

Universidad de las Ciencias Informáticas

Facultad 1



Título: *Módulo de análisis de trazas de procesos del Sistema Único de Identificación Nacional de la República de Cuba*

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: *Ivan Campos Cesar*

Tutor(es): *Ing. Adrian Torres Graña*

Ing. Reynier Blanco Zambrano

La Habana, 2012

...Las ideas nacen de los conocimientos y de los valores éticos. Una parte importante del problema estaría resuelta tecnológicamente, la otra hay que cultivarla sin descanso o de lo contrario se impondrán los instintos más primarios.

La tarea que los graduados de la UCI tienen por delante es grandiosa. Espero que la cumplan, y la cumplirán.

EL ROBO DE CEREBROS, 17 de Julio de 2007.

Reflexiones del Comandante en Jefe Fidel Castro Ruz.

Datos de contacto

Autor: Ivan Campos Cesar

- ✓ Estudiante 5to año, Facultad 1, UCI, Curso 2011 – 2012.

Correo electrónico: icampos@estudiantes.uci.cu

Tutor: Ing. Adrian Torres Graña

- ✓ Ingeniero en Ciencias Informáticas, UCI, 2010.
- ✓ Especialista de Automatización de la Dirección de Tecnologías y Sistema, MININT.

Correo electrónico 1: agrana@uci.cu

Correo electrónico 2: atorres10@graduados.uci.cu

Tutor: Ing. Reynier Blanco Zambrano

- ✓ Ingeniero en Ciencias Informáticas, UCI, 2009.
- ✓ Arquitecto del Sistema Único de Identificación Nacional de la población.

Correo electrónico: rblanco@uci.cu

Dedicatoria

El presente trabajo que representa mi esfuerzo por superarme tanto en la vida profesional como en lo personal va dedicado a:

La revolución cubana, que se ha llevado a cuestras la tarea de hacer una sociedad con todos y para el bien de todos.

María M. Cesar Rodríguez e Ivan E. Campos Llácer, mis padres; que me han dado 24 años de sus vidas para hacerme crecer y aprender a enfrentarme a la vida por muy drástico que sean los acontecimientos.

Juan Armando Cesar y a Onel Campos, mis abuelos; los que siempre quisieron ver a su nieto llegar hasta aquí y para ello trabajaron hasta el último minuto de sus vidas con tal de que de ellos aprendiera lo mejor.

Tita, mi hermana; la que también me ha enseñado que siempre que se desea, se le puede dar un giro a la vida y se puede ser perseverante.

Ivan Campos Cesar

Agradecimientos

Mis agradecimientos van dirigidos a todas las personas que de una forma u otra me han permitido llegar a este momento.

A María M. Cesar, mi madre; por haberme guiado siempre por el camino que me ha servido para llegar hasta aquí. Agradecer que me hayas enseñado que la perseverancia es la clave para llegar a cualquier sitio.

A Ivan E. Campos, mi padre; por haberme enseñado que no hace falta tener a un grupo de personas para construir nuestra casa, que con nosotros dos y los deseos de hacerlo bastaba. Agradezco me hayas enseñado que uno puede lograr siempre lo que se proponga.

A los dos les agradezco me hayan enseñado que la familia lo es todo.

A mi abuelo Armando, que siempre quiso ver este momento, en el que uno de sus nietos se hiciera un profesional.

A mi abuelo Onel, que me enseñó que no hay obstáculos que detengan a una persona que tiene deseos de vivir.

Al resto de mis familiares, que siempre me han apoyado y me han dado lo mejor de sí.

A mis tutores, que sin ellos no hubiera llegado hasta aquí.

A mis amistades, algunas de las cuales me han acompañado durante mis 24 años y en su mayoría han estado siempre para apoyarme, divertirme o pasar un mal momento. En especial a Jose Miguel, Mary, Lisi, Vida, Yennia y Yonnis que siempre han estado pendiente de mi como persona y amigo.

A Fidel y a la Revolución por haber creado una universidad como esta.

Ivan Campos Cesar

Declaración de autoría

Declaro que soy el único autor de este trabajo y se le reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste se firma la presente a los ___ días del mes _____ del año _____.

Firma del Autor
Ivan Campos Cesar

Firma del Tutor
Ing. Adrian Torres Graña

Firma del Tutor
Ing. Reynier Blanco Zambrano

Opinión de los tutores

Firma del Tutor

Ing. Adrian Torres Graña

Firma del Tutor

Ing. Reynier Blanco Zambrano

Resumen

La minería de procesos es una disciplina de investigación joven ubicada entre el aprendizaje computacional y la minería de datos, y el modelado y análisis de procesos (1). Las técnicas y algoritmos de minería de procesos concentran al proceso como un todo. Estos son los encargados de dar tratamiento a la información extraída de las trazas que dejan los sistemas que automatizan los principales procesos de la sociedad. El trabajo propone un módulo de análisis de las trazas de los procesos que son ejecutados por el Sistema Único de Identificación Nacional (SUIN), que utiliza técnicas de minería de procesos para el descubrimiento, la detección de cuellos de botella y la realización de predicciones de tiempo. El principal objetivo es analizar las trazas generadas por el SUIN y poder realizar mejoras en la ejecución de los procesos. Para realizar la investigación, son utilizados los métodos histórico – lógico, analítico – sintético, hipotético – deductivo y la modelación, así como la entrevista. Se realiza un estudio de las principales soluciones que realizan análisis de trazas, así como de las técnicas y algoritmos de minería de procesos existentes. La investigación concluye con la selección del algoritmo alfa para el descubrimiento de los procesos. Además emplea los análisis operacionales y de rendimiento para la detección de cuellos de botella y las predicciones de tiempo. Se obtiene como resultado una solución que permite realizar el análisis de las trazas generadas por los procesos de negocio del SUIN que posibilita identificar mejoras en la ejecución de dichos procesos.

Palabras claves: *minería de procesos, proceso, trazas.*

Índice de contenido

Índice de tablas..... X

Índice de figuras XII

Introducción 13

Capítulo 1 Fundamentación teórica 18

 Introducción..... 18

 1.1 Enfoque a procesos..... 18

 1.2 Análisis de técnicas, algoritmos y soluciones existentes.....21

 1.2.1 Técnicas y algoritmos21

 1.2.2 Soluciones existentes25

 1.2.3 Solución propuesta30

 1.3 Ambiente de desarrollo.....30

 1.3.1 Metodología de desarrollo.....30

 1.3.2 Lenguajes utilizados.....31

 1.3.3 Plataforma de desarrollo33

 1.3.4 Control de versiones36

 1.3.5 Manejo de datos36

 1.3.6 Herramienta de modelado.....37

 Conclusiones parciales.....38

Capítulo 2 Solución propuesta39

 Introducción.....39

 2.1 Modelo de dominio39

 2.1.1 Descripción del modelo de dominio.....40

 2.2 Descripción de roles41

2.3 Catálogo de requisitos	42
2.3.1 Especificación de los requisitos funcionales.....	42
2.3.2 Descripción de los requisitos.....	43
2.4 Especificación de los requisitos no funcionales	46
2.5 Arquitectura	47
2.6 Patrones de diseño.....	49
2.7 Especificación de clases.....	53
2.7.1 Diagrama de clases del diseño	53
2.7.2 Clases conectoras.....	54
2.7.3 Clases gestoras	55
2.7.4 Clases entidad	56
2.8 Modelo de datos.....	56
Conclusiones parciales.....	58
Capítulo 3 Implementación y pruebas	59
Introducción.....	59
3.1 Estándares de codificación.....	59
3.2 Tratamiento de errores	60
3.3 Implementación	61
3.3.1 Diagrama de componentes	61
3.3.2 Diagrama de despliegue	62
3.4 Prototipos de interfaz del módulo	63
3.5 Pruebas.....	65
3.5.1 Pruebas de caja blanca.....	66
3.5.2. Pruebas de caja negra	67

3.6 Beneficios del módulo	70
Conclusiones parciales.....	70
Conclusiones generales.....	72
Recomendaciones	73
Bibliografía.....	74
Anexos.....	79
Anexo 1: Descripción de requisitos funcionales.....	79
Anexo 2: Pruebas unitarias	93
Anexo 3: Casos de prueba	94
Anexo 4: Prototipos de interfaz del módulo	104
Anexo 5: Diagrama de clases del diseño.....	109
Anexo 6: Modelo de datos.....	110

Índice de tablas

Tabla 1 Descripción de roles.....41

Tabla 2 Requisitos funcionales del módulo.42

Tabla 3 Especificación del RF 3 Buscar procesos según criterio de búsqueda.45

Tabla 4 Descripción de la clase conectora XmlProcessConnector.....55

Tabla 5 Descripción de la clase gestora MiningProcessController.56

Tabla 6 Descripción de la clase entidad Process.56

Tabla 7 Descripción del CPRF 1 Extraer datos de los sistemas fuente.69

Tabla 8 Resultado de la ejecución del CPRF 1 Extraer datos de los sistemas fuente.69

Tabla 9 Descripción del RF 1 Extraer datos de los sistemas fuente.80

Tabla 10 Descripción del RF 2 Transformar datos.81

Tabla 11 Descripción del RF 3 Buscar procesos según criterio de búsqueda.83

Tabla 12 Descripción del RF 4 Mostrar datos de los procesos, instancias de procesos y actividades.....85

Tabla 13 Descripción del RF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa. 86

Tabla 14 Descripción del RF 6 Analizar rendimiento del proceso.....87

Tabla 15 Descripción del RF 7 Mostrar resultados del análisis de rendimiento.88

Tabla 16 Descripción del RF 8 Mostrar cuellos de botella.....89

Tabla 17 Descripción del RF 9 Analizar operacionalmente el proceso.....90

Tabla 18 Descripción del RF 10 Mostrar predicciones de tiempo.....91

Tabla 19 Descripción del RF 11 Generar fichero *.mxml.....92

Tabla 20 Descripción del CPRF 1 Extraer datos de los sistemas fuente.95

Tabla 21 Resultado de la ejecución del CPRF 1 Extraer datos de los sistemas fuente.95

Tabla 22 Descripción del CPRF 3 Buscar procesos según criterio de búsqueda.97

Tabla 23 Resultado de la ejecución del CPRF 3 Buscar procesos según criterio de búsqueda.98

Tabla 24 Descripción del CPRF 4 Mostrar datos de los procesos, instancias de procesos actividades. ...100

Tabla 25 Descripción del CPRF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.
..... 101

Tabla 26 Descripción del CPRF 7 Mostrar resultados del análisis de rendimiento. 101

Tabla 27 Descripción del CPRF 8 Mostrar cuellos de botella. 102

Tabla 28 Descripción del CPRF 10 Mostrar predicciones de tiempo. 102

Tabla 29 Descripción del CPRF 11 Generar fichero *.xml 103

Índice de figuras

Fig. 1 Modelo de dominio.....	40
Fig. 2 Arquitectura del módulo de análisis de trazas de procesos.....	48
Fig. 3 Ejemplo del patrón de diseño subclase.....	50
Fig. 4 Ejemplo del uso del patrón Controlador.....	53
Fig. 5 Fragmento del Diagrama de clases de diseño.....	54
Fig. 6 Fragmento del modelo de datos.....	57
Fig. 7 Diagrama de componentes.....	62
Fig. 8 Diagrama de despliegue.....	63
Fig. 9 Prototipo de interfaz para la selección de la fuente de datos.....	64
Fig. 10 Prototipo de interfaz para buscar procesos desde ficheros MXML.....	65
Fig. 11 Prueba unitaria aplicada al método ConvertPath.....	66
Fig. 12 Resultados de las pruebas unitarias.....	67
Fig. 13 Resultados de las iteraciones.....	70
Fig. 14 Prueba unitaria aplicada al método GetAllProcess.....	93
Fig. 15 Prueba unitaria aplicada al método GetProcessById.....	93
Fig. 16 Prototipo de interfaz para la selección de la fuente de datos.....	104
Fig. 17 Prototipo de interfaz para buscar procesos desde ficheros MXML.....	105
Fig. 18 Prototipo de interfaz para mostrar datos de procesos, instancias de proceso y actividades seleccionadas.....	106
Fig. 19 Prototipo de interfaz para mostrar el modelo de descubrimiento del proceso seleccionado.....	106
Fig. 20 Prototipo de interfaz para mostrar los resultados del análisis de rendimiento.....	107
Fig. 21 Prototipo de interfaz para mostrar cuellos de botella.....	107
Fig. 22 Prototipo de interfaz para mostrar predicciones de tiempo.....	108
Fig. 23 Diagrama de clases del diseño extendido.....	109
Fig. 24 Modelo de datos extendido.....	110

Introducción

Los procesos son una parte integral del mundo de hoy, en los servicios de manejo y en funciones internas en empresas, organismos gubernamentales y organizaciones. Si bien hay gran cantidad de sistemas disponibles para apoyar la ejecución de tales procesos, las prácticas actuales para el seguimiento y el análisis de esta ejecución dejan mucho que desear.

