

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

**Personalización del plugin Heuristic Miner del ProM para la detección de
incongruencias en los procesos del Sistema de Información Hospitalaria del CESIM**

AUTORES

Mayrelis Abreu Mazorra

Alejandro Marcial Reyes Gainza

TUTOR

Ing. Arturo Orellana García

COTUTOR

Ing. Eiler Efraín Sarmiento Avila

“Año 57 de la Revolución”

La Habana, junio, 2015.

Declaración de autoría

Declaramos ser autores de la presente investigación y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de junio del año 2015.

Mayrelis Abreu Mazorra

Firma del Autor

Alejandro Marcial Reyes Gainza

Firma del Autor

Arturo Orellana García

Firma del Tutor

Eiler Efrain Sarmiento Avila

Firma del Tutor

Datos de contacto

Síntesis del tutor

Ing. Arturo Orellana García: Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Labora en el Centro de Informática Médica (CESIM). Se desempeña como tutor de las asignaturas Práctica Profesional IV a Práctica Profesional VII y ha sido analista de los proyectos SIAPS, Telemedicina Domiciliaria y Ensayos Clínicos del Centro de Informática Médica. Ha tutorado dos trabajos de diploma. Actualmente es estudiante de maestría.

Correo electrónico: aorellana@uci.cu

Ing. Eiler Efraín Sarmiento Avila: Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2014. Recién Graduado en Adiestramiento. Se desempeña como Arquitecto Principal del Sistema de Gestión de Ensayos Clínicos del Departamento Sistemas Especializados en Salud (SES), del Centro de Informática Médica (CESIM).

Correo electrónico: eiler@uci.cu

Agradecimientos

Agradecer primeramente a mis padres por ser mis guías de toda la vida, por enseñarme que con trabajo y esfuerzo podemos lograr todas nuestras metas, por sus consejos y enormes sacrificios en el transcurso de la carrera. Porque sin ustedes yo nunca hubiese logrado este sueño, por ser lo más preciado que tengo en la vida.

A mi tía Maydalis por estar siempre ahí cuando la he necesitado, por complacerme en todos mis caprichos, gracias por apoyarme en mis decisiones y por ser otra madre para mí.

A mi Mima querida gracias por ser la mejor abuela del mundo, protegerme, escucharme y darme tanto amor.

A mi hermano Magdiel, quien ha contribuido a que me esfuerce cada vez más para ser un buen ejemplo.

A mis primitas Mayi y Male por su cariño infinito.

A mi novio por su amor, cariño y comprensión, por haber sido mi apoyo incondicional durante todo este tiempo, por soportar mis malcriadeces y verse convertido en alguien imprescindible e importante en mi vida.

A la familia de mi novio por sus atenciones desde que me conocieron, en especial a Mayi, Lidia y mi suegro.

A mi compañero de tesis gracias por confiar en mí para el desarrollo de este trabajo y formar parte de este momento especial.

A Ary, Mary y Yele por todos los momentos malos y buenos vividos juntas, por siempre aconsejarme y estar presente cuando las necesito.

A mis compañeros de aula por ser personas tan especiales y compartir tantos momentos juntos, nunca los olvidaré.

A Arturo, por siempre estar ahí dispuesto a apoyarnos y brindarnos su ayuda sin importar las reiteradas veces que lo necesitamos.

Agradecimientos

Al tribunal y a Viquillon por sus consejos y ayuda en la realización de este trabajo.

A todos los que de una forma u otra han formado parte de mi vida en esta universidad y han hecho que este lugar sea inolvidable para mí. Gracias

Mayrelis

A mis padres Alejandro y Susana por su apoyo incondicional en todo momento, por su dedicación, amor, amistad, por sus valiosos consejos que me sirvieron de tanto en la vida, y por hacer que me sienta orgulloso al decir que soy su hijo.

A mi abuela Marta, por preocuparse por mí, por sus noches desvelada rezando para que yo saliera bien en las pruebas, por sus innovaciones culinarias, y por sobre todo por ser mi abuela.

A mi novia Lisney por su amor, comprensión, complicidad, por hacerme sentir especial, por enseñarme como es amar y ser amado, por completar mi ser y representar un punto de inflexión en mi vida.

A mis suegros y a toda la familia de mi novia por acogerme como un miembro más.

A mi extenso pelotón de tíos y tías por ser tan especiales conmigo.

A mis primos, en especial a Eduardito, Rubén y Roger por ser como hermanos.

A Soraida, Armandito, Luis, Omar, Ramiro y sus hijos por ser mi familia aquí en La Habana.

A todos mis compañeros de aula con los que he compartido experiencias inolvidables a lo largo de estos 5 años.

A todos mis amigos conocidos aquí en la UCI que siempre los recordaré.

A mis antiguos compañeros de apartamento Reinier, el Gosti, Ortega, el Yoyo y todos los demás quienes fueron las primeras personas que conocí aquí en la UCI.

A mi compañera de tesis por su dedicación y entrega para poder lograr juntos esta difícil tarea.

Agradecimientos

A mis tutores Arturo y Eiler por su apoyo sin el cual no hubiera sido posible este resultado.

Al tribunal y oponente por brindarnos su ayuda y buen juicio permitiendo realizar el trabajo satisfactoriamente.

Y a todos los profesores que han ayudado en mi formación como Ingeniero en Ciencias Informáticas.

Alejandro

Dedicatoria

A toda mi familia que siempre me han apoyado y me han dado fuerzas para llegar hasta aquí y hacer realidad este sueño. Los amo y siempre los llevo en mi corazón.

Mayrelis

A mis padres Alejandro y Susana.

A mi abuela Marta.

A mi novia Lisney.

A toda mi familia y amigos.

Y a todas las personas que de una forma u otra me han ayudado a lo largo de mi carrera.

Alejandro

Resumen

La mayoría de las organizaciones utilizan sistemas de información para gestionar y optimizar sus procesos. Estos sistemas almacenan la información de los eventos asociados a las actividades de los procesos de negocio, denominados registro de eventos. El descubrimiento de modelos a partir de la información contenida en dicho registro es lo que se denomina Minería de Procesos.

Para el descubrimiento de modelos de procesos se han desarrollado diferentes técnicas entre las que se encuentra la *Heuristic Miner*, la cual es capaz de tratar con el ruido y la baja frecuencia, además es poco sensible al carácter incompleto de los registros, y puede utilizarse para expresar el comportamiento principal almacenado en un registro de eventos.

La investigación se centró en el desarrollo de una personalización de la técnica *Heuristic Miner* del ProM para detectar incongruencias en los procesos del Sistema de Información Hospitalaria del Centro de Informática Médica (CESIM), debido a que en estudios realizados anteriormente a partir de sus registros de eventos, se demostró que existen incongruencias entre la definición formal de los procesos y lo que ocurre realmente en el sistema.

Para el desarrollo del componente se emplea como servidor de aplicación JBoss 4.2.2, Eclipse 3.4.2 como plataforma de desarrollo, Java 1.6 como lenguaje de programación, Hibernate 3.3 como ORM (*Object Relational Mapping*) para la persistencia de los datos y Seam 2.1.1 como el marco de trabajo integrador.

Como resultado se obtuvo un componente de software, que permite identificar incongruencias en los procesos clínicos administrativos, generando modelos precisos y comprensibles para usuarios no expertos en Minería de Procesos y de este modo se eliminan anomalías en la ejecución de las actividades de proceso, que atentan contra el correcto flujo de eventos del sistema.

Palabras clave: *Heuristic Miner, Minería de Procesos, modelos, registro de eventos, sistemas de información.*

Índice

Introducción.....	1
Capítulo I: Fundamentación Teórica.....	6
1.1 Gestión basada en procesos.....	6
1.2 Modelo de proceso.....	7
1.3 Minería de Procesos	7
1.3.1 Perspectivas de la Minería de Procesos	10
1.4 Registro de eventos	10
1.5 Sistemas que utilizan Minería de Procesos para la detección de incongruencias en sus procesos.....	12
1.5.1 ARIS Process Performance Manager (ARIS PPM)	12
1.5.2 QPR Process Analyzer.....	13
1.5.3 Discovery Analyst.....	14
1.5.4 ProM	14
1.5.5 Disco.....	15
1.6 Metodología, lenguaje, tecnologías y herramientas a utilizar	15
1.6.1 Guía de desarrollo de software	15
1.6.2 Metodología de desarrollo de software.....	16
1.6.3 Ambiente de desarrollo	16
1.6.4 Lenguaje de programación.....	16
1.6.5 Lenguaje de Modelado.....	17
1.6.6 Tecnologías	17
Java Server Faces (JSF) 1.2.....	17
Java Platform Enterprise Edition (JavaEE) 5.0	17
Java Persistence API (JPA)	17
Hibernate 3.3.....	17
SEAM 2.1.1.....	18
Java Runtime Environment (JRE)	18
Jboss Server 4.2.2	18
Facelets 1.1	18
RichFaces 3.3.1	19
Enterprise JavaBeans (EJB) 3.0.....	19
1.6.7 Herramientas	19

Capítulo II: Propuesta de solución.....	22
2.1 Patrones arquitectónicos.....	22
2.1.1 Modelo Vista Controlador.....	22
2.2 Patrones de diseño	23
2.3 Particularidades de <i>plugin Heuristic Miner</i> del ProM	28
2.3.1 Algoritmo <i>Heuristic Miner</i>	28
2.4 Flujo de información de la personalización del <i>plugin Heuristic Miner</i>	30
2.5 Parámetros del <i>plugin Heuristic Miner</i>	31
2.6 Integración del <i>plugin Heuristic Miner</i> al HIS	33
2.6.1 Etapa 1: Generación del registro de eventos.....	33
2.6.2 Etapa 2: Análisis de las configuraciones para el HIS.....	34
2.6.3 Etapa 3: Visualización de los modelos	36
2.7 Estructura del componente	36
Capítulo III: Validación de la propuesta de solución	43
3.1 Sistema de Información Hospitalaria del CESIM	43
3.2 Diseño de la validación	43
3.3 Detección de incongruencias en los procesos del HIS	43
3.4 Conclusión de los procesos analizados.....	46
3.5 Análisis de las configuraciones del componente <i>Heuristic Miner</i>	47
Conclusiones.....	51
Recomendaciones.....	52
Referencias Bibliográficas	53
Bibliografía	57
Anexos	62

Introducción

La mayoría de las empresas utilizan sistemas de información, con el objetivo de gestionar y optimizar la ejecución de sus actividades. La correcta gestión de sus procesos de negocio permite lograr un mejor entendimiento del funcionamiento de la institución. (1) De ahí, que el enfoque basado en procesos, constituye un instrumento básico, pues cada vez más, el éxito de toda organización depende de que sus procesos estén alineados con su estrategia, misión y objetivos. (2)

No obstante, el modelado de estos procesos no es algo trivial, debido a que es necesario un amplio conocimiento de los mismos para modelarlos con la suficiente exactitud. En el modelado de los procesos existe la presencia de incongruencias, traducándose en la falta de relación o correspondencia entre la definición formal del proceso y lo que ocurre realmente. (3)

En la actualidad existen técnicas que permiten la construcción automática de modelos de procesos basados en registros de trazas, en las cuales se almacena la información de los eventos asociados a las actividades de los procesos de negocio. (1) El descubrimiento de estos modelos a partir de la información contenida en dichas trazas es lo que se denomina Minería de Procesos.

La Minería de Procesos es una disciplina de investigación relativamente joven que se ubica entre la inteligencia computacional y la minería de datos, por una parte; y la modelación y análisis de procesos, por otra. (4) La idea de la Minería de Procesos es descubrir, monitorear y mejorar los procesos reales a través de la extracción de conocimiento de los registros de eventos disponibles en los actuales sistemas de información. (4)

La aplicación de la Minería de Procesos constituye una tecnología potencial en sectores críticos que utilicen sistemas de información, específicamente en la salud, pues en el mismo intervienen una serie de procesos y datos que son vitales convertir en conocimiento relevante y percepciones fiables, para una mejor calidad en la atención a los pacientes. En investigaciones realizadas en el sector hospitalario, se ha demostrado que en los procesos quirúrgicos permite controlar la duración de las intervenciones y el consumo de recursos quirúrgicos. Estos resultados ayudan a la toma de decisiones

relacionadas con la planificación de quirófanos o con la secuenciación de turnos de los cirujanos. (3) También es utilizada para el empleo de recursos en el área de Emergencias, tras descubrir horarios de mayor afluencia a este sector hospitalario. Además, en la obtención de conocimientos sobre casos reales de procesos de ginecología y oncología, reflejando la importancia de uso de la Minería de Procesos. (5) Básicamente, la Minería de Procesos se divide en tres tipos: el descubrimiento, la verificación de conformidad y el mejoramiento. El descubrimiento es el área más explotada en la Minería de Procesos, se basa en tomar un registro de eventos y producir un modelo sin usar ninguna información a-priori. (4) Para muchas organizaciones es sorprendente ver que las técnicas de descubrimiento existentes son realmente capaces de descubrir los procesos reales meramente basado en las muestras de ejecución en los registros de eventos. Entre las técnicas de descubrimiento se destaca la *Heuristic Miner*, la cual aplica algoritmos basados en heurística, tomando en cuenta las frecuencias y secuencias de los eventos, para la construcción del modelo de proceso. Estos algoritmos descubren casi todas las estructuras comunes de control (decisión, secuencia, lazos) y son robustos ante ruidos. Se obtiene como resultado modelos precisos y comprensibles.

En Cuba, a partir del proceso de informatización de la sociedad con el objetivo de elevar la calidad y rapidez de los servicios brindados uno de los sectores beneficiados es la salud, el cual en mucho tiempo no ha sido favorecido en la utilización de técnicas y funciones gerenciales como la planificación y el control. Esto conlleva a la inadecuada utilización de los recursos, la incidencia de malas prácticas clínicas, procesos ineficientes y la incorrecta organización de los servicios. (6) Las dificultades que presenta actualmente el sistema hospitalario afecta en gran medida el desempeño de sus profesionales y la satisfacción de los pacientes. En dicho sector es de vital importancia un modelado preciso en sus procesos, debido a que es el primer paso para el análisis y optimización de los mismos, que permita reducir tiempos de espera de los pacientes sin dejar de garantizar una correcta atención sanitaria.

El Centro de Informática Médica (CESIM), perteneciente a la Universidad de las Ciencias Informáticas (UCI), desarrolla un Sistema de Información Hospitalaria o HIS (*Hospital Information System*, por su nombre en inglés), el cual tiene como objetivo la informatización de los procesos hospitalarios en el nivel de atención secundario, permitiendo la gestión clínica y administrativa de los procesos médicos para elevar la

calidad de atención a los pacientes. Dicho sistema, tiene implementado una bitácora, como vía para recopilar información de la actividad de los usuarios y los procesos que ocurren en el sistema. Sobre el mismo fue desarrollado un componente para la extracción y transformación de las trazas de ejecución en un registro de eventos. Para el análisis de estos registros es necesario hacer uso de herramientas externas (ProM y Disco) las cuales generan modelos de procesos de la ejecución real de los procesos en sistemas automatizados. Estas herramientas presentan inconvenientes para la usabilidad y la comprensión de los modelos generados. En estudios realizados a partir de registros de eventos obtenidos del HIS y la herramienta ProM, se demostró que existen incongruencias entre la definición formal de los procesos y lo que ocurre realmente en el sistema. Entre los resultados obtenidos, se comprobó que existen transiciones entre actividades del proceso que nunca fueron ejecutadas por el sistema y se detectaron secuencias en el sistema que no están definidas en el proceso de negocio. Lo que conlleva a que puedan ocurrir anomalías en la ejecución de las actividades de proceso, atentando contra el correcto flujo de eventos del sistema. Igualmente, con el estudio realizado se comprobó que los modelos generados por la herramienta ProM no se adecuan a la comprensión de usuarios no expertos en el área de la Minería de Procesos.

Por tanto, a partir de la situación anterior se define como **problema a resolver**

¿Cómo detectar incongruencias en los procesos del Sistema de Información Hospitalaria del CESIM?

Enmarcado en el **objeto de estudio**: La gestión basada en procesos hospitalarios. Centrándose en el **campo de acción**: El análisis de procesos hospitalarios a partir de la técnica *Heuristic Miner* de Minería de Procesos.

Para dar solución al problema planteado, se define como **objetivo general**: Desarrollar una personalización del *plugin*¹ *Heuristic Miner* del ProM para el Sistema de información Hospitalaria del CESIM, que permita detectar incongruencias en el comportamiento de los procesos, a partir de su registro de eventos.

¹ plugin: Un plugin es aquella aplicación que en un programa informático, añade una funcionalidad adicional o una nueva característica al software. Por lo tanto, puede nombrarse al plugin como un complemento.

