actividades, estableciendo vínculos entre los procesos reales y sus datos, por un lado, y los modelos de proceso, por otro lado. Como se mencionó, el universo digital y el universo físico se alinean cada vez más (Van der Aalst, 2016). Existe una variedad de estudios de casos e informes de experiencia de Minería de Procesos en áreas comerciales, tales como: Atención médica, administración pública, transporte y educación, con un enfoque particular en las aplicaciones de la Minería de Procesos en la Gestión de Procesos de negocios (Rubín et al., 2014).

Dado que la Minería de Procesos es un campo relativamente nuevo en comparación con las disciplinas que lo forman (Rubín et al., 2014), su potencial de crecimiento es bastante alto, tanto en términos teóricos como prácticos. Se ha observado como la Minería de Procesos se utiliza para verificar el cumplimiento, diagnosticar desviaciones, identificar cuellos de botella, mejorar el rendimiento, predecir tiempos de flujo y recomendar acciones en diversos ámbitos (Van der Aalst, 2016).

1.1 Conceptualización de la Minería de Procesos

La Ciencia de Datos se define como un campo interdisciplinario que tiene sus raíces en disciplinas como la estadística, la minería de datos y las bases de datos. Este surgimiento se debió en gran medida a la disponibilidad cada vez mayor de datos y la necesidad de extraer valor de estos recursos (Van der Aalst, 2016). En contraste, la Ciencia de Procesos se presenta como una disciplina que fusiona el conocimiento proveniente de las tecnologías de la información y las Ciencias administrativas. Su enfoque se centra en la gestión y mejora de los procesos de negocio, y abarca áreas adicionales como la automatización de procesos, la gestión de flujos de trabajo y la Minería de Procesos (Van der Aalst, 2016). Una definición de Minería de Procesos es la

propuesta por Van der Aalst (2016) como la unión de la Ciencia de datos y Ciencia de Procesos (Figura 1), con el propósito de realizar tres tipos de minería (Rubín et al., 2014; Aguirre Mayorga y Rincón García, 2015): 1) Descubrimiento de procesos, 2) Verificación de la conformidad y 3) Mejora de procesos.

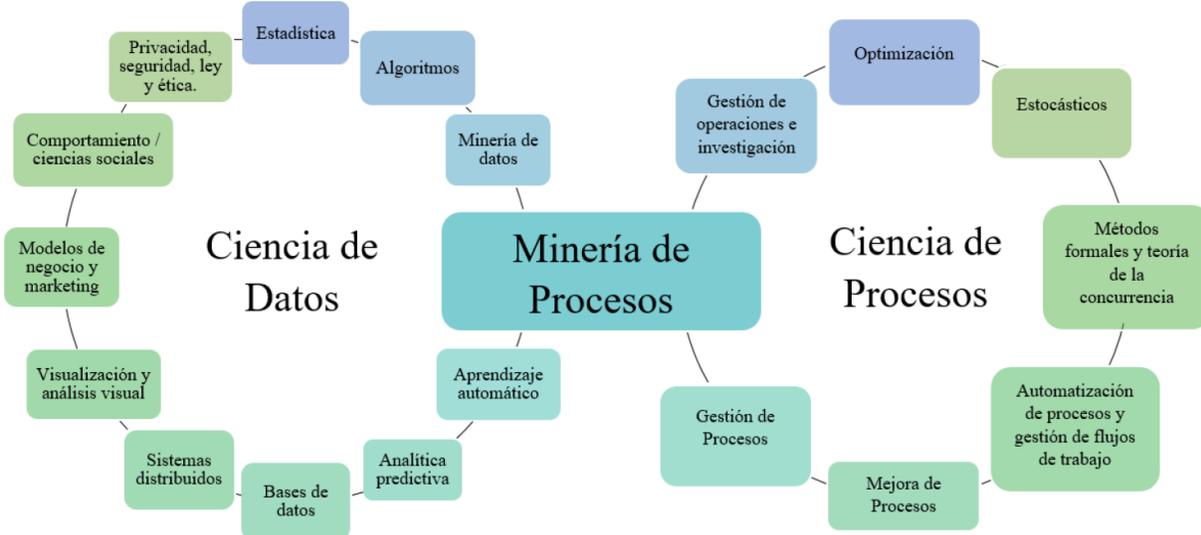


Figura 1. Minería de Procesos como parte de la Ciencia de datos y Ciencia de procesos.
Traducida de (Van der Aalst, 2016).

La Minería de Procesos actúa como un vínculo entre la Ciencia de Datos y la Ciencia de Procesos (Van der Aalst, 2016). Al combinar datos de eventos y modelos de procesos, es posible abordar una amplia gama de cuestiones relacionadas con la conformidad y el rendimiento a través del análisis de registros de eventos.

1.1.1 Registro de Eventos

Los datos generados por la realización de las actividades establecidas en los modelos de procesos se recopilan en registros de eventos (Van der Aalst, 2016). Estos registros se presentan en forma de tablas que contienen información sobre las actividades que han sido ejecutadas (Rivas y Bayona-Oré, 2019).

Como se describe en (Van der Aalst, 2016) el concepto de registros de eventos se basa en los siguientes supuestos (Kneuper, 2018) representados en la Figura 2 a través del meta modelo del registro de eventos de la Minería de Procesos:

- Un registro de eventos contiene datos de varios procesos (generalmente uno).
- Un proceso (o su ejecución) consta de casos o instancias individuales.
- Un caso consta de diferentes eventos, donde cada evento pertenece exactamente a un caso.
- Se ordenan los eventos que pertenecen a un caso. Este orden se puede utilizar para analizar el flujo de control del proceso.
- Los eventos tienen atributos adicionales, que incluyen al menos la actividad de proceso realizada por el evento, además de atributos opcionales como la descripción de la actividad, marca de tiempo (timestamp) o las personas involucradas.

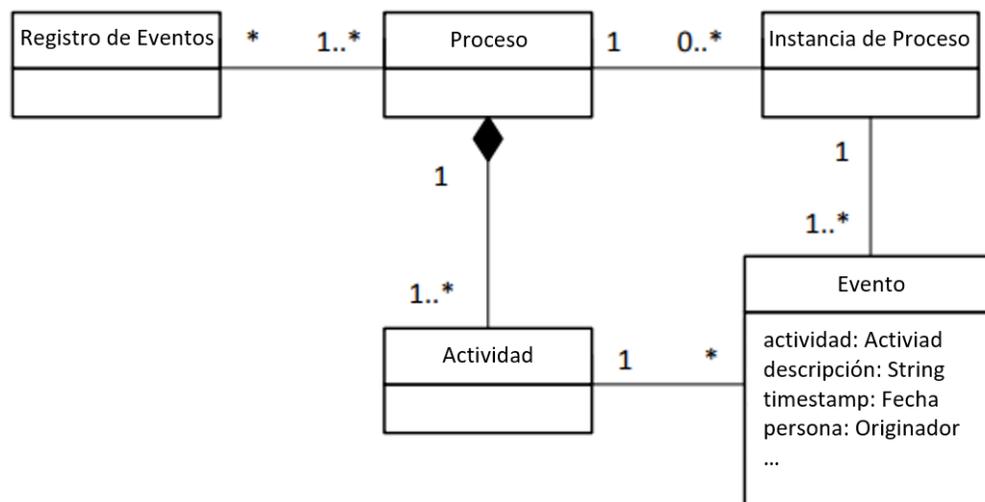


Figura 2. Meta modelo de Minería de Procesos (Poncin et al., 2011).

La mayoría de los registros de eventos almacenan información adicional sobre los eventos (Van der Aalst, 2016). La información que contenga un registro de eventos dependerá de las técnicas de Minería de Procesos y las preguntas en cuestión que se necesiten responder con su análisis.

Después de definir los registro de eventos, se muestra en la Tabla 1 un fragmento correspondiente a un registro de eventos relacionado a un proceso de inscripción para conocer sus elementos mínimos y las características que lo conforman. Cada línea representa un evento, los cuales se

encuentran agrupados por casos. Cada evento contiene información sobre el caso (instancia del proceso), un identificador de evento (ID Evento) y una marca de tiempo. Este último puede variar en su especificación de una forma general a una más detallada (inicio y fin de la actividad) o en formatos diferentes dependiendo del sistema de información o repositorio donde se encuentre. En el ejemplo, los eventos están asociados a un recurso (nombre del usuario que realizó la actividad) y un costo. Las columnas ID Caso, ID Evento, Timestamp y Actividad son las mínimas requeridas por el registro de eventos.

Tabla 1. Fragmento de un registro de eventos.

ID Caso	ID Evento	Timestamp	Actividad	Recurso	Costo	...
1	35654803	06-09-2018 : 15.02	Solicitud de registro	Sofía	50	...
1	35654822	06-09-2018 : 16.06	Examinación regulatoria	Elisa	400	...
1	35654824	07-09-2018 : 16.22	Revisión de ficha	Miguel	100	...
2	35654847	07-09-2018 : 16.52	Decisión	Mía	200	...
2	35654858	15-09-2018 : 11.47	Pagar inscripción	Mateo	200	...
...

La información puede variar dependiendo de las propiedades que se necesiten para formular cualquier registro de eventos. Sin embargo, los requisitos mínimos para la aplicación de la Minería de Procesos son que cualquier evento puede estar relacionado tanto con un caso como con una actividad y que los eventos, dentro de un caso, deben estar ordenados (Van der Aalst, 2016).

De este modo, un registro de eventos se convierte en una herramienta que posibilita descubrir, monitorear y analizar el cumplimiento de regulaciones, al mismo tiempo que permite verificar la integridad y la credibilidad de los datos relativos a los procesos fundamentales de la entidad. (Van der Aalst, 2016, Pérez-Alfonso et al., 2013).

1.1.2 Tipos de Minería de Procesos

La Minería de Procesos no se limita al descubrimiento de procesos solamente; los registros de eventos se pueden usar para verificar la conformidad y mejorar los modelos de procesos existentes (Van der Aalst, 2016). Por esta razón la Minería de Procesos cuenta con tres principales tipos de

minado que también se les puede referir como técnicas de Minería de Procesos. Como se observa en la Figura 3, cada tipo de minado tiene su propia entrada y salida dependiendo de la técnica a implementar.

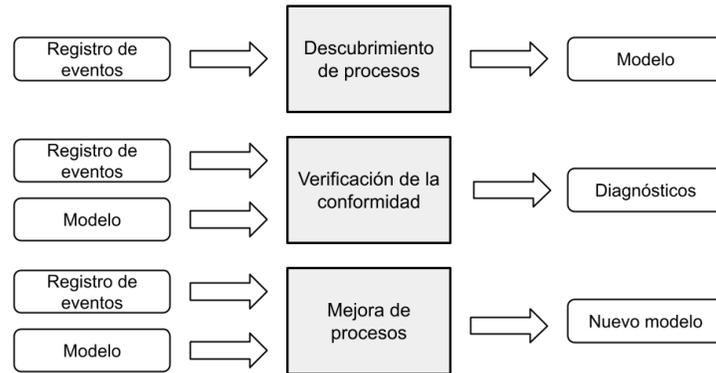


Figura 3. Tipos básicos de Minería de Procesos explicados en términos de entrada y salida (TFPM – IEEE, 2012).

El primer tipo de Minería de Procesos es el descubrimiento, siendo la técnica más destacada. El descubrimiento toma un registro de eventos como entrada y produce un modelo. El modelo descubierto es típicamente un modelo de proceso (por ejemplo, un diagrama de actividad de Petri net, BPMN, EPC o UML) (Van der Aalst, 2016, Liu et al., 2016). Las técnicas de descubrimiento de procesos tienen sus raíces en varias disciplinas, como la minería de datos, la inteligencia computacional y el aprendizaje automático (Erdem y Demirörs, 2017). Desde el año 1998 hasta la actualidad se han propuesto varias técnicas de descubrimiento, las cuales utilizan enfoques muy diferentes, por ejemplo, los enfoques heurísticos, de programación de lógica inductiva, regiones basadas en estado, regiones basadas en lenguaje y algoritmos genéticos (Van der Aalst, 2015).

El segundo tipo de Minería de Procesos es la verificación de conformidad. Las técnicas de verificación de conformidad necesitan un registro de eventos y un modelo como entrada; el resultado consiste en información de diagnóstico que muestra diferencias y puntos en común entre el modelo y el registro (Van der Aalst, 2016, Liu et al., 2016). La verificación de conformidad se puede aplicar a diferentes tipos de modelos como los de procedimiento, modelos organizacionales, modelos de procesos declarativos, reglas/políticas comerciales, leyes, entre otras (Van der Aalst, 2016, TFPM – IEEE, 2012). En la última década se han desarrollado varias

técnicas de verificación de conformidad, las técnicas más avanzadas crean alineaciones, es decir, el comportamiento observado se relaciona con el comportamiento modelado incluso si hay desviaciones (Van der Aalst, 2015).

El tercer tipo de Minería de Procesos es la mejora. La mejora toma un registro de eventos y un modelo existente como entradas e intenta mejorar o ampliar el modelo utilizando información adicional en el registro (Van der Aalst, 2016, Liu et al., 2016). Las técnicas de mejora de procesos utilizan una combinación de datos históricos, modelos aprendidos y el estado actual del modelo (Van der Aalst, 2015). Un tipo de mejora es la reparación, es decir, modificar el modelo para reflejar mejor la realidad; otro tipo es la extensión, que consiste en agregar una nueva perspectiva al modelo de proceso mediante la correlación cruzada con el registro (Van der Aalst, 2016). Por ejemplo, al usar marcas de tiempo en el registro de eventos, se puede extender el modelo para mostrar cuellos de botella, niveles de servicio, tiempos de rendimiento y frecuencias (Van der Aalst, 2016, TFPM – IEEE, 2012).

En general, al obtener un modelo de procesos minado con algún tipo de Minería de Procesos se muestra solo su flujo de control. Sin embargo, al extender estos modelos, se agregan perspectivas adicionales para proporcionar una buena caracterización de los aspectos que el proceso de minería pretende analizar (Van der Aalst, 2016). Por lo tanto, de forma integradora a los tres tipos de minería (Descubrimiento, Conformidad y Mejora), se pueden identificar las perspectivas de flujo de control, organizacional, del caso y del tiempo.

1.1.3 Perspectivas de Minería de Procesos

Para definir lo que es y en qué consisten las perspectivas de Minería de Procesos, se establecieron los elementos integradores que intervienen para representar su significado: perspectiva y Minería de Procesos. La Real Academia Española (RAE) define el término *perspectiva* como “punto de vista desde el cual se considera o se analiza un asunto”. Van der Aalst (2016) define que el propósito de la Minería de Procesos es descubrir, monitorear y mejorar procesos reales a partir de los registros de eventos para derivar modelos de procesos. A partir de lo anterior, se puede definir que las perspectivas son un punto de vista del análisis de los registros de eventos

procesados por medio de los tipos de Minería de Procesos para caracterizar los modelos de procesos derivados.

Las perspectivas como lo son la de flujo de control, organizacional, del caso y del tiempo ofrecen una descripción de los aspectos del proceso de minado, permiten correlacionarse de forma cruzada para encontrar información relevante y pueden superponerse parcialmente entre sí (Van der Aalst, 2016).

Flujo de control es responsable de analizar el orden de las actividades. Su objetivo es encontrar una buena caracterización de todos los caminos posibles, por ejemplo, expresados en términos de una red de Petri o alguna otra notación (Van der Aalst, 2016, Caldeira et al., 2016).

Por otra parte, la organizacional se concentra en destacar los recursos involucrados en el proceso, que incluyen a individuos, sistemas y roles, y examina cómo estos elementos están interconectados. Su propósito principal radica en organizar la estructura organizativa al categorizar a las personas en función de sus roles y las unidades organizativas correspondientes, así como en proporcionar una representación visual de la red social existente en la organización. (Van der Aalst, 2016, Caldeira et al., 2016).

Con respecto a la del caso, se dedica a analizar las características de los casos, los cuales pueden ser definidos por la secuencia de actividades que siguen en el proceso o por las personas responsables de llevar a cabo esas actividades. No obstante, también es posible caracterizar los casos en función de los valores asociados a los elementos de datos que involucran. (Van der Aalst, 2016).

Por último, la del tiempo se encuentra vinculada a la frecuencia de los eventos y su duración, lo que brinda la capacidad de anticipar la cantidad de tiempo que falta para la finalización del proceso, identificar situaciones de cuellos de botella, supervisar el aprovechamiento de los recursos y evaluar los niveles de servicio. (Van der Aalst, 2016, Caldeira et al., 2016).

Para obtener un modelo totalmente integrado con las perspectivas se establece un enfoque de cinco pasos (Van der Aalst, 2016) el cual contempla la incorporación de las perspectivas de flujo

de control, organizacional, del caso y del tiempo. Los pasos del enfoque (representados en la Figura 4) son los siguientes:

1. **Obtener un registro de eventos.** Se debe extraer un registro de eventos de una variedad de sistemas; esto representa ser un proceso iterativo en el cual se exploran los registros y filtran procesos.
2. **Crear o descubrir un modelo.** Existen técnicas para el descubrimiento de procesas; técnicas como la minería heurística o la genética pueden ser usadas para obtener un modelo. También los modelos hechos a mano pueden ser usados.
3. **Conectar eventos del registro a actividades en el modelo.** Este paso es esencial para agregar información y perspectivas al modelo. Los eventos en el registro y las actividades en el modelo tienen que estar conectados.
4. **Ampliar el modelo.** Se encuentra dividido en 4 sub-pasos.
 - a. **Agregar la perspectiva organizacional:** es posible realizar un análisis de una red social y luego identificar unidades organizativas que establecen vínculos entre las actividades y conjuntos de recursos.
 - b. **Agregar la perspectiva del tiempo:** las marcas de tiempo y las frecuencias se pueden emplear para comprender las distribuciones de probabilidad que adecuadamente representan los intervalos de espera y de servicio.
 - c. **Agregar la perspectiva del caso:** emplear los atributos presentes en el registro para la minería de decisiones; esta práctica indica cuáles datos son significativos y deben ser incorporados en el modelo.
 - d. **Agregar otras perspectivas:** según la información contenida en el registro, es factible incorporar otras perspectivas al modelo, como la inclusión de detalles relacionados con riesgos y costos.
5. **Retornar el modelo integrado.** Se pueden utilizar para el análisis y exploración de diferentes estrategias de mejora.

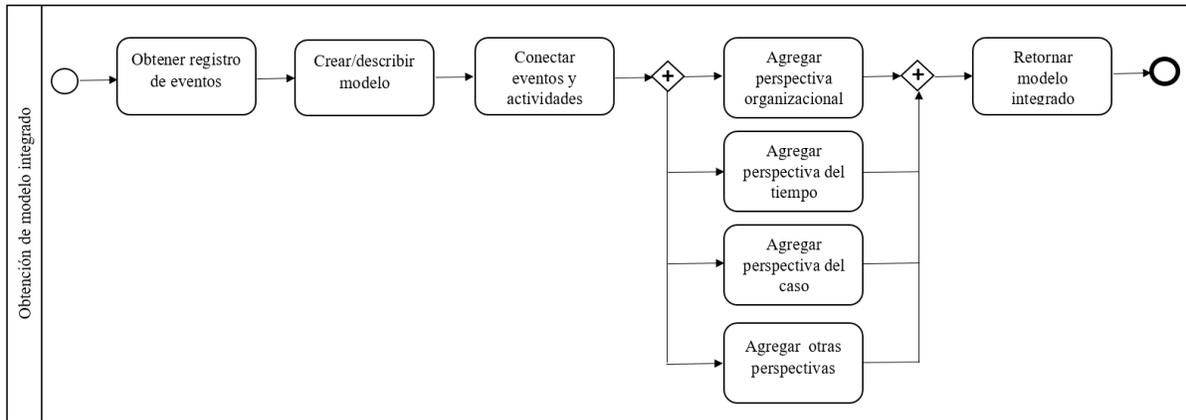


Figura 4. Enfoque para la obtención de un modelo integrado (Van der Aalst, 2016).

Por todo lo anterior, resulta conveniente durante el análisis observar las perspectivas de flujo de control, organizacional, del caso y del tiempo desde un mismo modelo de procesos para su comprensión.

1.2 Planteamiento del Problema

Una de las primeras publicaciones sobre Minería de Procesos fue en 1998 por Cook y Wolf (Cook y Wolf, 1998) en la cual se utilizó la construcción de modelos de procesos de software a partir de un registro de eventos. En ese momento no se le denominó como Minería de Procesos (el término se comenzó a establecer a principios de los 2000), pero a partir de ese entonces, la Minería de Procesos ha tenido una gran diversidad de aplicaciones en diversos ámbitos convirtiéndose hoy en día en una disciplina posicionada. Sin embargo, aunque la Minería de Procesos tuvo sus primeras aplicaciones en el área de procesos de software, su investigación se desvió hacia los procesos de negocios. Esta desviación ocurrió debido a que los procesos de software generalmente no se encuentran tan estructurados como los procesos de negocios (Kneuper, 2018).

Actualmente, la tendencia de la Minería de Procesos ha recobrado su camino de origen y se han expuesto estudios sobre su aplicación en la Ingeniería de Software. Algunos ejemplos de dichos estudios son los presentados por (Van der Aalst, 2015; Keith y Vega, 2016; Caldeira et al., 2016; Dong et al., 2017; Štolfa, 2018; Marques et al., 2018; Kneuper, 2018). Asimismo, diversos

artículos como (Rubin et al., 2014; Liu et al., 2016; Erdem et al., 2018) manejan el término *Software Process Mining* (Minería de Procesos de Software) para referirse a esta área de investigación.

La Minería de Procesos ofrece técnicas genéricas que pueden emplearse en cualquier tipo de proceso (si hay datos correctos disponibles) haciendo posible su aplicación al desarrollo de software (Keith y Vega, 2016), causando que la Minería de Procesos de Software sea un campo de investigación en desarrollo (Erdem et al., 2018; Rubin et al., 2014). Por lo tanto, la Minería de Procesos de Software es la aplicación de la Minería de Procesos en el desarrollo de software. Permite extraer conocimiento de los datos de ejecución del software, lo que ayuda a los analistas de esta área a comprender mejor el comportamiento del mismo (Liu et al., 2016).

Existen problemáticas detectadas por diferentes autores durante el proceso del desarrollo de software en dónde la intervención de la Minería de Procesos se contempla como una solución realizable. A continuación, se muestra una relación sin priorizar de las problemáticas identificadas por parte del desarrollo de software, asimismo, desafíos que la Minería de Procesos debe afrontar. Cada uno de los problemas identificados que se presentan en el siguiente listado se vieron motivados por las siguientes razones:

- **Limitar a las perspectivas de Minería de Procesos:** Las investigaciones solo se enfocan en los procesos desde la perspectiva de flujo de control, mientras que se ha prestado menos atención a las demás perspectivas (Zhao y Zhao, 2014; Van der Aalst, 2016).
- **Presentar fallas en el modelado y desviaciones al aplicar los procesos:** Las empresas de software enfrentan dificultades como errores durante la creación del modelo de procesos y desviaciones al ejecutar los procesos establecidos, lo que puede resultar en gastos considerables (Van der Aalst, 2015).
- **Elegir una representación adecuada del modelo a obtener:** El modelo a obtener debe de contar con una representación adecuada para que sea posible su comprensión y analizarlo adecuadamente (Burattin, 2015).
- **Disponer de datos adecuados proporcionados por los sistemas de información conscientes de los procesos:** Los sistemas de información pueden no disponer de datos

adecuados, lo que conlleva a las situaciones de: 1) Falta de datos para aplicar algoritmos de Minería de Procesos (Alpha, heurístico, genético, difuso, entre otros) y 2) La cantidad de detalles en los datos es excesiva y la complejidad del modelo resultante lo hace poco práctico (Keith y Vega, 2016).

- **Estructurar de manera apropiada los procesos de software:** Los procesos de software generalmente no están bien estructurados en comparación con los procesos de negocios, con una variación considerable entre diferentes instancias del mismo proceso (Kneuper, 2018). Asimismo, se ha identificado que el uso de herramientas de Minería de procesos representa un reto acerca del nivel de conocimiento técnico (Flores-Rios, 2016) requerido por los responsables de proyectos, responsable del programa de mejora de procesos (SPI), de software o de desarrollo. Éstos roles tienen que decidir cuáles son las herramientas más adecuadas para analizar los registros de eventos o verificar el comportamiento de los procesos (Ailene, 2011). También, poseer conocimiento especializado en algoritmos o técnicas de Minería de procesos dependiendo el tipo de análisis según los formatos de archivos que las herramientas utilizan (Urrea-Contreras et al., 2016).

Se propone incorporar a la Minería de Procesos en el desarrollo de software como un soporte para el diagnóstico, tratamiento y prevención de problemas relacionados con el software. Adicionalmente, emplear las perspectivas para explorar de manera más eficiente los registros de eventos desde los distintos puntos de vista del flujo, organizacional, del caso y del tiempo con la finalidad de resolver problemas identificados en los procesos de software y minimizar su impacto.

1.3 Justificación

Los estudios vinculados a la Minería de Procesos de Software exponen que es posible utilizar la Minería de Procesos para detectar discrepancias en los procesos de software, obteniendo un gran volumen de datos que se pueden usar para planificar correcciones y realizar mejoras (Keith y Vega, 2016). De manera general, los estudios del área se centran en el uso de la Minería de Procesos con la finalidad de mejorar productos y procesos de software.

La Minería de Procesos se contempla como una solución realizable al momento de observar su incorporación en el proceso desarrollo de software. Se han descrito diferentes formas de llevar a cabo dicha incorporación; una de ellas es el uso de la Minería de Procesos como herramienta de los procesos de software, por ejemplo, para el análisis de requisitos o investigar la ejecución de los sistemas de software, otra manera es la aplicación directa de la Minería de Procesos a los datos de los procesos de software (Kneuper, 2018), la cual describe en la Figura 5 a continuación.

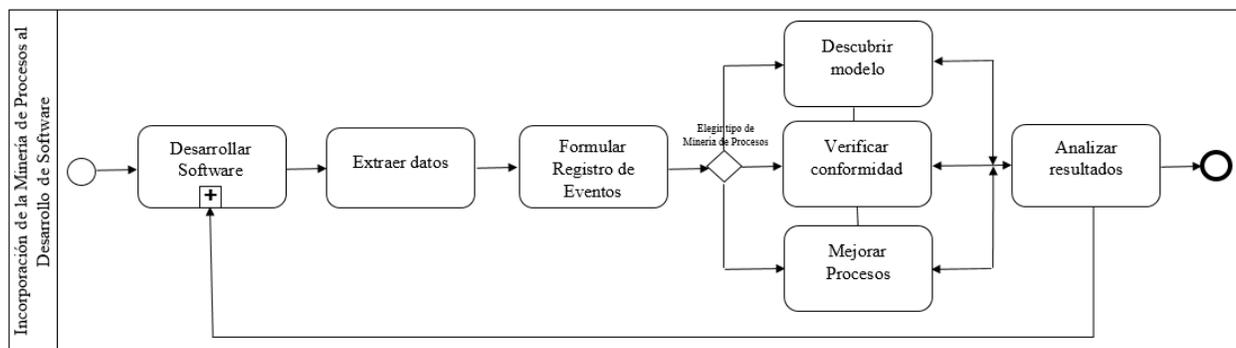


Figura 5. Incorporación de la Minería de Procesos al desarrollo de software.

En la Figura 5, se observa que el desarrollo de software se encuentra como subproceso el cual comprende los sistemas de software en ejecución y los datos de procesos de software. Estos datos son extraídos y transformados para poder crear un registro de eventos que posteriormente es procesado por los tipos de minado de la Minería de Procesos. Los resultados obtenidos serán analizados con la finalidad de enriquecer al desarrollo de software con la obtención de nuevos modelos y diagnósticos.

