



Universidad de las Ciencias Informáticas

Facultad 6

Trabajo de diploma para optar por el título

Ingeniero en Ciencias Informáticas

**MÓDULO DE REGISTRO Y TRANSFORMACIÓN DE TRAZAS DE EVENTOS PARA LA
VERSIÓN 2.0 DE SAGI**

Autor: Yasmany Sosa Pérez

Tutor: Ing. Vania Elena Yanes León

Ing. Yosbel Fonseca Mendoza

La Habana, junio de 2015.

“Año 57 de la Revolución”

“He fallado una y otra vez en mi vida, por eso he conseguido el éxito”

Michael Jordan

DECLARACIÓN DE AUTORÍA

Declaro ser el único 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 2015.

Yasmany Sosa Pérez

Firma del autor

Ing. Vania Elena Yanes León

Firma de la tutora

Ing. Yosbel Fonseca Mendoza

Firma del tutor

DATOS DE CONTACTO:

Tutora: Ing. Vania Elena Yanes León: Especialista B en Ciencias Informáticas. Diseñadora y Analista de Reportes del proyecto Elecciones.

Email: veyanes@uci.cu

Tutor: Ing. Yosbel Fonseca Mendoza: Recién Graduado en Adiestramiento. Desarrollador del proyecto SIGE.

Email: yfons@uci.cu

Autor: Yasmany Sosa Pérez

Universidad de las Ciencias Informáticas

La Habana, Cuba

Email: ysosa@estudiantes.uci.cu

AGRADECIMIENTOS:

Es mi deseo, como sencillo gesto de gratitud, agradecer:

A todas mis amistades que tanto me han ayudado durante tantos años de estancia en la UCI a todos les debo mucho, sin mencionar nombre porque siempre se me va quedar alguien fuera, Gracias.

A mis amigos y compañeros de los grupos 6105, 6205, 6202, 6305, 6405, 6505 gracias por soportarme, por la ayuda que me dieron siempre y por los muchos momentos buenos que compartimos.

A mis tutores gracias por la ayuda, paciencia y empeño que tuvieron conmigo y con este documento para que quedara lo mejor posible.

Yosbel gracias por todo lo que me ayudaste antes, gracias por ser mi amigo, mi tutor, mi dúo de tesis, mi oponente y mi tribunal.

A mi profe Yezenia gracias por los tantos repasos a ti te debo llegar a 5to año sin mundiales, sino es por ti no sé qué me hubiese hecho, mil gracias por todo.

*A mi familia que siempre me apoyo para que yo estuviese ahí, a mis abuelos por su apoyo y por el siempre buen consejo de **Estudia** cada vez que venía para la UCI. A mis padres que sin importar el que siempre me han apoyado incondicionalmente, gracias por dejarme equivocarme y por dejarme decidir mi camino, se lo mucho que se han sacrificado para que yo estuviese ahí, por eso más que para mí esto es para ustedes.*

A todos, ¡MUCHAS GRACIAS!

DEDICATORIA:

Dedicado a todos los que me ayudaron a lograr mi sueño a mi familia pero muy especialmente para mi mamá esto es por ti y para ti.

Yo.

RESUMEN

La Minería de Proceso es una disciplina de investigación que proporciona técnicas para descubrir, monitorear y mejorar los procesos en una variedad de dominios de aplicación. Las organizaciones dedican mucho tiempo y esfuerzo a analizar sus procesos en aras de mejorar su desempeño organizacional y operacional. Las trazas que registra el Sistema Automatizado de Gestión de Información (SAGI) carecen de nociones de procesos impidiendo la creación de modelos de procesos utilizando técnicas de Minería de Procesos. Esta investigación se enfoca en la creación de un Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI. Con el desarrollo del módulo se obtienen trazas de alta calidad, aptas para su uso en estudios de Minería de Procesos, para descubrir, monitorear y mejorar los procesos que maneja la ONEI.

Palabras clave: Minería de Procesos, Modelos de procesos, Traza de eventos.

ABSTRACT

Process Mining research is a discipline that provides techniques to discover, monitor and improve processes in a variety of application domains. The organizations devoted much time and effort to analyze their processes in order to improve its organizational and operational performance. The traces that records the Automated Information Management System (SAGI) lack notions of preventing processes modeling processes using Process Mining techniques. This research focuses on the creation of a registry and Transformation Module Event Trace for the version 2.0 of SAGI. With the development of high-quality trace module, are obtained suitable (for use in studies of mining processes), to discover, monitor and improve the processes that manage the ONEI.

Keywords: Mining Process, Process models, Trace Events.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS.....	7
1.1- Principales conceptos	7
1.2- Desempeño de la Minería de Procesos desde registro de eventos.....	8
1.3- Niveles de madurez para los registros de eventos.....	9
1.4- Elementos teóricos de la extracción de trazas de proceso desde DAIS.....	10
1.5- Atributos del registro de eventos y extensiones de XES.....	11
1.6- Herramientas para la extracción de trazas de los sistemas de información	13
1.6.1- <i>Eventifier</i>	13
1.6.2- <i>XES Mapper (XESame)</i>	13
1.7- Herramientas y tecnologías a utilizar	14
1.8- Metodología de desarrollo del software	17
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA	19
2.1- Modelo conceptual.....	19
2.2- Propuesta de solución.....	21
2.3- Especificación de requisitos	23
2.4- Patrones de Casos de Uso	26
2.5- Diagrama de casos de uso del sistema	27
2.6- Diagramas de clases.....	32
2.7- Diagramas de secuencia.....	34
2.8- Patrones arquitectónicos	36
2.9- Patrones de diseño	37
2.10- Modelo de datos.....	39
2.11- Diagrama de despliegue	41

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS	42
3.1- Diagrama de componentes.....	42
3.2- Estándares de codificación	43
3.3- Pruebas de software	45
3.4- Resultados de las pruebas	52
3.5- Interfaces de la aplicación	53
CONCLUSIONES	57
RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS	59

ÍNDICE DE FIGURAS

Fig. 1: Registro de eventos de los PAIS.....2

Fig. 2: Registro de eventos de los DAIS.....3

Fig. 3: Modelo de Dominio.20

Fig. 4: Evidencia del patrón CRUD total.....26

Fig. 5: Diagrama de Casos de Uso del Sistema.....27

Fig. 6: Diagrama de clases del diseño para el CU Definir Proceso.33

Fig. 7: Diagrama de secuencia correspondiente al escenario Adicionar Proceso.....35

Fig. 8: Representación del Patrón Modelo Vista Controlador.36

Fig. 9: Evidencia del patrón Bajo Acoplamiento.37

Fig. 10: Evidencia del patrón Alta Cohesión.....37

Fig. 11: Evidencia del patrón Experto.....38

Fig. 12: Evidencia del patrón Creador.38

Fig. 13: Evidencia del patrón Controlador.39

Fig. 14: Modelo de datos.....40

Fig. 15: Diagrama de Despliegue.41

Fig. 16: Diagrama de Componentes.....43

Fig. 17: Longitud de la línea y Comentarios.45

Fig. 18: Colocación de llaves.45

Fig. 19: Resultados de las pruebas de caja negra.....52

Fig. 20: Interfaz de usuario Adicionar Proceso.....53

Fig. 21: Interfaz de usuario Configurar Proceso.54

Fig. 22: Registro de evento en formato XES.55

ÍNDICE DE TABLAS

Tabla 1: Descripción textual del CU Definir Proceso.28

Tabla 2: Descripción de las variables.....47

Tabla 3: Sección 1: Caso de prueba Adicionar Proceso.....48

Tabla 4: Sección 2: Caso de prueba Modificar Proceso.49

Tabla 5: Sección 3: Caso de prueba Eliminar Proceso.....51

Tabla 7: Sección 4: Caso de prueba Listar Proceso.....52

INTRODUCCIÓN

Cualquier organización que desee permanecer en el tiempo debe adaptarse a los cambios que surgen constantemente y para ello, tiene que adaptar sus procesos a los mismos. Para guiar estos cambios en las organizaciones, con el objetivo de disponer de una metodología para la realización de sus actividades, definir responsables para las mismas, así como establecer objetivos e indicadores para su control se recurre a la Gestión de Procesos de Negocio BPM¹.

Para realizar una correcta gestión de los procesos en una organización es necesario realizar un análisis profundo con el objetivo de mejorar el desempeño y la optimización de los procesos de negocio. Este análisis depende en gran medida de la existencia de modelos de procesos de alta calidad, lo cual ha demostrado que existe una falta de alineación entre los modelos hechos manualmente y la realidad. La Minería de Procesos se centra en resolver estos problemas realizando una conexión directa entre los modelos y los datos de eventos de los procesos (Van Der Aalst, 2011).

El punto de partida para cualquier actividad de Minería de Procesos son los eventos registrados. Cada evento representa una actividad (un paso bien definido en algún proceso) y se relaciona a un caso particular (una instancia de proceso). Cada instancia de proceso tiene asociado un registro de eventos que almacena de cada evento: el recurso (persona o dispositivo) que ejecuta o inicia la actividad, la marca de tiempo, o elementos de datos registrados (unidad de observación de un formulario) (IEEE Task Force On Process Mining, 2011). La calidad de un resultado de Minería de Procesos en gran medida depende de este registro de eventos. Las trazas registradas (registro de eventos) deben tener una cantidad de información mínima para que la aplicación de técnicas de Minería de Procesos sea viable y se obtenga un resultado útil.

En la actualidad se desarrollan dos tipos de sistemas de información según la forma en que generan sus trazas. Por un lado se encuentran los Sistemas de Información Conscientes de los Procesos (PAIS²) y por otra parte, los Sistemas de Información Conscientes de los Datos (DAIS³). Los PAIS gestionan y ejecutan

¹ Del inglés: Business Process Management.

² Del inglés: Process-aware information systems.

³ Del inglés: Data-aware information system.

los procesos operativos relacionados con las personas, aplicaciones y/o fuentes de información sobre la base de modelos de procesos. Estos sistemas generan trazas de gran calidad pues se almacenan las instancias de procesos con sus respectivos eventos (ver Fig. 1). En cambio los DAIS guardan las trazas secuenciales sin guardar las instancias de los procesos por lo que no se conoce cuáles eventos pertenecen a qué instancia de proceso (ver Fig. 2). A causa de esto dichas trazas poseen poca calidad y la información almacenada es pobre.

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	...

Fig. 1: Registro de eventos de los PAIS.

event id	properties				
	timestamp	activity	resource	cost	...
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	...
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	...

Fig. 2: Registro de eventos de los DAIS.

Debido a la forma en que guarda sus trazas, dentro de los DAIS se encuentra el Sistema Automatizado de Gestión de Información (SAGI) para la Oficina Nacional de Estadísticas e Información (ONEI), desarrollado en el Centro de Tecnologías de Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas (UCI). Este tiene como objetivo principal la automatización de los procesos de gestión estadística en Cuba, desglosándose en objetivos específicos asociados a los procesos de la gestión de la información estadística.

Junto a las exigencias de la ONEI el sistema guía el proceso de diseño y captura de formularios en un organismo, controla la seguridad de la información y permite la visualización de datos a directivos a través de reportes así como la gestión de encuestas. Las funcionalidades de SAGI están divididas por los siguientes módulos: Gestión de la configuración, Gestor de plantillas, Entrada de datos, Herramientas, Generador de reportes, Administración y Seguridad.

Dentro del módulo de Seguridad se gestionan las trazas de Datos, Acción, Excepción y Rendimiento pero ninguna de ellas contiene la información suficiente para realizar análisis utilizando técnicas de Minería de Procesos. Por otra parte los métodos de detección y control, chequeo de conformidad y mejora de

procesos que se emplean en SAGI son los tradicionales, basados en largas entrevistas y reuniones de grupo, con el fin de tratar de entender cómo estos funcionan. Los resultados de estos métodos generalmente: no contemplan el comportamiento propio del proceso, son subjetivos y habitualmente aproximaciones de la realidad. En la práctica, estos análisis suelen acabar representando cómo debería llevarse a cabo el proceso y no como se está ejecutando en realidad. Estas situaciones pueden traer consigo:

- Ineficiente gestión del tiempo necesario para realizar determinada actividad.
- Contradicción entre la definición de la ejecución de los procesos y lo que sucede en realidad.
- Modelos inexactos de los procesos.

Estas situaciones no favorecen la toma de decisiones.

La ONEI, no está ajena a estas situaciones por lo que decidió aprovechar los datos de eventos generados por SAGI en una forma significativa, por ejemplo, para proveer un mejor entendimiento de sus procesos de negocio, identificar cuellos de botella, anticipar problemas, registrar violaciones de políticas y simplificar procesos.

Analizando la situación anterior se identifica como problema de la investigación: ¿Cómo garantizar que las trazas generadas por el SAGI contengan información suficiente para realizar análisis de Minería de Procesos?

Teniendo en cuenta el problema de la investigación se ha definido como objeto de estudio: la Minería de Procesos.

El estudio se centra en el campo de acción definido por: el registro de trazas para la Minería de Procesos.

Con el propósito de solucionar los problemas encontrados se ha determinado como objetivo de la investigación: Desarrollar un Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI que permita realizar análisis de Minería de Procesos en la ONEI.

Para dar cumplimiento al objetivo, se plantearon las siguientes preguntas científicas:

1. ¿Cuáles son los fundamentos teóricos que se necesitan para obtener un Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI, que cumpla los objetivos propuestos?

2. ¿Qué tecnologías, herramientas y metodología de desarrollo de software permiten la creación del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI?
3. ¿Cómo validar el correcto funcionamiento del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI?

