



**Universidad de las Ciencias Informáticas**

**Facultad 3**

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

**TÍTULO: COMPONENTE PARA LA EXTRACCIÓN DE  
REGISTROS DE EVENTOS EN FORMATO XES DEL SISTEMA  
DE PLANIFICACIÓN DE ACTIVIDADES**

**Autor:** Leyriel Zurita González

**Tutor(es):** Ing. Maria Teresa Rosales González

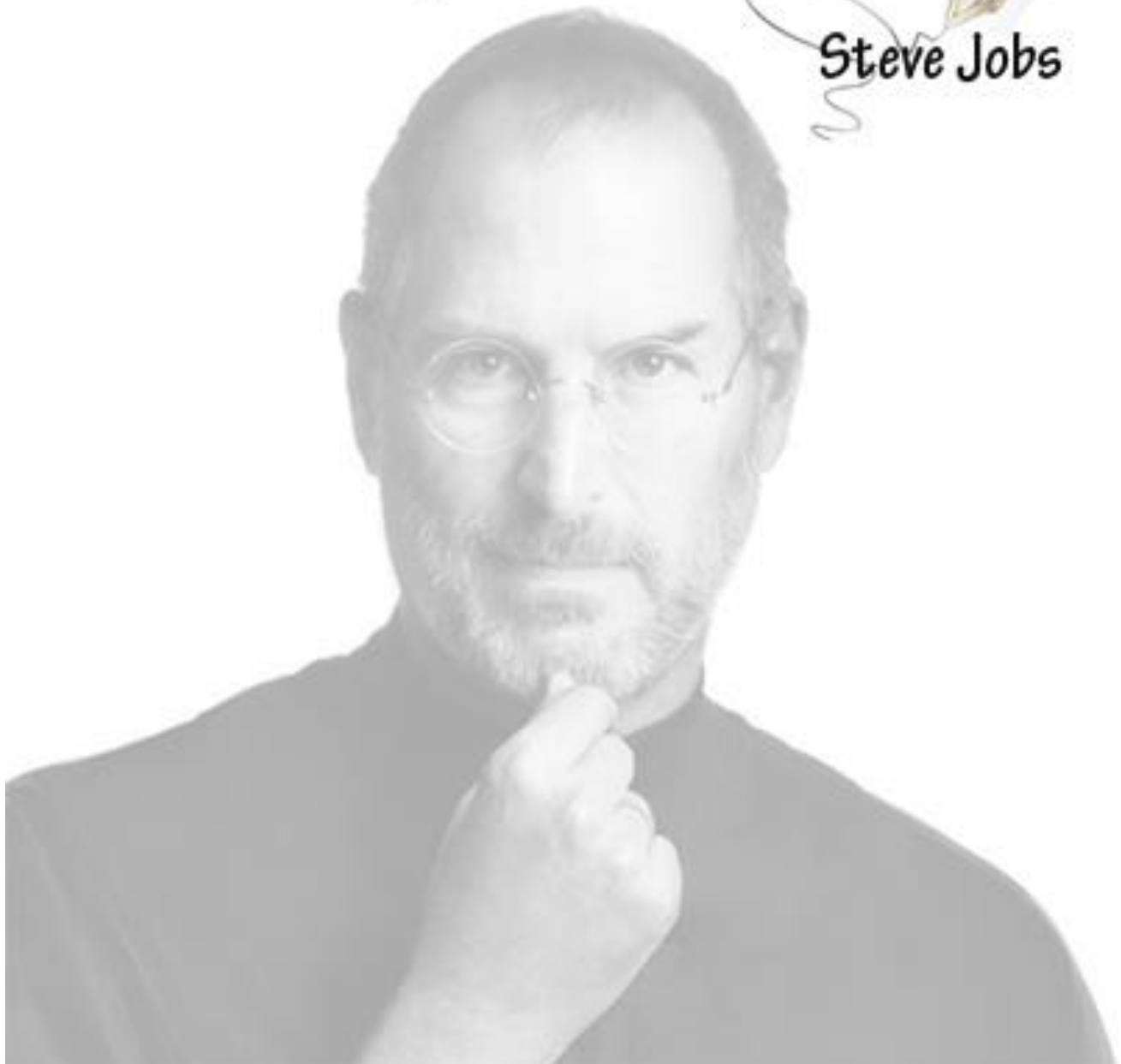
Ing. Abraham Calas Torres

Junio del 2016



**“Estoy convencido de que la mitad de lo que separa a los emprendedores exitosos de los no exitosos... es la perseverancia”**

**Steve Jobs**



## **DECLARACIÓN DE AUTORÍA**

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Leyriel Zurita González**

\_\_\_\_\_  
Firma del Autor

**Ing. María Teresa Rosales González**

\_\_\_\_\_  
Firma del Tutor

**Ing. Abraham Calás Torres**

\_\_\_\_\_  
Firma del Tutor

## DATOS DE CONTACTO

**Autor:** Leyriel Zurita González

**Edad:** 23 años

**Institución:** Universidad de las Ciencias Informáticas

**Correo:** [lzurita@estudiantes.uci.cu](mailto:lzurita@estudiantes.uci.cu)

**Currículo:** Estudiante de la Facultad 3 de la Universidad de las Ciencias Informáticas.

**Tutor:** Maria Teresa Rosales González

**Edad:** 26 años

**Institución:** Universidad de las Ciencias Informáticas

**Título:** Ingeniero en Ciencias Informáticas

**Correo:** [mtrosales@uci.cu](mailto:mtrosales@uci.cu)

**Currículo:** Ingeniero en Ciencias Informáticas, graduado en 2012 en la Universidad de las Ciencias Informáticas. Labora desde hace 4 años en el Centro de Informatización y la Gestión de Entidades en el proyecto Sistema de Planificación de Actividades, se desempeña como especialista de SIPAC en la Universidad de Ciencias Informáticas.

**Tutor:** Abraham Calás Torres

**Edad:** 27 años

**Institución:** Universidad de las Ciencias Informáticas

**Título:** Ingeniero en Ciencias Informáticas

**Correo:** [acalas@uci.cu](mailto:acalas@uci.cu)

**Currículo:** Ingeniero en Ciencias Informáticas, graduado en 2011 en la Universidad de las Ciencias Informáticas. Labora desde hace 5 años en el Centro de Informatización y la Gestión de Entidades en el departamento de Desarrollo de Componente, se desempeña con el rol de programador en la Universidad de Ciencias Informáticas.

## DEDICATORIA

*Dedico mi Trabajo de Diploma:*

*A mi Mamá por ser la persona más importante en mi vida, y darme tanto amor y cariño todo este tiempo, sin importar nada siempre estuviste para mí.*

*Gracias mamá.*

*A mi hermano por estar junto a mi mamá en mi ausencia. Muchas Gracias*

## **AGRADECIMIENTOS**

*Durante estos 5 años la Universidad me ha dado experiencias inolvidables, pero también personas que han hecho y estado en los momentos más felices y difíciles de mi carrera, por eso hoy quisiera agradecerles, primeramente:*

*A mi Mamá, por haberme dado la vida y estar siempre cuando lo necesite a pesar de todo, teníamos un sueño en común y hoy lo estamos haciendo realidad, ya soy Ingeniero mamá, Te Quiero... y Gracias.*

*A mi hermanito, por estar con mi mamá mientras estaba lejos, aunque dando dolores de cabeza.*

*A mi tutora María Teresa por toda la ayuda, el tiempo dedicado y la paciencia, y a mi tutor Abraham por toda la ayuda.*

*A mis primas Yeisy y Mariela por toda la ayuda que me dieron durante este tiempo y darme aliento para seguir adelante, también a la alegría de la familia mi pequeña niña LizLeisy.*

*A mis Primos Dayana, Papito y Dayami por haberme ayudado en estos años.*

*A mis tías China, Yamile, María Luisa por haberme ayudado e incitado a que siguiera adelante.*

*A Amauri, por haber estado ahí desde pequeño y haberme educado como su hijo.*

*A mis 4 mejores amigas por estar siempre ahí cuando lo necesite, por aguantarme mis malcriadeces, Indira gracias por ayudarme siempre, por estar a mi lado, aunque discutiéramos, estábamos el uno para el otro, a Maridely por ser siempre la más sensata de nosotros, tener siempre las cosas claras y darme buenos consejos, a Elais por ser tan fina y divina, a Nerelys que te voy a decir, por el cariño, la amistad, los consejos, dedicarme tu tiempo y ayudarme en todo, a las 4 las quiero mucho.*

*A mis otras tres chicas Arianna, Verónica y Yanía, por haber estado este año junto a mí, haberme dado consejo cuando los necesite, por ser tan geniales chicas.*



*A mis amigos Arlenís, Ricardo, Yení, Lesvy y Aidel por haberme dejado entrar en sus vidas y compartir con ustedes estos grandes momentos.*

*A Osmany por haber llegado en el momento más oportuno.*

*A Rene por haberme ayudado en estos tiempos.*

*A Lisbet y Karel Ernesto que, aunque no estén hoy aquí, fueron un apoyo, y por haberme enseñado tanto.*

*A mis compañeros y amigos de la Regional Nadíanis, Nailín, Yeneisy, Yusmaray, Mabel, Juani, Daniel Orlando, Yandith, Noslen, Barbaro, Maikel, Oreste, Daniel A, los jimaguas, que a pesar de todo nunca me dieron la espalda y a mi grupo en general, a todos los que pasaron por ahí.*

*A Dunia, Yuneisy, Jordania, Grisel y Aismara por haberme educado tanto en estos años, por los consejos, les estoy agradecido.*

*A los muchachos del Consejo de la FEU de la UCI por haberme dejado formar parte de ese grupo de dirigentes estudiantiles.*

*A los chicos de MegaDance y Kumbiamba mis dos grupos de baile de la Universidad.*

*A la Facultad Regional de las Ciencias Informáticas en Ciego de Ávila por haberme educado durante cuatro años y haberme formado como Ingeniero.*

*A la Federación Estudiantil Universitaria por convertirme en una persona responsable, preparada y darme experiencias inolvidables.*

*A la Universidad de las Ciencias Informáticas por haberme acogido el último año de mi carrera y terminar mi formación.*

*¡MUCHAS GRACIAS A TODOS!*

## RESUMEN

En Cuba es sumamente importante lograr una planificación de actividades eficiente en concordancia con los recursos que se disponen. Esta planificación se elabora siguiendo lo establecido por la Instrucción No.1 del Presidente de los Consejos de Estado y de Ministros, para la Planificación de los objetivos y actividades a todos los niveles de dirección. Con el avance tecnológico se han ido desarrollando sistemas que permiten llevar a cabo una correcta planificación teniendo en cuenta las regulaciones existentes para este proceso. En la Universidad de las Ciencias Informáticas (UCI) se desarrolló el Sistema de Planificación de Actividades (SIPAC), el cual informatiza los procesos de planificación de actividades y objetivo. SIPAC genera elementos que no se registran para el posterior análisis de un proceso determinado. El objetivo de esta investigación es desarrollar un componente que permita extraer los registros de eventos y exportarlos a formato XES. El punto de partida para aplicar técnicas de minería de procesos es un registro de eventos, asumiendo que es posible registrar eventos secuencialmente. Para dar formato a un registro de eventos, se define el formato XES, basado en experiencias prácticas de MXML, cuyo principal propósito es ofrecer un formato de intercambio de registros de eventos entre herramientas y dominios de aplicaciones. Como resultado de la investigación se obtuvo un componente que permite extraer las trazas a formato XES del Sistema de Planificación de Actividades.

**Palabras Claves:** minería de procesos, planificación, procesos, trazas, XES.



## ÍNDICE GENERAL

<b>DECLARACIÓN DE AUTORÍA .....</b>	<b>III</b>
<b>DATOS DE CONTACTO .....</b>	<b>IV</b>
<b>DEDICATORIA .....</b>	<b>V</b>
<b>AGRADECIMIENTOS .....</b>	<b>V</b>
<b>RESUMEN.....</b>	<b>1</b>
<b>ÍNDICE GENERAL .....</b>	<b>IX</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>XI</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>VIII</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1 Introducción .....	5
1.2 Conceptos asociados.....	5
1.3 Planificación de objetivo y actividades en Cuba.....	5
1.4 Sistema de planificación de actividades (SIPAC).....	6
1.5 Minería de procesos.....	7
1.6 Registros de eventos.....	9
1.7 Formato XES .....	12
1.7.1 Extensiones de XES .....	13
1.8 Desafíos en la extracción de registros de eventos.....	15
1.9 Estado del arte .....	17
1.9.1 Valoración del estado del arte .....	18
1.10 Metodología de desarrollo para la actividad productiva en la UCI (AUP-UCI).....	18
1.11 Tecnologías para el desarrollo de la solución.....	20
1.11.1 Herramientas para validación de la solución.....	24
1.12 Conclusiones parciales.....	25
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN .....</b>	<b>26</b>
2.1 Introducción .....	26
2.2 Propuesta de solución.....	26
2.3 Modelo de conceptual.....	26
2.4 Requisitos de software.....	27
2.4.1 Técnicas de captura de requisitos .....	28
2.4.2 Requisitos funcionales.....	28
2.4.3 Validación de los requisitos funcionales .....	30

2.4.4 Requisitos no funcionales.....	30
2.5 Modelo de datos.....	31
2.6 Tratamiento de los desafíos en la extracción de registros de eventos en SIPAC. ....	33
2.7 Modelo de diseño.....	34
2.7.1 Diagrama de clases de diseño.....	34
2.8 Patrones de diseño.....	37
2.8.1 Patrones de Diseño de Asignación de Responsabilidades (GRASP).....	37
2.8.2 Patrones de Comportamiento (GoF).....	39
2.9 Diseño de la solución en términos de componentes. ....	39
2.10 Conclusiones parciales.....	40
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN.....</b>	<b>41</b>
3.1 Introducción.....	41
3.2 Normas y estándares de codificación .....	41
2.9.1 Nomencladores de las Clases .....	41
2.9.2 Nomencladores de las funciones.....	42
2.9.3 Nomencladores de las variables.....	42
2.9.4 Nomencladores de los comentarios.....	43
3.3 Funciones <i>trigger</i> .....	43
3.4 Solución para la extracción de registros de eventos en formato XES.....	45
3.5 Métricas de software .....	46
3.5.1 Tamaño operacional de clase (TOC).....	47
3.5.2 Relaciones entre clases (RC) .....	49
3.6 Pruebas de software .....	51
3.6.1 Método de caja blanca.....	52
3.6.2 Método de caja negra .....	54
3.7 Pruebas de aceptación .....	58
3.8 Pruebas de rendimiento.....	59
3.9 Validación del fichero XES mediante un caso de estudio.....	62
3.10 Impacto de la solución .....	62
3.11 Conclusiones parciales.....	63
<b>CONCLUSIONES .....</b>	<b>64</b>
<b>RECOMENDACIONES .....</b>	<b>65</b>
<b>REFERENCIAS.....</b>	<b>66</b>

## ÍNDICE DE TABLAS

TABLA 1. DESCRIPCIÓN DE LOS REQUISITOS FUNCIONALES.....	28
TABLA 2. HISTORIA DE USUARIO RF 5.GENERAR TRAZAS.....	29
TABLA 3. DESCRIPCIÓN DE LAS ENTIDADES DEL MODELO DE DATOS.....	32
TABLA 4. DESCRIPCIÓN DE LOS TRIGGER.....	43
TABLA 5. ATRIBUTOS DE CALIDAD EVALUADOS POR LA MÉTRICA TOC.....	47
TABLA 6. CRITERIOS DE EVALUACIÓN PARA LA MÉTRICA TOC.....	47
TABLA 7. INSTRUMENTO DE EVALUACIÓN DE LA MÉTRICA.....	48
TABLA 8. ATRIBUTOS DE CALIDAD EVALUADOS POR LA MÉTRICA RC.....	49
TABLA 9.CRITERIOS DE EVALUACIÓN PARA LA MÉTRICA RC.....	49
TABLA 10. INSTRUMENTO DE EVALUACIÓN DE LA MÉTRICA RC.....	50
TABLA 11. CASO DE PRUEBA PARA LA APLICACIÓN.....	55
TABLA 12. DESCRIPCIÓN DE LAS VARIABLES.....	56
TABLA 13. TABLA DE NO CONFORMIDADES DE LA APLICACIÓN Y EL DOCUMENTO.....	57
TABLA 14. CASO DE PRUEBA ESCENARIO 1.....	60
TABLA 15. CASO DE PRUEBA ESCENARIO 2.....	60
TABLA 16. RESULTADO DE LAS PRUEBAS ESCENARIO 1.....	61
TABLA 17. RESULTADOS DE LAS PRUEBAS ESCENARIO 2.....	61

## ÍNDICE DE FIGURAS

FIGURA 1. POSICIONAMIENTO DE LOS TRES TIPOS PRINCIPALES DE MINERÍA DE PROCESOS: DESCUBRIMIENTO, CONFORMIDAD Y MEJORAMIENTO. ....	8
FIGURA 2. RESUMEN DE UN REGISTRO DE EVENTOS.....	10
FIGURA 3. EJEMPLO DE UN REGISTRO DE EVENTOS.....	10
FIGURA 4. REGISTRO DE EVENTOS EN FORMATO XES.....	15
FIGURA 5. MODELO CONCEPTUAL.....	27
FIGURA 6. MODELO DE DATOS.....	31
FIGURA 7. DCD_GENERAR TRAZAS, CAPA DE PRESENTACIÓN Y NEGOCIO.....	34
FIGURA 8. DCD_GENERAR TRAZAS, CAPA DE ACCESO A DATOS.....	35
FIGURA 9. FUNCIONAMIENTO DE UNA APLICACIÓN CON MVC.....	36
FIGURA 10. EJEMPLO DE PATRÓN EXPERTO EN CLASE DATEVENTO.....	38
FIGURA 11. EJEMPLO DE PATRÓN CONTROLADOR EN CLASE CMPGESTIONARTRAZASCONTROLLER.....	38
FIGURA 12. EJEMPLO DE PATRÓN CREADOR EN CLASE CMPGESTIONARTRAZASMODEL.....	38
FIGURA 13. DIAGRAMA DE COMPONENTE.....	40
FIGURA 14. EJEMPLO DE NOMENCLADORES DE LAS FUNCIONES EN EL CONTROLADOR.....	42
FIGURA 15. EJEMPLO DE IDENTIFICADOR.....	42
FIGURA 16. EJEMPLO DE VARIABLE.....	43
FIGURA 17. EJEMPLO DE COMENTARIO.....	43
FIGURA 18. EJEMPLO DE FUNCIÓN TRIGGER: FT_INSERTARPLAN_CASOS(). ....	44
FIGURA 19. INTERFAZ PRINCIPAL.....	46
FIGURA 20. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC PARA EL ATRIBUTO RESPONSABILIDAD.....	48
FIGURA 21. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC PARA EL ATRIBUTO COMPLEJIDAD.....	48
FIGURA 22. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA TOC PARA EL ATRIBUTO REUTILIZACIÓN.....	49
FIGURA 23. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC PARA EL ATRIBUTO ACOPLAMIENTO.....	50
FIGURA 24. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC PARA EL ATRIBUTO COMPLEJIDAD DE MANTENIMIENTO.....	51
FIGURA 25. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC PARA EL ATRIBUTO CANTIDAD DE PRUEBAS .....	51
FIGURA 26. RESULTADOS DE LA EVALUACIÓN DE LA MÉTRICA RC PARA EL ATRIBUTO REUTILIZACIÓN.....	51
FIGURA 27. CÓDIGO FUENTE DE LA FUNCIONALIDAD DAMETRAZASPERSONALIZADAS.....	52
FIGURA 28. GRAFO DE FLUJO ASOCIADO A LA FUNCIONALIDAD DAMETRAZASPERSONALIZADAS.....	53
FIGURA 29. PORCIENTO DE NO CONFORMIDADES DE LA DOCUMENTACIÓN.....	58
FIGURA 30. PORCIENTO DE NO CONFORMIDADES DE LA APLICACIÓN.....	58
FIGURA 31. ÁRBOL DE CONFIGURACIÓN.....	59
FIGURA 32. CONFIGURACIÓN UTILIZADA EN JMeter con 200 HILOS.....	60
FIGURA 33. MODELO GENERADO A PARTIR DE LA EXTRACCIÓN DEL REGISTRO DEL PROCESO ELABORACIÓN DEL PLAN.....	62

## INTRODUCCIÓN

En Cuba el perfeccionamiento de la planificación de objetivos y actividades juega un rol estratégico en la institucionalización del país. Para ello, desde 2009 se comenzaron a dar los primeros pasos en la conceptualización de este proceso; definición de las principales políticas en las organizaciones, funciones y responsabilidades de los directivos en todos los niveles y la normalización de los documentos generados en este proceso, basándose en las mejores prácticas de la Dirección por objetivos y la Dirección estratégica.