Los modelos obtenidos se pueden extender con las perspectivas de Minería de Procesos, lo cual ayudará a identificar patrones de procesos de desarrollo, detectar tendencias y realizar predicciones (Caldeira et al., 2016). Las perspectivas pueden dar respuesta a preguntas tales como: la perspectiva organizacional. (“¿Cuáles son los roles organizacionales y qué recursos están realizando actividades particulares?”), la perspectiva del caso (“¿Qué características de un caso influyen en una decisión en particular?”), y la perspectiva del tiempo (“¿Dónde están los cuellos de botella en mi proceso?”) (Van der Aalst, 2016). Aportar una visualización más al modelo, no

solo la perspectiva de flujo de control, representa un beneficio al caracterizar más a detalle el proceso.

Por esta razón, se destaca la Minería de Procesos por su capacidad de descubrir procesos reales, evaluar su conformidad con respecto al modelo oficial y encontrar oportunidades de mejora teniendo un alto impacto en las organizaciones de desarrollo de software (Keith y Vega, 2016). Además, por su condición de poder extender un modelo que contiene diferentes aspectos del proceso como flujo de control e información organizacional y su utilización dentro en del desarrollo de software que resulta valiosa por las siguientes razones: 1) Utilizar técnicas de descubrimiento de procesos para capturar el comportamiento real del software, 2) Emplear técnicas de verificación de conformidad para detectar desviaciones, 3) Predecir problemas relacionados con el rendimiento o la conformidad a partir de la alineación de los modelos y el comportamiento real del software (Van der Aalst, 2015).

1.4 Preguntas de Investigación

En el contexto de las MiPyMEs especializadas en el desarrollo de software, esta investigación se enfoca en abordar interrogantes fundamentales que buscan enriquecer la comprensión y la eficiencia de sus procesos, además de mostrar la aplicabilidad de la Minería de Procesos en este sector. A continuación, se presentan las siguientes preguntas de investigación:

1. ¿Qué metodología de Minería de Procesos pueden implementar las MiPyMEs de desarrollo de software?
2. ¿Las MiPyMEs de desarrollo de software poseen procesos estructurados que posibiliten la extracción de registro de eventos para implementar la Minería de Procesos?
3. ¿El proceso de desarrollo de software ejecutado entre los diferentes sistemas de gestión de la información de las MiPyMEs se lleva a cabo de acuerdo al proceso documentado?
4. ¿Cuáles son los retos que enfrentan las MiPyMEs de desarrollo de software al aplicar la Minería de Procesos con las perspectivas de flujo de control, organizacional, del caso y del tiempo?

1.5 Objetivo General y Específicos

Objetivo General: Implementar las cuatro perspectivas de Minería de Procesos a través de análisis de registros de eventos reales bajo una metodología de Minería de Procesos para descubrir conocimiento generado por proyectos de desarrollo de software para los siguientes ciclos de mejora.

Objetivos Específicos (OE) se establecieron los siguientes:

- **OE1.** Implementar una metodología en el minado de registros de eventos reales para proyectos de desarrollo de software dentro de una MiPyME (Micro, pequeña y mediana empresa) mexicana, como una técnica de Ingeniería de Software.
- **OE2.** Desarrollar una interfaz de programación de aplicaciones (API por sus siglas en inglés) que permita relacionar los registros de eventos de los sistemas de gestión de software de la MiPyME para la extracción, transformación y carga de sus datos.
- **OE3.** Identificación de los retos en la implementación del preprocesamiento, la Minería de Procesos y el minado de las perspectivas en las MiPyME de desarrollo de software.

1.6 Metodología

Para lograr los objetivos planteados en esta investigación, la metodología a seguir comprende las siguientes fases:

- I. Aplicar una metodología de Minería de Procesos en una MiPyME mexicana dedicada al desarrollo de software. Se definirá el contexto empresarial, la elección y adaptación de una metodología de Minería de Procesos, la adquisición y preprocesamiento de datos, la aplicación de la metodología, y la evaluación de los resultados para mejorar los procesos de desarrollo de software. Realizar una estancia de investigación en la MiPyME de por lo menos un año.
- II. Identificar los APIs para relacionar los registros de eventos de los sistemas de gestión de software durante la etapa de pre-procesamiento de los datos.

- III. Analizar los resultados para identificar similitudes en los desafíos, la comparación entre proyectos de desarrollo de software y la generación de recomendaciones en el abordaje de los retos identificados.

1.7 Estructura del documento

El documento se estructura de la siguiente forma, el Capítulo 2 corresponde al estado del arte relacionada a las perspectivas de Minería de procesos en Ingeniería de software; el Capítulo 3 muestra el marco teórico de las diferentes metodologías y definiciones formales matemáticas de elementos de la Minería de procesos; el Capítulo 4 presenta la propuesta de la adaptación de la Metodología L* por medio de capas para brindar un enfoque de Ingeniería de Software, así como las notaciones matemáticas establecidas en esta investigación y su aplicabilidad en proyectos de Minería de procesos de software a través de áreas y temas de conocimiento para la identificación y análisis de escenarios; en el Capítulo 5 corresponde al análisis de la aplicación de las perspectivas en el estudio de caso de dos proyectos de software, en el Capítulo 6 se muestran las conclusiones y trabajo futuro.

Capítulo 2. Estado del Arte

A partir de la publicación del Manifiesto de la Minería de Procesos en el 2012, por parte de la IEEE Task Force on Process Mining (TFPM – IEEE, 2012) se dispuso el establecimiento de la Minería de Procesos como una disciplina y la definición de un conjunto de principios rectores y desafíos a abordar, concediendo un paso significativo para promover la investigación, desarrollo, educación, implementación y evolución de la misma. La Minería de Procesos ofrece técnicas genéricas que pueden emplearse en cualquier tipo de proceso haciendo posible su aplicación al desarrollo de software (Keith y Vega, 2016), generando que sea un campo de investigación en desarrollo (Rubin et al., 2014; Erdem et al., 2018).

En una búsqueda preliminar realizada se identificó que existen estudios de caso e investigaciones sobre la aplicación de la Minería de Procesos en la disciplina de Ingeniería de Software (Rubin et al., 2014; Liu et al., 2016; Erdem et al., 2018, Keith y Vega, 2016). Actualmente, con el avance que han logrado las investigaciones de Minería de Procesos, se ha incorporado una progresión en los estudios sobre la aplicación de la Minería de Procesos en la Ingeniería de Software. Algunos ejemplos de dichos estudios son los presentados por (Keith y Vega, 2016; Kneuper, 2018; Van der Aalst., 2015; Caldeira et al., 2016; Dong et al., 2017). El proceso de incorporación de la Minería de Procesos en las prácticas de la Ingeniería de Software depende, en gran medida, de la manera en cómo las organizaciones de desarrollo de software registran la ejecución de las actividades de sus procesos. La recopilación de datos y el pre-procesamiento se identifican como los aspectos más difíciles y críticos de la Minería de Procesos (Özdağoğlu y Kavuncubaşı, 2019), debido a que los datos pueden estar incompletos y/o distribuidos en varias fuentes de información (TFPM – IEEE, 2012). Esto implica un esfuerzo disciplinado y organizado que deben seguir las organizaciones de desarrollo de software para la adopción de la Minería de Procesos. Por lo anterior, es importante realizar una revisión sistemática que permita conocer y establecer la relación existente entre ambas disciplinas.

Una revisión sistemática de la literatura (SLR por las siglas de *Systematic Literature Review*) es un medio para identificar, evaluar e interpretar toda la investigación disponible relevante para una

pregunta de investigación en particular, un área temática o un fenómeno de interés (Kitchenham, 2004). En este proyecto son las perspectivas de Minería de Procesos para identificar su intervención en proyectos de desarrollo de software con la finalidad de conocer su estado en esta área.

El objetivo de la SLR es analizar sistemáticamente la literatura sobre la relación entre las perspectivas de Minería de Procesos en el contexto de la Ingeniería de Software, con el fin de trazar cuáles perspectivas se han definido, cuáles se han implementado en las fases del proceso de desarrollo de software y cuáles aplicaciones o implicaciones en la práctica se han identificado. La SLR se centra en identificar la situación en que se incorporan las perspectivas al desarrollo de software con la finalidad de analizar los desafíos y oportunidades en el campo del software para futuras prácticas.

Se desarrollaron las tres fases principales de la metodología de SLR (Kitchenham, 2007; Kitchenham, 2004): 1) Planificación, 2) Ejecución y 3) Análisis de resultados, las cuales están representados en la Figura 6. Cada fase cuenta con actividades propias y no son completamente secuenciales. Al final, se obtiene un informe o publicación de resultados sobre los Estudios Primarios (EP) del tema de interés.

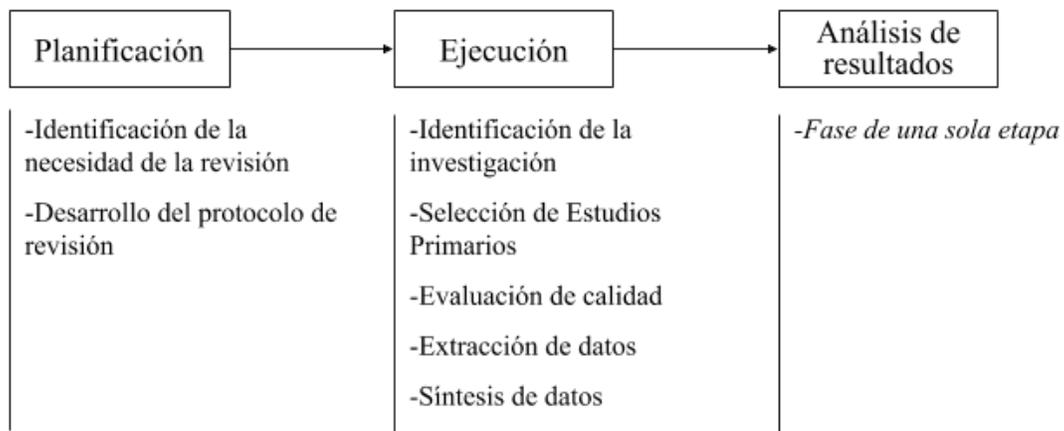


Figura 6. Diagrama del protocolo de la Revisión Sistemática de la Literatura.

2.1 Planificación

Para el desarrollo del protocolo de investigación se definieron los siguientes puntos: a) Preguntas de Investigación, b) Selección de fuentes, c) Cadena de búsqueda y d) Criterios de selección de inclusión y exclusión, los cuales se describen a continuación.

2.1.1 Preguntas de Investigación

Se establecieron dos Preguntas de Investigación (RQ) basadas en la necesidad de la revisión.

- RQ1: ¿Cuáles perspectivas de la Minería de Procesos se han definido y su definición en el contexto de la Ingeniería de Software?
- RQ2: ¿Cuáles son las aplicaciones prácticas dónde se han implementado perspectivas de Minería de Procesos en la disciplina de Ingeniería de Software?

La RQ1 está orientada a identificar las perspectivas existentes que han sido definidas para minar los procesos en el contexto de la Ingeniería de Software. Comprender sus definiciones y tipos permitirá a los practicantes de software conocer cuáles perspectivas se pueden aplicar en ciertos contextos. La ausencia de una definición o nomenclatura podría generar una brecha conceptual o de marco teórico en el empleo de la Minería de Procesos.

La RQ2 tiene como motivación conocer cuáles son las aplicaciones prácticas de las perspectivas en los proyectos de software, situando la fase o contexto. Asimismo, analizar los desafíos y oportunidades en la disciplina de Ingeniería de Software para futuras prácticas.

2.1.2 Selección de fuentes

Se seleccionaron cinco fuentes de datos considerando que reúnen la cobertura para llevar a cabo la SLR en el área de Ciencias de la Computación como se muestra en (Dong et al., 2017; Erdem et al., 2018; dos Santos Garcia et al., 2019; Ibarra-Esquer et al., 2017). Las fuentes fueron: 1) ACM Digital Library, 2) IEEE Xplore, 3) ScienceDirect, 4) Scopus y 5) Springer.

2.1.3 Cadena de búsqueda

Para obtener la mayor cantidad de estudios, se definió una cadena de búsqueda utilizando una combinación de los términos clave relacionados con las RQ. La cadena emplea las perspectivas mencionadas en (Van der Aalst, 2016), así como la combinación de Minería de Procesos junto con Ingeniería de Software, con el fin de encontrar los trabajos que estén relacionados con: 1) Process Mining, 2) Software Engineering, 3) Control-flow perspective, 4) Organizational perspective, 5) Case perspective y 6) Time perspective. La cadena de búsqueda fue adaptada para ajustarse a las especificaciones de cada motor de búsqueda.

("Process Mining" OR "Process Mining Perspective" OR "Software Engineering") AND ("Control-flow perspective" OR "Organizational perspective" OR "Case perspective" OR "Time perspective")

2.1.4 Criterios de Inclusión y Exclusión

Se incluyeron los artículos que cumplieron con los siguientes criterios:

- Estudios escritos en el idioma inglés.
- Estudios publicados entre el periodo del 2010 al 2020.
- Estudios que contengan al menos la palabra "Process Mining - Minería de Procesos" en el título.
- Estudios relacionados con la Ingeniería de Software.
- Estudios que contengan al menos la mención y/o aplicación de alguna perspectiva de Minería de Procesos.

Artículos con los siguientes criterios fueron excluidos:

- Estudios que no presenten perspectivas de Minería de Procesos.
- Estudios repetidos o que no utilice en aplicaciones prácticas las perspectivas de Minería de Procesos con Ingeniería de Software.
- Estudios que no se encuentren disponibles en texto completo.

2.2 Ejecución

Para esta fase se plantearon las siguientes etapas: 1) Identificación y selección de estudios primarios, 2) Criterios de calidad y 3) Extracción y síntesis de datos.

2.2.1 Identificación y Selección de Estudios Primarios

La selección de los estudios se realizó por medio de cinco pasos como se muestra en la Figura 7. Durante la aplicación del proceso, en los pasos 1 y 2, las fuentes se enumeraron (valor de la fuente F1 hasta n donde $n=5$) y se adaptó la cadena de búsqueda con respecto a cada una de ellas para extraer un conjunto completo de consultas. Como resultado se obtuvo la primera selección de estudios con un total de 145 estudios. En el paso 3 se aplicaron los criterios de inclusión y exclusión predefinidos tomando en cuenta el título, abstract (resumen) y palabras claves. De tal manera, que en el paso 4 se obtuvo una segunda selección reduciendo a 21 estudios. Finalmente, en la Tabla 2 se muestran los resultados del paso 5 al aplicarse los criterios de calidad obteniendo la selección final de 12 Estudios Primarios (EP).

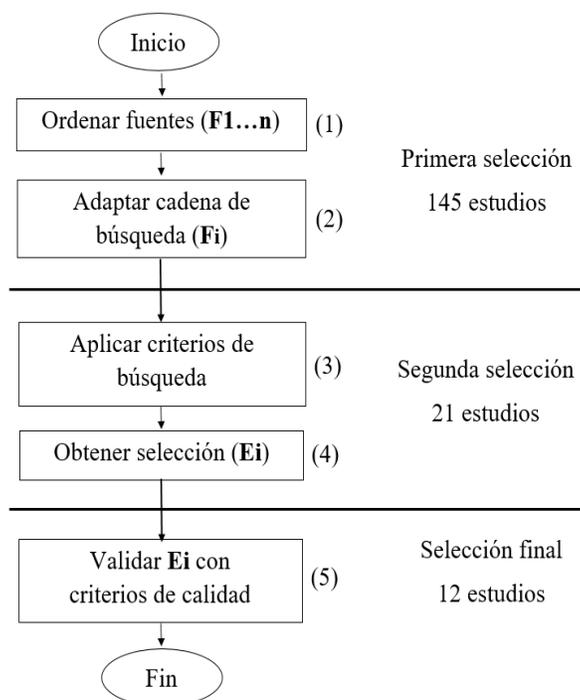


Figura 7. Proceso de selección de Estudios Primarios

Tabla 2. Resultados del proceso de selección

Fuente	Primera	Segunda	Final
ACM Digital Library	6	4	2
IEEE Xplore	14	3	2
ScienceDirect	26	2	1
Scopus	8	4	2
Springer	91	8	5
TOTAL	145	21	12

2.2.2 Criterios de calidad

En la Tabla 3 se muestran los criterios utilizados, así como la ponderación recibida por cada EP para la evaluación de la calidad, los cuales se retomaron de (Jovanović et al., 2020) para adaptarse a las dos RQ. Cada criterio se calificó entre 0 y 1 de acuerdo al nivel de cumplimiento. Se consideraron sólo aquellos estudios que obtenían una calificación entre 8-10 (Jovanović et al., 2020). Los autores tomaron el conjunto de la segunda selección para la evaluación de calidad, lo que condujo de los 21 estudios preseleccionados, solo 12 se clasificaron como EP (Tabla 4).

Tabla 3. Criterios de calidad por cada Estudio Primario

No.	Criterio	EP1	EP2	EP3	EP4	EP5	EP6	EP7	EP8	EP9	EP10	EP11	EP12
C1	¿El objetivo de la investigación está suficientemente explicado y bien motivado?	1	1	1	1	1	1	1	1	1	1	1	1
C2	¿Está el contexto de estudio claramente establecido?	1	1	1	1	1	1	1	1	1	1	1	1
C3	¿El diseño de la investigación está suficientemente preparado?	1	1	1	1	1	1	1	1	1	1	1	1
C4	¿Se describen adecuadamente la recopilación	1	1	1	1	1	1	1	1	1	1	1	1

	de datos y las medidas?												
C5	¿Son las medidas utilizadas en el estudio las más relevantes para responder la pregunta de investigación ?	1	1	1	1	1	1	1	1	1	1	1	1
C6	¿Se describe adecuadamente el análisis de datos utilizado en el estudio?	1	1	1	1	1	1	1	1	1	1	1	1
C7	¿Se describe claramente la interpretación de la evidencia?	1	1	1	1	1	1	1	1	1	1	1	1
C8	¿Se consideran y discuten las posibles explicaciones alternativas en el análisis?	1	1	1	0	0	1	1	1	0	0	1	1
C9	¿Los resultados del estudio están claramente establecidos y respaldados por los resultados?	1	1	1	1	1	1	1	1	1	1	1	1
C10	¿El documento discute limitaciones o validez?	0	1	0	1	1	1	0	0	1	0	0	1
	TOTAL	9	10	9	9	9	10	9	9	9	8	9	10

Tabla 4. Resultados del proceso de selección

Fuente	Selección			
	Primera	Segunda	Estudios Primarios Finales	
ACM Digital Library	6	4	2	(Gupta et al., 2014; Gupta, 2014)
IEEE Xplore	14	3	2	(Lemos et al., 2014; Sebu y Ciocarlie, 2014)
ScienceDirect	26	2	1	(dos Santos Garcia et al., 2019)
Scopus	8	4	2	(Samalikova et al., 2014; Caldeira et al., 2019)
Springer	91	8	5	(Li et al., 2011; Ailenei et al., 2012; Zhao y Zhao, 2014; Kalenkova et al., 2017; Bernardi et al., 2018)
TOTAL	145	21	12	

2.2.3 Extracción de información

Se utilizó como formato la síntesis de datos de los EP la Tabla 5, en la cual se registraron y analizaron bajo los siguientes indicadores: Código (numeración de identificación del EP), año (en que fue publicado el documento), título y autor(es).

Tabla 5. Listado de Estudios Primarios

Código	Año	Título	Autor(es)
EP1	2014	Process Mining Multiple Repositories for Software Defect Resolution from Control and Organizational Perspective.	M. Gupta et al.
EP2	2014	Nirikshan: Process Mining Software Repositories to Identify Inefficiencies, Imperfections, and Enhance Existing Process Capabilities	M. Gupta
EP3	2011	Using Process Mining in Software Development Process Management: A Case Study	A. M. Lemos et al.
EP4	2014	Applied process mining in software development: Case Study	M. L. Sebu y H. Ciocârlie
EP5	2019	Process mining techniques and applications –A systematic mapping study	C. dos Santos Garcia et al.
EP6	2014	Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice	J. Samalikova et al.
EP7	2019	Assessing Software Development Teams' Efficiency using Process Mining	J. Caldeira et al.

EP8	2011	A process mining based approach to knowledge maintenance	Ming Li, Lu Liu, Lu Yin y Yanqiu Zhu
EP9	2012	Definition and Validation of Process Mining Use Cases	I. Ailenei et al.
EP10	2014	Process Mining from the Organizational Perspective	W. Zhao y X. Zhao
EP11	2017	Process mining using BPMN: relating event logs and process models	A. A. Kalenkova et al.
EP12	2018	A systematic approach for performance assessment using process mining: An industrial experience report	S. Bernardi et al.

2.3 Análisis de resultados

Los resultados obtenidos se presentan con base a las RQ1 y RQ2 y las motivaciones que las definieron; de igual forma, se exponen con respecto a los aspectos identificados de los EP.

La Tabla 6 presenta la distribución de los EP por año de publicación, observando que el año 2014 presentó el mayor número de EP (5) y se establece una pausa en 2015 y 2016, volviéndose a presentar publicaciones de 2017 a 2019. Según el tipo de publicación, el 50% de los EP corresponden a revistas, mientras que el resto se publica en congresos. En cuanto al país de origen y el método de investigación de estudio de caso se encontró que Brasil (EP3 y EP5), China (EP8 y EP10), India (EP1 y EP2) y Holanda (EP6 y EP9) presentaron dos estudios de caso. Por otro lado, los países en los que sólo se identificó un EP fueron España (EP12), Portugal (EP7), Rumanía (EP4) y Rusia (EP11).

Tabla 6. Publicaciones por año

Año	2011	2012	2013	2014	2015	2016	2017	2018	2019
Frecuencia	2	1	0	5	0	0	1	1	2

Para facilitar la exposición de los resultados obtenidos, la información se presenta como subsecciones de acuerdo a: 1) Definiciones de perspectivas de Minería de Procesos y sus contextos de aplicación, 2) Aportación de las perspectivas de Minería de Procesos a la Ingeniería de Software e 3) Implementación de perspectivas de Minería de Procesos en estudio de casos.

2.3.1 Identificación y definición de perspectivas de Minería de Procesos

Durante el análisis de los EP seleccionados, se identificaron cuatro perspectivas: 1) Flujo de control, 2) Organizacional, 3) Caso y 4) Tiempo, las cuales se presentan en la Tabla 7. Con respecto a los nombres de las perspectivas, la perspectiva del tiempo se menciona como perspectiva de desempeño en EP6 y la perspectiva de flujo de control también se denomina perspectiva de proceso en EP1. Las definiciones de las perspectivas se clasificaron en tres tipos: 1) Definición propuesta por los autores; 2) Definición citada en el estudio; y 3) Mención del concepto sin definición.

Tabla 7. Definición de perspectivas de Minería de Procesos

Perspectiva	Definición/Concepto	Estudio
Flujo de Control (del proceso)	Identifica las actividades realizadas y el orden de su ejecución.	EP1
	Identifica todas las rutas posibles que un elemento podría seguir para alcanzar un estado cercano.	EP4
	Define un enfoque que consiste en analizar cómo cada tarea/actividad se sigue entre sí en un registro de eventos, e infiere un posible modelo para el comportamiento capturado en el proceso observado.	EP7
	Encuentra la dependencia entre las tareas mediante el análisis de las secuencias de eventos observadas en el registro.	EP8
Organizacional	Identifica los roles organizacionales, la distribución del trabajo entre los ejecutantes (recursos) y las actividades que puede ejecutar un recurso en particular.	EP1
	Verifica cuáles son los recursos involucrados en las actividades y cómo interactúan.	EP4
	Extrae estructuras organizacionales mediante la combinación de análisis de redes sociales, mapeo de comportamientos de recursos, colaboración de usuarios y análisis de roles mediante la mejora del modelo de proceso.	EP5
	Agrupar a los encargados de las tareas y encuentra las relaciones entre los grupos.	EP8
Caso	Analiza comportamientos específicos debido a valores específicos de los atributos.	EP4
Tiempo (del desempeño)	Identifica la causa del retraso del proceso actual.	EP1

1) Definición propuesta por los autores: Esta clasificación identifica los EP que presentan una definición de las perspectivas que no está relacionada con la cita de otros autores; también evalúa aquellos que presentan una definición orientada a la explicación práctica de la perspectiva de Minería de Procesos dentro del contexto de la Ingeniería de Software. Se encontraron cinco EP (41.6%) que cumplen con la definición conceptual de las perspectivas. Los estudios EP5, EP7, EP8 describen sólo una perspectiva, por otro lado, EP1 y EP4 muestran la definición de perspectivas como la de flujo de control, organizacional, del tiempo y del caso. En el marco de las cinco EP, la perspectiva organizacional es la más descrita por EP1, EP4, EP5 y EP8, la perspectiva de flujo de control sólo se menciona en EP1, EP4 y EP7 y las perspectivas de caso y tiempo se incorporan con su definición en una solo estudio, EP4 y EP1 respectivamente. Un aspecto a destacar es el consenso en la definición de la perspectiva organizacional y la perspectiva de flujo de control.

2) Definición citada en el estudio: Esta clasificación incluye aquellos EP que presentan una definición citada de otras investigaciones y aquellos autores que han proporcionado una definición de las perspectivas. Los estudios son EP6 y EP10. EP6 proporciona la definición de la perspectiva de flujo de control retomada de (de Medeiros et al., 2007), la perspectiva de desempeño de (Van Dongen et al., 2005), la perspectiva organizativa de (Van der Aalst et al., 2005) y la perspectiva de caso de (Samalikova et al., 2011) y (Rozinat y Van der Aalst, 2008). Por último, EP10 presenta las perspectivas de flujo de control, organizacional y del caso citadas en (Van der Aalst, 2016).

3) Mención del concepto sin definición: Esta clasificación incorpora EP que sólo hacen referencia a perspectivas sin el acompañamiento de una descripción. Se han identificado un total de cinco EP (41.6%) que mencionan las perspectivas flujo de control (EP3, EP11), organizacional (EP2, EP3, EP9), caso (EP9) y tiempo (EP9, EP12). Existe una particularidad de EP8 en la que se identifica la perspectiva de flujo de control, pero también se describe la perspectiva organizacional, por lo que se agrupó en la primera clasificación.

Respecto a la última clasificación "Mención", los resultados aportan evidencias de identificación de perspectivas de Minería de Procesos pero sin definir las por lo que su interpretación en proyectos de desarrollo o procesos de software podría darse en las siguientes situaciones: (1) no

se considera importante o necesario definir las, (2) son desconocidas o no entendidas conceptualmente por los implicados, o (3) no tienen el mismo significado para todos dentro de una problemática concreta.

2.3.2 Implementación de perspectivas de Minería de Procesos en estudios de caso

Del total de EP, sólo cuatro (33.3%) tenían una aplicación práctica EP1, EP2, EP3 y EP7, de los cuales EP1, EP3 y EP7 aplicaban la perspectiva de flujo de control. Además, EP1 aplicó la perspectiva organizacional, al igual que EP2. Existe una gran variedad de herramientas comerciales y gratuitas para la Minería de Procesos (Wang et al., 2012), por lo que fue importante identificarlas (en el 75% de los casos se aplicó la herramienta ProM). Las perspectivas de Minería de Procesos que se manejaron con las herramientas ProM y Disco fueron la de flujo de control y organizacional. Asimismo, la Tabla 8 concentra detalles más específicos de los casos de estudio, como las perspectivas aplicadas, el tipo de repositorio en el que se aplicó la minería, el tipo de extracción o pre-procesamiento de datos y las propiedades del registro de eventos (estructura, número de registros utilizados y/o antigüedad). A continuación, se detalla la aplicación práctica de cada EP:

Estudio de caso EP1: Se utilizaron datos del proyecto Google Chromium aplicando el minado a tres repositorios de software: el sistema de seguimiento de issues (Google ITS), el sistema de revisión por pares (Rietveld) y el sistema de control de versiones (Subversion). Los datos se extrajeron mediante JSON-RPC y XML-RPC, se integraron (los sistemas de información no estaban vinculados explícitamente) y se almacenaron en una base de datos MySQL. Se generó un registro de eventos de un año con 9744 eventos con cuatro atributos: caseID, timestamp, actividad y actor. El estudio se centró en el flujo de control y la perspectiva organizacional. El resultado fue la identificación de cuellos de botella, la definición y detección de antipatronos básicos y compuestos por parte de la perspectiva de flujo de control, y el descubrimiento de métricas como la entrega de trabajo, la subcontratación, los casos conjuntos y las actividades conjuntas por parte de la perspectiva organizacional (Gupta et al., 2014). Como se muestra en la Tabla 8, se utilizó la herramienta Disco para descubrir el modelo de proceso en tiempo de ejecución.