Los sistemas de información se están haciendo cada vez más entrelazados con los procesos y las operaciones que soportan. Por consiguiente, muchos de los eventos que se desencadenan en la ejecución de las aplicaciones necesitan ser seguidos y controlados. Para ello, estos son grabados en forma de trazas por los sistemas que automatizan los principales procesos de la sociedad. Estas trazas permiten comprobar que se realizan exactamente las funciones esperadas, y no otras. También son empleadas con el fin de monitorear las acciones que se realicen (acceso a ficheros, dispositivos, empleo de los servicios, etc.), y para detectar anomalías que puedan afectar la estabilidad o el correcto funcionamiento del sistema. Para ello se recurre a productos de software muy potentes y modulares que, entre otras funciones, rastrean los caminos que siguen los datos a través del programa. (1)

Para realizar un seguimiento y control sobre los procesos ejecutados por los sistemas de información, el auditor informático emplea, preferentemente, la información que genera el propio sistema. Se hace una revisión especial en el registro oficial de eventos (LOG), donde se recogen las modificaciones de los datos y se detalla la actividad general. (1) El análisis que se le realiza a los LOG es también extensible a los procesos de negocio que en estos registros se graban. El análisis de estos procesos va más allá de verificar las modificaciones realizadas; dicho análisis permite determinar cuellos de botella, mostrar puntos de decisión y hacer aplicables perspectivas a los procesos en función de conocer otros datos concernientes a su ejecución. Sin embargo, las organizaciones tienen problemas para extraer el valor de los datos registrados en las trazas; existiendo la minería de procesos, que dentro de la informática, constituye una vía de solución a dicho problema. El objetivo de la minería de procesos es extraer de los datos de eventos, la información relacionada con los procesos, para descubrir automáticamente un modelo de proceso, observando los eventos grabados por un sistema. (2) Para la realización de un análisis exhaustivo es recomendable la aplicación de técnicas y algoritmos de minería de procesos, mediante los cuales se puedan obtener modelos y datos que muestren lo que realmente sucede en la ejecución de los mismos.

En la Universidad de las Ciencias Informáticas (UCI), de conjunto con la Dirección de Identificación y Registro (DIR) del Ministerio del Interior (MININT), se trabaja en el desarrollo del Sistema Único de Identificación Nacional de la República de Cuba (SUIN); sistema que tiene como objetivo el control estricto de la identificación y registro de la población nacional o extranjera que reside temporal o permanentemente en el país. Está diseñado bajo el precepto de la creación de una Base de Datos Nacional, que se retroalimenta a partir de la información vigente y de la resultante de los distintos trámites que se realizan en las oficinas de identificación y registros. El SUIN es un sistema encargado de automatizar uno de los procesos más importantes de la sociedad. Teniendo en cuenta esto, debe ser un sistema ágil que no implique demoras que atenten contra la eficiencia en la realización de los trámites.

Es una solución desarrollada con un enfoque a procesos y permite la gestión de los diferentes procesos requeridos para la ejecución segura y eficiente de los trámites de identificación. De los mismos se generan trazas que permiten analizar y monitorear los procesos a partir de la información parcial¹ e histórica². Sin embargo, no cuenta con una herramienta que realice el análisis de los procesos, de manera que muestre en caso de demoras en un trámite, la causa o ubicación en la ejecución del proceso.

Algunos de los sistemas existentes que permiten el análisis de la información registrada por los procesos constituyen aplicaciones que consumen demasiado recurso computacional o no cumplen con algunos requerimientos. Dichos requerimientos están enfocados a garantizar tener en una misma aplicación web, que permita ser integrada al resto del SUIN, todas las funcionalidades que garanticen una visión profunda de lo que ocurre en la ejecución de los procesos, y con ello, poder aplicar mejoras o tomar decisiones. Entre estas demandas se pueden citar que el consumo de recursos computacionales esté acorde con las estaciones de trabajo, que se realice un análisis y descubrimiento de los procesos a partir de técnicas y algoritmos de minería de procesos. Además, se requiere que se pueda tener acceso desde cualquier estación de trabajo y que se obtengan informaciones medulares sobre los procesos y su comportamiento real. Se necesita también, la realización de otros análisis a los procesos tales como los operacionales y los de rendimiento. Estos últimos con el objetivo de identificar cuellos de botella y realizar predicciones de tiempo.

¹ Información almacenada sobre procesos que se encuentran en ejecución.

² Información almacenada sobre procesos que han sido ejecutados con anterioridad.

Dada la situación problemática planteada se define como **problema científico** de la investigación: ¿Cómo diseñar e implementar un módulo para el análisis de las trazas generadas por los procesos de negocio ejecutados en el Sistema Único de Identificación Nacional de la República de Cuba?

El **objeto de estudio** de la investigación está determinado por el análisis de las trazas generadas por los sistemas de información. El **campo de acción** se enmarca en el análisis de las trazas generadas por los procesos de negocio ejecutados en el Sistema Único de Identificación Nacional de la República de Cuba.

El **objetivo general** de la investigación es desarrollar un módulo que permita analizar mediante técnicas de minería de procesos, las trazas generadas por el Sistema Único de Identificación Nacional de la República de Cuba.

Para darle cumplimiento al objetivo planteado se realizarán las siguientes **tareas de la investigación**:

1. Estudiar los principales conceptos asociados al análisis de procesos de negocio.
2. Estudiar y caracterizar el estado actual de las diferentes soluciones relacionadas con el análisis de las trazas generadas por los sistemas de información.
3. Estudiar las técnicas y algoritmos de minería de procesos existentes.
4. Seleccionar las técnicas y algoritmos de minería de procesos que permitan el análisis de las trazas.
5. Realizar un estudio que permita fundamentar las herramientas informáticas y metodologías a utilizar en el desarrollo de la solución.
6. Definir los requisitos funcionales y no funcionales del módulo de análisis de trazas de procesos del SUIN.
7. Diseñar los prototipos de interfaz de usuario.
8. Realizar la descripción de los requisitos.
9. Realizar el diseño del módulo de análisis de trazas de procesos del SUIN.
10. Implementar el módulo de análisis de trazas de procesos del SUIN.
 - a) Implementar el buscador de trazas de procesos.
 - b) Implementar las técnicas y algoritmos de minería de procesos seleccionados.
 - c) Implementar la capa de acceso a datos.
 - d) Integrar el módulo implementado con el módulo de administración del SUIN.
11. Llevar a cabo las pruebas de aceptación.

- a) Realizar las pruebas unitarias.
 - b) Diseñar los casos de prueba.
12. Realizar las pruebas funcionales.

Se plantea como **hipótesis**: si se desarrolla un módulo para el análisis de trazas utilizando técnicas de minería de procesos, se logrará la detección de cuellos de botella, así como la realización de predicciones de tiempo en los procesos de negocio previamente ejecutados en el Sistema Único de Identificación Nacional de la República de Cuba.

Para asistir las tareas de la investigación se emplearán los siguientes **métodos científicos**:

Métodos teóricos:

1. **Histórico lógico**: Permite hacer un estudio de la problemática, así como analizar la evolución y el desarrollo de otros sistemas de análisis de trazas a niveles nacional e internacional.
2. **Analítico – sintético**: Se utiliza para procesar toda la información relacionada con la investigación realizada y de ella obtener las ideas fundamentales que permiten definir las que serán utilizadas para desarrollar la propuesta de solución.
3. **Hipotético – deductivo**: Se utiliza para, a partir de la hipótesis, llegar a nuevos conocimientos y predicciones. Se aplica en el análisis y construcción de la teoría científica.
4. **Modelación**: Permite la creación de modelos que visualizan una reproducción resumida de la realidad, relacionadas con el objeto de estudio y la solución que se propone.

Métodos empíricos:

- ✓ **Entrevista**: Se utiliza la entrevista como una conversación planificada con los futuros usuarios para obtener información acerca del problema que se investiga, influye en el análisis y diseño del producto de software.

Al término de la investigación, se podrá incluir en el SUIN un módulo que permita analizar mediante técnicas de minería de procesos, las trazas generadas por este sistema. Permitirá identificar cuellos de botella y realizar predicciones en la ejecución de los procesos. Posibilitando que estos resultados puedan ser usados como soporte para la toma de decisiones y la realización de posibles mejoras en la ejecución de los procesos.

El presente documento consta de tres capítulos, desarrollados a partir del estudio realizado de la información bibliográfica e institucional existente. La descripción de los mismos se presenta a continuación:

- ✚ **Capítulo 1: Fundamentación teórica de la investigación:** cuenta con el respaldo teórico de los temas tratados en el trabajo de investigación, necesarios para el correcto entendimiento de la solución propuesta. Se describen los conceptos fundamentales asociados al dominio del problema y al objeto de estudio, realizándose un análisis de la situación actual. Se presenta el ambiente de desarrollo necesario para el análisis y la implementación de la propuesta de solución.

- ✚ **Capítulo 2: Solución propuesta:** se describen las actividades del flujo de procesos a través de un modelo de dominio, el cual sirve de base para determinar lo que se va a desarrollar. Además se realiza el levantamiento de requisitos que contiene la definición de los requisitos funcionales y no funcionales del módulo, conjuntamente con los roles que intervienen. Se realiza el modelo de diseño del sistema que se va a implementar, en el que se definen y especifican el conjunto de clases a utilizar. Se describen la arquitectura y los patrones que serán usados para el desarrollo del módulo, y se presentan las funcionalidades y el modelo de datos usado.

- ✚ **Capítulo 3: Implementación y pruebas:** se describen los aspectos relacionados con el flujo de implementación y la validación del producto. Se presentan los estándares de codificación empleados, así como el tratamiento a errores dado. Se detallan los diagramas correspondientes al flujo de implementación y se describen los prototipos de interfaz de usuario. Para comprobar el correcto funcionamiento del módulo, se le realizan pruebas de caja blanca y de caja negra, las que permiten verificar la calidad del módulo.

