

Universidad de las Ciencias Informáticas Facultad 3



Componente para medir la calidad de las variantes de modelos de procesos durante la fase de diagnóstico de procesos.

Trabajo Final presentado en opción al título de Ingeniera
en Ciencias Informáticas

Autora: Elena Laurencio González

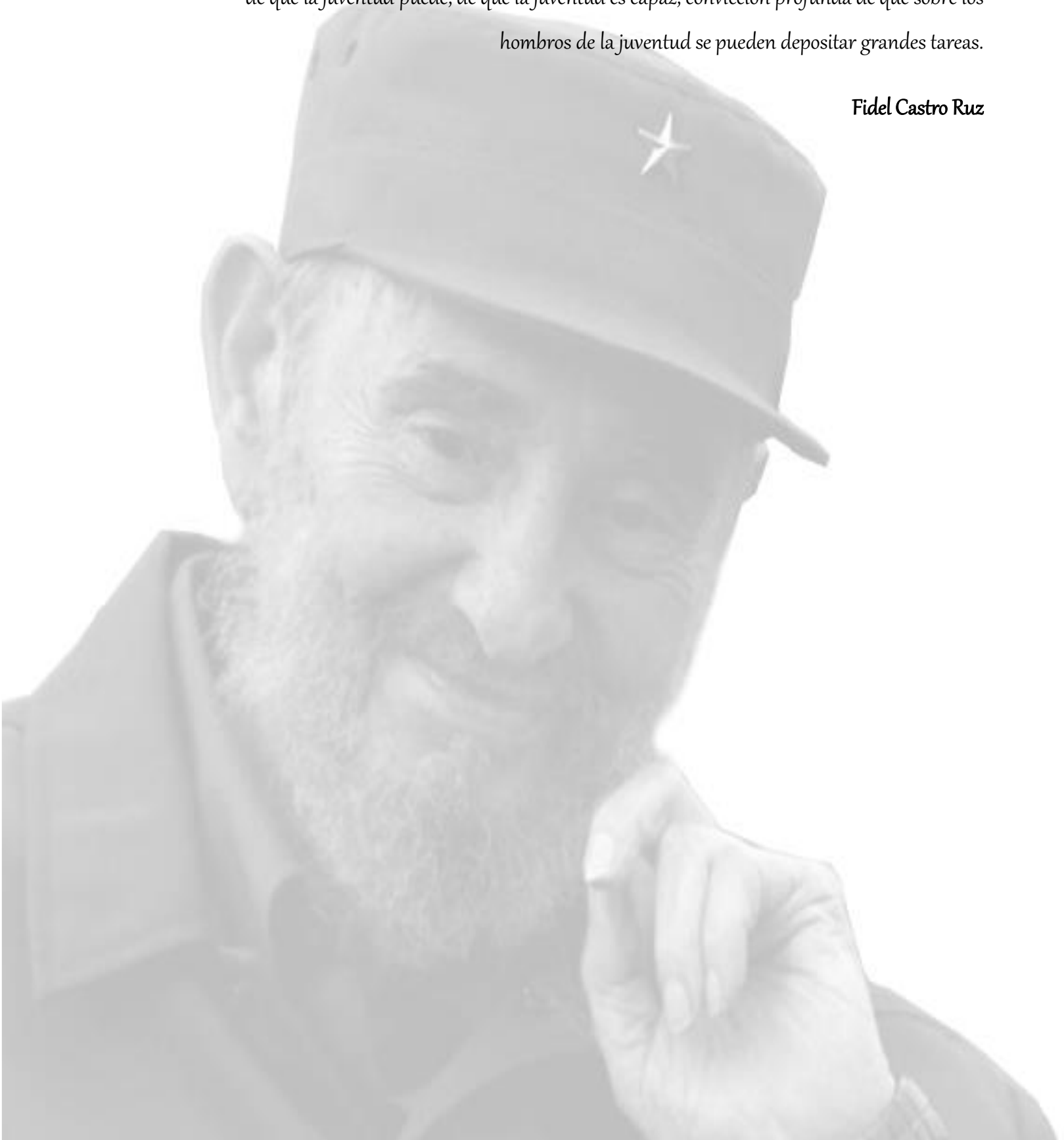
Tutores: Ing. Pedro Arango Astorga

Dr. Damián Pérez Alfonso

La Habana, Junio 2016

Creer en los jóvenes es ver en ellos además de entusiasmo, capacidad; además de energía, responsabilidad; además de juventud, ¡pureza, heroísmo, carácter, voluntad, amor a la patria, fe en la patria! ¡amor a la Revolución, fe en la Revolución, confianza en sí mismos!, convicción profunda de que la juventud puede, de que la juventud es capaz, convicción profunda de que sobre los hombros de la juventud se pueden depositar grandes tareas.

Fidel Castro Ruz



Agradecimientos

Mamita mía Elena González Rojas te agradezco por darme la vida y educarme como lo has hecho con mucho amor ya la vez con rectitud, gracias por consolarme, guiarme, por tus regaños y enseñanzas, gracias por todas las fuerzas que me has dado en los momentos tan difíciles que he pasado, especialmente estos cinco años de mi carrera para convertirme finalmente en ingeniera, yo sé que los hijos no pedimos venir al mundo, pero si me preguntaran yo te elegiría nuevamente como mi mentora. Te doy gracias papá mío Adalberto Fonte Blanco, por todo el amor que me has dado a pesar de yo no ser de tu sangre, me acogiste como una más de tus hijas y luchaste junto a mi mamá para que yo finalmente me hiciera una profesional de bien, siempre me hiciste sentir parte de esa gran familia que son mis hermanas Adelky Fonte y Liannet Fonte. No puede faltar mi abuela Corlinda Rojas Durand, que ha sido mi tutora mayor en el camino de la vida que aún es muy largo, te doy abuela mil gracias por hacerme toda una mujer, por entenderme y siempre comprenderme cuando yo más lo necesitaba. Tengo también una tía carnal, mi gran tía Yayi, que siempre me cuidó en las semanas de receso y me cuidaba como suya, te quiero mucho. Mis dos tíos del alma, que siempre han estado ahí a mí lado cuidándome ellos son Tía Gasmery (Marta) y Tío Santo. También agradezco a mi tío Pablo y a mi nueva tía María Esther que a pesar de llevar muy poco dentro de esta pequeña familia ya es una parte importante de ella y a Leonardo mi tío loco, que siempre estuvo pendiente de mí. No puede faltar mi más grande Tía Margot que siempre me ayudó sin condiciones, te doy las gracias por ayudarme y siempre esperarme a la hora que fuera aquellos días que yo llegaba de Guantánamo con el maletín que pesaba más que nosotras y siempre, siempre estar ahí al pendiente de mí guiándome y aconsejándome en este viaje que todos hacemos por la vida, te quiero mucho.

Agradezco a todos mis amigos no puede faltar mi gran hermana de escuela Grelenis Lirene que juntas pasamos muchas cosas inolvidables, a mi gran amigo Leunam que me ayudó mucho en mis primeros años de mi carrera, a mis fieles seguidores Osvaldo y Javier en los juegos de dominó, y no puede faltar a mi gran amigo, mi hermano y mi consejero Yordan, mi negro de la suerte sabes que te quiero mucho. También no quería dejar de mencionar a mis dos súper amigos de la Isla de la Juventud Jonel que siempre nos acompañó a Yamilka y a mí en todo, los quiero y siempre seguiremos siendo los 3 mosqueteros.

No puede faltar y no por ser los últimos son menos importantes, una nueva parte de mi familia, mis suegros Reina y Soto como le decimos todos, que me acogieron y me dieron mucho cariño, claro no puede faltar mi gran amor de toda la vida Yacer Soto, que siempre me ha guiado y me ha dado todo el amor del mundo y me hace la persona más feliz de este mundo.

Les agradezco a todos los que de una forma u otra, han aportado un granito de arena para contribuir a realizarme como Ingeniera en Ciencias Informáticas, a todos, muchas gracias y sepan que les estaré eternamente agradecida por la luz que me han dado en este paso tan importante para mi porvenir. ¡Muchas Gracias!

DECLARACIÓN JURADA DE AUTORÍA

Declaro ser autora del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Se autoriza a dicho centro para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Elena Laurencio González

Firma de la autora.

Ing. Pedro Arango Astorga

Dr. Damián Pérez Alfonso

Firma de los tutores

Resumen

La Minería de Procesos es una disciplina que impulsa el desarrollo de técnicas y herramientas para el diagnóstico de procesos, entre las que se encuentra *Variant Miner*. Esta técnica ha sido implementada como complemento del marco de trabajo ProM, la cual a partir de un registro de eventos genera un Árbol de Variantes y proporciona el Perfil de Diagnóstico correspondiente, en el cual se registran las evaluaciones realizadas a estos modelos, teniendo en cuenta los comportamientos asumidos y descartados en la identificación de los patrones de control de flujo. Además, brinda información referente a la frecuencia de cada traza dentro del registro de eventos.

La presente investigación tiene como objetivo principal desarrollar un componente para medir la calidad de las variantes de modelos de procesos, que facilite la selección de las mismas durante la fase de diagnóstico. Los principales aportes de esta investigación son: proporcionar al usuario una evaluación exacta de cada variante; generar una versión del árbol correspondiente, definida como entrada para crear una red de Petri y modificar el registro de eventos con una nueva codificación de las trazas; a partir de estas últimas se realiza el cálculo de las métricas y la integración de la biblioteca CoBeFra con el complemento *Variant Miner*. Todo ello permitió el desarrollo de una nueva versión del complemento *Variant Miner* compuesto por tres elementos: *VariantTree*, *ConvertVTreeToPetriNet* y la biblioteca CoBeFra, reforzando el trabajo y la cooperación entre sí.

Palabras claves: Árbol de Variantes, biblioteca CoBeFra, Minería de Procesos, Perfil de Diagnóstico, Red de Petri, *Variant Miner*.

ABSTRACT

Process Mining is a discipline that promotes the development of techniques and tools for process diagnostics, among which is Variant Miner. This technique has been implemented to complement the framework ProM, which from an event log generates a tree variants and provides the profile corresponding diagnosis, in which assessments made on these models are recorded, taking into account behaviors made and discarded in identifying patterns flow control. It also provides information regarding the frequency of each trace in the event log.

This research has as main objective to develop a component to measure the quality of the variants of process models, which facilitates the selection of the same during the diagnostic phase. The main contributions of this research are to provide the user with an accurate assessment of each variant; generate a version of the tree, defined as input to create a Petri net and modify the event log with new coding traces; from the latter calculating metrics and integration of CoBeFra library with add-Miner Variant it is performed. This enabled the development of a new version of the add Variant Miner composed of three elements: Variant Tree, ConvertVTreeToPetriNet and CoBeFra library, strengthening the work and cooperation with each other.

Keywords: Diagnostic Profile, library CoBeFra, Mining Process, Petri net, Tree Variants, Variant Miner.

INDICE

Introducción.....	1
CAPITULO 1: Minería de Procesos, técnica para el descubrimiento de procesos y el marco integral CoBeFra 5	
1.1 Minería de Procesos	5
1.1.1 Análisis de las métricas de la dimensión Aptitud	8
1.1.2 Análisis de las métricas de la dimensión Precisión.....	9
1.1.3 Análisis de la métrica de la dimensión Generalización	11
1.2 Descubrimiento de modelos de procesos.....	13
1.2.1 Minería de Variantes	13
1.3 Herramientas para evaluar modelos.....	15
1.3.1 Un análisis de conformidad del marco integral CoBeFra	15
1.4 Herramientas y Tecnologías.....	17
1.4.1 ProM	17
1.4.2 CoBeFra.....	17
1.4.3 Entorno de desarrollo integrado	18
1.4.4 Control de Versiones.....	18
1.4.5 Ingeniería de software asistida por computadora	18
1.4.6 Metodología de desarrollo	19
Conclusiones parciales del capítulo	19
CAPITULO 2: Propuesta de Solución	20
2.1 Introducción	20

2.2	Generar un Árbol de Variantes	20
2.3	Generar un modelo en red de Petri	24
2.4	Marco integral CoBeFra	26
	Conclusiones parciales del capítulo	28
CAPITULO 3: Validación de la solución propuesta		30
3.1	Pruebas Internas	30
3.1.1	Características de los registros de eventos artificiales	30
3.1.2	Resultados obtenidos aplicando G2	32
3.2	Coeficiente de correlación de Pearson	35
3.2.1	Análisis para la dimensión Aptitud	36
3.2.2	Análisis para la dimensión Precisión	36
3.2.3	Análisis para la dimensión Generalización	37
	Conclusiones parciales del capítulo	38
	Conclusiones generales	39
	Recomendaciones	40
	REFERENCIAS	41

Índice de Figuras

Figura 1 Representación de variantes de modelos.....	2
Figura 1.1: Tareas principales de la disciplina Minería de Procesos tomada de (Aalst 2012).	5
Figura 1.2: Componentes de la técnica Variants Miner tomado de (Damián Pérez Alfonso 2015a)....	14
Figura 2.1: Interfaz principal al ejecutar el Variant Miner.	20
Figura 2.2: Árbol de Variantes.	21
Figura 2.3: Estructura del componente Variant Miner representada mediante paquetes.	23
Figura 2.4: Reducción del árbol del subproceso analizado.	23
Figura 2.5: Representación del funcionamiento del plugin para convertir a red de Petri.	25
Figura 2.6: Modelo en red de Petri tomado de (Chaviano 2014).....	25
<i>Figura 2.7: Componentes del Plugin ConvertVTreeToPetriNet tomado de (Chaviano 2014).</i>	<i>26</i>
Figura 2.8: Arquitectura de la biblioteca CoBeFra tomado de (Brouke 2014).	27
Figura 2.9: Funcionamiento de la biblioteca CoBeFra.	28

Índice de Tablas

q

Tabla 1.1: Evaluación de las métricas de la dimensión Aptitud.....	8
Tabla 1.2: Evaluación de las métricas de la dimensión Precisión.	10
Tabla 1.3: Evaluación de las métricas de la dimensión Generalización.	11
Tabla 3.1: Características de los modelos de procesos artificiales tomado de (Damián Pérez Alfonso 2014).....	31
Tabla 3.2: Características de los registros de eventos.	31
Tabla 3.3: Resultado de la reducción de actividades.....	32
Tabla 3.4: Tiempo para crear un registro de eventos.	33
Tabla 3.5: Tiempo para crear una red de Petri.	33
Tabla 3.6: Tiempo consumido para el cálculo de las métricas.	34
Tabla 3.7: Tiempo promedio consumido para realizar cada operación.	34
Tabla 3.8: Comparación de los valores calculados mediante el componente y CoBeFra.	35
Tabla 3.9: Resultados obtenidos para la Aptitud.....	36
Tabla 3.10: Resultados obtenidos para la Precisión.	37
Tabla 3.11: Resultados obtenidos para la Generalización.	37

Introducción

Las empresas utilizan sistemas de información que son capaces de gestionar sus procesos de negocio, trayendo múltiples beneficios, entre los que se pueden encontrar: automatización de los procesos operativos, suministro de una plataforma de información necesaria para la toma de decisiones y usabilidad de la información (Kourdi 2008).