Estudio de caso EP2: Se propone un marco de investigación denominado Nirikshan con la funcionalidad de procesar datos de repositorios de software utilizando una perspectiva organizacional, con el objetivo de identificar ineficiencias e imperfecciones en la modelización del proceso en tiempo de ejecución (realidad) descubiertas mediante un registro de eventos. La perspectiva organizacional se aplicó para identificar los roles organizativos, los patrones de interacción y la distribución del trabajo entre los ejecutantes (recursos) y las actividades ejecutadas por un recurso concreto. El enfoque propuesto consiste en extraer datos de procesos de repositorios de software como el sistema de seguimiento de issues (por ejemplo, Bugzilla, JIRA), el sistema de revisión por pares (Rietveld, Gerrit), el sistema de control de versiones (Mercurial, Git), las plataformas de alojamiento de proyectos (source- forge) y los sitios web de QnA. También, la extracción de datos a través de API (si están disponibles), XML-RPC, JSON-RPC, etc. Como resultados se menciona un estudio de caso sobre datos extraídos de Bugzilla ITS del navegador de código abierto Firefox y el proyecto Core. El registro de eventos constaba de 12234 informes de problemas de Firefox y 24253 informes de problemas de Core (Gupta, 2014).

Estudio de caso PS3: Se aplicó la perspectiva de flujo de control explorando un registro de eventos del proceso de desarrollo de software de una empresa brasileña, obteniendo información de los últimos cinco años de ejecución de más de 2000 instancias de proceso. El registro de eventos fue pre-procesado utilizando los informes producidos por la herramienta ProM. Los campos que formaron el registro de eventos son: taskID, tipo de evento, timestamp y originador, permitiendo el descubrimiento de la perspectiva del flujo de control (Lemos et al., 2011). La Minería de Procesos se implementó para la verificación de la conformidad. Como resultado, se produjo un conjunto de informes indicando las inconsistencias detectadas entre un modelo de proceso y su correspondiente registro de ejecución, demostrando la utilidad de la Minería de Procesos como herramienta para la gestión de procesos de desarrollo de software.

Estudio de caso PS7: Se realizó una evaluación de la eficiencia de equipos de desarrollo de software aplicando la perspectiva de flujo de control. El estudio consistió en 8 equipos de programadores a los que se asignó la misma tarea de calidad de software. Los registros de eventos se almacenaron en una base de datos para convertirlos a formato XES e importarlos a la herramienta ProM para su análisis mediante la perspectiva de flujo de control. Las propiedades

del registro de eventos incluían el elemento del equipo como identificador del caso, categoryName y command- Name, marcas de tiempo para el inicio y fin de las actividades y otras propiedades como el recurso en el proceso (Caldeira et al., 2019). Los modelos resultantes se compararon a través de la perspectiva de flujo de control con un modelo de referencia de mejores prácticas obteniendo una verificación de conformidad de los modelos de proceso menos complejos.

Tabla 8. Caracterización de las perspectivas de la Minería de Procesos en casos prácticos

EP	Perspectivas	Repositorios	Extracción/ Pre- procesamiento	Herramienta	Propiedades del registro de eventos
EP1	Flujo de control Organizacional	Issue Tracking System (Google ITS), Peer Code Review (Rietveld), Version Control System (Subversion)	JSON-RPC XML-RPC	Disco	Estructurado con los atributos caseID, timestamp, actividad y actor.
EP2	Organizacional	Issue Tracking System (Bugzilla)	API (si es disponible), XML-RPC, JSON-RPC	ProM, Disco	La estructura del registro de eventos no se presenta. Conformado por 12,234 (Firefox) y 24,253 (Core) reporte de problemas.
EP3	Flujo de control	No establecido	Pre- procesamiento mediante informes elaborados por la herramienta de Minería de Procesos	ProM	Estructurado con los atributos taskID, tipo evento, timestamp y originador. Registro real con más de 2000 instancias del proceso.
EP7	Flujo de control	No establecido	Almacenamiento de eventos en la base de datos para su conversión al formato eXtensible Event Stream (XES)	ProM	Estructurado con los atributos elementos de equipo (Identificador de caso), categoryName, commandName, timestamps.

2.3.3 Perspectivas de Minería de Procesos en la Ingeniería de Software

Para la aplicación práctica de la Minería de Procesos en Ingeniería de Software, se consideraron los tópicos de las Áreas de Conocimiento (Software Engineering Education Knowledge - SEEK) (LeBlanc et al., 2014). De esta forma, cuatro tópicos de la Ingeniería de Software fueron apoyados por los resultados obtenidos de las perspectivas de la Minería de Procesos (Tabla 9). La aplicación práctica en EP1 y EP2 se orientó principalmente a Métricas y su Gestión y Herramientas y Técnicas de Diseño respectivamente; mientras que EP3 estaba relacionada con la categoría de Gestión de Procesos de Software y EP7 con la Práctica Profesional de Ingeniería de Software.

Tabla 9. Áreas de aplicación de perspectivas de MP

Áreas de aplicación	Estudio
Resolución de problemas de software	EP1
Repositorios de software	EP2
Calidad de software	EP3
Eficiencia en los equipos de desarrollo de software	EP7

Tomando los resultados y contribuciones de los EP de la sección anterior, se identificaron las aportaciones de las perspectivas de Minería de Procesos a la Ingeniería de Software. Las perspectivas se aplicaron para detallar los procesos de software analizados y se representan en la Tabla 10 junto con el estudio de donde surge la aportación.

La perspectiva de flujo de control se implementó para la definición de patrones, orden de ejecución de tareas, comparación de las frecuencias de las actividades contra la mejor práctica de un proceso y el hallazgo de tareas adicionales del proceso. Por parte de la perspectiva organizacional, se encuentra la obtención de gráficas de interacción social de los roles, identificación de roles y la obtención de un modelo de organización de mantenimiento del conocimiento. En la perspectiva del tiempo solo se observa la identificación de cuellos de botella, frecuencia de eventos y trazas únicas.

Tabla 10. Aportación de las perspectivas de Minería de Procesos en la Ingeniería de Software

Perspectiva	Aportación	Estudio
Flujo de Control (del proceso)	Definir patrones para el proceso y detectar su existencia en el modelo de proceso de tiempo de ejecución.	EP1
	Descubrir cuáles tareas se realizan realmente y su orden de ejecución	EP3
	Comparar variantes de proceso menos complejas con el proceso de mejor práctica en función de las frecuencias de las actividades.	EP7
	Encontrar tareas adicionales e ineficientes y las dependencias de las tareas.	EP8
Organizacional	Construir redes sociales basadas en relaciones como la causalidad (entrega y subcontratación de trabajo), actividades conjuntas y casos conjuntos.	EP1
	Identificar roles organizacionales, patrones de interacción y distribución del trabajo entre los ejecutantes (recursos) y las actividades ejecutadas por un recurso en particular.	EP2
	Obtener un modelo de organización de mantenimiento del conocimiento.	EP8
Tiempo	Identificar cuellos de botella, frecuencia de eventos y trazas únicas.	EP1

La Minería de Procesos y la Ingeniería de Software son dos disciplinas que pueden realizar una completa integración teórica y práctica para formar un enfoque transdisciplinario donde cada una aporta elementos para la comprensión de un problema unificado. Esta situación permite el uso de perspectivas de Minería de Procesos dentro de la Ingeniería de Software, que de acuerdo con la presente SLR, se confirma como un campo de investigación en desarrollo, resultado de investigaciones que hasta el momento han demostrado evidencias hacia su definición y uso.

En cuanto a las definiciones, constituye una visión desde el punto de vista práctico; en general, las definiciones pueden parecer similares a las encontradas, por lo que es posible identificar un consenso entre los autores que puede utilizarse para el establecimiento formal y completo de una definición, logrando formar el interés de los profesionales del software en el uso de las perspectivas de Minería de Procesos.

En lo que se refiere a las aplicaciones prácticas, la visión de la transdisciplinariedad de los elementos de las perspectivas de Minería de Procesos en la Ingeniería de Software está representada en los estudios de casos mencionados. En estos estudios, las perspectivas de Minería

de Procesos fueron un factor de influencia positiva en la caracterización de los procesos analizados. Su uso, por el momento, no se centra en ninguna etapa específica del desarrollo de software, pero los esfuerzos logran validar sus propuestas en áreas como Métricas, Gestión, Herramientas y Técnicas de Diseño y Práctica Profesional de Ingeniería de Software. La perspectiva con mayor implantación es la de flujo de control, seguida de la organizacional, observándose menor frecuencia de estudio hacia las perspectivas de caso y tiempo en comparación con las dos primeras. Observando como reto el aplicar la Minería de Procesos al análisis de las perspectivas menos destacadas (caso y tiempo) y obtener así un modelo integrado con las perspectivas.

Los resultados de la SLR sitúan a las perspectivas de Minería de Procesos en la Ingeniería del Software y el impacto positivo que tienen en sus aportaciones. Destaca la importancia del trabajo constante a realizar para mejorar los resultados en las aplicaciones de las perspectivas de Minería de Procesos. También se distingue por observar su evolución de manera conceptual y la oportunidad de establecer definiciones formales con la incorporación de diferentes términos a medida que la Minería de Procesos se desarrolla en la Ingeniería de Software. De esta forma, las definiciones serán ampliadas y complementadas con expresiones propias dentro del dominio de la Minería de Procesos y la Ingeniería de Software.

Capítulo 3. Marco Teórico

Este capítulo se adentra en metodologías propuestas y utilizadas en el ámbito de la Minería de Procesos, como lo son el Modelo de Ciclo de Vida L*, la Metodología PM², *Process Comparison Methodology* (PCM) y *Process Diagnostics Method* (PDM). Cada una de estas metodologías aporta enfoques dependiendo de la estructura de los procesos para el análisis. Asimismo, se introducirán las notaciones formales de los elementos de la Minería de Procesos, tales como, eventos, casos, trazas, registros de eventos y redes de Petri con el fin de contextualizar las definiciones en un ámbito matemático. Estas definiciones formales son esenciales para comprender los procesos y eventos al momento del minado.

3.1 Metodologías de Minería de Procesos

Con la creciente investigación de la aplicación de la Minería de Procesos, se han expuesto varias metodologías en la literatura, entre ellas se encuentra PDM (Bozkaya et al., 2009), el Modelo de ciclo de vida L* (Van der Aalst, 2016), la Metodología de Minería de Procesos PM² (Van Eck et al., 2015), y la Metodología PCM (Syamsiyah et al., 2017).

3.1.1 Modelo de ciclo de vida L*

El Modelo de ciclo de vida L* (Van der Aalst, 2016) describe los ciclos de vida de un proyecto típico de Minería de Procesos e interviene con temas más amplios como la mejora de procesos y el apoyo operativo para procesos estructurados. Esta metodología incorpora el enfoque para llegar a un modelo totalmente integrado que cubra la perspectiva organizacional, del tiempo y del caso (Van der Aalst, 2016). Se divide en 4 etapas las cuales son:

- 1) **Extraer:** Una vez iniciado el proyecto, se extraen datos esenciales para el proyecto de Minería de Procesos, incluyendo eventos, modelos, objetivos y preguntas. Se transforman datos brutos en registros de eventos aptos para el análisis.
- 2) **Crear un modelo de flujo de control y conectar el registro de eventos:** Se crea un modelo de flujo de control que refleja el proceso real. Los eventos se relacionan con las actividades del modelo.

- 3) **Crear un modelo de proceso integrado:** El modelo se enriquece con perspectivas adicionales (organizacional, de caso y temporal) para comprender mejor el proceso y realizar simulaciones. Se utilizan para responder preguntas y guiar acciones.
- 4) **Soporte operativo:** Esta etapa se centra en detectar, predecir y recomendar acciones en tiempo real para apoyar la toma de decisiones y la mejora continua de los procesos operativos.

El Modelo de Ciclo de Vida L* proporciona un marco sólido para la Minería de Procesos, desde la extracción inicial de datos hasta el apoyo operativo continuo, permitiendo la comprensión, optimización y mejora de los procesos empresariales.

3.1.2 Metodología PM2

La Metodología de Minería de Procesos PM² se diseñó para dar soporte a proyectos cuyo objetivo es mejorar el rendimiento de los procesos o el cumplimiento de las normas y reglamentos. Abarca una amplia gama de técnicas de Minería de Procesos y otras técnicas de análisis, y es adecuada para el análisis de procesos tanto estructurados como no estructurados (Van Eck et al., 2015). Esta metodología consta de 6 etapas:

- 1) **Planeación:** En esta etapa, se establecen los objetivos del proyecto, se seleccionan los procesos a analizar y determinar las preguntas de investigación.
- 2) **Extracción:** Se obtienen datos de eventos y, depende la información disponible, modelos de procesos relevantes para el análisis.
- 3) **Procesamiento de datos:** Los datos de eventos se procesan y transforman en registros de eventos adecuados para el análisis.
- 4) **Minado y análisis:** Se aplican técnicas de extracción de procesos a los registros de eventos para responder preguntas y obtener información sobre el rendimiento y conformidad.
- 5) **Evaluación:** Los resultados del análisis se relacionan con ideas de mejora y nuevas preguntas de investigación.
- 6) **Mejora de procesos y soporte:** Se implementan las mejoras identificadas en la evaluación para modificar la ejecución real del proceso y lograr una mejora continua.

La metodología PM² representa un enfoque estructurado para abordar la Minería de Procesos. Desde la identificación de preguntas de investigación hasta la aplicación de técnicas de análisis, PM² permite una visión completa de los procesos. Además, su enfoque en la mejora continua y el apoyo operativo asegura que los beneficios de la Minería de Procesos se traduzcan en acciones concretas y mejoras sostenibles.

3.1.3 Metodología PCM

El Método de Comparación de Procesos (PCM) se centra en el análisis de múltiples procesos. Esta metodología considera múltiples perspectivas, como el flujo de control, la organización, los datos, el rendimiento, entre otros (Syamsiyah et al., 2017). PCM se divide en 5 fases:

- 1) **Pre-procesamiento de datos:** En esta fase, se convierten los datos brutos en formatos de registro de eventos estandarizados para facilitar el análisis de Minería de Procesos. Los objetivos principales son refinar los datos de eventos y crear un registro de eventos con un conjunto de atributos de caso definidos.
- 2) **Análisis de alcance:** En esta etapa, se delimita el alcance del análisis, limitando el número de comparaciones futuras. La delimitación se basa en atributos, como los valores más frecuentes. El objetivo es reducir la complejidad del análisis y enfocarse en los aspectos más relevantes.
- 3) **Identificación de sub-registros comparables:** En esta fase, se identifican los sub-registros comparables, es decir, variantes del proceso que tienen suficientes similitudes para ser comparadas. Esto implica seleccionar conjuntos de variantes que tengan puntos en común y sean relevantes para el análisis.
- 4) **Comparación exhaustiva:** Una vez identificados los conjuntos de sub-registros comparables, se lleva a cabo una comparación detallada entre ellos. Se analizan aspectos como el flujo de control, el rendimiento, la utilización de recursos y el cumplimiento. El resultado es una colección de diferencias destacadas entre los sub-registros de entrada.
- 5) **Interpretación y validación:** En la última fase, se presentan e interpretan los resultados del análisis de Minería de Procesos. Los resultados se comunican al responsable del proceso y se validan con los participantes. La interpretación y validación son esenciales para asegurar que los resultados sean relevantes y útiles a nivel empresarial.

Estas etapas en conjunto forman la metodología PCM, que se centra en el análisis comparativo de múltiples procesos desde diversas perspectivas para identificar diferencias significativas y proporcionar información valiosa para la toma de decisiones empresariales.

3.1.4 Metodología PDM

El Método de Diagnóstico de Procesos (PDM) incorpora una amplia visión de los procesos de la organización en un corto período de tiempo destacando varias perspectivas del proceso. El resultado abarca la perspectiva del flujo de control, la perspectiva del rendimiento y la perspectiva organizacional (Bozkaya et al., 2009). PDM está compuesta por 6 fases:

- 1) **Preparación del registro:** En esta fase inicial, se preparan los registros de eventos para su posterior análisis. Esto implica la extracción y transformación de los datos brutos de eventos de sistemas de información en un formato adecuado para la Minería de Procesos. Se abordan cuestiones como la selección de la noción de caso y la identificación de actividades y sus eventos.
- 2) **Inspección del registro:** En esta fase, se realiza una inspección detallada del registro de eventos para comprender su estructura y características. Se recopilan estadísticas clave, como el número de casos, roles, eventos y otros aspectos relacionados con el tamaño y la complejidad del registro.
- 3) **Análisis de flujo de control:** Esta fase se centra en analizar el flujo de control del proceso, respondiendo a la pregunta de cómo se desarrolla el proceso en la práctica. Se verifica si el proceso se ajusta a una descripción existente y se identifican comportamientos inusuales o infrecuentes.
- 4) **Análisis de rendimiento:** Una vez comprendido el flujo de control, se utilizan los modelos de proceso para analizar el rendimiento. Se buscan posibles cuellos de botella o ineficiencias en el proceso.
- 5) **Análisis de roles:** Si el registro de eventos incluye información sobre quién ejecutó cada evento, se analizan los roles y las interacciones entre ellos en el proceso.
- 6) **Transferencia de resultados:** El objetivo de la metodología PDM es obtener información valiosa sobre los procesos de la organización y su relación con los sistemas de información. Los resultados revelan el comportamiento real observado en

el sistema, que a menudo difiere del proceso previsto debido a comportamientos no deseados o adaptaciones realizadas por los usuarios.

La metodología PDM se enfoca en obtener una visión integral de los procesos organizacionales, abarcando aspectos como el flujo de control, el rendimiento y los roles involucrados, mediante un proceso de análisis estructurado y detallado de registros de eventos.

3.1.5 Comparativa de las metodologías

Estas metodologías ofrecen enfoques diferentes para el análisis de procesos y su mejora, abordando diferentes aspectos y características. Por ejemplo, la metodología PDM se diferencia de L* y PM² en términos de su enfoque y alcance en la Minería de Procesos. A diferencia de L* y PM², que abarcan un amplio conjunto de técnicas de Minería de Procesos, PDM se centra en un conjunto más reducido de técnicas específicas. Por otro lado, PDM, L* y PM² realizan el análisis de un solo proceso, mientras que PCM analiza la comparación de múltiples procesos. Por último, la L* va dirigida a procesos estructurados y PM² es adecuada para el análisis de procesos tanto estructurados como no estructurados.

En la elección de una metodología de Minería de Procesos influyen diversos factores, como los objetivos del proyecto, la complejidad del proceso, la calidad de los datos y los recursos disponibles. Cada metodología tiene sus propias fortalezas y enfoques específicos, lo que significa que no existe una metodología única que sirva para todos los casos. Por lo tanto, es esencial realizar una evaluación y considerar estos factores antes de tomar una decisión.

3.2 Notación

Dentro de las etapas de preparación de los datos, es esencial identificar elementos clave del proceso, tales como eventos, casos, trazas, entre otros. Estos elementos han sido formalmente definidos a través de notaciones matemáticas que incluyen las relaciones entre ellos, como la relación entre eventos, casos específicos y actividades (Van der Aalst, 2016).

A continuación, se muestran las definiciones de evento, caso, traza y registro de eventos. Luego, se presenta la notación de red Petri para el modelado de los procesos. Se tomaron en cuenta las relaciones presentes en la Figura 8.

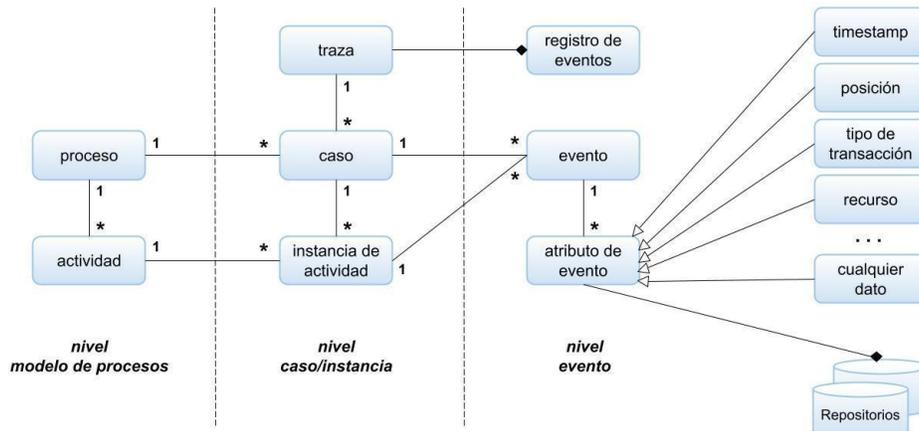


Figura 8. Representación de los elementos de Minería de Procesos.
Adaptada de Van der Aalst, 2016.

3.2.1 Definición de Evento

Un proceso describe la ejecución de conjunto de actividades; cada actividad ejecutada da lugar a uno o más eventos que registran información que describe la instancia del proceso, la actividad ejecutada, el tiempo de ejecución y datos adicionales como los atributos y valores involucrados (Van Eck et al., 2022).

Tabla 11. Comparativa de las definiciones y notaciones de evento

Autor	Definición y representación formal
(De Leoni et al., 2016)	Un evento e es una asignación de valores a características, es decir, $e \in C \rightarrow U$. En el resto $\varepsilon = C \rightarrow U$ es el universo de eventos.
(Diamantini et al., 2016)	Un evento σ es una instancia de una especificación de entidad, y se identifica por un identificador único. Denotaremos por Σ el universo de eventos, es decir, el conjunto de todos los posibles identificadores de eventos. Cada evento se describe mediante un conjunto de atributos. Sea A el conjunto de nombres de atributos; para cualquier evento $\sigma \in \Sigma$ y atributo $\alpha \in A$, $\#_{\alpha}(\sigma)$ es

	el valor del atributo α para el evento σ . Cada evento se describe al menos con un nombre de atributo etiquetado, tal que $\#_{\text{nombre}}(\sigma) = (\#_{\text{recurso}}(\sigma), \#_{\text{actividad}}(\sigma)) \in ES$, y por un atributo timestamp que identifica el momento en que se produjo el suceso.
(Senderovich, 2017)	Sea ε el conjunto de todos los eventos posibles, es decir, identificadores únicos de eventos. Sea ε^* el conjunto de todas las secuencias finitas sobre ε siendo $\epsilon \in \varepsilon^*$ la traza vacía. Los eventos se asocian con atributos, por ejemplo, marcas de tiempo, actividades, ubicaciones y recursos. Denotamos por A_ε el conjunto de todos los espacios de atributos de eventos: $A_\varepsilon = \{A_i \mid i \in I\}$, siendo A_i el espacio de atributos i_{th} y I es el conjunto de índices de atributos de eventos.
(Van Eck, 2022)	Un evento $e \in \varepsilon$ es una tupla de n pares nombre-pares de valores. Es decir, $e = ((a_1, v_1), (a_2, v_2), \dots, (a_n, v_n))$ con atributos $a_1, a_2, \dots, a_n \in Attr$, $\forall 1 \leq i < j \leq n : a_i \neq a_j$, y valores $v_1, v_2, \dots, v_n \in Val$.

3.2.2 Definición de Caso

Los casos, como los eventos, pueden tener atributos (Marin-Castro et al., 2021). Los casos son solicitudes individuales y por cada caso se puede registrar una traza de eventos (Van der Aalst, 2016).

Tabla 12. Comparativa de las definiciones y notaciones de Caso

Autor	Definición y representación formal
(Hompeš et al., 2016)	Sea C el universo de casos, es decir, el conjunto de todos los posibles identificadores de casos $n(c)$ es el valor del atributo $n \in N$ para el caso $c \in C$ ($n(c) = \perp$ si c no tiene ningún atributo llamado n).
(Senderovich, 2017)	El conjunto de casos es $C \subseteq \varepsilon^*$, es decir, un conjunto de secuencias finitas de eventos. Exigimos que cada evento aparezca como máximo una vez en algún caso. Un caso $c = \langle c_1, \dots, c_n \rangle \in C$ es, por tanto, una secuencia finita de sucesos tal que $c_i \in \varepsilon$, $i = 1, \dots, n$, siendo n el número de sucesos por caso.

(Seeliger et al., 2018)	Definimos C como el conjunto de todos los posibles identificadores de casos. Los casos también pueden tener atributos, así que para cada caso $c \in C$ y un atributo $a \in A$, definimos $\#_a(c)$ como el valor de un atributo a para el caso c .
(Jokonowo et al., 2019)	Sea C el caso. Un caso son las actividades del ciclo de vida del proceso. Para cualquier evento $c \in C$.

3.2.3 Definición de Traza

Los eventos de un caso se representan en forma de traza, es decir, una secuencia de eventos únicos (Marin-Castro et al., 2021). Estas trazas constituyen la entrada tradicional de muchos enfoques de descubrimiento de procesos y otras técnicas de Minería de Procesos (Van Eck et al., 2022).

Tabla 13. Comparativa de las definiciones y notaciones de Traza

Autor	Definición y representación formal
(Hompeš et al., 2016)	Una traza es una secuencia finita de eventos $\sigma \in \varepsilon^*$, tal que cada evento aparece sólo una vez, es decir, para $1 \leq i < j \leq \sigma $: $\sigma(i) \neq \sigma(j)$.
(De Leoni et al., 2016)	Una traza $t \in \varepsilon^*$ es una secuencia de eventos. Sea $T \in \varepsilon^*$ el universo de las trazas.
(Diamantini et al., 2016)	Una traza l es una secuencia finita de eventos, $l = \langle \sigma_1, \sigma_2, \dots, \sigma_n \rangle$, $\sigma_i \in \Sigma$, $\#_{timestamp}(\sigma_i) < \#_{timestamp}(\sigma_j)$, $i < j$.
(Van Eck, 2022)	Una traza $\sigma = \langle e_1, e_2, \dots, e_n \rangle \in \varepsilon^*$ es una secuencia finita de n eventos ordenados en el tiempo que han sido registrados para una instancia de proceso específica. Para todos los $e_i, e_j \in \sigma$ con $1 \leq i < j \leq n$, se cumple que (1) $\#_{time}(e_i) \leq \#_{time}(e_j)$, (2) $\#_{pi}(e_i) = \#_{pi}(e_j)$ y que (3) $e_i \neq e_j$.

3.2.4 Definición de Registro de eventos

Un registro de eventos almacena datos sobre las actividades que fueron registradas por los sistemas de información durante la ejecución de un proceso. Cada ejecución de una instancia de proceso da lugar a una secuencia de eventos (de Leoni et al., 2016 & Van Eck et al., 2022).

Tabla 14. Comparativa de las definiciones y notaciones de Registro de eventos

Autor	Definición y representación formal
(Hompeš et al., 2016)	Un registro de eventos es un conjunto de casos $L \subseteq C$ tal que cada evento aparece como máximo una vez en todo el registro, es decir, para cualquier $c, c' \in L$ tal que $c \neq c'$: $set(\hat{c}) \cap set(\hat{c}') = \emptyset$.
(De Leoni et al., 2016)	Un registro de eventos L es un conjunto múltiple de trazas, es decir $L \in (T)$.
(Diamantini et al., 2016)	Un registro de eventos L es el registro de una colección de trazas. Cada evento en una traza (registro de eventos) es único, es decir, $\forall \sigma_i, \sigma_j \in L, i \neq j \Rightarrow \sigma_i \neq \sigma_j$.
(Van Eck, 2022)	Un registro de eventos $L = \{\sigma_1, \dots, \sigma_n\} \subset \varepsilon^*$ es un conjunto finito de n trazas. Para todos los $\sigma, \sigma' \in L$ con dos eventos cualesquiera $e \in \sigma, e' \in \sigma'$ se cumple que $\#_{pi}(e) \neq \#_{pi}(e')$.