Capítulo 1 Fundamentación teórica

Introducción

En cualquier entidad donde se empleen sistemas informáticos es de gran prioridad mantener un control y seguimiento de su funcionamiento. De ahí la importancia que se le da al análisis de las trazas o LOG generados por los sistemas. Estos permiten un mejor control de todas las actividades que se desarrollan y la recuperación ante un incidente de seguridad. Pueden ser usados como evidencia legal, así como para la realización de análisis operacionales y de rendimiento del sistema. Permiten además, la detección de anomalías en las ejecuciones, y con ello poder determinar cuál o cuáles pueden llevar a fallos en la seguridad del sistema.

En el presente capítulo se expone la fundamentación teórica de la investigación, la cual da respaldo a la solución propuesta. Se abordan los principales conceptos asociados al dominio del problema y de forma general, se ofrece una visión del ambiente de desarrollo utilizado para la implementación de la solución.

1.1 Enfoque a procesos

La norma internacional ISO-9001 define un proceso como “una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados”. (3) Oscar Barros hace una importante distinción al introducir el concepto de valor agregado en la definición de proceso. Señala que “un proceso es un conjunto de tareas lógicamente relacionadas que existen para conseguir un resultado bien definido dentro de un negocio; por lo tanto, toman una entrada y le agregan valor para producir una salida. (4) Thomas Davenport, uno de los pioneros de la reingeniería, señala que un proceso, simplemente, es “un conjunto estructurado, medible de actividades diseñadas para producir un producto especificado, para un cliente o mercado específico. Implica un fuerte énfasis en “cómo” se ejecuta el trabajo dentro de la organización, en contraste con el énfasis en el “qué”, característico de la focalización en el producto.” (5)

Tomando en cuenta lo anteriormente expuesto, y enfocándolo al marco de la investigación, en este caso, los procesos de negocio consisten en subprocesos, instancias y actividades ejecutadas en un sistema. Un subproceso es parte de un proceso de mayor nivel que tiene su propia meta, propietario, entradas y salidas. Las actividades son partes de los procesos de negocio que no incluyen ninguna toma de decisión

ni vale la pena descomponer (aunque ello sea posible). Por ejemplo, “*Introduce Carné de Identidad*”, “*Registra los datos*”.

Minería de procesos

Según Wil M.P Van Der Aalst en su publicación “*Minería de Procesos. Descubrimiento, Ajustes y Mejoras de los Procesos de Negocio*”: la minería de procesos es una disciplina de investigación joven que permanece entre el aprendizaje de computadora y la minería de datos por un lado, y el modelado de procesos y el análisis por el otro. La idea es descubrir, monitorear y mejorar procesos legítimos extrayendo conocimientos de registros de eventos fácilmente disponibles en los sistemas. (2)

La idea básica de la minería de procesos es aprender de las ejecuciones observadas de un proceso y descubrir nuevos modelos. Por ejemplo, formular una red de Petri que puede reproducir un comportamiento observado, por el cual verificar la conformidad de un modelo verificando si el modelo de comportamiento hecho, corresponde al comportamiento observado, y en el que ofrecer la información extraída de los registros de eventos. (2)

Tradicionalmente, la minería de procesos ha sido usada para extraer modelos de registro de eventos y verificar o prolongar modelos existentes. Esto se ha visto útil para la mejora de procesos y su soporte. La minería de procesos, sobre la base de la información histórica grabada en registros de eventos, a menudo suministra una perspicacia sorprendente a directores, desarrolladores de sistemas, auditores, y usuarios finales. En muchos casos, no son explotadas a cabalidad sus particularidades, ni son tomados sus resultados como soporte para la toma de decisiones. Sin embargo, los algoritmos y las técnicas de minería de procesos existentes se concentran en el proceso como un todo. Estos algoritmos son los encargados de dar tratamiento a la información extraída de los registros de eventos en función de lo que se necesite: el descubrimiento, el chequeo de conformidades o la extensión. Se toma en cuenta además que la minería de procesos en el análisis de los procesos permite la aplicación de perspectivas tales como las de control de flujo, la organizacional, de casos y la perspectiva de tiempo. Dentro de los algoritmos de descubrimiento más usados se contemplan: *Algoritmo Alfa*, *Algoritmos de Minería Heurística*, *Algoritmos de Minería Genética* y *Algoritmos de Minería Difusa*. Por su parte, para el chequeo de conformidades y

extensión son usadas técnicas tales como las LTL³, para corroborar la existencia de anomalías en la ejecución de los procesos.

La aplicación de perspectivas como parte del análisis de procesos, permite ofrecer informaciones en forma de modelos y gráficos que apoyan el trabajo del auditor de sistemas de información. La perspectiva de control de flujo se concentra en el orden de ejecución de las actividades, su objetivo es encontrar una buena caracterización del proceso localizando todos los posibles caminos de su ejecución. Por otro lado la perspectiva organizacional se concentra en la información sobre los recursos que intervienen en los procesos, su objetivo es mostrar la estructura del proceso en términos de roles, agrupando a los actores según sean comunes en la ejecución de alguna actividad y mostrándolos a modo de una red. La perspectiva de casos se adentra en las propiedades de los casos, los que pueden ser caracterizados por sus ejecutores o su forma de ejecución dentro del proceso, también por los valores que este posea. Por su parte, la perspectiva de tiempo va más enfocada a los tiempos y frecuencia de ocurrencia de los eventos, esto permite la detección de los cuellos de botella, determinar la medida de los niveles de servicios, así como el monitoreo del empleo de los recursos y la realización de predicciones de tiempo. (2)

La minería de procesos define el trabajo con los LOG, y las trazas, elementos que son necesarios definir, enfocados a su función en la investigación.

LOG

Registro oficial de eventos durante un período de tiempo en particular. Para los profesionales en seguridad informática un LOG es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación. La mayoría de los LOG son almacenados o desplegados en el formato estándar ASCII, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones. (6)

Traza

La traza de un algoritmo (o programa) indica la secuencia de acciones (instrucciones) de su ejecución, así como, el valor de las variables del algoritmo (o programa) después de cada acción (instrucción). Las que son utilizadas para detectar indicios de hechos relevantes a los efectos de la seguridad que puedan afectar la estabilidad o el funcionamiento del sistema informático. (1)

³ Por sus siglas en inglés: *Linear Temporal Logic*.

1.2 Análisis de técnicas, algoritmos y soluciones existentes

En esta sección se describen las características, ventajas y desventajas de las técnicas, algoritmos y las soluciones existentes asociadas al análisis de trazas. Como resultado del siguiente análisis se fundamenta la necesidad de desarrollar la propuesta de solución en este trabajo de diploma.

1.2.1 Técnicas y algoritmos

Las técnicas y los algoritmos de minería de procesos son capaces de extraer conocimiento de los registros de eventos comúnmente disponibles en los sistemas de información actuales. Estos proveen nuevos medios para descubrir, monitorear y mejorar los procesos en una variedad de dominios de aplicación. Hay dos razones principales para el creciente interés en minería de procesos. Por un lado, se registran más y más eventos, proporcionando información detallada acerca de la historia de los procesos. Por otro lado, hay una necesidad de mejorar y apoyar los procesos de negocio en ambientes competitivos y que cambian rápidamente. (7)

Análisis de procesos

El análisis de los procesos de negocio en la minería de procesos está basado en el tratamiento de la información extraída de las fuentes de datos correspondientes. Es empleada en función de detectar anomalías en los procesos, dar seguimiento a alguno en específico, detectar cuellos de botella y realizar análisis operacionales y de rendimiento. Concibe además, la aplicación de perspectivas tales como las perspectivas de control de flujo, organizacional, de casos y la perspectiva de tiempo. Estas perspectivas brindan información que puede ser utilizada en pos de una mejor planificación o ejecución de los procesos.

Análisis de rendimiento

El rendimiento de un proceso se puede definir de diferentes maneras. Por lo general, se identifican tres dimensiones del desempeño: tiempo, costo y calidad. Para cada una de estas dimensiones de desempeño, se pueden definir diferentes indicadores claves de rendimiento (KPIs⁴). Al mirar la dimensión temporal de los siguientes indicadores de desempeño se pueden identificar sus objetivos y metas:

⁴ Por sus siglas en inglés: *Key Performance Indicators*.

- **El período de gestación** (también conocido como **tiempo de flujo**) es el tiempo total desde la creación del caso⁵ hasta la finalización del mismo. En términos de una WF-net⁶, este es el tiempo que tarda en ir desde su lugar de fuente i hasta el sumidero o . Uno puede medir el período de gestación corriente sobre todos los casos. Sin embargo, el grado de la discrepancia podría ser también importante, es decir, supone una diferencia si todos los casos toman más o menos dos semanas o si algunos toman sólo unas horas mientras que los otros toman más de un mes. El *nivel de servicio* es el porcentaje de casos con un período de gestación inferior a cierto valor de umbral, por ejemplo, el porcentaje de casos tratados en dos semanas.
- **El tiempo de servicio** es el tiempo realmente trabajado en un caso. Se puede medir el tiempo de servicio por actividad. Por ejemplo, el tiempo promedio necesario para tomar una decisión es de 35 minutos, o para todo el caso. Cuando se trata de concurrencia, el tiempo de servicio general (es decir, sumando los tiempos invertidos en las diversas actividades) puede ser más largo que el tiempo de espera o gestación. Sin embargo, normalmente el tiempo de servicio es sólo una fracción del tiempo de espera o gestación (en minutos en lugar de semanas).
- **El tiempo de espera** es el tiempo que un caso está a la espera de que un recurso esté disponible. Este tiempo puede ser medido por la actividad o para el caso en su conjunto.
- **El tiempo de sincronización** es el tiempo que una actividad todavía no está totalmente habilitada y a la espera de un disparador externo u otra rama paralela. A diferencia del tiempo de espera, la actividad no está totalmente activada, es decir, el caso está esperando para la sincronización en lugar de un recurso.

Considerando que el control se centra en la corrección (lógica) del proceso de modelado, el análisis de rendimiento tiene como objetivo mejorar los procesos con respecto al tiempo, costo o calidad. (2)

Al evaluar la información que proporciona el análisis de rendimiento se concluye en que su implementación constituye una opción viable para apoyar la solución de la situación problemática planteada. Los indicadores de desempeño: período de gestación y tiempo de servicio brindan información valiosa

⁵ Del inglés *Case*: refiérase a instancia de proceso.

⁶ Red de Workflow.

referente al comportamiento de los procesos ejecutados. Estas informaciones son imprescindibles para llevar a cabo la identificación de los cuellos de botella.

Análisis operacional

El análisis operacional, no se limita a los datos de eventos históricos. También se pueden reproducir las trazas parciales de los casos aún en marcha. Esto se puede utilizar para detectar desviaciones en tiempo de ejecución. Por lo tanto, una alerta puede ser generada antes de que el caso se complete. Igualmente, es posible predecir el tiempo de procesamiento restante o la probabilidad de ser rechazado un caso. Estas predicciones también se pueden utilizar para recomendar los siguientes pasos adecuados para avanzar en el caso. (2)

Teniendo en cuenta la ventaja que proporciona el análisis operacional, y su aporte en la mejora de los procesos de negocio, es seleccionado para ser implementado en la propuesta de solución. Este análisis permitirá llevar a cabo las predicciones de tiempo.