La Minería de Procesos es una disciplina dedicada al análisis de los procesos, partiendo de sus trazas de ejecución en los sistemas de información. Al conjunto de las trazas de un proceso se le denomina registro de eventos, ya que cada traza contiene una secuencia de eventos ordenados según su ocurrencia. Estos eventos reflejan la ejecución de las actividades del proceso al cual pertenece la traza. El análisis automático de la información contenida en los registros de eventos, permite a las técnicas de Minería de Procesos ofrecer resultados en menor tiempo y con mayor fiabilidad que las técnicas tradicionales de análisis y mejora de procesos. Debido a esto, dentro del área de Gestión de Procesos de Negocios, ha aumentado sostenidamente el interés en la Minería de Procesos en la última década. Importantes proveedores de software, universidades y empresas consultoras se han aliado junto al IEEE (Instituto de Ingeniería Eléctrica y Electrónica) para la investigación y el desarrollo de esta área (Damián Pérez Alfonso 2015b).

La Minería de Procesos tiene como objetivo el diagnóstico de procesos, su descubrimiento, el chequeo de conformidad entre la ejecución real y sus modelos de referencia y la mejora de los procesos. Ha demostrado su utilidad en diferentes etapas de la Gestión de Procesos de Negocios, incluyendo el diagnóstico, la ejecución, el ajuste y la reconfiguración. El diagnóstico de procesos ayuda a tener una visión general del proceso, de los aspectos más significativos del mismo y de las técnicas que pueden ser más útiles en su posterior análisis. Esta etapa de la Gestión de Procesos de Negocios abarca el análisis de rendimiento, la detección de anomalías, la identificación de patrones comunes y su ubicación en el contexto de ejecución del proceso (Damián Pérez Alfonso 2015b).

Se han desarrollado un conjunto de técnicas de Minería de Procesos útiles en la etapa de diagnóstico que al ser aplicadas generan un modelo a partir de un registro de eventos. Sin embargo, dichas técnicas suelen ser afectadas por el ruido, la ausencia de información, el número de tareas involucradas, las dimensiones y la complejidad del registro de eventos.

El ruido y la ausencia de información son características de los procesos cuya ejecución varía para distintas instancias del proceso. El 80 % de la variabilidad de un proceso suele encontrarse en menos del 20 % de las trazas del registro de eventos. Aunque las técnicas de Minería de Procesos suelen descartar este comportamiento poco frecuente, en él se encuentran anomalías y desviaciones

que son omitidas de los modelos resultantes. Dichos modelos son incompletos desde el punto de vista de los objetivos a lograr en el diagnóstico. En contraposición los modelos que reflejan todo el comportamiento suelen ser densos ya que no jerarquizan los comportamientos adecuadamente, dificultando la comprensión del proceso (Damián Pérez Alfonso 2015b).

Para resolver la afectación que provocan el ruido y la ausencia de información en la comprensión del proceso es necesario controlar el impacto estructural de estas características. Sin embargo, considerar ciertos comportamientos infrecuentes como ruido y/o asumir que faltan determinadas evidencias de la ejecución del proceso, son estimaciones que sólo pueden ser confirmadas a partir del contexto de ejecución particular del proceso analizado. Por tanto, la técnica *Variants Miner* propone la construcción de varios modelos de procesos integrados en una única representación y con información sobre los comportamientos descartados y asumidos en la construcción de cada modelo, más conocido como "Minería de variantes". De esta manera se puede decidir en función del contexto cuáles son las anomalías y desviaciones, así como identificar su impacto en la estructura de control de flujo del proceso (Damián Pérez Alfonso 2015b).

Al aplicar la técnica Minería de Variantes se obtiene un Árbol de Variantes (Figura 1), el cual estará compuesto por las diferentes variantes de descomposición, los patrones de control de flujo, los subprocessos y las actividades del proceso que se está analizando y un Perfil de Diagnóstico.

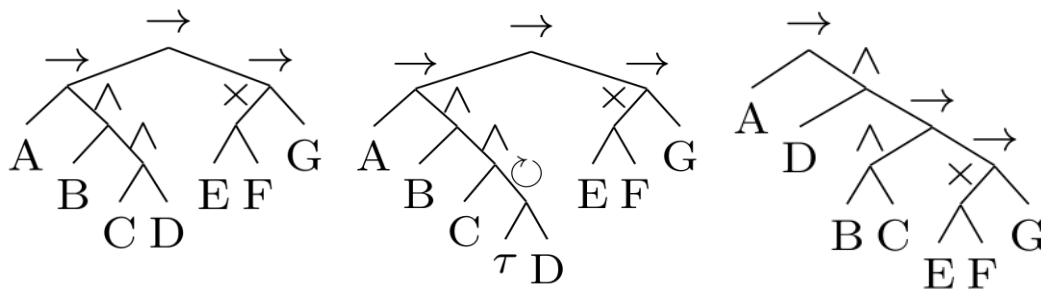


Figura 1 Representación de variantes de modelos.

El Perfil de Diagnóstico representa en forma de barras las evaluaciones de las diferentes variantes, estos resultados se obtienen midiendo la cantidad de comportamientos asumidos y descartados, durante la identificación de los diferentes patrones de control de flujo. Estos valores no son exactos son una aproximación del valor real, por lo cual se hace necesario diagnosticar los modelos para llegar a obtener un valor real. Además, el usuario al visualizar las distintas variantes, no puede inferir a simple vista cuál de ellas será la que represente eficientemente su proceso teniendo en cuenta los valores actuales, por lo tanto, se le hace difícil seleccionarlo, trayendo consigo que tome, en muchos casos, un modelo que no sea capaz de capturar eficientemente el comportamiento existente en la realidad.

En búsqueda de una solución factible se define como **problema a resolver**: ¿Cómo facilitar la selección de variantes de modelos de procesos durante la fase de diagnóstico de los mismos?

En concordancia con lo anterior se toma como **objeto de estudio**: La Minería de Procesos, centrando el **campo de acción**: La evaluación de la calidad de los modelos de procesos.

En respuesta al problema planteado se define como **objetivo general**: Desarrollar un componente para medir la calidad de las variantes de modelos que facilite la selección de las mismas durante la fase de diagnóstico de procesos.

Como **idea a defender** de esta investigación se plantea lo siguiente: Si se desarrolla un componente para medir la calidad de las variantes de modelos de procesos se facilitará la selección de variantes de modelos durante la fase de diagnóstico de los mismos.

A continuación se especifican los **objetivos específicos** de esta investigación:

- Analizar los enfoques utilizados en la medición de la calidad de los modelos de procesos para la identificación de ventajas e inconvenientes.
- Diseñar un componente para evaluar la calidad de las variantes de modelos de procesos generados durante la fase de diagnóstico de procesos.
- Implementar el componente para evaluar la calidad de las variantes de modelos de procesos generados durante la fase de diagnóstico de procesos.
- Validar la herramienta implementada mediante la utilización de registros de eventos especiales que posean la capacidad de recrear la mayor cantidad de características que pudieran existir en un entorno real.

Se espera obtener como **resultado**: un componente para medir la calidad de las variantes de modelos de procesos durante la fase de diagnóstico de procesos. Para dar cumplimiento a los objetivos específicos, se desarrollan las siguientes **tareas**:

- Análisis de los principales conceptos y trabajos relacionados con la calidad de modelos de procesos.
- Caracterización de las principales métricas relacionadas con la calidad de los modelos de procesos.
- Diseño de un componente para medir la calidad de las variantes de modelos de procesos.
- Implementación del componente diseñado permitiendo su integración a un marco de trabajo existente para la Minería de Procesos.
- Generación de los datos necesarios para validar la propuesta.

Para el desarrollo de la investigación se utilizan los siguientes métodos científicos:

Métodos teóricos

- **Analítico – sintético:** Permite realizar el estudio de las necesidades y sintetizar las ideas para lograr el desarrollo de un complemento que brinde la solución al problema planteado.
- **Histórico-lógico:** Permite conocer la evolución de los diferentes conceptos y tendencias referentes al objeto de estudio de la investigación, y de esta forma obtener un mayor conocimiento y saber aplicarlo en la solución del problema presentado.
- **Modelación:** Permite trabajar con los modelos generados a partir de los registros de eventos y de la propia solución, además de los diagramas y artefactos que genera la metodología utilizada.

Métodos Empíricos

- **Medición:** Permite aplicar métricas para medir la calidad de los modelos de procesos.

La investigación está estructurada en tres capítulos:

Capítulo 1: Se caracteriza la Minería de Proceso, haciendo énfasis en el diagnóstico de procesos. Además se realiza un estudio crítico de las diferentes dimensiones de calidad. Se analizan y evalúan las diferentes métricas que caracterizan dichas dimensiones. Se define *Variant Miner* como técnica de descubrimiento de modelos de procesos, y como herramienta para evaluarlos al marco integral CoBeFra, así como la metodología y las herramientas para el desarrollo del componente.

Capítulo 2: Se caracteriza la estructura del componente desarrollado. Además, se describe el algoritmo implementado para la creación de nuevos Registros de Eventos y Árboles de Variantes para ser usados posteriormente. Se realiza una descripción de la arquitectura de CoBeFra, y las modificaciones que se le hicieron para convertirla en una biblioteca. Se explica detalladamente todo el proceso de evaluación mediante el uso del nuevo componente.

Capítulo 3: Se describen las pruebas internas aplicadas al nuevo componente mediante el uso de cinco registros de eventos generados artificialmente, a los cuales se le sustrajo información provocando ruido en un 20%. Se hace una breve caracterización de los resultados por cada una de las etapas recogidos en una tabla y se define el tiempo promedio por cada una de las operaciones que se realizan. Se evalúa la correlación entre los resultados por cada dimensión mediante Pearson.

CAPITULO 1: Minería de Procesos, técnica para el descubrimiento de procesos y el marco integral CoBeFra

En este capítulo se presentan los principales conceptos asociados a la Minería de Procesos. En el diagnóstico de procesos, una de las fases de la Gestión de Procesos de Negocio, se han empleado técnicas de Minería de Procesos. La utilización de técnicas para el diagnóstico de procesos permite identificar los patrones de control de flujo, detectar las desviaciones presentes en el registro de eventos y ofrecer una vista general del proceso. Se realiza un análisis de las principales dificultades de las técnicas de Minería de Procesos utilizadas para el diagnóstico, reflejando como solución la utilización de la técnica *Variant Miner*. ProM es un marco de trabajo para la Minería de Procesos concebido para admitir la adición de complementos y de esta manera posibilitar el desarrollo de nuevos algoritmos y técnicas.

1.1 Minería de Procesos

La ejecución de los procesos de negocio, habitualmente es registrada por sistemas de información tales como los sistemas de Planificación de Recursos Empresariales, sistemas de Gestión de la Relación con los Clientes, sistemas de Gestión de Cadena de Suministros, u otros de propósitos más específicos. Estos registros almacenan las trazas de ejecución de los procesos de negocio y se denominan registros de eventos.

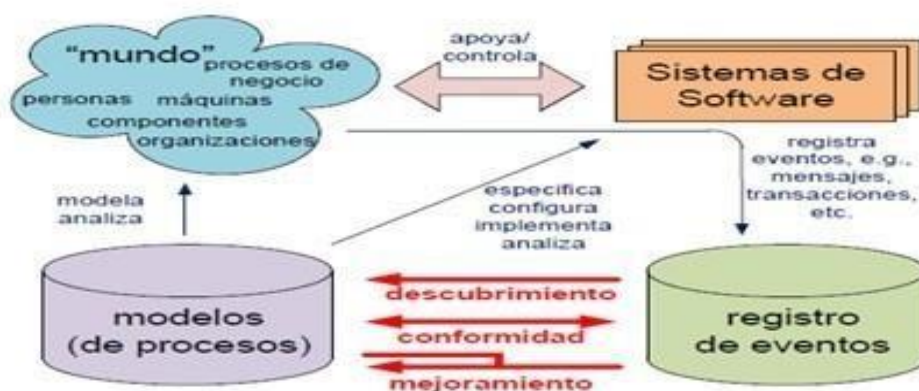


Figura 1.1: Tareas principales de la disciplina Minería de Procesos tomada de (Aalst 2012).