Para dar cumplimiento al objetivo general se definen las siguientes tareas de la investigación:

1. Elaborar el marco teórico metodológico referente al análisis de procesos hospitalarios, la Minería de Procesos y los conceptos asociados con el objeto de estudio.
2. Personalizar la técnica *Heuristic Miner* para la identificación de incongruencias en los comportamientos y patrones frecuentes de los procesos del Sistema de Información Hospitalaria del CESIM.
3. Desarrollar las interfaces del componente, que permitan modelar las instancias de proceso a partir del registro de eventos utilizando la técnica *Heuristic Miner*.
4. Validar la solución propuesta a partir de su capacidad de detectar incongruencias en los procesos del Sistema de Información Hospitalaria del CESIM.

Los **métodos científicos** utilizados para desarrollar la investigación fueron:

Métodos teóricos

- Histórico-lógico: se utilizó para analizar los antecedentes, esencia y evolución del área de estudio, permitiendo crear una línea evolutiva de su desarrollo.
- Análisis-síntesis: se utilizó para el estudio y análisis de las principales características y conceptos relacionados con la Minería de Procesos.
- Deductivo-inductivo: se utilizó para analizar y definir los algoritmos a utilizar a partir de la información que brinde el registro de eventos.
- Modelación: se utilizó para la creación y el análisis de las diferentes propuestas de solución que arrojan las herramientas y seleccionar la que más se adecua a la investigación.

Métodos empíricos

- Observación: Utilizado como medio para adquirir conocimiento sobre el campo de estudio a través de la interacción directa con la herramienta.

El documento está estructurado en tres capítulos:

Capítulo 1. Fundamentos teóricos de la investigación: Aborda los conceptos fundamentales relacionados con la investigación, un análisis de los sistemas que utilizan Minería de Procesos para la detección de incongruencias en sus procesos. Además, se describen las herramientas y tecnologías a utilizar.

Capítulo 2. Propuesta de solución: Describe el diseño de la propuesta de solución, el patrón arquitectónico y los patrones de diseño a utilizar. Se abordan los temas relacionados con el principio de funcionamiento y análisis del algoritmo *Heuristic Miner*. Además, la selección de los elementos estructurales y las interfaces del componente, junto con su comportamiento.

Capítulo 3. Validación de la propuesta de solución: Se valida la solución propuesta a partir de la capacidad de detectar incongruencias en los procesos del Sistema de Información Hospitalaria. Se muestran los resultados obtenidos a partir de analizar algunos procesos del HIS.

Capítulo I: Fundamentación Teórica

El presente capítulo tiene como objetivo el análisis y comprensión de los conceptos relacionados con la investigación. Se refleja el estado del arte y sus características fundamentales, basados en un estudio de los sistemas que utilizan Minería de Procesos para la detección de incongruencias en sus procesos. Se describen las herramientas y tecnologías propuestas a utilizar en el desarrollo del componente.

1.1 Gestión basada en procesos

Según la norma ISO 9000:2005 un **proceso** se define como; “un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados”. Para que las organizaciones operen de manera eficaz, tienen que identificar y gestionar numerosos procesos interrelacionados y que interactúan.

La identificación y gestión sistemática de los procesos empleados en la organización y en particular las interacciones entre tales procesos, se conocen como enfoque basado en procesos. La gestión por procesos está dirigida a realizar procesos competitivos y capaces de reaccionar autónomamente a los cambios mediante el control constante de la capacidad de cada proceso, la mejora continua, la flexibilidad estructural y la orientación de las actividades hacia la plena satisfacción del cliente y de sus necesidades. (7)

El ciclo de vida de la Gestión de Procesos de Negocio abarca las siete fases de un proceso de negocio y los sistemas de información que lo soporta (**Figura 1**).

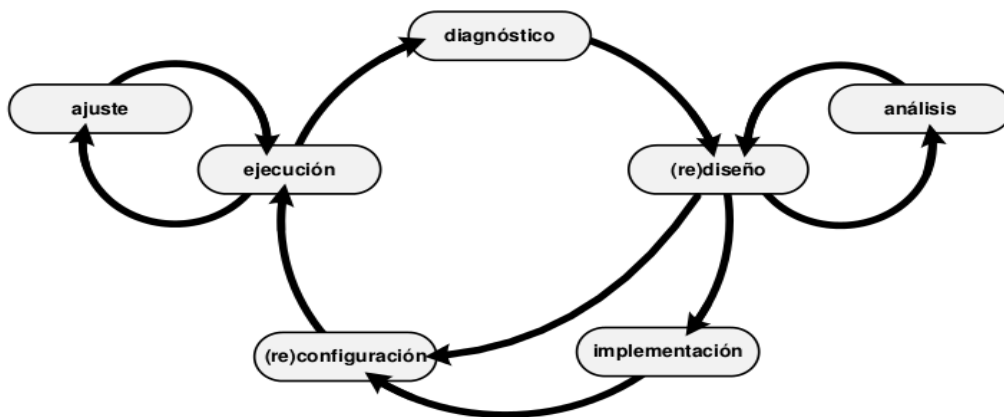


Figura 1: Ciclo de vida de BPM

Fuente: (4)

Primeramente se diseña un proceso, el cual es convertido en un sistema ejecutable en la fase de implementación. En la fase de (re) diseño se crea un nuevo modelo de proceso o se adapta un modelo de proceso existente. En la fase de análisis se analiza un modelo candidato y sus alternativas. Después de la fase de (re) diseño, se implementa el modelo (fase de implementación) o se (re) configura un sistema existente (fase de (re) configuración). En la fase de ejecución se ejecuta el modelo diseñado. Durante la fase de ejecución el proceso es monitoreado. También, se podrán realizar pequeños ajustes sin rediseñar el proceso (fase de ajuste). En la fase de diagnóstico se analiza el proceso ejecutado y la salida de esta fase podría iniciar una nueva fase de rediseño del proceso (4).

1.2 Modelo de proceso

Un modelo es una representación de una realidad compleja. Modelar es desarrollar una descripción lo más exacta posible de un sistema y de las actividades llevadas a cabo en él. Cuando un proceso es modelado, con ayuda de una representación gráfica pueden apreciarse con facilidad las interrelaciones existentes entre distintas actividades, analizar cada actividad, definir los puntos de contacto con otros procesos, así como identificar los subprocesos comprendidos. Al mismo tiempo, los problemas existentes pueden ponerse de manifiesto claramente dando la oportunidad para iniciar acciones de mejoras (8).

1.3 Minería de Procesos

La Minería de Procesos es una disciplina científica cuyo fin es descubrir, monitorear y mejorar los procesos reales, a través de la extracción de conocimiento de los registros de eventos, disponibles en los actuales sistemas de información. Esta disciplina provee un puente importante entre la minería de datos, la modelación y análisis de procesos de negocios (4).

Existen tres tipos principales de Minería de Procesos (**Figura 2**): el descubrimiento de proceso, la verificación de conformidad y el mejoramiento de modelos.

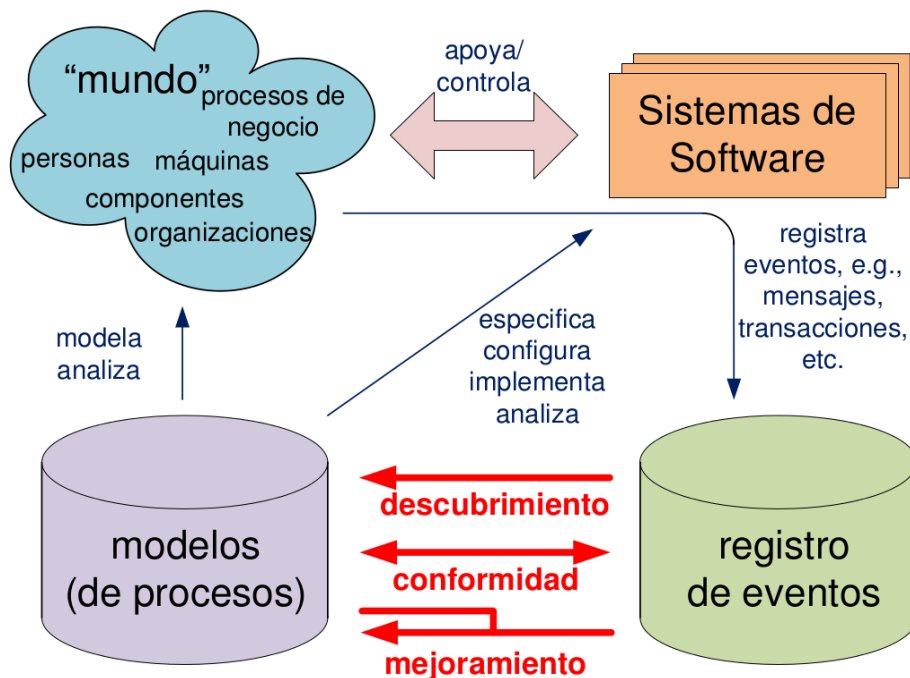


Figura 2: Representación de los tipos de Minería de Procesos

Fuente: (4)

La verificación de conformidad tiene como objetivo comparar un modelo de proceso existente con un registro de eventos del mismo proceso. La verificación de conformidad puede ser usada para revisar si la realidad, tal como está almacenada en el registro de eventos, es equivalente al modelo y viceversa. El mejoramiento de modelos consiste en analizar la información almacenada en los registros de eventos, con el fin de ajustar o extender el modelo para que refleje mejor lo que ocurre durante su ejecución. El descubrimiento es el área más explotada en la Minería de Procesos, se basa en tomar un registro de eventos y producir un modelo sin usar ninguna información a-priori. Para muchas organizaciones es sorprendente ver que las técnicas existentes son realmente capaces de descubrir los procesos reales meramente basado en las muestras de ejecución en los registros de eventos.

Existe una serie de algoritmos de descubrimiento para el mapeo de un registro de eventos hacia un modelo de proceso (4), tales como:

- **Alpha Miner:** Es uno de los primeros algoritmos de descubrimiento de procesos. Se basa en la captura, desde un registro de evento, de las relaciones siguientes usando el sistema binario. Asume que el registro está completo, por lo que presenta problemas para lidiar con el ruido y el comportamiento infrecuente o incompleto. No tiene en cuenta las frecuencias de las actividades (4).

- **Genetic Miner:** Este algoritmo simula la evolución en la naturaleza. Parte de una población inicial de individuos (en este caso procesos que son candidatos a la solución) a la que se le realizan varias iteraciones del algoritmo genético acercándose cada vez más a la solución apropiada (9).
- **Fuzzy Miner:** Con la aplicación de esta técnica se obtiene un modelo basado en gráficos, capaz de proporcionar una vista de alto nivel de un proceso, con la abstracción de los detalles no deseados. Asimismo, permite agrupar tareas, aunque considera que cada tarea pertenece a un único nodo; y es empleada en el diagnóstico del registro de eventos, donde posibilita realizar análisis preliminares al descubrimiento, basados en la correlación entre las tareas y la importancia de una secuencia (10).
- **Heuristic Miner:** Este algoritmo, creado en 2006 por Weijters y otros, tiene en cuenta las frecuencias de los eventos. Este dato es usado para derivar las relaciones causales y de paralelismo. La *Heuristic Miner*, es capaz de tratar con el ruido y la baja frecuencia, además es poco sensible al carácter incompleto de los registros, y puede utilizarse para expresar el comportamiento principal (es decir, no todos los detalles y excepciones) almacenado en un registro de eventos (11).

Los modelos descubiertos pueden ser representados utilizando múltiples notaciones, dentro de las que se encuentran Redes de Petri y Redes de Flujo de Trabajo (4).

De los algoritmos de descubrimiento anteriormente mencionados el *Heuristic Miner*, es el idóneo para la detección de incongruencias, pues es considerado el algoritmo más apropiado y robusto para entornos reales en términos de precisión y compresibilidad de los modelos que se obtiene (12). Por otra parte, este algoritmo toma en cuenta las frecuencias y secuencias de eventos al construir un modelo de proceso. La idea básica es que las rutas poco frecuentes no deben ser incorporadas en el modelo, lo cual posibilita dirigir las técnicas de mejora hacia los elementos más críticos del proceso. Tanto la representación parcial proporcionada por las redes causales y el uso de frecuencias hace que el enfoque sea mucho más robusto que otras técnicas desarrolladas. Esta técnica es capaz de modelar la presencia de:

- **Tareas ocultas:** representan ejecuciones normales de un proceso que por sus condiciones excepcionales ocurren de forma aislada.
- **Alternativas no libres:** constructores de control de flujo donde las alternativas y la concurrencias coinciden, por tanto una tarea A depende indirectamente de

una tarea B; o sea en un punto de división o unión, la elección de un camino u otro puede estar en dependencia de las alternativas escogidas en otra parte de dicho proceso.

- **Lazos:** tarea o conjunto de tareas que se repiten varias veces dentro de un proceso.

Si el modelo se obtiene a partir de la notación BPMN (Business Process Management Notation) permite realizar análisis comparando los modelos resultantes de la ejecución del sistema con los modelos definidos para el proceso de negocio. De ahí que la presente investigación está orientada a la personalización de este algoritmo para el Sistema de Información Hospitalaria del CESIM, apoyando al personal clínico administrativo con una herramienta que evite la ocurrencia de cuellos de botellas, desviaciones de recursos y anomalías en la ejecución de las actividades de proceso, que atenta contra el correcto flujo de eventos del sistema.

1.3.1 Perspectivas de la Minería de Procesos

La Minería de Procesos pretende cubrir las siguientes perspectivas:

- **Perspectiva de control de flujo:** Se basa en el control del orden de ejecución de las actividades. El objetivo es encontrar una buena caracterización de todos los caminos posibles (4).
- **Perspectiva de Casos:** Se enfoca en las propiedades de los casos (caracterización por rutas, actores u otros elementos) (13).
- **Perspectiva organizacional:** El objetivo es estructurar la organización al clasificar a las personas en términos de las funciones y roles, así como mostrar la relación entre los distintos usuarios (construir una red social) (13).
- **Perspectiva de Tiempo:** Se relaciona con la ocurrencia y la frecuencia de los eventos. Posibilita descubrir cuellos de botella, medir niveles de servicio, monitorear la utilización de recursos y predecir el tiempo restante de los casos en ejecución (13).

1.4 Registro de eventos

El registro de eventos es la colección de eventos utilizados como entrada para la Minería de Procesos. Las técnicas de Minería de Procesos asumen que es posible registrar eventos secuencialmente (4), donde cada proceso está compuesto por casos que no son más que instancias del mismo. Un caso consiste en eventos que representan pasos

bien definidos dentro del proceso; los eventos poseen atributos, entre los atributos más usados se encuentran: actividad, recursos, costo y tiempo.

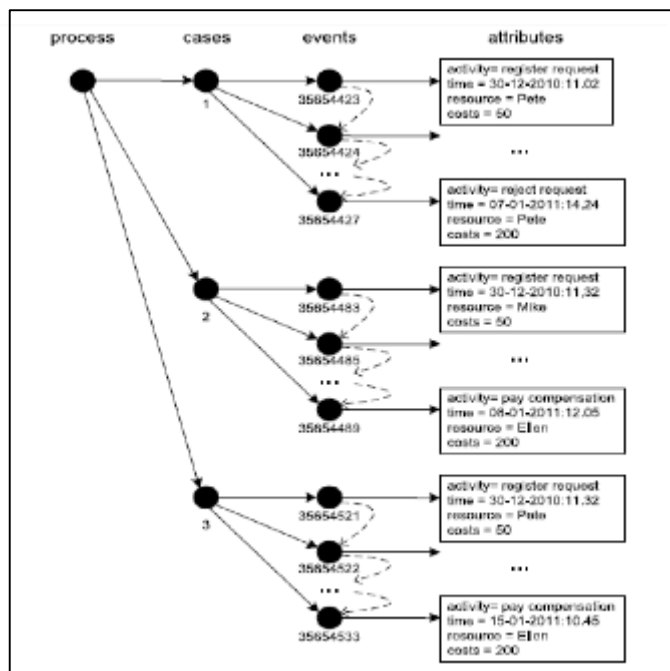


Figura 3: Estructura de un registro de eventos

Fuente: (4)

Para asegurar un análisis de Minería de Procesos exitoso, además del formato de almacenamiento del registro de eventos se debe garantizar su calidad. La misma se define a partir de tres aspectos fundamentales: confiabilidad, completitud y seguridad. La confiabilidad consiste en que los eventos deben ser confiables, es decir, debería ser seguro asumir que los eventos registrados realmente ocurrieron y que los atributos de los eventos son correctos. La completitud se relaciona a que los registros de eventos deberían ser completos, dado un determinado contexto, no puede faltar ningún evento. También, cualquier evento registrado debe tener una semántica bien definida. Por otra parte los datos de eventos son seguros, si se tienen en cuenta consideraciones de privacidad y seguridad al registrar los eventos (4).