3.2.5 Definición de Petri net

Los modelos de procesos son representaciones del comportamiento de un proceso o sistema, existiendo varias notaciones de modelado (Van der Aalst, 2016). Los modelos de procesos describen un lenguaje con el orden ejecución de las actividades y las opciones que se pueden realizar, también conocido como flujo de control. Los modelos de procesos suelen visualizarse como un gráfico, aunque también se han desarrollado otras representaciones (Van Eck et al., 2022). Las redes de Petri son un lenguaje de modelado de procesos que permite modelar la concurrencia. Las redes de Petri utilizan una notación de círculos que representan lugares y cuadrados que representan transiciones con flechas que las conectan (Li, 2019).

Tabla 15. Comparativa de las definiciones y notaciones de Petri net

Autor	Definición y representación formal
(Leemans, 2018)	Sea A el alfabeto de actividades y sea τ una etiqueta especial (silenciosa) tal que $\tau \notin A$. Una red de Petri es una tupla $PN = (P, T, F, \ell)$, donde: <ul style="list-style-type: none"> - P es un conjunto finito de lugares - T es un conjunto finito de transiciones, tal que $P \cap T = \emptyset$ - F es un subconjunto finito de arcos dirigidos: $F \in B((P \times T) \cup (T \times P))$

	<ul style="list-style-type: none"> - ℓ es una función de etiquetado de transiciones: $\ell: T \rightarrow A \cup \{\tau\}$
(Kalenkova et al., 2017)	Una red de Petri es una tupla $PN = (P, T, F)$ con P el conjunto de lugares, T el conjunto de transiciones, $P \cap T = \emptyset$, y $F \subseteq (P \times T) \cup (T \times P)$ la relación de flujo.
(Li, 2019)	Una red de Petri es una tupla $N = (P, T, F)$ donde P es un conjunto finito de lugares, T es un conjunto finito de transiciones tal que $P \cap T = \emptyset$, y $F \subseteq (P \times T) \cup (T \times P)$ es un conjunto de arcos dirigidos, llamados relaciones de flujo. Una red de Petri marcada es un par (N, M) , donde $N = (P, T, F)$ es una red de Petri y $M \in \mathbb{B}(P)$ es un subconjunto sobre P que denota el marcado de la red.
(Van Zelst, 2019)	Sea P un conjunto de lugares y sea T un conjunto de transiciones <i>s.t.</i> $P \cap T = \emptyset$. Sea $F = (P \times T) \cup (T \times P)$ la relación de flujo. Además, dejemos que Σ denote el universo de etiquetas, que $\tau \notin \Sigma$ y que $\lambda: T \rightarrow \Sigma \cup \{\tau\}$ denota la función de etiquetado de las transiciones. Una red de Petri N , es una tupla $N = (P, T, F, \lambda)$.
(Syamsiyah, 2020) (Van Eck, 2022)	Una red de Petri es una tupla $\mathcal{N} = (P, T, F, l)$ donde: <ul style="list-style-type: none"> - P es un conjunto de lugares - T es un conjunto de transiciones, $P \cap T = \emptyset$ - $F \subseteq (T \times P) \cup (P \times T)$ es una relación de flujo. - $\ell: T \rightarrow U_A \cup \{\tau\}$ es una función de etiquetado.

3.3 Aplicabilidad de las definiciones formales

La aplicabilidad de las definiciones formales se puede centrar en tres aspectos esenciales en el campo de la Minería de Procesos: la representación en lenguaje formal, los algoritmos y técnicas, y la identificación de atributos.

1) Representación en lenguaje formal: Los formalismos proporcionan un lenguaje formal y una notación matemática estandarizada que los autores muestran para una comunicación clara y precisa entre los expertos en Minería de Procesos. Al tener un lenguaje común, los profesionales pueden compartir y discutir sus ideas, hallazgos y resultados de manera más efectiva, lo que facilita la colaboración y el intercambio de conocimientos en el campo. La notación matemática

utilizada en los formalismos identificados muestra varias ventajas. En primer lugar, brinda un nivel de dominio y exactitud que no siempre se puede lograr con descripciones en lenguaje natural. La notación matemática permite una representación más rigurosa de los conceptos (por lo regular los artículos seleccionados contemplaban una sección de conceptos preliminares) y relaciones dentro de los procesos, lo que evita ambigüedades en la interpretación de los resultados. Otra ventaja de la notación matemática es su capacidad para simplificar y generalizar conceptos complejos. Al representar los procesos de manera abstracta y simbólica, se pueden identificar patrones y regularidades que pueden pasar desapercibidos en las descripciones verbales o visuales. Esto permite una comprensión más profunda de los procesos y la capacidad de aplicar resultados y conocimientos a diferentes contextos y dominios.

2) Algoritmos y técnicas: Además de permitir la representación de los procesos, los formalismos en la Minería de Procesos desempeñan un papel crucial en la aplicación de algoritmos y técnicas (*Handover of work, Token replay, Alineaciones*) para analizar los datos de los procesos y extraer conocimiento que permite la mejora y optimización de los procesos (Urrea-Contreras et al., 2021). Los formalismos proporcionan una estructura y un marco de referencia para aplicar estas técnicas de análisis de datos. Por ejemplo, mediante el uso de redes de Petri, se pueden aplicar algoritmos de descubrimiento de procesos que permiten identificar patrones recurrentes, desviaciones, cuellos de botella y oportunidades de mejora en los procesos. Además, los formalismos permiten la definición de métricas y medidas de rendimiento que se utilizan en la evaluación y comparación de los procesos. Estas métricas pueden ser utilizadas para identificar áreas problemáticas, evaluar la eficiencia y efectividad de los procesos, y tomar decisiones basadas en datos para la optimización y reingeniería de los mismos.

3) Identificación de atributos: Las representaciones matemáticas muestran los diferentes atributos relevantes en el análisis de los procesos. Al utilizar fórmulas matemáticas, es posible expresar de manera precisa las relaciones entre los atributos, lo que facilita la identificación de patrones y tendencias entre ellas. Las representaciones matemáticas proporcionan un marco para la identificación y el análisis de los atributos en la Minería de Procesos, lo que contribuye a una comprensión más profunda y precisa de los procesos analizados. Tal es el caso de la definición de perspectiva por (Hompe et al., 2016) (Tabla 16), la cual contempla los atributos y la proyección

de múltiples perspectivas. Por otro lado, más en lo particular se encuentra la descripción de la perspectiva de flujo de control por (de Leoni et al., 2016), donde la descripción de esta perspectiva se realiza basada en las notaciones de manipulación de traza, exponiendo definiciones relacionadas al número de ejecuciones de actividad, primera ocurrencia de actividad, actividad previa en la traza y actividad actual.

Tabla 16. Definición y representación formal de perspectiva

Autor	Definición y representación formal
(Hompeš et al., 2016)	Sea P una perspectiva. $\iota_P : C \rightarrow R^m$ denota la función que asigna un caso a un vector real de longitud m según la perspectiva P . m es el número de atributos en P , por ejemplo, el número de recursos diferentes en el registro. c/P denota la proyección del caso $c \in L$ a una perspectiva P . Además que $c \iota_{\{P_1, P_2, \dots, P_K\}} = c \left[\iota_{P_1} \parallel (\iota_{P_2}) \parallel \dots \parallel \iota_{P_K} \right]$, es decir, se concatenan los vectores de perfil resultantes de la proyección a múltiples perspectivas.

Por otro lado, el modelar un proceso de software por medio de una red Petri permite conocer el flujo de trabajo, las actividades, transiciones y condiciones que pueden presentar. En la Figura. 9, se observa como en la red Petri las posiciones (P) se representan como círculos y se numeran de P1 al P14 para indicar su identificación única. Las transiciones (T) se representan como barras verticales y también se numeran del T1 al T14. Mediante flechas dirigidas, se indican las relaciones de incidencia entre las posiciones y las transiciones. Una flecha que va de una posición a una transición indica que la transición consume recursos de esa posición (por ejemplo de P2 a T2 o T3), mientras que una flecha que va de una transición a una posición indica que la transición genera o produce recursos en esa posición (por ejemplo de T10 a P11). Esta representación permite visualizar las interacciones entre las posiciones y las transiciones, así como sus conjuntos, proporcionando una comprensión clara de cómo se produce el flujo de recursos en el sistema.

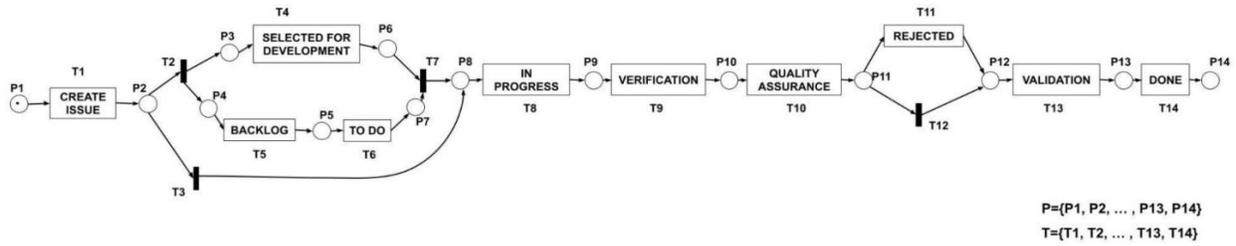


Figura 9. Red Petri de los estados de transición de un proceso de software.

Se observa como existen flujos paralelos entre las posiciones P3 y P4 que dependen de los eventos que describen el inicio de la actividad y su terminación entre P6 y P7 respectivamente. El analizar en conjunto y aplicar las definiciones de caso y traza se pueden institucionalizar mejores prácticas y/o enfoques para la gestión de proyectos de software, la asignación de recursos pero sobre todo, la toma de decisiones basadas en diagnósticos, la obtención de nuevos modelos o los datos existentes y generados por las empresas de desarrollo de software.

Capítulo 4. Una propuesta de aplicación de la Minería de Procesos en proyectos de software

Este capítulo tiene como propósito presentar la propuesta de la adaptación de la Metodología L* por medio de capas para brindar un enfoque de Ingeniería de Software. Así como las notaciones matemáticas formales utilizadas en la Minería de Procesos para su integración en una guía técnica de MoProPEI-MP (Urrea Contreras et al., 2018) y su aplicabilidad en proyectos de Minería de procesos de software a través de áreas y temas de conocimiento para la identificación y análisis de escenarios.

4.1 Descripción de la metodología L* en capas

Para guiar el minado de los procesos de software, se implementó la metodología del modelo de ciclo de vida L* (Van der Aalst, 2016) (Figura 10) por capas, representado por 3 fases: 1) Preprocesamiento, 2) Minería de procesos, y 3) Minería de perspectivas. Estas tres fases se llevan a través de 5 pasos (Van der Aalst, 2016):

Fase de Pre-procesamiento: En la fase inicial se presenta el paso 1) Obtener un registro de eventos, donde los datos se extraen de los sistemas de información.

Fase de Minería de Procesos: En esta fase se observan los pasos 2) Crear o descubrir un modelo de proceso, que se centra en las técnicas y algoritmos de descubrimiento de procesos, y 3) Conectar los eventos del registro con las actividades en el modelo, lo cual es esencial para proyectar la información en los modelos y añadir perspectivas, los eventos en el registro y las actividades del modelo se pueden conectar utilizando la técnica de repetición (verificación de la conformidad).

Fase de Minado de perspectivas: En la última fase se describen los pasos 4) Ampliar el modelo, en el que se integran perspectivas de Minería de Procesos como la organizacional, del caso y la del tiempo; y 5) Retornar el modelo integrado, que puede utilizarse para diversos fines, como obtener una visión holística del proceso y servir de entrada para otras herramientas y enfoques.

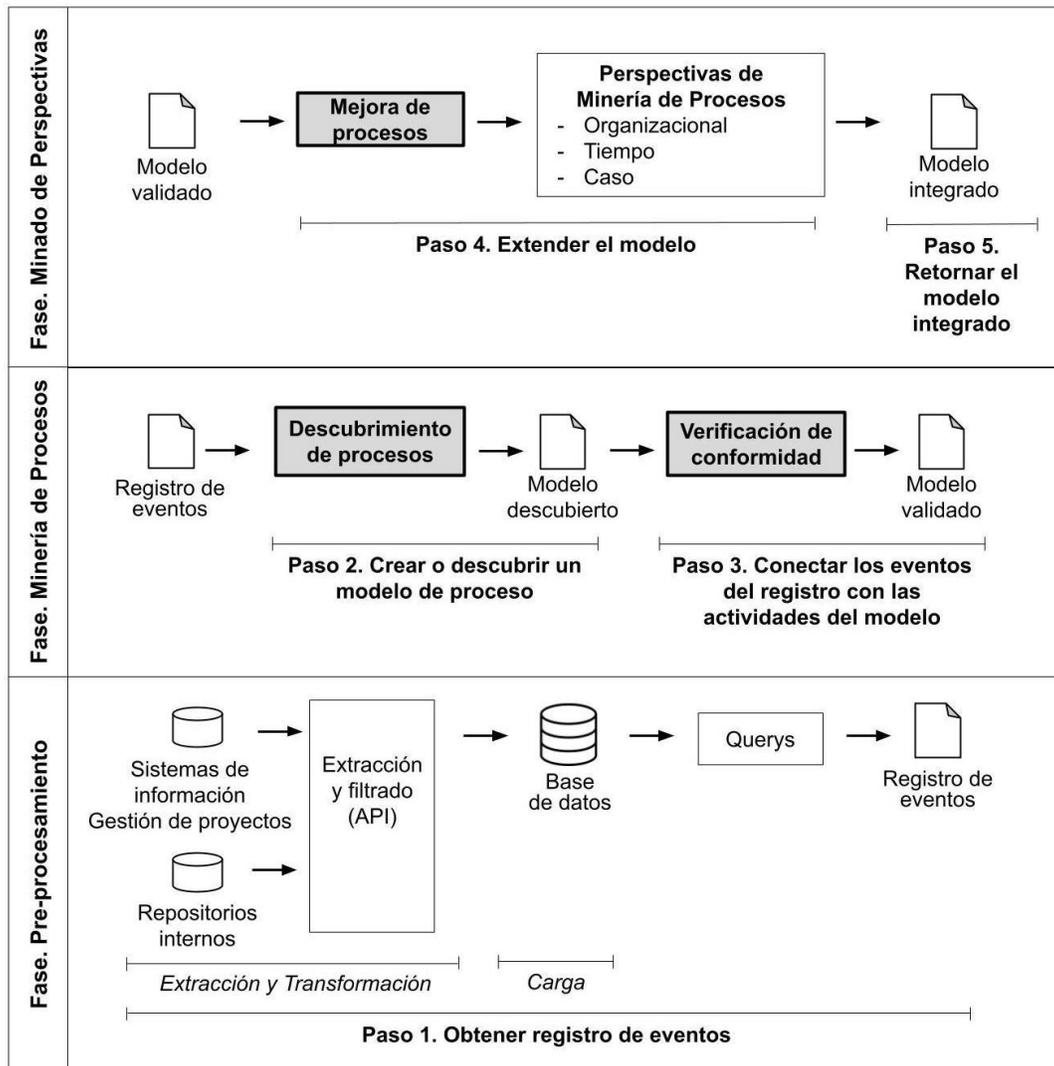


Figura 10. Aplicación de la Metodología L* para el proyecto de desarrollo de software

Se seleccionó la metodología L* con el fin de aplicar las perspectivas de Minería de Procesos en los procesos de software. Por lo general, los procesos de software no se caracterizan por ser altamente estructurados (Kneuper, 2018) y aun cuando L* tiene su enfoque principal en procesos

estructurados, se puede ajustar y emplear de manera eficaz en situaciones menos estructuradas, por lo que para su adaptación se consideraron los siguientes factores:

- **Extracción y transformación de los datos:** La extracción y transformación adecuada de los datos son aspectos esenciales en la Minería de Procesos, especialmente en contextos de procesos menos estructurados. En este sentido, los APIs desempeñan un papel fundamental al proporcionar un medio estructurado y programático para acceder a datos. Esto simplifica significativamente la extracción y transformación de información de sistemas y aplicaciones externas, permitiendo un manejo más adecuado de la información y posibilitando la limpieza de los datos.
- **Integración de perspectivas:** La metodología L* permite la integración de perspectivas a medida que se avanza en el análisis del proceso. Aunque la metodología puede inicialmente centrarse en el flujo de control, perspectivas como la organizacional, caso y del tiempo pueden implementarse en pasos posteriores para profundizar en otros aspectos del proceso. Esta flexibilidad brinda la oportunidad de obtener una visión más completa del proceso de software.

Para ejecutar la metodología adaptada, en el paso 1, durante el proceso ETL (*Extract, Transformation and Load*), se desarrolló un script en Python para obtener datos de los sistemas involucrados en el proyecto de desarrollo de software, Jira y Github. El script permitió la conexión con las APIs de los sistemas, lo que permitió obtener información detallada sobre el proceso¹.

4.1.1 Actividades ETL: Desarrollo de un API

En el proceso de extracción y filtrado se utilizaron REST API y GraphQL, donde se eligió GraphQL por su capacidad para acceder a los datos mediante consultas y su flexibilidad a la hora de definir la información requerida. Los resultados de las consultas se guardaron en formato JSON para estructurar los datos antes de ser almacenados finalmente en la base de datos no relacional

¹ Revisar Anexo 2.

MongoDB. En primer lugar, se extrajeron y filtraron los issues de Jira (Figura 11), obteniendo como atributos el issue ID, la fecha de creación y la información del creador.

```
query = {
  'jql': 'project = %s' % (project_name),
  'startAt' : offset,
  'maxResults' : maxResultGetList
}
```

Figura 11. Query utilizado para obtener el listado de *issues* del proyecto.

Tras obtener la lista de issues relacionadas con el proyecto, se consultaron los datos del campo changelog para extraer en específico el estado de Jira, debido a que contiene información sobre los cambios de estado de cada issue, lo que permite obtener un registro detallado de los movimientos de los issues a lo largo del tiempo y analizar el flujo de control del proceso de desarrollo. Se establecieron filtros a través de la API para obtener específicamente el estado requerido. De este modo, se obtuvieron atributos como el issue ID, el estado predecesor, el estado posterior, la fecha del cambio y quién lo realizó. Una vez filtrados los datos, se almacenaron junto con los atributos de la primera extracción relacionados con el issue ID (Figura 12).

```
changelog_csv_array = []
changelog_csv_array.append
([issue["key"], "", "To Do",
issue["created"], issue["creator"]])

for change in changelog:
  for item in change["items"]:
    if(item["field"] == 'status'):
      changelog_csv_array.append
      ([issue["key"], item["fromString"],
      item["toString"], change["created"],
      change["author"]["displayName"]])
return changelog_csv_array
```

Figura 12. Filtrado y vinculación de los atributos obtenidos de Jira.

Para los registros asociados a actividades de codificación, se implementó una consulta para filtrar aquellas issues relacionadas con Jira y Github (Figura 13). Se obtuvieron atributos con los detalles de los commits, como la referencia del commit, la fecha de creación, quién lo creó y la URL de acceso al cambio.

```

graphQLQuery = """query MyQuery {
  jira {
    issue(id: "ari:cloud:jira:4e503e73") {
      id
      devInfoDetails {
        commits {
          details {
            created
            entityUrl
            providerCommitId
            author {
              name
            }
          }
        }
      }
    }
  }
}""" % (issue["id"])

```

Figura 13. Query para la obtención de los *commits* por *issue*.

Una vez extraídos y filtrados los datos, se almacenaron en la base de datos NoSQL MongoDB para realizar las consultas necesarias y obtener el registro de eventos, finalizando así el Paso 1. En el paso 2, se descubrió el modelo de proceso en red de Petri aplicando un algoritmo de descubrimiento al registro de eventos. Como propone L*, en el paso 3 se aplicó la técnica de conformidad Token Replay para conectar los eventos del registro con las actividades del modelo descubierto y generar un modelo validado. El paso 4 consistió en ampliar el modelo validado con las perspectivas de caso, organización y tiempo para caracterizar mejor el proceso mediante los atributos de proceso disponibles y establecer un modelo integrado. Por último, el paso 5 consistió en utilizar el modelo integrado para crear una visión general del proceso extraído.

4.2 Propuestas de definiciones matemáticas

En el capítulo 3 se presentaron las definiciones y representaciones formales de cada uno de los distintos elementos de la Minería de Procesos para conocer el conocimiento semántico del significado y representación de las relaciones entre ellos. Este enfoque consistió en analizar cómo se conciben y se expresan matemáticamente estas relaciones dentro del ámbito de la Minería de Procesos.

De esta manera, las notaciones se compararon para establecer un lenguaje formal y estructurado, ofreciendo un conocimiento explícito para representar las actividades y facilitar la interpretación, el análisis y modelado de procesos. Para determinar las definiciones formales, se establecen universos conceptuales con el propósito de proporcionar un contexto para las definiciones, permitiendo una comprensión estructurada de los elementos de la Minería de Procesos.

En esta investigación se proponen las siguientes definiciones las cuales pueden ser utilizadas en proyectos de desarrollo de software:

Definición 1 (Universos). Se utilizan las siguientes notaciones para los universos:

- ε denota el universo de eventos únicos, es decir, el conjunto de todos los posibles identificadores que se refieren a eventos concretos,
- A denota el conjunto de todos los posibles nombres de atributos,
- V denota el conjunto de todos los posibles valores de atributos,
- C denota el universo de casos.

Definición 2 (Evento). Un evento e es una asignación de valores, el cual se describe mediante un conjunto de atributos. Sea $e \in \varepsilon$ para cualquier evento, $a \in A$ para los atributos y $v \in V$ para los valores, denotamos que, $e = ((a_1, v_1), (a_2, v_2), \dots, (a_n, v_n))$, en donde $a \subseteq v$.

Definición 3 (Caso). El conjunto de casos es $C \subseteq \varepsilon^*$, es decir, un conjunto de secuencias finitas de eventos. Cada evento aparece como máximo una vez en algún caso. Un caso $c = \langle c_1, c_2, \dots, c_n \rangle \in C$. Los casos también pueden tener atributos, así que para cada caso c , un atributo $a \in A$, definimos $c = \langle c_1, a_1, (c_2, a_2), \dots, (c_n, a_n) \rangle$.

Definición 4 (Traza). Una traza puede ser vista como un caso, o una secuencia finita de eventos. Sea t una traza y ε^* el conjunto de todas las secuencias finitas sobre ε , $t \in \varepsilon^*$ por lo que $t = \langle e_1, e_2, \dots, e_n \rangle$, donde n es el número de eventos por traza.

Definición 5 (Registro de eventos). Un registro de eventos L es una secuencia finita de trazas donde $L = (t_1, t_2, \dots, t_n)$.

Definición 6 (Petri net). Una red de Petri es una tupla $N = (P, T, F, l)$ donde:

- P es un conjunto de lugares (posiciones)
- T es un conjunto de transiciones, $P \cap T = \emptyset$ (transiciones, conjuntos finitos)
- $F \subseteq (T \times P) \cup (P \times T)$ es una relación de flujo (conjunto de arcos dirigidos)
- $\ell : T \mapsto \mathcal{A} \cup \{\tau\}$ es una función de etiquetado.

De acuerdo a la definición de los elementos de Minería de Procesos y sus relaciones al extraer los datos de los repositorios se sugiere el siguiente orden: identificar los eventos (Definición 2) agrupándolos en sus correspondientes casos (Definición 3) para crear un registro de eventos (Definición 5). Una vez definido el registro de eventos, se genera un modelo basado en los datos extraídos que puede ser representado con una red Petri (Definición 6) para posteriormente verificar la conformidad de los eventos que es representado por medio de las trazas (Definición 4). Este modelo podría extenderse con las perspectivas de Minería de Procesos con la identificación de los atributos correspondientes (Tabla 17).

La Tabla 17 presenta la relación de atributos que se utilizan en el proceso de minado de las perspectivas de Minería de Procesos. Estos atributos son variables que se extraen y analizan para obtener información relevante sobre el flujo, el tiempo y la relación organizacional de los procesos. La Tabla 17 muestra una lista de los atributos relacionados a las perspectivas de flujo de control, organizacional, del caso y del tiempo, junto con ejemplificaciones de atributos como el orden de actividades, timestamp y caso. Estos atributos pueden incluir características como la duración de una actividad, el estado de un evento, el rol de un participante, la secuencia de actividades, entre otros.

La relación de atributos en la Tabla 17 proporciona una guía para los investigadores y profesionales que realizan el minado de procesos, ayudándoles a identificar los atributos relevantes que deben considerar en su análisis. Esta relación de atributos sirve como un recurso útil para comprender y aplicar las perspectivas de Minería de Procesos en diferentes contextos y proyectos.

Tabla 17. Relación de atributos para el minado de perspectivas de Minería de Procesos

Flujo de Control	Organizacional	Caso	Tiempo
Actividad	ID Actividad	<i>Caso***</i>	<i>Timestamp*</i>
<i>Orden Actividades*</i>	Actividades (Tipo)		Horas invertidas
	<i>Timestamp*</i>		Horas efectivas
	Horas invertidas		
	<i>Recursos**</i>		
<i>Orden Actividades*</i> Timestamp (Inicio/Fin) <ul style="list-style-type: none"> - Fecha DD/MM/AA - Hora HH:MM 			
<i>Recursos**</i> Roles Estructuras organizacionales <ul style="list-style-type: none"> - Departamentos - Equipos/proyectos 			
<i>Caso***</i> El atributo de caso dependerá del análisis			

Es importante destacar que los atributos específicos y su relevancia pueden variar dependiendo del dominio de aplicación y los objetivos del minado. Por lo tanto, esta tabla brinda una visión general de los atributos comunes utilizados en el minado de perspectivas, pero se recomienda adaptarla y ajustarla según las necesidades y particularidades de cada proyecto de software.

4.3 Análisis de Escenarios

El analizar las representaciones formales de los elementos de Minería de procesos permitió constatar que el conocimiento de un dominio específico se representa por distintos niveles de formulación. Uno de los propósitos fue el identificar cómo los roles deberán realizar sus actividades relacionadas a su dominio de aplicación siguiendo el flujo de la Figura 8. Para esto, se han identificado escenarios que representan problemáticas identificadas al momento de la ejecución de las actividades en proyectos de Minería de procesos. Los escenarios corresponden a:

- 1) Creación de un registro de eventos;
- 2) Asociación de los sistemas de software;
- 3) Uso de

algoritmos y herramientas de Minería de Procesos; y 4) Aplicación de las perspectivas de Minería de Procesos. Dichos escenarios fueron seleccionados en función de la revisión de problemáticas identificadas en la literatura (Van der Aalst, 2016; Keith y Vega, 2016; Zhao y Zhao, 2014; Burattin, 2015; Kneuper, 2018). En cada uno de ellos se especifican los roles Responsable del proyecto SPI, Líder de proyecto, Gestor de procesos, Experto en Minería/Analista de datos. Estos dos últimos poseen conocimiento tácito asociado a la identificación de variables para la creación y análisis de registros de eventos dando sentido a la información extraída. A continuación se presentan los escenarios en las Tablas 18 a la 21.