El Sistema de Planificación de Actividades (SIPAC), desarrollado por la Universidad de las Ciencias informáticas UCI y el Grupo de Planificación de Actividades del Consejo de Ministros, informatiza la Instrucción no.1 del Presidente de los Consejos de Estado y de Ministros para la Planificación de los objetivos y actividades en los Órganos, Organismos de la Administración Central de Estado, Entidades nacionales y Administraciones locales del Poder Popular. SIPAC posibilita la gestión de las actividades a todos los niveles organizacionales, permite interrelacionar objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas principales en las entidades, apoya el sistema informativo de gobierno y centraliza la información referente a la planificación.

Con el fin de aprovechar la información que se registra en los sistemas informáticos para aprender sobre el comportamiento de los procesos, y de esta manera detectar fuentes de problemas y oportunidades, surge la disciplina de minería de procesos.

La minería de procesos toma los datos registrados por los sistemas informáticos de la empresa como un punto de partida, extrae las distintas variaciones del proceso, y automáticamente las convierte en una visualización entendible (Aalst, 2014). Como los datos en los que se basa la minería de procesos son los existentes en los sistemas que están en uso, se obtiene una visualización del proceso real.

Es importante tener en cuenta que para comenzar con la minería de procesos se necesita partir de un registro de eventos (*log* de eventos). Todas las técnicas de minería de procesos asumen que es posible registrar eventos secuencialmente tal que cada evento se refiera a una actividad y se relacione a un caso particular (Aalst, 2011).

Para dar forma a un registro de evento y poder aplicar técnicas de minería de procesos, en un principio se definió el estándar *Mining eXtensible Markup Language* (MXML), el cual se basa en *eXtensible Markup Language* (XML) para el intercambio de registro de eventos. Sin embargo, en el

2010 *eXtensible Event Stream* (XES) lo reemplaza como el nuevo formato para minería de procesos, con el propósito de ofrecer un formato de intercambio de registro de eventos entre herramientas y dominios de aplicación (Buijs, 2010).

Si se lleva este enfoque a SIPAC; las modificaciones a los elementos se registran en una tabla o entidad en la que se almacenan atributos como: el elemento que sufrió la modificación, el autor de la misma, el tipo de acción (si es una modificación o un cambio de estado), y la modificación como tal. Dicha tabla es utilizada posteriormente para crear las notificaciones que el sistema envía al usuario que creó el elemento, sin embargo, no se almacenan en el sistema los datos necesarios para completar el registro de eventos que analizan las herramientas destinadas a la minería de procesos y por lo tanto se dificulta el análisis del comportamiento de los procesos en SIPAC, y de esta manera la detección de fuentes de problemas y oportunidades.

La demora para realizar los análisis es otros de los factores que influye, debido a que no poseen una estructura basada en los procesos que ejecuta, tampoco posee un mecanismo que permita exportar los registros a diferentes formatos que estén estandarizados.

Ante la problemática planteada se deriva el siguiente **Problema de la investigación**: ¿Cómo garantizar la extracción de las trazas en el Sistema de Planificación de Actividades?

A partir del problema expuesto, se centra el estudio de la investigación en la extracción de trazas, lo cual constituye el **Objeto de estudio**. Acotando el **Campo de acción** a la extracción de trazas en el Sistema de Planificación de Actividades. Definiendo como **Objetivo general**, desarrollar una solución informática que permita la extracción de trazas en el Sistema de Planificación de Actividades.

Para satisfacer el Objetivo general se definieron los siguientes **Objetivos específicos**:

- Confeccionar el marco teórico para fundamentar la investigación.
- Realizar el análisis y diseño para obtener y describir la solución a implementar teniendo en cuenta las necesidades del cliente.
- Implementar una solución que responda a las necesidades del cliente.
- Validar la solución propuesta mediante pruebas de caja blanca y caja negra.

Para orientar la investigación se propusieron las siguientes **Tareas Investigativas**:

- Revisión bibliográfica acerca de la confección de los registros de trazas en la minería de procesos.

- Estudio del estado del arte de los sistemas que generan registros de eventos.
- Caracterización del formato XES y sus funcionalidades dentro de la minería de procesos.
- Selección de las herramientas a utilizar para dar solución al problema propuesto.
- Elaboración del informe de la investigación.
- Validación de la solución utilizando técnicas de caja blanca y caja negra.

Los **Métodos teóricos** que se utilizaron para darle cumplimiento a los objetivos planteados anteriormente fueron:

**Analítico-Sintético**, al identificar los conceptos y definiciones más importantes relacionados con la descripción de algunos documentos que permita generar luego, una propuesta adecuada a la situación planteada y la tecnología estudiada.

**Histórico-Lógico**, permitió llevar a cabo un estudio evolutivo y cronológico de las diferentes herramientas que permitan desarrollar la solución al fenómeno, sus características, ventajas, desventajas y la selección de la más adecuada.

Los **Empíricos** que se utilizaron fueron la **Observación** haciendo un estudio visible de cómo se registraban las trazas en SIPAC en tiempo real y observando si durante los registros ocurría alguna anomalía.

**Entrevista**, se utiliza como herramienta para detectar las necesidades del cliente a través del intercambio con él. Para ello se realizan una serie de encuentros con el personal de SIPAC para definir los problemas existentes.

El presente trabajo está estructurado en **tres capítulos**:

En el **capítulo I**, “Fundamentación teórica”, se hace referencia a los principales conceptos teóricos existentes dentro del objeto de estudio, se describe el entorno en el que se desarrolla la problemática, se analizan las posibles soluciones y las características de los sistemas similares. Se exponen herramientas y tecnologías utilizadas en el transcurso de la investigación.

En el **capítulo II**, “Análisis y diseño de la solución”, se lleva a la práctica los conocimientos adquiridos durante la revisión bibliográfica y el estudio teórico realizado. Se fundamenta la propuesta que cumple con los objetivos trazados, la que incluye el desarrollo de un componente que dé solución al objetivo de la investigación y se especifican sus principales características. También se brinda una



descripción general de la propuesta de solución y de cómo debe funcionar, se especifican los requisitos y se modela el negocio en general.

En el **capítulo III**, “Implementación y validación de la solución”, se describen los estándares de codificación utilizados en la nomenclatura del código, además del diseño del diagrama de componentes que guíe la implementación y se prueba la propuesta de solución mediante pruebas de caja blanca y caja negra.



## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En el presente capítulo se describen los conceptos asociados a la minería de procesos, la metodología de desarrollo a utilizar y los conceptos asociados al objeto de estudio. Se brinda una descripción del entorno en el cual se desarrolla la problemática y se analizan las posibles soluciones. Se realiza un estudio de las técnicas de minería de procesos, descubrimiento, conformidad y mejoramiento.

### 1.2 Conceptos asociados

Para lograr un mejor entendimiento de la investigación propuesta, es importante conocer primeramente los siguientes conceptos asociados:

Un **proceso** es un conjunto de actividades o eventos que se realizan o suceden (alternativa o simultáneamente) con un fin determinado. Este término tiene significados diferentes según la rama de la ciencia o la técnica en que se utilice (Sanz, 2010).

Una **Instancia de proceso**, en el ámbito de la minería de procesos, hace referencia a la entidad siendo ejecutada por el proceso que es analizado. Los **eventos o actividades** son pasos bien definidos en el proceso, son el conjunto de operaciones o tareas que se llevan a cabo para cumplir las metas de un proceso (Mining, 2011).

“La **planificación** es un proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos” (Cortiñas, 2010).

"Es una técnica para minimizar la incertidumbre y dar más consistencia al desempeño de la empresa" (Cortiñas, 2010).

### 1.3 Planificación de objetivo y actividades en Cuba.

La planificación en Cuba se regula mediante la Instrucción No.1 del Presidente de los Consejo de Estados y de Ministros, el cual establece el contenido y los procedimientos principales con vistas al proceso de planificación del Gobierno de la República de Cuba. Se aplica atendiendo a la estructura, las misiones y características propias, en los niveles de dirección de la Asamblea Nacional del Poder Popular, el Consejo de Estado, en todos los niveles organizacionales (Ruz, 2011).

La Instrucción No.1 tiene como propósito definir los objetivos de trabajo que aseguran el continuo desarrollo del país, con el empleo racional y eficiente de los recursos disponibles a corto, mediano y largo plazos, así como la medición y evaluación sistemática de estos. Lograr una adecuada organización y coordinación de todos los factores implicados, que intervienen en el proceso, en interés del cumplimiento de los objetivos de trabajo. Garantizar eficiencia y eficacia en la implementación de lo planificado y en la comprobación de su ejecución mediante las diferentes formas de control (Ruz, 2011).

#### **1.4 Sistema de planificación de actividades (SIPAC).**

El Sistema de Planificación de Actividades (SIPAC) se desarrolló por la Universidad de las Ciencias Informáticas y lo certifica el Grupo de Planificación de Actividades de la Secretaría del Consejo de Ministros de la República de Cuba, la cual se guía por la Instrucción no.1 del Presidente de los Consejos de Estado y de Ministros explicada en el epígrafe anterior. Con el objetivo de informatizar y homogeneizar el proceso de planificación a corto, mediano y largo plazo que se realiza en todos los niveles de dirección del país. SIPAC se encuentra basado en los principios de independencia tecnológica, implementando funcionalidades generales de los procesos asociados a la planificación por objetivos el sistema permite:

- Interrelacionar los objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento y cumplimiento de los objetivos y actividades en las entidades.
- Puntualizar las actividades que debe efectuar cada usuario, como parte de la planificación a corto plazo, posibilita una mayor coincidencia entre lo que aspira la dirección y lo que debe proponerse cada miembro de la organización.
- Gestiona los posibles involucrados que dirigen o ejecutan el proceso de planificación de objetivos y actividades.
- Compartimenta la información mediante la gestión de los permisos entre los diferentes niveles involucrados en la planificación, la gestión de los principales nomencladores
- Genera reportes como el plan de actividades anual, el plan de actividades mensual, plan de trabajo individual, puntualizaciones del plan mensual y el resumen de cumplimiento del plan de trabajo según lo establecido en la Instrucción No.1.
- Informatiza los procesos de Elaboración del plan, Aprobación-conciliación, Puntualización del plan, Ejecución y control del plan, Evaluación de los objetivos (CEIGE, 2015).

SIPAC ha sido desplegado en la mayoría de los Organismos de la Administración Central del Estado, 3 Órganos del Estado: Fiscalía General de la República (FGR), Asamblea nacional del poder popular (ANPP), Contraloría general de la república (CGR), Consejos de Administración Provinciales y Consejos de Administración Municipales de las provincias Pinar del Río y Artemisa, Comisión para la Implementación y Desarrollo de los Lineamientos y Secretaría del Consejo de Ministros. De igual forma en Oficina del historiador de la Habana (OHLH), Grupo azucarero AZCUBA, Instituto de planificación física, Tribunal supremo popular.

Desde mayo del 2013 se comienza a desplegar en la UCI. SIPAC se ha convertido además en una plataforma que le da visibilidad a la UCI como soporte de la industria cubana del software en el país.

SIPAC es una aplicación Web que utiliza como lenguaje de programación PHP, como servidor web Apache y servidor de base de datos PostgreSQL. Está desarrollado en el marco de trabajo SAUXE v1.0, quien utiliza ExtJS v2.2 para la visualización de la interfaz de usuario, ZendFramework para la lógica del negocio, y Doctrine para el acceso a la base de datos.

SIPAC no posee ningún mecanismo para analizar sus procesos. Una de las variantes para lograr este objetivo, es aplicar la minería de procesos mediante técnicas de descubrimiento, mejora y chequeo de conformidad. Para esto es necesario extraer las trazas de la ejecución real de los procesos y permitir que se exporten a formato XES. El cuál es el formato estándar definido que utilizan herramientas de minería de procesos como ProM y Disco.

### **1.5 Minería de procesos**

Los sistemas de información registran enormes cantidades de eventos, de los cuales muchos son guardados de forma no estructurada, lo que ocasiona pérdida de tiempo y esfuerzo en la extracción de los mismos (Aalst, y otros, 2011).

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. La idea de la minería de procesos es descubrir, monitorear y mejorar los procesos reales y no los procesos supuestos, a través de la extracción de conocimiento de los registros de eventos ampliamente disponibles en los actuales sistemas. La minería de procesos incluye el descubrimiento automático de procesos, el chequeo de conformidad basándose en el monitoreo de desviaciones al comparar el modelo y el registro de eventos, la minería de redes sociales y organizacionales, la construcción automática de modelos de simulación, la extensión de modelos, la

reparación de modelos, la predicción de casos, y las recomendaciones basadas en historia (Aalst, 2011).

El punto de partida de la minería de procesos es un registro de eventos. Todas las técnicas de minería de procesos asumen que es posible registrar los eventos secuencialmente tal que cada *log* se refiera a una actividad y se relacione a un caso. Los registros podrían almacenar información adicional acerca de las actividades que se ejecutan. De hecho, siempre que sea posible, las técnicas de minería de procesos usan información extra, tales como el recurso que ejecuta o inicia la actividad, la marca de tiempo, o elementos de datos registrados con el evento.

Los registros de eventos pueden ser utilizados para realizar tres tipos de minería de procesos (Figura 1), el primer tipo es el descubrimiento. Una técnica de descubrimiento toma un registro de eventos y produce un modelo sin usar ninguna información a-priori. El descubrimiento de procesos es la técnica más destacada. El segundo tipo es el chequeo de conformidad, aquí, se compara un modelo de proceso existente con un registro de eventos del mismo proceso. La verificación de conformidad puede ser usada para chequear si la realidad, tal como está almacenada en el registro de eventos, es equivalente al modelo y viceversa. El tercer tipo de minería de procesos es el mejoramiento, el cual pretende extender o mejorar un modelo de proceso existente usando la información acerca del proceso real almacenada en algún registro de eventos. Mientras la verificación de conformidad mide el alineamiento entre el modelo y la realidad, este tercer tipo de minería de procesos busca cambiar o extender el modelo a-priori.



**Figura 1.** Posicionamiento de los tres tipos principales de minería de procesos: descubrimiento, conformidad y mejoramiento.

**Fuente:** Manifiesto sobre Minería de Procesos, Autor: Wil van der Aalst y otros, IEEE Task Force on Process Mining.

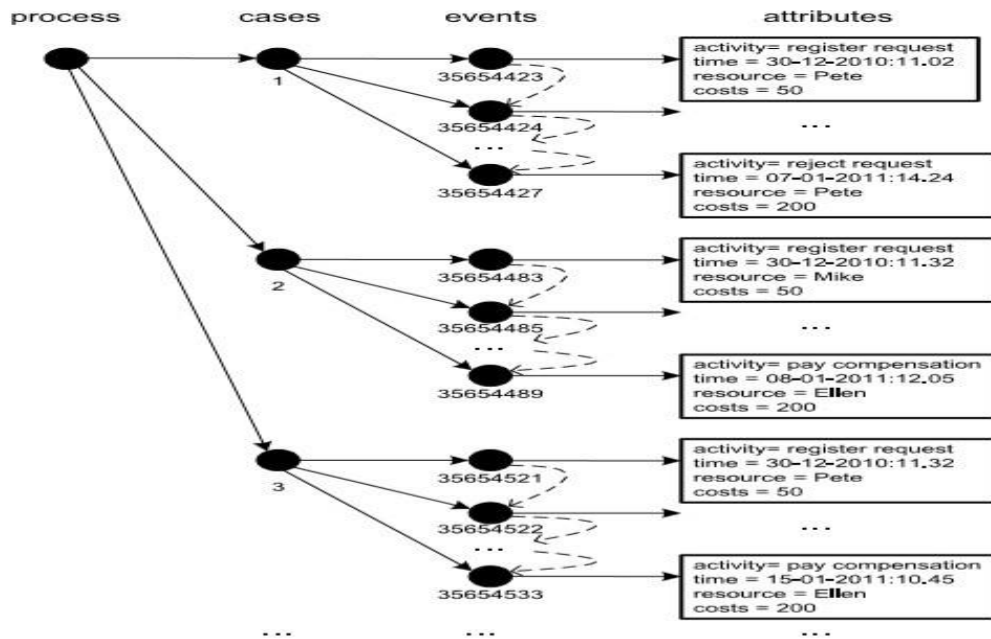
La minería de procesos es una disciplina impulsora del desarrollo de técnicas y herramientas para el análisis de los procesos de negocio que forman parte de los registros de eventos. Estos análisis se centran en brindar una mejor comprensión de los procesos para mejorar su rendimiento (Aalst, 2011).

Luego del estudio realizado sobre las técnicas de minería de procesos, la investigación estará enmarcada en la etapa de descubrimiento. Se desarrolla un procedimiento para detectar eventualidades creando un registro de eventos para descubrir modelos y lograr análisis de procesos en el Sistema de Planificación de Actividades. Esta técnica es la más destacada y utilizada en la actualidad porque permite descubrir cómo se ejecutan realmente los procesos en un sistema.

## **1.6 Registros de eventos**

La mayoría de los sistemas de información modernos utilizan mecanismos para registrar la ejecución real de los procesos, debido a que estos poseen un registro de trazas (Aalst, y otros, 2011). Las técnicas de minería de procesos asumen que es posible registrar eventos, donde cada proceso está compuesto por casos que no son más que instancias del mismo (Figura 2). Cada caso se compone de eventos que representan pasos bien definidos dentro del proceso; los casos contienen atributos o propiedades, donde los más usuales son la actividad que representan, la fecha y el usuario. Los registros podrían almacenar información adicional acerca de los eventos. De hecho, siempre que sea posible, las técnicas de minería de procesos usan información extra, tales como el recurso que ejecuta o inicia la actividad, la marca de tiempo del evento, o elementos de datos registrados con el evento.

Una instancia de un proceso, en el ámbito de la minería de procesos, hace referencia a la entidad siendo ejecutada por el proceso que es analizado. Los eventos se refieren a instancias del proceso. A su vez, un evento expresa el estado de una actividad para una instancia particular de un proceso, el inicio, conclusión o cancelación de una actividad (Aalst, 2011). El registro de eventos es la colección de eventos utilizados como entrada para la minería de procesos. Los eventos no necesitan ser almacenados en un archivo de registro por separado (i.e., los eventos pueden estar dispersos en diferentes tablas de bases de datos).



**Figura 2.** Resumen de un registro de eventos.

**Fuente:** Wild M. P. van der Aalst, Procces Mining, Dosccovery, Conformance and Enhancement of Business Procces.

Los casos y cada evento en un registro utilizan un identificador mientras que cada atributo de los eventos puede ayudar a extender el modelo con información extra. En la figura 3 se expone un ejemplo de un registro de eventos donde se puede observar los identificadores y los atributos de los eventos.

Case id	Event id	Properties Timestamp	Activity	Resource	Cost	...
1	35654423	30-12-2010:11.02	Register request	Pete	50	...
	35654424	31-12-2010:10.06	Examine thoroughly	Sue	400	...
	35654425	05-01-2011:15.12	Check ticket	Mike	100	...
	35654426	06-01-2011:11.18	Decide	Sara	200	...
	35654427	07-01-2011:14.24	Reject request	Pete	200	...
2	35654483	30-12-2010:11.32	Register request	Mike	50	...
	35654485	30-12-2010:12.12	Check ticket	Mike	100	...
	35654487	30-12-2010:14.16	Examine casually	Pete	400	...
	35654488	05-01-2011:11.22	Decide	Sara	200	...
	35654489	08-01-2011:12.05	Pay compensation	Ellen	200	...
3	35654521	30-12-2010:14.32	Register request	Pete	50	...
	35654522	30-12-2010:15.06	Examine casually	Mike	400	...
	35654524	30-12-2010:16.34	Check ticket	Ellen	100	...
	35654525	06-01-2011:09.18	Decide	Sara	200	...
	35654526	06-01-2011:12.18	Reinitiate request	Sara	200	...
	35654527	06-01-2011:13.06	Examine thoroughly	Sean	400	...
	35654530	08-01-2011:11.43	Check ticket	Pete	100	...
	35654531	09-01-2011:09.55	Decide	Sara	200	...
35654533	15-01-2011:10.45	Pay compensation	Ellen	200	...	
4	35654641	06-01-2011:15.02	Register request	Pete	50	...
	35654643	07-01-2011:12.06	Check ticket	Mike	100	...
	35654644	08-01-2011:14.43	Examine thoroughly	Sean	400	...
	35654645	09-01-2011:12.02	Decide	Sara	200	...
	35654647	12-01-2011:15.44	Reject request	Ellen	200	...
...	...	...	...	...	...	...