Niveles de madurez

La combinación de los tres aspectos de calidad se refleja en los niveles de madurez definidos en el Manifiesto de Minería de Proceso para los registros de eventos:

Nivel 1: Los eventos se registran manualmente, los eventos registrados podrían no corresponder a la realidad y podrían faltar eventos.

Nivel 2: Los eventos se registran automáticamente, no se sigue un enfoque sistemático para decidir qué eventos se registran, podrían faltar eventos o estos podrían no registrarse correctamente.

Nivel 3: Los eventos se registran automáticamente, pero no se sigue un enfoque sistemático para registrar los eventos. Hay un nivel de garantía de que los eventos registrados calzan con la realidad (el registro de eventos es confiable pero no necesariamente completo).

Nivel 4: Los eventos se registran automáticamente y de manera sistemática y confiable, a diferencia del nivel 3 da soporte de manera explícita a nociones tales como instancia de proceso (caso) y actividad.

Nivel 5: El registro de eventos es de excelente calidad (confiable y completa) y los eventos están bien definidos. Los eventos se registran de manera automática, sistemática y segura. Se toman en cuenta adecuadamente consideraciones acerca de la privacidad y la seguridad (4).

Las técnicas de Minería de Procesos pueden ser aplicadas a registros de eventos en nivel 5, nivel 4 y nivel 3. En principio, también es posible aplicar Minería de Procesos utilizando registros de eventos en nivel 2 o nivel 1. Sin embargo, el análisis de dichos registros de eventos es generalmente problemático y los resultados no son confiables. De hecho, no tiene mucho sentido aplicar Minería de Procesos a registros de eventos en el nivel 1. Para obtener beneficio, las organizaciones deben apuntar a registros de eventos en el nivel de calidad más alto posible (4).

1.5 Sistemas que utilizan Minería de Procesos para la detección de incongruencias en sus procesos.

A nivel internacional se han desarrollado varias soluciones informáticas que utilizan Minería de Procesos para la detección de incongruencias en sus procesos empresariales, con el propósito de potenciar la efectividad en la toma de decisiones y mejorar la calidad de sus servicios. A continuación se realiza una breve reseña de algunos de estos sistemas.

1.5.1 ARIS Process Performance Manager (ARIS PPM)

ARIS Process Performance Manager (ARIS PPM) es una herramienta que permite a las empresas supervisar y analizar el rendimiento y la estructura de sus procesos empresariales. ARIS PPM optimiza continuamente los flujos de trabajo internos y

externos, realizando de este modo una contribución crucial para el éxito de la empresa. Distintos departamentos también utilizan el software satisfactoriamente para supervisar acuerdos de nivel de servicio relacionados con procesos básicos, como por ejemplo el tratamiento de pedidos, el aprovisionamiento, el mantenimiento y la actualización, el transporte y la logística, el tratamiento de préstamos y las transacciones de valores en el sector financiero y la gestión de nuevas solicitudes y reclamaciones de seguros, entre otros.

La funcionalidad exclusiva de ARIS PPM permite a los usuarios visualizar automáticamente modelos de procesos reales basándose en datos del sistema de aplicación y, de este modo, realizar comparaciones de los objetivos frente a los valores reales de los flujos de trabajo (14).

Ventajas:

- El rendimiento de los procesos integrales resulta transparente.
- Medición del rendimiento de los procesos (cantidad, tiempo, coste, calidad).
- Supervisión perfecta de los valores de destino y del estado actual.
- Sistema de advertencia temprana eficiente para indicadores de rendimiento críticos.
- Análisis de comunicación y actividades de departamentos y otras unidades organizativas.

1.5.2 QPR Process Analyzer

QPR Process Analyzer es un software de Minería de Procesos escalable utiliza los datos almacenados en los sistemas de apoyo a las empresas, permitiendo reducir el tiempo y el costo en la mejora de los procesos. *QPR Process Analyzer* es fácil de usar y rápido en el procesamiento y análisis de datos. Dirigido a responsables de los procesos, analistas de procesos de negocios, desarrolladores, consultores e incluso administradores de sistema (15).

Ventajas:

- Aumenta los ingresos mediante el descubrimiento y la eliminación de los cuellos de botella del proceso.
- Mejora la eficiencia operativa y reduce los costos a través del rendimiento de los procesos optimizados y llevando a cabo el proceso de seguimiento como una actividad en curso.
- Velocidad para la toma de decisiones con el descubrimiento de procesos

automatizados.

- Rápido tiempo de emitir el descubrimiento y la resolución.

1.5.3 Discovery Analyst

El producto extrae los procesos de negocio de las aplicaciones de negocio en tiempo real para crear modelos de procesos que se pueden utilizar para fines de visualización y de comparación para validar y ampliar los modelos e integrar a la perfección con otras herramientas compatibles con BPMN. *Discovery Analyst* ofrece las siguientes funciones claves (16):

- Grabación: Utiliza su software de reconocimiento semántico para capturar las imágenes de la pantalla que son utilizados por una organización y asociar estos con las etapas del proceso y las decisiones no intrusiva, en tiempo real, a medida que se llevan a cabo por los empleados. Graba lo que un usuario ha presionado para llegar a una pantalla en particular, lo que ayuda a entender la lógica y mapa ramificación compleja.
- Visualización: Los usuarios pueden simular las actividades de procesos de negocio.
- Negocio de Generación de proceso: se genera automáticamente mapas de procesos de negocio en formato estándar BPMN y apoya la exportación modelo directo en Microsoft Word, iGrafx e IBM WebSphere Business Modeler para su posterior mejora y transformación.
- Medición: Captura tiempo real dedicado por los empleados que realizan el proceso de negocio, incluyendo el tiempo para cada actividad y el tiempo total de proceso. Estas métricas de desempeño del proceso para cada actividad del proceso y todo el proceso se pueden exportar a Microsoft Excel.
- Comparación: Comparación de los procesos descubiertos se puede hacer y la diferencia resaltada.
- Generación de documentación: Estructurado y documentación trazable incluyendo mapas de procesos, etapas de proceso y pantallas de la aplicación asociados se puede generar de forma automática.

1.5.4 ProM

La herramienta ProM de código abierto y distribuida gratuitamente, ha sido el estándar impuesto para la Minería de Procesos durante la última década. Permite el proceso de descubrimiento, la comprobación de la conformidad, análisis de redes sociales, la

minería de organización, la minería de decisión (4). La herramienta requiere experiencia en Minería de Procesos y no está respaldada por una organización comercial. Por lo tanto, tiene las ventajas y desventajas comunes para el software de código abierto (17). *Framework* extensible que soporta una amplia variedad de técnicas de Minería de Procesos en forma de *plugins*, uno de ellos es el *Heuristic Miner*.

1.5.5 Disco

Disco es una aplicación completa de Minería de Procesos desarrollada por *Fluxicon* en 2009, es la herramienta más usada dentro de las privativas, posee una licencia gratuita limitada con fines académicos. Implementada con el objetivo de ser una herramienta profesional para el apoyo a las organizaciones en el control de sus procesos. Ha sido diseñada para hacer la importación de datos por las marcas de tiempo de detección automática, recordando los valores de configuración, y por la carga de los conjuntos de datos con alta velocidad. También es totalmente compatible con las herramientas académicas ProM 5 y 6 (4).

A partir del estudio de los sistemas antes mencionados se puede concluir que *ARIS Process Performance Manager*, *QPR Process Analyzer* y *Discovery Analyst* son sistemas privativos, con fines comerciales diseñados a la medida para los sectores industrial y empresarial. Aparte de que no se corresponde con las políticas de independencia tecnológica que lleva a cabo Cuba, a través del uso de software libre para el desarrollo de aplicaciones. Por otro lado, Disco y ProM son herramientas líderes en el área de Minería de Procesos, siendo Disco de licencia gratuita limitada con fines académicos y ProM de código abierto distribuida gratuitamente, pero no está enfocada a la usabilidad, por lo que es recomendable para personas que tienen conocimientos de las técnicas y algoritmos de Minería de Procesos, ya que se requiere cierta experiencia para seleccionar las herramientas a utilizar y para entender los parámetros a configurar. Además, sería complejo el uso de una herramienta externa al HIS.

1.6 Metodología, lenguaje, tecnologías y herramientas a utilizar

A continuación se describen y analizan la metodología, lenguaje, tecnologías y herramientas a utilizar en el proceso de desarrollo del componente.

1.6.1 Guía de desarrollo de software

Modelo de Integración de Capacidad y Madurez o CMMI (*Capability Maturity Model Integration* por su nombre en inglés) es un modelo de madurez para la mejora de los procesos en el desarrollo de productos y de servicios. Consiste en las mejores prácticas

que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento.

Las denominaciones anteriores de CMMI para la ingeniería de sistemas y la ingeniería del software (CMMI-SE/SW) son reemplazadas por el título “CMMI para desarrollo” (CMMI-DEV), reflejando así realmente la integración completa de estos cuerpos de conocimiento y la aplicación del modelo en el seno de una organización. CMMI para desarrollo propone una solución integrada y completa para las actividades de desarrollo y de mantenimiento aplicadas a los productos y a los servicios (18).

1.6.2 Metodología de desarrollo de software

Ayudado en el (CMMI-DEV v1.3), se utiliza la metodología de desarrollo Proceso Unificado Ágil o AUP (*Agile Unified Process*, por su nombre en inglés), la cual es una versión simplificada del Proceso Unificado de Desarrollo o RUP (*Rational Unified Process*, por su nombre en inglés). Esta describe una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. En la variación de AUP definida para la actividad productiva de la UCI, se decide para el ciclo de vida de los proyectos las fases: Inicio, Ejecución y Cierre, ver [Anexo 1](#). Además, se definen 7 disciplinas, ver [Anexo 2](#) y 11 roles, ver [Anexo 3](#) (19).

1.6.3 Ambiente de desarrollo

Un marco de trabajo (*framework*, por su nombre en inglés) para aplicaciones web es un software o conjunto de librerías, que está diseñado para dar soporte al desarrollo de sitios y en general a la construcción de cualquier aplicación web. Entonces un *framework* trata de facilitar aquellas actividades comunes realizadas durante el desarrollo de la aplicación, como por ejemplo: acceso a la base de datos, uso de plantillas, manejo de sesiones, separación de aspectos de programación; además de promover la reutilización de código (20).

1.6.4 Lenguaje de programación

Java es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principio de los años 90's, recientemente fue adquirida por *Oracle Corporation* (21). Una de las principales características de este lenguaje, es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Además, es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas.

1.6.5 Lenguaje de Modelado

Unified Modeling Language (UML)

Es un lenguaje que se utiliza para especificar, visualizar, construir y documentar los artefactos de sistemas intensivos de software. UML es gratuito, accesible a todos, y conforma la colección de las mejores técnicas de ingeniería que han probado ser un éxito en el modelamiento de sistemas grandes y complejos (22).

1.6.6 Tecnologías

A continuación se describen las tecnologías y herramientas de código abierto, utilizadas para el desarrollo del componente, las cuales cumplen con las políticas de independencia tecnológica definidas en Cuba para la informatización de la sociedad.

Java Server Faces (JSF) 1.2

Framework que define un modelo de componentes de interfaz de usuario y de eventos. Permite manejar el estado de los componentes de interfaz de usuario, manejar sus eventos, la validación y conversión del lado del servidor y centralizar la navegabilidad de las páginas de la aplicación. JSF es el marco estándar que proporciona Java para construir aplicaciones web, sigue al patrón Modelo Vista Controlador proporcionando una manera de validar datos y devolver los resultados al cliente (23).

Java Platform Enterprise Edition (JavaEE) 5.0

Java versión 5 o JEE 5 es una plataforma de programación (parte de la Plataforma Java) para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N niveles distribuida (24).

Java Persistence API (JPA)

JPA, es la API de persistencia desarrollada para la plataforma *JavaEE* e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs) (25).

Hibernate 3.3

Herramienta que realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado

es *Object/Relational Mapping* (ORM) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Hibernate no solo realiza esta transformación sino que proporciona capacidades para la obtención y almacenamiento de datos de la base de datos reduciendo el tiempo de desarrollo (26).

SEAM 2.1.1

SEAM es una potente plataforma de desarrollo de código abierto para construir aplicaciones ricas de Internet en Java. Seam integra tecnologías como *JavaScript* asíncrono y XML (AJAX), *JavaServer Faces* (JSF), *Java Persistence Api* (JPa), *Enterprise Java Beans* (EJB 3.0) y *Business Process Management* (BPM). (27) SEAM elimina la capa artificial que existe entre EJB 3.0 y JSF y provee un consistente sistema de anotaciones para integrar estos dos *frameworks*. Comparada con aplicaciones desarrolladas en otros *frameworks*, las aplicaciones SEAM son conceptualmente simples y requieren significativamente menos código (en Java y en XML) para obtener las mismas funcionalidades.

Java Runtime Environment (JRE)

JRE es el acrónimo de *Java Runtime Environment* (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas. JVM (máquina virtual Java) es una instancia de JRE en tiempo de ejecución. Este interpreta el código Java y está compuesto por las librerías de clases estándar que implementan el API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto (28).

Jboss Server 4.2.2

Jboss Server es el servidor de aplicaciones Java más utilizado actualmente en el mercado. Cientos de profesionales y desarrolladores de código abierto han contribuido a su creación y desarrollo. Es una plataforma certificada que cumple con las especificaciones de J2EE para el despliegue y desarrollo de aplicaciones empresariales Java, aplicaciones web y portales (29).

Facelets 1.1

JavaServer Facelets es un *framework* para plantillas centrado en la tecnología JSF (*JavaServer Faces*), por lo cual se integran de manera muy fácil. *Facelets* llena el vacío

entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF (27).

RichFaces 3.3.1

JBoss RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Provee facilidades de validación y conversión de los datos proporcionados por el usuario, administración avanzada de recursos como imágenes, código *Javascript* y Hojas de Estilo en Cascada (CSS) (30).

Enterprise JavaBeans (EJB) 3.0

Permite realizar la administración automática de transacciones, seguridad, escalabilidad, concurrencia, distribución, acceso a ambientes portables y persistencia de datos. Incorpora el estándar JPA como el principal API de persistencia para aplicaciones EJB3. (31) Su objetivo es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.

JGraph 2.1

Es una librería gráfica de Java gratuita que proporciona objetos y algoritmos gráfico, bajo la licencia BSD. JGraph proporciona funcionalidad para la visualización e interacción con gráficos. Algunos ejemplos de aplicaciones que es posible escribir con ella son un editor de flujo de trabajo, un organigrama, una herramienta de modelado de procesos de negocio, una herramienta UML. Además se puede exportar a formatos como .SVG (Gráficos vectoriales), XML, y otros.

1.6.7 Herramientas

Entorno Integrado de Desarrollo Eclipse 3.4.2

Eclipse Ganymede 3.4.2 es un entorno de desarrollo integrado de código abierto, portable y multiplataforma. Este fue diseñado originalmente por la empresa IBM y actualmente, es desarrollado por la Fundación Eclipse, una organización independiente, sin ánimo de lucro que fomenta una comunidad de código abierto. Se basa en el uso de módulos (*plugins*), lo cual hace posible el trabajo en múltiples lenguajes de programación como son Java, C++, PHP, Perl y se le puedan añadir, además, otras funcionalidades (32).

PostgreSQL 9.3

Es un poderoso Sistema Gestor de Base de Datos de código abierto que ha ganado una fuerte reputación debido a su rentabilidad, ingeniería de datos, exactitud y fácil administración. Soporta distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. y permite la creación de tipos propios. Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit. Posee interfaces de programación para lenguajes como C, C++, Java, Perl y Python así como una documentación excepcional (33).

PgAdmin III 1.10

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Facilita enormemente la administración e incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, entre otras (34).

Visual Paradigm 8.0

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML contribuye a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (35).

Conclusiones parciales

Con el estudio de los conceptos necesarios para comprender el campo de acción de la investigación y los elementos relacionados a esta, se logró crear las bases para el desarrollo de la investigación. El análisis de los sistemas que utilizan Minería de Procesos para la detección de incongruencias, permitió establecer las vías de desarrollo para la propuesta de solución. La asimilación de las tecnologías y herramientas, permitirá obtener un componente con las características funcionales y arquitectónicas para su integración al HIS del CESIM.

Capítulo II: Propuesta de solución

En el presente capítulo se describe el patrón arquitectónico y los patrones de diseño a utilizar. Se abordan los temas relacionados con el principio de funcionamiento y análisis del algoritmo *Heuristic Miner*. Además, se explican las etapas para la integración del componente en el HIS.

2.1 Patrones arquitectónicos

Los patrones arquitectónicos, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones (36).