Tabla 18. Áreas y temas de conocimiento para la identificación del Escenario 1 Creación de un registro de eventos

Escenario 1. Obtención de un registro de eventos	
Descripción	Para la aplicación de la Minería de procesos es esencial la creación de un registro de eventos extraído de las distintas fuentes de datos (sistemas de información, repositorios, documentación, etc.). Los eventos como mínimo deben estar relacionados a un caso, una actividad y un <i>timestamp</i> , pero pueden contener más información la cual se pueda minar a través de las perspectivas de Minería de procesos (Flujos de control, organizacional, del caso y del tiempo).
Rol(es) involucrado(s)	Experto en Minería de procesos
Área(s) de conocimiento	<ul style="list-style-type: none"> - Conocimiento de los sistemas de software - Conocimiento de la disciplina de Minería de procesos - Conocimiento del proceso de desarrollo de software
Temas de conocimiento	<ul style="list-style-type: none"> - Conocimiento del dominio de los sistemas de software - Extracción de información - Extracción de un registro de eventos - Perspectivas de Minería de procesos - Herramientas y algoritmos de Minería de procesos - Conocimiento tópicos de desarrollo de software - Documentos de entrada - Documentos de salida - Identificación de roles

Tabla 19. Áreas y temas de conocimiento para la identificación del Escenario 2 Asociación de los sistemas de software

Escenario 2. Asociación de los sistemas de software	
Descripción	Para el desarrollo de software existe una diversidad de sistemas que ayudan a la gestión del proceso como lo son los Issue Tracking System y Version Control System. De forma general, los procesos de software se ven involucrados en tener dispersa su información en varios sistemas, por lo que conocer las relaciones entre ellos y la forma en que se vinculan con las actividades y otros elementos del proceso es necesaria para la obtención de un registro de eventos.
Rol(es) involucrado(s)	Experto en Minería de procesos
Área(s) de conocimiento	- Conocimiento de los sistemas de software - Conocimiento del proceso de software
Temas de conocimiento	- Estructura relacional entre los sistemas de software - Dependencia de los sistemas de software - Actividades del proceso de software

Tabla 20. Áreas y temas de conocimiento para la identificación del Escenario 3 Uso de herramientas y algoritmos de Minería de procesos

Escenario 3. Uso de herramientas y algoritmos de Minería de procesos	
Descripción	Actualmente, existen varias herramientas de Minería de procesos tanto de acceso libre como comerciales con una gran cantidad de plugins que añaden funcionalidad a las herramientas a partir de nuevos algoritmos y/o técnicas. Se debe conocer las características de las principales herramientas y su compatibilidad con los procesos y sistemas de software.
Rol(es) involucrado(s)	Experto en Minería de procesos
Área(s) de conocimiento	- Conocimiento de los sistemas de software - Conocimiento de la disciplina de Minería de procesos
Temas de conocimiento	- Extracción de información - Extracción de registro de eventos - Tipos de minado de la Minería de procesos - Perspectivas de Minería de procesos - Herramientas y algoritmos de Minería de procesos

Tabla 21. Áreas y temas de conocimiento para la identificación del Escenario 4 Minado de las perspectivas de Minería de proceso

Escenario 4. Minado de las perspectivas de Minería de procesos	
Descripción	Las perspectivas de Minería de procesos ofrecen una caracterización completa del proceso de software. Su minado dependerá de la información que contengan los distintos sistemas o repositorios disponibles. Asimismo, las herramientas y algoritmos de Minería de procesos influyen en el minado de las perspectivas.
Rol(es) involucrado(s)	Experto en Minería de procesos
Área(s) de conocimiento	<ul style="list-style-type: none"> - Conocimiento de los sistemas de software - Conocimiento de la disciplina de Minería de procesos - Conocimiento del proceso de software
Temas de conocimiento	<ul style="list-style-type: none"> - Estructura de los sistemas de software - Dependencia entre los sistemas de software - Conocimiento del dominio de los sistemas de software - Extracción de información - Extracción de un registro de eventos - Tipos de minado de la Minería de procesos - Perspectivas de Minería de procesos - Herramientas y algoritmos de Minería de procesos - Conocimiento del desarrollo de software - Actividades del proceso de software - Documentación de entrada - Documentación de salida - Participación de roles

Capítulo 5. Resultados del Estudio de caso

Este capítulo presenta la aplicación de la Metodología L* adaptada en capas en una empresa de desarrollo de software. El objetivo es la obtención de un modelo integrado de procesos relacionado a un proyecto de desarrollo de software de tal manera que la empresa pueda conocer si la ejecución de los procesos son acordes a la documentación de los mismos, descubrir si existen desviaciones dentro de su modelo e identificar actividades no conformes y con base a los hallazgos detectados establecer acciones correctivas para hacer más eficientes sus procesos. El proyecto bajo estudio fue minado a partir de los registros de eventos almacenados en los sistemas para la gestión de proyectos de desarrollo de software: *Issue Tracking System* (Jira) y SIMIo. Siendo este último un desarrollo propio y privado de la organización. Asimismo, se contempló *Version Control System* (GitHub) para realizar consultas sobre la información de los *commits*. Con base a lo anterior, las metas del estudio son las siguientes:

- 1) Identificar los datos asociados a los registros de eventos en cada uno de los repositorios de acuerdo al proceso específico de desarrollo.
- 2) Extraer los datos de los repositorios para construir un registro de eventos.
- 3) Aplicar técnicas de descubrimiento de procesos para generar un modelo de procesos a partir del registro de eventos.
- 4) Aplicar verificación de la conformidad para comparar el registro de eventos y el modelo de procesos descubierto.
- 5) Extensión de un modelo de proceso con información extraída del registro de eventos.

5.1 Caracterización de la empresa de desarrollo de software

La empresa de software se encuentra establecida en Mexicali, Baja California, con una cartera de proyectos a nivel regional, nacional e internacional (Australia y Estados Unidos). Cuenta con 12 años de antigüedad y con un total de 15 empleados correspondiendo a una pequeña y mediana empresa. En el 2019 obtuvo la certificación en nivel 2 de CMMI-DEV v1.3.

5.1.1 Proceso de desarrollo de la MiPyME de software

Las metodologías específicas para el desarrollo y mantenimiento de software adoptadas por la MiPyME son: Scrum (Scrum-institute, 2017) y Kanban (Brechner, 2015). En la Figura 14, se muestra el flujo de proceso de desarrollo actual de la organización en dos bloques, el primero comprende la definición del proyecto, requerimientos, diseño, configuración de ambientes de trabajo, ciclos y desarrollo Kanban. El segundo bloque integra las actividades de desarrollo, integración y despliegue continuo en el entorno de DevOps.

El primer bloque del proceso inicia cuando el Departamento de proyectos le entrega al Arquitecto de software las características generales propuestas del software para que realice el Análisis de requerimientos en el que también se diseña un prototipo el cual es validado por el cliente. Una vez validado, el Gestor de proyectos define las historias de usuario basado en las características y en el prototipo. Posteriormente, el Arquitecto de software establece el Diagrama de solución que integra el modelo de datos y se definen las tecnologías que se van a utilizar. El Desarrollo Kanban se administra por medio de *issues*, con las características del software a realizar y el trabajo técnico.

El segundo bloque comienza durante el Desarrollo del software, cuando un programador realiza una protección del código, lo que procede a la Integración Continua (CI) y Despliegue Continuo (CD). Por parte de la CI se compila la protección del código para realizar pruebas automáticas y después realizar su empaquetamiento y almacenamiento. Con respecto a CD, contempla un ambiente de pruebas controlado y de staging donde se lleva a cabo la validación de los cambios con información real del proceso. Por último, el ambiente de producción con el producto de software final validado por el cliente.

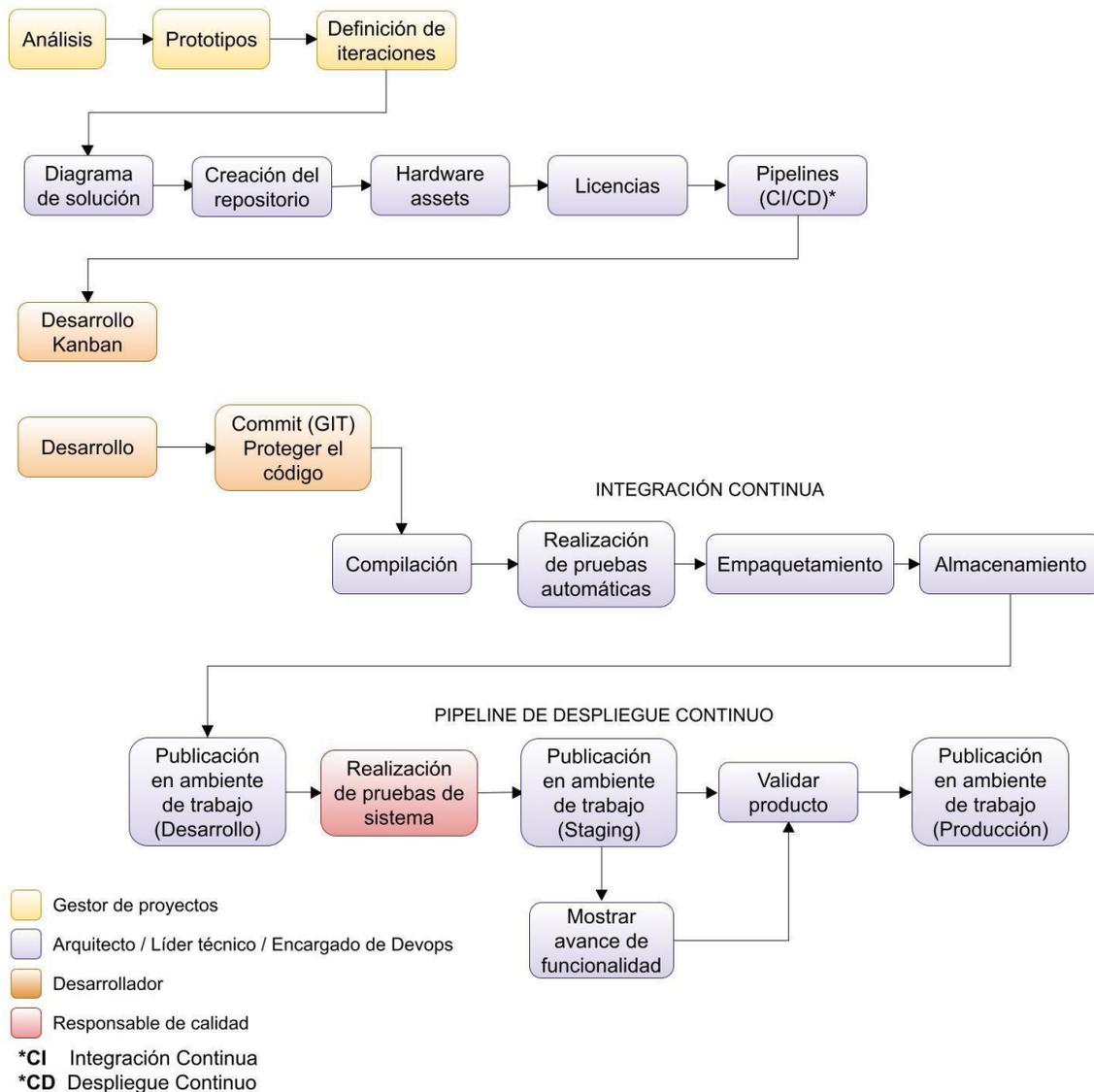


Figura 14. Diagrama del flujo del proceso de desarrollo de la empresa de software.

5.1.2 Repositorios

Las herramientas de apoyo para almacenar los registros de eventos en el proceso de desarrollo de software por la MiPyME son: Jira®, Github® y SIMIo. A continuación se describen brevemente cada una de ellas.

- **JIRA** ®

Jira es la herramienta de software (Atlassian, s.f.) en la que se da inicio al proceso de desarrollo, con la creación del tablero Kanban configurado por el siguiente flujo que detonan los cambios de estado: *Selected for Development*, *To Do*, *In Progress*, *Verification*, *Quality Assurance*, *Rejected*, *Validation* y *Done*. De esta manera, se da el seguimiento a los *issues* (*Issue Tracking System*). Los requerimientos se encuentran en historias de usuario las cuales son planificadas por sprints y distribuidas entre el equipo de proyecto de manera auto-organizada en la ejecución de sus tareas. La constante actualización de la información del sistema y su disposición para todos los integrantes del proyecto permite realizar un análisis del seguimiento de los *issues* presentes en un proyecto de software.

La actualización automática de la información permite la exportación de un historial completo con la utilización de filtros o sentencias que Jira proporciona; así como *datasets* de acuerdo a los diferentes flujos que puede seguir el proceso. En Jira se pueden integrar aplicaciones en una solución de desarrollo de productos estrechamente integrada para los equipos que necesitan hacer más en menos tiempo (Atlassian, s.f.). Por ejemplo, la aplicación *Time in Status* puede generar reportes relacionados con el tiempo que se ha mantenido un estado dentro del *issue*, su promedio de duración, fecha de entrada a un estado en particular, transiciones y recuento de los estados. Por otra parte, con la aplicación *Issue History* for Jira se puede supervisar los cambios de los *issues* seleccionados, realiza un seguimiento de todas las actualizaciones que se han realizado en un *issue*, elegir las claves de los *issues* que se desea supervisar los cambios, y ordenarlos por fecha y clave.

- **GITHUB** ®

Una vez que las tareas se encuentran en estado de *To Do*, el flujo de control de versiones continúa a través de la implementación de una interface entre Jira y el *Version Control System* GitHub (GitHub, s.f.) a partir de los *issues* existentes al momento de pasar del estado *To Do – In Progress* al detonarse la creación de un *branch* o un *commit*. La interfaz es relacionada por la empresa con el propósito de mantener una trazabilidad entre la planificación del proyecto y la gestión del código en cuanto a la colaboración de escritura del código entre los diferentes programadores,

probar y desplegar las versiones que se van generando dónde puede añadir anotaciones y debatir sobre determinados cambios realizados en un *commit* específico.

- **SIMIo**

SIMIo es un software desarrollado por la MiPyME para llevar un registro detallado del trabajo realizado por los miembros del equipo de proyecto. Este software completa el flujo de eventos almacenado en Jira y GitHub. De SIMIo se puede descargar un informe con el nombre del proyecto, nombre del miembro del equipo que registra el trabajo, su rol, fecha, detalles del trabajo, horas invertidas de trabajo y horas efectivas. La MiPyME definió una fórmula para calcular las horas efectivas del proyecto que junto con el informe brindan información significativa para la toma de decisiones por parte del Gestor de proyectos y la alta dirección.

5.1.3 Actividades del proceso de software

Para definir las actividades de los proyectos, se analizaron los estados del tablero Kanban presentados en Jira (Tabla 22), con el fin de proporcionar una visión del proceso de desarrollo de software. Mediante el estudio y análisis de los estados, se obtuvo información sobre cómo se realizan las actividades y cómo fluyen las tareas en el proceso. Cada estado representa una etapa distinta en el ciclo de vida de una tarea o elemento del proyecto. Desde la creación en el *backlog* hasta la finalización en el estado "*Done*", cada estado refleja una acción o conjunto de acciones que deben realizarse para avanzar en el proceso de desarrollo de software de la MiPyME.

Tabla 22. Descripción de los estados de Jira utilizados por la MiPyME.

Estado de Jira	Descripción
<i>Backlog</i>	Comprende una lista priorizada de tareas pendientes en el proyecto, utilizada para planificar y gestionar las actividades futuras del equipo de desarrollo.
<i>Selected for Development</i>	Agrupar los elementos del <i>backlog</i> que han sido elegidos para ser trabajados en el sprint o iteración actual.
<i>To Do</i>	Contiene las tareas que aún no han comenzado y están pendientes de ser asignadas a miembros del equipo para su ejecución.
<i>In Progress</i>	Presenta las tareas o elementos de trabajo en los que están trabajando activamente los miembros del equipo, es decir, que están en proceso de desarrollo.

<i>Verification</i>	Indica tareas o elementos de trabajo que han sido completados por el equipo de desarrollo y están en proceso de revisión.
<i>Quality Assurance</i>	Muestra las pruebas y revisiones de las tareas para garantizar su calidad.
<i>Validation</i>	Consiste en validar las tareas después de que hayan pasado por el proceso de desarrollo y control de calidad.
<i>Rejected</i>	Contiene las tareas o elementos de trabajo que no cumplen las normas o requisitos y necesitan correcciones antes de avanzar en el proceso.
<i>Done</i>	Comprende las tareas completadas listas para su entrega y despliegue.

5.2 Análisis del Proyecto 1

El proyecto 1 corresponde a una aplicación Web para la gestión de servicios de un gimnasio que contempla las funciones de: registro de clientes, pago de membresía, renovación de cuotas y control de acceso. Este proyecto tuvo una duración aproximada de 2 meses (Mayo – Junio del 2021). El equipo de proyecto estaba integrado por los roles de (Tabla 23): Gestor de proyectos, Líder técnico, Programador y Responsable de calidad. El Gestor de proyectos, así como el Líder Técnico poseen 10 años de antigüedad en la empresa. El rol de programador fue desempeñado por 2 miembros del equipo, un programador junior con una experiencia de 2 años trabajando en la MiPyME y de tiempo completo dedicado al proyecto; mientras que el programador 2 cuenta con un nivel *medium* y 6 años de experiencia, a quien se le asignaron los *issues* de mayor complejidad y al mismo tiempo participaba en otro proyecto de desarrollo de software.

Tabla 23. Roles del equipo del proyecto.

Rol	Antigüedad
Gestor de proyectos	10 años
Líder técnico	10 años
Programador 1	junior – 2 años
Programador 2	medium – 6 años
Responsable de calidad	5 años

Según lo expuesto por el Gestor de proyectos, el proyecto inició con la intervención de dos programadores que posteriormente por la carga de trabajo y la situación económica de la empresa

debido a la pandemia se terminó con un solo programador. Asimismo, al momento de terminar este proyecto, el Responsable de calidad dejó de laborar en la empresa con una antigüedad de 5 años.

5.2.1 Obtención del Registro de Eventos

En la primera fase (Pre-procesamiento) se realizó una primera extracción de los eventos registrados de las herramientas Jira y SIMIo. El historial de los estados o cambios del flujo de las actividades (*issues*) documentados en el tablero Kanban se obtuvieron del repositorio de Jira, mientras que del repositorio de SIMIo se obtuvo la descripción ampliada de las actividades registradas por los programadores. La relación de los registros de eventos extraídos en cuanto a su trazabilidad con SIMIo se realizó manualmente por la ausencia de un atributo explícito para su relación con Jira. La trazabilidad manual se efectuó en colaboración con el Gestor de proyectos, la cual consistió en revisar los eventos obtenidos de SIMIo para relacionarlos con la clave del *issue* de Jira al que hacían referencia. Con base en esta trazabilidad se corrigieron las inconsistencias de algunos registros de eventos y se asignaron las etapas de desarrollo de software (Planeación, Desarrollo, Pruebas y Despliegue).

Una vez realizada la trazabilidad entre los repositorios, el registro de eventos está conformado con las cardinalidades y relaciones de los elementos de la Minería de Procesos. De esta forma, de los repositorios y los elementos de datos que almacenaban, se definieron los siguientes atributos que describen a los eventos: 1) Clave *Issue* como identificador del caso, 2) Estado del *issue* a modo de definir las actividades, 3) *Timestamp* (fecha y hora) que especifica la ejecución de la actividad, 4) Rol que realiza la actividad, 5) Actor quien realiza la actividad y 6) Etapa de desarrollo de software a la que pertenece la actividad. Los primeros cinco atributos pertenecen al registro de Jira y el último (etapa de desarrollo de software) corresponde a la trazabilidad manual entre SIMIo y Jira.

La Tabla 24 muestra las actividades en el registro de eventos ordenadas de mayor a menor por el número de frecuencia. Cada instancia de actividad está relacionada con un caso, por tal motivo, se seleccionó como actividad al estado del *issue* para poder establecer los casos y que estos estuvieran en función con la clave del *issue*. Como se observa en la Figura 15, el fragmento del registro de eventos representa los eventos que integran el caso del *issue* 8, en donde cada columna

y renglón corresponden a un atributo y evento respectivamente. Por cada caso se puede registrar una traza de eventos que corresponde a la ejecución del caso de principio a fin (Van der Aalst, 2016). De esta forma, las trazas del registro de eventos marcaron los posibles caminos a través del flujo de actividades del modelo de proceso.

Tabla 24. Actividades en el registro de eventos del proyecto de software ordenados por su frecuencia Proyecto 1.

Actividad	Frecuencia
<i>Done</i>	65
<i>Validation</i>	64
<i>Quality Assurance</i>	63
<i>Create Issue</i>	62
<i>Verification</i>	62
<i>In Progress</i>	54
<i>Selected For Development</i>	43
<i>To Do</i>	40
<i>Backlog</i>	27
<i>Rejected</i>	1
<i>Total:</i>	481

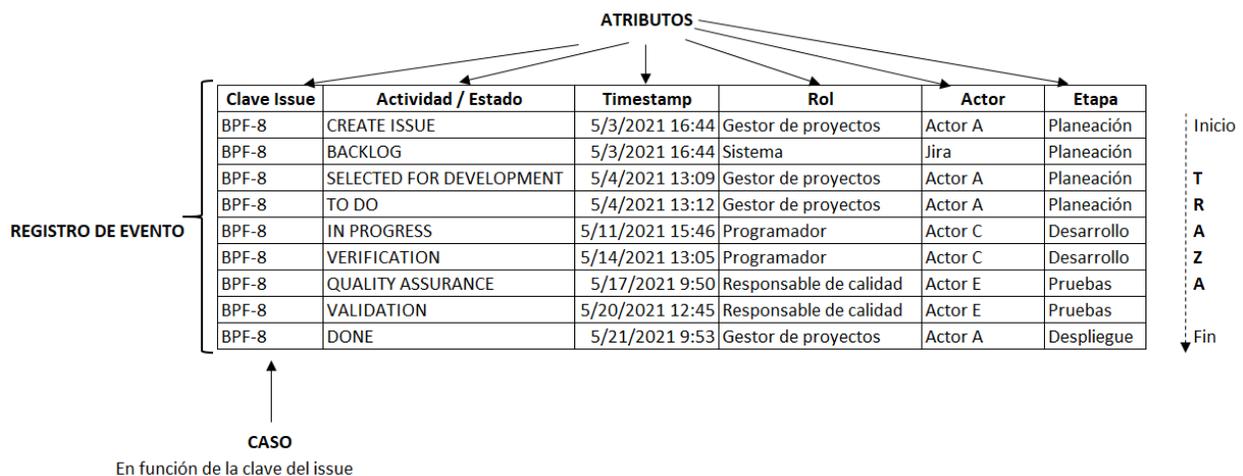


Figura 15. Fragmento del registro de eventos del proceso de desarrollo de software.

Con lo anterior, la obtención del registro de eventos inicial mostró un total de 488 eventos distribuidos en 62 casos que posterior a su depuración se obtuvo un total de 481 eventos sin la

eliminación de un caso. La depuración se debió a una duplicidad del estado de la actividad y el tiempo de ejecución (*timestamp*). Es decir, no hubo un cambio de estado. Por ejemplo, de *In Progress* a *In Progress*, *To Do* a *To Do* y *Done* a *Done*.

Extraer los registros de eventos de los dos repositorios de software Jira y SIMIo, así como su trazabilidad manual permitió extraer importantes conocimientos sobre el flujo de control. El reto de la obtención del flujo de control se encontró en la identificación de los casos y actividades. Asimismo, durante el proceso de desarrollo de software, se dificulta el reconocimiento de las actividades recurrentes cuando sólo se analizan los sistemas de control de versiones (Bala y Mendling, 2018). Por lo anterior, se resalta la importancia de seguir la ejecución de las actividades a través de los distintos repositorios donde el proceso de software deja evidencias de realización (registro de eventos).

5.2.2 Creación y descripción del modelo

En la segunda fase (Minería de Procesos), se aplicó la minería inductiva con el propósito de generar el modelo de procesos real en un formato visual representado por redes Petri. Este modelo se obtuvo con el soporte del plugin Mine Petri net with Inductive Miner de la herramienta ProM a partir del análisis de los registros de eventos. ProM permite el manejo de registro de eventos generando modelos BPMN, flujos de trabajo, redes Petri, descubrimiento de procesos, verificación de la conformidad y visualización de agrupamiento de trazas (Urrea-Contreras *et al.*, 2021). En la Figura 16, se muestra el modelo descubierto a través de una red petri y su diagrama BPMN con los diferentes flujos que los estados de transición de los *issues* presentaron de acuerdo al tablero Kanban.

Posteriormente, se analizaron las trazas con base en su similitud, de acuerdo al enfoque de perfiles, que utilizan las características de los eventos de cada traza, con el fin de descubrir modelos de procesos simples que en conjunto representan un modelo de procesos complejo (Reséndiz, 2019).

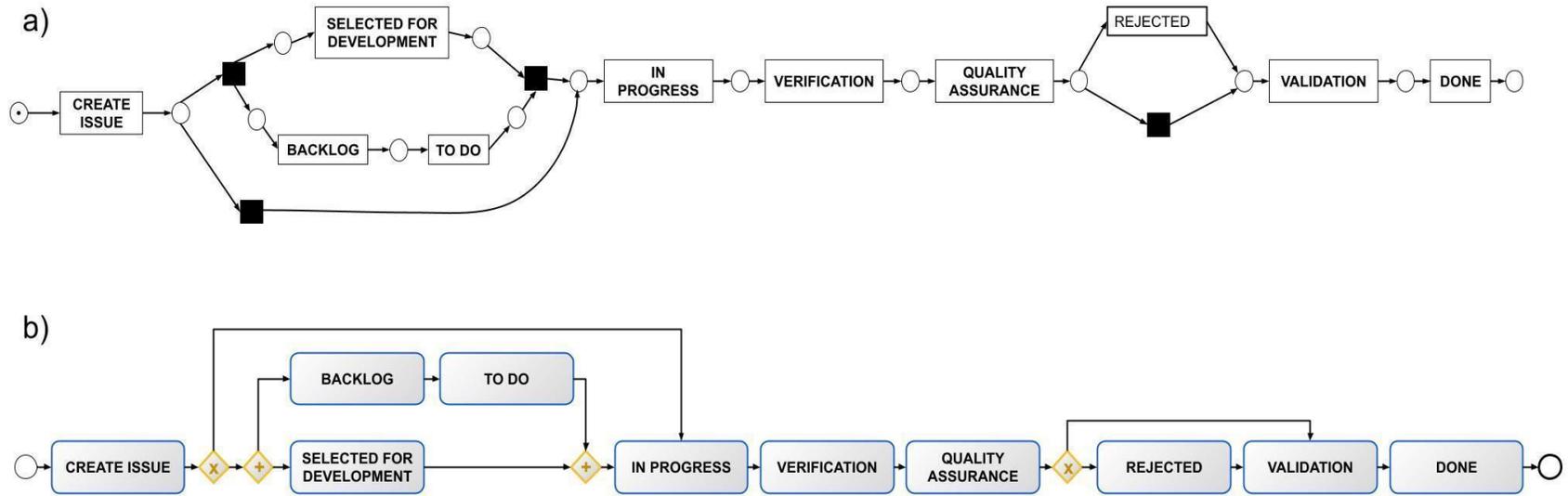


Figura 16. Modelo de procesos real de los estados de transición de los issues en Jira Proyecto 1
a) Red Petri, b) Diagrama BPMN.

A partir de la revisión del registro de eventos, se observó que la traza con menor longitud es la conformada por 3 eventos y la de mayor con 13 eventos. Mientras que un 56.45% de las trazas están compuestas por 6 o 9 eventos (Figura 17). Se identificó que la traza con mayor frecuencia está relacionada con 20 casos, lo que representa el 32.26% del registro de eventos.

Dicha traza está conformada por las actividades $\sigma_1 = \langle \text{Create Issue, In Progress, Verification, Quality Assurance, Validation, Done} \rangle$ (Figura 18).