A su vez, se definieron las siguientes tareas de la investigación:

1. Elaboración de los elementos introductorios para fundamentar el tema de investigación.
2. Análisis de los principales conceptos y trabajos relacionados con la definición de XES⁴, sus extensiones y análisis de Minería de Procesos para asumir postura de investigación.
3. Estudio de herramientas existentes que realizan la extracción de registros de eventos en formato XES para el desarrollo del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.
4. Revisión bibliográfica sobre las herramientas, tecnologías y metodologías, para fundamentar su selección.
5. Definición de los requisitos para gestionar las necesidades del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.
6. Confección del modelo de casos de uso del sistema del módulo para visualizar y especificar el comportamiento del sistema.
7. Elaboración del modelo de diseño para describir cómo se debe implementar el Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.
8. Implementación del módulo para el registro y transformación de trazas.
9. Realización de pruebas funcionales para validar el correcto funcionamiento del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.

Para el desarrollo de la presente investigación se aplicaron los siguientes métodos:

Métodos teóricos

Analítico – Sintético: Se emplea principalmente durante la revisión del estado del arte, con el fin de conocer los principales conceptos de la investigación. Se estudia sobre la actualidad del tema buscando los resultados obtenidos en el objeto de estudio identificado y la definición de las herramientas

⁴ Del inglés: eXtensible Event Stream.

seleccionadas para el desarrollo del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.

Métodos Empíricos

Entrevista: Se realizaron entrevistas a especialistas y expertos del área que sirvieron de guía en el desarrollo de la investigación.

Encuesta: Se realizaron encuestas de satisfacción con el cliente, para complementar los resultados obtenidos en las pruebas y conocer su opinión con respecto a los resultados alcanzados.

Con el desarrollo del módulo se esperan los siguientes resultados:

- Definir y mejorar las trazas a utilizar en el Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI hacia el formato XES.
- Obtener el fichero XES para permitir la utilización de técnicas de Minería de Procesos.

Este documento se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS: En este capítulo 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, características y viabilidad. Se propone la utilización de una serie de herramientas y tecnologías a emplear en el desarrollo de la solución.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA: En el presente capítulo se hace el análisis y diseño de la propuesta del sistema. En el análisis se especifican los requisitos funcionales y no funcionales, las descripciones de los casos de uso y el diagrama de casos de uso del sistema. En el diseño se generan los artefactos correspondientes a esta etapa, se describe el patrón arquitectónico y los patrones de diseño utilizados en la implementación, se describen los diagramas de clases del diseño y el modelo de despliegue.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA: En este capítulo se convierten los elementos del diseño en elementos de implementación y se presenta la propuesta de solución del sistema. Se describen los estándares de codificación utilizados en la nomenclatura del código y se valida el correcto funcionamiento del sistema mediante los casos de prueba aplicados al módulo.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Introducción

En este capítulo se realiza un estudio de los conceptos fundamentales relacionados con la extracción de trazas de proceso en sistemas de información. Adicionalmente se detallan los aportes teóricos de los autores que han abordado este tema. Se analizan las herramientas existentes para la extracción de trazas de proceso generadas por DAIS. Finalmente se describen las herramientas, tecnologías y la metodología para el desarrollo de la solución.

1.1- Principales conceptos

Un **Sistema de Información** es un tipo particular de sistema de trabajo que usa tecnología de la información para captar, transmitir, almacenar, recuperar, manipular o mostrar información. Los sistemas de información se dividen en dos ramas, DAIS y PAIS. Los PAIS gestionan y ejecutan procesos operacionales involucrando personas, aplicaciones y/o fuentes de información basados en modelos de procesos (Dumas, y otros, 2005).

Se considera como **proceso** a un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados (ISO, 2000). Una **Actividad** es un paso bien definido en el proceso (IEEE Task Force On Process Mining, 2011). Está compuesta por tareas y se entiende como **tarea** a una actividad atómica que está incluida dentro de un proceso. Por su parte un **Sub-Proceso** es un proceso que está incluido en otro proceso (OMG, 2006).

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, por ejemplo, el inicio, conclusión o cancelación de una actividad (IEEE Task Force On Process Mining, 2011). El **registro de eventos** es la colección de eventos utilizada como entrada para la Minería de Procesos. No necesitan ser almacenados en un archivo de registro por separado (por ejemplo, los eventos pueden estar dispersos en diferentes tablas de bases de datos).

Minería de Procesos

Se entiende por **Minería de Procesos** a la disciplina de investigación que permite descubrir, monitorear y mejorar los procesos reales a través de la extracción de conocimiento de los registros de eventos ampliamente disponibles en los actuales sistemas de información (IEEE Task Force On Process Mining, 2011).

El **Descubrimiento de procesos**, la **Verificación de conformidad** y el **Mejoramiento de modelos**, son los tres tipos básicos de Minería de Procesos. El descubrimiento de procesos permite obtener un modelo de proceso a partir de un registro de eventos. A su vez la Verificación de Conformidad analiza si la realidad (según consta en un registro de eventos) se ajusta al modelo y viceversa. Su objetivo es detectar las discrepancias y medir su gravedad. Por otra parte el Mejoramiento de Modelos permite extender o mejorar, a partir de la información extraída de un registro de eventos, los modelos. Por ejemplo, se pueden identificar cuellos de botella representando un registro de eventos en un modelo de proceso, mientras se examinan las marcas de tiempo (IEEE Task Force On Process Mining, 2011).

1.2- Desempeño de la Minería de Procesos desde registro de eventos

Para demostrar este desempeño se analiza un registro de eventos de un proceso de la vida real, tomado de una entidad financiera holandesa, utilizando técnicas de Minería de Procesos. En particular, se utilizó la técnica de alineación para obtener información sobre el flujo de control y el rendimiento de la ejecución del proceso. Se demuestra que las alineaciones entre los registros de eventos y modelos de procesos descubiertos, revelan las desviaciones que se producen con frecuencia. Estos descubrimientos pueden ser explotados para reparar los modelos de procesos originales, reflejando mejor la realidad. Todos los análisis de este trabajo se llevan a cabo mediante plugins de la aplicación de código abierto para la Minería de Procesos ProM (Adriansyah, y otros, 2012).

Registros de eventos o trazas de proceso

Para formalizar la estructura de los registros de eventos a utilizar en la Minería de Procesos se han definido dos estándares: (Lenguaje de Marcado Extensible para la Minería) MXML⁵ y (Flujo de Eventos Extensible) XES. MXML es un formato basado en XML para el intercambio de registros de eventos. MXML fue el primer estándar que surgió en el 2003 y fue adoptado por la herramienta de Minería de Procesos

⁵ Del inglés: Mining Extensible Markup Language.

ProM. MXML establece una notación estándar para almacenar fechas, recursos y tipos de transacciones. XES en el 2010 reemplaza a MXML como el nuevo formato para Minería de Procesos independiente de la herramienta (IEEE Task Force On Process Mining, 2011). Es un estándar basado en experiencias prácticas de MXML, menos restrictivo y extensible. Tiene como propósito ofrecer un formato de intercambio de registros de eventos entre herramientas y dominios de aplicaciones. Su objetivo principal es permitir el análisis de los procesos operativos basados en sus registros de eventos (Günther, 2009).

1.3- Niveles de madurez para los registros de eventos

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 semántica. La confiabilidad consiste en que debería ser seguro asumir que los eventos registrados realmente ocurrieron y que sus atributos son correctos. La completitud se relaciona a que los registros de eventos deberían ser completos, dado un determinado contexto no puede faltar ningún evento. Además, cualquier evento registrado debe tener una semántica bien definida. La combinación de los tres aspectos de calidad se refleja en los niveles de madurez definidos por la IEEE 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: Hay un nivel de garantía de que los eventos registrados concuerdan 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).

1.4- Elementos teóricos de la extracción de trazas de proceso desde DAIS

Los sistemas de información almacenan los datos producidos en su base de datos pero se distorsionan diferentes aspectos y se descuida la naturaleza basada en eventos de la ejecución de los procesos. Por esta razón el descubrimiento de procesos generalmente comienza a partir de un registro de eventos (Rodríguez, y otros, 2012).

Un registro de eventos puede ser visto como la secuencia de eventos $E = \langle e_1, e_2, e_3, \dots, e_m \rangle$ donde: $e_i = \langle id, tname, pname, piid, ts \rangle$.

e_i es un evento de una instancia de un proceso, **id** es el identificador del evento, **tname** es el nombre de la tarea a que el evento está asociado, **pname** es el nombre del tipo de proceso, **piid** el identificador de la instancia del proceso y **ts** la fecha del evento.

Reconstruir un registro de eventos significa decidir cuándo inferir la existencia de un evento en la base de datos y llenar cada uno de los atributos del evento con valores relevantes. Estos valores pueden ser extraídos de la base de datos o pueden ser especificados por el experto del dominio. Específicamente para el atributo id, asignar un identificador a un evento, significa reconocer la existencia de dicho evento. El valor de pname se puede obtener solamente del experto en el dominio que es quien sabe cuál es el proceso que se trata de identificar. El piid se utiliza para agrupar los eventos en instancias de procesos. El atributo ts se utiliza para ordenar cronológicamente los eventos siendo este un requerimiento esencial para el descubrimiento de procesos (Rodríguez, y otros, 2012).

Al ser los eventos grabaciones atómicas de actividades ejecutadas, estos no tienen duración. Para almacenar los diferentes estados que una actividad puede tener, cada evento debe ser de un cierto tipo. Los tipos de eventos más usados son el “iniciado” y el “completado” para indicar el inicio y la completitud de una actividad respectivamente. Es posible almacenar eventos de un solo tipo el cual generalmente es “completado”. Para ciertos tipos de análisis es imperativo adicionar el tipo “iniciado” como cuando es necesario medir el tiempo de procesamiento. Los tipos de eventos “iniciado” y “completado” se usan para distinguir entre el tiempo de espera y el tiempo de procesamiento.

1.5- Atributos del registro de eventos y extensiones de XES

El registro de eventos contiene un registro de elementos como su raíz el cual contiene todas las trazas. Este registro de elementos también puede contener atributos. Como el registro de elementos solo se crea una vez, el impacto de incluir muchos atributos en el registro es mínimo (J.C.A.M. , Buijs, 2010). Por otra parte es de importancia incluir información relevante describiendo el contenido del registro de eventos y su origen. Los siguientes atributos deben ser tomados en consideración para su adición en el registro de elementos:

Nombre de Proceso: El nombre del proceso al cual el registro le graba su ejecución.

Fuente de Datos: Una descripción del Sistema de Información del cual se extrae el registro de eventos.

Organización Fuente: El nombre de la organización que provee los datos.

Descripción: Una breve descripción del contenido del registro de eventos.

Versión: Un identificador para diferenciar versiones de registros de eventos.

Autor: Nombre y detalles de contacto del que definió la conversión.

Proyecto de Minería de Procesos: Una referencia del Proyecto de Minería de Procesos o el propósito del registro de eventos.

Extensiones de XES

La extensión de XES más importante es **concept** la cual especifica un nombre para el registro de eventos, la traza y los eventos. Los nombres de las trazas deberán incluir algún identificador único y los nombres de los eventos deben proveer el nombre de la actividad ejecutada. La extensión **concept** define un atributo **instance** para eventos el cual representa un identificador de la instancia de la actividad la cual generó el evento. Incluir este atributo en el registro de eventos, permite enlazar los eventos con lo almacenado en la fuente de datos para futuros análisis.

Otra extensión importante es "time". La cual 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 preferentemente al nivel de segundos y en algunos casos de milisegundos. Muchos eventos pueden ser ejecutados dentro del mismo segundo, por esto es la necesidad de la precisión de milisegundos. Esto es necesario pues de esta forma los eventos tendrán un único instante de tiempo dentro de la traza.

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”. En la mayoría de los casos con usar el tipo “iniciado” y “completado” es suficiente pero existen otros tipos de eventos. 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 se pueden detectar a un grupo de actores desempeñando tareas similares. Para lograr la visualización 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 “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.

La última de las cinco extensiones estándares de XES analizar es “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 un entendimiento más a fondo de un evento. Otro ejemplo pudiese referirse a una ontología de usuarios en el atributo “resource” de la extensión “organizational” (Günther, 2009).

Pueden existir otras extensiones y atributos que definan los atributos relacionados de 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 como “concept: name” y “time: timestamp”, deben ser especificados en el registro de eventos para que la Minería de Procesos se pueda aplicar exitosamente.

1.6- Herramientas para la extracción de trazas de los sistemas de información

Antes de iniciar el desarrollo de un proyecto de software es muy importante realizar un estudio detallado de los sistemas existentes que realizan la extracción de trazas de proceso de los sistemas DAIS. A pesar de descuidar la naturaleza basada en eventos de los datos, si se pueden relacionar los atributos de los eventos a campos en la base de datos, se podrán realizar estudios utilizando técnicas de Minería de Procesos. Las herramientas estudiadas se describen a continuación.

1.6.1- Eventifier

Eventifier es una herramienta que ayuda en la reconstrucción de un log de eventos desde bases de datos operacionales donde existen evidencias de la ejecución de instancias de procesos. Primeramente, el experto en el dominio identifica los eventos en la base de datos operacional, los ordena y le asocia datos. Todas estas actividades están soportadas por el Extractor de eventos el cual ayuda al experto del dominio de una manera interactiva. El resultado de este primer paso es una serie de eventos los cuales todavía no están relacionados. La correlación es dirigida por el Componente correlacionar, el cual ayuda al experto del dominio de manera interactiva a identificar los mejores atributos y condiciones para reconstruir trazas de proceso. El resultado del proceso completo es un registro de eventos listo para aplicársele Minería de Procesos (Rodríguez, y otros, 2012).

1.6.2- XES Mapper (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 pues 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 y Mysql. 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 Procesos (J.C.A.M. , 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. XESame posee ciertas limitaciones, pues al estar desarrollada en Java para su conexión a base de datos requiere un driver ODBC⁶. El driver ODBC tiene conflictos con el uso de funciones SQL en la definición de la conversión. La principal desventaja de XESame es que sólo es aprovechable si existe alguna evidencia de la ejecución de un proceso (J.C.A.M. , Buijs, 2010).