Para el lograr la integración del componente se utiliza el patrón arquitectónico Modelo Vista Controlador (MVC).

2.1.1 Modelo Vista Controlador

El patrón conocido como Modelo Vista Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres capas diferentes (37), como se explicará a continuación:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Controla el flujo entre la vista y el modelo (los datos).

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras capas. Esta separación permite construir y probar el modelo independientemente de la representación visual (37).

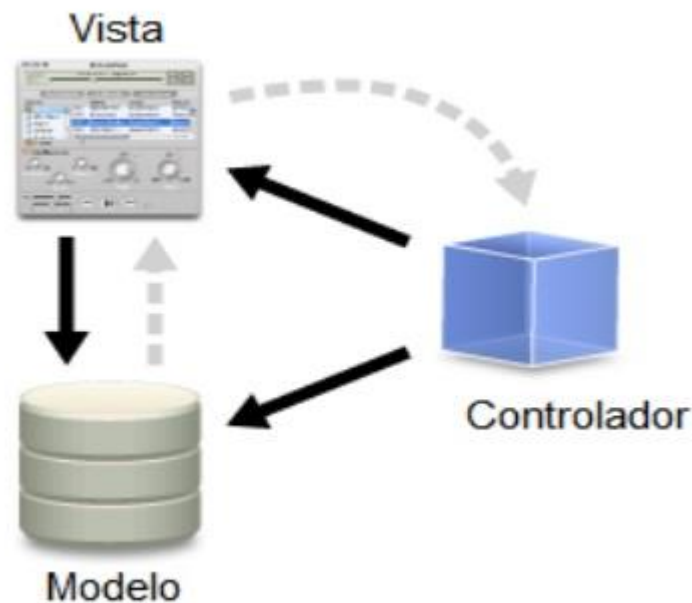


Figura 4: Representación del patrón MVC

Fuente: Elaboración propia.

Ventajas del estilo Modelo Vista Controlador

- **Soporte de múltiples vistas:** Dado que la vista se encuentra separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- **Adaptación al cambio:** Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celular.

2.2 Patrones de diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (38).

Ventajas del uso de patrones

Proporcionan una estructura conocida por todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos. Así la incorporación de un nuevo programador, no requerirá conocimiento de lo realizado anteriormente por otro. Permiten tener una estructura de código común a todos los proyectos que implemente una funcionalidad genérica. La utilización de patrones de diseño, permite ahorrar grandes cantidades de tiempo en la construcción de software (38).

En el desarrollo de la solución se utilizan los patrones para la asignación de responsabilidades (General Responsibility Assignment Software Patterns) más conocidos como GRASP, entre ellos el patrón experto, creador, alta cohesión, bajo acoplamiento y el controlador, con el objetivo de describir los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Experto

- ✓ Problema: ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?
- ✓ Solución: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
- ✓ Explicación: Experto es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la cosa real que representa, por lo que ofrece una analogía con el mundo real.
- ✓ Beneficios: Con la utilización de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener (39).
- ✓ Este patrón se evidencia en la clase AnalisisProcesos, la cual contiene la información necesaria para cumplir la responsabilidad asignada.

Creador

- ✓ Problema: ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?

- ✓ Solución: Asignarle a la clase B la responsabilidad de crear una instancia de la clase A en uno de los siguientes casos:
 - B agrega los objetos A.
 - B contiene los objetos A.
 - B registra las instancias de los objetos A.
 - B utiliza específicamente los objetos A.
 - B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).
 - B es un creador de los objetos A.
 - Si existe más de una opción, prefiera la clase B que agregue o contenga la clase A.
- ✓ Explicación: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.
- ✓ Beneficios: Brinda un soporte a un bajo acoplamiento, lo que supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización (39).
- ✓ Se evidencia en la clase MineríaHeuristica la cual es la encargada de crear instancias u objeto de la clase ActivitiesMappingStructures.

Bajo Acoplamiento

- ✓ Problema: ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización? El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras clases. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras. Este tipo de clases no es conveniente, ya que presentan los siguientes problemas:
 - Los cambios de las clases afines ocasionan cambios locales.
 - Son más difíciles de entender cuando están aisladas.
 - Son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen.
- ✓ Solución: Asignar una responsabilidad para mantener bajo acoplamiento.

- ✓ Explicación: El bajo acoplamiento es un principio que se debe tener siempre en cuenta durante las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades.
- ✓ Beneficios: Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar (39).
- ✓ Este patrón es evidenciado, por ejemplo, en la clase HeuristicMiner.

Alta Cohesión

- ✓ Problema: ¿Cómo mantener la complejidad dentro de límites manejables? En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. No conviene este tipo de clases pues presentan los siguientes problemas:
 - Son difíciles de comprender.
 - Son difíciles de reutilizar.
 - Son difíciles de conservar.
 - Son delicadas: las afectan constantemente los cambios.

Las clases con baja cohesión a menudo representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado a otros objetos.

- ✓ Solución: Asignar una responsabilidad de modo que la cohesión siga siendo alta.
- ✓ Explicación: El patrón Alta Cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón

evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande.

- ✓ Beneficios: Con el uso de este patrón mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifican el mantenimiento y las mejoras en funcionalidad. A menudo se genera un bajo acoplamiento. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico (39).
- ✓ Este patrón es evidenciado en todas las clases que contiene el sistema.

Controlador

- ✓ Problema: ¿Quién debería encargarse de atender un evento del sistema? Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema.

Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación.

- ✓ Solución: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:
 - El “sistema” global (controlador de fachada).
 - La empresa u organización global (controlador de fachada).
 - Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
 - Un manejador artificial de todos los eventos del sistema de un caso de uso (controlador de casos de uso).
- ✓ Explicación: La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. Otros medios de entrada son los mensajes externos o las señales procedentes de sensores como sucede en los sistemas de control de procesos. En todos los casos, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. La misma

clase controladora debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso.

- ✓ Beneficios: Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la de la interfaz. Al delegar a un controlador la responsabilidad de la operación de un sistema entre las clases del dominio favorece la reutilización de la lógica para manejar los procesos afines del negocio en aplicaciones futuras (39).
- ✓ El uso de este patrón es evidenciado en la clase controladora AnalisisProcesos.

2.3 Particularidades de *plugin Heuristic Miner* del ProM

La característica más importante de la *Heuristic Miner* es la robustez ante el ruido y excepciones, debido a que se basa en la frecuencia de patrones es posible centrarse en el comportamiento principal en el registro de eventos.

- **Excepciones:** Patrones pocos frecuentes.
- **Ruido:** Cuando el registro de eventos contiene comportamiento raro e infrecuente no representativo para el comportamiento típico del proceso. (i.e. deformaciones por falta de uno o más eventos de la traza, eventos intercambiados o tareas del inicio o el final de las trazas inexistentes).

La representación interna del modelo de proceso de la *Heuristic Miner* es llamada red heurística. Sin embargo utilizando ProM es posible convertir la red heurística en una red de Petri. Después es posible usar varios *plugins* del ProM asociados con redes de Petri (11).

2.3.1 Algoritmo *Heuristic Miner*

El algoritmo *Heuristic Miner*, está basado en la construcción de un grafo de dependencias a partir de los distintos eventos que se producen dentro de una traza de ejecución. Este grado de dependencia, representará el flujo de trabajo inferido por el sistema. El algoritmo *Heuristic Miner* solo considera el orden de los eventos en la instancia.

El grafo de dependencias contendrá como nodos, los eventos ocurridos en las trazas de ejecución. Los arcos se calculan mediante la función $a \Rightarrow wb$, siendo a y b eventos de las trazas. La función se define formalmente como:

$$a \Rightarrow_w b = \frac{|a > wb| - |b > wa|}{|a > wb| + |b > wa| + 1}$$

La función $a >_w b$ representa si el evento b se alcanza directamente desde a con únicamente un paso. $|a >_w b|$ representa el número de veces que tras un evento a ocurre b en la traza de la instancia.

Dados a y b , $a \Rightarrow_w b$ es un número entre -1 y 1 . Cuanto más cercano a -1 menor probabilidad de ser una dependencia, mientras que cuanto más cercano a 1 mayor probabilidad de serlo. Por otro lado, si es cero, significa no existen datos de esa relación. Si el resultado de esta función es mayor que un umbral dado, se dibujará un arco entre a y b . Los eventos y sus arcos asociados formarán, finalmente el flujo de trabajo inferido.

El algoritmo construye flujo de trabajo muy sencillo. Sin embargo, este algoritmo, está pensado para inferir un flujo de trabajo con patrones sin acciones paralelas ni sincronizaciones. Esto es debido a que las características de los grafos impiden la definición de procesos paralelos (40).

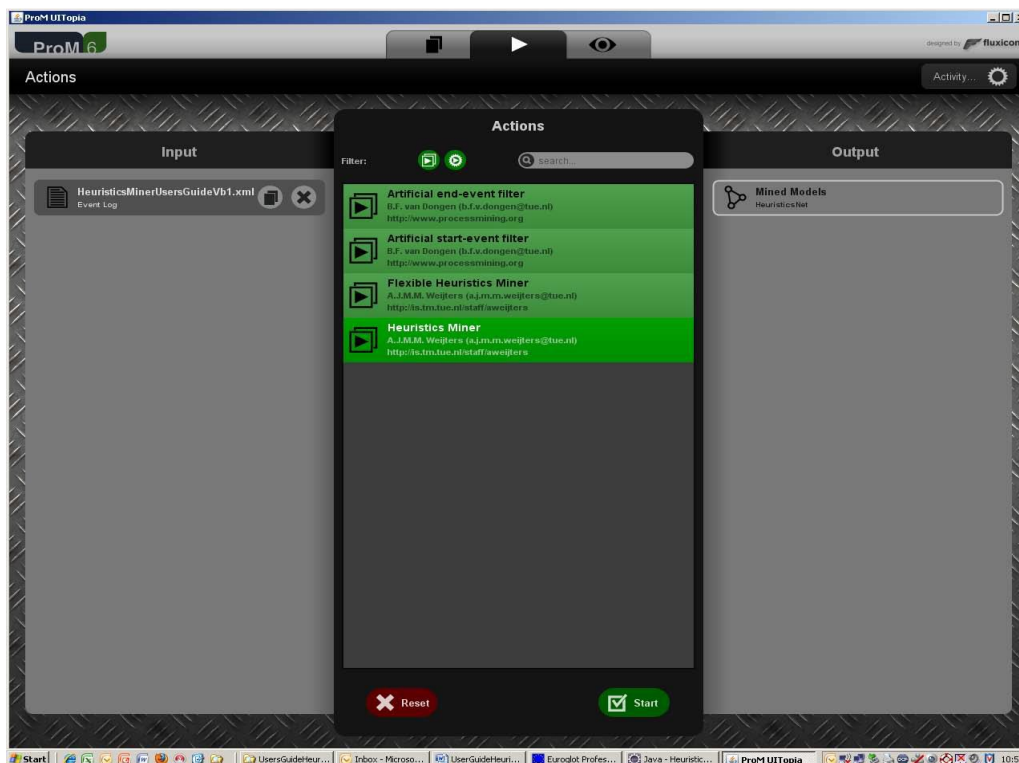


Figura 5: *Plugin Heurisc Miner* del ProM

Fuente: (ProM)

2.4 Flujo de información de la personalización del plugin *Heuristic Miner*

A continuación se muestra el flujo de información, teniendo en cuenta las características del sistema.

El HIS cuenta con una bitácora de procesos en su base de datos donde se almacenan los eventos del sistema. Para utilizar la información almacenada sobre las actividades de procesos, se utiliza el componente para la extracción y transformación de las trazas de ejecución, el cual tiene como objetivo recolectar estas trazas y estructurarlas jerárquicamente para luego proceder a organizarlas en un registro de eventos. Este registro de eventos constituye el punto de partida para su análisis aplicando la técnica *Heuristic Miner* teniendo como resultado la visualización de modelos de proceso. Teniendo en cuenta que el plugin *Heuristic Miner* desarrollado para la herramienta ProM no está apto para su uso por parte del personal no especializado en Minería de Procesos, por lo que es necesario su personalización y adaptación a las tecnologías en las que está desarrollado el HIS. Por último, se propone su integración al sistema para facilitar su uso y aplicación por parte del personal clínico administrativo. Para una mayor comprensión de la visualización, se realizaron los productos de trabajo correspondientes, ver [Anexo 4](#) y [Anexo 5](#).

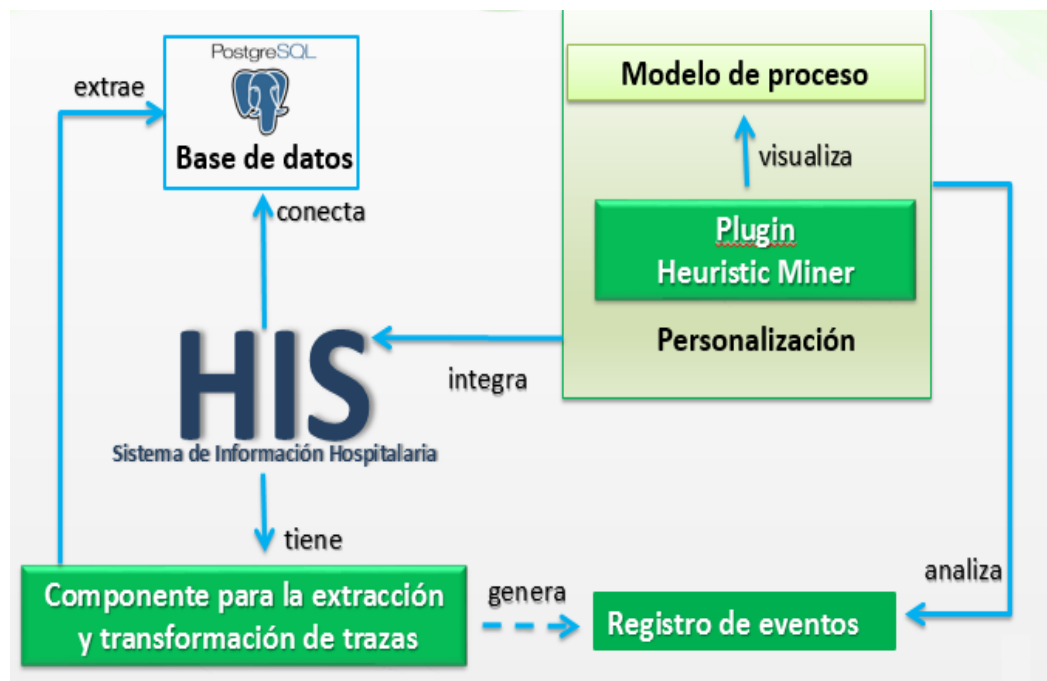


Figura 6: Flujo de información de la personalización del plugin *Heuristic Miner*

Fuente: Elaboración propia.

2.5 Parámetros del *plugin Heuristic Miner*

A continuación se muestran un conjunto de parámetros definidos para el *plugin Heuristic Miner* del ProM 6.4, para luego seleccionar los parámetros a utilizar en la personalización a desarrollar:

- **Relative-to best:** Umbral que permite dibujar el arco entre dos actividades si la diferencia entre la medida de dependencia entre estas dos actividades y la "mejor" medida de dependencia en todo el modelo es menor que el valor especificado en el umbral, en caso contrario el arco no se dibuja en el modelo (41).
- **Dependency:** Umbral que permite mostrar los arcos en el modelo de proceso cuyo valor de dependencia entre las actividades que este conecta sea mayor o igual que el valor especificado en el umbral, en caso contrario el arco no se dibuja en el modelo. La medida de dependencia indica cuán certera es la relación entre dos actividades, un valor alto (cerca de 1) significa que estamos seguros que hay una relación de dependencia entre las actividades conectadas. Esta relación se calcula, entre las actividades a y b, como (42):

$$a \Rightarrow wb = \frac{|a > wb| - |b > wa|}{|a > wb| + |b > wa| + 1}$$

- **Lenght-one loop:** Umbral que permite mostrar los arcos que conectan una actividad con ella misma si el valor de la dependencia del arco es mayor o igual que el valor del umbral. En caso contrario no se dibuja el arco en el modelo. El valor de la medida de dependencia para estos casos se calcula de la siguiente manera siendo a una actividad del proceso (42):

$$a \Rightarrow wa = \frac{|a > wa|}{|a > wa| + 1}$$

- **Lenght-two loop:** Umbral que permite mostrar un bucle entre dos actividades a y b si la cantidad de veces que a es seguido directamente por b y luego ocurre a de nuevo está igual o por encima del valor especificado en el umbral, este bucle se puede representar con la expresión $a \Rightarrow w^2b$ y se calcula con la formula (42):

$$a \Rightarrow w^2b = \frac{|a >^2wb| - |b >^2wa|}{|a >^2wb| + |b >^2wa| + 1}$$

- **Long distance:** Umbral que permite mostrar un arco entre dos actividades si el valor de la relación de dependencia de larga distancia entre estas actividades es mayor o igual al valor especificado en el umbral. Dos actividades a y b están en una "relación de larga distancia" si hay una dependencia entre ellos, pero no es una sucesión directa. Esta relación se expresa mediante la fórmula:

$$a \Rightarrow^1wb = \frac{|a >>>> wb|}{|b| + 1}$$

Donde $|a >>>> wb|$, indica el número de veces que una es directamente o indirectamente (es decir, si hay otras actividades diferentes entre a y b) (42).