Las técnicas de Minería de Procesos, permiten realizar recomendaciones y predicciones, teniendo en cuenta el análisis de los datos actuales e históricos. Estas pueden clasificarse en tres grupos como se muestra en la Figura 1. El primer grupo engloba las destinadas al descubrimiento de modelos de procesos, el segundo representa el chequeo de conformidad y el tercero el mejoramiento de los procesos (Adriansyah 2011).

Un elemento fundamental en la Minería de Procesos es el diagnóstico de procesos, es decir, la evaluación automática del modelo de proceso asociado. Estos modelos descubiertos describen las dependencias causales entre las actividades del proceso (BSc. 2011).

Para evaluar estos modelos existen dos formas, modelo-modelo y modelo-log. El método modelo-modelo evalúa la paridad entre el modelo descubierto y un modelo de referencia del proceso y el método modelo-log evalúa la paridad entre el modelo descubierto y el registro de eventos a partir del cual se generó (BSc. 2011).

Las dos formas se basan en las dimensiones de calidad para evaluar los modelos, pero la variante modelo-modelo no es recomendable su aplicación en esta investigación debido a que se analizan informaciones provenientes de registros de eventos y no de modelos de referencia.

Existen cuatro dimensiones para medir la calidad de los modelos de procesos descubiertos que se denominan Aptitud, Precisión, Generalización y Simplicidad:

Aptitud: el modelo debe permitir el comportamiento presente en el registro de eventos (Adriansyah 2012).

Precisión: el modelo no debe permitir comportamiento completamente sin relación con el que se puede apreciar en el registro de eventos. La generalización de un modelo puede conducir a un ajuste insuficiente, más conocido como *underfitting*. En este caso el modelo obtenido permite mucho más comportamiento del presente en el registro de eventos (A. Adriansyah 2010).

Generalización: el modelo descubierto debe generalizar el comportamiento observado en el registro de eventos. El registro de eventos contiene solamente ejemplos de comportamiento del proceso, por ende, si el modelo está excesivamente ajustado al comportamiento observado, puede ser que otra muestra del mismo proceso genere un modelo completamente diferente (A. Adriansyah 2010).

Simplicidad: el modelo debe ser tan simple como sea posible. Relacionado con la estructura del modelo, la cual está determinada por el vocabulario de la notación de modelo seleccionado (De Weerd 2013).

En la investigación no se analiza la dimensión Simplicidad debido a que está orientada a la evaluación de modelos en notación Red de Petri.

Cada una de las dimensiones posee un conjunto de métricas por las que son medidas. Seleccionar las métricas adecuadas para evaluar las dimensiones es fundamental, para que al confeccionar los diferentes modelos se obtengan buenos resultados.

1.3 Métricas de calidad

Para determinar qué parámetros son útiles y cuáles no, se presenta un número de requisitos que una métrica debe cumplir, los que pueden ser requisitos generales que definen las propiedades deseables de una métrica y requisitos específicos de proyectos (BSc. 2011).

Requisito 1 (**validity**) las métricas se comporta de una manera predecible. Es decir, cuando la propiedad que se está midiendo mejora, también lo hace el valor de la métrica y la métrica no se ve afectada por los cambios que no afecten la propiedad que se está midiendo (BSc. 2011).

Requisito 2 (**stability**) significa que la métrica sólo mide la propiedad para ser medida y no se ve afectada por los cambios que no afecten a la propiedad que se está midiendo (BSc. 2011).

Requisito 3 (**analizability**) se ocupa de las propiedades del valor de la métrica. En la práctica, esto significa que el valor de la métrica debe estar en el dominio $[0, 1]$, y que 0 es el peor valor, mientras que 1 es el mejor valor (BSc. 2011).

Requisito 4 (**reproducibility**) significa que la métrica debe ser objetiva, de manera que cuando diferentes personas utilizan la misma métrica para medir una propiedad, sus resultados son comparable (BSc. 2011).

Requisito 5 (**localizability**) significa que no es suficiente simplemente devolver un valor, la métrica debe permitir ubicar el problema (BSc. 2011).

Requisito 6 (**Model-Log metrics**) tiene que ver con lo bien que un determinado modelo corresponde a las trazas en un registro de eventos, a diferencia de las métricas modelo-modelo que asignan una puntuación basada en lo parecidos que son un modelo descubierto y un modelo de referencia. Aquí la investigación se centra en métricas modelo-Log, ya que en situaciones de la vida real modelos de referencia no pueden estar presentes o pueden no ser exactos. Por lo tanto, en su caso la evaluación comparativa se realiza en los registros de la vida real, no sólo se mide la calidad del modelo

descubierto, sino una combinación de la calidad del registro de eventos con respecto al modelo de referencia y la calidad del modelo descubierto (BSc. 2011)

A continuación se observarán varias tablas en las cuales se recoge la información referente al cumplimiento de los requisitos de cada métrica por cada dimensión. La primera columna contendrá los nombres de las métricas, la segunda columna, en un rango de 0-6, las violaciones que ha cometido la métrica y a continuación en las próximas 6 columnas los requisitos en la nomenclatura R_i , donde $i = \{1...6\}$, el 0 indicará que la métrica no viola el requisito y el 1 que sí lo ha violado.

1.1.1 Análisis de las métricas de la dimensión Aptitud

Las métricas que evalúan la dimensión aptitud son las encargadas de analizar cuán apto es un modelo de proceso descubierto. Se han desarrollado una serie de métricas que evalúan esta dimensión. Las métricas estudiadas fueron:

Tabla 1.1: Evaluación de las métricas de la dimensión Aptitud.

Métricas	Violaciones	R1	R2	R3	R4	R5	R6
Completeness	1	1	0	0	0	0	0
Completeness (PF complete)	2	0	0	1	0	1	0
Continuos Parsing Measure	2	0	0	1	0	1	0
Fitness	0	0	0	0	0	0	0
HMM event fitness	1	0	0	0	0	0	1
HMM model Fitness/HMM trace fitness	1	1	0	0	0	0	0
Parcing Measure	1	1	0	0	0	0	0

Recall	2	1	0	0	0	0	1
--------	---	---	---	---	---	---	---

Teniendo en cuenta el cumplimiento de los requisitos reflejado en la tabla anterior, se selecciona la métrica Fitness.

La métrica **Fitness** se define de la siguiente forma:

$$f = \frac{1}{2} \left(\frac{1 - \sum_{i=1}^k n_i m_i}{\sum_{i=1}^k n_i e_i} \right) + \frac{1}{2} \left(1 - \sum_{i=1}^k n_i \frac{r_i}{\sum_{i=1}^k n_i p_i} \right)$$

Donde:

n_i : Número de instancias del proceso.

m_i : Número de tokens ausentes.

K : Número de trazas diferentes agregadas al registro de eventos.

r_i : Número de tokens restantes.

c_i : Número de tokens consumidos.

p_i : Número de tokens producidos durante la generación del registro de eventos.

Esta métrica analiza los tokens y las actividades que se generan durante la ejecución de un registro de eventos. Así mide la cantidad de actividades existentes, que pueden estar representadas o no, en el modelo. El valor de la métrica oscila entre 0 y 1. La métrica analiza redes de Petri y penaliza el comportamiento adicional representado (BSc. 2011).

1.1.2 Análisis de las métricas de la dimensión Precisión

Las métricas de esta dimensión evalúan que el comportamiento presente en el modelo se encuentre presente en el registro de eventos. Esta dimensión es importante cuando se quieren detectar anomalías en las empresas, ya sean fraudes o desviaciones en el proceso. Las métricas estudiadas son:

Tabla 1.2: Evaluación de las métricas de la dimensión Precisión.

Métricas	Violaciones	R1	R2	R3	R4	R5	R6
Appropriateness	3	1	1	0	0	1	0
Behavioral Appropriateness	3	1	0	1	0	1	0
Behavioral Precision	2	0	0	0	0	1	1
ETCprecision	1	0	0	0	0	0	1
HMM model precision	1	1	0	0	0	0	0
HMM total precision	1	1	0	0	0	0	0
Improved Appropriateness	3	1	1	0	0	1	0
Improved/Advanced Behavioral Appropriateness	0	0	0	0	0	0	0
Precision	1	0	0	0	0	0	1
Soundness	1	1	0	0	0	0	0

Las métricas Behavioral Precision, ETCprecision y Precision, requieren de un modelo de referencia, además del registro de eventos y un modelo extraído como se argumenta antes, no podemos asumir que existe un modelo de este tipo o que es preciso. Por último, la métrica Behavioral Appropriateness penaliza un modelo basado en el número medio de actividades que están habilitados durante la reproducción. Esto significa que un modelo que permite paralelismo siempre tiene una puntuación de precisión que es inferior a la óptima, incluso si el modelo extraído corresponde exactamente con el comportamiento en el registro de eventos. Esto deja solo una métrica que se puede utilizar, Improved/Advanced Behavioral Appropriateness (BSc. 2011).

La métrica **Improved/Advanced Behavioral Appropriateness** se define de la siguiente forma:

$$2 * S_P^m \vee \dot{\iota}$$

$$\dot{\iota} \frac{(S_F^l \cap S_F^m)}{2 * (S_F^m)} + \frac{(S_P^l \cap S_P^m)}{\square}$$

$$a_B = \dot{\iota}$$

S_f : Relación de seguidores.

S_p : Relación de predecesores.

La relación S_f y S_p es simétrica, cuando existe una actividad que no es seguidora ni es seguida por ninguna otra no se representa. Esta métrica se enfoca en analizar las relaciones existentes en el modelo, desechando las actividades aisladas (ROZINAT 2008).

1.1.3 Análisis de la métrica de la dimensión Generalización

La dimensión generalización es la encargada de que el modelo descubierto generalice todo el comportamiento registrado. Es decir, debe mostrar todo el comportamiento que está presente en el registro de eventos sin llegar a un nivel de detalle elevado. Esta dimensión no es muy utilizada en estudios de Minería de Procesos en entornos reales (Brouke 2014). En el marco de trabajo CoBeFra están implementadas una serie de métricas de esta dimensión. Las métricas estudiadas son:

Tabla 1.3: Evaluación de las métricas de la dimensión Generalización.

Métricas	Violaciones	R1	R2	R3	R4	R5	R6
Behavioral Recall	1	0	0	0	0	0	1
Causal Footprints	2	0	0	0	0	1	1
Negative Event Generalization	0	0	0	0	0	0	0

Las métricas Behavioral Recall y Causal Footprints necesitan un modelo de referencia además del registro de eventos y no se puede asumir que existe un modelo o que sea general. Además, Causal Footprints no brinda la información necesaria del problema por el cual su valor podría ser bajo.

Luego de analizados los resultado se selecciona la métrica **Negative Event Generalization** ya que es la única que cumple con todos los requisitos.

La métrica **Negative Event Generalization** se define de la siguiente manera:

$$r_B^P = \left(\frac{\sum_{i=1}^K n_i TP_i}{\sum_{i=1}^K n_i TP_i + \sum_{i=1}^K n_i FN_i} \right)$$

El resultado de la métrica se obtiene mediante el análisis de cada secuencia de eventos ejecutada. Los valores de TP y FN se inicializan en cero. A partir de estos valores cada secuencia es analizada y cada vez que ocurre una transición los valores de TP se incrementan en uno. Además, cuando la transición no se activa pero se obliga en el modelo, el valor de FN se incrementa en uno.

Donde:

K: Número de secuencias agrupadas.

n_i : Número de instancias de procesos.

TP_i : Número de actividades analizadas correctamente.

FN_i : Número de eventos disparados por cada transición en las secuencias agrupadas.

Esta métrica analiza los eventos positivos y negativos generados en el modelo. Los eventos positivos son aquellos que se encuentran en el registro de eventos, y los negativos los que genera el algoritmo de descubrimiento ante cualquier característica del registro de eventos que incide en él. Los valores de esta métrica oscilan entre 0 y 1 (BSc. 2011).

El estudio de las métricas por cada dimensión permite seleccionar cuáles utilizar para evaluar los modelos de procesos descubiertos teniendo en cuenta sus características. Las métricas son importantes para obtener modelos de procesos de calidad evaluando los modelos descubiertos.

1.2 Descubrimiento de modelos de procesos

El proceso de descubrimiento es una de las tareas más complejas y atendidas en la Minería de Procesos. Su objetivo es la construcción de un modelo de proceso a partir de la información obtenida de un registro de eventos (Aalst 2012). El descubrimiento de un modelo de proceso requiere que el registro de eventos contenga información suficiente, es decir, no exista ausencia de información.

Entre las características propias de los registros de evento se encuentran:

- **Ruido:** según (Aalst) es el comportamiento raro e infrecuente registrado en el registro de eventos y que no es representativo del comportamiento típico o común del proceso.
- **Ausencia de información:** ausencia en las trazas, de eventos relativos a la ejecución de actividades, que dificulta la identificación de patrones de control de flujo del proceso. Si las trazas no contienen ningún evento de alguna actividad, a este tipo de actividad se le denominará actividad invisible (Herrera 2012).
- **Tareas Duplicadas:** son los eventos que se encuentran de forma reiterada en un registro de eventos y aunque poseen el mismo identificador, hacen referencia a actividades diferentes (Aalst 2012).

1.2.1 Minería de Variantes

Dentro del grupo de técnicas de Minería de Procesos se encuentra *Variants Miner*, más conocida como “Minería de Variantes”, la cual resuelve afectaciones tales como el ruido y la ausencia de información en la comprensión del proceso.