Concluido el análisis realizado sobre los sistemas existentes que realizan la extracción de trazas de proceso de los sistemas DAIS se concluye lo siguiente:

- Ambas herramientas, por la condición de ser una aplicación desktop, desarrollada en Java, no pueden ser integrada a SAGI como un módulo más.
- Estas herramientas no son aprovechables pues no existe, en la base de datos de SAGI, evidencia de la ejecución de al menos un proceso. Además esta condición limita la usabilidad en sistemas de información debido a la variación de los datos.

1.7- Herramientas y tecnologías a utilizar

El desempeño de un proyecto está antecedido por la investigación de las tecnologías a utilizar, así como las ventajas y desventajas de su aplicación. A continuación se describe la metodología, herramientas y tecnologías a utilizadas en el desarrollo del módulo, así como sus principales características.

Apache 2.2

Es un servidor web modular, multiplataforma y de código abierto usado para compartir páginas web estáticas y dinámicas hacia Internet. Su principal objetivo es proporcionar servicios HTTP en sincronización con los estándares HTTP actuales de forma segura y eficiente. Su adaptabilidad, robustez y estabilidad lo han hecho mundialmente popular (The Apache Software Foundation, 2015).

⁶ Del inglés: Open DataBase Connectivity.

Se utilizó como servidor web puesto que consta de diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web. Tiene la capacidad de crear sitios privados seguros, es uno de los servidores web más implementado entre los distintos que ofrecen servicios a internet y es el servidor que define SAGI para su ejecución dentro de los requisitos no funcionales.

Symfony 2.5

Symfony es un framework PHP diseñado para optimizar el desarrollo de aplicaciones web, proporcionando herramientas para agilizar aplicaciones complejas. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Integra plenamente uno de los frameworks ORM existentes para PHP llamado Doctrine, el cual es el encargado de la comunicación con la base de datos, permitiendo un control total de los datos con relación al motor de base de datos, sin importar cuál sea, la mayoría de ellos se encuentran soportados por Doctrine. Las sentencias SQL no son generadas por el programador sino por el mismo Doctrine. Symfony automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (Eguiluz, 2011).

Se utilizó como framework de desarrollo Symfony, puesto que contiene variedad de herramientas y clases para conseguir acortar el tiempo de desarrollo de aplicaciones web complejas. Internamente Symfony crea una nueva estructura de directorios para los proyectos, desaparecen los plugins y se sustituyen por un elemento mucho más potente llamado bundle. Posee buenas prácticas de PHP y de diversos frameworks como Django, Spring, Java entre otros. Otro aspecto importante es que la versión 2.0 de SAGI se desarrollará utilizando este framework por lo que para la integración sólo habría que instalar el bundle que corresponde al módulo implementado.

ExtJS 4.2

ExtJS es una biblioteca de Javascript para el desarrollo de aplicaciones web, utiliza técnicas AJAX, DHTML además de flexibilizar el manejo de componentes como el DOM es compatible con la mayoría de los navegadores. Incluye un conjunto de controles para el desarrollo de interfaces en los desarrollos web (Sencha Ext JS, 2015).

Se utilizó ExtJS puesto que permite crear aplicaciones complejas utilizando componentes predefinidos, provee una experiencia consistente sobre cualquier navegador y es la biblioteca que utiliza actualmente el SAGI.

Netbeans 8.0

Es un ambiente integrado de desarrollo de código abierto para desarrolladores de software usando mayormente Java o algún otro lenguaje de programación como PHP, Python, Ruby. Puede ser usado para la creación de cualquier tipo de aplicaciones de escritorio, web y mobile. También permite el desarrollo de aplicaciones usando conjuntos de componentes modulares de software que han logrado expandir sus funcionalidades con numerosos plugins (NetBeans IDE, 2015).

Se utilizó como entorno de desarrollo NetBeans, pues brinda soporte para Symfony lo que ayuda a los desarrolladores en la elaboración de aplicaciones complejas. Es apoyado por una gran comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación. Además sus actualizaciones y complementos son fáciles de instalar, consume pocos recursos con pocos módulos activados y aun así funciona correctamente.

Visual Paradigm 8.0

Visual Paradigm es una plataforma de modelado diseñado para apoyar a los arquitectos de sistemas, desarrolladores y diseñadores para acelerar el proceso de análisis y diseño de aplicaciones empresariales. La tecnología de modelado UML⁷ facilita el diseño, visualización, gestión y documentación de los artefactos generados durante el proceso de desarrollo de software (Visual Paradigm, 2015). Para el sistema que se va a desarrollar se utilizó como herramienta CASE Visual Paradigm puesto que permite la realización de un buen diseño para el proyecto. Permite la construcción de sistemas de software de forma fiable a través de la utilización del enfoque orientado a objetos. Visual Paradigm permite realizar ingeniería tanto directa como inversa y está diseñado para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas y arquitectos.

Se decidió usar Visual Paradigm debido a todas las características mencionadas anteriormente. Además, es una herramienta CASE profesional, la cual soporta todo el ciclo de vida de desarrollo de software.

⁷ Del inglés: Unified Modeling Language.

También se tuvo en cuenta que esta es una de las herramientas usadas por la Universidad y DATEC. La misma posee una licencia comercial y la UCI cuenta con esta para su uso.

PostgreSQL 9.1

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. La estabilidad, potencia, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez el sistema (PostgreSQL, 2015). Se utilizó como servidor de bases de datos pues presenta buen rendimiento y rapidez a la hora de hacerle una petición al servidor. Además los mecanismos o servicios que brinda utilizan poco procesamiento.

El sistema gestor de base de datos que utiliza SAGI es PostgreSQL por lo que su utilización, evita incompatibilidades y permite un mejor acoplamiento del Módulo de Registro y Transformación de Trazas de Eventos.

1.8- Metodología de desarrollo del software

El desarrollo de software puede ser un reto para los desarrolladores, pues crear uno en el menor tiempo posible y con calidad, puede ser casi imposible si no se cuenta con un proceso que ayude a agilizar su desarrollo. Por tales motivos se utilizan las metodologías de desarrollo de software, estas proponen un proceso disciplinado con el fin de construir un software con calidad.

OpenUP.

Es un proceso de desarrollo de software mínimamente suficiente, es decir, que solamente el contenido fundamental es incluido. Es una metodología ágil, por esto provee una buena opción para muchos tópicos dentro de un proyecto, tales como reducidos equipos de trabajo, conformidad, seguridad o aplicaciones críticas. OpenUP tiene las características esenciales de Unified Process que aplica enfoques iterativo e incremental dentro de un ciclo de vida estructurado. Está basada en casos de uso, manejo de riesgos y tiene una arquitectura centralizada para guiar el desarrollo de software (Balduino, 2007).

Conclusiones del capítulo:

Como resultado del estudio realizado en este capítulo, se concluye que la información que almacenan las trazas generadas por SAGI no es suficiente para realizar estudios utilizando técnicas de Minería de Procesos. El estudio de las herramientas para la extracción de trazas de proceso a partir de bases de datos, permitió identificar que estas dependen de la evidencia de la ejecución previa de los procesos. Esto representa una debilidad pues la evidencia puede ser inconsistente y los datos en ocasiones son eliminados o sufren variaciones. Se seleccionó PHP como lenguaje de programación, Netbeans 8.0 como entorno de desarrollo, Visual Paradigm 8.0 como herramienta de modelado y OPEN UP como metodología para guiar el proceso de desarrollo de software. Para modelar y visualizar los resultados se eligió el lenguaje de modelado UML.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En el presente capítulo se realiza el análisis y diseño a la propuesta del sistema. En el análisis se especifican los requerimientos funcionales y no funcionales, las descripciones de los casos de uso y los actores así como la relación entre ellos mediante el diagrama de casos de uso del sistema. En el diseño se generan los artefactos correspondientes a esta etapa, se describe la arquitectura y se definen los patrones de diseño, se describen los diagramas de clases del diseño y el modelo de despliegue.

2.1- Modelo conceptual

Los principales conceptos y relaciones de la solución se presentan en la Fig. 3. El presente modelo conceptual ilustra el funcionamiento del registro de trazas una vez implementado el Módulo para el Registro y Transformación de trazas de eventos. SAGI es un sistema de gestión estadística que está compuesto por diferentes módulos que generan trazas. Del mismo modo SAGI tiene definido procesos de negocio los cuales poseen eventos asociados. Estas 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 experto de dominio es quien configura los procesos delimitando los datos a capturar.

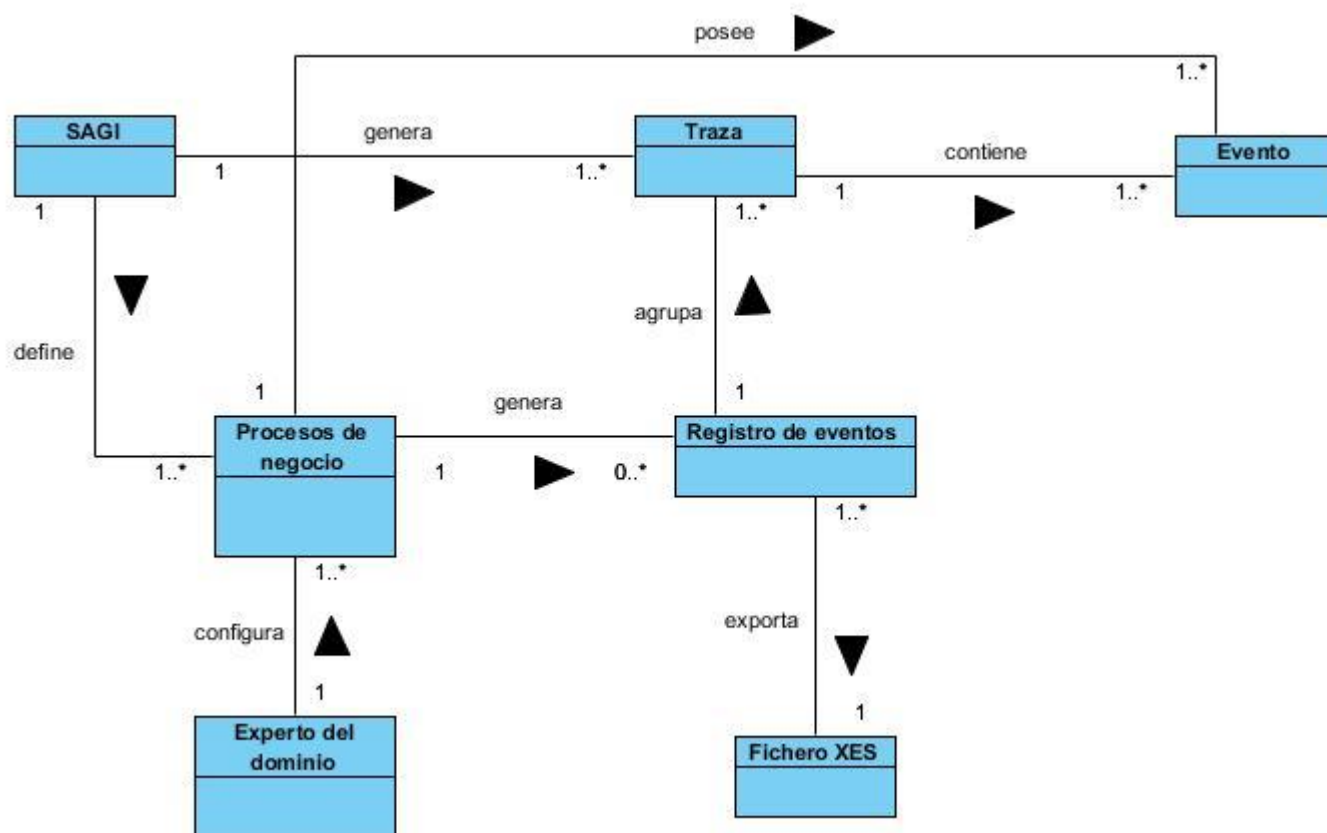


Fig. 3: Modelo de Dominio.

SAGI: Sistema de gestión estadística que está compuesto por los módulos: Gestión de la configuración, Gestor de plantillas, Entrada de datos, Herramientas, Generador de reportes, Administración y Seguridad.

Procesos de negocio: Gestión de Encuestas y Formularios.

Registro de eventos: Es la colección de eventos generados por los procesos de negocio que se utiliza como entrada para la Minería de Procesos. Los mismos pueden ser almacenados en diferentes tablas de bases de datos.

Traza: Almacena información sobre los eventos que ocurren en SAGI.

Evento: Expresa el estado de una actividad por ejemplo, el inicio, conclusión o cancelación de una actividad realizada en los procesos de negocio.

Fichero XES: Formato al cual se exporta el registro de eventos para su uso en análisis de Minería de Procesos.

Experto del dominio: Es quien realiza la configuración de los procesos.

2.2- Propuesta de solución

Como se explicó en el capítulo anterior, las trazas de proceso 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 del registro de trazas de proceso por un experto del dominio a partir de las trazas que generan los módulos de SAGI y su exportación como registro de eventos en formato XES. Para lograr lo anteriormente planteado se decidió crear Historial de datos. El uso del módulo de Historiales resuelve el problema de la pérdida y/o variación de los datos. La información contenida en las trazas, es referente al usuario que realizó los cambios y a las particularidades de los datos alterados. Una vez almacenada toda la información necesaria se procede, a petición del usuario, a la generación de las trazas de proceso. Las trazas de proceso pueden ser exportadas como registro de eventos al formato XES para la realización de futuros análisis utilizando técnicas de Minería de Procesos.

Historial de datos

El módulo de Historiales permite crear tablas, espejo de tablas específicas en la base de datos de SAGI. Esta réplica captura todos los datos de la original. Además, la misma registra la operación realizada sobre los datos (insertar, modificar o eliminar) y el instante de tiempo de dicha operación. Estas tablas espejo persisten en el tiempo y no sufren variaciones en sus datos. El uso del módulo de Historiales permite superar una de las principales deficiencias identificadas en las herramientas XESame y Eventifier.