- **All tasks connected:** Heurística que en caso de estar activada se mostraran todas las actividades del modelo conectadas ignorando de esta forma los umbrales *Relative-to best* y *Dependency* en caso contrario si se toman en cuenta los umbrales mencionados.
- **Long distance dependency:** Heurística que permite decidir si tener en cuenta el umbral *Long distance* al modelar el proceso.
- **Ignore loop dependency thresholds:** Heurística que decide si ignorar o no los umbrales *Length-one loop* y *Length-two loop*.

En la (Tabla 1), se muestra el tipo de dato de los parámetros.

Tabla 1: Parámetros del *plugin Heuristic Miner*.

Nombre del parámetro	Tipo de dato
<i>All tasks connected</i>	Boolean
<i>Dependency threshold</i>	Float
<i>Length-1-loop threshold</i>	Float
<i>Length-2-loop threshold</i>	Float
<i>Long distance</i>	Int
<i>Relative to best</i>	Float
<i>Long distance dependency</i>	Boolean

<i>Ignore loop dependency thresholds</i>	Boolean
--	---------

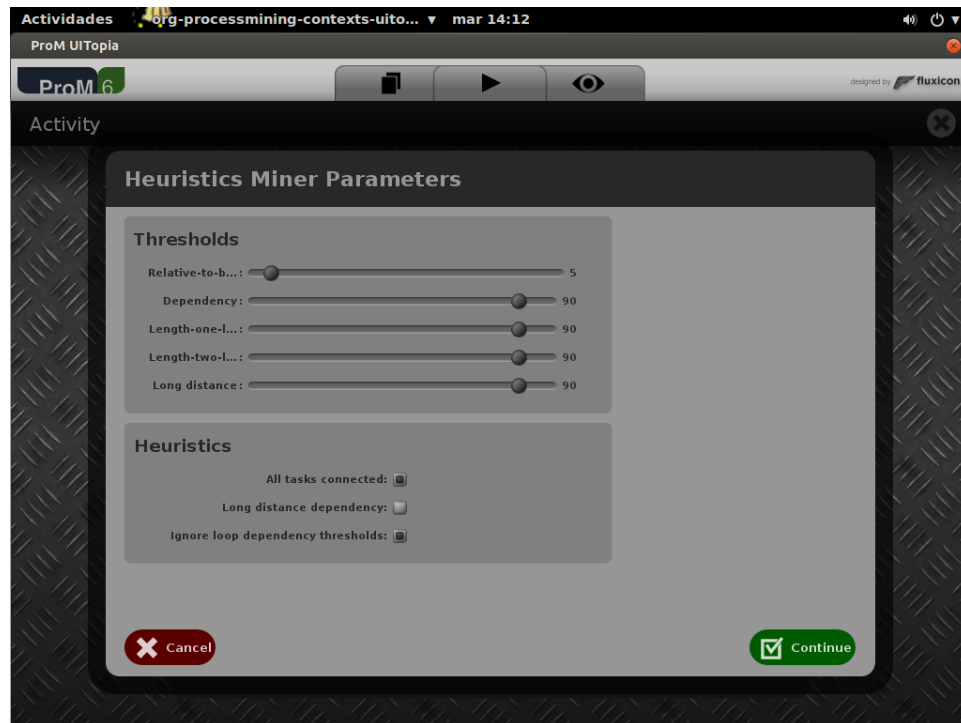


Figura 7: Parámetros del *plugin Heuristic Miner* del ProM

Fuente: (ProM).

2.6 Integración del *plugin Heuristic Miner* al HIS

El *plugin Heuristic Miner*, tiene como objetivo fundamental incorporar al HIS una herramienta que detecte incongruencias en los procesos del sistema. Para su correcta integración se definen tres etapas fundamentales.

2.6.1 Etapa 1: Generación del registro de eventos

A partir del componente para la extracción y transformación de las trazas de ejecución, se obtiene un fichero de extensión .xes, el cual contiene la información del registro de evento que permitirá aplicar la técnica *Heuristic Miner* en el HIS, en la **Figura 8** se muestra una instancia perteneciente al proceso “Solicitar producto”, en ella se detallan los atributos y tareas que conforman cada evento.

```
<event>
<string key="org:resource" value="UNDEFINE"/>
<date key="time:timestamp" value="2012-02-29T08:37:53.146+01:00"/>
<string key="concept:name" value="ver_detalle_sol_bq"/>
<string key="lifecycle:transition" value="start"/>
</event>
```

Figura 8: Registro de evento del proceso “Solicitar producto”

Fuente: Elaboración propia.

org:resource: Es el nombre de quien ejecuta la actividad.

time:timestamp: La fecha y la hora en que se ejecutó la actividad.

concept:name: Corresponde al nombre de la actividad.

lifecycle:transition: Ciclo de vida de la transición, en este caso puede tomar dos valores (*start*, *complete*).

Luego se procede a ejecutar la técnica para generar el modelo de flujo de actividades de los procesos en el sistema.

2.6.2 Etapa 2: Análisis de las configuraciones para el HIS

A partir de la necesidad de crear un componente intuitivo y fácil de utilizar para el personal clínico administrativo, garantizando así la correcta interacción con el sistema, se decide prescindir visualmente de todos los parámetros de entrada con que cuenta el *plugin* original, los cuales dificultan la comprensión y el entendimiento de los modelos generados por parte de los usuarios, por esto obtendrán un valor por defecto. El componente desarrollado cuenta con un conjunto de configuraciones que constituyen la base fundamental para la obtención de modelos de procesos precisos en el ámbito hospitalario. A continuación se presentan dichas configuraciones:

Medida de transición: Mide el comportamiento de la relación entre las actividades.

- **Dependencia:** Define cuan certera es la relación entre dos actividades. El valor de esta dependencia se acota de 0-1, cuanto más próximo a 1 indica seguridad en la relación de dependencia entre las actividades conectadas. Se muestra en el arco que une las actividades.
- **Frecuencia:** Número de veces que se repite un evento en el sistema en un intervalo de tiempo determinado.

Flujo principal: Muestra en color fuerte las actividades más frecuentes en el modelo.

Semántica: El lenguaje a utilizar en la representación gráfica de los modelos de procesos.

1) **Sin semántica:** Se representa el modelo de proceso en una red heurística, la cual muestra la frecuencia y dependencia entre las transiciones.

- **Red heurística:** Los rectángulos son las tareas; los arcos indican la dependencia entre tareas. El número dentro de cada tarea indica la cantidad de veces que las tareas se llevan a cabo (es decir, el registro de eventos contiene 497 ocurrencias

de tarea D). El número de los arcos indica la frecuencia o dependencia entre las transiciones.

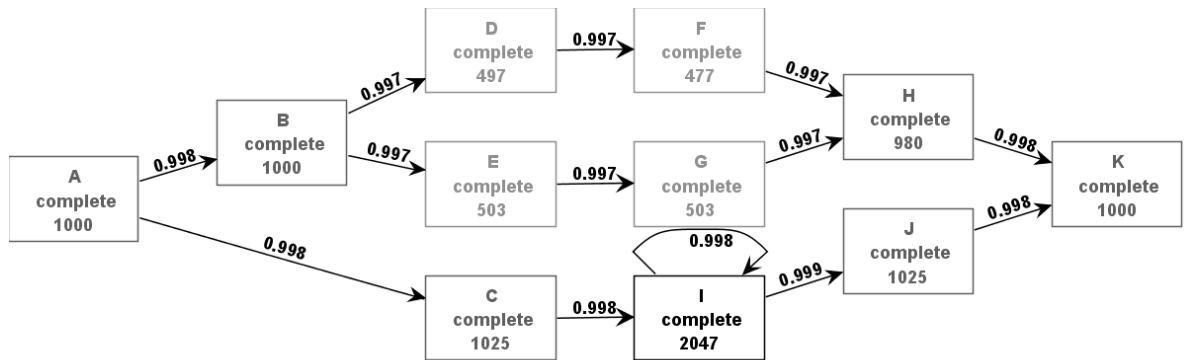


Figura 9: Modelo de proceso, representado mediante red heurística

Fuente: (11).

2) **Con semántica:** Se representa el modelo mediante la notación BPMN, la cual proporciona diagramas de flujo detallados, con suficiente información como para poder analizar el proceso y simularlo.

➤ **Notación BPMN:** El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio sea fácil entender el flujo y el proceso. Entre las categorías básicas del flujo de proceso se encuentran los elementos (43):

- **Eventos:** Están representados gráficamente por un círculo y describen algo que sucede.
- **Actividades:** Se representan por un rectángulo con sus vértices redondeados y describe el tipo de trabajo que será realizado.
- **Control de Flujo:** Se representan por una figura de diamante y determinan si se bifurcan o se combinan las rutas dependiendo de las condiciones expresadas.

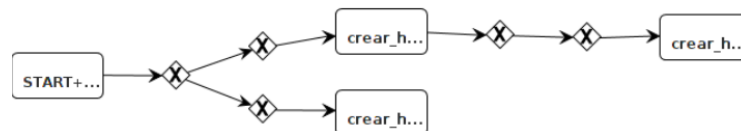


Figura 10: Modelo de proceso, representado mediante la notación BPMN

Fuente: Elaboración propia.

2.6.3 Etapa 3: Visualización de los modelos

Para la visualización de los modelos es necesario seleccionar el proceso, el tipo de análisis y el intervalo de fecha de inicio y fin, luego se procede a la opción “**Analizar**”, (Figura 12):



Figura 12: Interfaz para analizar el proceso

Fuente: Elaboración propia.

A partir de seleccionar la opción, se muestran las diferentes configuraciones a utilizar que descubren como resultado un modelo de proceso, (Figura 13):

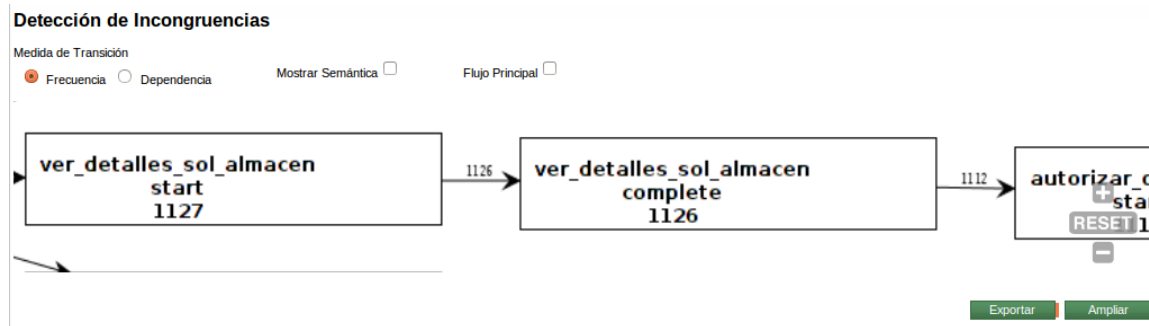


Figura 13: Configuraciones del componente *Heuristic Miner*

Fuente: Elaboración propia.

Luego de obtener el modelo de proceso, se puede aumentar la imagen en otra pestaña, mediante la opción “**Ampliar**” y guardar la imagen como (.PNG), a través del botón “**Exportar**”.

2.7 Estructura del componente

El componente está compuesto por varios paquetes. A continuación se muestra su estructura y las clases principales contenidas en cada uno:

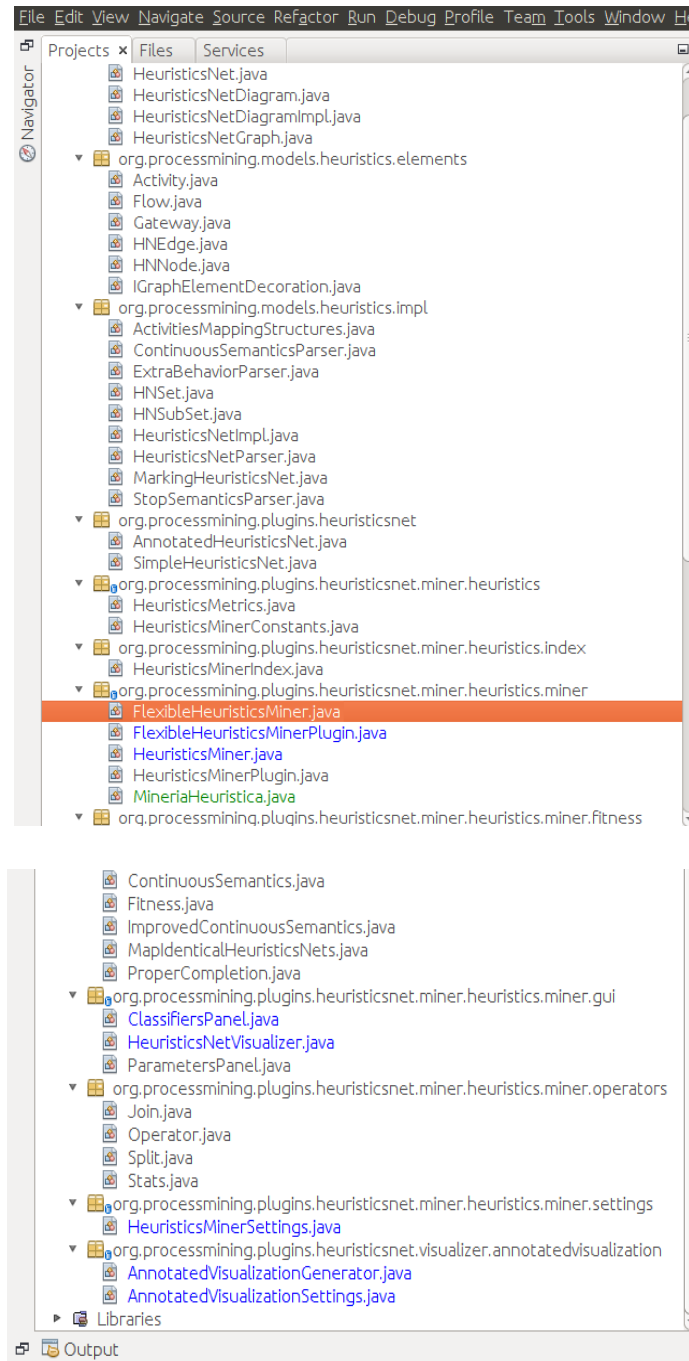


Figura14: Estructura del componente *Heuristic Miner*

Fuente: Elaboración propia.

org.processmining.models.heuristics: Este paquete contiene la definición general de la estructura de la red heurística y su visualización.

Clases Principales:

HeuristicsNet: Interfaz común de los objetos de la red heurística.

HeuristicsNetDiagram: Clase interfaz que define el diagrama correspondiente a una red heurística.

HeuristicsNetDiagramImpl: Clase que implementa el diagrama correspondiente a una red heurística.

HeuristicsNetGraph: Hereda de la clase HeuristicsNetDiagramImpl y se encarga de crear una representación visual de una red heurística.

org.processmining.models.heuristics.elements: Este paquete contiene los elementos que componen la visualización o modelo de proceso de una red heurística como son las actividades, las aristas, compuertas, entre otros.

Clases Principales:

Activity: Define la estructura de las actividades del modelo.

Flow: Define la estructura de las aristas del modelo.

Gateway: Define la estructura de las compuertas del modelo.

org.processmining.models.heuristics.impl: Este paquete contiene las clases que implementan la estructura de una red heurística y transforma un registro de eventos en una red heurística.

Clases Principales:

ActivitiesMappingStructures: Ayuda a la creación de la red heurística garantizando que los índices asignados a las actividades de la red heurística sean los mismos una vez realizado el proceso inverso.

ContinuousSemanticsParser: Implementa la interface HeuristicsNetParser. Reproduce las trazas de un registro de eventos en una red heurística usando una semántica continua. Esto significa que el proceso de análisis tiene lugar incluso cuando las actividades no están permitidas durante la reproducción del registro. En cambio, los problemas encontrados durante el análisis son registrados y pueden ser recuperados a través de los diversos métodos "get" que se proporcionan en esta clase.