**Figura 3.** Ejemplo de un registro de eventos.

**Fuente:** Wild M. P. van der Aalst, Procces Mining, Discovery, Conformance and Enhancement of Business Process.

Para formalizar la estructura de los registros de eventos a utilizar en la minería de procesos uno de los estándares definido es: *eXtensible Event Stream* (XES). XES tiene como principal propósito ofrecer un formato de intercambio de registros de eventos entre herramientas y dominios (Gunther, 2009).

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 define porque los registros de eventos deberían ser completos, o sea, dado un determinado contexto no puede faltar ningún evento. Los datos de eventos son **seguros** si se tienen en cuenta consideraciones de privacidad y seguridad al registrar los eventos (Gunther, 2009). Los registros de eventos que se van a generar a partir de los datos almacenados en SIPAC, deben cumplir con estas variables de calidad, permitiendo que sean registros con información confiable, completa y segura a la hora de utilizarlo en diferentes análisis.

### **Niveles de madurez**

La combinación de los tres aspectos de calidad se refleja en los **niveles de madurez** definidos por la *IEEE Task Force On Process Mining*, 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 completo) 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 (IEEE Task Force On Process Mining, 2011).

## 1.7 Formato XES

El formato XES es un estándar XML para los registros de eventos. El estándar ha sido adoptado por la IEEE Fuerza de Trabajo de la Minería de procesos, como el formato de intercambio de registros de eventos por defecto. Una vez obtenido el registro de eventos en formato XES, se debería ser capaz de analizar este registro de todas las maneras posibles en el ámbito de la minería de procesos (Buijs, 2010). Los datos requeridos por el formato XES son, la marca de tiempo, identificador, nombre de la actividad y recurso.

Los registros de eventos, como ocurren en la práctica, pueden tomar una cantidad de formas diferentes y ejemplificaciones. Cada arquitectura de sistema que incluye algún tipo de mecanismo de registro ha desarrollado hasta ahora su propia solución. XES es un estándar y su propósito es proporcionar un formato reconocido generalmente para el intercambio de datos de registro de eventos entre las herramientas y los dominios de aplicación. Su objetivo principal es para la minería de procesos, es decir, el análisis de los procesos operativos basados en sus registros de sucesos. Sin embargo, XES ha sido diseñado para ser también adecuado para la minería en general de datos, minería de texto, y el análisis estadístico. En el diseño de la norma XES, los siguientes objetivos se han utilizado como principios rectores.

- **Simplicidad:**

Usar la forma más sencilla posible para representar la información. Los registros XES deben ser fáciles de analizar y generar, y deben ser igualmente legibles.

- **Flexibilidad:**

El estándar XES debe ser capaz de capturar los registros de eventos de cualquier plano, no importa el dominio de aplicación o de soporte de los procesos que se observan. Por lo tanto, XES pretende mirar más allá de la minería de procesos y los procesos de negocios, y se esfuerza por ser un estándar general para los datos de registro de eventos.

- **Extensibilidad:**

Debe ser fácil extender el estándar en el futuro. La extensión del estándar debe ser lo más transparente posible, al tiempo que mantiene la compatibilidad hacia atrás y hacia adelante. En el



mismo sentido, debe ser posible extender el estándar para requisitos especiales, para dominios específicos de la aplicación, o para las implementaciones de herramientas específicas.

- **Expresividad:**

Mientras se busca un formato genérico, los registros de eventos serializados en XES deben contener la menor pérdida de información posible. Por lo tanto, todos los elementos de información deben ser fuertemente tipados, y este es un método genérico para adjuntar semánticas interpretadas por humanos (Standard, 2010).

### 1.7.1 Extensiones de XES

Para las trazas y eventos pueden ser definidos dos tipos de atributos. El primer tipo son esos atributos que pueden ser especificados para trazas y eventos. Algunos de estos atributos son requeridos por algoritmos de minería de procesos. La mayoría de estos atributos son definidos por las cinco extensiones estándar de XES. El segundo tipo de atributo son los atributos de datos. Los atributos de datos almacenan información adicional acerca del objeto al cual la traza se refiere o la actividad ejecutada. Algunos algoritmos requieren atributos específicos en el registro de eventos, pero no todos los atributos definidos por las extensiones estándar deben estar siempre presentes en el registro. Algunas veces la información requerida no se encuentra en la fuente de datos (Orellana, y otros, 2015).

La extensión de XES más importante es la **concept** la cual especifica un nombre para el registro de eventos, la traza y los eventos. Proporcionar nombres a cada elemento es fácil y muy informativo. Los nombres de las trazas deberán incluir algún identificador único. Los nombres de los eventos deben proveer el nombre de la actividad ejecutada representada por el evento. La extensión **concept** define un atributo “*instance*” para eventos. Este atributo representa un identificador de la instancia de la actividad la cual generó el evento. Incluyendo este atributo en el registro de eventos permite enlazar los eventos con los registros en la fuente de datos para futuros análisis.

Otra extensión importante es la **time**. Esta extensión especifica el atributo que representa el instante de ocurrencia del evento. Mediante la grabación del instante de tiempo de ocurrencia del evento, los eventos pueden ser ordenados. Además, esto permite realizar análisis de duración y desempeño. Es importante lograr recoger la mayor cantidad de información acerca del momento de la ejecución del evento, preferiblemente al nivel de segundos y en algunos casos de milisegundos. Especialmente en una base de datos operacional siendo consultada concurrentemente, muchos eventos pueden ser ejecutados dentro del mismo segundo, por esto es la necesidad de la precisión de milisegundos. Esto

es necesario porque es muy importante que los eventos tengan un único instante de tiempo dentro de la traza. Si poseen mayor precisión los instantes de tiempo que manejan los algoritmos mejoraran drásticamente en su ejecución.

Cuando los eventos tienen nombre y fecha (instante de tiempo), es importante reconocer que pueden ser de diferentes tipos. Los eventos son grabaciones atómicas y por eso no tienen duración. Como las actividades sí tienen duración, los eventos pueden ser de diferentes tipos, cada uno de estos grabando un diferente estado de una actividad. Los tipos de eventos son proporcionados por la extensión **lifecycle**. Dos de los tipos de eventos más comunes son el “iniciado” y “completado”. Estos tipos de eventos indican el inicio y la completitud respectivamente de una actividad. Contando con estos dos tipos de eventos nos permite estimar tiempo de procesamiento y tiempo de espera en un análisis de desempeño. En la mayoría de los casos se utiliza el tipo “iniciado” y “completado”. Si no existe información acerca del inicio y fin de una actividad se usa solamente el “completado” como tipo de evento. Otra dimensión del análisis de minería de procesos trata sobre la distribución del trabajo. Relacionando eventos, recursos o grupos de recursos, la distribución del trabajo entre las diferentes unidades puede ser visualizada. Además, las redes sociales pueden ser construidas para indicar qué actores trabajaron juntos en algunos casos entregando el trabajo, etc.

También se pueden detectar a un grupo de actores desempeñando tareas similares. Para lograr un descubrimiento de estas redes, el actor o grupo que ejecutó el evento debe ser registrado con cada evento. Lo anterior se define como la extensión **organizational**. Esta extensión define tres diferentes atributos para eventos. El atributo más comúnmente usado es el “*resource*” el cual recoge el nombre o identificador del actor que ejecutó el evento. Adicionalmente el rol del recurso se puede almacenar también con el atributo “*role*”. En añadidura también se puede almacenar el grupo al que el usuario pertenece en el atributo “*group*”. La decisión de cuáles de los atributos usar depende de la información disponible y del nivel de detalle deseado. En alguno de los casos son necesarios los análisis sobre el rol o el grupo (Gunther, 2009).

La última de las cinco extensiones estándares de XES es la **semantic**. Esta extensión agrega el atributo “*modelReference*” a todos los elementos en el registro de eventos. Dicho atributo se refiere a un modelo conceptual en una ontología (externa) la cual permite cambiar el nivel de detalle en el modelo de proceso mediante la expansión o el colapso de los nodos de eventos. Otro ejemplo, es usarlo para referirse a una ontología de usuarios en el atributo “*resource*” de la extensión **organizational** (Gunther, 2009).

Pueden existir otras extensiones y atributos que definan los atributos relacionados en un proceso. Las cinco extensiones discutidas anteriormente por otra parte, capturan los atributos más frecuentemente usados. Al menos una parte de los atributos definidos por estas extensiones, deben ser especificados en el registro de eventos para que la minería de procesos se pueda aplicar exitosamente.

A continuación, se muestra un ejemplo sencillo de un registro de eventos en formato XES.

```

- <log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7">
  <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
  <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
  <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
  <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
  <extension name="Semantic" prefix="semantic" uri="http://www.xes-standard.org/semantic.xesext"/>
- <global scope="trace">
  <string key="concept:name" value="__INVALID__"/>
</global>
- <global scope="event">
  <string key="concept:name" value="__INVALID__"/>
  <string key="lifecycle:transition" value="complete"/>
</global>
<classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
<classifier name="Event Name" keys="concept:name"/>
<classifier name="Resource" keys="org:resource"/>
<string key="source" value="ProcessLogGenerator"/>
<string key="concept:name" value="Process 3"/>
<string key="lifecycle:model" value="standard"/>
- <trace>
  <string key="concept:name" value="67"/>
  <string key="description" value="67"/>
  - <event>
    <string key="org:resource" value="Alain"/>
    <date key="time:timestamp" value="2010-04-20T15:31:00.000-04:00"/>
    <string key="concept:name" value="Tarea: Creación"/>
    <string key="description" value="Tarea: Creación"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
  - <event>
    <string key="org:resource" value="Alain"/>
    <date key="time:timestamp" value="2010-04-20T15:31:00.000-04:00"/>
    <string key="concept:name" value="Tarea: Creación"/>
    <string key="description" value="Tarea: Creación"/>
    <string key="lifecycle:transition" value="complete"/>
  </event>
</trace>
+ <trace></trace>
</log>

```

**Figura 4.** Registro de Eventos en Formato XES.

**Fuente:** Componente para la extracción de registros de eventos en formato XES del Sistema de Gestión de Ensayos Clínicos (SIGEC), Lilian Valdés Medina, Lian Jaime Ramayo Méndez

## 1.8 Desafíos en la extracción de registros de eventos

Para la extracción de registros de eventos es necesario tener presentes algunos de los inconvenientes o desafíos que pueden surgir durante la extracción de los datos. Por tanto se hace necesario realizar un estudio de dichos desafíos, con el objetivo de tenerlos en cuenta en la creación de los eventos en SIPAC.

### Desafío No.1: Correlación

La correlación de eventos es el proceso mediante el cual se busca asociar los eventos para encontrar información útil. Los eventos en un registro se agrupan por caso. Este simple requisito puede ser bastante difícil, ya que requiere la correlación de eventos, es decir, los eventos necesitan estar

relacionados entre sí. Al diseñar registros a partir de cero, es bastante fácil de hacer frente a este problema. Sin embargo, cuando se trata de una variedad de fuentes de datos o sistemas interconectados, son necesarios esfuerzos adicionales para correlacionar eventos (Aalst, 2011).

### **Desafío No.2: *Timestamps*<sup>1</sup>**

Los eventos se deben solicitar por casos (instancias de procesos). Al combinar datos de diferentes fuentes, en la mayoría de los casos tiene que depender de las marcas de tiempo para clasificar los eventos (en orden de aparición). Esto puede ser problemático debido a los múltiples relojes y diferencias en las grabaciones a la hora de registrar un evento. Por lo tanto, puede haber diferencias significativas entre el tiempo real de un evento que se lleva a cabo y su marca de tiempo en el registro. Como resultado, el orden de los eventos no es fiable, por ejemplo, causa y efecto pueden ser invertidas (Aalst, 2011).

### **Desafío No.3: Intervalo de tiempo**

Los casos pueden tener una vida útil que se extiende más allá del período grabado, por ejemplo, un caso se inició antes del comienzo del registro de eventos o se sigue ejecutando cuando la grabación se detuvo. Por lo tanto, es importante darse cuenta que los registros de eventos típicamente solo proporcionan una instancia de un proceso corriendo. Cuando la duración media de un caso es corto en comparación con la longitud de la grabación, lo mejor es resolver este problema mediante la eliminación de los casos incompletos. Teniendo cuáles son las actividades iniciales y finales en un evento, por lo que es fácil filtrar el registro y eliminar los que no estén en el rango de las actividades que marcan el inicio y fin (Aalst, 2011).

### **Desafío No.4: Alcance**

El cuarto problema es la determinación del alcance del registro de eventos. Los sistemas pueden tener miles de tablas con los datos relevantes para el negocio. Se necesita conocimiento del dominio para localizar los datos requeridos y poder crear los registros. El alcance depende de los datos disponibles y las preguntas que necesitan ser contestadas (Aalst, 2011).

### **Desafío No.5: Granularidad**

La granularidad representa el nivel de detalle al que se desea almacenar la información sobre el negocio que se esté analizando. Mientras mayor sea el nivel de detalle de los datos, se tendrán mayores posibilidades de análisis, ya que los mismos podrán ser resumidos. Es decir, los datos que

---

<sup>1</sup> *Timestamp* es una secuencia de caracteres, que denotan la hora y fecha en la cual ocurrió determinado evento.

posean granularidad fina podrán ser resumidos hasta obtener una granularidad media o gruesa (Aalst, 2011).

### **1.9 Estado del arte**

Se analizaron dos sistemas que poseen componentes que generan registros de eventos basados en sus procesos, además del mapeador XESame que genera registros a partir de bases de datos operacionales.

#### **Componente para la extracción de registros de eventos en formato XES del Sistema de Gestión de Ensayos Clínicos (SIGEC).**

Este componente tiene como objetivo contribuir al análisis de los procesos de ensayos clínicos en el Centro de Inmunología Molecular (CIM), a partir de la información almacenada en las trazas de procesos. El componente está destinado a los usuarios que tengan el rol de analistas de procesos. Está desarrollado bajo el lenguaje de programación Java. Es un componente para generar registros de eventos en el sistema antes mencionado, que permite aplicar técnicas de minería de procesos para la toma de decisiones sobre la ejecución de los procesos y disminuir los tiempos concebidos para su análisis. Este componente genera sus registros a partir de la información que posee en la base de datos, los extrae y permite aplicarle técnicas en herramientas como el ProM, los registros están bien definidos basándose en los procesos que genera el estudio de Ensayos Clínicos en SIGEC (Medina, y otros, 2015).

#### **Componente para la extracción de registros de eventos en formato XES del sistema ZUN Suite.**

El componente permite extraer de las bases de datos de los hoteles que utilizan el sistema de información ZUN Suite, datos para conformar un registro de eventos en formato XES para su posterior análisis. Este análisis está centrado en aplicar técnicas de minería de procesos que permitan encontrar deficiencias o anomalías en los procesos de gestión hotelera, además de brindar posibles soluciones a los problemas encontrados y lograr disminuir los tiempos de ejecución de estos procesos. El componente es capaz de extraer el registro de eventos, centralizar los errores, las posibles soluciones, filtrar el registro de eventos de acuerdo a diferentes parámetros como son temporadas altas o bajas, nacionalidad, fecha de inicio y fecha fin, entre otras, además brindar información específica relacionada con los procesos de gestión hotelera. (Corella, y otros, 2015)

#### **XESame**

XESame es un mapeador de XES que provee una manera genérica para extraer un registro de eventos desde una fuente de datos. Una de las potencialidades de XESame es su facilidad de uso, ya que no requiere habilidades de programación. XESame le permite a un experto en el dominio

especificar cómo el registro de eventos debe ser extraído de la base de datos operacional de un sistema de información determinado y convertido a XES o MXML. El sistema brinda varias interfaces para la definición de la conversión en las cuales se pueden definir los tipos de gestores de bases de datos a usar como son PostgreSQL, Mysql, etc. La herramienta permite establecer las relaciones entre las tablas de la base de datos y sus respectivos campos en el registro de eventos teniendo en cuenta las extensiones de XES definidas. Una vez obtenido el registro de eventos en formato XES, se debería ser capaz de analizar este registro de todas las maneras posibles en el ámbito de Minería de proceso (Buijs, 2010).

Usando XESame se pueden especificar cada evento y cada actividad de manera independiente. El registro, sus trazas y eventos poseen atributos para almacenar información. Los atributos pueden contener otros atributos anidados para almacenar información más detallada. Además de seleccionar cuáles de los atributos han de ser incluidos en el registro de eventos, también se pueden seleccionar qué trazas o eventos se han de incluir. Esto se puede lograr limitando el número de trazas a extraer y/o el número de eventos (Buijs, 2010).

### **1.9.1 Valoración del estado del arte**

Con la investigación realizada se obtuvo como resultado, que cada uno de los componentes o módulos estudiados transformaban y exportaban las trazas de forma diferente, teniendo en cuenta que basan su desarrollo en las tecnologías y herramientas de los sistemas donde se van a incluir. Los componentes para SIGEC y ZUNSuite están implementados en Java y generan sus registros a partir de los procesos de negocio en los sistemas donde se integran, además las extracciones que realizan no permiten ser configurables para otros procesos. XESame posee ciertas limitaciones, pues al estar desarrollada en Java cuando realiza la extracción de una base de datos con gran cantidad de información tardaría bastante tiempo, lo que genera problemas con las consultas SQL por la gran cantidad de información. Referente al estudio realizado se puede decir que los registros de eventos que exportan están estandarizados a formato XES, siendo este un formato que permite realizar análisis de los procesos que se van ejecutando en los diferentes sistemas. Todo lo antes mencionado evidencia la necesidad de incorporar a SIPAC un componente para la extracción de registros de eventos en formato XES.

## **1.10 Metodología de desarrollo para la actividad productiva en la UCI (AUP-UCI)**

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de 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 (Sánchez, 2015).

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, que es Ejecución y se agrega la fase de Cierre.

**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.

**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.

**Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Atendiendo a lo anteriormente planteado en la presente investigación se determinó que la fase en la cual se va a interactuar es la fase de ejecución, porque en esta se efectúan las actividades requeridas para desarrollar el software, se realiza todo el modelado de la solución para un mejor entendimiento y se elaboran los requisitos, la arquitectura del sistema, se lleva a cabo el desarrollo y se entrega la solución al cliente.

AUP en su variación UCI define 7 disciplinas las cuales son el modelado del negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación y las pruebas de aceptación, para el desarrollo de la solución propuesta se trabajará en las siguientes 5 disciplinas:

**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. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado de negocio.

**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, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.

**Implementación:** En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.

**Pruebas internas:** 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.

**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.

Por las características que presenta la investigación, en la cual no se va a modelar el negocio porque esta descrito en el expediente de proyecto de SIPAC 2.0, la variante a escoger para el desarrollo del producto es el Escenario 4 el cual plantea generar para la descripción de los requisitos Historias de usuario (HU).

## **1.11 Tecnologías para el desarrollo de la solución.**

### **Herramientas CASE**

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software. Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un potencial efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales (Anónimo, 2014).

### **Visual Paradigm 8.0**

Visual Paradigm es una herramienta UML libre y 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 ayuda 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. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Soporta UML versión 2.1, permite modelado colaborativo con CVS y Subversión, generación de código, ingeniería inversa, generación de bases de datos (transformación de diagramas entidad-relación en tablas de la base de datos), importación y exportación a ficheros XML, distribución automática de diagramas, entre otras características (Paradigm, 2012).

#### **Apache 2.4**

Apache 2.4 es una tecnología gratuita de código abierto compatible con muchos Sistemas Operativos. Tiene todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Tiene una alta configuración en la creación y gestión de registros de actividad. Apache permite la creación de ficheros de registro a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Apache es el servidor web por excelencia, su facilidad de configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Se ejecuta en gran cantidad de sistemas operativos, lo que lo hace prácticamente universal. Es altamente configurable de diseño modular por lo que resulta muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos para Apache que son adaptables a este, y están disponibles para su instalación cuando sean necesarios. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor y es posible configurarlo para que ejecute un determinado script cuando esto suceda (Apache, 2012).

#### **PostgreSQL 9.4**

Es un servidor de base de datos relacional, libre. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Como toda herramienta de software libre PostgreSQL tiene entre otras ventajas que cuenta con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y algo muy importante es que dicha herramienta es multiplataforma. Además de sus ofertas de soporte, cuenta con una importante comunidad de profesionales y entusiastas de PostgreSQL de los

que los centros de desarrollos pueden obtener beneficios y contribuir. Está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas (PostgreSQL, 2013).

## PHP 5.6

PHP (acrónimo de "PHP: Preprocesador de Hipertexto ") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP es un soporte para una gran cantidad de bases de datos. Al nivel más básico, PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP, también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. PHP es un lenguaje libre que puede ser utilizado en casi todos los sistemas operativos existentes (Group, 2016).

## Marco de Trabajo

Un framework o marco de trabajo es una estructura de apoyo definida a través de la cual un proyecto de software se puede organizar y desarrollar. Persigue acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo, como el uso de patrones (Framework, 2006).