Para el desarrollo de la solución se determinó la implementación de cinco módulos. Un módulo para la gestión de las conexiones, tres para la configuración del registro de trazas de proceso y uno para la generación de trazas de proceso y administración de los procesos. Además de las funcionalidades que permiten la exportación de trazas de proceso.

Dichos módulos se describen a continuación:

Módulo primero: Gestión de la conexión

Gestiona las conexiones a bases de datos definidas exclusivamente para la definición y configuración de un proceso. Dichas conexiones pueden ser independientes para cada proceso o pueden ser compartidas entre varios procesos.

Módulo segundo: Definición de proceso

Permite la definición de un proceso. Esta definición consiste en la selección de una o más tablas asociadas a la ejecución del proceso, la definición del nombre (único para cada proceso), fuente de datos, descripción del proceso y la selección de la tabla principal del proceso la cual debe poseer el identificador de instancia de proceso.

Módulo tercero: Gestión de eventos

Este módulo permite definir eventos con su nombre (único para cada proceso) y descripción asociados a un proceso determinado.

Módulo cuarto: Configuración de proceso

Este módulo permite la configuración de los datos necesarios para generar las trazas de un proceso previamente definido. La configuración de un proceso consiste en la asociación de los atributos de cada uno de los eventos a columnas de tablas de bases de datos. Estas asociaciones pueden estar bajo ciertas condiciones especificadas. Los atributos de los eventos son las extensiones definidas por el estándar XES anteriormente.

Módulo quinto: Administración de proceso

Administra los procesos previamente definidos y configurados. La administración de un proceso está compuesta por la validación, activación o desactivación de un proceso y generación de trazas de proceso. La validación de un proceso consiste en verificar que todos los atributos de todos los eventos tengan columnas asociadas y que las condiciones especificadas para todos los eventos posean tipos de datos compatibles. La validación de un proceso es un requisito previo para la activación del mismo. La activación de un proceso consiste en la creación de historiales de las tablas cuyas columnas forman parte de los atributos de los eventos y/o las condiciones de los mismos, también registra de la activación del proceso. La desactivación de un proceso consiste en el registro de la desactivación del mismo. La generación de trazas de proceso consiste en construir dichas trazas con los datos de los historiales en conjunto con las trazas de datos teniendo en cuenta el rango de tiempo en que el proceso ha estado activo. Una vez que el proceso ha estado activo al menos una vez, cualquier cambio en la configuración o definición (excluyendo nombre, descripción y fuente de datos) del mismo se refleja como una nueva

versión del proceso. Si un proceso recibe cambios en su configuración, este se invalida y desactiva automáticamente en caso de estarlo. Un proceso se invalida y desactiva automáticamente si recibe cambios en su definición que no incluyan el nombre del proceso, descripción y/o fuente de datos del mismo.

Funcionalidad exportar trazas de proceso a formato XES

Esta funcionalidad permite exportar las trazas de proceso a formato XES previamente generadas por el Módulo de administración de proceso. La exportación permite especificar el proceso al que se le quiere exportar las trazas y la versión del proceso a exportar.

2.3- Especificación de requisitos

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. A continuación se presentan los requisitos separados en funcionales y no funcionales (Pressman, 2005).

Requisitos funcionales

Los requisitos funcionales son las condiciones que el sistema debe alcanzar o cumplir y definen el comportamiento interno del sistema. A continuación se enumeran.

RF1. Listar conexiones: Este sistema deberá ser capaz de listar las conexiones existentes.

RF2. Adicionar conexión: Este sistema deberá brindar la posibilidad de adicionar una conexión.

RF3. Modificar conexión: Mediante este requisito el sistema deberá brindar la posibilidad de modificar una conexión.

RF4. Eliminar conexión: A través de este requisito, el sistema deberá brindar la oportunidad de eliminar una conexión.

RF5. Probar conexión: Este sistema deberá brindar la posibilidad de probar una conexión.

RF6. Listar procesos: Este sistema deberá ser capaz de listar los procesos existentes.

RF7. Adicionar proceso: Este sistema deberá brindar la posibilidad de adicionar un proceso.

RF8. Modificar proceso: Mediante este requisito el sistema deberá brindar la posibilidad de modificar un proceso.

- RF9.** Eliminar proceso: A través de este requisito, el sistema deberá brindar la oportunidad de eliminar un proceso.
- RF10.** Buscar conexión: Este sistema deberá brindar la posibilidad de buscar una conexión.
- RF11.** Listar esquemas: Este sistema deberá ser capaz de listar todos los esquemas de la conexión seleccionada.
- RF12.** Buscar esquema: Este sistema deberá brindar la posibilidad de buscar un esquema.
- RF13.** Listar tablas: Este sistema deberá ser capaz de listar todas las tablas de los esquemas seleccionados.
- RF14.** Buscar tabla: Este sistema deberá brindar la posibilidad de buscar una tabla.
- RF15.** Listar eventos: Este sistema deberá ser capaz de listar los eventos del proceso seleccionado.
- RF16.** Adicionar evento: Este sistema deberá brindar la posibilidad de adicionar un evento.
- RF17.** Modificar evento: Mediante este requisito el sistema deberá brindar la posibilidad de modificar un evento.
- RF18.** Eliminar evento: A través de este requisito, el sistema deberá brindar la oportunidad de eliminar un evento.
- RF19.** Listar procesos y eventos: Este sistema deberá ser capaz de listar los procesos y eventos.
- RF20.** Adicionar condiciones a los eventos: Este sistema deberá brindar la posibilidad de adicionar condiciones al evento.
- RF21.** Listar condiciones de los eventos: Este sistema deberá ser capaz de listar las condiciones del evento.
- RF22.** Adicionar configuración de eventos: Este sistema deberá brindar la posibilidad configurar el evento.
- RF23.** Modificar configuración de los eventos: Mediante este requisito el sistema deberá brindar la posibilidad de modificar las condiciones del evento.
- RF24.** Adicionar instancia del proceso: Este sistema deberá brindar la posibilidad de adicionar instancia del proceso.
- RF25.** Adicionar atributo del negocio: Este sistema deberá brindar la posibilidad de adicionar atributos del negocio.
- RF26.** Modificar atributo del negocio: Mediante este requisito el sistema deberá brindar la posibilidad de modificar los atributos del negocio.

RF27. Eliminar atributo del negocio: A través de este requisito, el sistema deberá brindar la oportunidad de eliminar un atributo del negocio.

RF28. Listar atributo del negocio: Este sistema deberá ser capaz de listar los atributos del negocio.

RF29. Validar proceso: Este sistema deberá brindar la posibilidad de validar un proceso.

RF30. Activar proceso: Este sistema deberá brindar la posibilidad de activar un proceso.

RF31. Desactivar proceso: Este sistema deberá brindar la posibilidad de desactivar un proceso.

RF32. Generar trazas de proceso: Este sistema deberá brindar la posibilidad de construir dichas trazas con los datos de los historiales en conjunto con las trazas de datos.

RF33. Listar trazas de proceso: Este sistema deberá ser capaz de listar las trazas de proceso generadas por el módulo de administración.

Buscar trazas de proceso

RF34. Exportar trazas de proceso: Este sistema deberá brindar la posibilidad de exportar como fichero XES las trazas de proceso generadas.

Requisitos no funcionales.

Los requisitos no funcionales constituyen propiedades o cualidades que el producto debe tener, son requisitos que imponen restricciones en el diseño o en la implementación. Son características que hacen al producto atractivo, usable, rápido o confiable. A continuación se muestran los requisitos del módulo (Pressman, 2005).

Usabilidad.

RnF 1: Tendrá un ambiente sencillo y todas las opciones estarán al alcance de un clic, será fácil de manejar por cualquier experto del dominio en SAGI aunque no tenga mucha experiencia en el ámbito de la Minería de Procesos.

Soporte.

RnF 2: La aplicación contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

Portabilidad.

RnF 3: El sistema podrá ser ejecutado en sistema operativo Windows xp o superior o Linux.

Software.

En el cliente: Se debe disponer para ejecutar la aplicación:

RnF 4: Navegador Web preferentemente Firefox versión 38.

RnF 5: Sistema operativo Windows xp o superior o Linux.

En el servidor: Se debe disponer para instalar la aplicación:

RnF 6: Sistema operativo Windows xp o superior o Linux en cualquiera de sus distribuciones.

RnF 7: Un servidor Apache 2.2 o superior.

RnF 8: Un servidor de base de datos PostgreSQL 9.1.

Hardware.

En el cliente: Se debe disponer para ejecutar la aplicación:

RnF 9: Requerimientos mínimos: Procesador Pentium II a 133Mhz con 256 Mb de memoria RAM.

RnF 10: Tarjeta de red.

En el servidor: Se debe disponer para poder instalar la aplicación:

RnF 11: Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.

RnF 12: Al menos 40Gb de espacio libre en disco duro.

RnF 13: Tarjeta de red.

2.4- Patrones de Casos de Uso

CRUD total: Este patrón se evidencia en el CU Gestionar Conexión pues en él se agrupan los siguientes requisitos funcionales: Adicionar Conexión, Modificar Conexión y Eliminar Conexión y Listar Conexión.

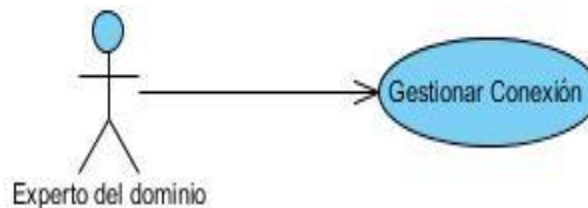


Fig. 4: Evidencia del patrón CRUD total.

2.5- Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema es la representación del comportamiento del software desde el punto de vista del usuario. Permite visualizar las funcionalidades en casos de uso, el entorno del sistema en actores, y las relaciones entre ambos. Aunque la parte más visible de dicho modelo son los diagramas de casos de uso, suele ir acompañado de una especificación textual de cada uno de los casos de uso.

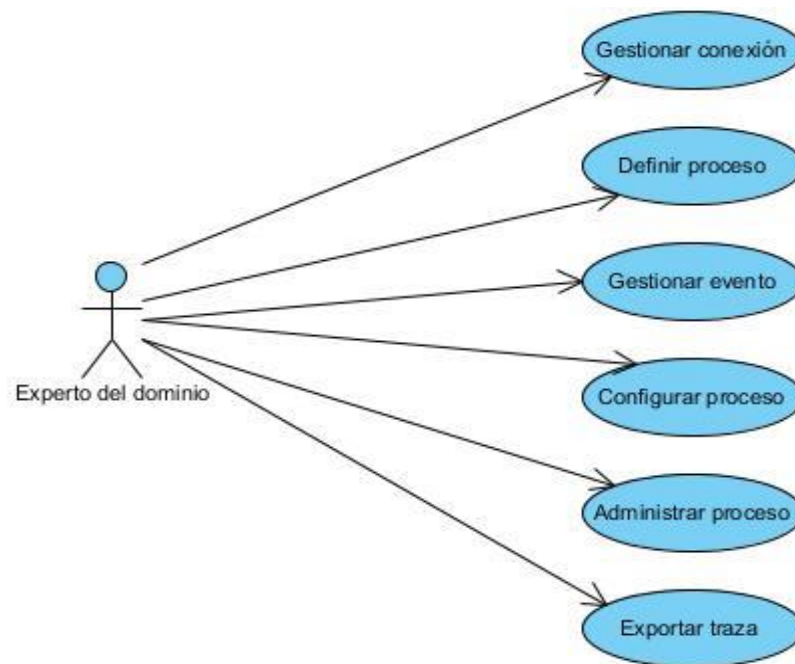


Fig. 5: Diagrama de Casos de Uso del Sistema.

En el diagrama de casos de uso del sistema de la Fig. 5, el actor Experto del dominio es el encargado de iniciar los casos de uso: Gestionar conexión, Definir proceso, Gestionar evento, Configurar proceso, Administrar proceso y Exportar traza. De esta forma, se agruparon los 34 requisitos funcionales en seis casos de uso, siendo de alta prioridad para la arquitectura del módulo Definir proceso y Configurar proceso.

Descripción de los Casos de Uso del Sistema.

La descripción de los casos de uso del sistema narra detalladamente los eventos que los actores realizan para completar un proceso o funcionalidad a través de la aplicación. A continuación se describe el más significativo los Casos de Uso presentes en el Diagrama de Casos de Uso del Sistema.

Tabla 1: Descripción textual del CU Definir Proceso.