ExtraBehaviorParser: Hereda de la clase ContinuousSemanticsParser. Construye un objeto del tipo ExtraBehaviorParser el cual reproducirá las trazas de un registro de eventos en una red heurística dada. El aspecto aleatorio de este proceso de reproducción está basado en un generador aleatorio proporcionado.

HeuristicsNetImpl: Implementación de una red heurística.

HeuristicsNetParser: Interfaz de los analizadores sintácticos para los objetos de HeuristicNet.

StopSemanticsParser: Reproduce las trazas en un registro de eventos en una red heurística usando una parada de semántica. Esto significa que el analizador sintáctico para cuando alguna actividad no está habilitada durante la reproducción del registro de eventos.

org.processmining.plugins.heuristicsnet: Este paquete contiene las clases que definen los dos tipos de red heurística, las cuales son red simple o una red con anotaciones.

Clases Principales:

SimpleHeuristicsNet: Hereda de la clase HeuristicsNetImpl. Define una red heurística simple o sea sin mostrar la semántica.

AnnotatedHeuristicsNet: Hereda de la clase SimpleHeuristicsNet. Agrega a la Red Heurística la semántica, es decir permite que cuando se genere la visualización se muestran las compuertas como los splits y joins.

org.processmining.plugins.heuristicsnet.miner.heuristics

En este paquete están contenidas las métricas que utiliza el plugin para el procesamiento interno que realiza y las constantes que son los parámetros por defecto de la técnica.

Clases Principales:

HeuristicsMetrics: Define las métricas que utiliza el *plugin*.

HeuristicsMinerConstants: Define las constantes que son los parámetros por defecto de la técnica

org.processmining.plugins.heuristicsnet.miner.heuristics.index: En este paquete se encuentra la clase HeuristicsMinerIndex que implementa el índice de la red heurística.

org.processmining.plugins.heuristicsnet.miner.heuristics.miner: Contiene las clases que definen la minería heurística y generan la red heurística a partir de estas.

Clases Principales:

FlexibleHeuristicsMinerPlugin: Clase Plugin, la cual mina un registro de evento dado utilizando la minería heurística y tiene como resultado la red heurística.

HeuristicsMiner: Define la técnica minería heurística

org.processmining.plugins.heuristicsnet.miner.heuristics.miner.fitness: Este paquete contiene las clases que calculan el ajuste del modelo de proceso generado con respecto al log que se está minando.

Clases Principales:

Fitness: Interfaz de las clases para calcular el ajuste de la red heurística con respecto al log.

org.processmining.plugins.heuristicsnet.miner.heuristics.miner.gui: Contiene las clases que se encargan de generar la visualización de la red heurística.

Clases Principales:

HeuristicsNetVisualizer: Clase encargada de generar la visualización la cual contiene el modelo de proceso.

org.processmining.plugins.heuristicsnet.miner.heuristics.miner.operators: Especifica los operadores del modelo de proceso como son las diferentes compuertas de este.

Clases Principales:

Join: Implementa la compuerta Join.

Split: Implementa la compuerta Split.

org.processmining.plugins.heuristicsnet.miner.heuristics.miner.settings:

Contiene la clase HeuristicsMinerSettings en la cual se definen los parámetros de entrada que utiliza la técnica para el procesamiento del registro de eventos.

org.processmining.plugins.heuristicsnet.visualizer.annotatedvisualization:

Contiene las clases que definen las configuraciones de la visualización.

Clases Principales:

AnnotatedVisualizationSettings: Contiene las configuraciones de la visualización de la red heurística.

Conclusiones parciales

Con la utilización de los principales patrones de diseño y el patrón arquitectónico, se garantizó la organización estructural del componente. Además, al definir las etapas para la integración del *plugin Heuristic Miner* al HIS, permitió guiar el desarrollo de la personalización, logrando crear interfaces intuitivas para modelar los procesos y exportar los modelos obtenidos como imagen en formato PNG.

Capítulo III: Validación de la propuesta de solución

En este capítulo se realiza la validación del componente *Heuristic Miner* para el HIS, para la detección de incongruencias en sus procesos, mostrando los resultados obtenidos.

3.1 Sistema de Información Hospitalaria del CESIM

El HIS provee la informatización de los procesos fundamentales de la capa clínica del nivel secundario de atención médica y está orientado a satisfacer las necesidades de almacenamiento, procesamiento, recopilación e interpretación de la información relacionada con la atención al paciente y la labor administrativa de todas las actividades de la institución hospitalaria (44).

Se realizan diferentes procesos en dependencia del área o servicio, los cuales requieren la optimización de los recursos humanos y materiales, así como minimizar los inconvenientes y retrasos que existen. A partir de los procesos del sistema, aplicando el componente personalizado, se obtienen modelos que permiten detectar incongruencias en los procesos, lo cual luego de los análisis por parte del personal clínico administrativo autorizado permite incidir sobre la calidad de los servicios que se prestan.

3.2 Diseño de la validación

El objetivo fundamental de la validación es comprobar que el componente desarrollado detecta incongruencias en los procesos del HIS. Por tal motivo, la validación se divide en dos etapas.

- **Primera etapa:** Dirigida a la detección de incongruencias en algunos procesos del HIS.
- **Segunda etapa:** Orientada a especificar otros tipos de análisis que permiten enriquecer los factores necesarios para una correcta toma de decisiones sobre la ejecución de las actividades de procesos.

3.3 Detección de incongruencias en los procesos del HIS

Para validar la solución propuesta se analizarán los procesos “Solicitar producto”, “Procesar solicitud” y “Desincorporar producto”, pertenecientes al módulo Almacén, el cual se encarga de gestionar el flujo de actividades sobre los distintos movimientos que puede tener un producto en un almacén. Los modelos de procesos de negocio mostrados a continuación están realizados a partir de los diagramas de flujo de trabajo

generados por JBPM (Java Business Process Management), que es el motor de control de flujo del sistema, el cual administra las instancia de los procesos.

Análisis del proceso “Solicitar producto”

A partir del registro de eventos se procede a utilizar el componente *Heuristic Miner* para generar el modelo de flujo de actividades del proceso “Solicitar producto”, mediante la notación BPMN.

El registro de eventos fue generado en las fechas comprendidas entre el 10 de abril del 2008 y 9 de abril del 2015, el resultado es el modelo que se muestra en la **Figura 15**, donde se evidencia irregularidades con respecto al proceso de negocio (**Figura 16**). (i.e la actividad “Autorizar despacho”, debe estar representada como una tarea final del proceso, sin embargo en el modelo aparece como un nodo intermedio). Además aparecen relaciones que no están definidas en el proceso de negocio como; las actividades “Modificar pedido” y “Ver detalle de solicitud” al igual que las actividades “Aceptar niveles” y “Ver detalle de solicitud”.

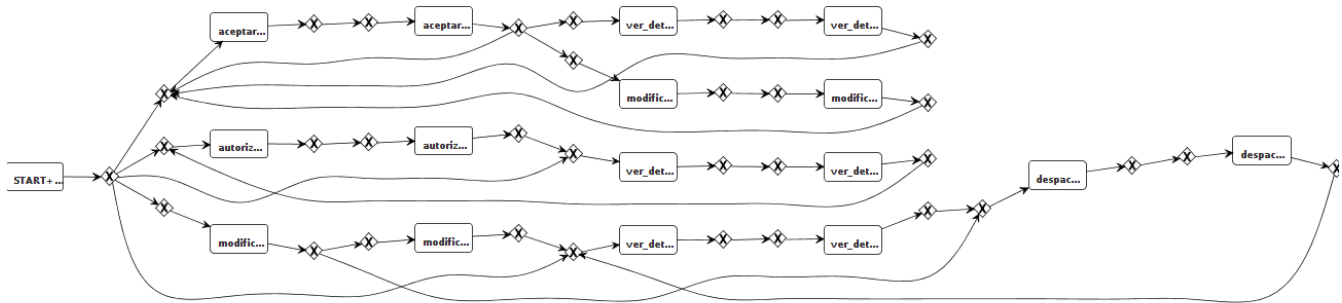


Figura 15: Modelo BPMN del proceso “Solicitar producto” obtenido del componente *Heuristic Miner*

Fuente: Elaboración propia.

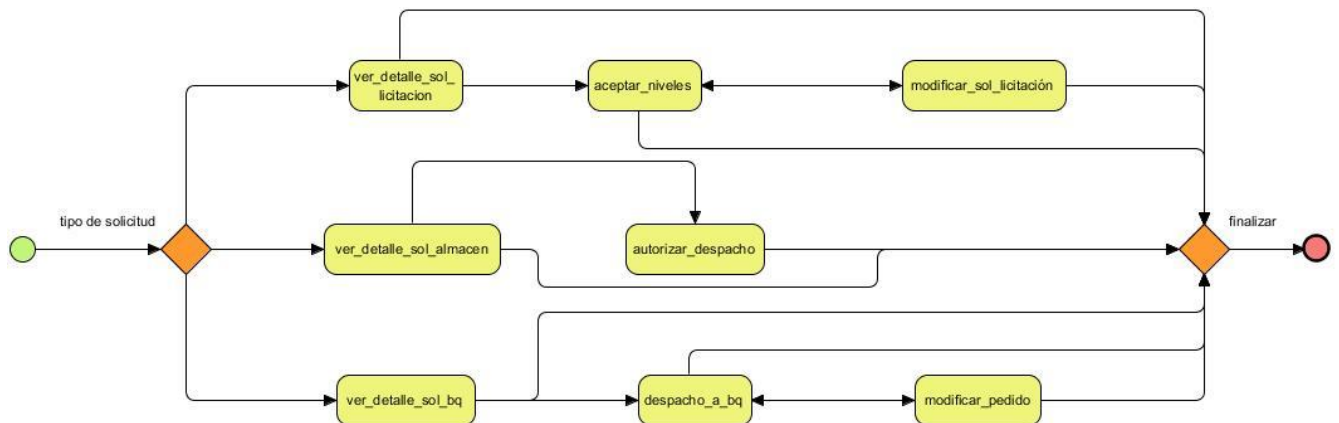


Figura 16: Diagrama de Procesos del Negocio del proceso “Solicitar producto”

Fuente: Elaboración propia.

Análisis del proceso “Procesar solicitud”

El registro correspondiente al proceso fue generado entre las fechas 1 de mayo del 2010 y 9 de junio del 2015, donde se obtiene como resultado el modelo de la **Figura 17**, el cual se compara con el modelo de proceso de negocio de la **Figura 18**, y se comprueba que el flujo de actividades obtenido no se corresponde con el diseñado originalmente en el proceso de negocio, porque de cinco actividades que deben ejecutarse en el proceso, solamente dos son las que están registradas en el modelo de proceso generado por el componente, hay actividades que nunca fueron ejecutadas por el sistema, tales como “Procesar oferta” y “Aprobar adjudicación”.



Figura 17: Modelo BPMN del proceso “Procesar solicitud” obtenido del componente

Heuristic Miner del HIS

Fuente: Elaboración propia.

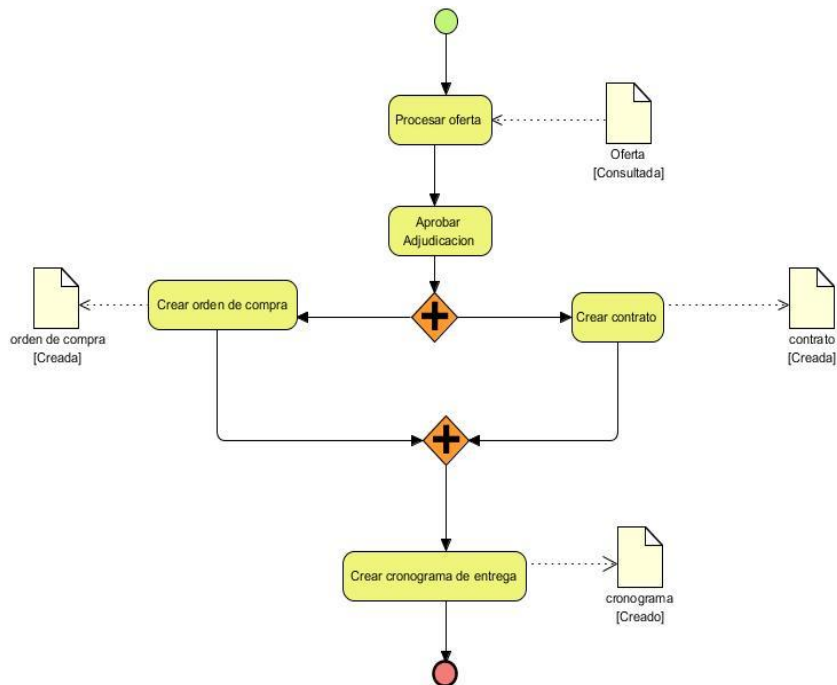


Figura 18: Diagrama de Procesos del Negocio del proceso “Procesar solicitud”

Fuente: Elaboración propia.

Análisis del proceso “Desincorporar producto”

En este caso fue generado un registro de eventos comprendido entre el 17 de enero del 2008 y el 2 de abril del 2014. Como en los procesos analizados anteriormente, el proceso “Desincorporar producto”, presenta irregularidades con respecto al proceso de negocio, a partir de la lógica del proceso de negocio, la primera actividad a realizar debe ser “Desincorporar producto” y la próxima “Acta de desincorporación”, sin embargo esta transición no se modela, pues la actividad “Desincorporar producto” no se representa en el modelo, como se observa (Figura19).



Figura 19: Modelo BPMN del proceso “Procesar solicitud” obtenido del componente *Heuristic Miner* del HIS

Fuente: Elaboración propia.

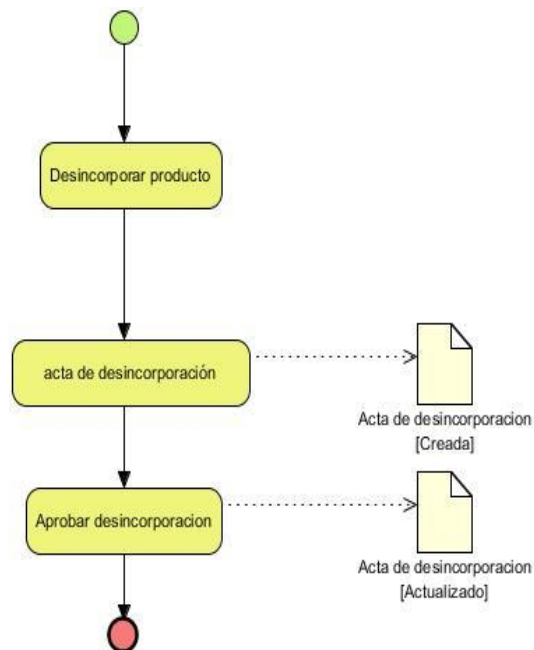


Figura 20: Diagrama de Procesos del Negocio del proceso “Desincorporar producto”

Fuente: Elaboración propia.

3.4 Conclusión de los procesos analizados

Luego del análisis realizado, a modo general se puede evidenciar la presencia de incongruencias en los procesos, puesto que los modelos generados por el componente

difieren con la definición de los procesos de negocio. Durante el período de 4 a 7 años, que fueron ejecutados los procesos, existen actividades que nunca fueron generados por el sistema, también la presencia de transiciones que no están definidas en el proceso de negocio. Por tanto, se demuestra que existen inconsistencias en la ejecución de los procesos del HIS, limitando un correcto análisis sobre la ejecución de algunas actividades de proceso que nunca fueron realizadas.

3.5 Análisis de las configuraciones del componente *Heuristic Miner*

Otras posibilidades que ofrece el componente es el uso de configuraciones, que permiten obtener información significativa acerca de las actividades y sus relaciones.