**SAUXE 1.0:** Sauxe es el marco de trabajo que se empleará para el desarrollo de la solución que se propone. Fue desarrollado en la UCI como fruto del paradigma de independencia tecnológica por el que aboga el país. Este marco de trabajo, fusionado bajo tecnología totalmente libre (entre ellas PHP, PostgreSQL, Apache) posee el desarrollo de tecnologías propias basadas en otros marcos de trabajo como ZendFramework para el manejo de la lógica de negocio, Doctrine para el acceso a datos y ExtJS para la capa de presentación. Cuenta con una arquitectura en capas que a su vez presenta en su capa superior un MVC(Modelo Vista Controlador). Contiene un conjunto de componentes reutilizables que proveen la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo (Baryolo, 2011).

**ExtJS 2.2:** ExtJS es el marco de trabajo que se empleará para el desarrollo de la capa de presentación. Está basado completamente en la programación orientada a objeto. Cada objeto contiene: propiedades, métodos y eventos. Basa toda su funcionalidad en JavaScript a través de

bibliotecas. Así, en tiempo de ejecución, carga y crea todos los objetos HTML (*HyperText Markup Language* o Lenguaje Marcado de Hipertexto) a través del uso intenso de DOM (Modelo de Objetos de Documentos). Los datos son obtenidos con AJAX (*Asynchronous JavaScript And XML* o JavaScript asíncrono y XML) a través de XML. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos. Además, permite que exista un balance entre el Cliente – Servidor, posibilitando que la carga de procesamiento se distribuye, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo (Cuter, y otros, 2011).

**Zend Framework 1.11:** Se utilizará el marco de trabajo Zend Framework para el manejo de la lógica de negocio. Este marco de trabajo de código abierto brinda facilidades de uso y funcionalidades. Está diseñado para la versión 5 de PHP y posee buenas capacidades de ampliación. Proporciona un sistema de caché de forma que se puedan almacenar diferentes datos, así como los componentes que forman la infraestructura del patrón Modelo-Vista-Controlador. Consta de mecanismos de filtrado y validación de entradas de datos. Permite convertir estructuras de datos PHP a JSON y viceversa, para su utilización en aplicaciones AJAX y provee capacidades de búsqueda sobre documentos y contenidos (Framework, 2010).

**Doctrine 1.0:** Para la capa de acceso a datos se empleará Doctrine. Este es un sistema ORM (en inglés Object Relational Mapper) para PHP 5.2 o superior que incorpora una DBL (capa de abstracción a base de datos). Uno de sus rasgos importantes es la habilidad de escribir opcionalmente las preguntas de la base de datos orientada a objeto. Esto les proporciona una alternativa poderosa a diseñadores de SQL, manteniendo un máximo de flexibilidad sin requerir la duplicación del código innecesario. Además, exporta una base de datos existente a sus clases correspondientes y convierte clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos (Vesterinen, 2012).

### **PhpStorm 10**

JetBrains PhpStorm es una multi-plataforma para PHP desarrollado sobre la plataforma JetBrains IntelliJ IDEA, además es un IDE comercial, pero la comunidad de PHP en la UCI posee una licencia de aprendizaje para utilizarlo.

PhpStorm proporciona un editor para PHP, HTML y JavaScript, con el análisis de código en marcha, la prevención de errores y refactorizaciones automáticas para PHP y el código JavaScript. PhpStorm soporta código PHP 5.3, 5.4, 5.5 y 5.6 (proyectos modernos y heredados), incluyendo generadores,

co-rutinas, palabras claves, listas *foreach*, espacios de nombres, los cierres, los rasgos y la sintaxis de matrices. Incluye un editor de SQL en pleno derecho con resultados en consulta editables. PhpStorm se basa en IntelliJ IDEA, que está escrito en Java. Los usuarios pueden ampliar el IDE mediante la instalación de *plugins* creados para la plataforma de IntelliJ o escribir sus propios *plugins*. Todas las funciones disponibles en WebStorm se incluyen en PhpStorm, que añade soporte para PHP y bases de datos (JetBrains, 2000-2016).

### 1.11.1 Herramientas para validación de la solución

#### JMeter 2.4

Es un software de código abierto, creado 100% en Java destinado a carga funcional, comportamiento de prueba y medición del rendimiento. Originalmente fue diseñado para probar las aplicaciones Web, pero desde entonces se ha expandido a otras funciones de prueba.

JMeter puede ser utilizado para probar el rendimiento tanto en los recursos estáticos y dinámicos (archivos, Servlets, Perl, Java Objects, bases de datos y consultas, servidores FTP y mucho más). Puede ser utilizado para simular una carga pesada en un servidor, de red o un objeto para poner a prueba su resistencia o para analizar el rendimiento general en diferentes tipos de carga. Puede usarlo para hacer un análisis gráfico de rendimiento o para probar su servidor, script, comportamiento del objeto con cargas pesadas concurrentes.

Presenta un marco completo que permite el muestreo simultáneo de muchas discusiones y toma de muestras simultáneas de diferentes funciones de los grupos de hilos separados. Además un cuidadoso diseño de interfaz gráfica de usuario que permite una operación más rápida y más precisa en cuanto a tiempo (Rosales, 2013).

La herramienta JMeter fue seleccionada para realizar las pruebas de rendimiento al sistema, debido a la implementación de los *trigger* en la base datos.

#### ProM 6.5.1

ProM es la herramienta más usada en el campo de la minería de procesos. Es de software libre, multiplataforma y está diseñada para que en ella se puedan desarrollar y ejecutar algoritmos. Es un marco extensible que es compatible con una amplia variedad de técnicas de minería de procesos en forma de *plugins*. Es independiente de la plataforma, ya que es implementado en Java y puede ser descargado sin ningún costo. Está publicado bajo una licencia de código abierto (Technology, 2014).

La última versión del producto es la 6.4. Para el trabajo con la herramienta se utilizan como entrada un registro de eventos solamente o un registro de eventos y un modelo de proceso (Verbeek, y otros, 2010).

La herramienta ProM fue seleccionada para inspeccionar los registros de eventos obtenidos a partir del componente desarrollado, con el fin de validar la calidad de los registros de eventos.

### **1.12 Conclusiones parciales**

El análisis de la problemática y los conceptos necesarios para comprender el campo de acción de la investigación, permitió entender la necesidad del problema propuesto y la posible solución que se podía implementar.

El análisis de las herramientas de extracción de registros de eventos en formato XES demostró, que a pesar de su utilidad estas no pueden ser integradas a SIPAC, debido a que el sistema está desarrollado en SAUXE, bajo el lenguaje de programación PHP, y procesa grandes cantidades de información en su base de datos, por lo que es necesario desarrollar un componente que permita generar y exportar registros de eventos, que faciliten el análisis de los procesos en SIPAC.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

### 2.1 Introducción

En este capítulo se describe la propuesta de solución denominada Componente para la extracción de registros de eventos en formato XES del Sistema de Planificación de Actividades. Para lograr un mejor entendimiento del componente se presenta la propuesta de solución. Adicionalmente, se presenta el modelo conceptual. Se identifican y describen los requisitos funcionales y no funcionales. Se presenta el patrón arquitectónico y los patrones del diseño empleados en el desarrollo de la solución. Posteriormente se muestran los artefactos generados durante la fase de análisis y diseño.

### 2.2 Propuesta de solución.

SIPAC genera en su base de datos un histórico donde almacena un registro de acciones de cada usuario, esta información es insuficiente porque no se sabe a qué procesos pertenecen los registros en esta tabla y solamente se puede interpretar la acción que se realizó. No se almacena toda la información necesaria para realizar un posterior análisis de cómo se ejecutan los procesos en el sistema.

Los registros de eventos de los procesos son el tipo de traza que posee la información suficiente para realizar estudios utilizando técnicas de minería de procesos. La solución se enmarca en la configuración de registros de eventos por un usuario a partir de las trazas que genera SIPAC y luego poder exportarlas a formato XES. Por tanto, es importante extraer los procesos del sistema mediante registros que contengan la información suficiente y necesaria para realizar análisis de ellos. Estos procesos son Elaboración del plan, Aprobación y conciliación, Puntualización, Ejecución y control del plan, y Evaluación de los objetivos.

El componente constará con una interfaz que permitirá al usuario seleccionar el proceso del cual quiere extraer las trazas, luego las actividades que utilizará y por último los recursos que le interesan del proceso. Una vez almacenada toda la información necesaria sobre el proceso se procederá a petición del usuario a la generación y exportación del registro de eventos.

### 2.3 Modelo de conceptual.

El Modelo de dominio o Modelo conceptual es una representación visual de los conceptos u objetos que se manejan en el dominio del sistema. Se utiliza para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema. Es un medio para

comprender el sector de negocio al cual el sistema va a servir. Este modelo puede ser usado como punto de partida para el diseño del sistema. Los objetos o conceptos incluidos en el Modelo de dominio no describen clases u objetos del software; sino entidades o conceptos del mundo real que están asociados al problema en cuestión (Larman, 2005).

El presente modelo conceptual ilustra el funcionamiento del registro de trazas una vez implementado el componente para la extracción y transformación de trazas de eventos a formato XES.

SIPAC es un Sistema de Planificación de Actividades el cual ejecuta varios procesos. Los procesos de negocio poseen eventos asociados. Elaboración del plan, Aprobación y conciliación, Puntualización del plan, Ejecución y control, y Evaluación de los objetivos son procesos. Las trazas generadas contienen los eventos definidos en los procesos de negocio. Un proceso puede generar registros de eventos los cuales agrupan trazas de un proceso y son exportados al formato XES, mediante el uso de sus extensiones. El usuario es quien configura los procesos delimitando los datos a capturar.

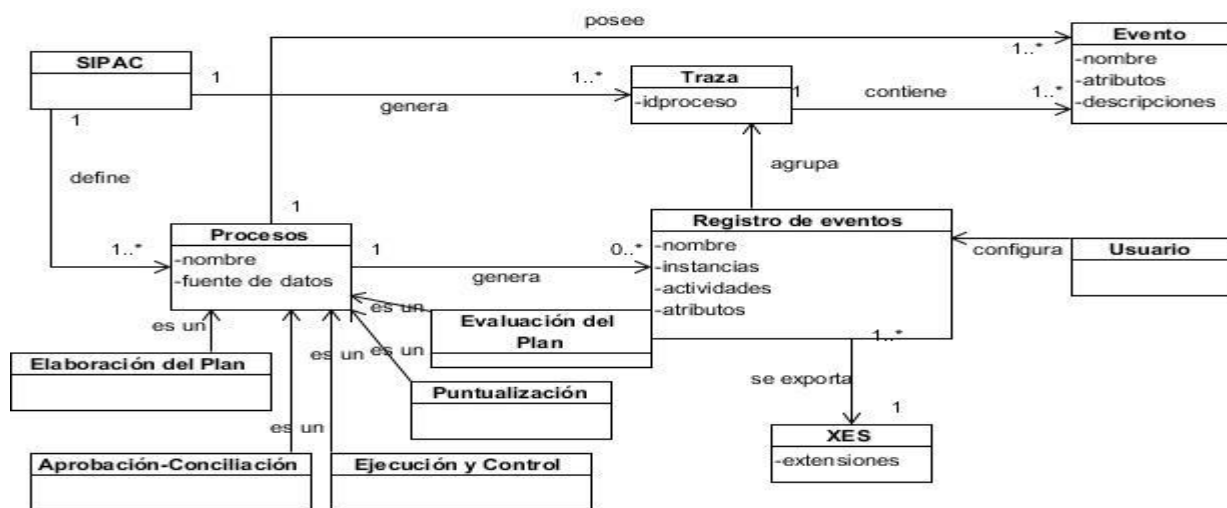


Figura 5. Modelo conceptual.

Fuente: Elaboración del autor

## 2.4 Requisitos de software.

El proceso de desarrollo de software comprende en sus etapas tempranas la definición de tareas orientadas a captar las necesidades o características para satisfacer el sistema que se vaya a crear o modificar. Como resultado de esta captación, se obtendrán los requisitos con los que deberá cumplir la solución (Pressman, 2002).

### 2.4.1 Técnicas de captura de requisitos

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema (Escalona, y otros, 2010). Para extraer los requisitos del sistema se utilizaron las siguientes técnicas:

**Entrevistas:** Se utilizó en los diferentes encuentros con el cliente, realizando preguntas con el objetivo de obtener toda la información posible sobre la visión que el entrevistado tiene de los requisitos y de esta forma comprender los propósitos de la solución buscada.

**Tormenta de ideas:** Esta técnica se implementó mediante la realización de reuniones y encuentros con todos los involucrados en el desarrollo del sistema donde cada cual expresa sus ideas y criterios. Su objetivo fundamental es dar una visión general de las necesidades del sistema.

### 2.4.2 Requisitos funcionales.

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares (Sommerville, 2005).

**Tabla 1.** Descripción de los Requisitos Funcionales.

Requisitos Funcionales	Descripción
Configurar trazas	El sistema permite al usuario configurar el registro de eventos que quiere obtener. Seleccionando: proceso, actividades y recursos.
Listar procesos	El sistema muestra un listado con todos los procesos que posee a partir de los definidos.
Listar actividades	El sistema muestra un listado con todas las actividades que pertenecen a un proceso determinado.
Listar recursos	El sistema muestra los recursos que posee el registro para que el usuario pueda escoger lo que desee para exportar los registros.
Generar trazas	Genera a partir de la definición de los procesos, las actividades y los recursos un registro de eventos.



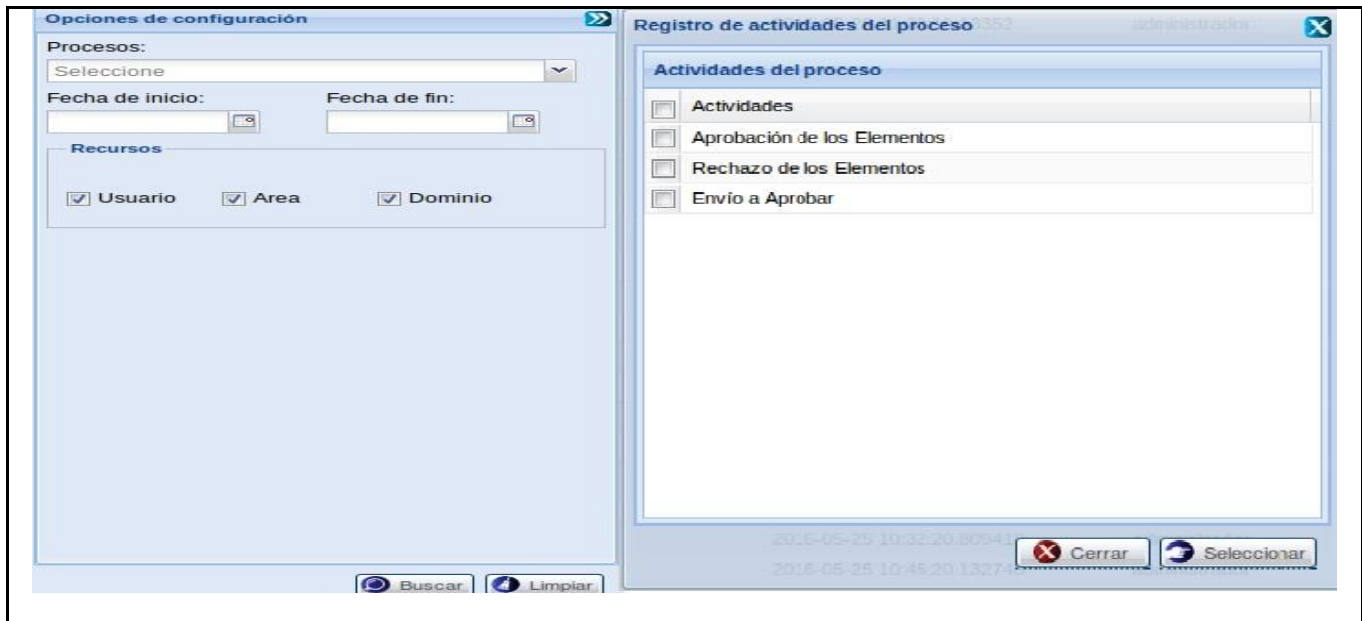
Exportar registro de evento a formato XES	Luego de generado el registro de evento el componente exportara este a formato XES para poder aplicarles técnicas de minería de procesos.
---	---

### 2.4.2.1 Historias de usuario.

De los requisitos funcionales identificados fueron diseñadas sus historias de usuario. Se escogió uno de los requisitos para mostrar un ejemplo de estas historias de usuarios, en este caso es el RF5. Generar trazas, las demás historias de usuarios dirigirse al Anexo No 1.

**Tabla 2.** Historia de Usuario RF 5. Generar trazas.

<b>Número:</b> 5		<b>Nombre del requisito:</b> Generar trazas	
<b>Programador:</b> Leyriel Zurita González		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta		<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en Desarrollo:</b> Poca experiencia del desarrollador en las tecnologías de desarrollo del proyecto.		<b>Tiempo Real:</b> 12 horas	
<b>Descripción:</b> Este requisito permite seleccionar uno de los procesos existentes en SIPAC, a partir de esa selección se muestra al usuario las actividades del proceso en una ventana emergente. El usuario debe seleccionar al menos una de ellas. Luego podrá o no seleccionar el rango de tiempo deseado para el análisis, una vez realizada la selección podrá generar el registro de eventos.			
<b>Observaciones:</b> N/A			
<b>Prototipo de interfaz:</b>			



### 2.4.3 Validación de los requisitos funcionales

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Escalona, y otros, 2002).

Para validar que los requisitos anteriormente identificados y descritos cumplan con las expectativas del cliente y el equipo de desarrollo, se emplea la técnica de validación de requisitos: construcción de prototipos por cada requisito funcional.

También es importante señalar que se realizaron diversas revisiones, por la analista principal del proyecto, a los artefactos relacionados con la disciplina de levantamiento de requisitos para lograr una correcta interpretación de la información transmitida, los señalamientos planteados fueron recogidos y aplicados posteriormente.

### 2.4.4 Requisitos no funcionales.

Los requerimientos no funcionales son requisitos que imponen restricciones en el diseño o implementación. Son propiedades o cualidades que el producto debe tener (Pressman, 2002).

Los requisitos no funcionales especificados son definidos en la arquitectura del sistema y se encuentran descritos en el expediente del proyecto SIPAC 2.0. A continuación se definen algunos que son relevantes para esta solución en específico:

## 2.5 Modelo de datos

En la fase de análisis y diseño se genera de acuerdo con la línea de desarrollo del centro el modelo de datos. El mismo tienen el propósito de garantizar que los datos persistentes sean almacenados coherente, eficazmente y definir el comportamiento que debe ser implementado en la base de datos (Pressman, 2005).

En la Figura 6 se puede observar el modelo de datos de la solución en la cual se ilustran las tablas o entidades generadas en la base de datos de SIPAC, para dar solución a la problemática y las relaciones entre las mismas. Teniendo en cuenta que se generan 8 nuevas entidades, para crear registros de eventos con la información necesaria, en la base de datos se van a implementar 8 funciones *trigger*<sup>2</sup> que van a permitir registrar las acciones que se realicen en el sistema de forma tal que se pueda guardar los eventos asociados a un proceso determinado. Los *trigger* se disparan cuando ocurren en algunas de las tablas existentes inserciones o modificaciones de los elementos que poseen, permitiendo insertar en el modelo de datos diseñado los datos necesarios para generar los registros de eventos.