Objetivo	Definir un proceso.	
Actores	Experto de dominio	
Resumen	<p>El CU se inicia cuando el Experto de dominio va a realizar algunas de las siguientes operaciones:</p> <ul style="list-style-type: none"> • Adicionar un proceso: cuando el Experto de dominio introduce los datos necesarios y crea un proceso, finalizando así el CU. • Modificar un proceso: cuando el Experto de dominio necesita modificar datos del proceso, finalizando así el CU. • Eliminar un proceso: cuando el Experto de dominio elimina un proceso, finalizando así el CU. 	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Que exista al menos una conexión disponible.	
Postcondiciones	<p>En dependencia de la acción del Experto de dominio:</p> <ol style="list-style-type: none"> 1. Se adiciona un proceso en el sistema. 2. Se modifica un proceso en el sistema. 3. Se elimina un proceso en el sistema 4. No se registran los datos. 5. El proceso se desactiva en caso de estar activado si se modifican valores además de: Nombre y Descripción. Igualmente el proceso se invalida en caso de estar validado si se modificaron valores además de: Nombre y Descripción. Se crea una nueva versión del proceso si se modificaron valores además de: Nombre y Descripción. 	
Flujo de eventos		
Flujo básico < Adicionar >		
	Actor	Sistema
		1. Muestra un listado de las conexiones disponibles.
2.	Selecciona la conexión y presiona el botón Siguiente.	3. Muestra un listado de los esquemas asociados a la conexión seleccionada.
4.	Selecciona uno o varios esquemas y presiona el	5. Muestra un listado de las tablas

	botón Siguiente .	asociadas a los esquemas seleccionados.
6.	Selecciona una o varias tablas y presiona el botón Siguiente .	7. Muestra una interfaz dividida en dos campos, en la parte izquierda se insertan los datos del nuevo proceso y en la derecha se muestra un listado de tablas, una de ellas contendrá el identificador de instancia de proceso.
8.	Inserta los datos en la parte izquierda de la interfaz en los campos: Nombre, Fuente de datos y Descripción.	9. Valida los datos.
10.	Selecciona la tabla principal del proceso la cual contendrá la instancia de proceso en la parte derecha de la interfaz.	
11.	Presiona el botón Aceptar .	12. Muestra un mensaje de información: “Elemento adicionado satisfactoriamente” .
Sección 1: “Adicionar”		
Flujos alternos		
Nº Evento < 1.a Adicionar proceso dejando campos vacíos. >		
	Actor	Sistema
2.	Deja un campo vacío.	1. Muestra un mensaje de error: “Existen campos inválidos” .
3.	Volver al paso 8 del flujo básico.	
Flujos alternos		
Nº Evento < 1.b Adicionar un proceso con un nombre ya existente. >		
	Actor	Sistema
1.	Escribe el nombre de un proceso ya definido.	2. Muestra un mensaje de error: “Error, existe un proceso con el mismo nombre” .
3.	Volver al paso 8 del flujo básico.	
Flujos alternos		

Nº Evento < 1.d El usuario cancela la acción >		
	Actor	Sistema
1.	Presiona el botón Cancelar .	2. No guarda los datos.
Sección 2: "Modificar"		
Nº Evento < Modificar proceso >		
	Actor	Sistema
		1. Muestra el listado con los procesos previamente definidos.
2.	Selecciona un proceso y presiona el botón Modificar .	3. Muestra un listado de las conexiones disponibles y la conexión del proceso seleccionada.
4.	Selecciona otra conexión de las existentes si se desea.	
5.	Presiona el botón Siguiente .	6. Muestra un listado de los esquemas asociados a la conexión seleccionada.
7.	Selecciona los esquemas deseados.	
8.	Presiona el botón Siguiente .	9. Muestra un listado de las tablas asociadas a los esquemas seleccionados.
10.	Selecciona una o varias tablas.	
11.	Presiona el botón Siguiente .	12. Muestra una interfaz dividida en dos campos, en la parte izquierda se modifican los datos del proceso seleccionado y en la derecha se muestra un listado de tablas, una de ellas contendrá el identificador de instancia de proceso.
13.	Modifica los datos en la parte izquierda de la interfaz en los campos: Nombre, Fuente de datos y Descripción.	14. Valida los datos.

15.	Selecciona la tabla principal del proceso la cual contendrá la instancia de proceso en la parte derecha de la interfaz.	
16.	Presiona el botón Aceptar .	17. Muestra un mensaje de información: “Elemento Modificado satisfactoriamente” .
Flujos alternos		
Nº Evento < 2.a Modificar un proceso dejando campos vacíos >		
	Actor	Sistema
1.	Deja un campo vacío.	2. Muestra un mensaje de error: “Existen campos inválidos” .
Flujos alternos		
Nº Evento < 2.b Modificar un proceso con un nombre ya existente >		
	Actor	Sistema
1.	Escribe el nombre de un proceso existente.	2. Muestra un mensaje de error: “Error, existe un proceso con el mismo nombre” .
Flujos alternos		
Nº Evento < 2.c El usuario cancela la acción. >		
	Actor	Sistema
1.	Presiona el botón Cancelar .	2. No guarda los datos.
Sección 3: “Eliminar”		
Nº Evento < Eliminar proceso >		
	Actor	Sistema
		1. Muestra el listado con los procesos previamente definidos.
2.	Selecciona el proceso a eliminar y presiona el botón Eliminar .	3. Muestra un mensaje de confirmación: “Está seguro que desea eliminar el/los elemento(s) seleccionado(s)” .
4.	Presiona el botón Aceptar .	5. Muestra un mensaje de información: “Elemento(s) eliminado(s)”

		satisfactoriamente”.
		6. Concluye el Caso Uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

2.6- Diagramas de clases

Los diagramas de clases describen la estructura de un sistema. Estos diagramas representan las clases de la solución, así como las principales relaciones que existen entre las mismas. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas aplicando ingeniería directa e inversa. A continuación se muestra el diagrama de clase para el caso de uso Definir proceso. El resto de los Diagramas de Clases del Diseño se encuentran en el anexo del trabajo.

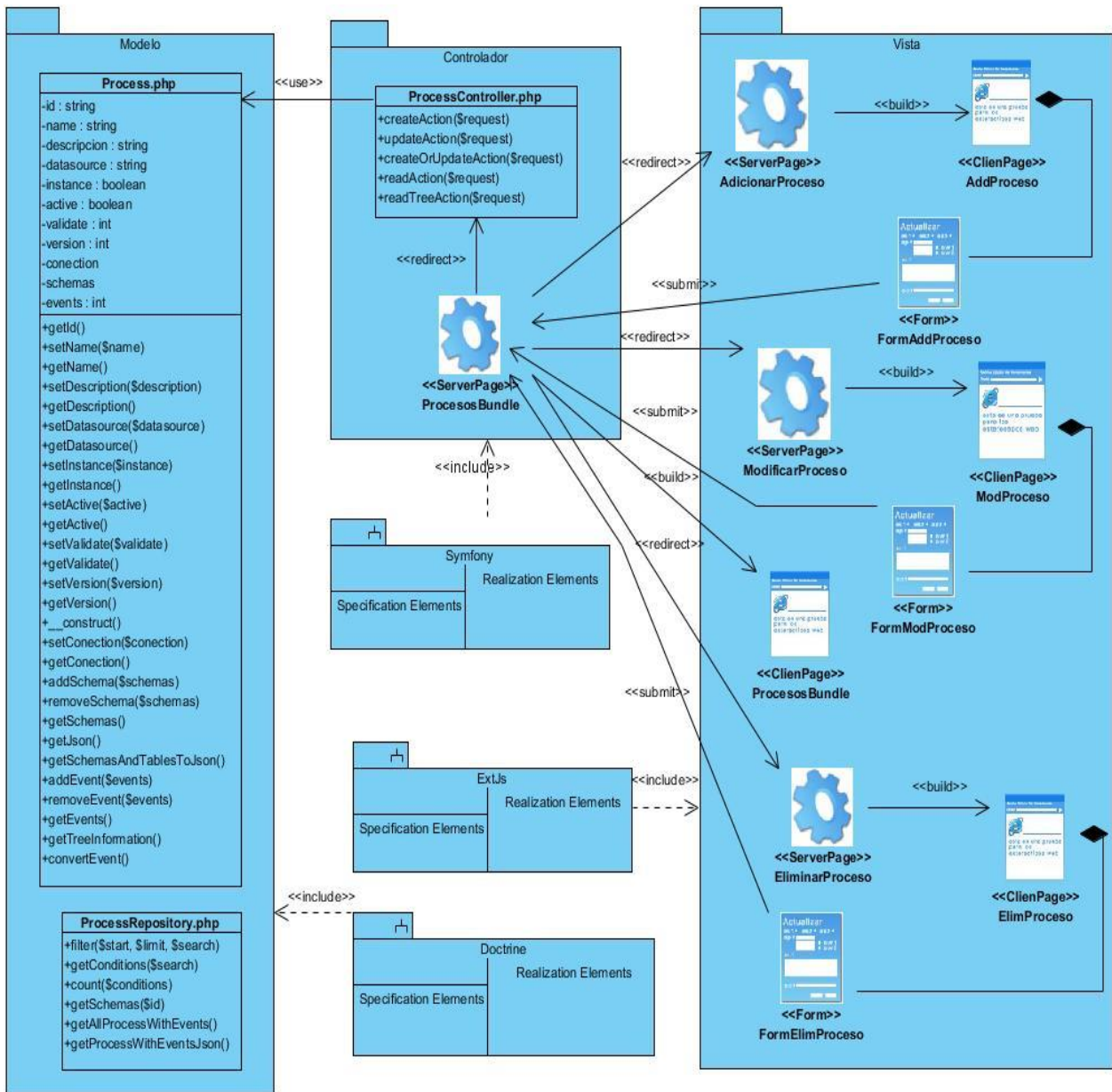


Fig. 6: Diagrama de clases del diseño para el CU Definir Proceso.

En este diagrama, los elementos del modelo corresponden a las clases generadas por el ORM Doctrine, modelado por el subsistema de igual nombre. Las acciones, elementos que construyen las respuestas del servidor, están implementadas en el lenguaje PHP, con la dependencia del subsistema de Symfony comprobadas por el controlador frontal ProcesosBundle. Las solicitudes se realizan por medio de la

tecnología AJAX y las respuestas son devueltas específicamente en formato JSON. Los elementos del lado del cliente corresponden a componentes específicos del negocio, en lenguaje JavaScript utilizando los componentes del subsistema ExtJS.

2.7- Diagramas de secuencia

Los diagramas de secuencia muestran cómo las instancias específicas de las clases trabajan juntas para alcanzar un objetivo común. Al construirlos se toman decisiones claves acerca del funcionamiento futuro del sistema y permiten representar su comportamiento dinámico. A continuación se muestra el diagrama de secuencia para el caso de uso Adicionar proceso. El resto de los Diagramas de Secuencia se encuentran en el anexo del trabajo.

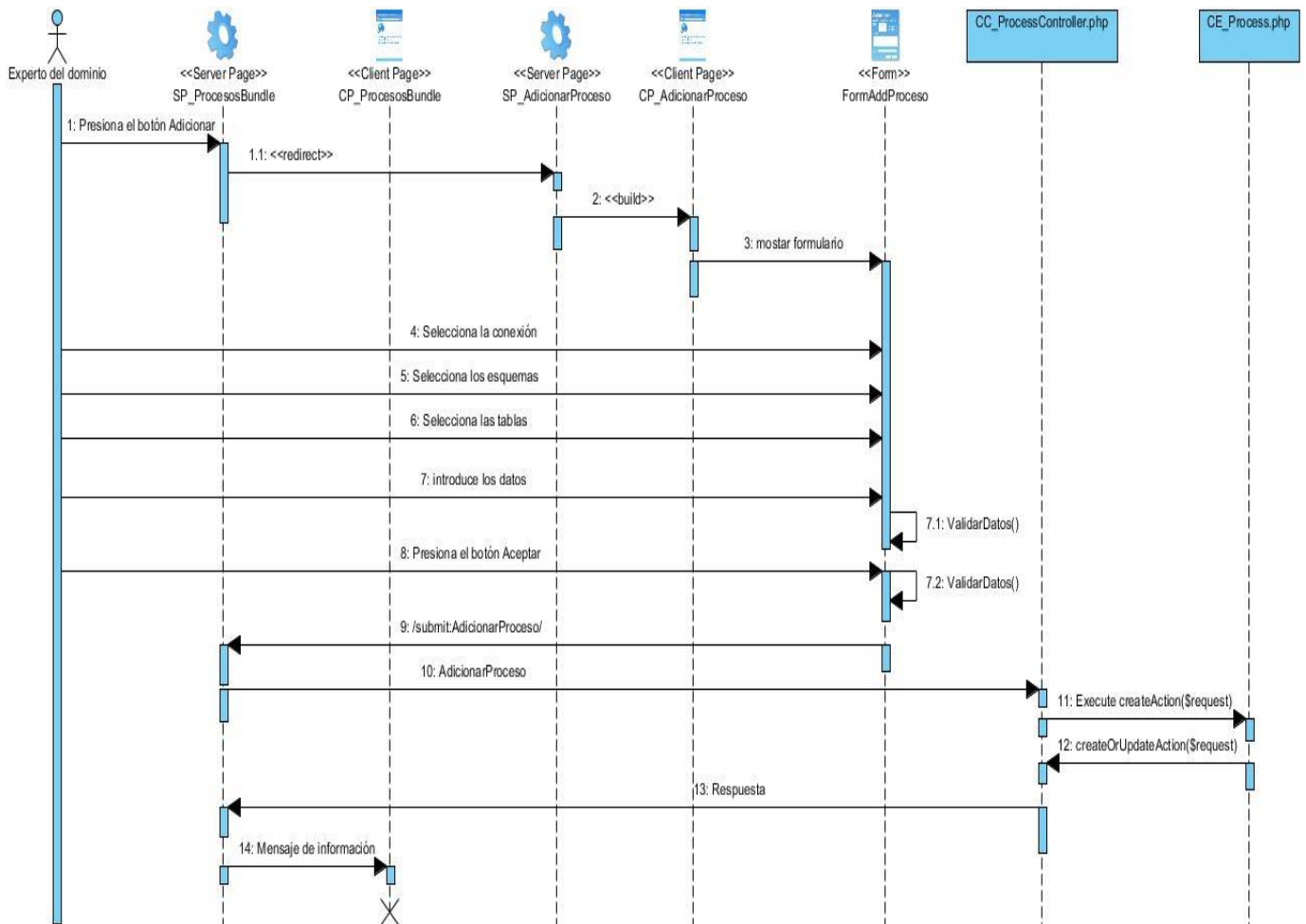


Fig. 7: Diagrama de secuencia correspondiente al escenario Adicionar Proceso.

En este diagrama de secuencia el actor Experto del dominio selecciona la opción Adicionar para registrar un proceso en el sistema. El controlador frontal ProcesosBundle se encarga, a través de la operación “redirect”, de enviar la solicitud a la página servidora AdicionarProceso que a su vez mediante la operación “build” abre la página cliente AdicionarProceso la cual contiene un formulario donde el usuario introduce los datos para crear un proceso y envía estos datos mediante la operación “submit” al controlador frontal. Este a su vez le envía a ProcessController.php la tarea de ejecutar createAction(\$request), posteriormente le envía a ProcesosBundle la respuesta de la acción ejecutada mostrando este el resultado mediante la página cliente del mismo nombre.

2.8- Patrones arquitectónicos

Modelo Vista Controlador (MVC).