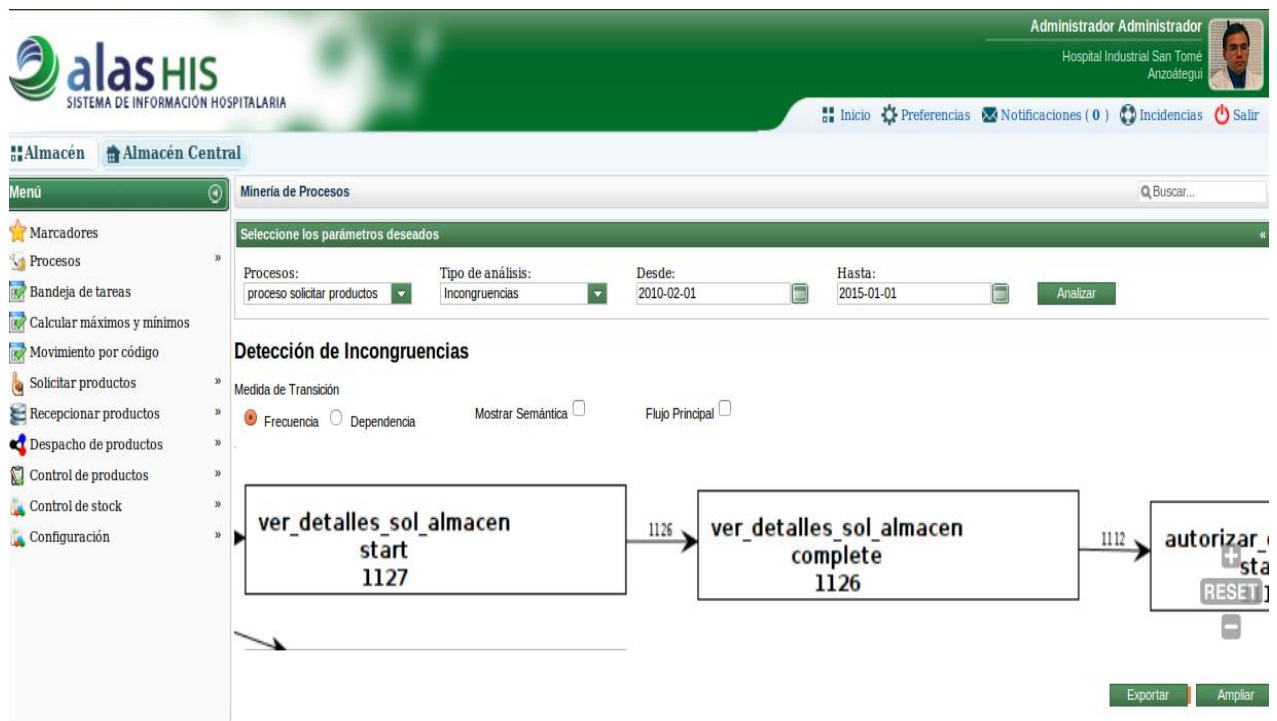


Figura 21: Interfaz del componente *Heuristic Miner*

Fuente: Elaboración propia.

Para un el entendimiento de estas configuraciones, se toma el proceso “Solicitar producto” y es modelado a partir de una red heurística.

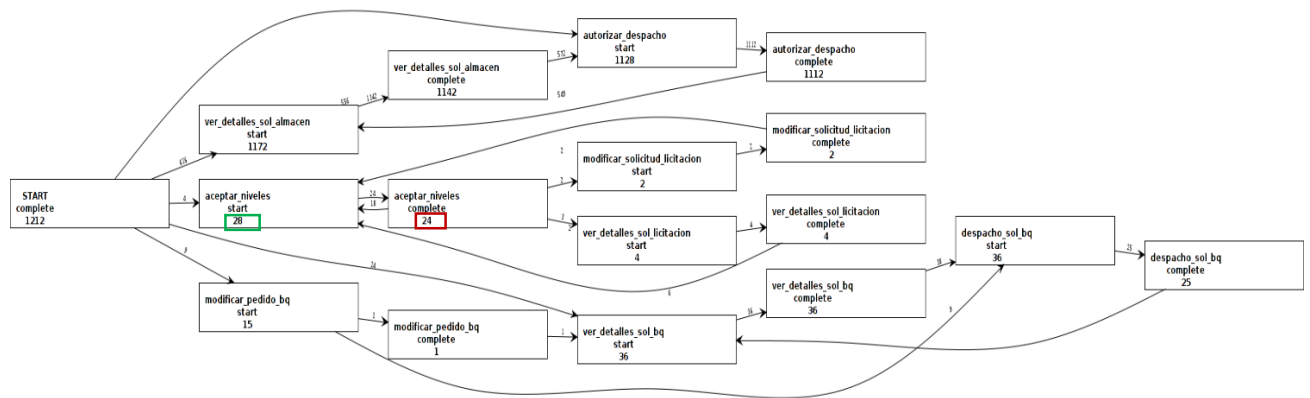


Figura 22: Modelo de red heurística del proceso “Solicitar producto”

Fuente: Elaboración propia.

El modelo obtenido en la **Figura 22** define dos nodos por cada actividad, en este caso puede tomar dos valores (*start* o *complete*), los cuales están asociados al ciclo de vida de la actividad. Las actividades de tipo *start* (el número marcado en verde en el nodo define cuantas veces fue iniciada esa actividad) son las que comienzan la actividad; mientras las de tipo *complete* (el número marcado en rojo en el nodo define cuantas veces fue terminada esa actividad) son las que finalizan la actividad (i.e. la actividad “Aceptar niveles”, fue iniciada en 28 ocasiones y de ellas se completaron 24).

Frecuencia

El modelo representado en la **Figura 23**, fue generado teniendo en cuenta la frecuencia, la cual permite definir la cantidad de veces que fue ejecutada cada transición y actividad. Como resultado en el proceso analizado, se identifica el flujo de actividades más frecuente, con un valor de 1142 el compuesto por las actividades, “Ver detalles de solicitud” y “Autorizar despacho”, se muestra en un círculo rojo en la figura.

Capítulo III: Validación de la propuesta solución

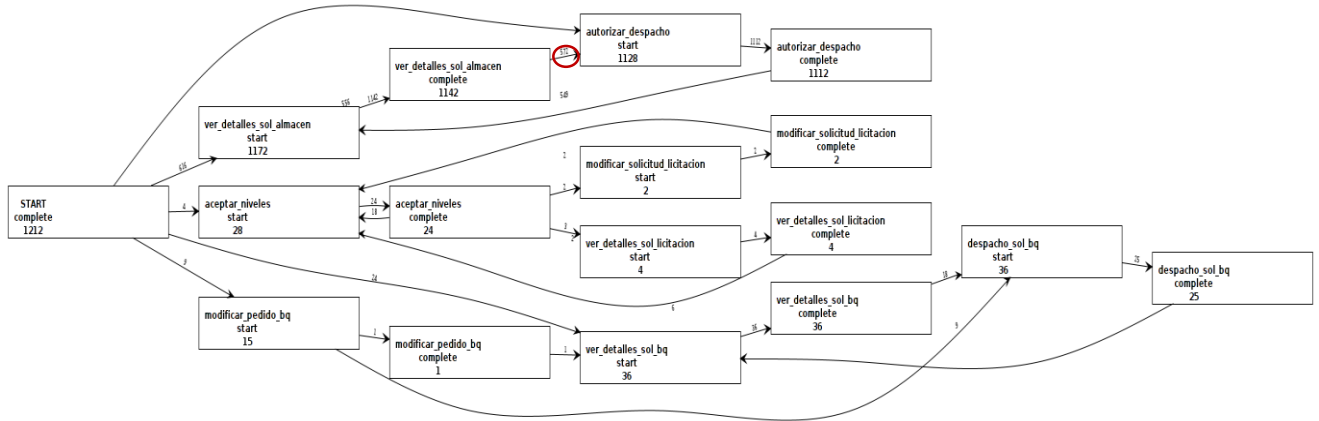


Figura 23: Modelo del proceso “Solicitar producto“, mediante la configuración de frecuencia

Fuente: Elaboración propia.

Dependencia

La **Figura 24** representa el modelo generado a partir de la configuración de dependencia, donde se define cuan certera es la relación entre dos actividades (i.e. al analizar el modelo se puede concluir que las actividades con mayor dependencia son “Ver detalle de solicitud” y “Autorizar despacho” con un valor de 0.998), en la figura se señala en un círculo rojo.

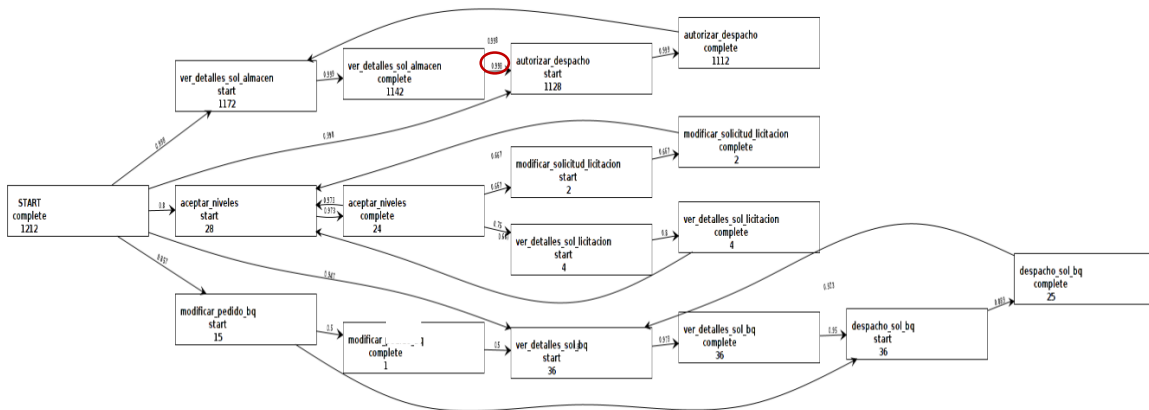


Figura 24: Modelo del proceso “Solicitar producto“, mediante la configuración de dependencia

Fuente: Elaboración propia.

Flujo principal

Por otro lado, el flujo principal del modelo (representada con mayor énfasis en el modelo obtenido) se muestra a partir de las actividades con mayor frecuencia en el proceso.

Como se muestra en la **Figura 25**, el flujo principal está representado por las actividades “Ver detalles de solicitud” y “Autorizar despacho” representando un 95 % de la ejecución de las actividades.

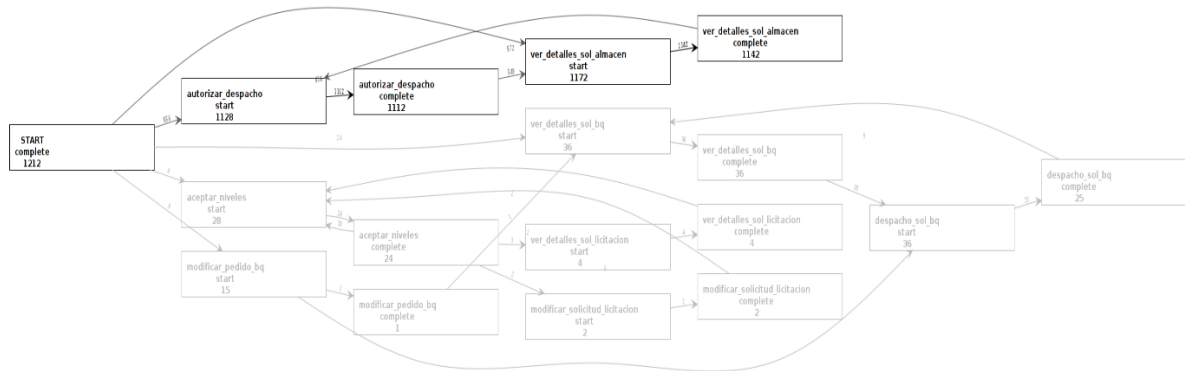


Figura 25: Flujo principal del proceso “Solicitar producto”

Fuente: Elaboración propia.

Conclusiones parciales

Al validar la solución propuesta, permitió detectar incongruencias en relación con las secuencias y ausencia de las actividades en los procesos, observándose poca solidez de los procesos ejecutados, con los modelos reales del proceso, debido a que los modelos obtenidos presentan un alto nivel de ajuste sin embargo muestran poca correspondencia con el modelo de negocio. A partir de los análisis obtenidos se demostró que el componente desarrollado permite detectar incongruencias y realizar otros tipos de análisis como la frecuencia, dependencia y flujo principal.

Conclusiones

Con la presente investigación se le dio cumplimiento al objetivo general propuesto, por lo que se concluye que:

- Se analizaron los principales conceptos relacionados al objeto de estudio de la investigación lo que permitió, definir el marco teórico conceptual que sustenta la investigación y seleccionar el *plugin Heuristic Miner* de Minería de Procesos como propuesta de solución a la problemática planteada.
- Con el desarrollo e integración de la personalización del *plugin Heuristic Miner* para HIS, se logró crear interfaces intuitivas, para modelar las instancias de proceso a partir del registro de eventos y obtener modelos que permiten detectar incongruencias en el comportamiento de los procesos del sistema.
- Al validar la solución propuesta se demostró que el componente desarrollado permite, detectar incongruencias en los procesos del HIS y realizar análisis de frecuencia, dependencias y del flujo principal de las actividades de proceso.

Recomendaciones

Para dar continuidad a este trabajo, los autores proponen las siguientes recomendaciones:

- Desarrollar un componente que permita interpretar los modelos obtenidos.
- Desarrollar un componente que compare el flujo de actividades de negocio definidas con los modelos obtenidos.

Referencias Bibliográficas

1. **Castillo Pino, Argelio and Díaz Fernández, Daniel.** *Sistema para el tratamiento de procesos desestructurados.* UCI. La Habana : s.n., 2013. p. 6.
2. **Nogueira.** *Fundamentos para el Control de Gestion empresaria.* La Habana : Pueblo y Educación., 2004. p. 52.
3. **Dios, Manuel Alejandro, Torres, José Manuel Framiñán and Domínguez, Roberto.** *Modelado y análisis de un proceso quirúrgico mediante técnicas de minería.* Sevilla : s.n., 2010. p. 1.
4. **van der Aalst, W, Adriansyah, A and Medeiros,, A.K.A.** *Manifiesto sobre Minería de Procesos.* s.l. : IEEE, 2011.
5. **Mans, R. S., et al.** *Discovering Simulation Models. Information Systems.* 2009.
6. **Hernández, Nariño Arialys.** *Contribución a la gestión y mejora de procesos en instalaciones hospitalarias del territorio matancero.* Matanzas : s.n., 2010. p. 4.
7. **Beltrán Sanz, Jaime, et al.** *Guía para una gestión basada en procesos .* 2001.
8. **Brunnelo, Miguel and Marcelo Rocha.** http://e-conomicas.eco.unc.edu.ar/archivos/_2/U3-ModProc-11.pdf. *Modelado de procesos.* [Online] 2011.
9. **Bratosin, C., Sidorova, N. and Van Der Aalst, W.** *Distributed genetic process mining using sampling.* 2011.
10. **Günther, C. W. and Van der Aalst, W. M. P.** *Fuzzy Mining: Adaptive Process Simplification Based on Multi-Perspective Metrics.* 2007.
11. **Weijters, A.J.M.M and Ribeiro, J.T.S.** *HeuristicsMiner 6.0: Users Guide.* The Netherlands : s.n., 2011.

12. **Weijters, A.J.M.M., van der Aalst, W.M.P. and Alves de Medeiros, A.K.** . *Process Mining with the HeuristicsMiner*. The Netherlands. : Department of Technology Management, Eindhoven University of Technology, 2006.
13. **Chamorro, Maria.** *Método para alicar minería de procesos a la distribución de bebestibles no alcohólicos*. Santiago de Chile : s.n., 2013.
14. **ARIS Process Performance Manager** . Software. [Online] 2011. [Cited: diciembre 1, 2014.] http://www.softwareag.com/corporate/products/aris_platform/aris_controlling/aris_process_performance/overview/default.asp.
15. **QPR.** QPR. [Online] 2011. [Cited: diciembre 2, 2014.] <http://www.qpr.com/>.
16. **Analyst, Discover.** Stereologic. [Online] 2011. [Cited: diciembre 2, 2014.] <http://www.stereologic.com/>.
17. **Van der Aalst, W.M.P. and Van der Alst, W.M.P.** "ProM: The Process Mining Toolkit," in *Proceedings of BPM*. Germany : s.n., 2009. Vol. 489.
18. **Beth Chrissis, Mary , Konrad, Mike and Shrum, Sandy** . *CMMI: Guía para la integración de procesos*. Madrid : s.n., 2009.
19. **Sánchez, Tamara Rodriguez.** *Metodología de desarrollo para la Actividad productiva UCI*. La Habana : s.n.
20. **Mendoza, Vázquez Iván.** *Definición de un Framework para aplicaciones Web con navegación sensible a concerns*. La Plata. Argentina : s.n., 2012. p. 45.
21. **García de Jalón, J.,, Rodriguez Iñigo, M. J., and Alfonso Brazález, A. I., L.** *Aprenda Java como si estuviera en primero*. España : s.n., 2000. p. 78.
22. **UML.** Unified Modeling Language™ (UML®) Resource Page. [Online] [Cited: mayo 7, 2015.] <http://www.uml.org/>.
23. **JSF.** <http://www.desarrolloweb.com>. *Java Server Faces, Primer artículo de un pequeño manual sobre esta tecnología*. [Online] s.f. [Cited: enero 16, 2015.] <http://www.desarrolloweb.com/articulos/2380.php..>

24. **Franky, C.** Researchgate. [Online] 2010. [Cited: enero 20, 2015.] http://www.researchgate.net/profile/Consuelo_Franky/publication/228753991_Java_EE_5_%28sucesor_de_J2EE%29_el_reto_de_volver_a_empezar/links/0fcfd50916764d54ee000000.pdf.
25. **API, JAVA J2EE. JPA - Java Persistence.** <http://luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>. *JAVA J2EE*. [Online] s.f. [Cited: enero 20, 2015.] <http://luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>.
26. **Hibernate.** Hibernate. [Online] s.f. [Cited: enero 20, 2015.] <http://hibernate.org/>.
27. **Liu, X.** Java.net. *Developing applications with Facelets, JSF and JS*. [Online] 2011. [Cited: enero 20, 2015.] <http://today.java.net/pub/a/today/2006/08/29/developing-with-facelets-jsf-jsp.html>.
28. **Lucifer, P.** Searchsoa. *Java Runtime Environment - JRE*. [Online] s.f. [Cited: enero 20, 2015.] <http://searchsoa.techtarget.com/definition/Java-Runtime-Environment>.
29. **Jboss.** Jboss. [Online] s.f. [Cited: 3 febrero, 2015.] <http://www.jboss.org/jbossas/docs/>.
30. **Hernandis, J. A.** Versión Cero. [Online] 2011. [Cited: febrero 4, 2015.] <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>.
31. **Rumbaugh, J., and Jacobson, I., Booch. G.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Pearson Education. 2007.
32. **Ganymede.** Eclipse Ganymede. [Online] s.f. [Cited: febrero 4, 2015.] <https://www.eclipse.org/ganymede/>.
33. **PostgreSQL.** PostgreSQL. [Online] [Cited: febrero 4, 2015.] <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.
34. **PgAdmin.** PgAdmin . [Online] [Cited: febrero 5, 2015.] <http://www.pgadmin.org/>.
35. **Visual Paradigm.** Visual Paradigm Essential . [Online] [Cited: mayo 7, 2015.] <https://www.udemy.com/visual-paradigm-essential/>.