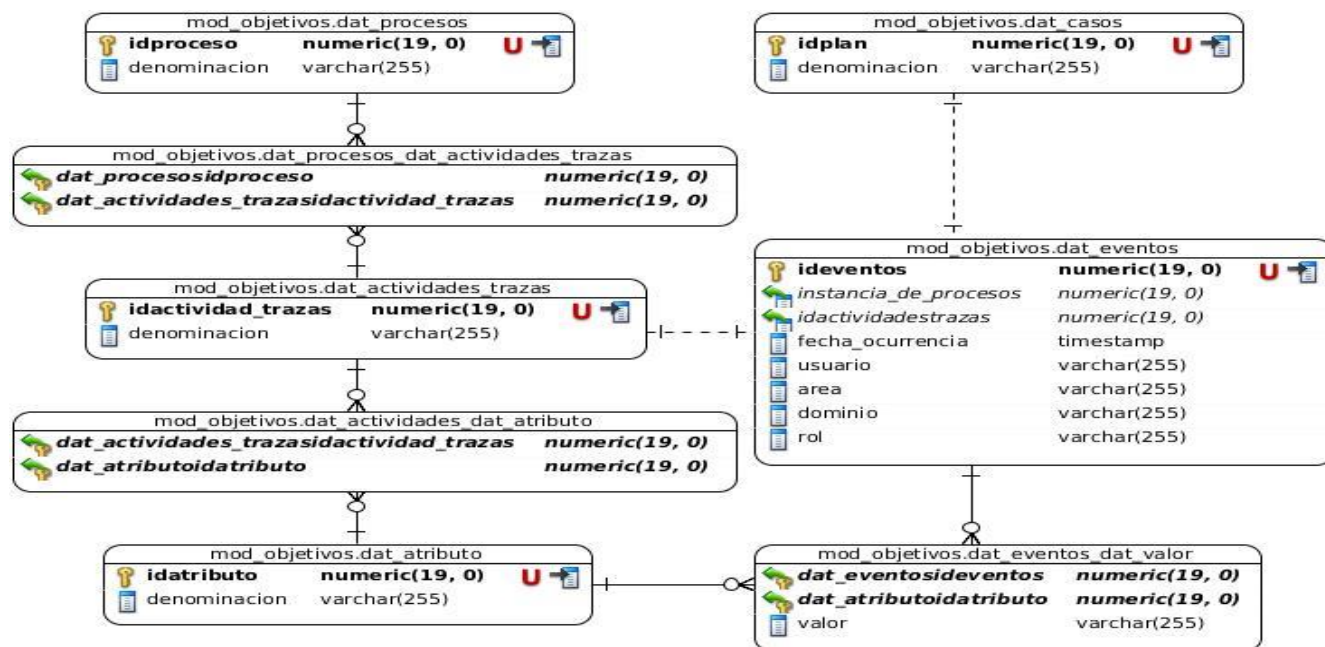


Figura 6. Modelo de datos.

Fuente: Elaboración del autor

<sup>2</sup> *Trigger* (Disparadores): define una acción que la base de datos debe llevar a cabo cuando se produce algún suceso relacionado con la misma. Los disparadores pueden utilizarse para completar la integridad referencial, también para imponer reglas de negocio complejas o para auditar

Para lograr un mejor entendimiento del modelo de datos, a continuación se propone una breve descripción de sus entidades o tablas que lo componen y así obtener una visión del objetivo que se quiere alcanzar con el modelo diseñado.

**Tabla 3.** Descripción de las entidades del modelo de datos.

Entidades	Descripción
mod_objetivos.dat_procesos	Esta entidad tendrá definida los procesos que actualmente se informatizan en SIPAC, mediante un identificador y la denominación de los procesos.
mod_objetivos.dat_procesos_dat_actividades_trazas	Esta entidad va a registrar la relación entre los procesos y las actividades de las trazas, cuales actividades pertenecen a un proceso determinado.
mod_objetivos.dat_actividades_trazas	La entidad tendrá definida todas las actividades que se pueden ejecutar en el sistema mediante un identificador y la denominación de la actividad.
mod_objetivos.dat_actividades_trazas_dat_atributos	Esta entidad va a registrar la relación entre las actividades trazas y los atributos, mediante identificadores.
mod_objetivos.dat_atributo	Registra los atributos de las actividades que se van generando en el registro de eventos.
mod_objetivos.dat_eventos_dat_valor	Registra la relación entre los atributos y los eventos que se van insertando en dat_eventos, mediante sus identificadores.
mod_objetivos.dat_eventos	En esta entidad se insertará registro de eventos como tal, permitiendo guardar la instancia de procesos, la actividad que se ejecutó, su instante de tiempo y los atributos asociados a esa actividad como son el usuario, el área al que pertenece y el dominio.
mod_objetivos.dat_casos	En esta entidad se van a insertar los planes que se crean en el sistema, partiendo que estas serán las instancias de procesos, esta tabla básicamente guarda la ejecución de los procesos.

## 2.6 Tratamiento de los desafíos en la extracción de registros de eventos en SIPAC.

En el capítulo anterior se realizó un estudio sobre los principales desafíos para la extracción de registros de eventos. Una vez que se van creando los *log*<sup>3</sup>, se hace necesario que estos cumplan con los desafíos de Correlación, *Timestamps*, Intervalo de tiempo, Alcance y Granularidad. Durante la extracción se tuvieron en cuenta estos aspectos para guiar la conformación de los registros de eventos a partir del estudio previo realizado.

### **Desafío No.1: Correlación**

Se garantiza a partir de la relación de los eventos a los casos, en SIPAC están definidos 5 procesos fundamentales. Cuando ocurre un evento se asocia a una instancia, por tanto existe un grupo de actividades definidas para cada proceso.

### **Desafío No.2: *Timestamps***

Los registros de eventos en la base de datos, se van registrando a partir de funciones *trigger* que están implementados en diferentes tablas, al ocurrir una acción de inserción o modificación estos permitirán insertar los eventos asociados a una instancia de proceso. Como esto ocurre en tiempo real, los *trigger* van a permitir capturar el *Timestamps* de los log, con niveles de detalles de hasta milisegundos, permitiendo conocer el momento exacto de su ejecución.

### **Desafío No.3: Intervalo de tiempo**

Se definió a partir de la configuración que se realiza para generar el registro, permitiendo seleccionar en intervalo de tiempo para el cual se quiere realizar el análisis. A partir del tiempo que el usuario proporcione se creara el registro mediante una búsqueda que realiza.

### **Desafío No.4: Alcance**

Se resuelve a partir de los *trigger*, SIPAC posee una base de datos en la cual se generan grandes cantidades de datos en diferentes momentos. Los *trigger* van a permitir, sin importar en donde estén los datos registrados, crear registros con esa información. Mientras se vayan realizando acciones en el sistema van obteniendo los datos que necesita para el *log*.

### **Desafío No.5: Granularidad**

Es importante que los registros contengan niveles de detalles a partir de la información que registra. La Granularidad se resuelve en los registros que se están construyendo a partir de los atributos que se incluyen. En este caso los log poseen el usuario que realizó la acción, el rol del usuario, el área a

---

<sup>3</sup> *log*: registro

la que pertenece y el dominio al cual está asociado, permitiendo dar mejores niveles de detalles en la información que se genera.

## 2.7 Modelo de diseño.

Uno de los artefactos generados en la fase de análisis y diseño es el modelo de diseño. Este artefacto está compuesto por los diagramas de clases, los cuales serán la guía para la implementación de la solución.

### 2.7.1 Diagrama de clases de diseño

El diagrama de clases de diseño representa las clases de la solución, así como las principales relaciones que existen entre las clases. A continuación, se muestra el diagrama de clase de diseño del componente, el cual se agrupo por paquetes.

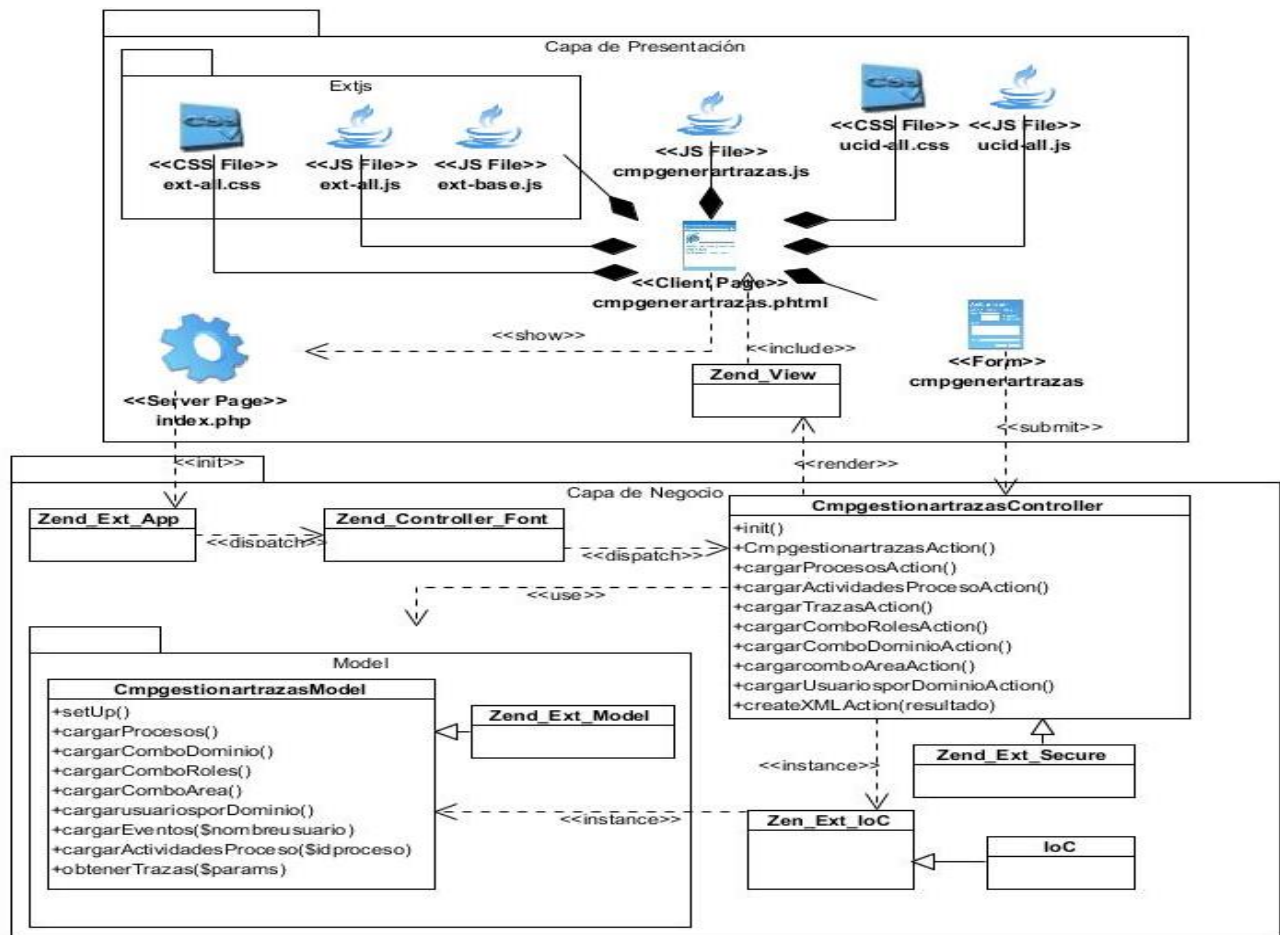


Figura 7. DCD\_Generar trazas, Capa de presentación y negocio.

Fuente: Elaboración del autor

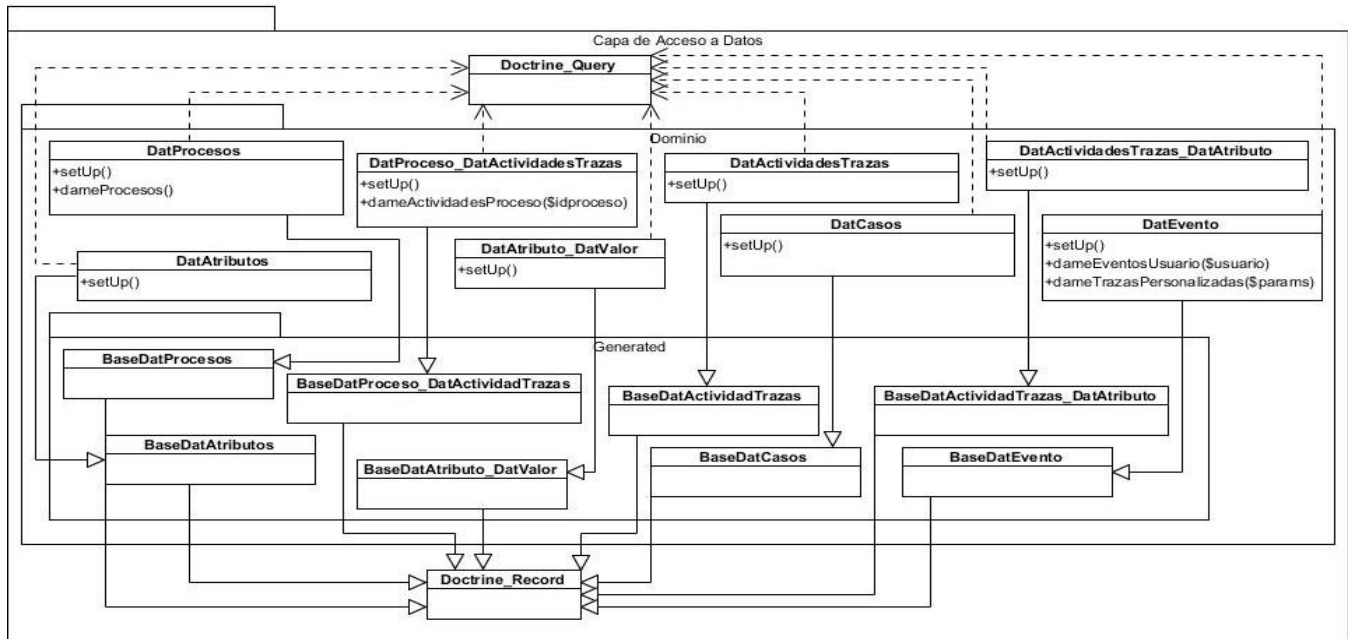


Figura 8. DCD\_Generar trazas, Capa de acceso a datos.

Fuente: Elaboración del autor

### Estilo arquitectónico

El estilo arquitectónico definido por el Departamento de Componente, donde se desarrolla el marco de trabajo Sauxe, es el **Estilo N capas**. Se identificaron cuatro niveles o capas fundamentales. La base tecnológica Sauxe cuenta con la capa de presentación, control, acceso a datos y base de datos (Baryolo, 2010).

- **Capa de presentación:** esta es la capa encargada de elaborar y mostrar las interfaces de los usuarios. La capa de presentación está compuesta principalmente por el marco de trabajo ExtJS y centra su desarrollo en tres componentes fundamentales archivos JS, archivos CSS, páginas clientes y páginas servidoras (Baryolo, 2010).
- **Capa de control o negocio:** en esta capa se encuentra ubicado el marco de trabajo Zend y se implementa el patrón Modelo-Vista-Controlador (Baryolo, 2010).
- **Capa de acceso a datos:** está ubicado el marco de trabajo Doctrine, como *Object Relation Manager* (ORM) para la comunicación con el servidor de datos mediante el protocolo PDO (Protocolo de conexión a base de datos basado en objetos) (Baryolo, 2010).
- **Capa datos:** en esta capa se encuentra ubicada los servidores de base de datos y los archivos XML donde se almacena la información de la aplicación (Baryolo, 2010).

## Patrón arquitectónico

El patrón de arquitectura MVC (Model View Controller), permite realizar la programación multicapa, separando en tres componentes distintos los datos de una aplicación, la interfaz de usuario y la lógica de control. Este patrón está diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. En aplicaciones web, donde la vista es el HTML y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y el controlador representa la lógica del negocio. Además presenta la ventaja de que la conexión entre el Modelo y sus Vistas es dinámica, o sea se produce en tiempo de ejecución, no en tiempo de compilación (Romero, y otros, 2012).

**Modelo:** representa la información con la que trabaja la aplicación, o sea, su lógica de negocio. El modelo no tiene conocimiento específico de los Controladores o las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene la responsabilidad de mantener las relaciones necesarias entre el Modelo y sus Vistas, y notificar a las vistas cuando cambia el modelo (Romero, y otros, 2012).

**Vista:** convierte el modelo en una página web, que facilita al usuario interactuar con ella. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio modelo (Romero, y otros, 2012).

**Controlador:** es el encargado de procesar las interacciones del usuario y ejecuta los cambios adecuados en el modelo o la vista. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el modelo a través de una referencia al propio modelo (Romero, y otros, 2012).



Figura 9. Funcionamiento de una aplicación con MVC.



Fuente: asp.net mvc 2| Mi blog técnico

El patrón arquitectónico Modelo-Vista-Controlador (MVC) se emplea en la capa de control y es el encargado de separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes.

En el caso de la solución propuesta el patrón va a estar reflejado porque se separó la vista, del modelo y el controlador, permitiendo un mejor entendimiento, la vista va a estar reflejada por los js<sup>4</sup>, css<sup>5</sup> y la interfaces que se muestran al usuario para que pueda realizar las peticiones, el modelo se evidencia mediante la clase CmpgenerartrazasModel, la cual va a permitir realizar las peticiones a la base de datos, el controlador mediante la clase CmpgenerartrazasController, permite enviar peticiones desde la vista hasta el modelo y darles las respuestas al usuario mediante las vistas de las respuestas que envió el modelo.

## 2.8 Patrones de diseño

Los patrones de diseño son soluciones a problemas repetidos en la construcción de software y en ocasiones pueden incluir sugerencias para aplicar estas soluciones en diversos entornos (Kaisler, 2005).

En el diseño que propone este trabajo, se utilizaron algunos patrones de diseño los GRASP (traducido al español Patrones Generales de Software de Asignación de Responsabilidades) y los GoF (traducido al español Grupo de Cuatro) y, para dar solución y tratar de evitar diferentes problemas que pudieran aparecer durante la implementación de la solución.

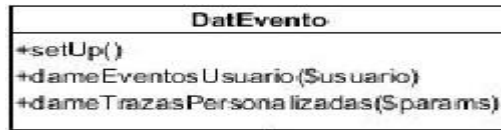
### 2.8.1 Patrones de Diseño de Asignación de Responsabilidades (GRASP)

**Experto:** Se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos (Kaisler, 2005). En este caso las clases DatProcesos, DatProcesosDatActividadesTrazas, DatActividadesTrazas, DatAtributoDatActividadesTrazas, DatAtributo, DatAtributoDatValor, DatEventos, DatCasos las cuales contienen la información necesaria para generar las trazas y exportarlas a XES.

---

<sup>4</sup> guardan funciones y variables globales que se ejecutarán en la página web pudiendo llamar a sus funciones desde cualquier subpágina sin tener que incrustar scripts en cada una de ellas y ahorrando así código.

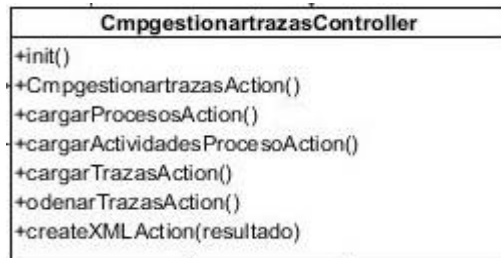
<sup>5</sup> es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.



**Figura 10.** Ejemplo de patrón Experto en clase DatEvento

Fuente: Elaboración del autor

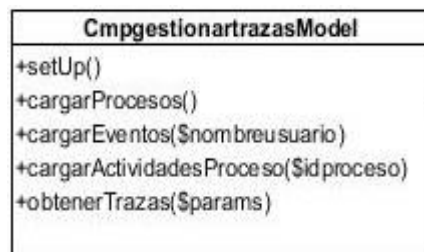
**Controlador:** el patrón controlador funciona como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la interfaz quien recibe los datos del usuario y los envía a las distintas clases según el método invocado (Kaisler, 2005). La clase controladora en el componente es CmpgestionartrazasController porque es el intermediario entre la interfaz de la aplicación y la implementación de los métodos.



**Figura 11.** Ejemplo de patrón Controlador en clase CmpgestionartrazasController

Fuente: Elaboración del autor

**Creador:** se encarga de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Su intención es encontrar un creador que necesite conectarse al objeto creado en alguna situación (Kaisler, 2005). Se evidencia su uso en la clase CmpgestionartrazasController la cual se encarga de crear las instancias de la clase CmpgestionartrazasModel, para así usar las funcionalidades de esta.



**Figura 12.** Ejemplo de patrón Creador en clase CmpgestionartrazasModel

Fuente: Elaboración del autor

**Bajo acoplamiento:** este patrón expresa que entre las clases deberán existir pocas ataduras, es decir, estas estarán lo menos relacionadas posible, de forma tal que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, incrementando la reutilización y disminuyendo la dependencia entre las clases (Kaisler, 2005). El uso de este patrón se evidencia en la poca relación existente entre las clases que conforman el componente.

**Alta cohesión:** propone que la información que almacena una clase debe de ser coherente y debe estar, en la medida de lo posible, relacionada con la clase. Al realizar un cambio en una clase con alta cohesión, todos los métodos que pueden verse afectados, estarán a la vista, en el mismo archivo. Incrementa la claridad, la reutilización y la facilidad de comprensión del diseño (Kaisler, 2005). Este patrón fue utilizado en el diseño del paquete de funcionalidad de manera general.

### 2.8.2 Patrones de Comportamiento (GoF)

**Cadena de responsabilidades (Comportamiento):** es principalmente usado en el tratamiento de Excepciones, si se produce un error en una consulta a la base de datos esta es manejada por el Modelo quien crea una excepción de tipo `ZendExt_Exception` que debe ser enviada al Controlador, quien la envía a la Vista traducida y esta muestra en un lenguaje entendible al usuario la notificación del error.