La técnica *Variants Miner* está conformada por cuatro etapas que se ejecutan secuencialmente y al igual que el resto de las técnicas para el diagnóstico, utiliza como entrada un registro de eventos (Figura 1). El objetivo perseguido por *Variants Miner* es construir variantes de procesos a partir de diferentes descomposiciones en subprocesos considerando patrones de control de flujo. Las variantes de descomposición obtenidas se muestran en un Árbol de Variantes, asociado al cual se obtiene un Perfil de Diagnóstico. *Variants Miner* ha sido implementada como complemento del marco de trabajo ProM. Genera como salida un Árbol de Variantes, con su respectivo Perfil de Diagnóstico (Damián Pérez Alfonso 2015a).

Variantes de proceso: Las variantes de un modelo de proceso o variantes de procesos, son modelos de un proceso que describen el mismo proceso de negocio, y poseen algunas diferencias estructurales. Las diferencias están dadas por los patrones de control de flujo que se utilizan en

secciones equivalentes del proceso y la presencia de determinadas actividades (Damián Pérez Alfonso 2015a).

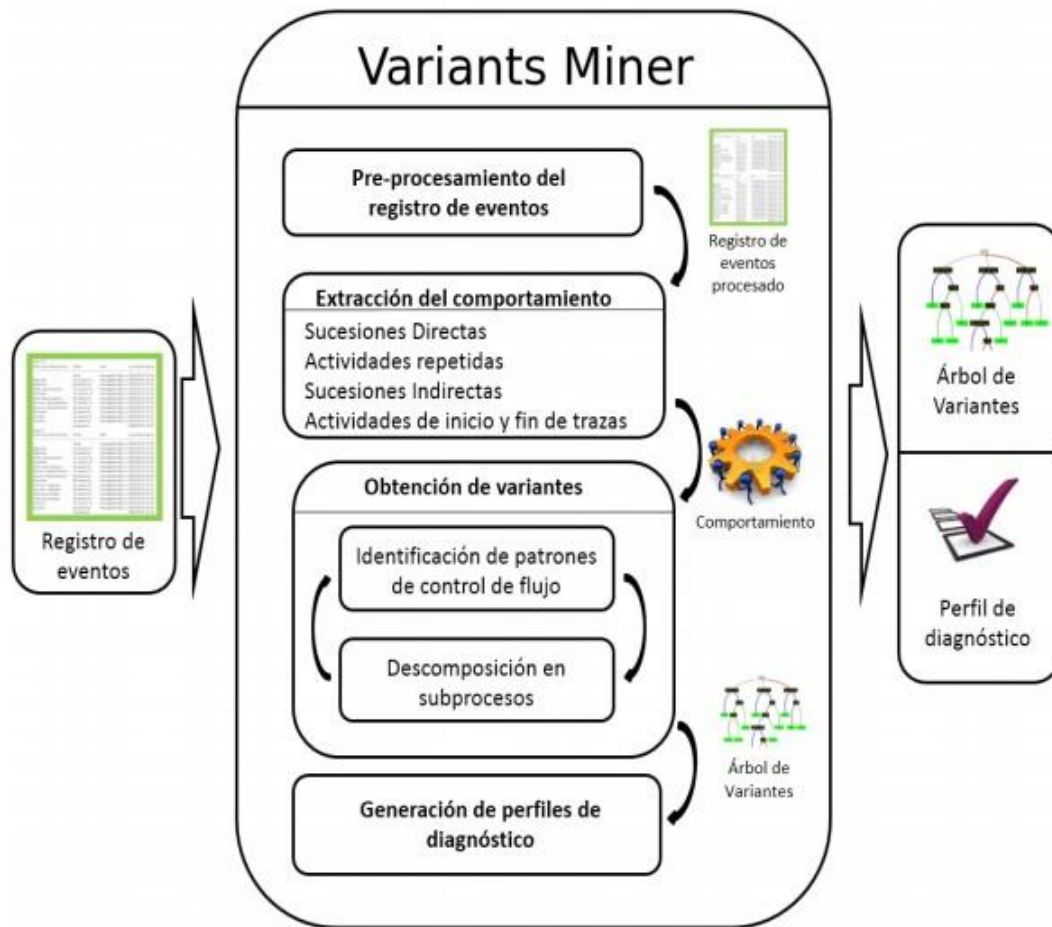


Figura 1.2: Componentes de la técnica Variants Miner tomado de (Damián Pérez Alfonso 2015a).

Árbol de variantes: un Árbol de Variantes (V_P) es una representación de diferentes descomposiciones en subprocesos aplicadas al proceso P a partir de un registro de eventos (Gamma et al.). Se denota a N_V como el conjunto de todos los nodos de V_P y por $\Gamma_V \subset N_V$ el conjunto de todos los nodos hoja. Se define $n_{s_i} \in N_V$: $0 < i \leq |S_P|$ como un nodo subproceso que representa al subproceso $s_i \in S_P$ y l_i la sección del registro de eventos asociada a s_i , y $n_{i w_j} \in N_V \wedge n_{i w_j} \in / \Gamma_V$: $1 < j \leq |W|$, como nodo patrón que representa una descomposición del subproceso de acuerdo a cierto patrón de control de flujo $w_j \in W$. El nodo raíz de V_P es n_{s_1} ya que $s_1 \in S_P \wedge s_1 = P$. Un nodo subproceso n_{s_i} puede tener ninguno o varios nodos hijos $n_{i w_j}$. Un nodo $n_{i w_j}$ tiene dos o más nodos hijo n_{s_i} (Reynaldo López Jiménez 2014).

Perfil de Diagnóstico: se denota por D_V al Perfil de Diagnóstico del Árbol V_P tal que: $\forall n_{i w_j} \in N_V \exists d_{i w_j} \in D_V$. Denotando por f a la estimación de Aptitud, por p a la estimación de Precisión, el

comportamiento descartado como $\hat{\eta}_{ijw}$ y el comportamiento asumido como θ_{ijw} . La información de diagnóstico d_{ijw} está constituida por la tupla $\{f, \rho, \eta_{ijw}, \theta_{ijw}\}$. Los comportamientos η_{ijw} y θ_{ijw} se refieren al subproceso s_i del nodo n_{si} padre de n_{ijw} . Se denota por β_{wjl_i} al comportamiento del patrón w_j expresado en la sección del registro de eventos l_i . El valor de f expresa la relación entre $\hat{\eta}_{ijw}$ y β_{wjl_i} . De manera similar ρ expresa la relación entre $\hat{\theta}_{ijw}$ y β_{wjl_i} . En el Perfil de Diagnóstico se incluyen los comportamientos descartados, ya que estos pueden ser considerados como ruido, anomalías o desviaciones del proceso. A su vez, los comportamientos asumidos durante la descomposición, reflejan situaciones de ausencia de información, las cuales pueden estar asociadas también a anomalías durante la ejecución del proceso. La presentación de ambos tipos de comportamiento como resultado de la técnica, responde directamente a los intereses del diagnóstico. Se pretende, de esta manera, contribuir a la comprensión del proceso en la dimensión pragmática (Reynaldo López Jiménez 2014).

1.3 Herramientas para evaluar modelos

La evaluación de las diferentes variantes de modelo de proceso es esencial para los usuarios, lo cual de acuerdo al resultado que se obtenga servirá como base para escoger el más apropiado. Para realizar esta evaluación se han implementado un conjunto de herramientas integradas en un marco.

1.3.1 Un análisis de conformidad del marco integral CoBeFra

CoBeFra, un conjunto de herramientas utilizables integral para evaluar la bondad de un modelo de proceso o para facilidad de referencia del rendimiento de diferentes modelos en contra entre sí, mediante múltiples métricas de chequeo de conformidad. Como tal se presenta la arquitectura de un marco de referencia integral extensible, lo que permite un cálculo uniforme, comparativo y repetible de las métricas de proceso de conformidad. Este tipo de arquitectura se considera como muy valiosa para los investigadores de la Minería de Procesos, ya que facilita significativamente el desarrollo y la evaluación de descubrimiento de procesos, así como técnicas de control de conformidad (Brouke 2014).

1.3.1.1 Facilidad de uso

El primer requisito de diseño pone énfasis en la facilidad de uso. Con este marco de referencia, el objetivo es ofrecer una interfaz sencilla para importar registros de eventos y modelos de procesos, para el mapeo de cada evento clase (es decir, el nombre de la actividad y del ciclo de vida de transición) a una de las actividades (tareas, transiciones) en el modelo de proceso, para la configuración de las diferentes métricas y, por último, para la inspección y la exportación de los

resultados obtenidos. Con el fin de simplificar las instalaciones experimentales que requieren una gran cantidad de secuencias de comandos y el procesamiento por lotes, se ha reforzado el desacoplamiento de la interfaz de usuario y la lógica de programación en CoBeFra, permitiendo que cada paso (log y configuración modelo, la configuración métrica y procesamiento resultado) sea ejecutado de forma pura (Brouke 2014).

1.3.1.2 Gestión de computación

Una de las limitaciones del marco es que no es fácil de configurar un entorno en el que múltiples comprobaciones de conformidad se pueden ejecutar al mismo tiempo. Por lo tanto, el enfoque es permitir primero al usuario configurar todas las entradas de registro de modelo, junto con la lista de métricas deseadas para funcionar y su configuración. Después, el cálculo de las métricas en sí se inicia; se ha puesto en marcha un gestor de computación que permite ejecutar varios cálculos métricos en paralelo. Por último, ya que algunas métricas pueden consumir una gran cantidad de tiempo antes de terminar, una opción para cancelar de forma manual y automáticamente una métrica que se añadió al procedimiento de cálculo. Esto permite a los investigadores imponer fácilmente los límites basados en el tiempo durante la ejecución de experimentos (Brouke 2014).

1.3.1.3 Extensibilidad

Por último, el marco está diseñado para poder ser ampliado con otras o futuras métricas de conformidad (Brouke 2014). El marco CoBeFra se integra en la PROM 6 y reutiliza varias bibliotecas y componentes existentes, disponibles en el marco de trabajo ProM. Se puede iniciar como cualquier otro plugin de ProM. La importación y exportación de objetos de entrada se realiza a través de la PROM estándar por la arquitectura proporcionada, y la visualización de los resultados obtenidos de las métricas se disoció en un plugin de visualización independiente. Una vez que se inicia CoBeFra, el marco descarta todas las dependencias de las bibliotecas (Brouke 2014).

1.3.1.4 Comparación con otras herramientas

La herramienta CPN Tools es la más extendida y avanzada para la edición, simular y analizar las redes de Petri coloreadas. Aunque las últimas versiones han hecho que sea más fácil de modelar una simple (no coloreada) red de Petri, la interfaz de usuario particular de esta herramienta viene con una curva de aprendizaje bastante empinada (Brouke 2014).

CoBeFra (Marco Integral): Es una plataforma que tiene implementada una serie de métricas que permiten la evaluación de los modelos de procesos. Las métricas implementadas en él, están divididas en las cuatro dimensiones de calidad. CoBeFra analiza registro de eventos y modelos de procesos que deben estar en redes de Petri. Los resultados obtenidos de la evaluación de las métricas con respecto un modelo correspondiente con el registro de eventos, pueden ser exportados (De Weerd 2013).

Luego de analizadas ambas herramientas, se determina que la idónea para utilizar en la evaluación de los modelos es CoBeFra, debido a que es de fácil utilización, permite adicionar otras métricas que sean implementadas posteriormente y principalmente el usuario no necesariamente debe contar con un vasto conocimiento, sino con solo conocer los elementos básicos puede utilizarla.

1.4 Herramientas y Tecnologías

Para el desarrollo de la investigación se tuvieron en cuenta una serie de herramientas y tecnologías, que permiten la realización de la experimentación y el desarrollo del componente.

1.4.1 ProM

ProM es un marco de trabajo para el desarrollo de herramientas de Minería de Procesos en un ambiente estandarizado. Está desarrollado en Java, está concebido para admitir la adición de complementos y de esta manera posibilitar el desarrollo de nuevos algoritmos y técnicas en el campo de la Minería de Procesos. Los complementos necesitan determinada cantidad de parámetros de entrada y producen uno o varios objetos de salida. Los parámetros de entrada pueden ser registros de eventos u objetos obtenidos a partir del procesamiento realizado por otros complementos, mientras que los objetos de salida obtenidos pueden ser empleados como parámetros de entrada de otros complementos. Este marco de trabajo cuenta con más de 550 complementos, cada uno de los cuales posibilita realizar diferentes análisis (Aalst 2011).

1.4.2 CoBeFra

CoBeFra es una plataforma que tiene implementada una serie de métricas que permiten la evaluación de los modelos de procesos. Las métricas implementadas en él, están divididas en las cuatro dimensiones de calidad. Los modelos de procesos que analiza solo pueden estar en redes de Petri. Los resultados obtenidos de la evaluación de las métricas a un registro de eventos pueden ser

exportados. Es un sistema implementado en Java, al cual se le pueden incorporar otras métricas según se vayan desarrollando (De Weerd 2013).

1.4.3 Entorno de desarrollo integrado

De acuerdo con ORACLE (2013), **NetBeans IDE** es un proyecto exitoso de código abierto con una gran cantidad de usuarios, una comunidad en constante crecimiento y cerca de 100 socios en todo el mundo. Proporciona un amplio soporte para las últimas tecnologías de Java. Soporta integración con diferentes herramientas para el control de versiones y Sistemas Gestores de Bases de Datos (SGBD). Es un producto libre, gratuito y sin restricciones de uso. Su empleo está determinado por la experiencia del equipo de desarrollo en su utilización, en virtud de minimizar el tiempo necesario para la familiarización con nuevos lenguajes y herramientas de desarrollo.