Minería heurística

Los algoritmos de minería heurística usan una representación similar a las redes causales. Estos algoritmos toman en cuenta las frecuencias de los eventos y las secuencias cuando construyen el modelo del proceso. La idea básica es que los senderos infrecuentes no sean incluidos en el modelo, y el uso de las frecuencias hace el enfoque mucho más robusto que la mayoría de los otros enfoques. El objetivo de la minería heurística es extraer una red causal del registro de eventos. Los nodos del gráfico de dependencia corresponden a cada juego de actividades y los arcos a las relaciones de dependencia. (2)

El enfoque presentado por la heurística es muy genérico y puede ser aplicado a otras representaciones. Puede formular modelos jerárquicos y las actividades menos frecuentes pueden ser cambiadas a subprocesos.

El algoritmo de minería heurística es una extensión muy fuerte del algoritmo alfa, pero también tiene dificultades con los conceptos de espacios opcionales y libres. Esto último, trae como dificultad que existan actividades opcionales que no sean contempladas en el modelo. Por esta causa, y debido a que las actividades con menos frecuencia de ejecución estarían siendo eximidas del modelo, no se contemplan estos algoritmos para el desarrollo de la solución.

Minería genética

Los algoritmos de minería genética usan enfoques evolutivos mediante procedimientos iterativos para imitar el proceso de la evolución natural. Tales enfoques no son deterministas y dependen de la aleatoriedad para encontrar las nuevas alternativas. Muchas decisiones de diseño tienen que ser tomadas para el desarrollo de un algoritmo de minería de procesos genético. El enfoque describe una variante de redes causales similares a la notación usada para el algoritmo de minería heurístico, sin embargo, muchas otras representaciones pueden ser posibles. (2)

La minería de procesos genética es flexible y robusta. De la misma manera que las técnicas de minería heurísticas, puede arreglárselas con el ruido y la no terminación. El enfoque también puede ser adaptado y extendido fácilmente. Desafortunadamente, de la misma manera que la mayoría de los enfoques evolutivos, la minería de procesos genética no es muy eficiente para los modelos de grandes registros de eventos. Podría tardar mucho tiempo el descubrir un modelo.

El SUIN constituye un sistema de gran envergadura que será empleado para todos los procesos de registro e identificación de la población. Por tales razones, será capaz de generar gran cantidad de información en sus registros oficiales de eventos, esto impide que la minería genética sea empleada como técnica para el descubrimiento de los procesos que en él se ejecutan.

Minería difusa

Constituye una alternativa para simplificar el modelo de procesos. Este enfoque ve modelos de procesos como si fueran mapas geográficos (mapas de carreteras o mapas de excursión). Dependiendo del mapa, los caminos insignificantes y las ciudades pueden ser retirados, y las calles y los suburbios pueden ser fusionados en estructuras más grandes. De esta manera, el enfoque provee un juego extensible de parámetros para determinar las actividades y los arcos que tienen que ser incluidos. Además, el enfoque puede formular modelos jerárquicos. Las actividades menos frecuentes pero que están atentamente relacionadas con las otras actividades frecuentes son agrupadas en subprocesos. (2)

Debido a que el objetivo de la solución es que se pueda mostrar el comportamiento real de los procesos ejecutados por el SUIN; la simplificación que hace el algoritmo de minería difusa no sería una opción factible para cumplir este objetivo. Por tales razones no será empleado dicho algoritmo para el descubrimiento de los procesos.

Algoritmo alfa

El algoritmo alfa es una técnica de descubrimiento que toma la información existente en un registro de eventos y produce un modelo sin usar ninguna otra información a priori, explicando en una red de Petri el comportamiento registrado. Si el LOG también contiene información sobre algunos recursos, con el algoritmo también se pueden descubrir modelos de recursos relacionados por ejemplo: una red social que muestra cómo trabajan en conjunto las personas en una organización. Esto ayuda a ubicar los puntos de decisión en el proceso.

El algoritmo es simple y muchas de sus ideas han sido arraigadas en las técnicas más complicadas y robustas. Además, garantiza producir un modelo de procesos correcto que no contiene actividades duplicadas, ni actividades que no están registradas en el LOG.

El algoritmo alfa garantiza producir un modelo de procesos correcto. El modelo resultante no contiene actividades duplicadas (dos transiciones etiquetadas con la misma actividad), ni transiciones en silencio (actividades que no están registrados en el registro de eventos). (2)

Teniendo en cuenta las ventajas que posee el algoritmo para el descubrimiento de los procesos de negocio ejecutados por el SUIN, así como su fácil empleo; se propone para ser utilizado en la solución a desarrollar.

1.2.2 Soluciones existentes

Las herramientas de Análisis de Procesos de Negocio (BPA⁷) son utilizadas para analizar los procesos de negocio, modelar procesos y diseñar flujos de trabajo de procesos. Son fundamentales en la optimización y racionalización de la organización, y se utilizan con frecuencia en combinación con la arquitectura empresarial (AE) y con las herramientas de gestión de procesos de negocios (BPM⁸). (8)

BONITA SOFT: Gestor de procesos de negocios BPM

BonitaSoft es el primer editor y líder de soluciones BPM. Uno de los objetivos de BonitaSoft es democratizar el BPM con una solución fácil e intuitiva que permita minimizar el costo de implantación, proporcionando soluciones de BPM flexibles y potentes para las organizaciones.

⁷ Por sus siglas en inglés *Business Process Analysis*.

⁸ Por sus siglas en inglés *Business Process Management*.

Las aplicaciones de Bonita comprenden toda la gama de proyectos de BPM, desde la migración de Sistemas de Información hacia una Arquitectura Orientada a Servicios (SOA⁹), hasta la automatización de los procesos de administración ERP y procesos de venta con interacciones humanas para los procesos de aprobación.

Características:

- Guarda, organiza y archiva todos sus procesos en el repositorio central de la organización.
- Gestiona datos de sus procesos bajo diversos formatos como Java Objects, XML o como documentos adjuntos.
- Exporta los diseños de procesos en pdf, jpeg, png, bmp, gif y svg.
- Guarda y administra versiones provisionales de su diseño mientras se modela un proceso.
- Simula la ejecución de procesos con parámetros, como el costo, duración, el consumo de recursos, calendario, entre otros, e identificar los candidatos para la optimización.
- Le permite seguir el proceso y recibir alertas en tiempo real.
- Permite implementar informes personalizados para obtener estadísticas de los procesos y los casos.

Bonita Open Solution cuenta con tres características que la convierte en un potente gestor de procesos de negocios; entre estas se encuentran: un innovador estudio de diseño de procesos, un potente motor de ejecución de procesos y una interfaz de usuario sencilla y de fácil utilización. (9)

Sin embargo, dentro de sus funcionalidades no están incluidas las predicciones de tiempo, además de que no contempla el formato en que son generadas las trazas del SUIN. Estas funcionalidades constituyen unas de las requeridas para dar solución al problema planteado, y no se encuentran dentro de las funciones que brinda el BonitaSoft. Por estas razones, se decide no utilizarlo como sistema para el análisis de trazas.

⁹ Por sus siglas en inglés: **S**ervices **O**riented **A**rchitecture.

Arís Process Performance Manager

ARISPPM¹⁰ es una herramienta que permite a las empresas monitorizar y analizar el rendimiento y la estructura de sus procesos de negocio. Proporciona una tecnología clave para evaluar los procesos de negocio y para identificar oportunidades de optimización. Posee un proceso de descubrimiento que ofrece visualización gráfica de los resultados de flujos de trabajo empresariales, hasta el nivel de transacción individual.

ARIS Performance Manager permite a las organizaciones lograr una velocidad excepcional en el proceso de análisis de la información y extraer e importar los datos necesarios a través de la gestión directa de datos en la memoria principal. (10)

Sin embargo, ARIS Process Performance Manager, no posee dentro de las funcionalidades los análisis operacionales, los que permiten realizar las predicciones de tiempo. Esta es una de las funcionalidades requeridas para la aplicación que se desea implementar, además de que se necesita contar con un buscador de procesos por usuarios. Todas estas funcionalidades deben estar unidas en una sola aplicación, y el ARISPPM no las contiene. Por tales razones, no se empleará para la solución.

MEGA Suite

La MEGA Suite es una solución completa que ayuda a los ejecutivos a entender como la gente y los procesos ofrecen valor. Las organizaciones utilizan BPA para representar y mejorar sus procesos de negocio, los vinculan a los objetivos estratégicos y la arquitectura de negocio, controlan y evalúan los que hayan sido ejecutados de acuerdo con las normativas y restricciones de riesgo. La MEGA Suite permite hacer todo esto unido en una solución, con el resultado de los recursos optimizados y la excelencia operativa. Se compone de cuatro categorías de herramientas: herramientas de modelado, herramientas de control, herramientas de transformación y herramientas de comunicación.

MEGA Process proporciona capacidades para modelar, documentar y mejorar los procesos de negocio y estructuras organizativas. MEGA Process ayuda a comparar y optimizar los procesos mediante pruebas, múltiples escenarios y sus posibles resultados. (8)

¹⁰ Por sus siglas en inglés *Arís Process Performance Manager*.

La MEGA Suite proporciona capacidades para modelar y propiciar mejoras a los procesos de negocio. Sin embargo, no realiza análisis operacionales ni de rendimiento. Estas funcionalidades son de gran importancia para cumplir el objetivo que se persigue con el análisis de las trazas del SUIN, por tales razones no se contempla para dar solución a la problemática planteada.

Intel(R) Trace Analyzer and Collector 8.0 for Windows and Linux

Con sus características únicas de minería de procesos, permite automáticamente recopilar y analizar rápidamente la actividad del sistema para visualizar lo que está sucediendo en su entorno.

El procesador Intel(R) es un colector de seguimiento, el rastreo que realiza la colección es basado en eventos de seguimiento en las aplicaciones. Puede analizar los datos de seguimiento recogidos por las zonas activas de rendimiento y los cuellos de botella. El producto es completamente seguro para subprocesos, y se integra con C/ C + +, Fortran y los procesos y subprocesos múltiples. Además se puede consultar la programación del sistema. El procesador Intel (R) Trace Analyzer proporciona una manera conveniente de controlar la aplicación y las actividades recogidas por el colector de seguimiento de Intel a través de pantallas gráficas. Usted puede ver el nivel de detalle deseado, identificar rápidamente los puntos críticos de rendimiento y cuellos de botella, y analizar sus causas. La herramienta está disponible para Linux y Microsoft Windows.

Características principales:

- *Avanzada interfaz gráfica de usuario:* amigable interfaz de usuario, con un nivel de alta escalabilidad, apoyo de archivo de seguimiento estructurado.
- *La agregación y filtrado:* vistas detalladas del comportamiento de tiempo de ejecución agrupados por funciones o procesos.
- *A prueba de errores de seguimiento:* mejora de la funcionalidad, permitiendo terminar aplicaciones con detección de punto muerto.
- *La función de comparación:* comparar dos archivos de seguimiento y/o dos regiones (en uno o dos archivos de seguimiento).
- *Contador de línea de tiempo:* analiza los datos recogidos a través de la lucha contra el Rendimiento de las aplicaciones de interfaz de programación (PAPI) y OS módulos o mediante el uso manual de Intel Trace Collector AP. (11)

Intel(R) Trace Analyzer and Collector constituye una herramienta que permite la realización de análisis a los sistemas una vez integrada a ellos. Esta herramienta no permite la captura de datos o información para sus análisis desde fuentes externas, por consiguiente no podría ser analizada la información histórica guardada en ficheros ni en la base de datos nacional del SUIN. Además es sólo integrable con C/C++ y Fortran, sin embargo el SUIN se encuentra desarrollado utilizando la plataforma .NET, utilizando como lenguaje de desarrollo C#.