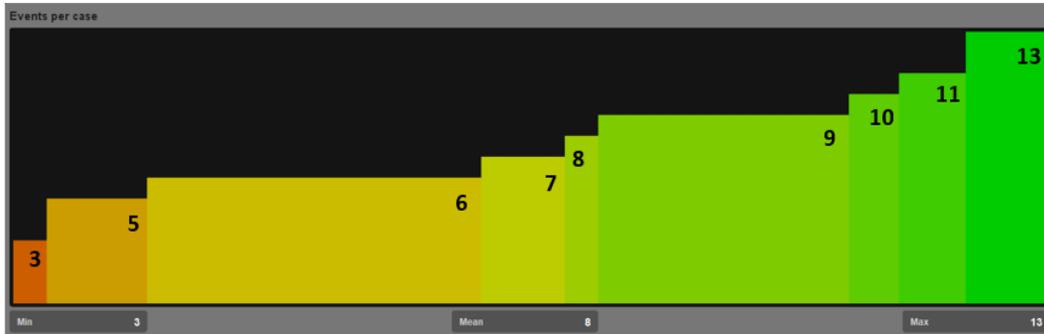


Figura 17. Histograma de eventos por trazas.



Figura 18. Secuencia de actividades de la traza con mayor presencia en el registro de eventos Proyecto 1.

En la Tabla 25 se muestran las 22 trazas² identificadas en el registro de eventos, se observa que la traza con mayor presencia corresponde a la solución de defectos (*Issue* tipo error) con un 32.26%, lo cual representa poco menos de un tercio del total del registro de eventos. Por otra parte, la segunda traza con mayor frecuencia corresponde a los *issues* de desarrollo (*Feature*) con un 24.19% lo que representa un cuarto del total del registro de eventos. Con respecto a las demás trazas, son ligeros cambios de seguimiento entre las trazas con mayor frecuencia, por ejemplo, para las trazas de *Issue* tipo error se puede ver la omisión o integración de los estados *In Progress*, *Quality Assurance* y *Rejected*. De igual manera, la traza de los errores es muy parecida a las trazas que corresponden a *Rework*, donde se integra los estados de *Selected For Development* y *To Do*. A lo que se refiere con la traza de Desarrollo (*Feature*), los cambios para originar nuevas trazas de debió al cambio repetitivo del *issue* entre los estados *Selected For Development* y *To Do*.

² Revisar Anexo 3 para consultar el conjunto de trazas totales con sus ID *Issues* correspondientes.

Tabla 25. Conjunto de trazas totales presentes en el registro de eventos Proyecto 1.
a= Create Issue, b= Backlog, c= Selected for Development, d= To Do, e= In Progress, f= Verification, g= Quality Assurance, h= Rejected, i= Validation y j= Done

Fase (Tipo de <i>issue</i>)	Traza	No. de casos	% del registro de eventos
Solución de defectos (Error) <i>Rework</i>	$\sigma_1 = \langle a, e, f, g, i, j \rangle$	20	32.26%
Desarrollo (<i>Feature</i>)	$\sigma_2 = \langle a, b, c, d, e, f, g, i, j \rangle$	15	24.19%
Solución de defectos (Error)	$\sigma_3 = \langle a, f, g, i, j \rangle$	5	8.06%
Desarrollo (<i>Feature</i>)	$\sigma_4 = \langle a, b, c, d, c, d, c, d, e, f, g, i, j \rangle$	2	3.23%
Solución de defectos (Error)	$\sigma_5 = \langle a, i, j \rangle$	2	3.23%
Solución de defectos (Error)	$\sigma_6 = \langle a, e, f, g, j, i, j \rangle$	2	3.23%
Desarrollo (<i>Feature</i>)	$\sigma_7 = \langle a, b, c, d, e, f, g, f, g, i, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_8 = \langle a, b, c, d, c, d, e, c, d, f, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_9 = \langle a, b, c, d, c, d, f, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{10} = \langle a, b, c, d, c, d, e, f, g, i, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{11} = \langle a, b, c, d, c, d, e, f, g, i, j \rangle$	1	1.61%
<i>Technical Labor</i>	$\sigma_{12} = \langle a, c, b, c, d, e, f, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{13} = \langle a, b, c, d, c, d, c, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{14} = \langle a, b, c, d, c, f, d, e, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{15} = \langle a, b, c, d, e, f, g, f, g, i, j \rangle$	1	1.61%
Solución de defectos (Error)	$\sigma_{16} = \langle a, e, f, i, j \rangle$	1	1.61%
<i>Rework</i>	$\sigma_{17} = \langle a, c, e, f, g, i, j \rangle$	1	1.61%
<i>Rework</i>	$\sigma_{18} = \langle a, c, d, e, f, g, i, j \rangle$	1	1.61%
Desarrollo (<i>Feature</i>)	$\sigma_{19} = \langle a, b, c, d, e, f, e, f, g, i, j \rangle$	1	1.61%
<i>Technical Labor</i>	$\sigma_{20} = \langle a, c, d, e, f, g, j \rangle$	1	1.61%
Solución de defectos (Error)	$\sigma_{21} = \langle a, e, f, g, i, j, i, j \rangle$	1	1.61%
Solución de defectos (Error)	$\sigma_{22} = \langle a, e, f, g, h, i, j \rangle$	1	1.61%
TOTAL		62	100%

5.2.3 Verificación de eventos y actividades

La verificación de la conformidad es el diagnóstico y la cuantificación de las discrepancias entre la realidad del proceso; es decir, la ejecución real registrada por los repositorios y los modelos de proceso (Mannhardt, 2018). De esta forma, la verificación de la conformidad verifica si la información que contienen los registros de eventos corresponde a los modelos organizacionales o da cumplimiento a políticas, regulaciones o reglas de negocio definidos previamente (TFPM-IEEE, 2012).

En la verificación de la conformidad se llevó a cabo con la aplicación de la técnica *Token Replay*, la cual diagnostica y cuantifica discrepancias, entre el comportamiento modelado y el observado (Van der Aalst, 2016). *Token Replay* mide el *fitness* a un nivel de eventos (eventos de una traza que se pueden reproducir en el modelo de procesos) brindando un resultado entre 0 y 1 donde 1 significa una condición perfecta de *fitness*. Un modelo tiene un *fitness* perfecto si todas las trazas del registro pueden ser reproducidas por el modelo de principio a fin (Van der Aalst, 2016).

Tomando en cuenta la relación del registro de eventos con las actividades del modelo de procesos descubierto, como resultado se obtuvo un *fitness* general de 0.935, es decir, alrededor del 6.5% de los eventos muestran desviaciones. Asimismo, el 61.3% de las trazas del registro se caracterizan por un nivel de *fitness* del 1. Para el resto de las trazas, en la Tabla 26 se observa el *fitness* resultante por cada traza del registro, seguido de las alineaciones de las actividades “*perdidas*” (*missing*) que se refieren a las actividades faltantes para que la traza pueda reproducirse a nivel óptimo, y las actividades “*erroneas*” (*wrong*) las cuales son aquellas actividades que presentan una doble reproducción en la traza y el modelo no lo permite, por lo que el movimiento de su transición se realiza a nivel registro y no en el modelo.

Las actividades *missing* (“perdidas”) fueron b= *Backlog*, e= *In Progress*, f= *Verification*, g= *Quality Assurance* y i= *Validation*, las cuales se observan en la Figura 19 de color amarillo representando su verificación de conformidad en el modelo y en la Tabla 27 con su porcentaje global de *fitness*. Los resultados de la Tabla 27 señalaron que la actividad *Backlog* cuenta con un *fitness* del 93.1%, *In Progress* 85.5%, *Verification* 93.5%, *Quality Assurance* 95.2% y *Validation* un 98.4%. Esto indica que la actividad *Validation* muestra desviaciones en 1.6% de sus eventos, lo cual se ve reflejado en la columna *missing* de la traza 20 al ser mencionado *Validation* solo una vez en la columna.

Al obtener un *fitness* del 93.5% con la verificación de conformidad, representando alrededor de 450 eventos reproducibles en el modelo de procesos descubierto, se concluye que el modelo queda validado para proceder a aplicar las perspectivas de Minería de Procesos según (Van der Aalst, 2016; Yang *et al.*, 2022).

Tabla 26. *Fitness* y alineaciones del conjunto de trazas totales presentes en el registro de eventos.
a= Create Issue, b= Backlog, c= Selected for Development, d= To Do, e= In Progress, f= Verification, g= Quality Assurance, h= Rejected, i= Validation y j= Done

Fase (Tipo de issue)	Traza	Fitness	Missing	Wrong
Solución de defectos (Error) <i>Rework</i>	$\sigma_1 = \langle a, e, f, g, i, j \rangle$	100%	-	-
Desarrollo (<i>Feature</i>)	$\sigma_2 = \langle a, b, c, d, e, f, g, i, j \rangle$	100%	-	-
Solución de defectos (Error)	$\sigma_3 = \langle a, e^*, f, g, i, j \rangle$	92.59%	e^*	-
Desarrollo (<i>Feature</i>)	$\sigma_4 = \langle a, b, c, d, c, d, c, d, e, f, g, i, j \rangle$	76.47%	-	c, d
Solución de defectos (Error)	$\sigma_5 = \langle a, e^*, f^*, g^*, i, j \rangle$	71.43%	e^*, f^*, g^*	-
Solución de defectos (Error)	$\sigma_6 = \langle a, e, f, g, j, i, j \rangle$	90.91%	-	j
Desarrollo (<i>Feature</i>)	$\sigma_7 = \langle a, b, c, d, e, f, g, f, g, i, g, i, j \rangle$	76.47%	-	f, g, i
Desarrollo (<i>Feature</i>)	$\sigma_8 = \langle a, b, c, d, c, d, e, c, d, f, g, i, j \rangle$	76.47%	-	c, d
Desarrollo (<i>Feature</i>)	$\sigma_9 = \langle a, b, c, d, c, d, e^*, f, g, i, j \rangle$	80.95%	e^*	c, d
Desarrollo (<i>Feature</i>)	$\sigma_{10} = \langle a, b, c, d, c, d, e, f, g, i, g, i, j \rangle$	76.47%	-	c, d, g, i
Desarrollo (<i>Feature</i>)	$\sigma_{11} = \langle a, b, c, d, c, d, e, f, g, i, j \rangle$	86.67%	-	c, d
<i>Technical Labor</i>	$\sigma_{12} = \langle a, c, b, c, d, e, f, g, i, j \rangle$	98.86%	-	c
Desarrollo (<i>Feature</i>)	$\sigma_{13} = \langle a, b, c, d, c, d, c, e^*, f^*, g, i, j \rangle$	69.05%	e^*, f^*	c, d
Desarrollo (<i>Feature</i>)	$\sigma_{14} = \langle a, b, c, d, c, f, d, e, f^*, g, i, j \rangle$	75.56%	f^*	c, d, f
Desarrollo (<i>Feature</i>)	$\sigma_{15} = \langle a, b, c, d, e, f, g, f, g, i, j \rangle$	86.67%	-	f, g
Solución de defectos (Error)	$\sigma_{16} = \langle a, e, f, g^*, i, j \rangle$	92.59%	g^*	-
<i>Rework</i>	$\sigma_{17} = \langle a, c, e, f, g, i, j \rangle$	90.91%	-	c
<i>Rework</i>	$\sigma_{18} = \langle a, b^*, c, d, e, f, g, i, j \rangle$	94.44%	b^*	-
Desarrollo (<i>Feature</i>)	$\sigma_{19} = \langle a, b, c, d, e, f, e, f, g, i, j \rangle$	86.67%	-	e, f
<i>Technical Labor</i>	$\sigma_{20} = \langle a, b^*, c, d, e, f, g, i^*, j \rangle$	87.88%	b^*, i^*	-
Solución de defectos (Error)	$\sigma_{21} = \langle a, e, f, g, i, j, i, j \rangle$	83.33%	-	i, j
Solución de defectos (Error)	$\sigma_{22} = \langle a, e, f, g, h, i, j \rangle$	100%	-	-

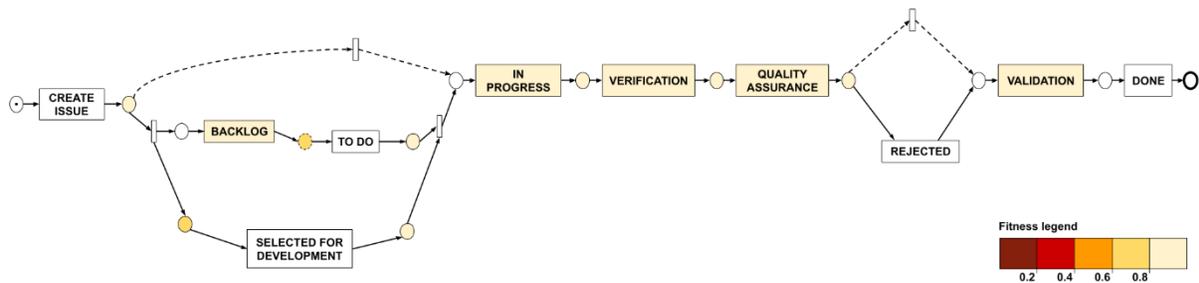


Figura 19. Resultados de la verificación de la Conformidad representado por el flujo de los estados de las actividades.

Tabla 27. Porcentaje de la verificación de la conformidad con respecto a la actividad Proyecto 1.

Estado/Actividad	Porcentaje de <i>fitness</i>
<i>Backlog (b)</i>	93.1%
<i>In Progress (e)</i>	85.5%
<i>Verification (f)</i>	93.5%
<i>Quality Assurance (g)</i>	95.2%
<i>Validation (i)</i>	98.4%

5.2.4 Extensión del Modelo de Procesos con las perspectivas

Perspectiva de Flujo de Control

En la perspectiva de flujo de control podemos observar el orden de las actividades que dentro del registro de eventos extraído son representadas por los estados de los *issues*. Como primera actividad, se realizó un análisis de asociación con las etapas del ciclo de vida de desarrollo de software y las actividades comprendidas en el registro de eventos (Tabla 28). En la Figura 20 se observa que se ejecutaron 3 iteraciones en el proyecto.

Tabla 28. Asociación del flujo de desarrollo del tablero Kanban con las etapas del ciclo de vida del desarrollo de software.

Estado / Actividad	Etapas
<i>Create Issue</i>	Planeación
<i>Backlog</i>	Planeación
<i>To Do</i>	Planeación
<i>Selected For Development</i>	Planeación
<i>In Progress</i>	Desarrollo
<i>Verification</i>	Desarrollo
<i>Quality Assurance</i>	Pruebas
<i>Validation</i>	Pruebas
<i>Rejected</i>	Pruebas
<i>Done</i>	Despliegue

En la Figura 21 se encuentra el modelo de procesos resultante donde se visualiza las actividades con mayor ejecución son *Create Issue*, *In Progress*, *Verification*, *Quality Assurance*, *Validation* y *Done* que corresponden a la traza con mayor frecuencia. Asimismo, la actividad con menor ejecución es *Rejected*, la cual solo se ejecutó una vez en el registro de eventos. En la Figura 22, se puede visualizar la incorporación de los flujos para que el modelo obtenga una verificación de la conformidad completa (100%). Los flujos incorporados están basados en la identificación de las trazas y las transiciones muestran la cantidad de veces que se utilizó el flujo.



Figura 20. Modelo de flujo de control de las etapas del ciclo de vida de desarrollo de software.

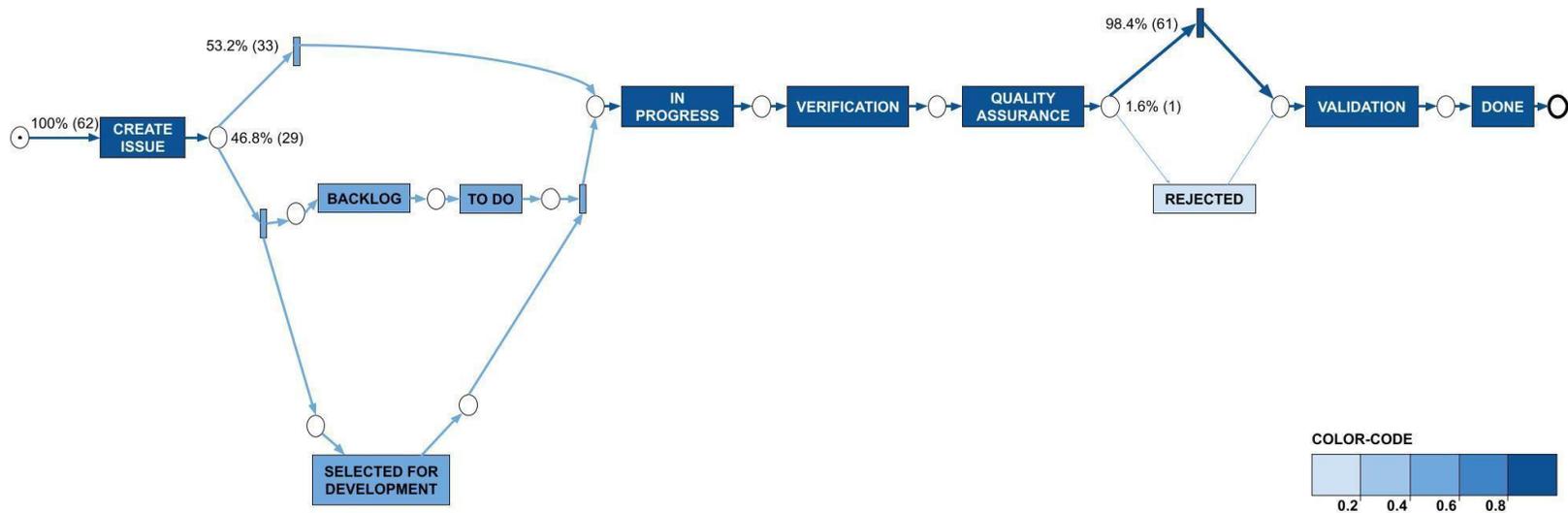


Figura 21. Modelo de flujo de control de las actividades (estados del *issue* de Jira).

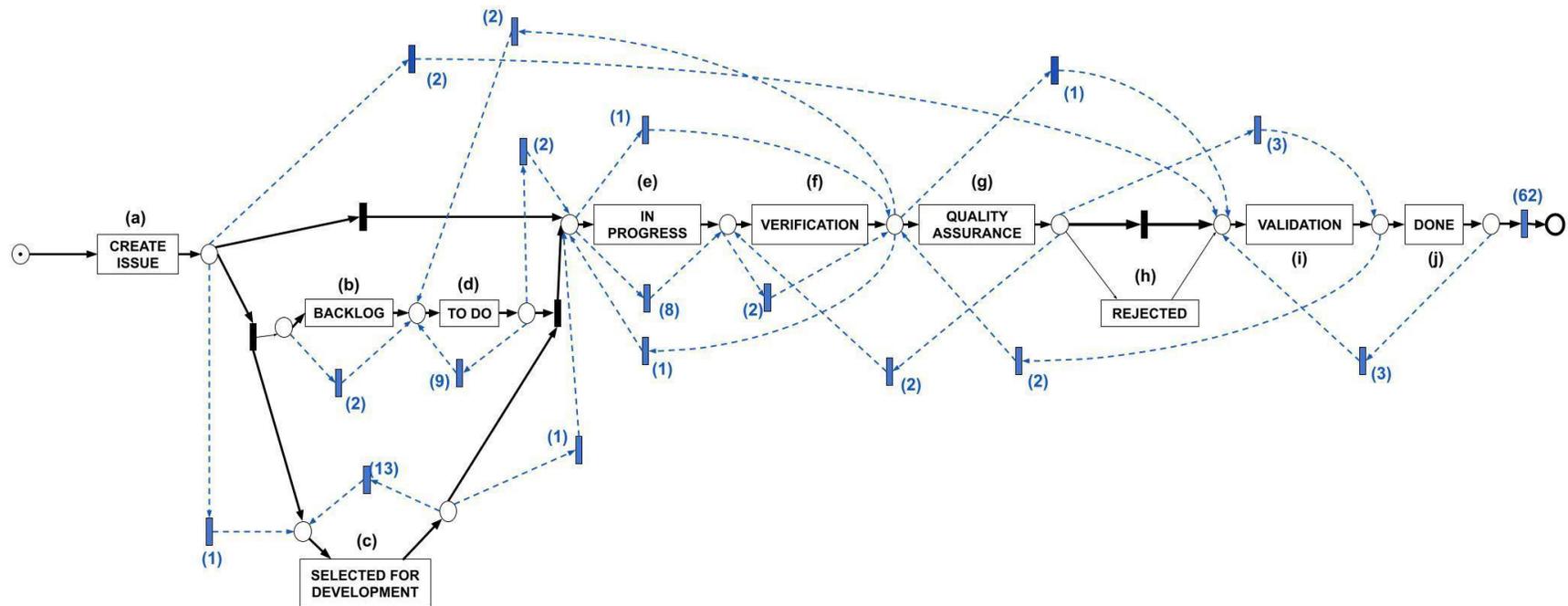


Figura 22. Modelo de flujo de control (Conformidad al 100%).

Perspectiva del Caso

Para el análisis de la perspectiva del caso, se estableció cada *issue* como un caso (traza), del cual se obtuvo que la traza con mayor frecuencia en el registro de eventos (Figura 23) corresponde a los *issues* etiquetados como *Error*, los cuales en la mayoría de los casos pasan a un estado de *In Progress* posterior a su creación para resolver el *issue*. De igual manera, se obtuvo la segunda traza con mayor presencia (15 trazas que representan el 24.19% del registro de eventos) que corresponde a los *issues* etiquetados como *Feature*. Como se observa en la Figura 23, los *features* conllevan un proceso que detalla con mayor claridad las etapas del desarrollo de software. Esto último se ve reflejado por el flujo de la traza y del diagrama del proceso de desarrollo de la empresa de software al ser semejantes. Si se toma el proceso de desarrollo completo descrito por la empresa, se deduce que los registros presentes en los repositorios detallan información a partir del desarrollo Kanban hasta que se realiza un despliegue en el ambiente de producción.



Figura 23. Secuencia de actividades de la segunda traza con mayor presencia en el registro de eventos Proyecto 1 (*Feature*).

Perspectiva Organizacional

La minería organizacional puede utilizarse para obtener información sobre los patrones de trabajo típicos, las estructuras organizativas y las redes sociales. Se basa en el atributo de recurso que puede contener información sobre máquinas, roles, departamentos y distribución de recursos (Gupta *et al.*, 2014, Van der Aalst, 2016).

La perspectiva organizacional se relaciona con los roles que están presentes en el registro de eventos. Para su análisis se utilizó la herramienta Disco la cual permitió la visualización tanto de los roles como de los actores. Por fines de confidencialidad solo se demostrará lo relacionado con los roles. En la Figura 24 se observa la frecuencia relativa de los roles con respecto a la traza con mayor frecuencia, dentro de la cual se indica el rol de programador tiene la mayor con un 50%, seguido del responsable de calidad con un 32.29%. Para esta traza, se encuentra el rol de sistema

con un 12.5% cuyas actividades son de manera automática, después se encuentra el Gestor de proyectos con 3.12% y por último el líder técnico con 2.08%.

Los resultados de una mayor participación por parte de los programadores son debido a que la información se extrajo de los repositorios en los que se registran las actividades realizadas por los mismos. Un ejemplo de este punto es el reporte generado por SIMIo el cual solo brindó información de los programadores debido a que se encontraba con una limitante en su versión, de la cual se indicó que para proyectos posteriores se vinculó con Jira para registrar las actividades de todos los involucrados en el proyecto.

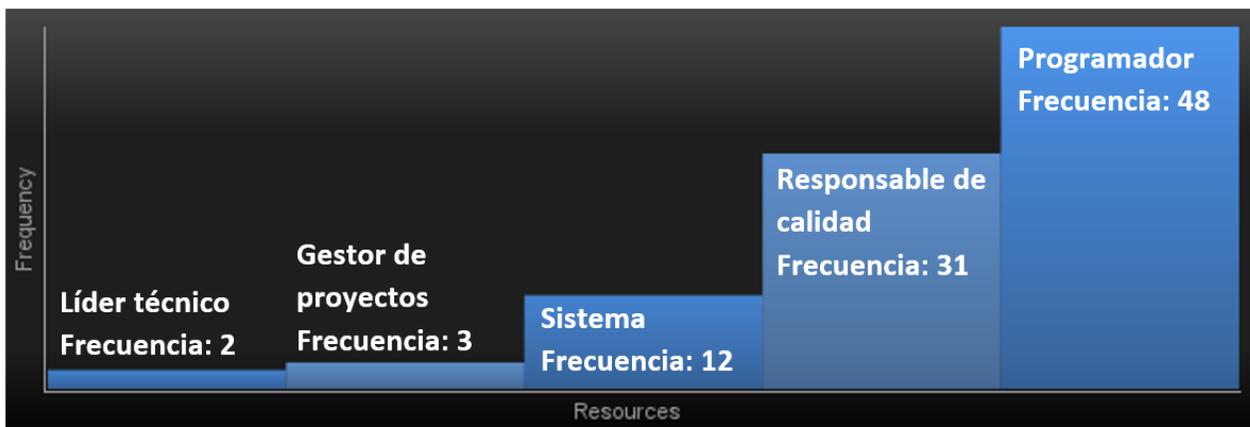


Figura 24. Frecuencia y porcentaje de participación Proyecto 1.

Perspectiva del Tiempo

La perspectiva del tiempo corresponde a la medición del tiempo obtenido por el análisis del atributo *timestamp*. Los elementos del *timestamp* del registro corresponden a MM/DD/AAAA HH:MM, teniendo una precisión de minutos. Se obtuvo el tiempo promedio entre actividades como se observa en la Figura 25. En la Tabla 29 se incluye también el tiempo mínimo y el tiempo máximo de las actividades. Se establece como tiempo promedio entre *Create Issue* y *Backlog* 4.4 segundos, lo cual sucede porque *Backlog* es una actividad automática que realiza el sistema posterior a la creación de un *issue*. Asimismo, *Backlog* obtuvo como tiempo máximo 1 minuto y como mínimo se tomó 0 debido a que el formato de *timestamp* no contempla segundos, ocasionando que las actividades con el mismo registro de *timestamp* que su estado antecesor se

tuviera que reflejar con 0 su tiempo de duración. Por ejemplo, en el *issue* BPF-8 el cambio de estado entre *Create Issue* a *Backlog* se realizaron en el mismo minuto (5/3/2021 16:44), a diferencia del *issue* BPF-20 el cual se realizó un minuto después el cambio de estado; *Create Issue* (5/4/2021 13:10) – *Backlog* (5/4/2021 13:11). Lo mismo ocurre en actividades como *In Progress*, *Verification* y *Done*, las cuales el cambio de su estado se realiza por acción de un rol.

En la Tabla 29 se observa que las actividades con mayor duración son *Selected For Development*, *To Do* y *Done*. Las dos primeras corresponden a que su espera de la transición a la siguiente etapa va ligada con los tiempos establecidos en el proyecto y las iteraciones del mismo. El estado *Done*, permanece hasta que el cliente apruebe el entregable, posteriormente se envía a producción. Por otra parte, la actividad de *In Progress* muestra una duración promedio de 1.3 días en comparación con el tiempo de 8.7 días en el estado de *To Do*.

Tabla 29. Especificación del Tiempo promedio, máximo y mínimo por actividad Proyecto 1.