1.4.4 Control de Versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo. Te permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo (Hosted on GitHub. 2014).

Git se presenta como un sistema distribuido, guarda a lo largo del tiempo una fotografía nueva que encierra el estado actual del archivo con todos sus cambios. Sin embargo, cuando un archivo no cambia, en lugar de guardar la misma fotografía varias veces, guarda una referencia a esa fotografía. De esta forma se optimizan los recursos del sistema. Todo es guardado en Git no por nombre, sino por la suma de comprobación de sus contenidos. Otro de los principios fundamentales de Git es el de no remover información. Cuando introduces un cambio en tu proyecto simplemente añade más información al repositorio, lo que permite que sea muy difícil estropear algo por error o que no puedas deshacer esas modificaciones. Además es Software Libre distribuido bajo los términos de la Licencia Pública General GNU versión 2 (Hosted on GitHub. 2014).

1.4.5 Ingeniería de software asistida por computadora

Para (Sommerville 2007)(*Computer Aided Software Engineering*) es el nombre con el que se identifica la herramienta utilizada para apoyar las actividades del proceso de software.

Visual Paradigm es una herramienta CASE multiplataforma que contribuye al desarrollo de sistemas de software fiables, mediante un enfoque orientado a objetos. Su preferencia está determinada por

las oportunidades que ofrece para la construcción de aplicaciones de calidad, con mayor rapidez y menor costo. Soporta el ciclo completo de desarrollo de software y permite su documentación en diferentes formatos, empleando UML como lenguaje de modelado. Posibilita la generación de código en una amplia gama de lenguajes de programación y se integra con los principales Entornos de Desarrollo Integrado (Paradigm 2005)

1.4.6 Metodología de desarrollo

El proceso de desarrollo de software, definido por Jacobson et al. (2000), cobra vital importancia la elección de la metodología de desarrollo apropiada. La misma, guía el proceso de desarrollo para alcanzar la satisfacción del cliente y del equipo.

La evaluación de las metodologías y la necesidad de una variante flexible a los cambios, preparada para ser ejecutada por cualquier equipo, dígase grande o pequeño, y agrupe todas las ventajas de las diferentes metodologías ya sean ágiles o pesadas, permitió identificar el “Proceso Unificado Ágil” como la alternativa más acertada (Sánchez 2015).

Para la actividad productiva de la UCI, se decide hacer una variación de la metodología AUP, de forma tal que la misma se adapte al ciclo de vida definido para la actividad, apoyándose en el Modelo CMMI-DEV v1.3, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. De sus tres fases: sólo se ejecutaran Ejecución y Cierre, y las disciplinas: Análisis y Diseño, Implementación y Pruebas Internas (Sánchez 2015).

Conclusiones parciales del capítulo

El estudio detallado de las técnicas de Minería de Procesos, permitió identificar la técnica Minería de Variantes como el eslabón principal para el desarrollo del componente, ya que resuelve afectaciones presentes en el registro de eventos como el ruido y la ausencia de información en la comprensión del proceso.

El análisis de las herramientas para evaluar modelos, permitió identificar al marco integral CoBeFra como herramienta principal para la evaluación de los modelos, mediante el empleo de las métricas de calidad seleccionadas.

Para el desarrollo del componente se definió como Metodología AUPuci en su variante cuatro, el entorno de desarrollo se utiliza Netbeans 8.0, para el modelado el Visual Paradigm en su versión 8.0, como marco de trabajo se utiliza el ProM.

CAPITULO 2: Propuesta de Solución

2.1 Introducción

El componente *Variant Miner* tiene integrados 2 plugin. El primero es *Variant Tree*, que a partir de un registro de eventos crea el Perfil de Diagnóstico y el Árbol de Variantes. Este plugin tiene implementado un algoritmo que posteriormente modifica el Árbol, creando una descomposición más sencilla, y reduce el registro de eventos. El segundo es *Convert VTree To Petri Net*, que recibiendo como entrada el Árbol modificado crea la red de Petri. La biblioteca CoBeFra se integra en el desarrollo de esta nueva versión del componente, la cual a partir del registro de eventos modificado y la red de Petri creada, calcula las métricas que tiene definidas, y los resultados se los envía al Perfil de Diagnóstico antes mencionado para ser mostrados al usuario.

A continuación se realiza una descripción más detallada de todo el proceso de evaluación de los modelos mediante cada uno de los elementos que integran a *Variant Miner*.

2.2 Generar un Árbol de Variantes

El componente *Variant Miner* es el encargado de obtener el Árbol de Variantes y el Perfil de Diagnóstico con la ayuda del Plugin *Variant Tree* (Figura 2.1).

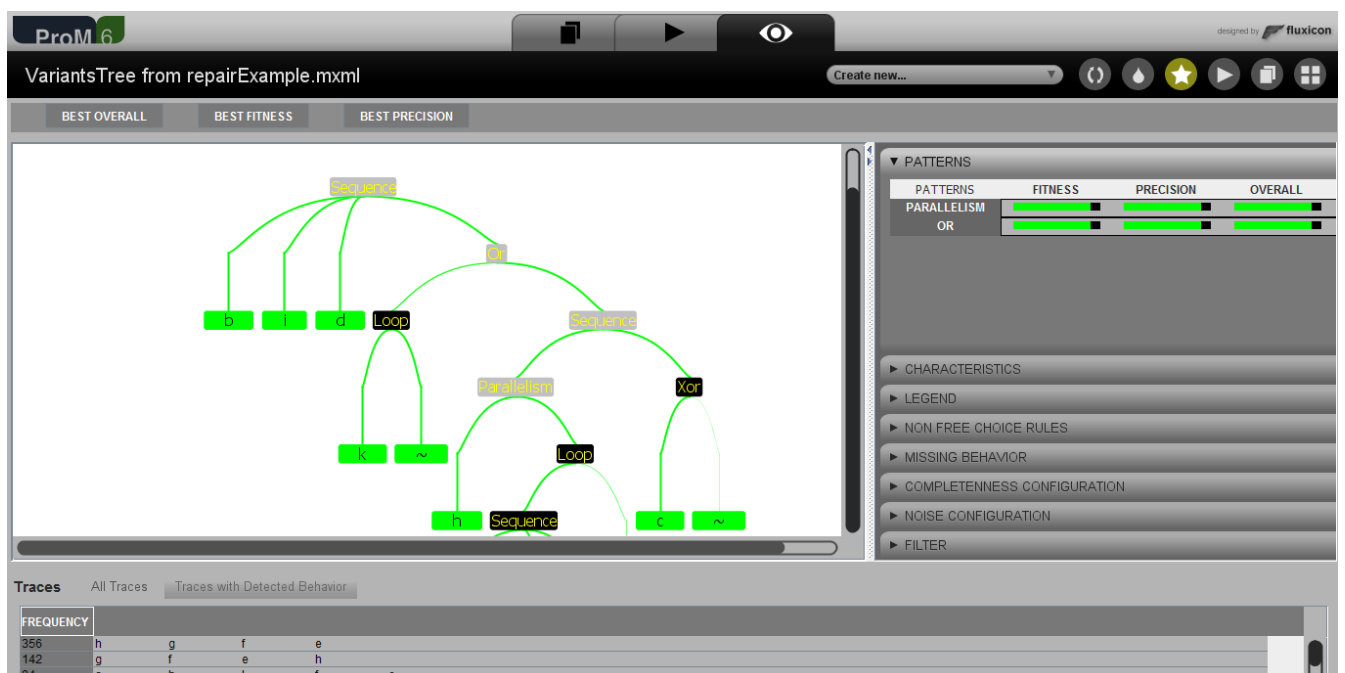


Figura 2.1: Interfaz principal al ejecutar el Variant Miner.

La idea general consiste en encontrar las posibles variantes de descomposición de un subproceso bajo determinados criterios impuestos por el usuario. Estos criterios son definidos en términos de umbrales para ruido (T_n) y completitud (T_c). A partir del registro de eventos y los umbrales definidos se genera el *Árbol de Variantes* (Figura 2.2) mediante el complemento *Variant Tree* el cual se encarga del pre-procesamiento (Reynaldo López Jiménez 2014).

Durante la etapa de pre-procesamiento se extraen y codifican las actividades presentes en el registro de eventos. Son agrupadas las trazas con igual secuencia de actividades y posteriormente ordenadas de acuerdo a su frecuencia. Son detectadas las características generales presentes en el registro de eventos, como son la cantidad de trazas diferentes y la cantidad de eventos diferentes. Posteriormente se procede a la extracción del comportamiento identificativo de los patrones de control de flujo, contenido en el registro de eventos ($B_{l_i} = B_{w_s l_i} \cup B_{w_x s_i} \cup B_{w_o s_i} \cup B_{w_p s_i} \cup B_{w_i s_i} | w_j \in W$). Para la obtención del árbol de variantes se realizan dos procedimientos fundamentales, la identificación de los patrones de control de flujo presentes en el subproceso y luego la descomposición del subproceso según los patrones identificados. A partir de cada nodo en el árbol de variantes se construye el Perfil de Diagnóstico, que muestra los resultados brindados por la biblioteca CoBeFra, que se describe más adelante (Reynaldo López Jiménez 2014).

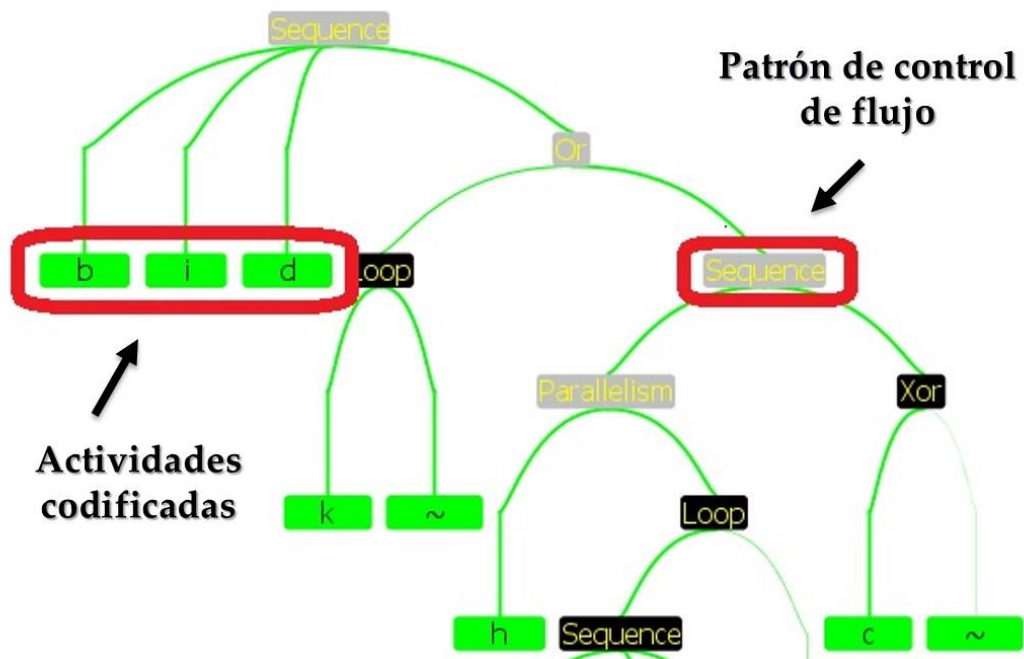


Figura 2.2: *Árbol de Variantes*.

Este complemento está estructurado en 7 paquetes (Figura 2.3): *configuration*, *preprocessing*, *tree*, *diagnosis*, *behavior*, *search* y *patterns*. Cada uno de estos paquetes contiene los componentes necesarios para la realización de funciones específicas dentro del complemento (Reynaldo López Jiménez 2014).

En el paquete *configuration* se encuentran los componentes encargados de obtener y almacenar los parámetros de configuración introducidos por el usuario. Los datos obtenidos son empleados durante la identificación de los patrones de control de flujo. El paquete *preprocessing* contiene los componentes encargados de realizar la codificación de las actividades presentes en el registro de eventos, extraer los datos generales del registro de eventos y agrupar las trazas iguales (Reynaldo López Jiménez 2014).

Los componentes pertenecientes a la estructura del Árbol de Variantes se encuentran ubicados en el paquete *tree*. En el paquete *diagnosis* se ubican los componentes asociados al Perfil de Diagnóstico. Los componentes presentes en el paquete *behavior* realizan la extracción del comportamiento relevante dentro del proceso. Dentro del paquete *search* se encuentran los componentes correspondientes al algoritmo de búsqueda empleado durante el descubrimiento de los patrones de control de flujo. Por último, en el paquete *patterns* se encuentran los componentes empleados para realizar el descubrimiento de los patrones de control de flujo (Reynaldo López Jiménez 2014).