Process Mining Workbench (ProM)

ProM es un marco genérico de código abierto para la aplicación de técnicas de minería de procesos en un entorno estándar. El marco de ProM recibe como registros de entrada archivos en formatos XES o formato MXML que contienen los datos oportunos de la ejecución de los procesos de los sistemas. En la actualidad, este marco tiene plugins para la aplicación de la minería de procesos en vista de realizar el análisis, monitoreo y conversión. Las técnicas de extracción del proceso con las que cuenta ofrecen una herramienta poderosa para evaluar la calidad y el rendimiento de los procesos de negocio.

ProM está disponible como distribución de archivos binarios para Windows, Mac OS X y Unix, y como el código fuente bajo los términos de la licencia CPL (hasta ProM 5.2) o la licencia GPL (a partir de ProM 6,0). Se requiere una instalación actual de Java Runtime Environment (JRE), versión 1.5/5.0 o superior (se recomienda la versión 5.0 para Windows, Linux y Mac OS X). (12)

El ProM constituye una poderosa herramienta para la aplicación de técnicas de la minería de procesos a los procesos de negocio ejecutados en un sistema. Sin embargo, para poder ser usada requiere la instalación de JRE, y con ello una estación de trabajo con cierto nivel de rendimiento debido a que durante su ejecución consume muchos recursos del sistema. El ProM tampoco permite ser integrado con aplicaciones web, de manera que a través de ella se le puedan proporcionar o extraer los datos. Por consiguiente no puede ser integrado con el SUIN. Además, este sistema web proporcionará sus funcionalidades desde cualquier estación de trabajo y estas serán de bajas prestaciones. Por todo lo anteriormente expuesto, no será usado el ProM como herramienta para el análisis de las trazas del SUIN.

1.2.3 Solución propuesta

A partir del estudio de las técnicas, algoritmos y soluciones existentes empleadas para el análisis de trazas; se han identificado una serie de agravantes que dificultan su uso en el entorno del negocio. Algunas de estas irregularidades son:

- El consumo excesivo de recursos del sistema, atentando contra la velocidad de las estaciones de trabajo con las que se cuenta. Teniendo en cuenta que estas son de bajas prestaciones.
- La dependencia de otros sistemas o herramientas para su funcionamiento.
- La realización del análisis de los datos de fuentes externas.
- La ausencia de funcionalidades que ayudarían a profundizar el análisis de las trazas, tales como: la búsqueda de procesos determinada por criterios, las predicciones de tiempo.

La idea esencial está en concentrar todas estas características y funcionalidades en una aplicación de fácil acceso y sin complicaciones o dependencias de otro software para su uso. Además, debe ser una solución que permita su integración con el módulo de administración del SUIN.

Por la importancia que se le concede al control y análisis de las trazas de los procesos que el SUIN genera, es necesario desarrollar un sistema informático que lo permita. A partir de dicha necesidad, se propone la realización del módulo de análisis de trazas de procesos del Sistema Único de Identificación Nacional de la República de Cuba.

1.3 Ambiente de desarrollo

A continuación se caracterizan la metodología, tecnologías y herramientas que permiten la realización del software propuesto, las cuales fueron definidas por las políticas del proyecto Identificación, Inmigración y Extranjería de la República de Cuba.

1.3.1 Metodología de desarrollo

En el desarrollo de cualquier sistema, es necesario conducir el proceso a través de una metodología. Esta define una guía sobre la que se apoyará el equipo de desarrollo, y de esta forma lograr una mayor organización durante las etapas de realización del software.

MSF¹¹ for CMMI¹²

Microsoft Solution Framework for CMMI es una metodología para el desarrollo de software centrada en el modelo de procesos y de equipo. Consta de cinco fases: Previsión, Planificación, Desarrollo, Estabilización e Implementación y es continuamente refinado por clientes, consultores y grupos de desarrollo de Microsoft.

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF for CMMI se centra en el modelo de procesos y de equipo, dejando en un segundo plano las elecciones tecnológicas.

MSF es un proceso alineado con CMMI que utiliza el mismo paradigma: iteraciones, roles, y reportes que proporcionan las métricas necesarias para valorar el estado del proyecto. Tiene documentos e informes más obligatorios que las versiones ágiles, y este proceso de desarrollo más formal, reduce el riesgo a la hora de desarrollar un software grande. Una de las ventajas de usar el proceso alineado con CMMI es la evaluación estándar por la cual se puede comparar la capacidad de desarrollar el software en otras organizaciones.

Características:

- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente. (13)

1.3.2 Lenguajes utilizados**CSharp (C#)**

Aunque para la plataforma .NET es posible programar en varios lenguajes, el C# es el lenguaje de propósito general diseñado por Microsoft para ser utilizado en ella.

Entre sus principales características se destacan:

¹¹ MSF: *Microsoft Solution Framework* (Framework de Soluciones de Microsoft).

¹² CMMI: *Capability Maturity Model Integration* (Integración de Modelos de Madurez de Capacidades).

- Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
 - El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera.
 - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código.
 - Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- Eficiente: en principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a como se hace en C++. Esto puede resultar vital para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grande. (14)

UML (Unified Modeling Language¹³)

El UML se define como un *"lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software ..."* (15). Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Su finalidad es describir modelos de sistemas (del mundo real y del mundo del software), basados en los conceptos de objetos. Los modelos se componen de otros modelos o artefactos, de diagramas y documentos que describen cosas. El UML especifica varios diagramas que permiten comprender el desarrollo del software, entre los que se encuentran los diagramas de clases, diagramas de despliegue, diagrama de componentes, etc.

¹³ Traducido al español: *Lenguaje Unificado de Construcción de Modelos*

El lenguaje UML no define un proceso estándar. Sus creadores admiten la importancia de contar con un lenguaje y un proceso muy sólidos para la construcción de modelos, ofrecen su recomendación porque la estandarización del proceso rebasaba entonces el ámbito de la definición del lenguaje. (16)

JavaScript

JavaScript es un lenguaje de script multiplataforma orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

El núcleo de JavaScript contiene un conjunto central de objetos, tales como Array (arreglos), Date (fechas) y Math (objetos matemáticos), además de un conjunto central de elementos del lenguaje tales como los operadores, estructuras de control y sentencias. El núcleo de JavaScript puede ser extendido para una variedad de propósitos complementándolo con objetos adicionales; por ejemplo:

- JavaScript del lado Cliente extiende el núcleo del lenguaje proporcionando objetos para el control del navegador (Navigator o cualquier Web browser) y su Modelo Objeto Documento [Document Object Model] (DOM). Por ejemplo, las extensiones del lado del cliente permiten a una aplicación ubicar elementos en un formulario HTML y responder a los eventos de usuario tales como los clic del mouse, entradas del formulario y navegación de páginas.
- JavaScript del lado Servidor extiende el núcleo del lenguaje proporcionando objetos relevantes para la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comunique con una base de datos relacional, proporcionar continuidad de la información desde una invocación de la aplicación a otra o efectuar la manipulación de archivos en un servidor. (17)

1.3.3 Plataforma de desarrollo

.Net Framework

Es la plataforma de desarrollo de código administrado de Microsoft. Está formado por una serie de herramientas y librerías con las que se pueden crear todo tipo de aplicaciones, desde las tradicionales

aplicaciones de escritorio hasta aplicaciones para XBOX (XNA) pasando por desarrollo web (ASP.NET), desarrollo para móviles (*Compact framework*), aplicaciones de servidor (WPF¹⁴, WCF¹⁵). (18)

Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 es un conjunto de herramientas que simplifica el desarrollo de aplicaciones para desarrolladores individuales o equipos, haciendo posible la creación de aplicaciones escalables y de alta calidad. Proporciona nuevos beneficios significativos, como la capacidad de optimizar su entorno de desarrollo con compatibilidad para varios monitores y la capacidad de soportar varias versiones de *.NET Framework* con una sola herramienta.

Presenta novedosas herramientas de pruebas que automatizan la realización de pruebas de la interfaz de usuario en aplicaciones basadas en web: pruebas manuales, pruebas de rendimiento de web, pruebas de carga, cobertura de código, generación de datos de pruebas.

Incluye una versión renovada del servidor *Team Foundation Server*, además de elementos nuevos como *LabManagement* que permite la creación y gestión de entornos virtuales. Brinda mejoras en herramientas como *Expression* y *SharePoint* presentes en versiones anteriores pero que ahora son mucho más funcionales. (18)

Framework .Net 4.0

Es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML. Contiene dos componentes principales: *Common Language Runtime* y la biblioteca de clases de *.NET Framework*. *Common Language Runtime* es el fundamento de *.NET Framework*. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la comunicación remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que promueven su seguridad y solidez. La biblioteca de clases, el otro componente principal de *.NET Framework*, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de

¹⁴ Del inglés *Windows Presentation Foundation*. Tecnología que permite la interacción de usuarios y aplicaciones.

¹⁵ Del inglés *Windows Communication Foundation*. Es un conjunto de tecnologías *.NET* para construir y poner en marcha sistemas conectados en la plataforma de Microsoft.

comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

Este *framework* brinda nuevas mejoras y características que lo hacen diferente a sus versiones anteriores, aunque posee la capacidad de funcionar en paralelo con ellas:

- ✓ Mejoras en *Common Language Runtime* (CLR), ASP.Net e innovaciones en los lenguajes Visual Basic y CSharp.
- ✓ *Entity Framework*, donde se permite a los desarrolladores programar con bases de datos relacionales usando objetos .NET y *Language Integrated Query* (LINQ).
- ✓ Servicios de datos de *Windows Communication Foundation* (WCF), componente que permite crear servicios y aplicaciones que usen protocolo de datos abierto para exponer y usar datos a través de la Web. (19)

Language Integrated Query (LINQ)

LINQ incluye patrones estándar y de fácil aprendizaje para consultar y actualizar datos, y su tecnología se puede extender para utilizar potencialmente cualquier tipo de almacén de datos. Visual Studio incluye ensamblados de proveedores para LINQ que habilitan su uso con colecciones de .NET Framework, bases de datos SQLServer, conjuntos de datos de ADO.NET y documentos XML (20).

Se traduce como consultas integradas en el lenguaje, y refleja fielmente el objetivo principal de las extensiones recién incorporadas a los lenguajes antes mencionados: abrir ante los desarrolladores la posibilidad de expresar de una manera uniforme e integrada consultas contra las más diversas fuentes de datos que estos utilizan en sus aplicaciones (colecciones de objetos en memoria, documentos XML o bases de datos relacionales, entre otras), mediante las expresiones de consulta que constituyen el principal reflejo en los lenguajes de la tecnología LINQ.

Las ventajas más importantes que aporta LINQ al mundo de la programación son:

- Permite reducir a una mínima expresión el conocido problema del desajuste de impedancias en la programación actual, provocado por las diferencias entre los modelos que proponen los lenguajes de propósito general y los que ofrecen los lenguajes de consulta específicos a fuentes de datos de diversa naturaleza, como SQL.