**Fachada (Estructural):** Este patrón proporciona una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan simplificando así la interacción con el subsistema. Se utiliza para proporcionar un fácil acceso a subsistemas complejos. En el diseño de la solución que se propone, la clase que se utiliza como fachada es `ZendExt_IoC`, para acceder a los servicios de otros componentes.

## 2.9 Diseño de la solución en términos de componentes.

En el diagrama de componentes se muestran los elementos de diseño de un sistema de software. Permite visualizar con más facilidad la estructura general del sistema y el comportamiento de los servicios que estos componentes proporcionan y utilizan a través de las interfaces (Cruz, y otros, 2011).

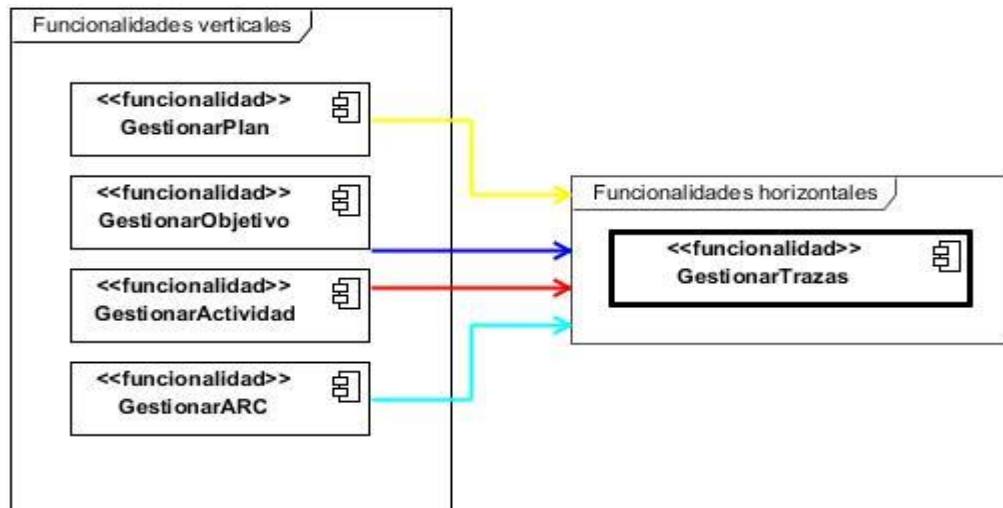


Figura 13. Diagrama de componente.

Fuente: Elaboración del autor

El Diagrama de componente diseñado permite observar el funcionamiento de la solución propuesta, todo el proceso de gestionar las trazas será una funcionalidad de la aplicación donde consumirá recursos de las funcionalidades existentes actualmente. A partir de los recursos que consume se van creando los registros de eventos y luego en la propia funcionalidad de Gestionar Trazas va permitir realizar configuraciones para generar las trazas y poder exportarlas.

## 2.10 Conclusiones parciales

El desarrollo del capítulo actual permitió llevar a cabo la captura, descripción y validación de los requisitos funcionales para comenzar con la implementación de componente.

El diseño de las clases principales del negocio permitió una mejor comprensión de cómo están estructuradas las clases que conforman la solución y la relación entre ellas. Esto permitió diseñar el modelo de datos correspondiente a la solución propuesta.

La definición de los patrones de diseño necesarios para el desarrollo posibilitó una entrada apropiada como punto de partida a las actividades de implementación, con la máxima de lograr una mayor calidad del producto y la satisfacción del cliente. Por consiguiente, una vez precisadas las funcionalidades y la estructura de la solución, es posible dar comienzo a la implementación de la solución.

## CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN

### 3.1 Introducción

En el presente capítulo se presenta la implementación y las validaciones realizadas al componente para la extracción de registros de eventos. Además, se visualizan los resultados obtenidos a partir del análisis de los datos proporcionados por la base de datos de SIPAC. Se realizan también, pruebas de software para comprobar que el componente desarrollado cumpla con los requisitos funcionales descritos en el capítulo anterior.

### 3.2 Normas y estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Constituyen una guía para el desarrollo en las líneas de producción, desde el punto de vista arquitectónico, con el propósito de lograr una estandarización del código. Permiten una mejor integración entre las líneas de producción y se establecerán las pautas que conlleven a lograr un código más legible y reutilizable, de tal forma que se pueda aumentar su mantenibilidad a lo largo del tiempo (Calleja, 2012). Los estándares utilizados en la codificación fueron los siguientes:

**Notación Húngara:** Esta convención se basa en definir prefijos para cada tipo de datos y según el ámbito de las variables. También es conocida como notación REDDICK (por el nombre de su creador). La idea de esta notación es la de dar mayor información al nombre de la variable, método o función definiendo en ella un prefijo que identifique su tipo de dato y ámbito. (Roldan, 2012)

**Notación PascalCasing:** Es como la notación húngara, pero sin prefijos. En este caso, los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula.

**Notación CamelCasing:** Es parecido al PascalCasing con la excepción que la letra inicial del identificador no debe estar en mayúscula.

#### 2.9.1 Nomencladores de las Clases

- **Clase controladora:**

Las clases controladoras después del nombre llevan la palabra: "Controller".

Ejemplo: CmpgenerartrazasController

- **Clase de los modelos:**

Business (Negocio)

Las clases que se encuentran dentro de Business después del nombre llevan la palabra: "Model".

Ejemplo: CmpgenerartrazasModel

- **Clases del Dominio:**

Las clases que se encuentran dentro de Domain reciben el nombre de la tabla en la Base de Datos.

Ejemplo: DatProcesos

- **Clases Generated(Dominio Base):**

Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: "Base" y seguido el nombre de la tabla en la Base de Datos.

Ejemplo: BaseDatProcesos

### 2.9.2 Nomencladores de las funciones

El nombre a emplear para las funciones de las clases del Dominio es utilizando la notación CamelCasing, donde se escribe la primera palabra en minúscula y demás en mayúscula. En el caso de ser una Acción de la clase controladora se escribirá la primera palabra en minúscula y las demás en mayúsculas terminadas por la palabra **Action**.

```
function cargarProcesosAction(){
    $objTrazasModel=new CmpgestionartrazasModel();
    $respuesta=$objTrazasModel->cargarProcesos();
    echo json_encode(array('data'=>$respuesta));
}
```

Figura 14. Ejemplo de nomencladores de las funciones en el controlador

Fuente: Elaboración del autor

### 2.9.3 Nomencladores de las variables

Las variables se escribirán en minúsculas, en el caso de ser el identificador de alguna clase se le adicionará a la palabra el prefijo id haciendo uso de la Notación Húngara, si es más de una palabra se escribirán juntas y en minúsculas todas.

```
public function setUp()
{
    parent::setUp();
    $this->hasMany('DatEventosDatValor', array('local' => ideventos, 'foreign' => ideventos));
}
```

Figura 15. Ejemplo de identificador

Fuente: Elaboración del autor

```
public function obtenerTrazasModel($params){
    $objDatEventos=new DatEventos();
    $datos=$objDatEventos->dameTrazasPersonalizadas($params);
    return $datos;
}
```

Figura 16. Ejemplo de variable

Fuente: Elaboración del autor

### 2.9.4 Nomencladores de los comentarios

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar para lograr una mejor comprensión del código.

```
/**
 * Controlador para gestionar trazas
 *
 * @author Leyriel Zurita Gonzalez
 * @version 1.0.0
 * @copyright SIPAC
 */
```

Figura 17. Ejemplo de comentario

Fuente: Elaboración del autor

## 3.3 Funciones *trigger*

Las funciones *trigger* implementadas en la base de datos constituyen el núcleo de la solución, porque permiten una vez insertado o modificado un elemento en las entidades, registrar en la entidad **dat\_casos** las instancias del proceso ejecutado y en la entidad **dat\_eventos** la ocurrencia de las actividades que se ejecutan en el sistema.

Tabla 4. Descripción de los trigger.

Nombre del <i>Trigger</i>	Descripción
<b>ft_insertarplan_casos():</b>	<b>Inicialización:</b> cuando se realiza una inserción en la entidad <b>dat_plan_pdo</b> . <b>Función:</b> 1- insertar en la entidad <b>dat_casos</b> la instancia de proceso. 2- insertar en <b>dat_eventos</b> , la actividad Creación del plan.
<b>ft_relacionar_elemento():</b>	<b>Inicialización:</b> cuando se realiza una inserción en la entidad <b>dat_elementos_dat_elementos</b> . <b>Función:</b> 1- insertar en <b>dat_eventos</b> , la actividad Relacionar actividad al plan.
<b>ft_relacionar_objplanarc():</b>	<b>Inicialización:</b> cuando se inserta en la entidad <b>dat_arc_dat_plan_objetivo</b> . <b>Función:</b> 1- insertar en la entidad <b>dat_eventos</b> la actividad Relacionar objetivos al plan.
<b>ft_inserta_elementos_eventos():</b>	<b>Inicialización:</b> cuando se realiza una actualización en la entidad

	<p>dat_elementos, del grado de cumplimiento de los elementos o cuando las actividades cambian su estado.</p> <p><b>Función:</b> 1- registrar las actividades de Enviado, Aprobado o Rechazado, Pospuesta o Suspendida de los elementos en la entidad dat_eventos.</p>
<b>ft_grado_estado_indicador():</b>	<p><b>Inicialización:</b> cuando se modifica el grado y el estado de cumplimiento del indicador en la entidad dat_indicadores.</p> <p><b>Función:</b> 1- insertar estas actividades en dat_eventos.</p>
<b>ft_modificar_actividadtiempo():</b>	<p><b>Inicialización:</b> cuando se actualiza el lugar, la fecha o la hora de las actividades registradas en la entidad dat_actividades.</p> <p><b>Función:</b> 1- insertar en dat_eventos las actividades Modificación del Lugar, la Hora y la Fecha, en dependencia del campo que se actualice.</p>
<b>ft_elementos_involucrados():</b>	<p><b>Inicialización:</b> 1- cuando se realiza una inserción en la tabla dat_elementos_dat_involucrados, al involucrar un usuario a un elemento.          2- cuando ocurre una actualización en el grado de cumplimiento de algún indicador.</p> <p><b>Función:</b> 1- insertar actividades Involucrar Usuario o Modificación del Grado de Cumplimiento del Indicador en dat_eventos.</p>
<b>ft_insertar_medida_eventos():</b>	<p><b>Inicialización:</b> 1- cuando se actualiza el valor real de la medida en la entidad dat_medida.</p> <p><b>Función:</b> 1- insertar la actividad Modificado Valor Real de la Medida en la entidad dat_eventos.</p>

```

1
2 BEGIN
3
4 IF((SELECT idversion FROM mod_objetivos.dat versiones INNER JOIN mod_objetivos.dat plan_pdo
5 ON mod_objetivos.dat versiones.idelementoorigen=mod_objetivos.dat plan_pdo.idplan WHERE NEW.idplan=mod_objetivos.dat versiones.idelementoorigen)= @)THEN
6 INSERT INTO mod_objetivos.dat casos(
7   idplan, denominacion)
8   VALUES (NEW.idplan,
9     (SELECT denelemento
10    FROM mod_objetivos.dat elementos WHERE NEW.idplan=mod_objetivos.dat elementos.idelemento));
11
12 INSERT INTO mod_objetivos.dat eventos(instancia de procesos, idactividadstrazas, fecha ocurrencia, usuario, area, dominio, rol)
13   VALUES (NEW.idplan, (SELECT idactividadtraza FROM mod_objetivos.dat actividades trazas WHERE denominacion LIKE '%planes'), now(),
14     (SELECT nombreusuario FROM mod_seguridad.seg_usuario WHERE idusuario=(SELECT idusuariocrea FROM mod_objetivos.dat elementos
15     WHERE idelemento=NEW.idplan)),
16     (SELECT denominacion FROM mod_estructuracomp.dat estructuraop INNER JOIN mod_seguridad.seg_usuario
17     ON(mod_estructuracomp.dat estructuraop.idestructuraop=mod_seguridad.seg_usuario.idarea) INNER JOIN
18     mod_objetivos.dat elementos ON(mod_seguridad.seg_usuario.idusuario=mod_objetivos.dat elementos.idusuariocrea)
19     WHERE NEW.idplan=mod_objetivos.dat elementos.idelemento),
20     (SELECT denominacion FROM mod_estructuracomp.nom dominio INNER JOIN mod_seguridad.seg_usuario
21     ON(mod_estructuracomp.nom dominio.iddominio=mod_seguridad.seg_usuario.iddominio) INNER JOIN
22     mod_objetivos.dat elementos ON(mod_seguridad.seg_usuario.idusuario=mod_objetivos.dat elementos.idusuariocrea)
23     WHERE NEW.idplan=mod_objetivos.dat elementos.idelemento),
24     (SELECT DISTINCT denominacion FROM mod_seguridad.seg rol INNER JOIN mod_seguridad.dat entidad_seg_usuario_seg_rol
25     ON(mod_seguridad.seg_rol.idrol=mod_seguridad.dat entidad_seg_usuario_seg_rol.idrol)
26     INNER JOIN mod_objetivos.dat elementos ON(mod_seguridad.dat entidad_seg_usuario_seg_rol.idusuario=mod_objetivos.dat elementos.idusuariocrea)
27     WHERE mod_objetivos.dat elementos.idelemento=NEW.idplan));
28
29 END IF;
30
31 RETURN NEW;
32
33 END
  
```

**Figura 18.** Ejemplo de Función Trigger: ft\_insertarplan\_casos().

**Fuente:** Elaboración del autor



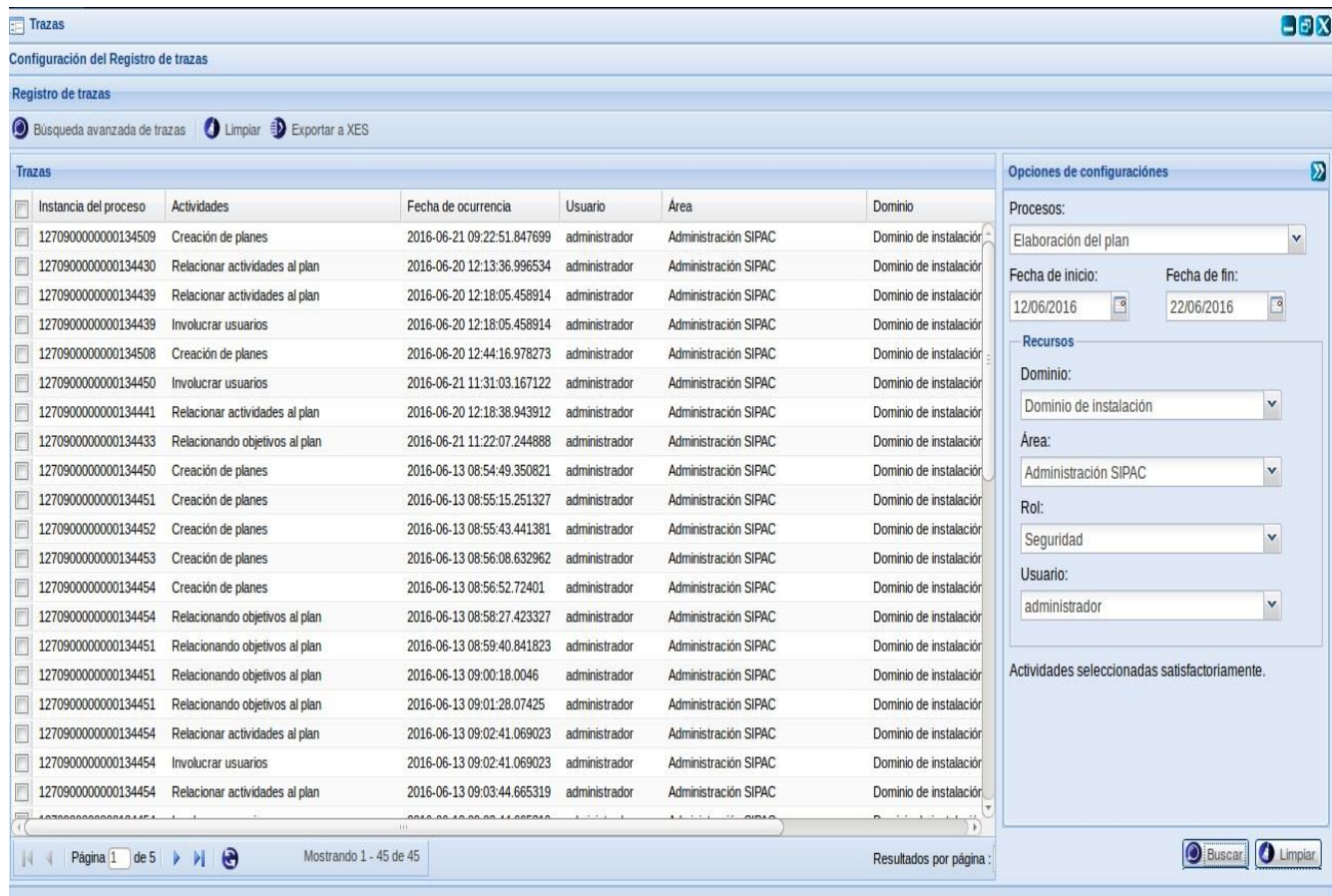
### 3.4 Solución para la extracción de registros de eventos en formato XES

La solución para la extracción de registros de eventos en formato XES de SIPAC, permite generar las trazas de ejecución de los procesos a partir de la definición de: procesos, actividades y recursos, con el objetivo de obtener el registro de eventos que incluya los datos necesarios para analizar y auditar los procesos del negocio gestionados en SIPAC. Además, posibilita al usuario configurar qué trazas quiere obtener mediante la selección del proceso que desea, las actividades de dicho proceso, el rango de fecha de ocurrencia del evento y recursos como: usuario generador, dominio y área a la que pertenece el mismo. Luego de generado el registro de evento la solución facilita exportar este a formato XES, lo que permite aplicar las técnicas de minería de procesos.

El componente posee varias funcionalidades que van a permitir al usuario configurar el registro de eventos para poder generarlos. En primer momento la interfaz de usuario va a mostrar todos los registros generados por la aplicación, luego en una ventana desplegable le va a permitir realizar la configuración del registro de eventos.

En la ventana desplegable el usuario podrá elegir el proceso que desee para realizar el análisis, una vez realizada esta acción le va a mostrar en una ventana emergente las actividades del proceso seleccionado, dándole la opción de escoger como mínimo una de ellas: si no escoge al menos una, no podrá generar el registro de eventos. Luego de seleccionada la actividad tendrá la opción de escoger el rango de tiempo que quiere analizar, la cual no es obligatorio porque puede dejar de seleccionar un rango de fechas y escoger todas las actividades ejecutadas sin importar su fecha, además podrá elegir los atributos que desea incluir en el registro.

Una vez realizadas estas acciones el usuario podrá generar el registro de eventos, refrescándose en la vista para que pueda consultarlos. Luego de generado tiene la opción de exportar a formato XES el registro de eventos para analizarlo aplicándole técnicas de minería de procesos.



**Figura 19.** Interfaz principal

Fuente: Elaboración del autor

### 3.5 Métricas de software

La evaluación de un producto, mediante métricas, es un aspecto fundamental a tener en cuenta, aunque las métricas del producto del software no suelen ser absolutas, brindan la posibilidad de evaluar la calidad a partir de varias reglas definidas claramente. Las métricas son también utilizadas para señalar áreas con problemas, de manera que se puedan desarrollar los remedios y mejorar el proceso del software (Pressman, 2005).

Las métricas empleadas están diseñadas para evaluar los siguientes atributos de calidad:

**Responsabilidad:** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta (Pressman, 2005).

**Complejidad de implementación:** Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado (Pressman, 2005).

**Reutilización:** Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software (Pressman, 2005).

**Acoplamiento:** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización (Pressman, 2005).

**Complejidad del mantenimiento:** Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto (Pressman, 2005).

**Cantidad de pruebas:** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (componente, modulo, clase, conjunto de clases, etc.) diseñado (Pressman, 2005).

### 3.5.1 Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

**Tabla 5.** Atributos de calidad evaluados por la métrica TOC.

Atributo de calidad	Modo en que lo afecta
Responsabilidad	Un aumento en el TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad implementación	Un aumento en el TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento en el TOC implica una disminución del grado de reutilización de la clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

**Tabla 6.** Criterios de evaluación para la métrica TOC.

Atributo	Categoría	Criterio
Responsabilidad	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2^*$ Prom.
	Alta	$> 2^*$ Prom.
Complejidad implementación	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2^*$ Prom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Media	Entre Prom. y $2^*$ Prom.