El patrón MVC es una arquitectura de diseño software para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio. Es una especialización de un modelo de capas, con la diferencia que se usa para entornos web como patrón por excelencia (Eguiluz, 2011). A continuación se describe cómo se utilizan las capas en el sistema.

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.

Capa de presentación: es la 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 y páginas clientes.

Capa de control: se implementa con una o varias clases Controladoras que son las clases encargadas de recibir las diferentes peticiones del experto de dominio y hacer las llamadas a las funciones necesarias para dar cumplimientos a estas peticiones. Como se muestra en la Fig. 8.

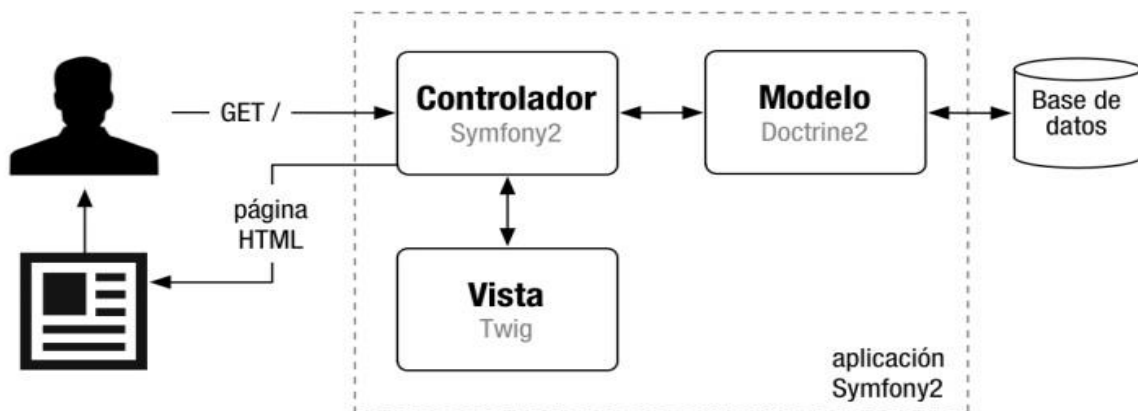


Fig. 8: Representación del Patrón Modelo Vista Controlador.

2.9- Patrones de diseño

Los patrones generales de software para asignación de responsabilidades GRASP⁸. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Visconti, y otros, 2008).

Bajo acoplamiento: Se centra tener las clases lo menos ligadas entre sí, posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. El uso de este patrón se aprecia en la clase `ConectionController.php` la cual contiene toda la información referente a las conexiones de forma tal que si ocurre algún cambio, no se tenga que realizar ninguna modificación en las demás clases.

```
ConectionController.php
+createAction($request)
+updateAction($request)
+testAction($request)
+createConnection($params)
+createOrUpdateEntity($request, $action)
+getParams($request)
```

Fig. 9: Evidencia del patrón Bajo Acoplamiento.

Alta cohesión: Propone que la información que almacena una clase debe de ser coherente y debe estar relacionada con la clase. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos. Ejemplo de esto es el controlador `ProcessConfigurationController.php` el cual al recibir un mínimo cambio, requiere una completa reestructuración de los métodos para su correcto funcionamiento.

```
AdminProcessController.php
+activateProcessAction($request)
+generateTraceProcessAction($request)
+disactivateProcessAction($request)
+validateProcessAction($request)
+buildAttrBusiness($pl, $row)
+removeEqualResult($array)
+functionh($table, $fields)
+functionhD($table, $fields)
```

Fig. 10: Evidencia del patrón Alta Cohesión.

⁸ Del inglés: General Responsibility Assignment Software Patterns.

Experto: Este patrón indica que la responsabilidad de hacer cualquier función debe recaer sobre la clase que tenga toda la información necesaria para hacerlo. De este modo se mantiene la información encapsulada y se obtiene un diseño con mayor cohesión. Se evidencia la utilización de este patrón en la clase `ProcessController.php` por ser la que posee los datos involucrados en la definición de un proceso.

```
ProcessController.php
+createAction($request)
+updateAction($request)
+createOrUpdateAction($request)
+readAction($request)
+readTreeAction($request)
```

Fig. 11: Evidencia del patrón Experto.

Creador: Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Se evidencia el uso de este patrón en la clase controladora `EventController.php` que se encarga de llevar el control de todos los eventos relacionados con el negocio e implementa las funcionalidades que dan respuesta a las peticiones del usuario.

```
EventController.php
+createAction($request)
+updateAction($request)
+createOrUpdateEntity($request, $action)
+getParams($request)
+readAction($request)
+validateEntity($entity)
```

Fig. 12: Evidencia del patrón Creador.

Controlador: Este patrón funciona como intermediario entre determinada interfaz y el algoritmo que la implementa de tal forma que es el controlador quien recibe los datos del usuario y quien los envía a las distintas clases según el método llamado. Se observa el uso de este patrón en la clase controladora `ProcessController.php` que tiene la responsabilidad de interactuar con las clases de acceso a datos, obtener la información necesaria y dar respuesta a las peticiones del cliente controlando así el flujo central de las acciones en el sistema.

```
ProcessController.php
+createAction($request)
+updateAction($request)
+createOrUpdateAction($request)
+readAction($request)
+readTreeAction($request)
```

Fig. 13: Evidencia del patrón Controlador.

2.10- Modelo de datos

Otro de los artefactos generados en la fase de análisis y diseño es el modelo de datos. El mismo tiene como propósito garantizar que los datos persistentes sean almacenados coherente, eficazmente y definir el comportamiento que debe ser implementado. El diseño conceptual de una base de datos proporciona mucha ayuda a la hora de visualizar las diferentes entidades que intervienen, así como también las relaciones entre ellas. A continuación se muestra el modelo de datos de la solución en el cual se ilustran las tablas de base de datos utilizadas y las relaciones entre las mismas.

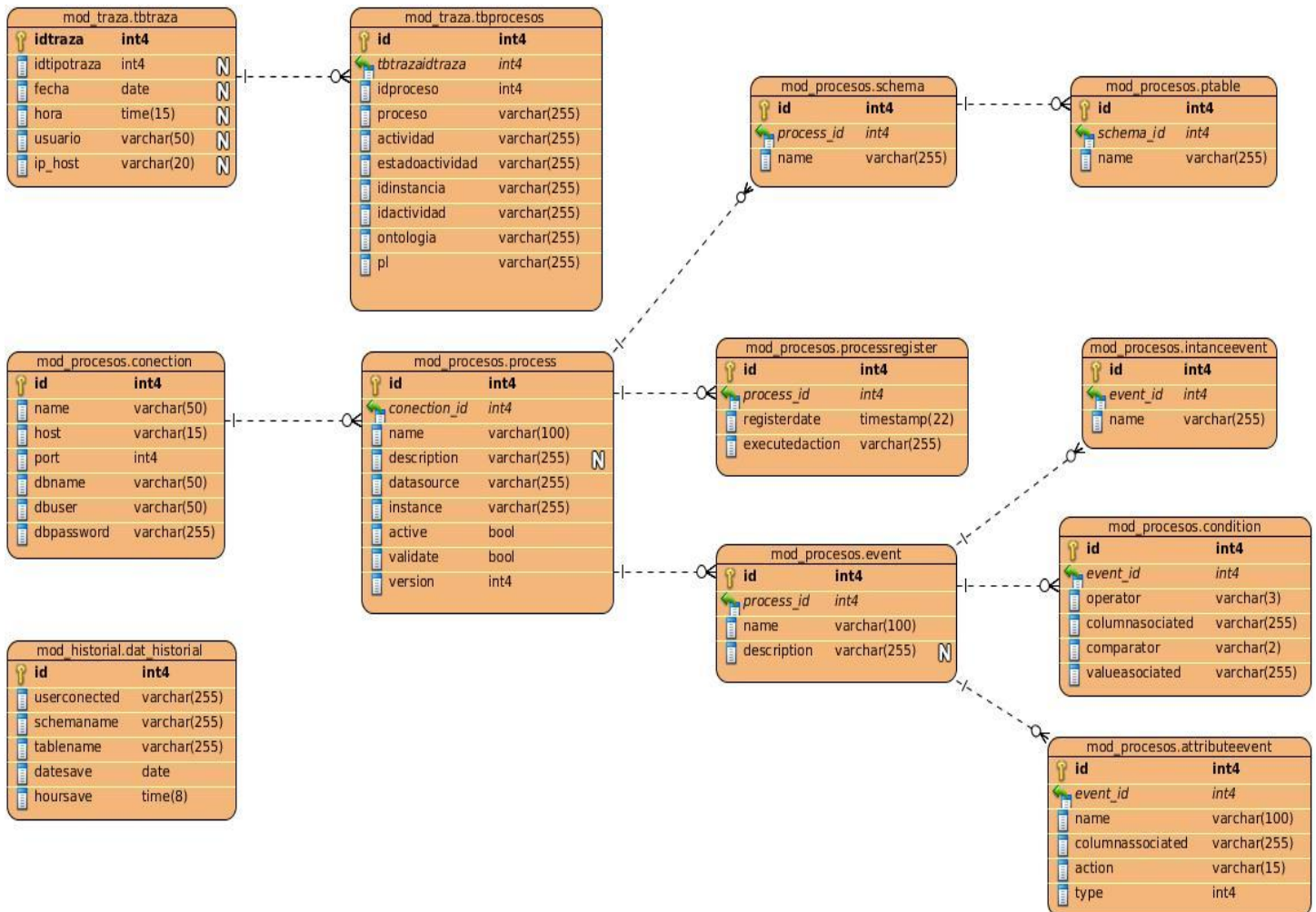


Fig. 14: Modelo de datos.

conection: Contiene toda la información relacionada con las conexiones a usar para la gestión de los procesos.

process: Contiene toda la información correspondiente a la definición del proceso, así como su estado de activación, validez y versión.

tbprocess: Almacena la información correspondiente a la generación de trazas de procesos.

event: Contiene toda la información relacionada a los eventos de los procesos definidos.

attributeevent: Contiene toda la información relacionada con la configuración de los atributos del evento.

condition: Contiene toda la información correspondiente a las condiciones del evento.

instanceevent: Almacena la instancia del evento.

processregistry: Esta tabla almacena la información referente a la activación y desactivación de los procesos, así como los intervalos de tiempo de los mismos.

schema: Contiene los esquemas que maneja el proceso.

ptable: Contiene las tablas de los esquema que utiliza el proceso.

dat_historial: Permite crear tablas espejo de tablas específicas en la base de datos, además almacena información sobre la fecha y hora que se creó o modificó un historial.

2.11- Diagrama de despliegue

Para evaluar de forma visual cómo se encuentran los componentes de la aplicación relacionados físicamente se elabora un diagrama de despliegue, este representa las relaciones físicas entre componentes de software y hardware en el sistema. En este caso la aplicación se encuentra hospedada en un servidor Web Apache y se comunica con el servidor de base de datos PostgreSQL. Como se muestra en la Fig. 155.

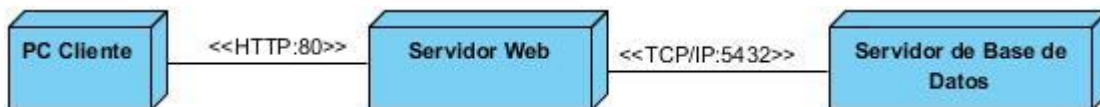


Fig. 15: Diagrama de Despliegue.

Conclusiones del capítulo:

La definición de los datos a registrar aseguró la completitud de la información necesaria para la creación de trazas de proceso. Se definieron los requisitos funcionales y no funcionales, los cuales debe cumplir el sistema. Se modelaron los artefactos correspondientes a la fase de Análisis y Diseño, los cuales constituyen la base de la implementación del producto. El uso de los patrones de diseño permitió ganar en reutilización y brindó robustez al diseño de la solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Introducción

En el presente capítulo se define el modelo de implementación para la creación de los módulos especificados en el capítulo anterior. Se describe el estándar de codificación utilizado para garantizar un correcto entendimiento del código. De esta forma se generan los artefactos pertenecientes a las fases de implementación y prueba de un software. De esta última fase se especifican los casos de pruebas realizadas al módulo para validar su correcto funcionamiento.

3.1- Diagrama de componentes

El diagrama de componentes representa la estructura física del sistema, su agrupación por paquetes y muestra las dependencias entre estos. Al iniciar la implementación todos los diagramas generados en fases anteriores se materializan en un sistema integrando las partes necesarias para la obtención de un producto final. Estas relaciones de dependencia se modelan a través del diagrama de componentes, como se muestra en la Fig. 166.