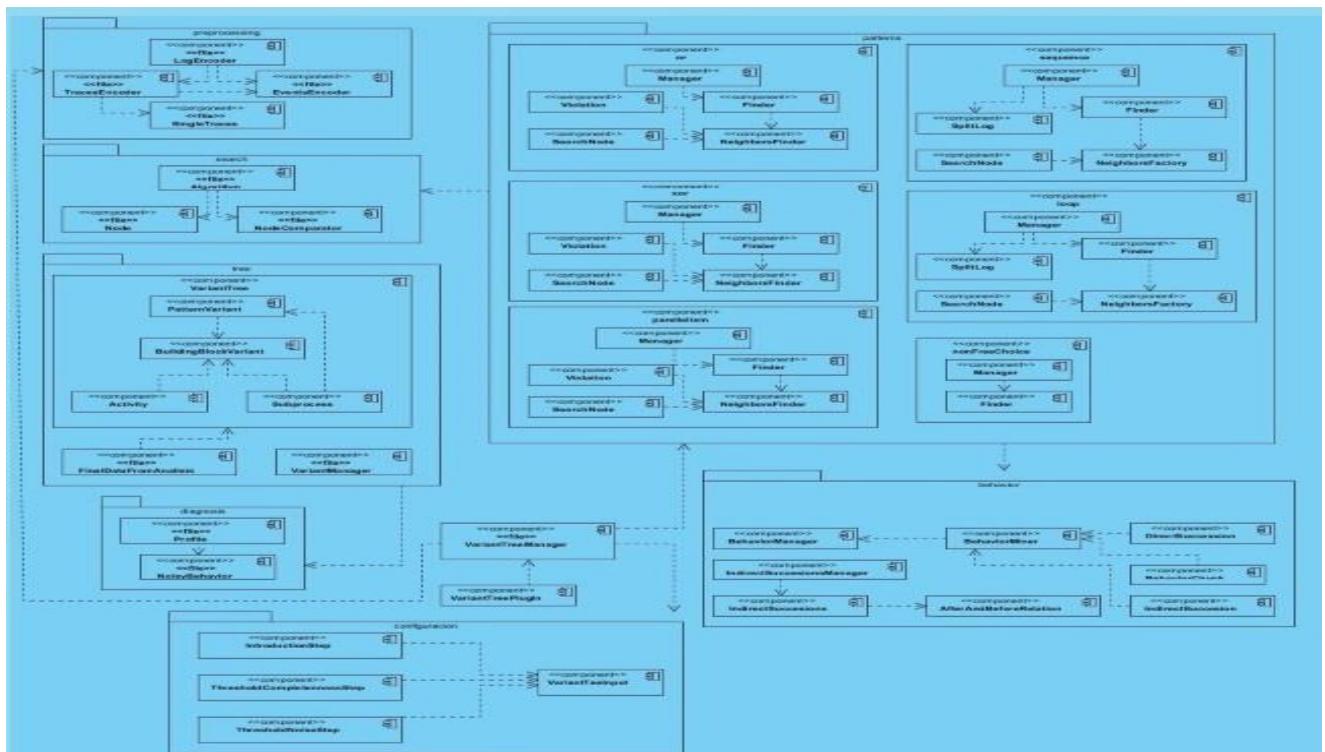


Figura 2.3: Estructura del componente Variant Miner representada mediante paquetes.

El usuario tiene la posibilidad de analizar el modelo completo y cada uno de sus subprocesos, seleccionando cada uno de los patrones de control de flujo presentes en cada nivel.

Internamente se ejecuta un algoritmo, añadido a la clase Patterns situada en el paquete VariantTree/Visualization/BehaviorPanel, mediante el cual, primero: se reduce el registro de eventos dejando solo las actividades que corresponden al subproceso que se analiza y segundo: se codifican las actividades alfabéticamente en orden Z-A, de manera tal que el subproceso quede reducido a las actividades relacionadas directamente con el patrón (Figura 2.4), para que sea utilizado como entrada para la conversión a red de Petri.

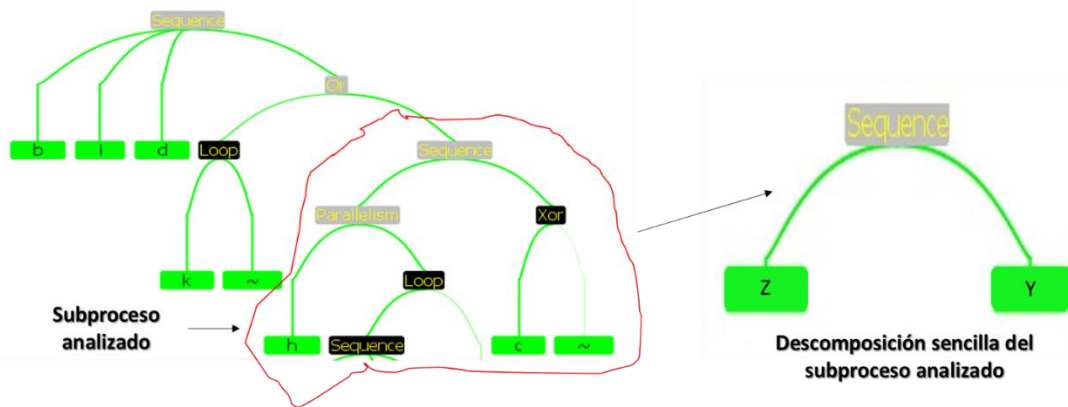


Figura 2.4: Reducción del árbol del subproceso analizado.

El algoritmo desarrollado tiene la siguiente estructura:

1. Determinar las reglas de conversión entre las actividades del subproceso.
Para cada hijo del subproceso actual, que sea un subproceso, se le asigna una etiqueta y a su vez se hace corresponder con las actividades del subproceso en cuestión. La complejidad algorítmica es de $O(n^3)$.
2. Filtrar los registros de eventos: reducir el registro de eventos a las actividades pertenecientes al subproceso que se está analizando.
Para el filtrado de los registros de eventos se utiliza el filtro XEventCondition, el cual filtra el registro de eventos por las actividades que pertenecen a un subproceso determinado. Luego se recorre el mismo filtrado y se reduce, teniendo en cuenta las reglas de conversión, creándolo como uno nuevo. La complejidad algorítmica de este paso es $O(n^3)$.
3. Editar subproceso: reduce el árbol en correspondencia con las etiquetas asignadas a los subprocesos.

Primeramente se recorren los hijos de la variante actual y, por cada hijo de tipo subproceso, se adiciona una actividad en correspondencia con la etiqueta asignada al mismo creando un nuevo árbol. La complejidad algorítmica es de $O(n)$.

4. Exportar el registro de eventos modificado.

Para exportar se utiliza el plugin ExportLongXES especificando la dirección donde se guarda el fichero y el registro de eventos a exportar. La complejidad algorítmica en este caso es $O(n)$.

5. Crear red de Petri a partir del árbol creado con el subproceso.

Para crear una red se utiliza el plugin VariantToPetriManager que recibe como entrada un subproceso y devuelve la red de Petri correspondiente. La complejidad algorítmica es igualmente $O(n)$.

6. Exportar la red de Petri.

Para la exportación se utiliza el método ExportPetriNetToPNMLorEPNMLFile, que recibe como entrada la red y la dirección donde se guarda. La complejidad algorítmica es de $O(n)$.

7. Calcular las métricas.

El proceso de calcular las métricas lo realiza la versión modificada de CoBeFra, como una biblioteca externa, la cual después de obtenidos los resultados, los exporta en un archivo "result.txt". La complejidad algorítmica de realizar esta operación es de $O(n^2)$.

8. Mostrar los resultados.

Se lee el fichero "result.txt" y se hace corresponder los resultados con las dimensiones de calidad presentes en el Perfil de Diagnóstico.

La complejidad del algoritmo de forma general es de $O(n^3)$, ello se debe a que dentro del paso 1 se hacen muchos recorridos dentro del propio registro para identificar cada uno de los subprocesos dentro del árbol.

Luego se hace necesario generar la red de Petri teniendo como base el nuevo árbol, debido a que para el cálculo de las métricas para evaluar los modelos, la biblioteca CoBeFra necesita el registro de eventos, antes creado y un modelo en red de Petri.

2.3 Generar un modelo en red de Petri

Para generar un modelo en red de Petri se utiliza el complemento Convert Vtree To Petri Net (traducción al español, Convertir un Árbol de Variantes a red de Petri) el cual a partir de un árbol devuelve un modelo en notación red de Petri (Figura 2.5), donde se representan las actividades y las relaciones entre ellas. Este modelo puede ser empleado por otros complementos para ser analizado.

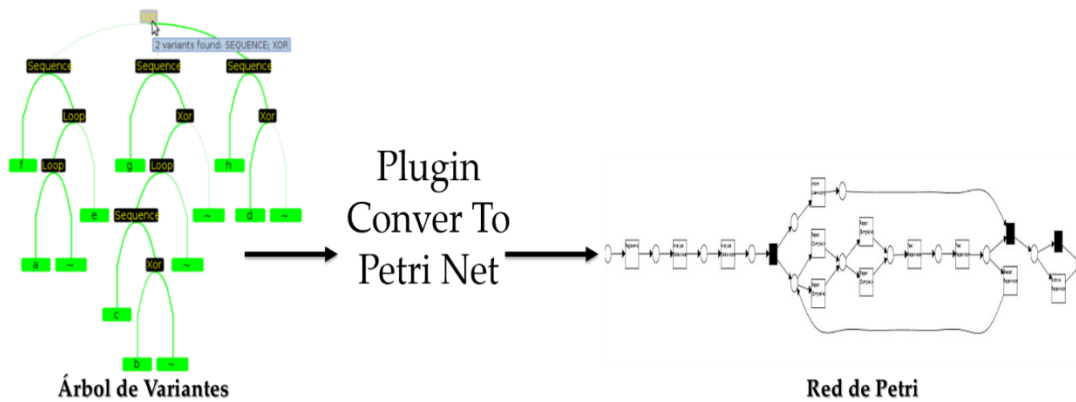


Figura 2.5: Representación del funcionamiento del plugin para convertir a red de Petri.

Una “red de Petri” es un grafo dirigido bipartito formado por nodos lugares y nodos transiciones conectados por arcos (Chaviano 2014) y se define de la siguiente forma:

Una red de Petri es un triplero $N = \{P, T, F\}$ 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, llamado relación de flujo (Aalst 2011). La Figura 2.6 es un ejemplo de una red de Petri c

Las redes de Petri emplean las siguientes estructuras de control para su creación (Chaviano 2014):

- Causalidad
- Paralelismo (AND-split - AND-join)
- Selección (XOR-split – XOR-join)
- Iteración (XOR-join - XOR-split)
- Restricciones de Capacidad (lazo de retroalimentación, exclusión recíproca, alternar).

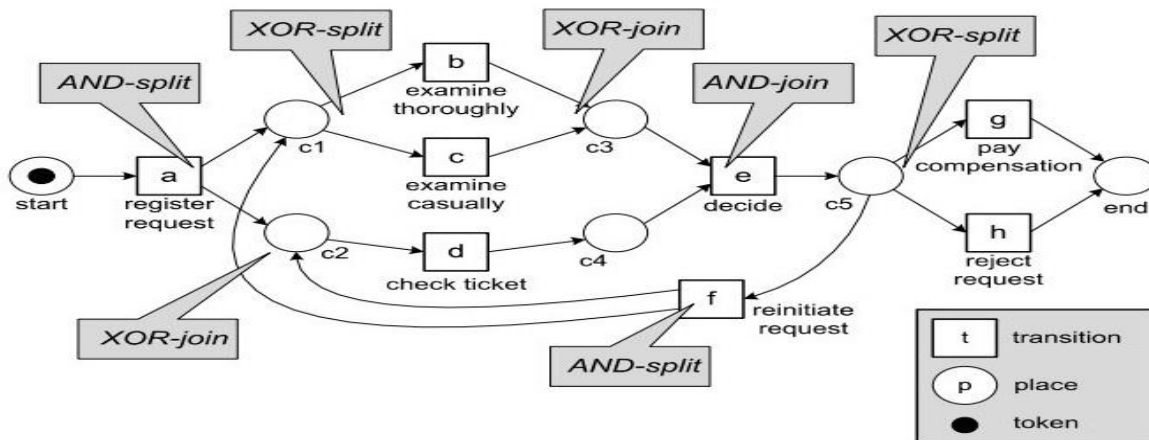


Figura 2.6: Modelo en red de Petri tomado de (Chaviano 2014).

En la figura 2.7 se puede observar los componentes del plugin para convertir a red de Petri tales como el Complemento Importar VTreeToPetriImportPlugin el cual se comunica con las bibliotecas del ProM por la interfaz UImportPlugin, el Complemento exportarVTreeToPetriExportPlugin, emplea las interfaces UIExportPlugin, UITopiaVariant y PluginVariant, el complemento VTreeToPetriPlugin utiliza UITopia, PluginVariant, Progress, UIContext y Petrinets, por último el componente VTreeToPetriManager emplea la interfaz Petrinets, además todas interactúan entre sí por medio de la interfaz Plugin (Chaviano 2014).