- Permite elevar el nivel de abstracción y claridad de la programación y en gran medida indicar únicamente qué se desea obtener, dejando a un motor de evaluación de expresiones los detalles sobre cómo lograr ese objetivo; seguridad mejorada, gracias a la verificación de sintaxis y de tipos por parte de los compiladores. (21).

1.3.4 Control de versiones

Una de las más importantes cuestiones a las que debe mantenerse atento el equipo de desarrollo de cualquier sistema informático es al control de versiones. Para facilitar el trabajo del equipo en este sentido, existen tecnologías y prácticas a seguir para supervisar la planificación del proyecto y mantener informado al equipo de desarrollo sobre los últimos cambios realizados en los ficheros del proyecto, en particular en el código fuente, la documentación o las páginas web.

Team Foundation Server 2010

Microsoft Visual Studio Team Foundation Server 2010 es la plataforma de colaboración central en la solución de administración del ciclo de vida de las aplicaciones de Microsoft, que automatiza el proceso de entrega de software y permite a las organizaciones administrar proyectos de desarrollo de software de forma eficiente durante su ciclo de vida. *Team Foundation Server 2010* permite al equipo ser más ágil, colaborar de un modo más eficaz y ofrecer software de más calidad, así como adquirir y compartir conocimiento institucional.

Los datos y artefactos del proyecto de seguimiento de elementos de trabajo, control de código fuente, compilaciones y herramientas de pruebas se acumulan en un almacén de datos, y eficaces informes que proporcionan tendencias históricas, seguimiento completo y visibilidad en tiempo real de la calidad y el progreso respecto al propósito empresarial. (22)

1.3.5 Manejo de datos

Todo sistema en funcionamiento maneja una cierta cantidad de datos que en ocasiones es voluminosa y necesita ser almacenada en contenedores que permitan recuperarla, actualizarla y eliminarla con el menor costo y la mayor rapidez posible.

Un SGBD es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de

manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando la seguridad e integridad de los mismos. Un SGBD permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. (23)

Oracle Database 11g

Oracle Database 11g proporciona nuevas e innovadoras funcionalidades que garantizan alto rendimiento, alta escalabilidad, fiabilidad y seguridad mediante el uso de plataformas Grid, asegurando altos niveles de calidad de servicio e incrementos de la flexibilidad de negocio, reduciendo además los costes de explotación. Con Oracle Database 11g los clientes pueden resolver las problemáticas de negocio más exigentes en todas las áreas, incluyendo aplicaciones transaccionales, de inteligencia de negocio y de gestión de contenidos.

Es el primer gestor de base de datos del mundo en incluir funcionalidades que permiten hacer pruebas de cambios en aplicaciones, simulando las cargas reales generadas por los usuarios en los entornos de producción. *Real Application Testing* permite reducir de manera drástica los tiempos, riesgos y costes derivados de la implantación de cambios, asegurando que las aplicaciones se comportarán de manera adecuada y predecible tras las modificaciones.

Incorpora un nuevo compilador Java que facilita un alto rendimiento en la ejecución de procedimientos almacenados sin necesidad de utilizar un compilador de terceros. Introduce un conjunto de funcionalidades en la línea de la auto-administración de la base de datos. Estas funcionalidades incluyen la generación automática de sentencias SQL, asistentes para reducir las tareas de particionamiento y ciclo de vida de los datos que proporciona una interfaz para visualizar incidencias en la base de datos. (21)

1.3.6 Herramienta de modelado

Altova UModel 2009

Altova UModel puede generar automáticamente diagramas de secuencia múltiple para las operaciones de ingeniería inversa en las clases de traza de la ejecución de aplicaciones. Modifica su código o modelos UML y añade un esquema único para el modelado de esquemas XML¹⁶ en UML. Permite generar la documentación en HTML, Word y RTF, así como hipervínculos a los diagramas UML para diagramas de

¹⁶ Por sus siglas en inglés: *eXtensible Markup Language* (lenguaje de marcas extensible)

otros archivos externos, o sitios Web. Cuando se integra con Visual Studio y Eclipse, sincroniza automáticamente diagramas UML, debido a que el código fuente se edita en estos IDEs.

Esta herramienta CASE¹⁷ combina una intuitiva interfaz visual con funciones de usabilidad superiores, que favorecen a los usuarios con las más completas ventajas en el modelado de software. Las características de Altova UModel 2009 para el desarrollo de *software* basado en las capacidades de modelado avanzado son:

- Soporte para los 14 tipos de diagramas UML.
- Generación de código fuente en lenguajes Java, C#, y VB.NET.
- Creación de diagramas de secuencia desde el código fuente de la ingeniería inversa.
- Generación de documentación personalizable de proyecto.
- Integración con sistemas de control de versiones.
- Estrecha integración con *Visual Studio* y *Eclipse* (24)

Conclusiones parciales

En el capítulo fueron estudiados varios conceptos referentes a los procesos de negocio. Se definió que el enfoque a procesos constituye una herramienta fundamental para el análisis de los procesos que son ejecutados en el SUIN. Fueron analizadas las técnicas y algoritmos de minería de procesos, así como, las soluciones existentes referentes al análisis de traza de procesos. Este análisis permitió identificar las agravantes que estas presentaban para dar solución a la problemática. Sin embargo, al ser evaluadas las ventajas del algoritmo alfa y su aporte para la solución, se determinó que sería esencial en la propuesta de solución. La selección de la metodología, lenguajes, plataforma y demás herramientas seleccionadas para el desarrollo de la solución permitirán diseñar prototipos de interfaz de usuarios agradable y fácil de interactuar. Así como una estructuración del módulo acorde con lo estipulado, lo que posibilitará una integración factible con el módulo de administración del SUIN.

¹⁷ *Ingeniería de Software Asistida por Computadora* (CASE) destinada a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero.

Capítulo 2 Solución propuesta

Introducción

En el presente capítulo se hace una descripción detallada de la propuesta de solución mediante el modelado del negocio. Además se realiza el levantamiento de requisitos que contiene la definición de los requisitos funcionales y no funcionales del módulo a desarrollar, conjuntamente con los roles que intervienen. También se realiza el modelo de diseño del módulo que se va a implementar, se definen y se especifican el conjunto de clases a utilizar. Se describen la arquitectura y los patrones a utilizar para el desarrollo del módulo. En este capítulo se presenta además el modelo de datos, así como la descripción del modelo de dominio realizado para el desarrollo de la investigación y la solución que se propone para dar respuesta al problema planteado.

2.1 Modelo de dominio

En la actualidad, no se lleva a cabo el análisis de los procesos de negocio ejecutados por el SUIN, por tales motivos no existe un negocio definido. Por consiguiente, es necesario diseñar un modelo de dominio para realizar el desarrollo de la propuesta de solución.

El modelo de dominio es un modelo de objetos del dominio que incorpora comportamientos y datos (25). Este modelo es la representación visual de las clases conceptuales u objetos del mundo real que reportan interés para la investigación o proyecto. El modelo de dominio puede ser empleado en casos donde el negocio no comprende actividades demasiado cambiantes. La determinación de realizar un modelo de dominio se toma a partir de no tener una claridad suficiente en los procesos como para modelar un negocio, no tener fronteras claramente determinadas para este o no existir una claridad de quiénes son los que inician el negocio y quienes se benefician con este. A continuación en la *Fig. 1* se muestra el modelo de dominio definido para el módulo a desarrollar.

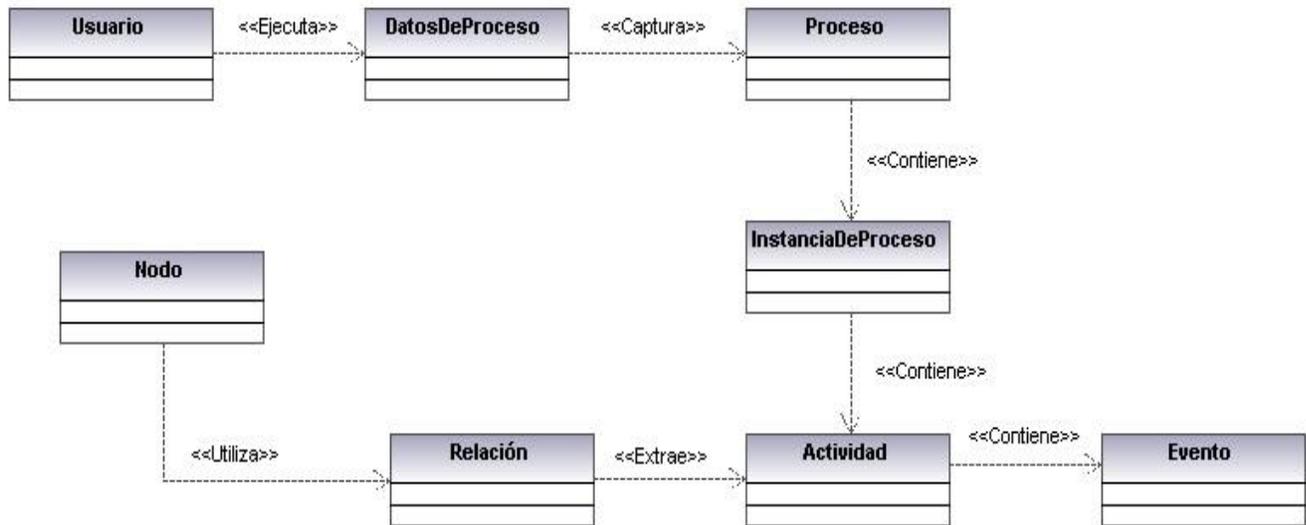


Fig. 1 Modelo de dominio.

2.1.1 Descripción del modelo de dominio

En el presente modelo de dominio se expone la captura de los datos referente a los procesos almacenados en las fuentes de datos correspondientes. Estos datos son transformados en las entidades **Proceso** que constituyen el o los procesos ejecutados por el **Usuario**. Dichas entidades contienen las **Instancias de Proceso**, las que a su vez contienen las **Actividades** ejecutadas en dichas instancias. En cada una de las actividades están contenidos los **Eventos**, e informaciones tales como el ejecutor, fecha y hora de ejecución, la localización u oficina desde donde fue ejecutada, etc. Estas informaciones son las utilizadas para realizar los análisis operacionales y de rendimiento, y con ello mostrar los tiempos de gestación de las instancias, los tiempos de servicio del proceso y las predicciones de tiempo de los procesos que se encuentran aún en ejecución. Luego son identificadas y extraída la **Relación** existente entre ellas, las que conformarían el **Nodo** para llevar a cabo el graficado y modelado del proceso.

Módulo de análisis de trazas de procesos

Para la realización del análisis de los procesos ejecutados por cualquier sistema, es recomendable el empleo de técnicas de minería de procesos con el fin de hacer más eficaz y factible el análisis.

La propuesta de solución permite definir la fuente de datos a utilizar para llevar a cabo los análisis, ya sea desde la información almacenada en la base datos como de la contenida en ficheros de trazas

estandarizados mediante el formato MXML. Cuenta con un buscador de procesos que está enfocado a la obtención de los procesos dado un criterio de búsqueda que está dado por un usuario, tipo de proceso, fecha de inicio, fecha de fin y/o la oficina desde donde fue ejecutado. Dentro de sus funcionalidades se encuentra el descubrimiento del proceso seleccionado luego de realizada la búsqueda. Este descubrimiento se mostrará a través de la imagen del modelo del proceso. Para ello se utilizará la técnica de descubrimiento llamada algoritmo alfa, la cual devuelve como resultado una representación en redes de Petri de lo que ocurrió realmente en el proceso con una perspectiva de control de flujo.