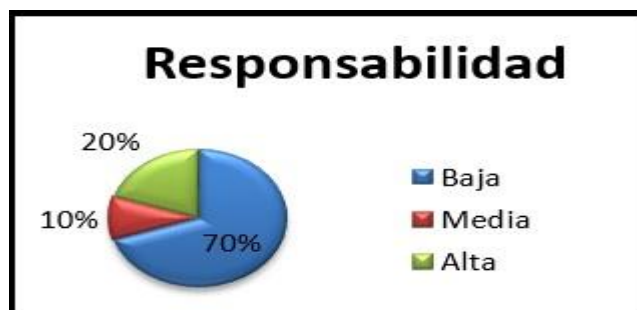
	Alta	<= Prom.
--	------	----------

### 3.4.1.1 Resultados obtenidos de la aplicación de la métrica TOC

Una vez obtenidos los resultados de la evaluación del instrumento de medición de la métrica TOC, se puede concluir que el diseño propuesto tiene una calidad aceptable teniendo en cuenta que el 80% de las clases empleadas en el sistema posee 5 operaciones o menos, lo que conlleva a evaluaciones positivas de los atributos de calidad involucrados. A continuación, se muestran los resultados obtenidos:

**Tabla 7.** Instrumento de evaluación de la métrica.

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
CmpgestionartrazasController	7	Alta	Alta	Baja
CmpgestionartrazasModel	5	Alta	Alta	Baja
DatProcesos	2	Baja	Baja	Alta
DatActividadesTrazas	1	Baja	Baja	Alta
DatProcesosDatActividadesTrazas	2	Baja	Baja	Alta
DatEventos	3	Media	Media	Media
DatCasos	1	Baja	Baja	Alta
DatAtributos	1	Baja	Baja	Alta
DatActividadesTrazasDatAtributos	1	Baja	Baja	Alta
DatAtributosDatValor	1	Baja	Baja	Alta



**Figura 20.** Resultados de la evaluación de la métrica TOC para el atributo Responsabilidad.

Fuente: Elaboración del autor



**Figura 21.** Resultados de la evaluación de la métrica TOC para el atributo Complejidad.

Fuente: Elaboración del autor



**Figura 22.** Resultados de la evaluación de la métrica TOC para el atributo Reutilización.

Fuente: Elaboración del autor

### 3.5.2 Relaciones entre clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

**Tabla 8.** Atributos de calidad evaluados por la métrica RC.

Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento en el RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesaria para probar una clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

**Tabla 9.** Criterios de evaluación para la métrica RC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad implementación	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2 \cdot$ Prom.
	Alta	$> 2 \cdot$ Prom.
Reutilización	Baja	$> 2 \cdot$ Prom.

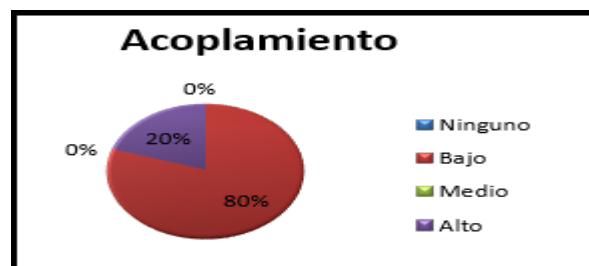
	Media	Entre Prom. y 2*Prom.
	Alta	<= Prom.
Cantidad de pruebas	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.

### 3.4.2.1 Resultados obtenidos de la aplicación de la métrica RC

Una vez obtenidos los resultados de la evaluación del instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto tiene una calidad aceptable teniendo en cuenta que el 60% de las clases empleadas posee 3 o menos dependencias de otras clases, lo que conlleva a evaluaciones positivas de los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización). A continuación, se muestran los resultados obtenidos:

**Tabla 10.** Instrumento de evaluación de la métrica RC.

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
CmpgestionartrazasController	6	Alto	Alta	Baja	Alta
CmpgestionartrazasModel	3	Alto	Media	Media	Media
DatProcesos	1	Baja	Baja	Alta	Baja
DatActividadesTrazas	1	Baja	Baja	Alta	Baja
DatProcesosDatActividadesTrazas	1	Baja	Baja	Alta	Baja
DatEventos	1	Baja	Baja	Alta	Baja
DatCasos	1	Baja	Baja	Alta	Baja
DatAtributos	1	Baja	Baja	Alta	Baja
DatActividadesTrazasDatAtributos	1	Baja	Baja	Alta	Baja
DatAtributos_DatValor	1	Baja	Baja	Alta	Baja



**Figura 23.** Resultados de la evaluación de la métrica RC para el atributo Acoplamiento.

**Fuente:** Elaboración del autor



Figura 24. Resultados de la evaluación de la métrica RC para el atributo Complejidad de Mantenimiento.

Fuente: Elaboración del autor



Figura 25. Resultados de la evaluación de la métrica RC para el atributo Cantidad de pruebas

Fuente: Elaboración del autor



Figura 26. Resultados de la evaluación de la métrica RC para el atributo Reutilización

Fuente: Elaboración del autor

### 3.6 Pruebas de software

El único instrumento adecuado para determinar el *status* de la calidad de un producto de software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada (software, 2010).

Existen dos enfoques principales para el diseño de casos de prueba:

1. El enfoque estructural o de caja blanca: que se basa en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
2. El enfoque funcional o de caja negra: que realiza pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa (Huanca, 2011).

### 3.6.1 Método de caja blanca

Las pruebas de caja blanca son un método de diseño que usa la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de prueba. Al emplear sus métodos, se puede garantizar que, al menos una vez, se ejecuten todas las rutas independientes del módulo. La técnica del camino básico permite derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control (Pressman, 2005).

A continuación se muestra un ejemplo de las pruebas de caja blanca realizadas a la solución, en el que para ejemplificar se escogió el método `dameTrazasPersonalizadas` de la clase `DatEventos`, lo primero es enumerar las sentencias del código y a partir del mismo se construye el grafo de flujo correspondiente.

```

public function dameTrazasPersonalizadas($params){
    $conn2 = Doctrine_Manager::connection(); ①
    $cadena=" SELECT * FROM mod_objetivos.dat_eventos where "; ②
    $cadena .= " mod_objetivos.dat_eventos.idactividadestrazas IN(" . implode(", ", $params->idactividades) . ")";
    if($params->fechainicio && $params->fechafin){ ③
        $cadena .= " and mod_objetivos.dat_eventos.fecha_ocurrencia between '$params->fechainicio' and '$params->fechafin' ";
    }
    elseif($params->fechainicio){ ④
        $cadena .= " and mod_objetivos.dat_eventos.fecha_ocurrencia >= '$params->fechainicio' ";
    }
    elseif($params->fechafin){ ⑤
        $cadena .= " and mod_objetivos.dat_eventos.fecha_ocurrencia <= '$params->fechafin' ";
    }
    $cadena .= " and mod_objetivos.dat_eventos.usuario = '$params->usuario' "; ⑥
    $result = $conn2->fetchAll($cadena); ⑦
    $total = count($result); ⑧
    $resultado['total'] = $total; ⑨
    $resultado['resultado'] = $result;
    return $resultado; ⑩
}

```

Figura 27. Código Fuente de la Funcionalidad `dameTrazasPersonalizadas`

Fuente: Elaboración del autor

Posteriormente se debe proceder a la elaboración del grafo de flujo teniendo en cuenta la enumeración.



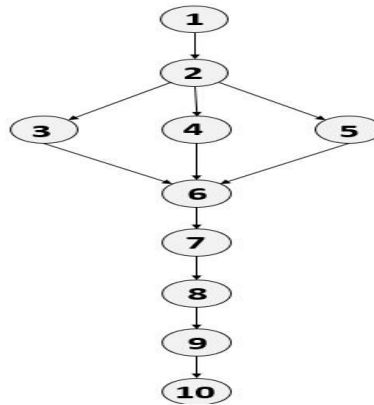


Figura 28. Grafo de flujo asociado a la funcionalidad dameTrazasPersonalizadas

Fuente: Elaboración del autor

### Cálculo de la complejidad ciclomática

La complejidad ciclomática es la métrica de software con que se define la cantidad de caminos independientes de cada una de las funcionalidades del programa y provee el límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez (Pressman, 2005).

La complejidad ciclomática se basa en la teoría gráfica y se calcula de tres maneras distintas, donde, para que el cálculo sea correcto, todas deben arrojar el mismo resultado:

1. El número de regiones corresponde a la complejidad ciclomática "**V (G)**".

$$V (G) = R \text{ (Donde } R \text{ es la cantidad total de regiones)}$$

$$V (G) = 3$$

2.  $V (G) = E - N + 2$  (Donde **E** es el número de aristas y **N** número de nodos)

$$V (G) = 11 - 10 + 2$$

$$V (G) = 3$$

3.  $V (G) = P + 1$  (Donde **P** es el número de nodos predicados)

$$V (G) = 2 + 1$$

$$V (G) = 3$$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, dando como resultado 3, lo que indica que existen 3 posibles caminos por donde el flujo puede circular, y determina el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

El cálculo arrojó que  $V(G) = 3$ , por lo que los posibles caminos básicos son:

- ✓ Camino Básico No 1: 1,2,3,6,7,8,9,10.
- ✓ Camino Básico No 2: 1,2,4,6,7,8,9,10.
- ✓ Camino Básico No 3: 1,2,5,6,7,8,9,10.

**Caso de prueba para camino básico N° 1(1,2,3,6,7,8,9,10):**

Si  $\$params \rightarrow fechaInicio$  and  $\$params \rightarrow fechaFin$ ,  $\$cadena = fechaOcurrencia$  between  $\$params \rightarrow fechaInicio$  and  $\$params \rightarrow fechaFin$ .

**Caso de prueba para camino básico N° 2(1,2,4,6,7,8,9,10):**

Si  $\$params \rightarrow fechaInicio$ ,  $\$cadena = fechaOcurrencia \geq \$params \rightarrow fechaInicio$ .

**Caso de prueba para camino básico N° 3(1,2,5,6,7,8,9,10):**

Si  $\$params \rightarrow fechaFin$ ,  $\$cadena = fechaOcurrencia \leq \$params \rightarrow fechaFin$ .

### 3.6.2 Método de caja negra

A las funcionalidades se le realizaron pruebas funcionales o de caja negra. Estas pruebas permiten obtener conjuntos de entrada que ejerciten los requisitos funcionales del software, complementándose con las pruebas de caja blanca, obteniendo errores en las siguientes categorías (Pressman, 2005):

- Funciones incorrectas o inexistentes.
- Errores en la interfaz.
- Errores en la estructura de datos.
- Rendimiento
- Inicialización y terminación.

En el presente trabajo se utilizará la técnica de partición de equivalencia pues esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software. Además, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. Para aplicar esta técnica se confeccionaron los diseños de caso de prueba para las funcionalidades del componente, donde se realizaron tres iteraciones para encontrar los errores y resolverlos.

### Diseño de caso de prueba

El objetivo del diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos. Para diseñar un caso de prueba, se selecciona las funcionalidades que se está probando, luego se selecciona un conjunto de entradas que ejecutan dicha funcionalidades, documenta las salidas esperadas o rangos de salida, y donde sea posible se diseña una prueba automatizada que prueba que las salidas reales y esperadas son las mismas (Sommerville, 2005).

### Casos de pruebas diseñados

A continuación se muestran los casos de prueba que han sido diseñados con el fin de cubrir todas las posibilidades, estos corresponden a los requisitos donde se describen los valores de entrada y los resultados observados aplicando el método de prueba de caja negra y utilizando la técnica de partición de equivalencia.

### Descripción general

El requisito se inicia cuando el usuario selecciona la opción de la funcionalidad Trazas en el sistema. Para configurar las trazas debe ir al panel de configuración y realizar las configuraciones necesarias para generar las trazas. Una vez realizada la configuración podrá generar las trazas y luego exportarlas a formato XES.

### Condiciones de ejecución:

El usuario debe seleccionar la funcionalidad Trazas en Recuperaciones.

Debe configurar las trazas al menos una vez.

**Tabla 11.** Caso de prueba para la aplicación

Escenario	Descripción	Variabes	Respuesta del sistema	Flujo central
EC 1.1 Configurar trazas	Mostrarle al usuario un panel donde tengas las opciones para la configuración.	Trazas	Muestra el panel donde el usuario podrá realizar la configuración.	1-El usuario selecciona la opción Trazas en la aplicación. 2-Luego Selecciona El botón "Búsqueda Avanzada de Trazas". 3-Se despliega el Panel de configuración.
EC 1.2 Listar procesos	Permite al usuario seleccionar el proceso del cual quiere generar sus trazas.	Procesos	Mostrarle al usuario un combo con los procesos del sistema para que pueda escoger uno.	1-En el Panel de Configuración, se mostrara un combo donde podrá seleccionar el proceso. 2-Selecciona el proceso que desee.

EC 1.3 Listar Actividades	Permite al usuario seleccionar las actividades de un proceso determinado.	ActividadesTrazas	Le muestra al usuario una ventana con las actividades que pertenezcan al proceso que seleccionó.	1-Luego de Seleccionado el proceso le mostrara una ventana emergente con las actividades. 2-Podrá seleccionar las actividades que desee, al menos una de ellas.
EC 1.4 Seleccionar Fecha	Permite al usuario seleccionar un rango de fechas que desee analizar las actividades que ocurrieron.	FechaInicio, FechaFin	Le muestra al usuario la forma de seleccionar un rango de fechas que desee, pero esta opción también puede dejarla en blanco, no es obligatoria.	1-En el panel de configuración se le mostrar la opción de seleccionar la fecha de inicio o fin. 2-Seleccionara el rango de fechas que desee. 3-Podrá dejar en blanco el rango de fechas. 4-Podrá seleccionar la fecha de inicio o de fin o ninguna de ellas.
EC 1.5 Generar trazas	Permite generar al usuario las trazas que desea analizar, luego de realizar la configuración	TrazasPersonalizadas	El sistema una vez realizada toda la configuración de como el usuario quiere las trazas, tendrá la opción de generarla en la parte principal del sistema.	1-Luego de configurado el registro de eventos. 2-Podrá seleccionar la opción "Buscar". 3-Se generarán automáticamente las trazas personalizadas.
EC 1.6 Exportar registro de trazas	Permite al usuario exportar las trazas generadas a formato XES.	TrazasPersonalizadas	El sistema da la opción de poder exportar el registro a formato XES, mediante un botón habilitado para ello.	1-Luego de generadas las trazas en la aplicación. 2-El usuario podrá exportar mediante la opción "Exportar", las trazas a XES.

**Tabla 12.** Descripción de las variables

Número	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Trazas	Lista de trazas	No	Posee un listado con todas las trazas generadas en la aplicación.
2	Procesos	Lista desplegable	No	Muestra un listado con los procesos que posee el sistema.
3	ActividadesTrazas	Lista de las actividades trazas	No	Muestra todas las actividades referentes al proceso que se seleccionó en un primer momento.

4	FechaInicio	Timestamp	Si	Si el usuario selecciona un rango de fecha esta tendrá la fecha de inicio de ese rango.
5	Fechafin	Timestamp	Si	Si el usuario selecciona un rango de fecha esta tendrá la fecha de fin de ese rango.
6	TrazasPersonalizadas	Lista de Trazas	No	Posee un listado con las trazas configuradas por el usuario.

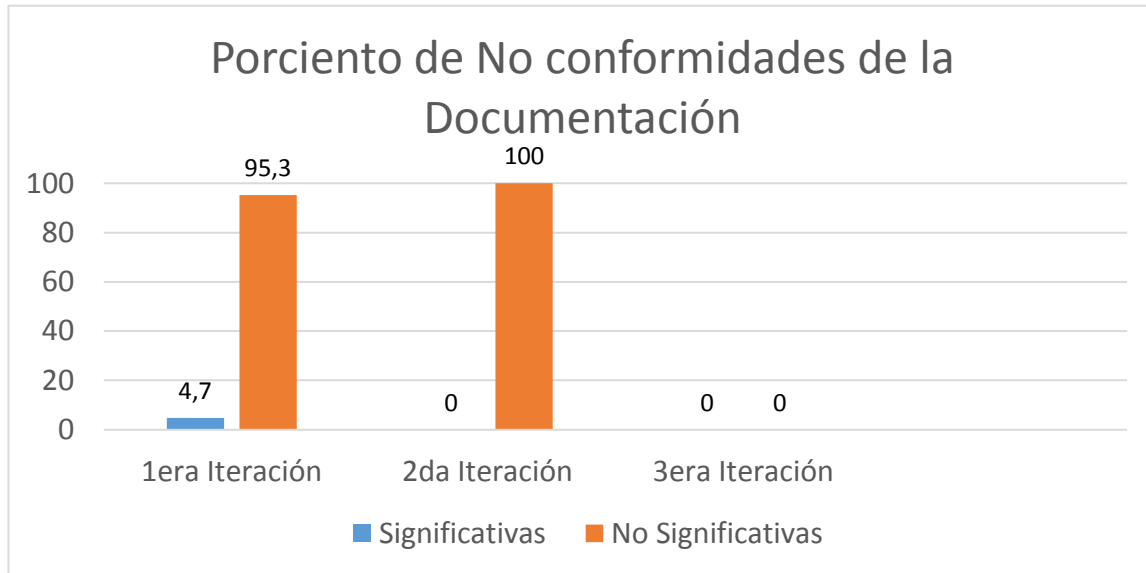
### Resultado de las Pruebas

Para comprobar la calidad de la solución se realizaron 3 iteraciones por Calidad del Proyecto SIPAC. Las no conformidades (NC) encontradas se clasificaron en: NC de la aplicación y NC del documento de Diseño de Casos de Prueba. En la siguiente tabla se muestran las iteraciones realizadas y la cantidad de NC encontradas en cada iteración.

**Tabla 13.** Tabla de No Conformidades de la aplicación y el documento

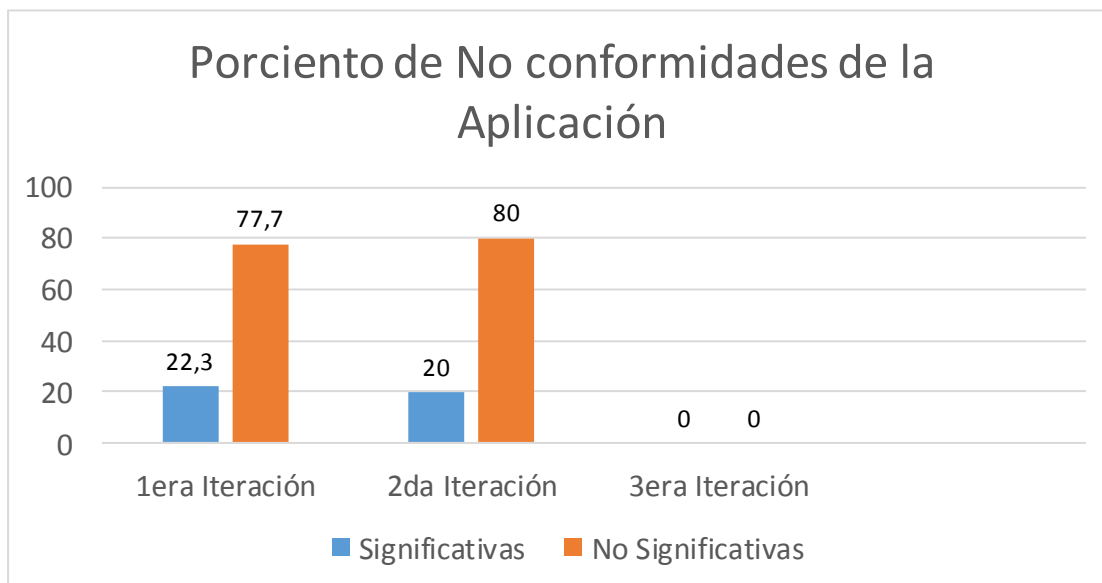
Tipo de no Conformidad	Documentación	Aplicación
<b>Primera Iteración</b>		
Significativa	1	2
No significativa	21	7
Total	22	9
<b>Segunda Iteración</b>		
Significativa	0	1
No significativa	5	5
Total	5	6
<b>Tercera Iteración</b>		
Significativa	0	0
No significativa	0	0
Total	0	0

Para lograr un mejor entendimiento se graficaron los valores obtenidos en las pruebas calculando el porcentaje que representaba con respecto a su total en cada iteración realizada.



**Figura 29.** Porcentaje de No Conformidades de la Documentación

**Fuente:** Elaboración del autor



**Figura 30.** Porcentaje de No Conformidades de la Aplicación

**Fuente:** Elaboración del autor

### 3.7 Pruebas de aceptación

Son pruebas que permiten que el cliente valide todos los requisitos. Las realiza el usuario final en lugar del responsable del desarrollo del sistema. Una prueba de aceptación puede ir desde un informal caso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. De

hecho, la prueba de aceptación puede tener lugar a lo largo de semanas o meses, descubriendo así errores acumulados que pueden ir degradando el sistema (Pressman, 2006).