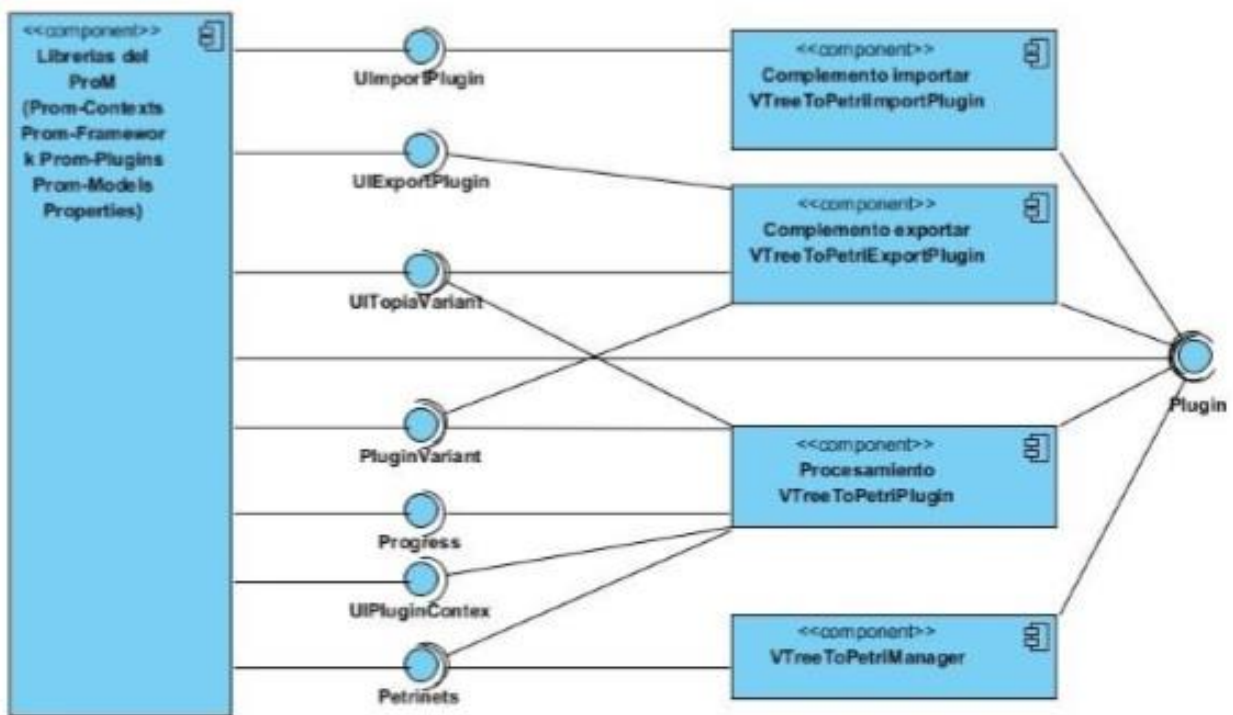


Figura 2.7: Componentes del Plugin ConvertVTreeToPetriNet tomado de (Chaviano 2014).

Luego de generada la red de Petri ya se tiene todos los elementos necesarios para calcular las métricas, para ello se emplea la biblioteca CoBeFra.

2.4 Marco integral CoBeFra

CoBeFra se encuentra integrado actualmente al ProM 6 como uno de sus complementos. Es un marco que permite por su diseño ejecutarse también como una herramienta independiente. Actualmente el usuario al ejecutarla por independiente tiene la posibilidad de interactuar mediante una interfaz.

Para el desarrollo del nuevo componente se le realizaron varias modificaciones a la herramienta CoBeFra: primero se fijaron las métricas a calcular para cualquier modelo; segundo, las actividades que no se pueda obtener información sobre ellas se declararon invisibles automáticamente; tercero, los resultados se almacenan en un archivo llamado “result.txt” y por último se convirtió la aplicación en una biblioteca para su integración con el componente **Variant Miner**.

A continuación se observa en la Figura 2.8 la arquitectura del CoBeFra. El paquete **Metrics Configuration UI** es el encargado de gestionar las métricas definidas exactamente para que sean calculadas en la evaluación de un modelo, el paquete **Input Handling** es el encargado de cargar el registro de eventos y el modelo en notación de Petri para evaluarlo, y por último el paquete **Result** es el encargado de calcular las métricas e ir guardando los resultados en un fichero denominado “result.txt”. Se eliminaron las interfaces, ya que el usuario no interactúa directamente con la aplicación, sino que se convirtió a una biblioteca para que se ejecute internamente al evaluar cada modelo.

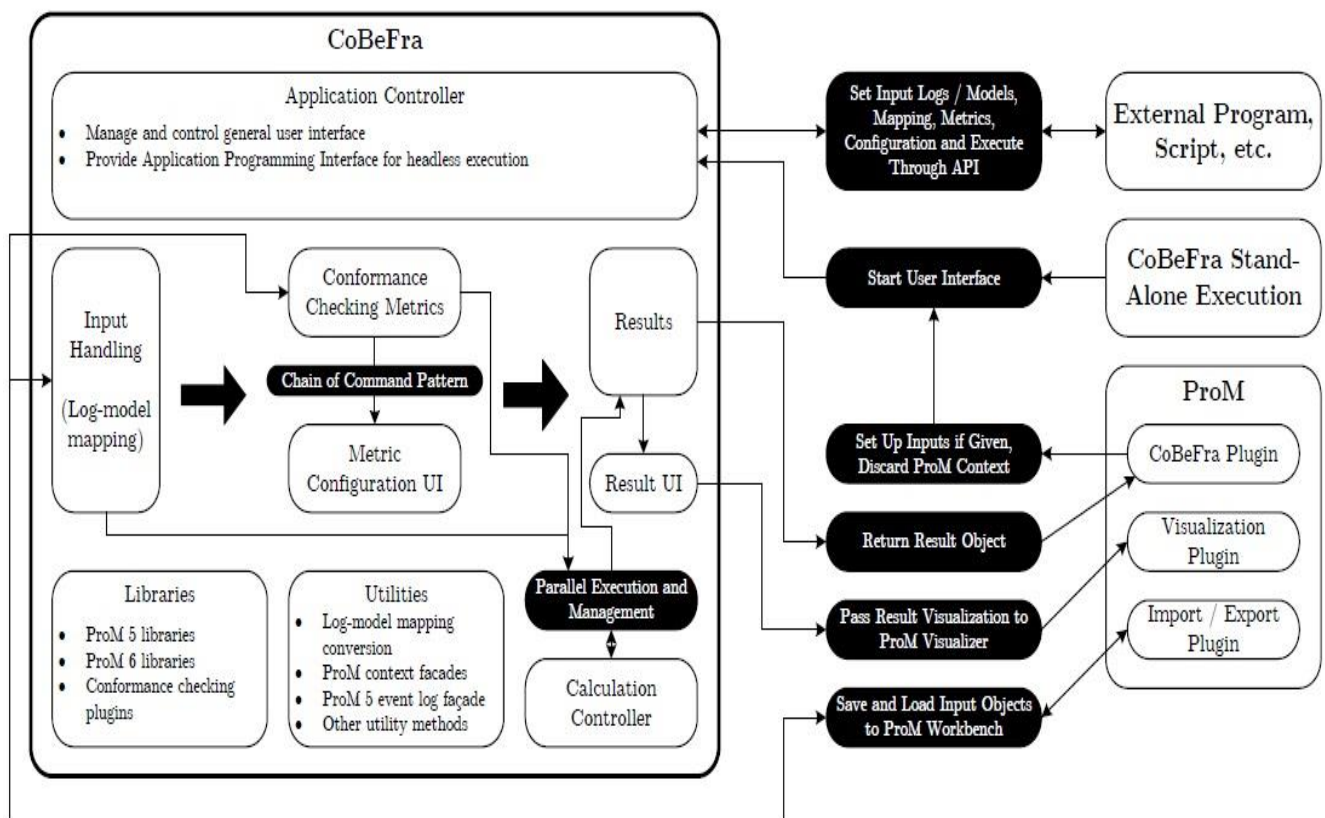


Figura 2.8: Arquitectura de la biblioteca CoBeFra tomado de (Brouke 2014).

De forma general, la biblioteca CoBeFra para evaluar los modelos mediante las métricas definidas, necesita como entrada el nuevo registro de eventos y el modelo en red de Petri generado

anteriormente. Luego, internamente mapea el registro de eventos y la red de Petri declarando las actividades invisibles y ahí calcula las métricas, enviándole los resultados al Perfil de Diagnóstico para ser mostrados al usuario (Figura 2.9).

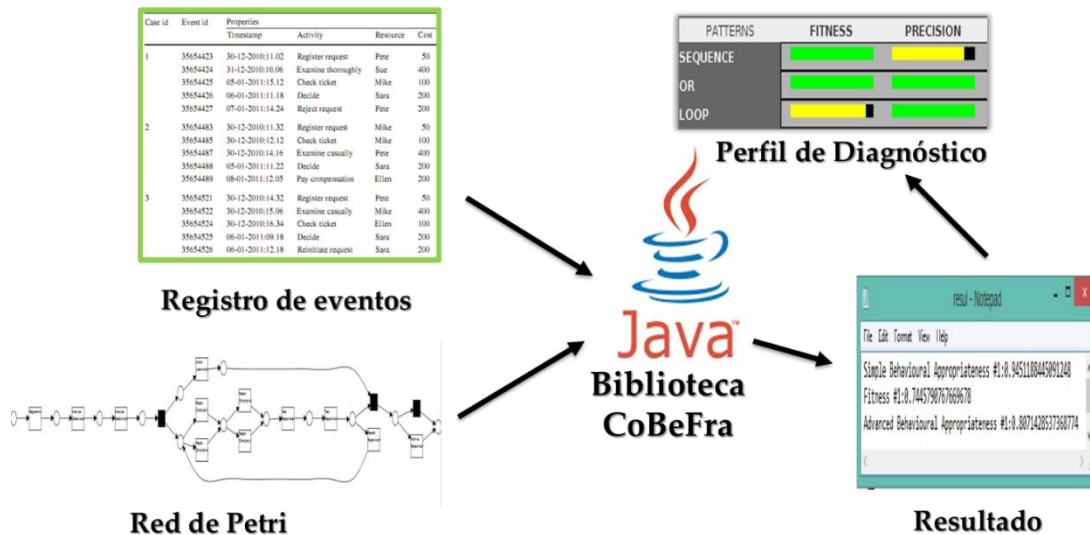


Figura 2.9: Funcionamiento de la biblioteca CoBeFra.

Conclusiones parciales del capítulo

Siguiendo la metodología AUPuci se generaron los artefactos necesarios para guiar y describir el desarrollo de la investigación, como el diagrama de paquete que representa la estructura del componente *Variant Miner* y el diagrama de componentes para representar los elementos que conforman el plugin *ConvertVTreeToPetriNet*.

El diseño del componente para evaluar la calidad de las variantes de modelos de procesos permitió modelar la solución, integrada por 7 pasos fundamentales, y cuya complejidad algorítmica es $O(n^3)$. Este diseño incluye la reducción de los registros de eventos y la transformación de los Árboles de Variantes en redes de Petri, pasos necesarios para realizar el cálculo de las métricas definidas.

La reutilización de los componentes *VariantTree* y *ConvertVTreeToPetriNet*, permitieron la creación del Árbol de Variantes, el Perfil de Diagnóstico, y la red de Petri. Además, se desarrolló el intercambio de datos entre el componente desarrollado y la biblioteca CoBeFra facilitando el cálculo de las métricas y la visualización de los resultados en el Perfil de Diagnóstico.

El componente desarrollado está compuesto por tres elementos fundamentales como son los plugins *VariantTree*, que es el encargado de generar el Árbol de Variantes y el Perfil de Diagnóstico, y

ConvertVTreeToPetriNet, el cual a partir de un Árbol de Variantes genera una red de Petri, y la biblioteca CoBeFra, encargada de calcular las métricas y enviar los resultados al Perfil de Diagnóstico.

CAPITULO 3: Validación de la solución propuesta

En el presente capítulo se realiza una descripción de las pruebas internas aplicadas al componente con el fin de evaluar la calidad y comprobar sus funcionalidades, verificando en todos los casos que los resultados sean los esperados.

Además, se muestran los resultados obtenidos a partir de la aplicación de un sociograma, para contrastar los resultados obtenidos por el componente con la aplicación CoBeFra. Finalmente se realiza una discusión de los resultados obtenidos.

3.1 Pruebas Internas

Las Pruebas Internas tienen como característica principal que son desarrolladas por el equipo de desarrollo, con el objetivo de probar que el componente desarrollado ejecuta todas las acciones correctamente para las que fue concebido y devuelve los resultados esperados. Para ello se evalúa en cuanto al tiempo que demora el componente en generar cada uno de los objetos necesarios y el cálculo de las métricas.

Para el desarrollo de las pruebas se hace necesario utilizar un grupo de registros de eventos con dos características principales como la ausencia de información y la presencia de ruido, en un 20%, recreando el comportamiento crítico que podría encontrarse en un entorno real.

3.1.1 Características de los registros de eventos artificiales

Para el desarrollo de las pruebas se generaron cinco registros de eventos a partir de modelos de procesos, estos fueron generados con la herramienta Process Log Generator, en la versión 1.4 (Burattin and Sperduti, 2011). Esta herramienta permite la construcción de modelos a partir de la combinación aleatoria de características como: cantidad de patrones anidados, probabilidad de Secuencia, Lazo, AND Split/Join y XOR Split/Join (Damián Pérez Alfonso 2014). Los modelos de procesos utilizados poseen las características descritas en la Tabla 3.1.

Tabla 3.1: Características de los modelos de procesos artificiales tomado de (Damián Pérez Alfonso 2014).

Características	Modelos de procesos				
	1	2	3	4	5
Actividades	10	8	15	10	13
Secuencia	7	6	9	6	10
Lazo	0	0	0	1	0
Paralelismo	1	1	1	0	1
Selección exclusiva	1	0	2	1	0

Por cada uno de los modelos de procesos descritos, se generó un registro de eventos, los cuales fueron utilizados para conformar el grupo G1. Para obtener los 5 registros de eventos que conforman G2 se extrajo, a cada registro de eventos perteneciente a G1, el 20% de sus eventos (Damián Pérez Alfonso 2014). En la Tabla 3.2 se muestran las características de los registros de eventos de cada grupo.

Tabla 3.2: Características de los registros de eventos.

Registro de eventos	Trazas	Grupo G1		Grupo G2	
		Eventos	Trazas únicas	Eventos	Trazas únicas
1	1500	10492	3	8394	62
2	1500	12000	2	9600	59
3	1500	16500	12	13200	231
4	1500	23096	90	18477	425
5	5000	35192	9	28154	182

A continuación se presentan los resultados obtenidos con el empleo de los registros de eventos pertenecientes a G2.