La solución incluye la realización de análisis operacionales y de rendimiento, los que brindarán informaciones relacionadas con los tiempos de ejecución de las actividades e instancias de los procesos. Siendo estos análisis las bases para realizar las predicciones de tiempo e identificar los cuellos de botella en los procesos. La propuesta le permite al usuario visualizar el tiempo promedio y el tiempo de ejecución que lleva el proceso que se encuentra ejecutando. Además, resalta en el modelo del negocio, la o las actividades que presentan demoras respecto al tiempo estándar de ejecución de ese tipo de actividad. El módulo permite también la navegación dentro del proceso, facilitando de esta manera ver características relacionadas a los procesos, casos o actividades seleccionadas.

2.2 Descripción de roles

Roles	Descripción
Auditor de trazas	<ul style="list-style-type: none"> Realiza la auditoria de las trazas generadas por los procesos ejecutados en el sistema. Realiza acciones tales como: la búsqueda de los procesos según el usuario, tipo de procesos, fecha de inicio, fecha de fin u oficina a auditar y realiza el descubrimiento y análisis de los procesos que lo requieran.
Técnico de auditoría de trazas	<ul style="list-style-type: none"> Lleva a cabo la generación de los ficheros *.mxml con las trazas de los procesos almacenados en la base de datos dado un período de tiempo según sea la necesidad.
Funcionarios	<ul style="list-style-type: none"> Lleva a cabo la ejecución de los procesos en el sistema.

Tabla 1 Descripción de roles.

2.3 Catálogo de requisitos

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que requiere el cliente, analizando necesidades, conformando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación. (26)

2.3.1 Especificación de los requisitos funcionales

Los requerimientos funcionales de un sistema describen la funcionalidad o los servicios que se espera que éste provea. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y no alteran la funcionalidad del sistema, esto quiere decir que los requisitos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (26)

La obtención de los requisitos funcionales constituye la actividad más importante dentro de la modelación del negocio. Teniendo en cuenta las disposiciones para las que se creará el módulo fueron definidos los requisitos funcionales que se muestran a continuación en la *Tabla 2*.

No.	Descripción
RF 1	Extraer datos de los sistemas fuente.
RF 2	Transformar datos.
RF 3	Buscar procesos según criterio de búsqueda.
RF 4	Mostrar datos de los procesos, instancias de procesos y actividades.
RF 5	Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.
RF 6	Analizar el rendimiento del proceso.
RF 7	Mostrar resultados del análisis de rendimiento.
RF 8	Mostrar cuellos de botella.
RF 9	Analizar operacionalmente el proceso.
RF 10	Mostrar predicciones de tiempo.
RF 11	Generar fichero *.mxml.

Tabla 2 Requisitos funcionales del módulo.

2.3.2 Descripción de los requisitos

Una vez que los requisitos funcionales han sido identificados y se han definido los roles, entonces se procede a la descripción de los mismos. Esta descripción facilita la implementación del sistema, ya que en ella son definidas las entidades que serán utilizadas y las condiciones que debe cumplir la aplicación inicialmente. Explica la secuencia de actividades entre el rol y el sistema para llevar a cabo las funcionalidades descritas, así como el resultado que se espera después de realizada cada una de las operaciones. Todos los requisitos funcionales definidos para el módulo en cuestión se encuentran descritos en el *Anexo 1*.

Para ejemplificar el levantamiento de requisitos funcionales del módulo de análisis de trazas de procesos, se realiza la descripción del *RF3¹⁸ Buscar procesos según criterio de búsqueda*.

RF3 Buscar procesos según criterio de búsqueda.

1.1. Permitir que el usuario introduzca los datos para la búsqueda de los procesos:

- a) Usuario
- b) Fecha inicio
- c) Fecha fin
- d) Oficina
- e) Tipo de proceso

1.1.1. Buscar procesos correspondientes: si selecciona Buscar: ver 3.3

1.1.2. Mostrar la página sin datos: si selecciona Cancelar.

1.2. Validar que los campos Fecha Inicio y Fecha Fin no se encuentren vacíos o estén correctamente.

1.3. Mostar lista de procesos coincidentes con los criterios de búsqueda y los datos generales del usuario.

1.3.1. Mostrar en la barra de navegación el texto “Sin resultados”: si no existen resultados coincidentes.

1.4. Mostrar lista de las instancias del proceso seleccionado.

1.5. Mostrar lista de las actividades de la instancia de proceso seleccionada.

1.6. Permitir al usuario retroceder a los resultados iniciales de la búsqueda y a las instancias de procesos del proceso seleccionado.

¹⁸ *Requisito Funcional.*

1.6.1. Mostrar resultados iniciales de la búsqueda: si selecciona atrás.

1.6.2. Mostrar lista de instancias del proceso: si selecciona el nombre del proceso en la barra de navegación.

1.6.3. Mostrar lista de instancias del proceso: si selecciona el Id de la instancia en la barra de navegación.

1.7. Mostrar página para seleccionar fuente de datos: si selecciona Cancelar.

Para un mejor entendimiento, en la *Tabla 3* se presenta la especificación del *RF 3 Buscar procesos según criterio de búsqueda*. En esta especificación se amplía el escenario de condiciones y validaciones que deben tenerse en cuenta para que tenga lugar el proceso al que corresponde el requerimiento.

RF 3 Buscar procesos según criterio de búsqueda.

Propósito	Mostrar los procesos que se han ejecutado o se encuentran en ejecución; tanto los almacenados en ficheros <i>.xml</i> como los almacenados en la base de datos nacional, teniendo en cuenta un criterio de búsqueda.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> • El auditor de trazas debe estar autenticado. • Debe haberse seleccionado la fuente de datos. • Deben haberse seleccionado la fecha inicio y la fecha fin. • Deben haberse seleccionado la oficina, el usuario o el tipo de proceso. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
Descripción	<p>3.1. Permitir que el usuario introduzca los datos para la búsqueda de los procesos:</p> <ul style="list-style-type: none"> f) Usuario g) Fecha inicio h) Fecha fin i) Oficina j) Tipo de proceso <p>3.1.1. Buscar procesos correspondientes: Si selecciona Buscar: ver 3.3</p> <p>3.2. Validar que los campos Fecha Inicio y Fecha Fin no se encuentren vacíos o estén</p>	

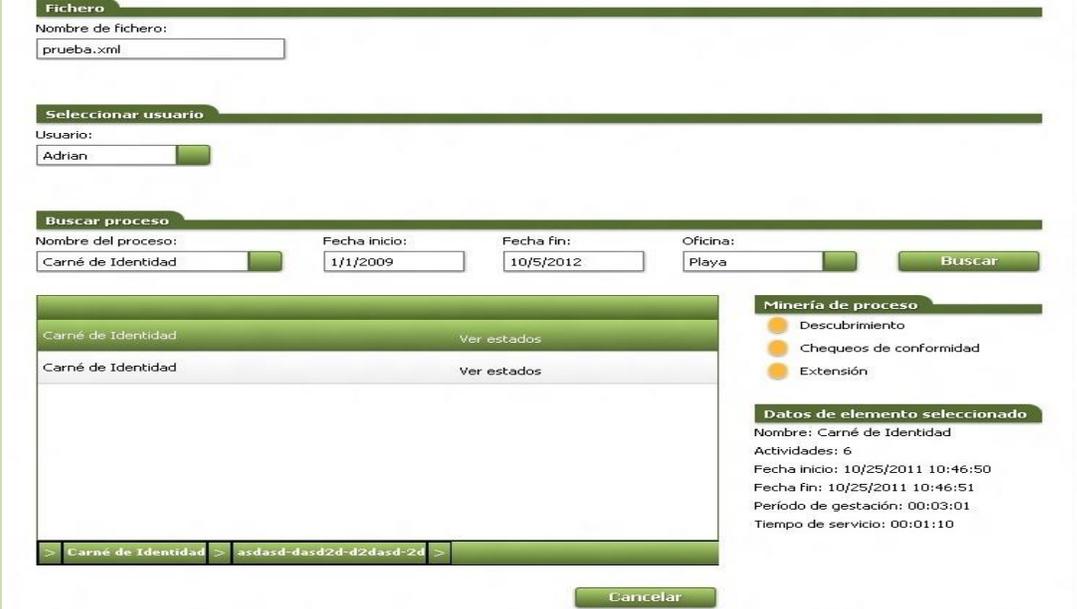
	<p>correctamente.</p> <p>3.3. Mostrar lista de procesos coincidentes con los criterios de búsqueda y los datos generales del usuario.</p> <p>3.3.1. Mostrar en la barra de navegación el texto: “Sin resultados”: si no existen resultados coincidentes.</p> <p>3.4. Mostrar lista de las instancias del proceso seleccionado.</p> <p>3.5. Mostrar lista de las actividades de la instancia de proceso.</p> <p>3.6. Permitir al usuario retroceder a los resultados iniciales de la búsqueda y a las instancias de procesos del proceso seleccionado.</p> <p>3.6.1. Mostrar resultados iniciales de la búsqueda: si selecciona atrás.</p> <p>3.6.2. Mostrar lista de instancias del proceso: si selecciona el nombre del proceso en la barra de navegación.</p> <p>3.6.3. Mostrar lista de instancias del proceso: si selecciona el Id de la instancia en la barra de navegación.</p> <p>3.7. Mostrar página para seleccionar fuente de datos: si selecciona Cancelar.</p>
<p>Validaciones</p>	<p>1. Los campos Fecha Inicio y Fecha Fin no deben estar vacíos.</p> <p>2. La fecha de inicio no puede ser mayor que la fecha de fin.</p>
<p>Prototipo</p>	

Tabla 3 Especificación del RF 3 Buscar procesos según criterio de búsqueda.

2.4 Especificación de los requisitos no funcionales

Según la IEEE los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. A continuación se muestran los requerimientos no funcionales definidos para el Módulo de Análisis de Trazas de Procesos del SUIN.

Usabilidad

RnF1. El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

- ✓ Conocimientos básicos relativos al uso de una computadora.
- ✓ Conocimientos básicos del sistema operativo Windows.
- ✓ Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

RnF2. El sistema será distribuido en idioma español.

RnF3. El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación.

- ✓ Menús laterales y desplegados que permitan el acceso rápido a la información.

Seguridad

RnF4. El sistema contará con diferentes niveles de acceso:

- ✓ De los usuarios: se realizará acorde con los roles autorizados para el uso de cada funcionalidad tanto a nivel de funciones de las aplicaciones como de información de las bases de datos.
- ✓ De las oficinas: se registrarán con los roles que le corresponda acorde a las funciones que le competan.
- ✓ De las estaciones de trabajo: se registrarán las estaciones de trabajo de forma tal que cada una accederá a las funcionalidades que le corresponda según su ubicación y función en la organización.

RnF5. El sistema registrará todas las acciones que se realizan.

- ✓ Se llevará el registro del tiempo de actividad y del lugar de acceso de cada usuario y estación de trabajo para cada acción.

- ✓ Se podrán realizar auditorías para la comprobación de las actividades realizadas en el sistema.