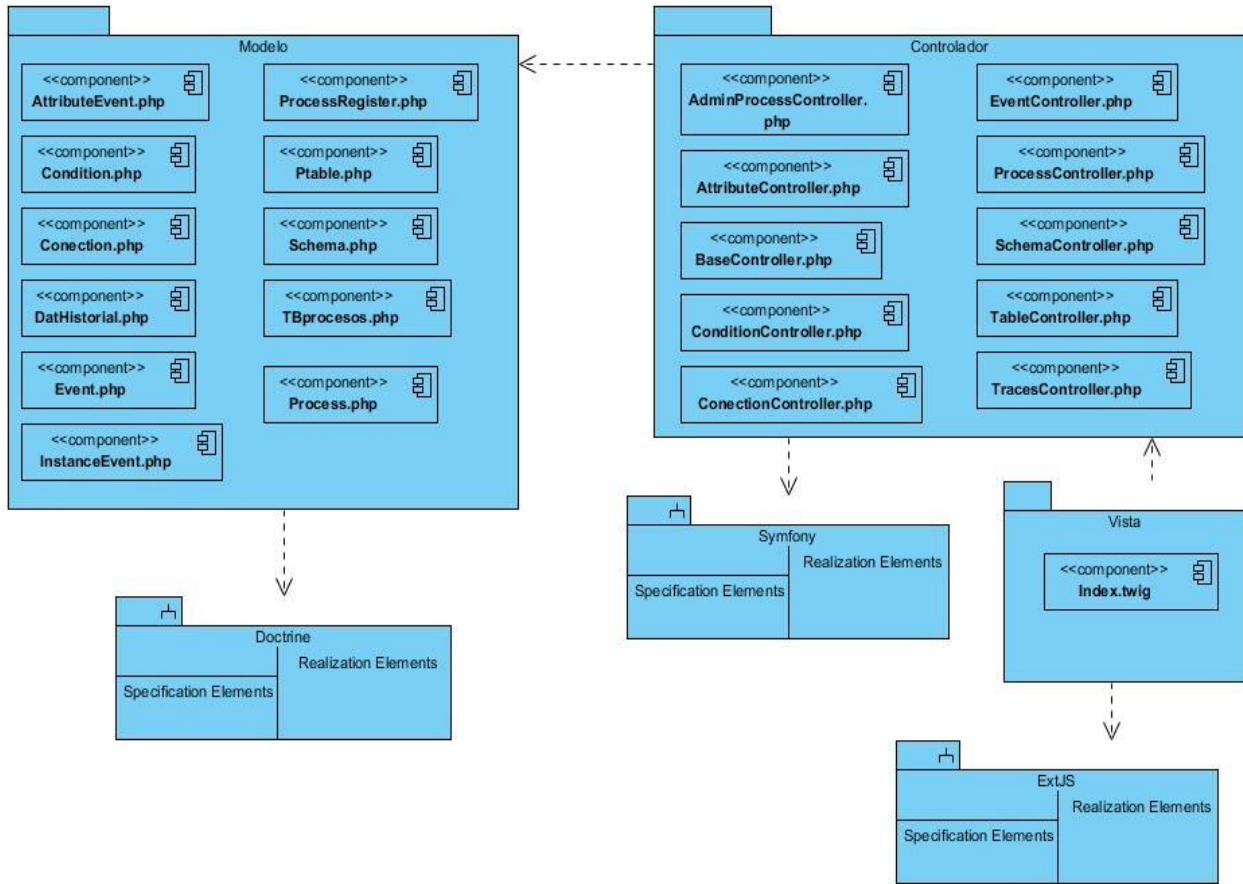


Fig. 16: Diagrama de Componentes.

En el diagrama anterior se muestran los componentes divididos en paquetes utilizando el patrón Modelo Vista Controlador. En el paquete Vista se localizan los componentes que pertenecen a las interfaces del módulo, empleando el subsistema ExtJS. En el paquete Controlador se encuentra el componente intermediario entre la vista y el modelo, este componente representa el núcleo del módulo y se encarga del control de las funcionalidades. Los componentes del Modelo corresponden a las clases generadas por el subsistema Doctrine las cuales implementan la lógica del negocio y se encargan del acceso a datos.

3.2- 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. El uso de estándares de codificación permite lograr un código más legible y reutilizable, de tal

forma que se pueda aumentar la sostenibilidad a lo largo del tiempo (PHP Framework Interop Group, 2015). A continuación se muestra el estándar utilizado:

- **Notación PascalCasing:** En este caso los identificadores y nombres de las 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.

Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing.

Ejemplo: ProcessController.

Nomenclatura según el tipo de clases

1. Clase controladora.

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

Ejemplo: EventController.

2. Clases de los modelos

Las clases que se encuentran dentro de Entity empiezan con mayúscula.

Ejemplo: Process.

Nomenclatura de las funciones

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing y en caso de ser una acción de la clase controladora se debe especificar el nombre de dicha acción en minúscula y seguido del sufijo "Action".

Ejemplo: createAction.

Nomenclatura de las variables

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing.

Ejemplo: \$process.

Longitud de la línea

Se evitan las líneas de más de 80 caracteres.

Comentarios

Los comentarios se utilizan para dar descripciones de código. Contienen sólo información que es relevante para la lectura y entendimiento del programa (ver Fig. 177).

```
/* id de conexión a la que pertenecen los datos */
$idConexión = $request->get('idConexión');
/* esquema de donde salen los datos */
$schemas = json_decode($request->get('schemas'));
/* tablas seleccionadas */
$tablas = json_decode($request->get('tables'));
/* datos generales del proceso */
$params = (array) json_decode($request->get('process'));
```

Fig. 17: Longitud de la línea y Comentarios.

Colocación de llaves

Las llaves se colocan al principio de los bloques (un bloque es cualquier código encerrado por llaves "{" y "}"). De esta forma se evita confundir a los programadores y no limitar la portabilidad del código dentro de su ámbito de visibilidad (ver Fig. 188).

```
public function createAction(Request $request) {
    try {
        $this->createOrUpdateEntity($request);
        return new Response(json_encode(array('success' => true)));
    } catch (\Exception $exc) {
        return new Response(json_encode(array('success' => false,
                                             'message' => $exc->getMessage())));
    }
}
```

Fig. 18: Colocación de llaves.

3.3- Pruebas de software

Las pruebas de software son un elemento esencial para la garantía de calidad del software y representan una revisión final de las especificaciones del diseño y de la codificación. Estas consisten en la dinámica de

la verificación del comportamiento de un programa en un conjunto finito de casos de prueba. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2005).

Cuando se aplican pruebas a un software, se tienen en cuenta una serie de objetivos en diferentes escenarios y niveles. Para comprobar el correcto funcionamiento del módulo, de los siguientes niveles de pruebas: unidad, integración, independiente, sistema, desarrollador y aceptación, se aplicaron los dos últimos.

Niveles de Prueba

Desarrollador: Este tipo de pruebas son ejecutadas por el equipo de desarrollo, básicamente consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes están codificados bajo condiciones de robustez, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada (Pressman, 2005).

Aceptación: Estas pruebas son básicamente pruebas funcionales sobre el sistema completo, buscan comprobar que se satisfacen los requisitos establecidos. Fueron realizadas por un miembro del equipo de SAGI mediante pruebas funcionales utilizando el método de caja negra. Al final de la prueba se emitió una carta de aceptación la cual hace constar que, el Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI, fue probado y cumple con cada uno de los requisitos funcionales definidos.

Tipo de Prueba

Funcionalidad: Las pruebas funcionales están basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Se elaboran mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático (Pressman, 2005).

Método de Prueba

Caja Negra: Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Mediante la utilización de este método

se demuestra que las funciones del software son operativas, la entrada se acepta de forma adecuada, se produce una salida correcta y la integridad de la información externa se mantiene (Pressman, 2005).

Técnica de prueba

Particiones de equivalencia: Divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. El diseño de casos de pruebas para esta técnica se basa en una evaluación de las clases de equivalencia que representan un conjunto de valores válidos y no válidos para una condición de entrada (Pressman, 2005). A continuación se presentan los casos de prueba para el CU Definir proceso.

Descripción general: Este caso de uso tiene la finalidad de gestionar todo lo referente a los procesos, ya sea adicionar, modificar, eliminar y listar los procesos.

Condiciones de ejecución: El sistema debe tener registrado al menos una conexión.

Tabla 2: Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Conexión	Campo de selección	No	Campo de selección que permite especificar la conexión del proceso.
2	Esquema	Campo de selección	No	Campo de selección que permite especificar el esquema del proceso.
3	Tablas	Campo de selección	No	Campo de selección que permite especificar las tablas del proceso.
4	Nombre	Campo texto	No	Campo alfanumérico con longitud máxima de 100 caracteres, indica el nombre del proceso
5	Fuente de datos	Campo texto	No	Campo alfanumérico con longitud máxima de 255 caracteres, indica la fuente de datos que origina el proceso.
6	Descripción	Campo texto	Si	Campo texto que acepta cualquier tipo de caracter, indica la descripción del proceso.

Tabla 3: Sección 1: Caso de prueba Adicionar Proceso.

Escenario	Descripción	Conexión	Esquema	Tablas	Nombre	Fuente de datos	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Adicionar proceso	En este escenario se realiza la adición de un nuevo proceso al sistema correctamente.	Conexión	mod_sige	nestado nmodeloe stado ntipodein dicador ntipodem odelo	Prueba	SAGI	(-)	Se adiciona correctamente el nuevo proceso. Muestra el mensaje de información "Elemento adicionado satisfactoriamente".	1. Definir proceso. 2. Seleccionar la opción "Adicionar". 3. Seleccionar la conexión. 4. Seleccionar los esquemas. 5. Seleccionar las tablas. 6. Se llenan los campos correctamente. 6. Se presiona el botón Aceptar.
EC 1.2 Adicionar proceso con campos en blanco.	En este escenario se realiza la adición de un proceso al sistema con datos en blanco.	Conexión	mod_sige	nestado nmodeloe stado ntipodein dicador ntipodem odelo	()	SAGI	(-)	El sistema valida que los campos obligatorios estén todos llenos y correctamente. Muestra el mensaje de error "Existen campos inválidos".	1. Definir proceso. 2. Seleccionar la opción "Adicionar". 3. Seleccionar la conexión. 4. Seleccionar los esquemas. 5. Seleccionar las tablas. 6. Se dejan campos obligatorios en blanco. 7. Se presiona el botón Aceptar.

EC 1.3 Cancelar operación	En este escenario se cancela la operación de adicionar un proceso.	(-)	(-)	(-)	(-)	(-)	(-)	Se cancela la operación de adicionar un proceso. No se guardan los datos en el sistema.	1. Definir proceso. 2. Seleccionar la opción "Adicionar". 3. Seleccionar la conexión. 4. Seleccionar los esquemas. 5. Seleccionar las tablas. 6. Se completan los datos. 7. Se presiona el botón Cancelar.
---------------------------------	---	-----	-----	-----	-----	-----	-----	--	---

Tabla 4: Sección 2: Caso de prueba Modificar Proceso.

Escenario	Descripción	Conexión	Esquema	Tablas	Nombre	Fuente de datos	Descripción	Respuesta del sistema	Flujo central
EC 2.1 Modificar proceso	En este escenario se realiza la modificación de un proceso existente en el sistema correctamente.	Conexión	mod_sige	nestado nmodelo estado ntipodein dicador ntipodem odelo	Prueba1	SAGI	(-)	Se modifica correctamente el proceso seleccionado. Muestra el mensaje de información "Elemento modificado satisfactoriamente".	1. Definir proceso. 2. Seleccionar el proceso a modificar. 3. Seleccionar la opción "Modificar". 4. Se llenan los campos correctamente. 5. Se presiona el botón Aceptar.
EC 2.2 Modificar proceso con	En este escenario se realiza la modificación	Conexión	mod_sige	nestado nmodelo estado	Prueba	()	(-)	El sistema valida que los campos obligatorios estén todos	1. Definir proceso. 2. Seleccionar el proceso a modificar.

campos en blanco	ción de un proceso existente en el sistema con datos en blanco.			ntipodeindicador				llos y correctamente. Muestra el mensaje de error "Existen campos inválidos".	3. Seleccionar la opción "Modificar". 4. Se dejan campos obligatorios en blanco. 5. Se presiona el botón Aceptar.
EC 2.3 Cancelar operación	En este escenario se cancela la operación de modificar proceso.	(-)	(-)	(-)	(-)	(-)	(-)	Se cancela la operación de modificar un proceso. No se guardan los datos en el sistema.	1. Definir proceso. 2. Seleccionar el proceso a modificar. 3. Seleccionar la opción "Modificar". 4. Seleccionar la conexión. 5. Seleccionar los esquemas. 6. Seleccionar las tablas. 7. Se completan los datos. 8. Se presiona el botón Cancelar.

Tabla 5: Sección 3: Caso de prueba Eliminar Proceso.

Escenario	Descripción	Conexión	Esquema	Tablas	Nombre	Fuente de datos	Descripción	Respuesta del sistema	Flujo central
EC 3.1 Eliminar proceso	En este escenario se realiza la eliminación de un proceso.	(-)	(-)	(-)	(-)	(-)	(-)	Se muestra un mensaje de confirmación "Está seguro que desea eliminar el/los elemento(s) seleccionado(s)". Se elimina el proceso.	<ol style="list-style-type: none"> 1. Definir proceso. 2. Seleccionar el proceso a eliminar. 3. Seleccionar la opción "Eliminar". 4. Se presiona el botón Aceptar del mensaje de confirmación.
EC 3.2 Cancelar operación	En este escenario se cancela la eliminación de un proceso.	(-)	(-)	(-)	(-)	(-)	(-)	Se cancela la operación de eliminar proceso.	<ol style="list-style-type: none"> 1. Definir proceso. 2. Seleccionar el proceso a eliminar. 3. Seleccionar la opción "Eliminar". 4. Se presiona el botón Cancelar del mensaje de confirmación.

Tabla 6: Sección 4: Caso de prueba Listar Proceso.

Escenario	Descripción	Conexión	Esquema	Tablas	Nombre	Fuente de datos	Descripción	Respuesta del sistema	Flujo central
EC 4.1 Listar procesos	En este escenario se listan los procesos existentes en el sistema.	(-)	(-)	(-)	(-)	(-)	(-)	Se listan todos los procesos existentes en el sistema.	1. Definir proceso.

3.4- Resultados de las pruebas

Se efectuaron tres iteraciones de prueba, donde se aplicaron los casos de prueba diseñados detectándose 16 no conformidades, las cuales fueron resueltas en la medida en que se detectaron. A continuación se muestra una gráfica que ilustra este resultado.



Fig. 19: Resultados de las pruebas de caja negra.

3.5- Interfaces de la aplicación

La definición de un proceso consiste en la selección de una o más tablas asociadas a la ejecución del proceso, la definición del nombre (único para cada proceso), fuente de datos, descripción del proceso y la selección de la tabla principal del proceso la cual debe poseer el identificador de instancia de proceso. Como se muestra en la Fig. 20.