36. **Jacobson, Ivar, Booch , Grady and Rumbaugh, James .** *El Proceso Unificado de Desarrollo de Software.* España : s.n., 2000.
37. **Castro Santiesteban, Roberto Amaury.** *Diseño e Implementación del módulo Entidades del plugins Génesis.* La Habana : s.n., 2011.
38. **MDN.** Microsoft Developer Network. ¿Qué es un Patrón? <http://msdn.microsoft.com>. [Online] [Cited: febreo 6, 2015.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx..>
39. **Hall, Larman - Prentice.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* s.f. pp. 38- 50.
40. **Fernández Llatas, Carlos .** *Representación, Interpretación y Aprendizaje de.* Universidad Politécnica de Valencia. España : s.n., 2007. p. 48.
41. **ma, Iula.** *How to Evaluate the Performance of Process Discovery Algorithms.* Department of Industrial Engineering & Innovation Sciences : s.n., 2012.
42. **Burattin, Andrea , Sperduti, Alessandro and van der Aalst, Wil M. P.** *Heuristics Miners for Streaming Event Data.* The Netherlands : s.n., 2012.
43. **P., Briol.** *BPMN, the Business Process Modeling Notation Pocket Handbook.* 2008. pp. 58-60.
44. **Oliva Romero and González García, Jandro.** *Procedimiento para la detección de eventualidades aplicando técnicas de minería de proceso en el Sistema de Información Hospitalaria xavia HIS.* La Habana : s.n., 2013.

Bibliografía

Analyst, Discover. Stereologic. [En línea] 2011. [Citado el: 2 de diciembre de 2014.] <http://www.stereologic.com/>.

API, JAVA J2EE. JPA - Java Persistence. <http://luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>. *JAVA J2EE*. [En línea] s.f. [Citado el: 20 de enero de 2015.] <http://luchorondon.blogspot.com/2009/04/jpa-java-persistence-api.html>.

ARIS Process Performance Manager . Software. [En línea] 2011. [Citado el: 1 de diciembre de 2014.] http://www.softwareag.com/corporate/products/aris_platform/aris_controlling/aris_process_performance/overview/default.asp.

Beltrán Sanz, Jaime, y otros. *Guía para una gestión basada en procesos* . 2001.
Bratosin, C., Sidorova, N. y Van Der Aalst, W. *Distributed genetic process mining using sampling*. 2011.

Brunello, Miguel y Marcelo Rocha. http://e-conomicas.eco.unc.edu.ar/archivos/_2/U3-ModProc-11.pdf. *Modelado de procesos*. [En línea] 2011.

Burattin, Andrea , Sperduti, Alessandro y van der Aalst, Wil M. P. *Heuristics Miners for Streaming Event Data*. The Netherlands : s.n., 2012.

Castillo Pino, Argelio and Díaz Fernández, Daniel. *Sistema para el tratamiento de procesos desestructurados*. UCI. La Habana : s.n., 2013. p. 6.

Castro Santiesteban, Roberto Amaury. *Diseño e Implementación del módulo Entidades del plugins Génesis*. La Habana : s.n., 2011.

Chamorro, Maria. *Método para alicar minería de procesos a la distribución de bebestibles no alcohólicos*. Santiago de Chile : s.n., 2013.

Dios, Manuel Alejandro, Torres, José Manuel Framiñán and Domínguez, Roberto. *Modelado y análisis de un proceso quirúrgico mediante técnicas de minería.* Sevilla : s.n., 2010. p. 1.

Diseño Arquitectónico. [En línea] [Citado el: 10 de febrero de 2015.] <http://disenoarquitectonicousta.blogspot.com/2014/09/quien-lo-realiza-aunque-un-ingeniero-de.html>.

Fernández Llatas, Carlos . *Representación, Interpretación y Aprendizaje de.* Universidad Politécnica de Valencia. España : s.n., 2007. pág. 48.

Fluxicon. [En línea] [Citado el: 19 de diciembre de 2014.] <http://fluxicon.com/blog/2011/05/process-mining-in-healthcare-case-study-no-1/>.

Franky, C. Researchgate. [Online] 2010. [Cited: enero 20, 2015.] http://www.researchgate.net/profile/Consuelo_Franky/publication/228753991_Java_EE_5_%28sucesor_de_J2EE%29_el_reto_de_volver_a_empezar/links/0fcfd50916764d54ee000000.pdf.

Ganymede. Eclipse Ganymede. [En línea] s.f. [Citado el: 4 de febrero de 2015.] <https://www.eclipse.org/ganymede/>.

García de Jalón, J.,, Rodríguez Iñigo, M. J., y Alfonso Brazález, A. I., L. *Aprenda Java como si estuviera en primero.* España : s.n., 2000. pág. 78.

Günther, C. W. y Van der Aalst, W. M. P. *Fuzzy Mining: Adaptive Process Simplification Based on Multi-Perspective Metrics.* 2007.

Hall, Larman - Prentice. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* s.f. págs. 38- 50.

Hernández, Nariño Arialys. *Contribución a la gestión y mejora de procesos en instalaciones hospitalarias del territorio matancero.* Matanzas : s.n., 2010. pág. 4.

Hernandis, J. A. Versión Cero. [En línea] 2011. [Citado el: 4 de febrero de 2015.] <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>.

Hibernate. Hibernate. [En línea] s.f. [Citado el: 20 de enero de 2015.] <http://hibernate.org/>.

JSF. <http://www.desarrolloweb.com>. *Java Server Faces, Primer artículo de un pequeño manual sobre esta tecnología.* [En línea] s.f. [Citado el: 16 de enero de 2015.] <http://www.desarrolloweb.com/articulos/2380.php>.

Jboss. Jboss. [En línea] s.f. [Citado el: febrero de 3 de 2015.] <http://www.jboss.org/jbossas/docs/>.

Jacobson, Ivar, Booch , Grady and Rumbaugh, James . *El Proceso Unificado de Desarrollo de Software.* España : s.n., 2000.

Liu, X. Java.net. *Developing applications with Facelets, JSF and JS.* [En línea] 2011. [Citado el: 20 de enero de 2015.] <http://today.java.net/pub/a/today/2006/08/29/developing-with-facelets-jsf-jsp.html>.

Lucifer, P. Searchsoa. *Java Runtime Environment - JRE.* [En línea] s.f. [Citado el: 20 de enero de 2015.] <http://searchsoa.techtarget.com/definition/Java-Runtime-Environment>.

Mans, R. S., y otros. *Discovering Simulation Models. Information Systems.* 2009.

ma, Iula. *How to Evaluate the Performance of Process Discovery Algorithms.* Department of Industrial Engineering & Innovation Sciences : s.n., 2012.

MDN. Microsoft Developer Network. ¿Qué es un Patrón? <http://msdn.microsoft.com>. [En línea] [Citado el: 6 de febrero de 2015.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Mendoza, Vázquez Iván. *Definición de un Framework para aplicaciones Web con navegación sensible a concerns.* La Plata. Argentina : s.n., 2012. pág. 45.

Modelado del análisis. <https://inge2software.files.wordpress.com>. [En línea] [Citado el: 15 de febrero de 2015.] <https://inge2software.files.wordpress.com/2012/.../modelado-de-analisis>.

Nogueira. *Fundamentos para el Control de Gestion empresaria.* La Habana : Pueblo y Educación., 2004. p. 52.

Oliva Romero y González García, Jandro. *Procedimiento para la detección de eventualidades aplicando técnicas de minería de proceso en el Sistema de Información Hospitalaria xavia HIS.* La Habana : s.n., 2013.

Orellana, Arturo y Corales Sanchez, Yovannys. *Minería de Procesos en salud. Caso de Estudio: modelado de los procesos del área de Emergencias.* La Habana : . Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014) "Excellence in Engineering To Enhance a Country's Productivity", 2014.

P., Briol. *BPMN, the Business Process Modeling Notation Pocket Handbook.* 2008. págs. 58-60.

Pérez, Alonso Damián y Yzquierdo Herrera, Raykenler. *Utilización de técnicas de minería de proceso en el entorno empresarial cubano.* La Habana, Cuba : s.n., 2013.

PgAdmin. PgAdmin . [En línea] [Citado el: 5 de febrero de 2015.] <http://www.pgadmin.org/>.

PostgreSQL. PostgreSQL. [En línea] [Citado el: 4 de febrero de 2015.] <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.

QPR. QPR. [En línea] 2011. [Citado el: 2 de diciembre de 2014.] <http://www.qpr.com/>.

Rumbaugh, J., y Jacobson, I., Booch. G. *El Lenguaje Unificado de Modelado. Manual de Referencia.* Pearson Education. 2007.

Sakipova, Dina Yaksilik Torres. *Algoritmos y técnicas de descubrimiento de procesos poco estructurados: estado del arte.* La Habana : s.n., 2014.

Van der Aalst, W, Adriansyah, A y Medeiros,, A.K.A. *Manifiesto sobre Minería de Procesos.* s.l. : IEEE, 2011.

Van der Aalst, W.M.P. and Van der Alst, W.M.P. "ProM: The Process Mining Toolkit," in *Proceedings of BPM.* Germany : s.n., 2009. Vol. 489.

Verbeek, H.M.W. . ProM 6 Tutorial. ProM. (n.d.). <http://www.promtool.org/prom6>. [En línea] 2010.

Weijters, A.J.M.M y Ribeiro, J.T.S. *HeuristicsMiner 6.0: Users Guide.* The Netherlands : s.n., 2011.

Weijters, A.J.M.M., van der Aalst, W.M.P. y Alves de Medeiros, A.K. . *Process Mining with the HeuristicsMiner*. The Netherlands. : Department of Technology Management, Eindhoven University of Technology, 2006.

Anexos

Anexo 1 Descripción de las Fases que propone Variación AUP-UCI para el ciclo de vida de los proyectos de la UCI.

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Anexo 2 Descripción de las Disciplinas que propone Variación AUP-UCI para el ciclo de vida de los proyectos de la UCI

Disciplinas AUP	Disciplinas Variación AUP-UCI	Objetivos Disciplinas (Variación AUP-UCI)
Modelo	Modelado de negocio (opcional)	El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito.
	Requisitos	El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
	Análisis y diseño	En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos,

		<p>incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.</p>
Implementación	Implementación	<p>En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.</p>
Prueba	Pruebas interna	<p>En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.</p>
	Pruebas de liberación	<p>Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.</p>

	Pruebas de Aceptación	Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.
Despliegue	Despliegue (Opcional)	Constituye la instalación, configuración, adecuación, puesta en marcha de soluciones informáticas y entrenamiento al personal del cliente.
Gestión de configuración	Se cubren con las áreas de procesos Planeación de proyecto, Monitoreo y control de proyecto y Administración de la configuración, que propone CMMI-DEV v1.3. Las mismas son áreas de procesos de gestión y soporte respectivamente.	
Gestión de proyecto		
Entorno		

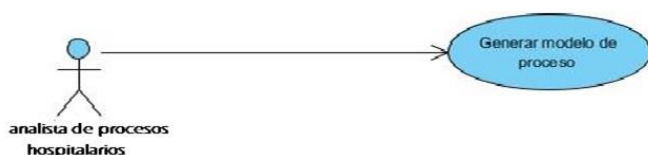
Anexo 3. Descripción de los Roles que propone Variación AUP-UCI para el ciclo de vida de los proyectos de la UCI.

Roles AUP	Roles Variación AUP-UCI
Administrador proyecto de	Jefe de proyecto
	Planificador
Ingeniero de procesos	Analista
Modelador ágil	Arquitecto de información (Opcional)
Desarrollador	Desarrollador
Administrador de configuración. la	Administrador de la configuración
Stakeholder	Stakeholder (Cliente/Proveedor de requisitos)
Administrador de pruebas	Administrador de calidad
Probador	Probador
Administrador de BD	Arquitecto de software (Sistema)
	Administrador de BD

Anexo 4 Especificación del requisito funcional: Generar modelo de procesos.

Nº	Nombre	Descripción	Prioridad para el cliente	Complejidad	Referencias cruzadas
1	Generar modelo de proceso	Permite obtener un modelo de proceso a partir de configurar los parámetros requeridos y seleccionar el tipo de análisis.	Alta	Alta	Ver: CESIM_SIGEC_Especificacion_de_casos_de_uso_Generar modelo de proceso. CU 1

Anexo 5 Diagrama de caso de uso



Anexo 6 Especificación del caso de uso: Generar modelo de proceso

Objetivo	Permite generar un modelo de proceso.
Actores	Analista de procesos hospitalarios.
Resumen	El caso de uso inicia cuando el actor accede a la opción Generar modelo de proceso, el sistema brinda la posibilidad de introducir los criterios para generar el modelo, el actor introduce los datos que considera como criterios para generar el modelo, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes genera el modelo de proceso. El caso de uso finaliza cuando queda generado el modelo de proceso.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Que existan datos de los eventos almacenados en el sistema.
Postcondiciones	Queda generado el modelo de procesos.

Flujo de eventos

Flujo básico

	Actor	Sistema
	El caso de uso inicia cuando se accede a la opción Generar modelo de proceso	
		Brinda la posibilidad de introducir los criterios para generar el modelo: <ul style="list-style-type: none"> • Procesos • Tipo de análisis • Desde • Hasta y permite: <ul style="list-style-type: none"> • Generar modelo de procesos Cancelar operación. Ver Evento 1 : "Cancelar operación."
	Introduce o selecciona los datos que considera como criterios para generar el modelo y selecciona la opción de Generar modelos	
		<ul style="list-style-type: none"> • Si el usuario deja valores obligatorios vacíos, ver Evento 2: "Existen campos obligatorios vacíos" • Si al buscar información para generar el modelo el sistema no encuentra ninguna, ver Evento 3: "No se encontró información."
		Genera el modelo a partir de los criterios seleccionados. Y permite: Exportar el modelo. Ver Evento 4 : "Exportar modelo de proceso". Ampliar el modelo. Ver Evento 5 : "Ampliar modelo de proceso".
		Termina el caso de uso

Flujos alternos

Evento 1. "Cancelar operación"

	Actor	Sistema
	Selecciona la opción de Cancelar operación.	
2.		Regresa a la vista anterior.
3.		Termina el caso de uso.

Evento 2. "Existen datos incompletos"

		Muestra un indicador sobre los campos incompletos.
		Regresa al paso 2 del Flujo Normal de Eventos .

Evento 3. "No se encontró información"

1.		Muestra el mensaje de información "No se encontró información que cumpla con los parámetros seleccionados."
2.		Regresa al paso 2 del Flujo Normal de Eventos .

Evento 4. "Exportar modelo de proceso"

	Selecciona la opción de Exportar	
		Exporta los modelos

Evento 5: "Ampliar modelo de proceso"

	Selecciona la opción de Ampliar	
		Amplia el modelo en otra pestaña
1.		Termina el caso de uso

Relaciones	CU incluidos	No existen
-------------------	---------------------	------------

	CU extendidos	
--	----------------------	--