Estado/Actividad	Tiempo		
	Promedio	Máximo	Mínimo
<i>Create Issue</i>	-	-	-
<i>Backlog</i>	4.4 s	1 m	-
<i>Selected For Development</i>	5 d	16.7 d	1 m
<i>To Do</i>	8.7 d	21.7 d	1 m
<i>In Progress</i>	1.3 d	7.1 d	-
<i>Verification</i>	1.1 d	10.9 d	-
<i>Quality Assurance</i>	10.8 h	4.9 d	6 m
<i>Rejected</i>	18 h	18 h	18 h
<i>Validation</i>	1.9 d	8.8 d	2 m
<i>Done</i>	8.6 d	22 d	-

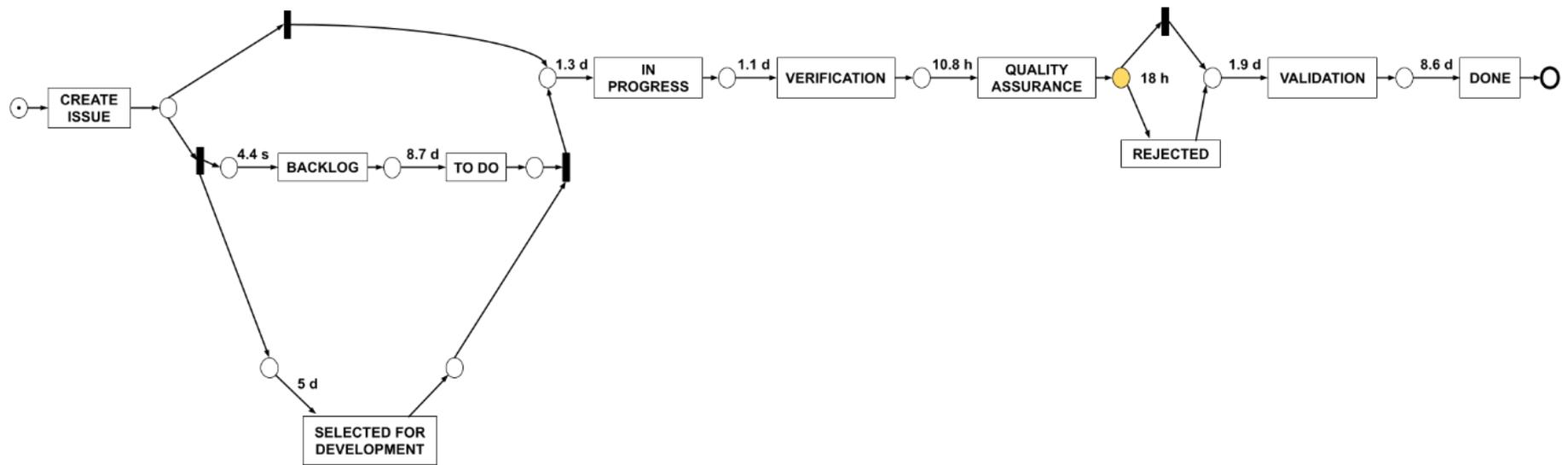


Figura 25. Modelo de procesos con tiempo promedio entre actividades.

5.3 Análisis del Proyecto 2

El proyecto 2 corresponde a una aplicación Web para la gestión de servicios de becarios que incluye las funcionalidades de: registro y gestión de afiliados, catálogo de afiliaciones, registro de cambios, gestión de nóminas, entre otras. Este proyecto se inició en diciembre de 2021 y aún se encuentra en desarrollo. El equipo del proyecto estuvo conformado por los roles de: Gestor de proyectos, Líder Técnico, Programador y Responsable de Calidad.

5.3.1 Obtención del Registro de Eventos

La información se extrajo de MongoDB definiendo consultas para seleccionar atributos del registro de eventos. Como se mencionó anteriormente, el estado del *issue* se estableció como una actividad, lo que permitió relacionar cada instancia de actividad con un caso. Para el proyecto 2, se identificaron 9 actividades que tienen lugar durante el proceso de desarrollo de software y que están registradas en los sistemas de información. Cada actividad representa una tarea específica y se registró cada vez que se realizó durante el proceso de desarrollo. La Tabla 30 muestra las actividades del registro de eventos ordenadas de mayor a menor frecuencia. La actividad más frecuente fue "*In progress*", seguida de "*Quality Assurance*" y "*Verification*". Con lo anterior, el registro de eventos identificó un total de 4055 eventos en 702 casos, con los atributos ID Issue, Activity, Timestamp y Role and Actor.

Tabla 30. Actividades en el registro de eventos del proyecto de software ordenados por su frecuencia Proyecto 2.

Actividad	Frecuencia
<i>In Progress</i>	724
<i>Quality Assurance</i>	683
<i>Verification</i>	661
<i>Done</i>	441
<i>Validation</i>	362
<i>Selected for Development</i>	361
<i>To Do</i>	349
<i>Backlog</i>	275
<i>Rejected</i>	199
<i>Total:</i>	4055

5.3.2 Creación y descripción del modelo

Se aplicó la minería inductiva con el objetivo de generar el modelo de procesos real en un formato visual, representadas por medio de redes Petri. Para ello, se utilizó la herramienta ProM, la cual permitió analizar los registros de eventos obtenidos. En la Figura 26 se muestra el modelo descubierto a través de una red Petri y su diagrama BPMN con los diferentes flujos que los estados de transición de los issues presentaron de acuerdo al tablero Kanban.

Una vez obtenido el modelo de procesos, se analizaron las trazas generadas a partir del registro de eventos. En total se identificaron 144 trazas, donde se observó que la traza con menor longitud estaba conformada por 1 evento, lo que indica que todavía no llega a su finalización, mientras que la traza de mayor longitud estaba compuesta por 17 eventos. Además, se identificó que la traza *Issue* tipo error (*Error*) es la de mayor frecuencia con 247 casos (Tabla 31), lo que representa el 35.19% del registro de eventos. Esta traza está conformada por las actividades $\sigma_1 = \langle \text{In progress, Verification, Quality Assurance, Done} \rangle$ (Figura 27).

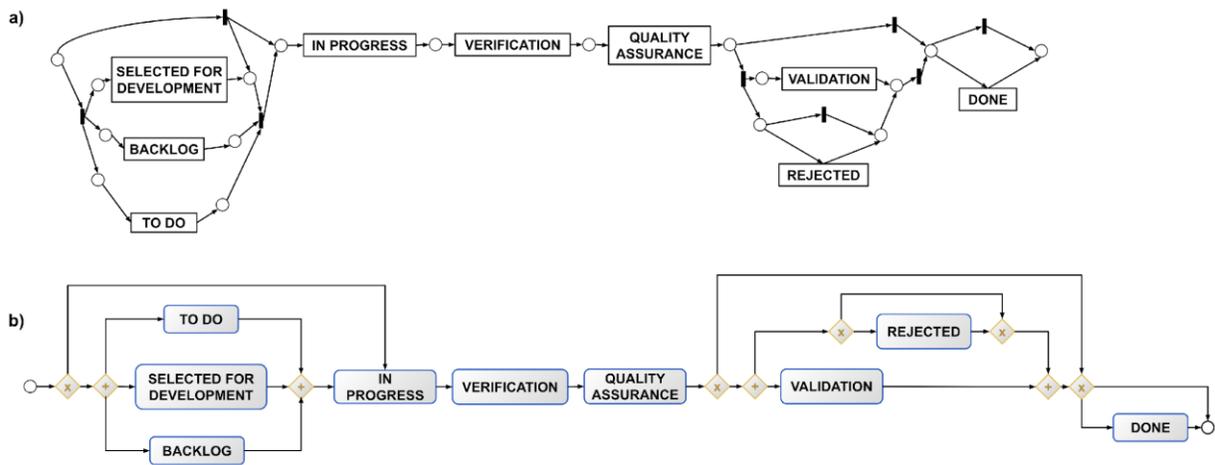


Figura 26. Modelo de procesos real de los estados de transición de los issues en Jira Proyecto 2
a) Red Petri, b) Diagrama BPMN.



Figura 27. Secuencia de actividades de la traza con mayor presencia en el registro de eventos Proyecto 2.

Tabla 31. Conjunto de trazas totales presentes en el registro de eventos Proyecto 2.
a= Backlog, b= Selected for Development, c= To Do, d= In Progress, e= Verification, f= Quality Assurance, g= Rejected, h= Validation y i= Done

Traza	No. de casos	% del registro de eventos
$\sigma_1 = \langle d, e, f, i \rangle$	247	35.19
$\sigma_2 = \langle a, b, c, d, e, f, h \rangle$	45	6.41
$\sigma_3 = \langle a, b, c, d, e, f, g, h \rangle$	40	5.70
$\sigma_4 = \langle d, e, f, h \rangle$	37	5.27
$\sigma_5 = \langle a, b, c, d, e, f, g, h, i \rangle$	21	2.99
$\sigma_6 = \langle a, b, c, d, e, f, h, i \rangle$	18	2.56
$\sigma_7 = \langle b, c, d, e, f, h \rangle$	16	2.28
$\sigma_8 = \langle a \rangle$	13	1.85
$\sigma_9 = \langle d, c, d, e, f, i \rangle$	12	1.71
$\sigma_{10} = \langle d, e, f, g, h \rangle$	12	1.71
$\sigma_{11} = \langle b, c, d, e, f, i \rangle$	9	1.28
$\sigma_{12} = \langle d, e, f, g \rangle$	9	1.28
$\sigma_{142} = \langle a, b, c, d, e, f, g, i, h, i \rangle$	1	0.14
$\sigma_{143} = \langle a, b, c, d, e, f, g, i \rangle$	1	0.14
$\sigma_{144} = \langle b, d, e, f, i \rangle$	1	0.14

5.3.3 Verificación de eventos y actividades

Se realizó la verificación de la conformidad utilizando la técnica Token Replay. Se evaluó la relación del registro de eventos con las actividades del modelo de procesos. El resultado general obtenido del análisis de Token Replay fue de 0.894 (89.4%), lo que indica que alrededor del 10.6% de los eventos presentan desviaciones en comparación con el modelo de procesos.

Del mismo modo, la Tabla 32 muestra los resultados de Fitness relativos a cada actividad. En concreto, muestra que *Backlog* tiene un *Fitness* del 83.6%, *Verification* 90.3%, *Validation* 90.7%, *Quality Assurance* 92.7%, *In progress* 94.2%, *To Do* 96.2% y *Selected for development* 99.7%. Esto indica que faltan actividades que impiden que se reproduzcan de forma óptima en las trazas y pueden afectar al flujo del proceso.

Tabla 32. Porcentaje de la verificación de la conformidad con respecto a la actividad Proyecto 2.

Estado/Actividad	Porcentaje de <i>fitness</i>
<i>Backlog</i>	83.6%
<i>Verification</i>	90.3%
<i>Validation</i>	90.7%
<i>Quality Assurance</i>	92.7%
<i>In Progress</i>	94.2%
<i>To do</i>	96.2%
<i>Selected for development</i>	99.7%
<i>Rejected</i>	100%
<i>Done</i>	100%

5.3.4 Extensión del Modelo de Procesos con las perspectivas

Perspectiva del Caso

En el análisis de la perspectiva del caso, cada *issue* se consideró como un caso o traza individual. Como se observó inicialmente en el modelo descubierto, los resultados muestran que la traza con mayor frecuencia en el registro de eventos corresponde a problemas de tipo Error, y la segunda traza más presente (que representa el 6.09% del registro) corresponde a problemas de tipo *Feature*. La Figura 28 muestra que los *issue* de tipo *Feature* implican un proceso más detallado que describe con mayor claridad las etapas del desarrollo de software. También puede observarse que la traza no incluye la actividad *Done*, lo que refleja que algunas tareas relacionadas con el proyecto siguen en curso y no han alcanzado su estado final. Esto último se debe a que el Proyecto aún no ha finalizado.



Figura 28. Secuencia de actividades de la segunda traza con mayor presencia en el registro de eventos Proyecto 2 (*Feature*).

Perspectiva Organizacional

Desde el punto de vista de la perspectiva organizacional, la Tabla 33 muestra la frecuencia de los roles con respecto a la traza más frecuente, en la que el rol de programador tiene la mayor frecuencia con un 44.16%, seguido del responsable de calidad con un 23%, el líder técnico con un 14.40% y el gestor de proyectos con un 12.25%. Para ciertas trazas, el rol de sistema está presente con un 6.14% cuyas actividades son automatizadas. Esto se debe a que Jira, como herramienta de gestión de proyectos, se encuentra automatizada.

Tabla 33. Frecuencia y porcentaje de participación Proyecto 1.

Rol	Frecuencia de eventos	Porcentaje de participación
Programador	1791	44.16%
Responsable de calidad	934	23.03%
Líder técnico	584	14.40%
Gestor de proyectos	497	12.25%
Sistema	249	6.14%
Total	4055	100%

Perspectiva del Tiempo

La perspectiva del tiempo se refiere a la medición del tiempo obtenido mediante el análisis del atributo timestamp. En el registro, los elementos de timestamp corresponden a MM/DD/AAAA HH:MM:SS. Se obtuvo el tiempo promedio, mínimo y máximo de las actividades que se presentan en la Tabla 34. Las actividades con mayor duración son *Backlog*, *Selected for development*, *To Do* y *Validation*, mientras que el estado *Done* también se considera una actividad con mayor duración porque permanece hasta que el cliente aprueba el entregable antes de ser enviado a producción.

Tabla 34. Especificación del Tiempo promedio, máximo y mínimo por actividad Proyecto 2.

Estado/Actividad	Tiempo		
	Promedio	Máximo	Mínimo
<i>Backlog</i>	2.2 d	134 d	-
<i>Selected For Development</i>	8.7 d	130.2 d	-
<i>To Do</i>	10.8 d	133.1 d	-
<i>In Progress</i>	15.1 h	18 d	-
<i>Verification</i>	16.9 h	26.2 d	1 s
<i>Quality Assurance</i>	9.6 h	16.1 d	1 s

<i>Rejected</i>	5 d	27.1 d	11 m
<i>Validation</i>	10.3 d	112.3 d	11.8 m
<i>Done</i>	4.9 d	52.8 d	-

5.4 Comparación de proyectos

5.4.1 Modelo Integrado

La Minería de Procesos permitió a la MiPyME identificar los cambios implementados hasta el momento entre el periodo de desarrollo del Proyecto 1 (Figura 29) y el Proyecto 2 (Figura 30). Aunque estos cambios aún están en proceso de integración, su identificación temprana permite trabajar en su mejora para lograr una mayor eficacia en las operaciones de la MiPyME. La aplicación de técnicas y perspectivas de Minería de Procesos permite tomar decisiones basadas en datos para mejorar su proceso de desarrollo de software y optimizar su rendimiento.

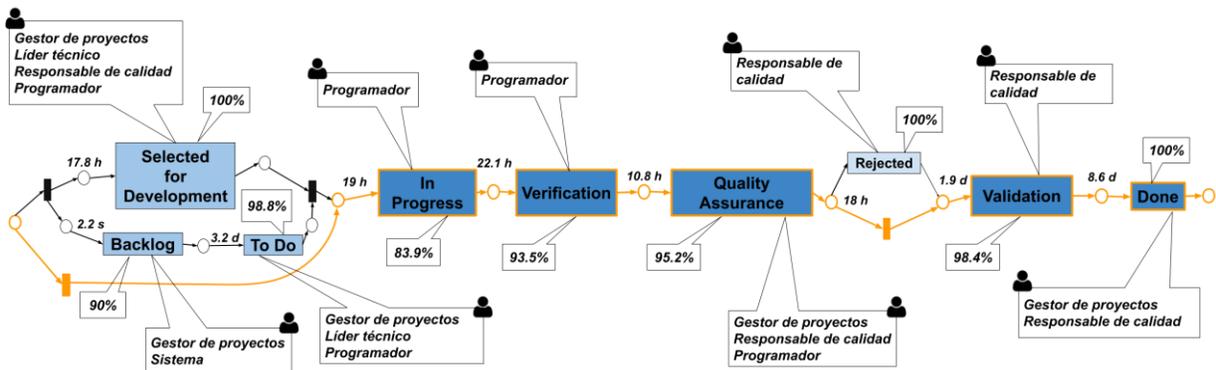


Figura 29. Modelo integrado del Proyecto 1.

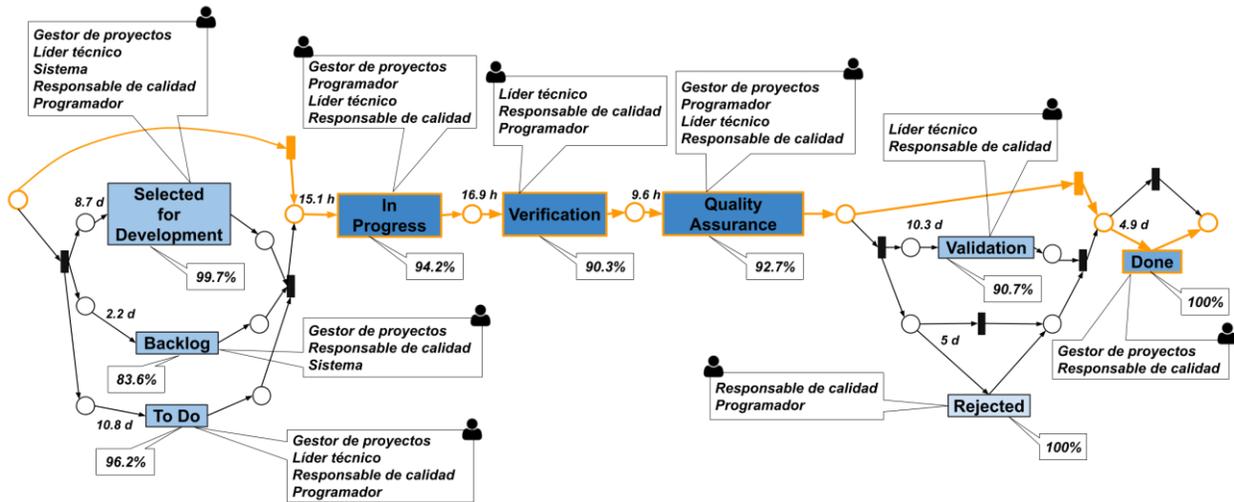


Figura 30. Modelo integrado del Proyecto 2.

La Tabla 35 presenta una comparación entre el Proyecto 1 y el Proyecto 2 en lo que respecta a la extracción de datos, los eventos, la identificación de la traza, la verificación de la conformidad y las perspectivas. En cuanto a la extracción de datos, en el Proyecto 1 se estableció la trazabilidad manual entre eventos, mientras que en el Proyecto 2 se desarrolló un script para automatizar las actividades ETL. En la identificación de trazas, el Proyecto 2 identificó un número significativamente mayor de trazas en el registro de eventos en comparación con el Proyecto 1, lo que sugiere una mayor diversidad o complejidad en el flujo de actividades en el segundo proyecto. En la verificación de la conformidad, ambos proyectos obtuvieron resultados globales satisfactorios, aunque el Proyecto 1 obtuvo un porcentaje ligeramente superior (93.5%) en comparación con el 89.4% del Proyecto 2.

Desde la perspectiva del flujo de control, el Proyecto 1 no reveló incoherencias, mientras que el Proyecto 2 identificó una incoherencia en la actividad de *Validation* en la traza más ejecutada, lo que indica una posible área de mejora en ese sentido. Desde la perspectiva del caso, en ambos proyectos se descubrió que las actividades del proceso de desarrollo de software se centran principalmente en los *issues* del tipo error: Proyecto 1 (32.26%) y Proyecto 2 (35.19%). En segundo lugar, a los *issues* tipo *Feature*: Proyecto 1 (25.19%) y Proyecto 2 (6.41%). Desde la perspectiva organizacional, a diferencia del Proyecto 2, experimentó una elevada rotación de personal y asignación de funciones. Por último, en la perspectiva del tiempo, teniendo en cuenta que la actividad *Done* es relevante para las tareas completadas y listas para su entrega y despliegue,

el Proyecto 1 tuvo una mayor duración con un tiempo medio (8.6 d) en comparación con el Proyecto 2 (4.9 d), lo que puede deberse a que, en el Proyecto 2, las tareas se liberaron sin ser validadas.

Tabla 35. Comparativa entre el Proyecto 1 y el Proyecto 2.

Campo	Proyecto 1	Proyecto 2
Extracción de datos	Trazabilidad manual entre eventos.	Desarrollo de un script para las actividades ETL.
Eventos	481 eventos.	4055 eventos.
Identificación de trazas	22 trazas identificadas en el registro de eventos.	144 trazas identificadas en el registro de eventos.
Verificación de la conformidad	Resultado global de <i>fitness</i> 0.935 (93.5%).	Resultado global de <i>fitness</i> 0.894 (89.4%).
Perspectiva de Flujo de control	Descubrimiento del flujo real de la empresa.	Inconsistencia en la actividad de <i>Validation</i> en la traza más ejecutada.
Perspectiva del caso	<i>Issues</i> tipo <i>error</i> (32.26%). <i>Issues</i> tipo <i>Feature</i> (35.19%).	<i>Issues</i> tipo <i>error</i> (25.19%). <i>Issues</i> tipo <i>Feature</i> (6.41%).
Perspectiva organizacional	No hubo cambios en la asignación de roles.	Rotación y cambio de asignación de roles.
Perspectiva del tiempo	Mayor tiempo promedio de duración en la actividad <i>Done</i> (8.6 d).	Menor tiempo promedio de duración en la actividad <i>Done</i> (4.9 d).

Al comparar los roles con la perspectiva organizacional en el Proyecto 2, se observan cambios tanto en el equipo de proyecto como en dirección. Por lo tanto, incluso cuando las MiPyMEs establecen un mejor flujo en sus actividades, los cambios de personal siguen siendo un reto (Eggert y Dyong, 2022). En este sentido, se identificó como reto la pérdida de conocimientos y experiencia del personal, que afecta a la correcta ejecución del modelo de procesos y, en consecuencia, a la calidad del producto del trabajo. Para superar este reto, es importante que las MiPyMEs cuenten con estrategias para integrar de forma eficiente a los nuevos miembros del equipo, así como para retener y aprovechar el talento existente. Otro reto es la falta de documentación relacionada con los procesos (Wijnhoven et al., 2023). La documentación permite transferir conocimientos explícitos entre los miembros del equipo. Durante la extracción del Proyecto 1, no existía un documento formal del proceso de desarrollo de software, por lo que la MiPyME tuvo que documentarlo antes de iniciar las actividades de extracción.

Del mismo modo, el registro de las actividades de los procesos a través de su flujo entre varios sistemas resulta ser todo un reto. En este sentido, durante el pre-procesamiento, se produjo una limitación en la extracción (Wijnhoven et al., 2023) porque los registros se encontraban en varios sistemas sin ninguna interfaz entre ellos. La falta de APIs, por tanto, se ha considerado otro reto a asumir (Vidoni, 2022). Por otro lado, al tener los datos en formatos diferentes, el siguiente reto fue preparar los datos del registro de eventos (Eggert y Dyong, 2022) para proporcionar un formato único y estandarizado. Esto también se debe al reto de la variabilidad de los registros entre herramientas o sistemas de apoyo al desarrollo de software (Vavpotič et al., 2022), como Jira y Github, que están vinculados pero se almacenan de forma diferente.

Además, otro reto a destacar entre el Proyecto 1 y el Proyecto 2 fue la comunicación. En el Proyecto 1, la comunicación era abierta y constante a través de reuniones, lo que ayudó a resolver las dudas e inquietudes que surgían sobre la Minería de Procesos. Mantener este nivel de comunicación permitió afrontar con éxito el reto de crear conciencia (Eggert y Dyong, 2022), es decir, lograr que la alta dirección y otros miembros del equipo comprendieran el proceso y comunicaran los resultados (Eggert y Dyong, 2022) con el fin de aplicarlos a la mejora de los procesos de la MiPyME. Sin embargo, en el Proyecto 2 la comunicación fue inadecuada, principalmente a través de mensajes y correo electrónico. Esto se debió al cambio en la dirección del proyecto, que priorizó el cumplimiento de la entrega al cliente. Lo que dificultó la interpretación de los datos, ya que se requiere un alto nivel de conocimiento de la unidad de negocio (Eggert y Dyong, 2022). Por tanto, un reto importante es la capacidad de las MiPyME para aplicar mejoras en los procesos. Esto puede requerir un cambio significativo en la cultura de la MiPyME, así como la asignación de recursos y la formación del personal.

La Minería de Procesos permitió a la MiPyME identificar los cambios implementados hasta el momento entre el periodo del desarrollo del Proyecto 1 y del Proyecto 2. Aunque estos cambios aún están en proceso de integración, su identificación temprana permite trabajar en la mejora para lograr una mayor efectividad en las operaciones de la MiPyME. En general, la aplicación de técnicas y perspectivas de la Minería de Procesos le permite a la MiPyME tomar decisiones con información basada en sus datos para mejorar su proceso de desarrollo de software y optimizar su rendimiento.

Capítulo 6. Conclusiones y Trabajo Futuro

6.1 Conclusiones

La Minería de Procesos es una disciplina fundamental para las MiPyMEs que buscan mejorar sus procesos de desarrollo de software. A través del análisis de trazas y la visualización de los flujos de trabajo, se pueden determinar oportunidades de mejora que permiten optimizar los procesos y aumentar la eficiencia y efectividad de las MiPyMEs.

Además de la optimización de procesos, la Minería de Procesos también puede ayudar a las MiPyMEs a identificar oportunidades de innovación y crecimiento. Al analizar las trazas de eventos, es posible identificar patrones emergentes, tendencias y posibles áreas de expansión. Esto puede abrir nuevas posibilidades de negocio, permitiendo a las MiPyMEs desarrollar nuevos productos o servicios, explorar nuevos mercados o implementar estrategias de diferenciación. La Minería de Procesos proporciona a las MiPyMEs una visión integral de sus procesos y les permite aprovechar el potencial de innovación y crecimiento.

Se propuso la adaptación de la Metodología L* mediante un enfoque de Ingeniería de Software por capas permitiendo el minado de dos proyectos de software. Asimismo, se introdujeron notaciones matemáticas formales de los elementos de la Minería de Procesos. La combinación de estos elementos ofrece un enfoque teórico y práctico para abordar la Minería de Procesos en el ámbito de desarrollo de software a través de diferentes áreas y temas de conocimiento para la identificación y análisis de escenarios. Esto brinda un marco de referencia para investigadores y profesionales del campo que desean mejorar sus prácticas en el ámbito.

La extensión de los modelos de procesos con las perspectivas de Minería de Procesos permitió obtener una comprensión más completa del proceso de desarrollo de software de la MiPyME. Las perspectivas del flujo de control, caso, organizacional y del tiempo son importantes para tener una

visión holística del proceso y permitir una mejor toma de decisiones por parte de la alta dirección y el equipo de mejora.

Se identificaron retos como la falta de documentación, la extracción de los datos, el cambio constante de personal y la variabilidad de los registros en los sistemas de desarrollo de software, que son algunos de los obstáculos que las MiPyMEs deben enfrentar al implementar la Minería de Procesos. Para superar estos retos, es necesario establecer una comunicación constante entre los diferentes actores involucrados en el proceso, hasta la alta dirección. La comunicación es clave para garantizar una interpretación precisa de los datos y para asegurar que la Minería de Procesos esté alineada con los objetivos y metas de la organización.

Es importante destacar que la Minería de Procesos no es una solución única y definitiva, sino un proceso continuo de mejora. Los cambios implementados en los procesos se encuentran en constante integración, lo que puede llevar al surgimiento de flujos no óptimos. Por lo tanto, es necesario que de forma iterativa se identifiquen oportunidades de mejora para lograr una mayor efectividad en las operaciones de las MiPyMEs.

6.2 Trabajo Futuro

Para continuar con la persistencia de la línea de investigación abordada y retomando los resultados obtenidos del minado de los dos proyectos, se plantean las siguientes acciones:

1. **Mejora de la extracción de datos:** Investigar y desarrollar técnicas y herramientas más eficientes para la extracción de datos de diferentes fuentes y sistemas. Esto incluye abordar desafíos como registros incompletos y limitaciones en las API de los sistemas.
2. **Desarrollo de técnicas de limpieza y preparación de datos:** Continuar investigando en la automatización de procesos de limpieza y preparación de datos para garantizar la calidad de la información utilizada en la Minería de Procesos. Esto implica abordar problemas como valores atípicos, datos faltantes o inconsistentes.
3. **Avance en la interpretación y visualización de resultados:** Investigar nuevas formas de interpretar y visualizar los resultados de la Minería de Procesos para facilitar su

comprensión y toma de decisiones. Esto puede incluir técnicas de visualización interactiva, gráficos intuitivos y herramientas de análisis avanzadas.