The screenshot shows a software window titled "Adicionar Proceso" with a close button in the top right corner. Below the title bar, it indicates "Paso 4 > Introducir datos del proceso." The window is divided into two main sections: "Datos del proceso" on the left and "Instancia de proceso" on the right. In the "Datos del proceso" section, there are three input fields: "Nombre:" with the value "disennarplantilla", "Descripción:" with the value "disennar plantilla", and "Fuente de datos:" with the value "sagi". The "Instancia de proceso" section features a search bar with the placeholder text "Buscar...". Below the search bar is a table with the following content:

	Nombre
<input checked="" type="checkbox"/>	mod_sige.tbdescriptorodelmodelo
<input type="checkbox"/>	mod_sige.tbindicadorpagina
<input type="checkbox"/>	mod_sige.tbpaginaaspecto

At the bottom of the window, there are three buttons: "Atrás" (with a left-pointing arrow), "Aceptar" (with a green checkmark icon), and "Cancelar" (with a red X icon).

Fig. 20: Interfaz de usuario Adicionar Proceso.

Una vez definido el proceso se procede a su configuración. La configuración de un proceso consiste en la asociación de los atributos de cada uno de los eventos a columnas de tablas de bases de datos. Estas asociaciones pueden estar bajo ciertas condiciones que el proceso debe cumplir. En caso de ser necesario pueden ser adicionados atributos del negocio, estos no forman parte del proceso como tal pero aportan información relevante al registro de eventos, además se especifica la instancia del proceso. Como se muestra en la Fig. 21.

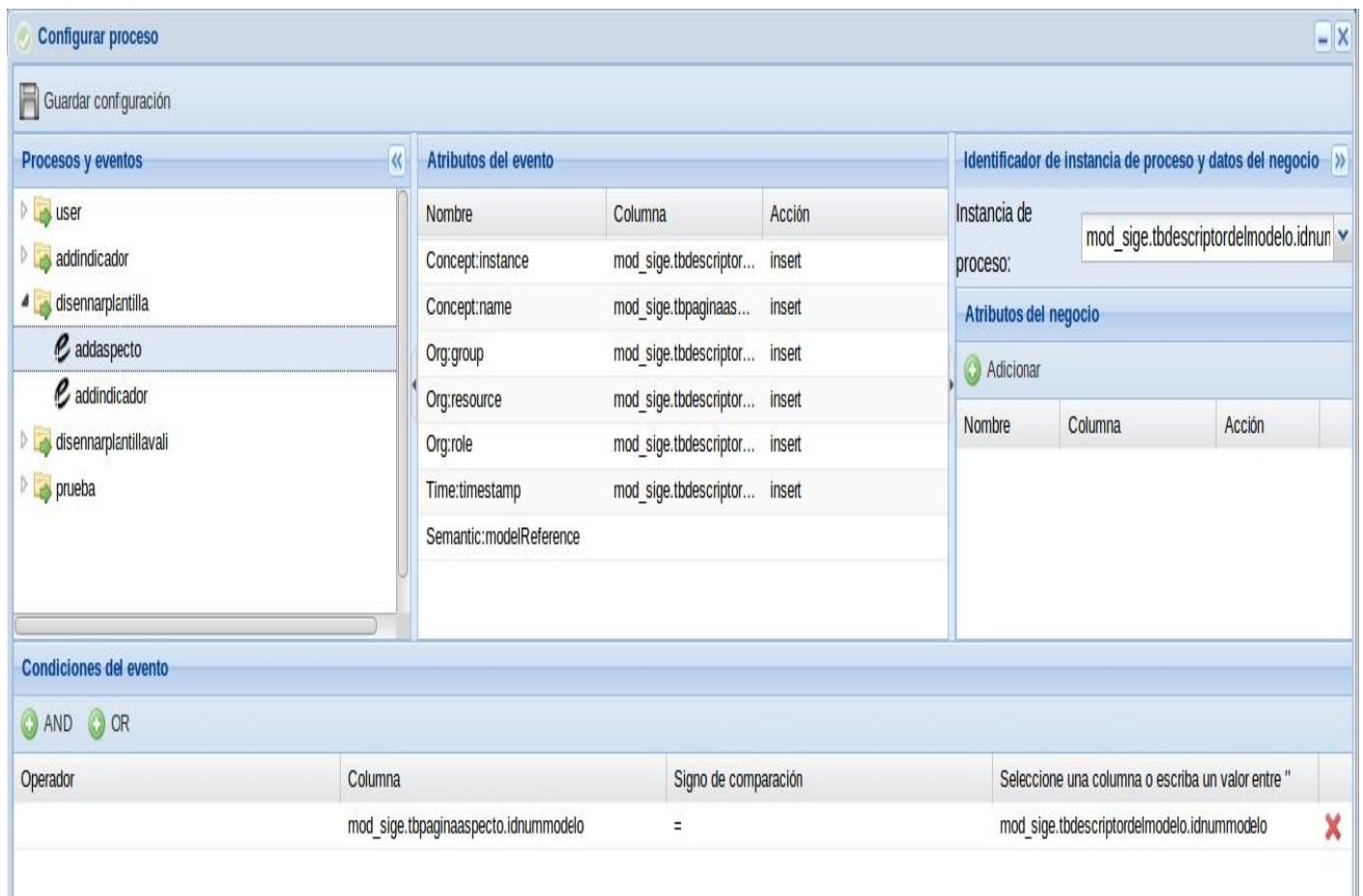


Fig. 21: Interfaz de usuario Configurar Proceso.

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

```
<trace>
  <string key="concept:name" value="Actualizar usuario"/>
  <event>
    <string key="org:resource" value="administrador"/>
    <date key="time:timestamp" value="2015-05-27T17:27:50-0400"/>
    <string key="lifecycle:transition" value="completado"/>
    <string key="concept:name" value="pepe"/>
    <string key="concept:instance" value="142"/>
    <string key="semantic:modelReference" value=" "/>
  </event>
  <event>
    <string key="org:resource" value="administrador"/>
    <date key="time:timestamp" value="2015-05-27T17:27:50-0400"/>
    <string key="lifecycle:transition" value="completado"/>
    <string key="concept:name" value="pepe"/>
    <string key="concept:instance" value="142"/>
    <string key="semantic:modelReference" value=" "/>
  </event>
  <event>
    <string key="org:resource" value="administrador"/>
    <date key="time:timestamp" value="2015-05-27T17:27:50-0400"/>
    <string key="lifecycle:transition" value="completado"/>
    <string key="concept:name" value="pepe"/>
    <string key="concept:instance" value="142"/>
    <string key="semantic:modelReference" value=" "/>
  </event>
</event>
```

Fig. 22: Registro de evento en formato XES.

Conclusiones del capítulo:

En el presente capítulo se elaboró el diagrama de componentes representando la organización y dependencias entre las clases del diseño estructuradas en paquetes de componentes. Se definió un estándar de codificación con el objetivo de garantizar la uniformidad del código en el desarrollo del módulo. Se realizaron pruebas al módulo para garantizar un producto sin errores y con un buen funcionamiento. Para esto se utilizó el método de caja negra acompañado de la técnica de partición de

equivalencia. Para determinar que el software posee una calidad aceptable se mostraron los resultados obtenidos durante la realización de las pruebas y las principales interfaces del mismo.

CONCLUSIONES

Teniendo en cuenta los objetivos trazados en la investigación se arribó a las siguientes conclusiones:

1. Los sistemas estudiados para la extracción de trazas de proceso al depender de la evidencia de la ejecución de procesos se les dificulta la completitud de las trazas debido a la variación de los datos.
2. La obtención de los artefactos necesarios durante el proceso de análisis y diseño permitió obtener como resultado un mayor entendimiento de cómo implementar el Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.
3. Se garantiza que los datos persistan en el tiempo asegurando la transformación de las trazas registradas por SAGI y su exportación a formato XES.
4. La realización de las pruebas así como la resolución de las no conformidades detectadas, permitieron comprobar el correcto funcionamiento del Módulo de Registro y Transformación de Trazas de Eventos para la versión 2.0 de SAGI.

RECOMENDACIONES

Al concluir la investigación se recomienda:

1. Realizar análisis de los procesos de negocio de SAGI, a partir de los ficheros XES que genera el módulo, utilizando las técnicas de Minería de Procesos.

REFERENCIAS BIBLIOGRÁFICAS

1. **Aalst, W.M.P. van der, Weijters, A.J.M.M., Maruster, L. 2012.** Workflow Mining: Discovering process models from event logs. [En línea] 2012.
2. **Adriansyah, A. y Buijs, J.C.A.M. 2012.** Mining process performance from event logs : the BPI Challenge 2012 case study. [En línea] 2012.
3. **Balduino, Ricardo . 2007.** Introduction to OpenUP. [En línea] Agosto de 2007. [Citado el: 26 de Febrero de 2015.] <http://www.eclipse.org/epf/general/OpenUP.pdf>.
4. **Bose, R.P.J.C., Aalst, W.M.P. van der. 2012.** Process diagnostics using trace alignment: Opportunities, issues, and challenges. [En línea] 2012.
5. **Caron, Filip, Vanthienen, Jan y Baesens, Bart . 2012.** Rule-Based Business Process Mining: Applications for Management. *Advances in Intelligent Systems and Computing*. [En línea] 2012. pp 273-282.
6. **Dumas, Marlon , M. van der Aalst, Wil y Hofstede, Arthur H. ter. 2005.** [En línea] 25 de octubre de 2005. [Citado el: 3 de Enero de 2015.] 0471663069.
7. **Eguiluz, Javier . 2011.** symfony.es. *Desarrollo web ágil con Symfony2*. [En línea] 29 de Diciembre de 2011. [Citado el: 6 de Noviembre de 2014.] <http://symfony.es/libro/>.
8. **Günther, Christian W. . 2009.** [En línea] 25 de noviembre de 2009. [Citado el: 9 de febrero de 2015.] <http://www.xes-standard.org/xesstandarddefinition>.
9. **IEEE Task Force On Process Mining. 2011.** Manifiesto sobre Minería de Procesos. [En línea] 2011. [Citado el: 9 de noviembre de 2014.] <http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:pmm-spanish-v1.pdf>.
10. **Ingeniería, Facultad de. 2014.** Centro de Posgrados y Actualización Profesional en Informática. *Universidad de la República de Uruguay*. [En línea] 27 de noviembre de 2014. [Citado el: 27 de noviembre de 2014.] <http://www.fing.edu.uy/node/5736>.

11. **ISO. 2000.** Norma Internacional ISO 9000-2000. [En línea] 2000.
12. **J.C.A.M. , Buijs, Joos. 2010.** Mapping Data Sources to XES in a Generic Way, Technische Universiteit Eindhoven, Eindhoven, The Netherlands. *Master Thesis*. [En línea] marzo de 2010. [Citado el: 9 de noviembre de 2014.] http://www.processmining.org/_media/xesame/xesma_thesis_final.pdf.
13. **NetBeans IDE. 2015.** <https://netbeans.org/features/index.html>. *NetBeans IDE Features*. [En línea] 2015. [Citado el: 26 de Febrero de 2015.]
14. **OMG. 2006.** Business Process Modeling Notation Specification. [En línea] 2006.
15. **Pérez Alfonso, Damián, y otros. 2012.** Conference: VII Conferencia Internacional de Ciencias Empresariales. [En línea] noviembre de 2012. [Citado el: 26 de noviembre de 2014.] https://www.researchgate.net/publication/237843306_Utilizacin_de_tcnicas_de_minera_de_proces_o_en_el_entorno_empresarial_cubano.
16. **PHP Framework Interop Group. 2015.** <http://www.php-fig.org/>. [En línea] 2015. [Citado el: 29 de Marzo de 2015.] <http://www.php-fig.org/psr/psr-1/es/>.
17. **PostgreSQL. 2015.** http://www.postgresql.org.es/sobre_postgresql. [En línea] 2015. [Citado el: 27 de Febrero de 2015.] <http://www.postgresql.org.es/>.
18. **Pressman, Roger S. 2005.** Software Engineering. *A Practitioner's Approach*. New York : Higher Education, 2005.
19. **Rodríguez, Carlos, y otros. 2012.** *Institute of Software Technology & Interactive Systems*. [En línea] 4 de Septiembre de 2012. [Citado el: 25 de Febrero de 2015.] <http://publik.tuwien.ac.at/showentry.php?ID=209003&lang=1>.
20. **Sencha Ext JS. 2015.** <http://www.sencha.com/products/extjs/>. [En línea] 2015. [Citado el: 27 de Febrero de 2015.] <http://www.sencha.com/>.
21. **Symfony. 2015.** *Desarrollo de aplicaciones web con Symfony*. [En línea] 2015. [Citado el: 27 de Febrero de 2015.] <http://symfony.com/doc/current/index.html>.

22. **The Apache Software Foundation. 2015.** The Apache Software Foundation. *Apache HTTP Server Project*. [En línea] 2015. [Citado el: 26 de Febrero de 2015.] <http://httpd.apache.org>.
23. **Van Der Aalst, Wil. 2011.** Process Mining. *Discovery, Conformance and Enhancement of Business Processes*. [En línea] 2011. [Citado el: 9 de noviembre de 2014.] <http://www.google.com/books?hl=en&lr=&id=I1KOAfiqfYc&oi=fnd&pg=PR4&dq=Discovery,+Conformance+and+Enhancement+of+Business+Processes&ots=K-kS69vVBL&sig=WLc3SuDDHQYJbX4dZpraXFzsxGI>. ISBN: 978-3-642-19344-6.
24. **Van Der, Heijden. 2012.** Process Mining Project Methodology. *Developing a General Approach to Apply Process Mining in Practice*. [En línea] agosto de 2012. [Citado el: 9 de noviembre de 2014.] http://alexandria.tue.nl/extra2/afstversl/tm/Van_der_Heijden_2012.pdf.
25. **Visconti, Marcello y Astudillo, Hernán . 2008.** *Fundamentos de Ingeniería de Software*. [En línea] 2008. [Citado el: 15 de Marzo de 2015.] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>.
26. **Visual Paradigm. 2015.** *Visual Paradigm for UML Standard*. [En línea] 2015. [Citado el: 27 de Febrero de 2015.] <http://visual-paradigm-for-uml-standard.soft112.com/>.