3.1.2 Resultados obtenidos aplicando G2

Los resultados obtenidos se basan en el tiempo que demora el componente en realizar el proceso de generación, de cada uno de los objetos necesarios para calcular las métricas y el valor obtenido para cada una de la dimensiones de calidad, que le son mostrados al usuario. A continuación se observan en una tabla los tiempos consumidos por cada uno de los registros de eventos, en milisegundos, y los resultados en un rango de 0-1.

Reducción de actividades

La reducción de las actividades se hace mediante la ejecución del algoritmo de codificación, recorriendo cada una de las trazas correspondientes a registro de eventos, obteniendo los siguientes resultados:

Tabla 3.3: Resultado de la reducción de actividades.

Registro de eventos	Tiempo en milisegundos (ms)
1	2313
2	1888
3	1394
4	1185
5	1854

Luego de reducir las actividades se necesita crear el nuevo registro de eventos, pero tomando solo las actividades que quedaron registradas.

Creación del nuevo registro de evento

Para crear el nuevo registro de eventos, basta con solo guardar las actividades que no fueron reducidas en un archivo y luego exportarlo como un registro de eventos. A continuación se observa en la tabla 3.4 el tiempo que demora la aplicación en realizar esta operación.

Tabla 3.4: Tiempo para crear un registro de eventos.

Registro de eventos	Tiempo en milisegundos (ms)
1	129
2	148
3	30
4	118
5	24

Después de creado el registro de eventos, el próximo paso es generar la red de Petri a partir del Árbol de Variantes modificado.

Creación de la red de Petri

Para crear una red de Petri solo se necesita como entrada un Árbol de Variantes y de acuerdo a cada patrón de control de flujo presente en el nodo raíz, se construye la red. De acuerdo al tamaño del Árbol y a la cantidad de patrones, así será el tiempo que consumirá el componente para realizar la operación. A continuación se muestran los resultados en cuanto a tiempo se refiere, de la construcción de la red de Petri para el cada uno de los Árboles de forma general.

Tabla 3.5: Tiempo para crear una red de Petri.

Registro de eventos	Tiempo en milisegundos (ms)
1	47
2	314
3	22
4	35

5	57
---	----

Después de creados todos los elementos necesarios se procede a calcular las métricas, una por cada dimensión de calidad. En la siguiente tabla se muestra el tiempo que demoró el componente en obtener los resultados.

Tabla 3.6: Tiempo consumido para el cálculo de las métricas.

Registro de eventos	Tiempo en milisegundos (ms)
1	15848
2	21905
3	17247
4	16645
5	26202

Analizando los resultados obtenidos por cada una de las operaciones que realiza el componente, se puede apreciar en la tabla 3.7, que el componente se tarda aproximadamente 2 segundos para reducir las actividades, un segundo para crear los registros de eventos nuevos, 0,1 segundo en crear la red de Petri y 20 segundos para calcular las métricas:

Tabla 3.7: Tiempo promedio consumido para realizar cada operación.

Componente	Tiempo promedio (ms)			
	Reducción de actividades	Creación del registro de evento	Creación de la red de Petri	Cálculo de las métricas
Variant Miner	1726	89	95	19567

Resultados obtenidos por cada Dimensión

Después de generados cada uno de los elementos necesarios para realizar el cálculo de las métricas, se compararon los resultados empleando los mismos datos, pero en este caso, con los obtenidos por la Herramienta CoBeFra, que se recogen en la Tabla 3.8 a continuación.

Tabla 3.8: Comparación de los valores calculados mediante el componente y CoBeFra.

Componente <i>Variant Miner</i>				Herramienta CoBeFra			
RE	Aptitud	Precisión	Generalización	RE	Aptitud	Precisión	Generalización
1	0,66	0,96	0,04	1	0,66	0,96	0,04
2	0,69	0,978	0,022	2	0,69	0,978	0,022
3	0,62	0,83	0,17	3	0,62	0,83	0,17
4	0,65	0,61	0,39	4	0,65	0,61	0,39
5	0,61	0,36	0,64	5	0,61	0,36	0,64

Para determinar correlación existente entre los resultados obtenidos por *Variant Miner* y CoBeFra, se utilizó el coeficiente de correlación de Pearson.

3.2 Coeficiente de correlación de Pearson

El coeficiente determina el grado de relación entre dos variables cuantitativas y se define como:

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

donde $\rho_{X,Y}$ es la covarianza de (X; Y), σ_X es la desviación típica de la variable X y σ_Y es la desviación típica de la variable Y. La interpretación del valor de $\rho_{X,Y}$ es la siguiente (Kenney 1962):

- Si $\rho_{X,Y} = 1$, existe una correlación positiva perfecta. El índice indica una dependencia total entre las dos variables, denominada relación directa.
- Si $0 < \rho_{X,Y} < 1$, existe una correlación positiva.

- Si $\rho_{X,Y} = 0$, no existe relación lineal entre las dos variables, es decir, la correlación es nula.
- Si $-1 < \rho_{X,Y} < 0$, existe una correlación negativa.
- Si $\rho_{X,Y} = -1$, existe una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables, denominada relación inversa.

3.2.1 Análisis para la dimensión Aptitud

A continuación se muestra la Tabla 3.9 los resultados obtenidos por *Variant Miner* y *CoBeFra* en el cálculo de la métrica para la dimensión Aptitud.

Tabla 3.9: Resultados obtenidos para la Aptitud.

RE	<i>Variant Miner</i>	RE	CoBeFra
1	0,66	1	0,66
2	0,69	2	0,69
3	0,62	3	0,62
4	0,65	4	0,65
5	0,61	5	0,61

A partir de los resultados mostrados en la tabla anterior, se aplica el coeficiente de correlación de Pearson, obteniéndose el resultado $\rho_{X,Y} = 1$, lo cual refleja que existe una correlación positiva perfecta.

3.2.2 Análisis para la dimensión Precisión

A continuación se muestra la Tabla 3.10 los resultados obtenidos por *Variant Miner* y *CoBeFra* en el cálculo de la métrica para la dimensión Precisión.

Tabla 3.10: Resultados obtenidos para la Precisión.

RE	<i>Variant Miner</i>	RE	CoBe Fra
1	0,96	1	0,96
2	0,978	2	0,978
3	0,83	3	0,83
4	0,61	4	0,61
5	0,36	5	0,36

A partir de los resultados mostrados en la tabla anterior, se aplica el coeficiente de correlación de Pearson, obteniéndose el resultado $\rho_{X,Y} = 1$, lo cual refleja que existe una correlación positiva perfecta.

3.2.3 Análisis para la dimensión Generalización

A continuación se muestra la Tabla 3.11 los resultados obtenidos por *Variant Miner* y *CoBeFra* en el cálculo de la métrica para la dimensión Generalización.

Tabla 3.11: Resultados obtenidos para la Generalización.

RE	<i>Variant Miner</i>	RE	CoBe Fra
1	0,04	1	0,04
2	0,022	2	0,022
3	0,17	3	0,17

4	0,39	4	0,39
5	0,64	5	0,64

A partir de los resultados mostrados en la tabla anterior, se aplica el coeficiente de correlación de Pearson, obteniéndose el resultado $\rho_{X,Y} = 1$, lo cual refleja que existe una correlación positiva perfecta.

Luego de analizados los 3 resultados, calculados aplicados Pearson, se aprecia que entre todos los valores existe una correlación positiva perfecta.

Conclusiones parciales del capítulo

Se definió como prueba para la validación del componente las Pruebas Internas. Para el desarrollo de las pruebas, se generaron cinco registros de eventos de manera artificial y se le extrajo el 20% de la información con el objetivo de introducir ruido y gran ausencia de información. Los resultados obtenidos por el componente desarrollado coinciden con los mostrados por la herramienta CoBeFra, con un tiempo promedio de 19 segundos.

Para el análisis de los resultados se aplicó el coeficiente de correlación de Pearson, demostrando que existe una correlación positiva perfecta entre los valores obtenidos por el componente desarrollado y la herramienta CoBeFra, lo cual permite afirmar que el resultado obtenido por el componente desarrollado es correcto.

Conclusiones generales

1. El análisis de la literatura publicada sobre métricas de calidad, llevó a la selección de las métricas *Fitness*, *Improved/Advanced Behavioral Appropriateness* y *Negative Event Generalization* para las dimensiones Aptitud, Precisión y Generalización respectivamente.
2. El marco integral CoBeFra es la técnica de chequeo de conformidad que mejor se ajusta a los intereses de la solución, ya que garantiza la evaluación de los modelos mediante las métricas seleccionadas y tiene menor curva de aprendizaje que el resto de las técnicas.
3. El diseño del componente para evaluar la calidad de las variantes de modelos de procesos permitió modelar la reducción de los registros de eventos y la transformación de los Árboles de Variantes en Redes de Petri, pasos necesarios para realizar el cálculo de las métricas definidas.
4. La reutilización de los componentes *VariantTree*, para crear el Árbol de Variantes y el Perfil de Diagnóstico, *ConvertVTreeToPetriNet*, para generar la Red de Petri, y la biblioteca CoBeFra, facilitó la implementación del componente propuesto en ProM.
5. Los resultados obtenidos por el coeficiente de correlación de Pearson para cada una de las dimensiones evaluadas, demuestran que existe una correlación positiva perfecta entre los valores obtenidos mediante *Variant Miner* y CoBeFra.

Recomendaciones

La biblioteca CoBeFra creada permitió la evaluación de los modelos mediante las métricas, las cuales se basan en el uso de varias bibliotecas para graficar los resultados. Se recomienda:

- Estudiar el funcionamiento de la biblioteca Graphviz con el objetivo de capturar las excepciones que pueda generar a la hora de graficar, o de lo contrario, eliminar el lanzamiento de la excepción para graficar, ya que no es un objetivo del usuario visualizarlo.
- Estudiar y analizar cuáles de las bibliotecas del ProM utiliza realmente la biblioteca CoBeFra con el objetivo de eliminar aquellas que sean innecesarias, para reducir el tiempo de cálculo de las métricas.

REFERENCIAS

- A. ADRIANSYAH, J. M. G., J. CARMONA. Based Precision Checking. 2010.
- AALST. Process Mining Discovery, Conformance and Enhancement of Business Eindhoven University of Technology, 2012.
- AALST, W. M. P. V. D., A. ADRIANSYAH. Process Mining. Discovery, Conformance and Enhancement of Business Processes. Eindhoven University of Technology, 2011.
- ADRIANSYAH, A., B. F. V. DONGEN AND W. M. P. V. D. AALST Conformance Checking Using Cost-Based Fitness Analysis. IEEE Computer Society, 2011.
- ADRIANSYAH, A. A. J. C. A. M. Mining Process Performance from Event Logs 2012.
- BROUKE, S. V. Advances in Process Mining: Artificial Negative Events and Other Techniques. Master, 2014.
- BSC., R. P. J. M. V. A. A Benchmark Set for Process Discovery Algorithms. Master Thesis Eindhoven University of Technology, 2011.
- CHAVIANO, Y. O. Complemento para transformar un Árbol de Variantes en una red de Petri. 2014.
- DAMIÁN PÉREZ ALFONSO, E. P. H., JUAN PEDRO FEBLES RODRÍGUEZ. Engineering Education Facing the Grand Challenges, What Are We Doing? In *Análisis de Variantes de Procesos de Negocio*. Santo Domingo, Dominican Republic: 13 th LACCEI Anual International Conference, 2015a, vol. 1, p. 10.
- DAMIÁN PÉREZ ALFONSO, E. P. H. A. P. F. R. TÉCNICA PARA EL DIAGNÓSTICO DE VARIANTES DE PROCESOS DE NEGOCIO. Master, 2014.
- DAMIÁN PÉREZ ALFONSO, E. P. H. A. P. F. R. Análisis de Variantes de Procesos de Negocio. In *Santo Domingo, Dominican Republic*,. 13 th LACCEI Annual International Conference., 2015b.
- DE WEERDT, J. B. A. B. A. V. A Comprehensive Benchmarking marco de trabajo (CoBeFra) for conformance analysis between procedural process models and event logs in ProM. 2013.
- GAMMA, E., R. HELM, R. JOHNSON AND J. VLISSIDES *Design Patterns Elements of Reusable Object-Oriented Software*. Edtion ed., 2004. ISBN 8131700070.
- HERRERA, R. I. Modelo para la estimación de información ausente en las trazas usadas en la Minería de Procesos. Doctorado, 2012.
- HOSTED ON GITHUB., A. Por qué Git es el sistema de control de versiones más popular. DESARROLLO DE SOFTWARE, OPEN SOURCE, 2014.
- JACOBSON, I., G. BOOCH AND J. RUMBAUGH *El Proceso Unificado de Desarrollo de Software*. Edtion ed. Madrid: Pearson Educación. S.A., 2000.
- KENNEY, J. F. Mathematics of statistics. 1962.

KOURDI. Analisis de Procesos. 2008.

ORACLE. Welcome to NetBeans. In *Welcome to NetBeans*. 2013.

PARADIGM, V. Visual Paradigm Suite User's Guide. In., 2005.

REYNALDO LÓPEZ JIMÉNEZ, E. P. H. Algoritmo para la detección de patrones de control de flujo en los registros de eventos reales., 2014.

ROZINAT, A. Conformance checking of processes based on monitoring real behavior. In W.M.P. VAN DER AALST. *Information System*. 2008, vol. 33, p. 64–95.

SÁNCHEZ, T. R. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana, Cuba. In. Universidad de las Ciencias Informaticas: La Habana, Cuba, 2015, p. 1:16.

SOMMERVILLE, I. *Software Engineering*. Edtion ed.: Pearson Education, 2007. ISBN 7-111-19770-4.