Estas pruebas fueron realizadas por la Jefa de Proyecto de SIPAC **Ing. Mairelys Fernández González**, la cual interactuó con el Componente para la extracción de registros de eventos en formato XES del Sistema de Planificación de Actividades SIPAC, realizando diferentes operaciones que permitían observar la puesta en funcionamiento del sistema. Las funcionalidades cumplían con las necesidades planteadas por el cliente al comenzar la investigación, de esta forma queda constancia en el Acta de aceptación emitida por el cliente la cual está en el Anexo No.2.

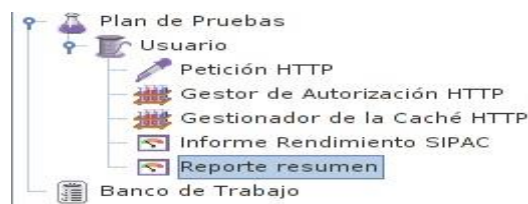
### 3.8 Pruebas de rendimiento

Con el objetivo de garantizar la calidad en el software se realizan los siguientes tipos de pruebas:

- **Pruebas de carga:** enfocada a evaluar el rendimiento del sistema y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios o transacciones que se espera en el ambiente de producción.

La prueba de carga para los servicios web se realizó utilizando la herramienta JMeter mencionada en el Capítulo I. La misma se configuró utilizando el árbol de configuración de la Figura 31 donde se realizaran cinco grupos de peticiones por usuario, las cuales estaban dadas de la siguiente manera:

- Insertar plan.
- Relacionar actividad al plan.
- Aprobar elementos.
- Modificar porcentaje de cumplimiento de los elementos.
- Relacionar objetivos al plan.



**Figura 31.** Árbol de configuración

**Fuente:** Elaboración del autor

JMeter permite establecer como variables principales a la hora de realizar una simulación las siguientes, para una mejor comprensión ver Figura 32:

- **Número de hilos:** Se corresponde con el número de procesos o usuarios concurrentes que van a ejecutar las peticiones, en este caso para los dos escenarios 200 usuarios.
- **Período de subida:** Es el tiempo en segundos que tardan en iniciarse todos los procesos, en este caso se empleó 1 segundo.
- **Control de bucle:** Es el número de veces que se van a lanzar las iteraciones por los 200 usuarios determinados en el número de hilos siendo 1 para este caso.

**Figura 32.** Configuración utilizada en JMeter con 200 hilos

Fuente: Elaboración del autor.

## Escenarios de pruebas

### Escenario No.1

**Tabla 14.** Caso de prueba escenario 1

Escenario 1		
Periodo de subida	Casos	Descripción
1 segundo	200 usuarios	Se utilizará la base de datos de la UCI, sin ninguna modificación, para observar los tiempos de respuestas.

### Escenario No.2

**Tabla 15.** Caso de prueba escenario 2

Escenario 2		
Periodo de subida	Casos	Descripción
1 segundo	200 usuarios	Se utilizará una base de datos en la cual están implementados 8 funciones <i>tigger</i> , y además estarán las tablas del modelo de datos diseñado, donde se van a insertar las ejecuciones de los <i>tigger</i> .



Las tablas que se muestran a continuación corresponden a los datos de entrada para cada uno de los escenarios y el resultado obtenido a partir de la ejecución. Este resultado en las pruebas permite constatar que los tiempos varían pero no de forma notable y por tanto no afecta considerablemente el rendimiento del sistema.

**Tabla 16.** Resultado de las pruebas escenario 1

Escenario 1					
Periodo de subida	Casos		Resultados para 200 usuarios		
1 segundo	200 usuarios	Peticiones	Solicitudes	Tiempo medio	Errores
		Insertar plan.	967	90.4/sec	0%
		Relacionar actividad al plan.	809	60.6/sec	0%
		Aprobar elementos.	783	62.8/sec	0%
		Modificar porcentaje de cumplimiento de los elementos.	739	50.9/sec	0%
		Relacionar objetivos al plan.	790	95.4/sec	0%

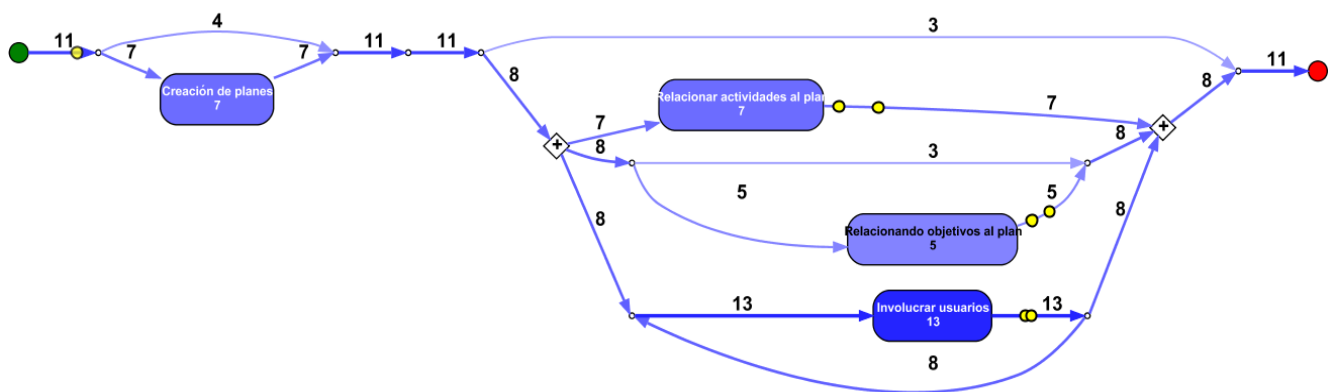
**Tabla 17.** Resultados de las pruebas escenario 2

Escenario 2					
Periodo de subida	Casos		Resultados para 200 usuarios		
1 segundo	200 usuarios	Peticiones	Solicitudes	Tiempo medio	Errores
		Insertar plan.	617	98.5/sec	0%
		Relacionar actividad al plan.	733	96.9/sec	0%
		Aprobar elementos.	727	98.7/sec	0%
		Modificar porcentaje de cumplimiento de los elementos.	1028	80.9/sec	0%
		Relacionar objetivos al plan.	738	98.5/sec	0%

### 3.9 Validación del fichero XES mediante un caso de estudio

Para realizar la validación del fichero XES a partir de un caso de estudio se utilizó el registro de eventos para el proceso Elaboración del plan, a partir de la interacción de un usuario con el sistema, el cual va a crear planes de trabajos, relacionar actividades a estos planes, relacionar objetivos e involucrar los usuarios que intervienen en esas actividades.

Este registro de eventos fue importado por la herramienta ProM 6.5.1. El hecho de que la herramienta ProM importara satisfactoriamente el registro de eventos en formato XES, implica que dicho registro posee una estructura sintácticamente correcta. Una vez importado, se aplicó el algoritmo “Inductive Visual Miner” (ver figura 33). Al generarse el modelo de proceso satisfactoriamente, se demuestra la completitud en la extracción realizada de las trazas de procesos. De esta forma se evidencia que la solución genera el registro de eventos de forma segura.



**Figura 33.** Modelo generado a partir de la extracción del registro del proceso Elaboración del plan

Fuente: Elaboración del autor

### 3.10 Impacto de la solución

La solución para la extracción de registros de eventos en formato XES se considera de gran impacto para SIPAC, debido a que permite generar las trazas del sistema centrándose en los procesos. De esta forma, las acciones que realizan los usuarios en el módulo de Planificación se asocian a las actividades definidas en el negocio. Al exportar el registro de eventos a formato XES, a este se le pueden aplicar técnicas de minería de procesos en herramientas como ProM y Disco. Esto posibilita realizar análisis objetivos de estos procesos, mediante preguntas e indicadores claves de rendimiento (KPIs). De esta forma se puede conocer el funcionamiento real de la planificación de objetivos y actividades de una organización que utilice SIPAC, en menor tiempo y con mayor fiabilidad que aplicando técnicas tradicionales de análisis de procesos.

### 3.11 Conclusiones parciales

En el desarrollo de este capítulo se pudo apreciar la importancia que tiene el proceso de prueba en el desarrollo de software, sus objetivos y alcance. Se abordaron las métricas de software que fueron aplicadas en la solución y se determinó que el diseño de la implementación estuvo correcto, tal como fue mostrado por las métricas.

Las pruebas de caja blanca y caja negra realizadas a la solución demostraron que esta respondía correctamente a los requisitos identificados a partir de las necesidades del cliente.

Luego de realizadas estas pruebas el cliente emitió un aval demostrando la importancia del solución y la aceptación del producto finalizado, teniendo en cuenta que cumplía con los requerimientos planteados por él.

## CONCLUSIONES

Con el desarrollo del componente para la extracción de registros de eventos a formato XES del Sistema de Planificación de Actividades se concluye lo siguiente:

El análisis de los principales conceptos, tecnologías y herramientas de extracción de registros de eventos a formato XES, permitió una mejor comprensión de la investigación obteniendo los basamentos teóricos necesarios para su desarrollo y demostró la necesidad de implementar una solución para la extracción de registros de eventos a formato XES a la medida para SIPAC.

A través del análisis y diseño se definió la solución a implementar para un mejor entendimiento de la misma, teniendo en cuenta las necesidades del cliente.

Se obtuvo la solución para la extracción de registros de eventos en formato XES de SIPAC, que responde a las necesidades del cliente, la cual fue comprobada a través de pruebas de software efectuadas al obtenerse resultados positivos.

Se validó el objetivo de la investigación mediante un caso de estudio, lo que permitió certificar que los registros de eventos exportados por la solución están correctamente estructurados y contienen la información necesaria para aplicar las técnicas de minería de procesos.



## RECOMENDACIONES

- Definir los indicadores para evaluar la ejecución de los procesos, luego de realizar la exportación de los registros a formato XES del Sistema de Planificación de Actividades.
- Definir las técnicas de minería de procesos para realizar los análisis luego de seleccionado los indicadores a medir para cada uno de los procesos de SIPAC.

## REFERENCIAS

1. **Aalst, Van Der. 2011.** *Process Mining Discovery Conformance and Enhancement of Business Processes*. 2011. ISBN 978-3-642-19344-6..
2. **Aalst, Van Der y P., W. M. 2011.** Conformance and Enhancement of Business Processes. [En línea] 2011. [Citado el: 11 de 11 de 2015.] <http://www.google.com/books?hl=en&lr=&id=I1KOAfiqfxYC&oi=fnd&pg=PR4&dq=Discovery,+Conformance+and+Enhancement+of+Business+Processes&ots=KkS69vVBL&sig=WLc3SuDDHQYJbX4dZpraXFzsgI>.
3. **Aalst, Wil Van Der. 2014.** *What is processmining?* processmining, 2014.
4. **Aalst, Wil van der, y otros. 2011.** *Manifiesto sobre Minería de Procesos*. s.l. : IEEE Task Force on Process Mining, 2011.
5. **Acevedo, Karel Ernesto Garateix. 2012.** *Sistema de Interoperabilidad para los servicios de gestion y destino turisticos*. Ciego de Ávila : Facultad Regional de las Ciencias Informáticas, 2012.
6. **Acuña, Karenny Brito. 2013.** Eumed.net. [En línea] 30 de 4 de 2013. [Citado el: 24 de 4 de 2016.] <http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.html>.
7. **Anónimo. 2014.** Herramienta CASE que da soporte al modelado visual con UML 8.0. [En línea] 2014. [Citado el: 20 de 11 de 2015.] <http://www.visual-paradigm.com..>
8. **Apache. 2012.** Subversion. [En línea] 2012. [Citado el: 12 de 1 de 2016.] <http://subversion.apache.org..>
9. **Baryolo, Oiner Gómez. 2010.** *Solución Informática de Autorización*. La Habana : Universidad de las Ciencias Informáticas, 2010. S/N.
10. **Buijs, J.C.A.M. 2010.** Mapping Data Sources to XES in a Generic Way. [En línea] 3 de 2010. [Citado el: 27 de 10 de 2015.] [http://www.processmining.org/\\_media/xesame/xesma\\_thesis\\_final.pdf](http://www.processmining.org/_media/xesame/xesma_thesis_final.pdf).
11. **Calleja, Manuel Arias. 2012.** Carmen. Estándares de codificación. [En línea] 2012. [Citado el: 29 de 4 de 2016.] <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf..>
12. **CEIGE. 2015.** *Manual de usuario para el Sistema de Planificación de Actividades*. Habana : UCI, 2015. CIG-SPA-N-i4302.
13. **Cobo, Angel y Gomez, Patricia. 2005.** PHP y MySQL- tecnologías para el desarrollo de aplicaciones web. s.l. : Ediciones Díaz Santos, 2005.
14. **Corella, Aliuska Batista y Martínez, Yordany Aguilera. 2015.** *Componente para la extracción de registros de eventos en formato XES del sistema ZUN Suite*. La Habana : Universidad de las Ciencias Informáticas, 2015. S/N.
15. **Cortiñas, Jesús L. 2010.** Apuntes de Gestión. [En línea] 2010. [Citado el: 25 de 1 de 2016.] <http://www.apuntesgestion.com/definicion-planificacion/>.
16. **Cruz, Victor favio, Gutierrez, Ever Dino y Mendivil, Luís Brian. 2011.** Slideshare. [En línea] LinkedIn Corporation, 7 de 4 de 2011. [Citado el: 26 de 4 de 2016.] <http://es.slideshare.net/uitron/diagrama-de-componentes-7551535>.
17. **Cuter, Blades, y otros. 2011.** *Learning Ext JS*. s.l. : Packt Publishing. Virginia : Escuela Tecnica Superior de Ingenieria, 2011.

18. Escalona, María José y Koch, Nora. 2010. Ingeniería de Requisitos en Aplicaciones para la Web - Un estudio comparativo. [En línea] 12 de 2010. [Citado el: 26 de 2 de 2016.] <https://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>. S/N.
19. Felipe, Wilfredo García, González, Daniel González y Quijano, Mercedes Martínez. 2010. *UN ACERCAMIENTO A LA PLANIFICACIÓN EN LA REALIDAD CUBANA ACTUAL*. Habana : Revista IPLAC, 2010. ISSN 1993-6850.
20. Fluxicon. 2012. Fluxicon Process Minnig for Professionals. [En línea] 2012. [Citado el: 16 de 1 de 2016.] <http://fluxicon.com/disco/>.
21. Framework, Zen. 2010. Zen Framework. [En línea] Zend Technologies Ltd. All rights reserved, 2010. [Citado el: 16 de 1 de 2016.] <http://framework.zend.com/about/>.
22. Framework, Zend. 2006. Zend Framework 2. [En línea] 2006. [Citado el: 15 de 1 de 2016.] <http://framework.zend.com/>.
23. Group, The PHP. 2016. PHP. [En línea] 3 de 3 de 2016. [Citado el: 24 de 2 de 2016.] <http://php.net/>.
24. Gunther, C W. 2009. *XES Extensible Event Stream standard definition*. 2009. S.l.: s.n..
25. Hedong, Yang. 2012. *Estimating Completeness of Event Logs*. s.l. : BPMcenter.org, 2012 , 2012. BPM reports ; 1204.
26. Heijden, Van Der. 2012. Master of Science in Operations Management and Logistics. *Master of Science in Operations Management and Logistics*. [En línea] THC, 2012. [Citado el: 25 de 10 de 2015.] [http://alexandria.tue.nl/extra2/afstversl/tm/Van\\_der\\_Heijden\\_2012.pdf](http://alexandria.tue.nl/extra2/afstversl/tm/Van_der_Heijden_2012.pdf).
27. Huanca, Daniel. 2011. Herramientas de Prueba de Software. [En línea] 2011. [Citado el: 14 de 5 de 2016.] <http://herrorsoft.zxq.net>.
28. IEEE Task Force On Process Mining. 2011. Manifiesto sobre Minería de Proceso. [En línea] 2011. [Citado el: 10 de 11 de 2015.] <http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:pmm-spanish-v1.pdf>.
29. Kaisler, Stephen H. 2005. *Software Paradigms*. New Jersey : Hoboken Sons, Inc, 2005. ISBN-0471483478.
30. Larman, Craig. 2005. *UML y Patrones. 2da Edición*. Argentina : Prentice Hall, 2005. ISBN 84-205-3438-2.
31. Martín, Juan. 2012. CheeksApp. [En línea] 19 de 9 de 2012. [Citado el: 11 de 12 de 2015.] <http://www.checkapps.net/2012/09/eventifier-recopilar-archivos-medios-sociales-eventos.html>.
32. Medina, Lilian Valdés y Méndez, Lian Jaime Ramayo. 2015. *Componente para la extracción de registros de eventos en formato XES del Sistema de Gestión de Ensayos Clínicos (SIGEC)*. La Habana : Universidad de las Ciencias Informáticas, 2015. S/N.
33. Orellana, A, Larrea, O.U y Pérez, D. 2015. *Generador de registros de eventos para el análisis de procesos en un sistema de información hospitalaria*. La Habana : CubaSalud2015, 2015.
34. Paradigm, Visual. 2012. Visual Paradigm. [En línea] 2012. [Citado el: 12 de 12 de 2015.] <http://www.visual-paradigm.com>.
35. Patrón Modelo-Vista-Controlador. Romero, Yenisleidy Fernández y González, Yanette Díaz. 2012. 1, La habana : Revista Telem@tica, 2012, Vol. 11. ISSN 1729-3804.
36. Peregrino, Carlos Alberto Torres y Alvarez, Héctor David Peguero. 2013. *Módulo para el registro y transformación de trazas de eventos a formato XES*. La Lisa, Habana : Universidad de las Ciencias Infromaticas, 2013.

37. **PostgreSQL. 2013.** PostgreSQL. [En línea] PostgreSQL, 2013. [Citado el: 12 de 1 de 2016.] <http://www.postgresql.org/docs/9.2/interactive/index.html>.
38. **Pressman, Roger. 2006.** *Ingeniería Software. Un enfoque práctico*. La Habana : Félix Varela, 2006. ISBN: 978-607-15-0314-5.
39. **Ramírez, Iván. 12.** Softonic, Guía de Software mas completa del mundo. [En línea] Softonic, 22 de 10 de 12. [Citado el: 16 de 1 de 2016.] [netbeans-ide.softonic.com](http://netbeans-ide.softonic.com).
40. **Roldan, Carlos Santana. 2012.** CodeJobs. [En línea] 2012. [Citado el: 30 de 4 de 2016.] <https://www.codejobs.biz/es/blog/2012/09/25/notacion-hungara>.
41. **Rosales, Yadriel Castro. 2013.** *Propuesta de herramienta para pruebas de rendimiento de carga y estrés a servidores streaming*. Habana : Universidad de las Ciencias Informáticas, 2013. S/N.
42. **Ruz, Raul Castro. 2011.** *Instrucción No1. Del Presidente de los Consejos de Estado y de Ministros*. La Habana : s.n., 2011.
43. **Sánchez, Mario Domínguez. 2006.** *Nómadas Revista Crítica de Ciencias Sociales y Jurídicas*. [En línea] 7 de 2006. [Citado el: 11 de 11 de 2015.] <http://pendientedemigracion.ucm.es/info/nomadas/8/mdominguez.htm>. ISSN 1578-6730.
44. **Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Lisa, Habana : Universidad de las Ciencias Informáticas, 2015. 1.2.
45. **Sanz, Emilio. 2010.** Consultores Documentales. [En línea] 12 de 5 de 2010. [Citado el: 10 de 1 de 2016.] <http://sorprendemos.com/consultoresdocumentales/?p=1142>.
46. **software, Pruebas de. 2010.** Pruebas de software. [En línea] 2010. [Citado el: 14 de 5 de 2016.] <http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
47. **Sommerville, I. 2005.** *Ingeniería del software. Séptima edición*. Madrid : S.I.: PEARSON EDUCACIÓN, S.A., 2005. ISBN 84-7829-074-5.
48. **Standard, An Introduction to the XES. 2010.** Flux Capacitor. [En línea] 2010. [Citado el: 11 de 11 de 2015.] <https://fluxicon.com/blog/2010/09/intro-to-xes/>.
49. **Technology., Eindhoven University of. 2014.** Process Mining. [En línea] 2014. [Citado el: 20 de 11 de 2015.] <http://www.processmining.org/prom/start>.
50. **Torres, Liudmila Rodríguez. 2012.** *Caracterización del proceso de planificación empresarial y su práctica en Cuba*. Moa, Holguin : Ciencia & Futuro, 2012.
51. **Verbeek, H.M.W., y otros. 2010.** *XES, XESame, and ProM6*. The Netherlands : Technische Universiteit Eindhoven, 2010. S.I.: s.n.
52. **Vesterinen, Konsta. 2012.** Doctrine. [En línea] 2012. [Citado el: 15 de 1 de 2016.] <http://www.doctrine-project.org/>.