4. **Integración de la Minería de Procesos en la gestión de procesos de software:** Trabajar en la integración de la Minería de Procesos con enfoques más amplios de gestión de procesos de software. Esto implica vincular la Minería de Procesos con marcos de trabajo y promover la sinergia entre ambas disciplinas.
5. **Investigación en técnicas de modelado de procesos:** Explorar y desarrollar técnicas de modelado de procesos que permitan capturar aspectos más complejos de los procesos, como la toma de decisiones o la integración de sistemas.
6. **Mejora de la visualización de perspectivas:** Investigar y desarrollar métodos de visualización más efectivos para representar las diferentes perspectivas extraídas de los datos de los procesos. Esto permitirá una comprensión más clara y detallada de la interacción entre las actividades, los roles y otros aspectos relevantes de los procesos.
7. **Promoción de la adopción de la Minería de Procesos en las MiPyMEs:** Trabajar en la concientización y capacitación de las MiPyMEs sobre los beneficios y aplicaciones de la Minería de Procesos. Esto puede incluir la realización de talleres, seminarios y la divulgación de casos de éxito en el uso de la Minería de Procesos en contextos similares.

Estas acciones se enfocan en superar los retos identificados, mejorar las técnicas y herramientas utilizadas, e impulsar la adopción de esta disciplina en las MiPyMEs. Esto permitirá aprovechar al máximo el potencial de la Minería de Procesos para mejorar la eficiencia, la calidad y la toma de decisiones en los procesos de software.

Referencias

- Aguirre Mayorga, H. S., & Rincón García, N. (2015). Minería de procesos: desarrollo, aplicaciones y factores críticos. *Cuadernos de Administración*, 28(50), (pp. 137-157).
- Ailenei, I., Rozinat, A., Eckert, A., & van der Aalst, W. M. (2012). Definition and validation of process mining use cases. In *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I 9* (pp. 75-86). Springer Berlin Heidelberg.
- Atlassian (s.f.). Jira Software. Recuperado de <https://www.atlassian.com/software/jira>
- Bala, S., Mendling, J., Schimak, M., & Queteschiner, P. (2018). Case and activity identification for mining process models from middleware. In *IFIP Working Conference on The Practice of Enterprise Modeling* (pp. 86-102). Cham: Springer International Publishing.
- Bernardi, S., Domínguez, J. L., Gómez, A., Joubert, C., Merseguer, J., Perez-Palacin, D., Romeu, A. (2018). A systematic approach for performance assessment using process mining: An industrial experience report. *Empirical Software Engineering*, 23, 3394-3441.
- Bozkaya, M., Gabriels, J., & van der Werf, J. M. (2009). Process diagnostics: a method based on process mining. In *2009 International Conference on Information, Process, and Knowledge Management* (pp. 22-27). IEEE.
- Brechner, E. (2015). *Agile project management with Kanban*. Pearson Education.
- Burattin, A. (2015). *Process mining techniques in business environments. Lecture Notes in Business Information Processing*, 207.
- Caldeira, J., & e Abreu, F. B. (2016). Software development process mining: Discovery, conformance checking and enhancement. In *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)* (pp. 254-259). IEEE.
- Caldeira, J., e Abreu, F. B., Reis, J., & Cardoso, J. (2019). Assessing software development teams' efficiency using process mining. In *2019 International Conference on Process Mining (ICPM)* (pp. 65-72). IEEE.
- Cook, J. E., & Wolf, A. L. (1998). Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7(3), 215-249.
- De Leoni, M., van der Aalst, W. M., & Dees, M. (2016). A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, 56, 235-257.
- de Medeiros, A. K. A., Weijters, A. J., & van der Aalst, W. M. (2007). Genetic process mining: an experimental evaluation. *Data mining and knowledge discovery*, 14, 245-304.
- de Pablos Heredero, C., Agius, J. J. L. H., Romero, S. M. R., & Salgado, S. M. (2019). *Organización y transformación de los sistemas de información en la empresa*. esic.

- Diamantini, C., Genga, L., & Potena, D. (2016). Behavioral process mining for unstructured processes. *Journal of Intelligent Information Systems*, 47(1), 5-32.
- Dong, L., Liu, B., Li, Z., Wu, O., Babar, M. A., & Xue, B. (2017). A mapping study on mining software process. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 51-60). IEEE.
- dos Santos Garcia, C., Meinheim, A., Junior, E. R. F., Dallagassa, M. R., Sato, D. M. V., Carvalho, D. R., Scalabrin, E. E. (2019). Process mining techniques and applications—A systematic mapping study. *Expert Systems with Applications*, 133, 260-295.
- Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A., Dumas, M., La Rosa, M., & Reijers, H. A. (2018). Process-aware information systems. *Fundamentals of business process management*, (pp. 341-369).
- Eggert, M., & Dyong, J. (2022). Applying process mining in small and medium sized IT enterprises—challenges and guidelines. In *International Conference on Business Process Management* (pp. 125-142). Cham: Springer International Publishing.
- Erdem, S., & Demirörs, O. (2017). An exploratory study on usage of process mining in agile software development. In *Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings* (pp. 187-196). Springer International Publishing.
- Erdem, S., Demirörs, O., & Rabhi, F. (2018). Systematic mapping study on process mining in agile software development. In *Software Process Improvement and Capability Determination: 18th International Conference, SPICE 2018, Thessaloniki, Greece, October 9–10, 2018, Proceedings 18* (pp. 289-299). Springer International Publishing.
- Flores-Rios, B. L. (2016). Modelo de evolución de la gestión del conocimiento en mipymes de acuerdo al nivel de madurez en un programa de mejora de procesos de software. Universidad Autónoma de Baja California.
- Ghasemi, M., & Amyot, D. (2020). From event logs to goals: a systematic literature review of goal-oriented process mining. *Requirements Engineering*, 25(1), 67-93.
- GitHub (s.f.). GitHub. Recuperado de <https://github.com/>
- Gupta, M., Sureka, A., & Padmanabhuni, S. (2014). Process mining multiple repositories for software defect resolution from control and organizational perspective. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (pp. 122-131).
- Gupta, M. (2014). Nirikshan: process mining software repositories to identify inefficiencies, imperfections, and enhance existing process capabilities. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 658-661).
- Hompes, B., Buijs, J. C. A. M., Van der Aalst, W. M. P., Dixit, P., & Buurman, J. (2016). Discovering deviating cases and process variants using trace clustering. In *Proceedings of the 27th Benelux Conference on Artificial Intelligence (BNAIC)*, November (Vol. 5, No. 6).
- Ibarra-Esquer, J. E., González-Navarro, F. F., Flores-Rios, B. L., Burtseva, L., & Astorga-Vargas, M. A. (2017). Tracking the evolution of the internet of things concept across different application domains. *Sensors*, 17(6), 1379.

- Jokonowo, B., Sarno, R., Rochimah, S., & Priambodo, B. (2019). Process mining: Measuring key performance indicator container dwell time. *Indones. J. Electr. Eng. Comput. Sci*, 16(1), 401-411.
- Jovanović, M., Mesquida, A. L., Mas, A., & Colomo-Palacios, R. (2020). Agile transition and adoption frameworks, issues and factors: a systematic mapping. *IEEE Access*, 8, 15711-15735.
- Kalenkova, A. A., van der Aalst, W. M., Lomazova, I. A., & Rubin, V. A. (2017). Process mining using BPMN: relating event logs and process models. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems* (pp. 123-123).
- Keith, B., & Vega, V. (2016). Process mining applications in software engineering. In *International Conference on Software Process Improvement* (pp. 47-56). Springer, Cham.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004), (pp. 1-26).
- Kitchenham, B. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University and University of Durham.
- Kneuper, R. (2018). *Software Processes and Life Cycle Models An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes*. by the registered company Springer Nature Switzerland AG.
- LeBlanc, R., Sobel, A., Diaz-Herrera, J. L., & Hilburn, T. B. (2014). *Software Engineering 2014-Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering a Volume of the Computing Curricula Series*. Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery.
- Leemans, M. (2018). Hierarchical process mining for scalable software analysis. Ph.D. dissertation, Eindhoven University of Technology.
- Lemos, A. M., Sabino, C. C., Lima, R. M., & Oliveira, C. A. (2011). Using process mining in software development process management: A case study. In *2011 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1181-1186). IEEE.
- Li, M., Liu, L., Yin, L., & Zhu, Y. (2011). A process mining based approach to knowledge maintenance. *Information Systems Frontiers*, 13, 371-380.
- Li, G. (2019). Process mining based on object-centric behavioral constraint (OCBC) models. Ph.D. Thesis, Eindhoven University of Technology, Mathematics and Computer Science, Eindhoven.
- Liu, C., van Dongen, B., Assy, N., & van der Aalst, W. M. (2016). Component behavior discovery from software execution data. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). IEEE.
- Mannhardt, F. (2018). Multi-perspective Process Mining. In *BPM (Dissertation/Demos/Industry)* (pp. 41-45).
- Marin-Castro, H. M., & Tello-Leal, E. (2021). Event log preprocessing for process mining: a review. *Applied Sciences*, 11(22), 10556.

- Marques, R., da Silva, M. M., & Ferreira, D. R. (2018). Assessing agile software development processes with process mining: A case study. In 2018 IEEE 20th Conference on Business Informatics (CBI) (Vol. 1, pp. 109-118). IEEE.
- Özdağoğlu, G., & Kavuncubaşı, E. (2019). Monitoring the software bug-fixing process through the process mining approach. *Journal of Software: Evolution and Process*, 31(7), e2162.
- Pérez-Alfonso, D., Yzquierdo-Herrera, R., & Lazo-Cortés, M. (2013). Recommendation of process discovery algorithms: a classification problem. *Res. Comput. Sci*, 61, (pp. 33-42).
- Poncin, W., Serebrenik, A., & Van Den Brand, M. (2011). Process mining software repositories. In 2011 15th European conference on software maintenance and reengineering (pp. 5-14). IEEE.
- Reséndiz, J. D. H., Leal, E. T., Castro, H. M. M., & Galván, G. R. (2019). Aplicación de técnicas de aprendizaje no supervisado para la agrupación de trazas en el dominio de Minería de Procesos (Application of unsupervised learning techniques for clustering traces in the Process Mining domain). *Pistas Educativas*, 41(133).
- Rivas, M. H., & Bayona-Oré, S. (2019). Algoritmos de minería de proceso para el descubrimiento automático de procesos. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (31), (pp. 33-49).
- Rozinat, A., & Van der Aalst, W. M. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1), 64-95.
- Rubin, V., Lomazova, I., & Aalst, W. M. V. D. (2014). Agile development with software process mining. In *Proceedings of the 2014 international conference on software and system process* (pp. 70-74).
- Samalikova, J., Kusters, R., Trienekens, J., Weijters, T., & Siemons, P. (2011). Toward objective software process information: experiences from a case study. *Software Quality Journal*, 19, 101-120.
- Samalikova, J., Kusters, R. J., Trienekens, J. J., & Weijters, A. J. M. M. (2014). Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice. *Journal of Software: Evolution and Process*, 26(7), 714-728.
- Scrum-institute (2017). *Scrum revealed training book* (3rd ed.). International Scrum Institute . <https://bit.ly/3rKMF31>
- Sebu, M. L., & Ciocarlie, H. (2014). Applied process mining in software development. In 2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI) (pp. 55-60). IEEE.
- Seeliger, A., Nolle, T., & Mühlhäuser, M. (2018). Process explorer: an interactive visual recommendation system for process mining. In *KDD Workshop on Interactive Data Exploration and Analytics*.
- Senderovich, A. (2017). Queue Mining: Service Perspectives in Process Mining. In *BPM (Demos)* (pp. 143-151).
- Syamsiyah, A., Bolt, A., Cheng, L., Hompes, B. F., Jagadeesh Chandra Bose, R. P., van Dongen, B. F., & van der Aalst, W. M. (2017). Business process comparison: A methodology and case study. In *Business Information Systems: 20th International Conference, BIS 2017, Poznan, Poland, June 28–30, 2017, Proceedings 20* (pp. 253-267). Springer International Publishing.

- Syamsiyah, A. (2020). In-database preprocessing for process mining. Eindhoven University of Technology
- Štolfa, J. (2018). Application of Process Mining in Intelligent Process Support. VSB – Technical University of Ostrava
- TFPM – IEEE Task Force on Process Mining. (2012). Process mining manifesto. In Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, (pp. 169-194). Springer Berlin Heidelberg.
- Urrea-Contreras, S. J., Flores-Rios, B. L., Astorga-Vargas, M. A., Ibarra-Esquer, J. E., & González-Navarro, F. F. (2018). MoProPEI-MP an Adaptation in the Selection of Process Mining Tools for Software Processes. Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento 2018, 151-162.
- Urrea-Contreras, S. J., Flores-Rios, B. L., Astorga-Vargas, M. A., Ibarra-Esquer, J. E., González-Navarro, F. F., Velazquez-Solis, P. E., ... & Albarracín, C. A. A. (2020). Propuesta de un marco de trabajo para la Identificación y Selección de algoritmos y herramientas de Minería de Procesos orientado a las organizaciones de desarrollo de software. Revista Ibérica de Sistemas e Tecnologías de Informação, (E25), (pp. 348-363).
- Urrea-Contreras, S. J., Flores-Rios, B. L., Astorga-Vargas, M. A., & Ibarra-Esquer, J. E. (2021). Process Mining Perspectives in Software Engineering: A Systematic Literature Review. In 2021 Mexican International Conference on Computer Science (ENC) (pp. 1-8). IEEE.
- Van der Aalst, W. M., Reijers, H. A., & Song, M. (2005). Discovering social networks from event logs. Computer Supported Cooperative Work (CSCW), 14, 549-593.
- Van der Aalst, W. M. (2015). Extracting event data from databases to unleash process mining. BPM-Driving innovation in a digital world, 105-128.
- Van der Aalst, W. (2016). Process mining: data science in action (Vol. 2). Heidelberg: Springer.
- Van Dongen, B. F., de Medeiros, A. K. A., Verbeek, H. M. W., Weijters, A. J. M. M., & van Der Aalst, W. M. (2005). The ProM framework: A new era in process mining tool support. In Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. Proceedings 26 (pp. 444-454). Springer Berlin Heidelberg.
- Van Eck, M. L., Lu, X., Leemans, S. J., & Van Der Aalst, W. M. (2015). PM²: a process mining project methodology. In Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015, Stockholm, Sweden, June 8-12, 2015, Proceedings (pp. 297-313). Cham: Springer International Publishing.
- Van Eck, M. L. (2022). Process Mining for Smart Product Design. Eindhoven University of Technology.
- Van Zelst, S. J. (2019). Process mining with streaming data. Eindhoven University of Technology.
- Vavpotič, D., Bala, S., Mendling, J., & Hovelja, T. (2022). Software process evaluation from user perceptions and log data. Journal of Software: Evolution and Process, 34(4), e2438.
- Vidoni, M. (2022). A systematic process for Mining Software Repositories: Results from a systematic literature review. Information and Software Technology, 144, 106791.

- Wang, J., Wong, R. K., Ding, J., Guo, Q., & Wen, L. (2012). Efficient selection of process mining algorithms. *IEEE Transactions on Services Computing*, 6(4), 484-496.
- Wijnhoven, F., Hoffmann, P., Bemthuis, R., & Bokseveld, J. (2023). Using process mining for workarounds analysis in context: Learning from a small and medium-sized company case. *International Journal of Information Management Data Insights*, 3(1), 100163.
- Yang, J., Ouyang, C., van der Aalst, W. M., ter Hofstede, A. H., & Yu, Y. (2022). OrdinoR: A framework for discovering, evaluating, and analyzing organizational models using event logs. *Decision Support Systems*, 113771.
- Zhao, W., & Zhao, X. (2014). Process mining from the organizational perspective. In *Foundations of Intelligent Systems: Proceedings of the Eighth International Conference on Intelligent Systems and Knowledge Engineering, Shenzhen, China, Nov 2013 (ISKE 2013)* (pp. 701-708). Springer Berlin Heidelberg.

Anexo 1. Productos académicos

Durante el desarrollo de la investigación se obtuvieron los siguientes productos académicos:

Publicaciones arbitradas en revistas y congresos

- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Félix Fernando González-Navarro, et al. **Applying process mining: The reality of a software development SME**. Applied Sciences.
- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Félix Fernando González-Navarro, et al. **Process Mining Model Integrated with Control Flow, Case, Organizational and Time Perspectives in a Software Development Project**. International Conference in Software Engineering Research and Innovation (CONISOFT), pp. 92-101. 2022. IEEE Índice: Scopus. Citado por 2
- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Félix Fernando González-Navarro. **Software Process Mining Projects based on the analysis of knowledge flows**. Mexican International Conference on Computer Science (ENC), pp. 57-73, 2022. IEEE Índice: Scopus
- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer. **Revisión de patentes de la Minería de Procesos en la Ingeniería de Software**, Revista Colombiana de Computación. pp. 6–19. 2022. México. ISSN: 2539-2115. Índice: Scopus Q4
- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Félix Fernando González-Navarro. **Proyectos de Minería de Procesos de Software basado en el análisis de flujos de conocimiento**. Abstraction & Application (A&A) Vol. 37. pp. 57–73. 2022. México. ISSN: 2007-2635. Índice: Latindex.

- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer. **Process Mining Perspectives in Software Engineering: A Systematic Literature Review**. Mexican International Conference on Computer Science (ENC), 2021. IEEE Índice: Scopus. Citado por 8.
- Silvia Jaqueline Urrea-Contreras, Brenda Leticia Flores-Rios, María Angélica Astorga-Vargas, Jorge Eduardo Ibarra Esquer, Félix Fernando González-Navarro, et al. **Propuesta de un marco de trabajo para la Identificación y Selección de algoritmos y herramientas de Minería de Procesos orientado a las organizaciones de desarrollo de software**. Revista Ibérica de Sistemas y Tecnologías de la Información (RISTI), E25, pp. 348-363, 2020.

Registro de Modelo de Utilidad

- Urrea Contreras, S. J., Flores Rios, B. L., Astorga Vargas, M. A., **Catálogo de conceptos básicos de la Minería de Procesos**. Número: 03-2021-042112204200-01. Instituto Nacional de Derecho de Autor (INDAUTOR). México. 2021.

Ponente por invitación a eventos académicos

- Ponente en el 1er coloquio internacional de investigación en Ingeniería de software. Universidad Veracruzana. Mayo 2023.
- Ponente en el seminario organizado por la Asociación Nacional de Universidades e Instituciones de Educación Superior a través de su Comité ANUIES-TIC y de la Comisión de Cultura de la Administración y Gestión Financiera mediante las TIC el 9 de junio de 2022.

Anexo 2. Desarrollo para las Actividades ETL

```
1 import requests, json, sys, csv, pymongo
2 from requests.auth import HTTPBasicAuth
3 from DTO.Evento import Evento
4 from DTO.JiraIssue import Jira_Issue
5 from DTO.Commit import Commit
6 import os
7 from dotenv import load_dotenv
8
9 load_dotenv()
10
11 headers = {
12     "Accept": "application/json"
13 }
14
15 auth = HTTPBasicAuth(os.getenv('JIRA_USER'), os.getenv('JIRA_KEY'))
16
17
18 client = pymongo.MongoClient(os.getenv('MONGO_URI'))
19 db = client["Proyecto2"]
20 eventos_collection = db["JiraLog"]
21 issues_collection = db.issues
22 commit_log_collection = db["GithubLog"]
23
24
25 maxResultGetList = 100
26
27 def get_issue_list(project_name):
28     list_of_issues = []
29     print('Obteniendo issues de proyecto: %s' % (project_name))
30     offset = 0
31     while True:
32         query = {
33             'jql': 'project = %s' % (project_name),
34             'startAt' : offset,
35             'maxResults' : maxResultGetList
36         }
37         response = requests.request(
38             'GET',
39             'https://%s/rest/api/3/search' % (os.getenv('ATLASIAN_BASE_URL')),
40             headers=headers,
41             params=query,
42             auth=auth
43         )
44         offset += maxResultGetList
45         jsonResult = json.loads(response.text)
46         for issue in jsonResult["issues"]:
47             list_of_issues.append(
48                 {
49                     'key' : issue["key"],
50                     'id' : issue["id"],
51                     'created' : issue["fields"]["created"],
```

Figura 31. Codificación para las actividades ETL (parte 1).

```

52         'creator' : issue["fields"]["creator"]["displayName"]
53     }
54 )
55 if offset > jsonResult["total"]:
56     print('Se obtuvieron %d de %d issues' % (jsonResult["total"], jsonResult["total"]))
57     break
58     print('Se obtuvieron %d de %d issues' % (offset, jsonResult["total"]))
59 return list_of_issues
60
61 def get_issue_changelog(issue):
62     offset = 0
63     changelog = []
64     while True:
65         query = {
66             'startAt' : offset,
67             'maxResults' : maxResultGetList
68         }
69         response = requests.request(
70             'GET',
71             'https://%s/rest/api/3/issue/%s/changelog' % (os.getenv('ATLASIAN_BASE_URL'), issue["key"]),
72             headers=headers,
73             params=query,
74             auth=auth
75         )
76         jsonResult = json.loads(response.text)
77         offset += maxResultGetList
78         changelog += jsonResult["values"]
79         if jsonResult["isLast"]:
80             break
81     changelog = [cl for cl in changelog if any(item["field"] == 'status' for item in cl["items"])]
82     changelog_csv_array = []
83     changelog_csv_array.append([issue["key"], "", "Create Issue", issue["created"], issue["creator"]])
84     changelog_csv_array.append([issue["key"], "Create Issue", "To Do", issue["created"], issue["creator"]])
85     for change in changelog:
86         for item in change["items"]:
87             if item["field"] == 'status':
88                 changelog_csv_array.append([issue["key"], item["fromString"], item["toString"], change["created"], change["author"]["displayName"]])
89     return changelog_csv_array
90
91
92 def get_commits_for_issue(issue):
93     specialHeader = {
94         "X-ExperimentalApi" : "JiraIssue, JiraIssueDevInfoDetails"
95     }
96     graphqlQuery = """query MyQuery {
97         jira {
98             issue(id: "ari:cloud:jira:%s:issue/%s") {
99                 id
100                 devInfoDetails {
101                     commits {

```

Figura 32. Codificación para las actividades ETL (parte 2).

```

102         details {
103             created
104             entityUrl
105             providerCommitId
106             author {
107                 name
108             }
109         }
110     }
111 }
112 }
113 }
114 }""" % (os.getenv('ATLASIAN_PROJECT_UUID'), issue["id"])
115 response = requests.post(
116     url='https://jdog.atlassian.net/gateway/api/graphql',
117     json={'query': graphqlQuery},
118     headers=specialHeader,
119     auth=auth
120 )
121 jsonResult = json.loads(response.text)
122 commits_csv_array = []
123 for commit in jsonResult["data"]["jira"]["issue"]["devInfoDetails"]["commits"]["details"]:
124     commits_csv_array.append([issue["key"], commit["providerCommitId"], commit["created"], commit["author"]["name"], commit["entityUrl"]])
125 return commits_csv_array
126
127
128 def main():
129     list_of_issues_in_project = get_issue_list(sys.argv[1])
130     count = 1
131     file_name = 'event_log_%.csv' % (sys.argv[1])
132     commit_log_file = 'commit_log_%.csv' % (sys.argv[1])
133     for issue in list_of_issues_in_project:
134         objeto_issue = Jira_Issue(issue["key"], sys.argv[1])
135         issues_collection.insert_one(objeto_issue.__dict__)
136         print('Procesando issue %s numero %d de %d...' % (issue["key"], count, len(list_of_issues_in_project)))
137         events = get_issue_changelog(issue)
138         with open(file_name, 'a') as csv_file:
139             writer = csv.writer(csv_file)
140             writer.writerow(events)
141         for evento in events:
142             objeto_evento = Evento(evento[0], evento[1], evento[2], evento[3], evento[4])
143             eventos_collection.insert_one(objeto_evento.__dict__)
144         commits = get_commits_for_issue(issue)
145         for commit in commits:
146             objeto_commit = Commit(commit[0], commit[1], commit[2], commit[3], commit[4])
147             commit_log_collection.insert_one(objeto_commit.__dict__)
148         with open(commit_log_file, 'a') as commit_csv:
149             writer = csv.writer(commit_csv)
150             writer.writerow(commits)
151         count += 1
152     print('Se han procesado los event logs de los issues del proyecto %s en el archivo %s' % (sys.argv[1], file_name))

```

Figura 33. Codificación para las actividades ETL (parte 3).

Anexo 3. Ejemplo de conjunto de trazas totales del estudio de caso

Anexo 1. Conjunto de trazas totales con sus *issues* correspondientes presentes en el registro de eventos.
a= Create Issue, b= Backlog, c= Selected for Development, d= To Do, e= In Progress, f= Verification, g= Quality Assurance, h= Rejected, i= Validation y j= Done

Traza	ID Issue
$\sigma_1 = \langle a, e, f, g, i, j \rangle$	BPF-23, BPF-24, BPF-25, BPF-26, BPF-27, BPF-28, BPF-30, BPF-31, BPF-32, BPF-33, BPF-34, BPF-35, BPF-36, BPF-37, BPF-39, BPF-40, BPF-53, BPF-58, BPF-64, BPF-66
$\sigma_2 = \langle a, b, c, d, e, f, g, i, j \rangle$	BPF-8, BPF-11, BPF-12, BPF-13, BPF-14, BPF-41, BPF-44, BPF-45, BPF-49, BPF-50, BPF-51, BPF-68, BPF-69, BPF-70, BPF-74
$\sigma_3 = \langle a, f, g, i, j \rangle$	BPF-21, BPF-22, BPF-55, BPF-56, BPF-57
$\sigma_4 = \langle a, b, c, d, c, d, c, d, e, f, g, i, j \rangle$	BPF-9, BPF-10
$\sigma_5 = \langle a, i, j \rangle$	BPF-61, BPF-62
$\sigma_6 = \langle a, e, f, g, j, i, j \rangle$	BPF-76, BPF-78
$\sigma_7 = \langle a, b, c, d, e, f, g, f, g, i, g, i, j \rangle$	BPF-15
$\sigma_8 = \langle a, b, c, d, c, d, e, c, d, f, g, i, j \rangle$	BPF-16
$\sigma_9 = \langle a, b, c, d, c, d, f, g, i, j \rangle$	BPF-17
$\sigma_{10} = \langle a, b, c, d, c, d, e, f, g, i, g, i, j \rangle$	BPF-18
$\sigma_{11} = \langle a, b, c, d, c, d, e, f, g, i, j \rangle$	BPF-19
$\sigma_{12} = \langle a, c, b, c, d, e, f, g, i, j \rangle$	BPF-20
$\sigma_{13} = \langle a, b, c, d, c, d, c, g, i, j \rangle$	BPF-46
$\sigma_{14} = \langle a, b, c, d, c, f, d, e, g, i, j \rangle$	BPF-47
$\sigma_{15} = \langle a, b, c, d, e, f, g, f, g, i, j \rangle$	BPF-48
$\sigma_{16} = \langle a, e, f, i, j \rangle$	BPF-52
$\sigma_{17} = \langle a, c, e, f, g, i, j \rangle$	BPF-60
$\sigma_{18} = \langle a, c, d, e, f, g, i, j \rangle$	BPF-63
$\sigma_{19} = \langle a, b, c, d, e, f, e, f, g, i, j \rangle$	BPF-67
$\sigma_{20} = \langle a, c, d, e, f, g, j \rangle$	BPF-72
$\sigma_{21} = \langle a, e, f, g, i, j, i, j \rangle$	BPF-75
$\sigma_{22} = \langle a, e, f, g, h, i, j \rangle$	BPF-77