RnF6. La conexión a los servidores de bases de datos y de aplicaciones es requerida para el correcto funcionamiento.

RnF7. El sistema garantizará la mayor exactitud en los datos.

- ✓ Deberá tenerse en cuenta la calidad de la fuente inicial de los datos para la migración de los mismos a las nuevas bases de datos.
- ✓ Deberán realizarse validaciones y comprobaciones automáticas en todos los casos posibles para garantizar la consistencia de los datos.

Interfaz

Interfaces de usuario

RnF8. Interfaz accesible e intuitiva, el manejo de las funcionalidades debe ser lo más intuitivo posible de manera que sean muy claras las acciones que se puedan realizar y la manera de hacerlas.

RnF10. Interfaz consistente con las pautas de diseño definidas para los sistemas del proyecto.

RnF11. Consistencia de la aplicación entre los distintos navegadores. Debe visualizarse y manejarse de la misma forma en los navegadores más comunes agrupados a continuación según su motor/intérprete de HTML, CSS y JavaScript:

- ✓ Trident (Internet Explorer a partir de la versión 7.0).
- ✓ WebKit (Google Chrome a partir de la versión 10.0).
- ✓ Gecko (Mozilla Firefox a partir de la versión 4.0).

2.5 Arquitectura

La arquitectura del módulo cuenta con cuatro capas lógicas que dan un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir al máximo el acoplamiento y aumentar la reutilización entre las mismas.

La distribución de las capas permite que se realicen grandes cambios sin tener que realizar cambios en las demás capas. Una vez que estas estén bien definidas la comunicación entre ellas se realizará solo a nivel de interfaces que permiten trabajar de manera transparente a las instancias reales. En la *Fig. 2* se muestra la arquitectura definida para el módulo de análisis de procesos del SUIN.

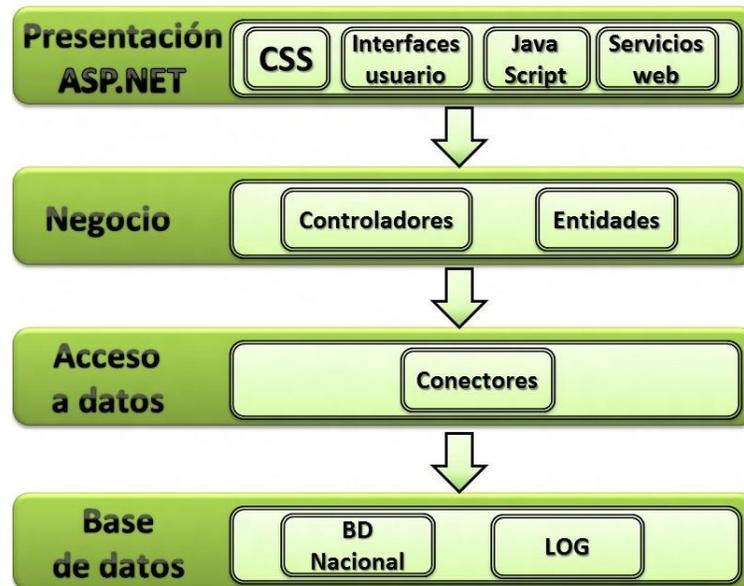


Fig. 2 Arquitectura del módulo de análisis de trazas de procesos.

Presentación

Está compuesta por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros JavaScript, CSS, Servicios Web, interfaces de usuario, etc. Esta capa se encuentra representada por el proyecto web de la aplicación y tiene interacción directa con la capa de negocio.

UIToolsBox

El UIToolsBox es una librería de componentes de interfaz de usuario que rige el modo en que se visualizan los controles en el navegador basado en las siguientes características:

- Define los controles básicos basándose en los controles de ASP.NET para mantener el comportamiento de las aplicaciones ASP.NET.
- Rige el comportamiento de las interfaces de usuario a partir de una capa basada en JQuery que le da un comportamiento más dinámico a los controles utilizados.
- Garantiza peticiones asíncronas al servidor a través de servicios Web que son consumidos desde JavaScript.

Negocio

En esta capa se recogen todos los controladores necesarios para darle solución a los requerimientos de negocio. Los controladores se encuentran definidos según el contexto en el que se desenvuelven y tienen la responsabilidad de manejar todas las operaciones sobre una entidad de negocio en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. Por cada entidad de negocio se crea un controlador y una interfaz que debe ser implementada por el acceso a datos que le dará soporte.

Acceso a Datos

La capa de acceso a datos está directamente relacionada con los controladores definidos en el negocio. Para establecer esta relación se hace uso de la interfaces de conectores y de la fábrica de conectores que define la capa de negocio. De esta manera es posible realizar cambios en esta capa sin que se vean afectadas las demás. Su principal función es realizar una implementación de las interfaces definidas en la capa de negocio y al mismo tiempo trabajar directamente con las fuentes de datos establecidas.

Base de Datos

Por la importancia que posee el desarrollo de procedimientos y vistas dentro del negocio, se ha decidido separar la base de datos en una nueva capa donde se concentra una pequeña parte de la lógica de las funcionalidades dentro de la aplicación. La única capa encargada de interactuar con esta es la de Acceso a Datos.

2.6 Patrones de diseño

Un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico y en medio de fuerzas que pueden tener un impacto en la manera en que se aplica y utiliza el patrón. La finalidad de cada patrón de diseño es proporcionar una descripción que le permita al diseñador determinar si el patrón es aplicable al trabajo actual, si se puede reutilizar (por ende, ahorrar tiempo del diseño) y si puede servir como guía para desarrollar un patrón similar, pero diferente en cuanto a la funcionalidad o estructura. (26)

Los patrones de diseño pueden incrementar o disminuir la capacidad de comprensión de un diseño o de una implementación. Además de esto, es necesario tener en cuenta que hacer uso de patrones significa en muchos casos disminución en el rendimiento de la aplicación, por ello es necesario lograr un equilibrio

entre la flexibilidad y el rendimiento dentro del sistema. Esto implica que no es razonable utilizar patrones hasta que no se tenga totalmente claro el dominio y el problema a tratar.

Para el desarrollo del módulo en cuestión se han tenido en cuenta un conjunto de patrones que permiten darle flexibilidad, además se identificaron un conjunto de patrones específicos para el desarrollo que brindan claridad y fortaleza a los diseños.

Encapsulamiento: propone esconder algunos componentes, permitiendo sólo accesos estilizados al objeto. Se hace uso de este patrón en casi todas las clases que componen al sistema permitiendo que estas solo posean como elementos públicos aquellos que son exclusivamente necesarios.

```
public class MiningProcessController
{
    public ConfigConnector ConfigConnector { get; set; }
    ...
}
```

Subclase: propone heredar miembros por defecto de una superclase, seleccionando la implementación correcta a través de resoluciones sobre qué implementación debe ser ejecutada. Se puede encontrar este patrón con más fuerza en las entidades de negocio que por su conceptualización las funciones y la información que almacenan pueden estar diferenciadas en cierta medida. A continuación, en la Fig. 3 se muestra un ejemplo del uso del patrón subclase en el módulo.

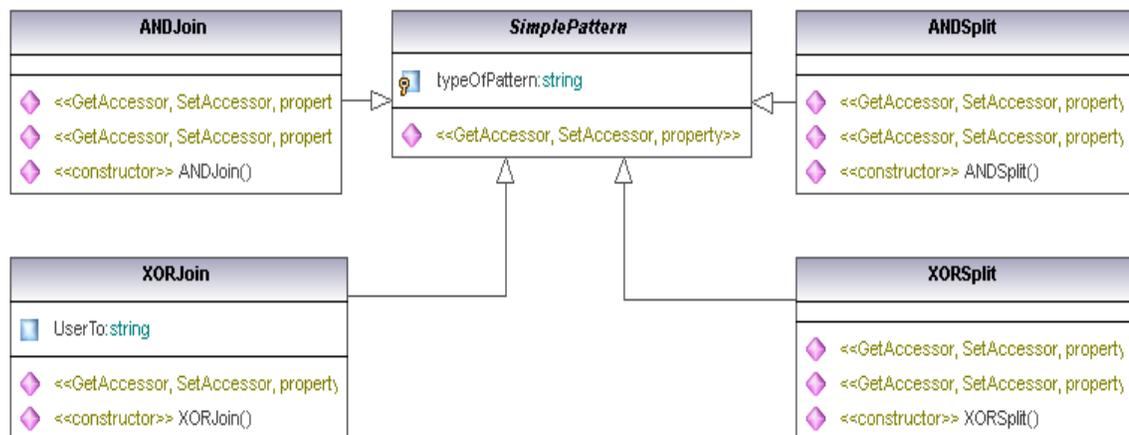


Fig. 3 Ejemplo del patrón de diseño subclase.

Excepciones: propone introducir estructuras de lenguaje para arrojar e interceptar excepciones. Se identificaron los diferentes tipos de errores a tratar dentro del sistema creando clases que permitan identificar cada tipo de error en el momento de ejecución.

Singleton: se asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia. Dicho patrón se utiliza en la clase **HefisFactory**, la cual está definida globalmente para todo el proyecto.

Ejemplo:

```
public class HefisFactory
{
    private static HefisFactory _instance { get; set; }

    public static HefisFactory Instance
    {
        get { return _instance ?? (_instance = new HefisFactory()); }
    }

    public IHefisServicesConnector HefisServices
    {
        get { return new HefisServicesConnector(); }
    }
}
```

Fábrica: provee de una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases. Su uso se encuentra centrado en la creación de los conectores correspondientes al acceso a datos que se esté utilizando, así como en la obtención de los servicios a utilizar. Un ejemplo de este patrón lo constituye la clase **HefisFactory**, la cual tiene como principal objetivo crear los conectores correspondientes a un componente de acceso de datos específico.

GRASP¹⁹: el uso de este conjunto de patrones está totalmente unido a cada componente desarrollado en el módulo, donde cada uno de ellos posee sólo las funcionalidades acorde a las particularidades que lo caracterizan. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Se aplican durante la elaboración de los diagramas de interacción, al asignar las responsabilidades a los objetos y al diseñar la colaboración entre ellos. Algunos de los patrones que describe GRASP son: *Experto, Creador, Alta Cohesión, Bajo*

¹⁹ Acrónimo que significa **General Responsibility Assignment Software Patterns** (patrones generales de software para asignar responsabilidades).

Acoplamiento y el patrón *Controlador*. (16) A continuación se describen ejemplos del uso de algunos de estos patrones en el desarrollo de la aplicación.

El patrón *Creador* guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. A continuación se muestra un ejemplo del empleo del patrón *Creador* en el módulo.

```
public class ProcessMiningService : System.Web.Services.WebService
{
    [WebMethod(EnableSession = true)]
    public List<Process> SearchProcessTest(string origin, string address, string user, string
nombreProceso, string startdate, string enddate, string office)
    {
        var sd = Convert.ToDateTime(startdate);
        var ed = Convert.ToDateTime(enddate);
        user = Convert.ToString(Session["SelectedUserId"]);
        var controller = new MiningProcessController (SessionHelper.ConfigConnector);
        return controller.SearchProcess(user, nombreProceso, sd, ed, office);
    }
.... }
```

En el ejemplo se muestra un fragmento de código de la clase ***ProcessMiningService*** donde para cada uno de los métodos se crean instancias de otras clases, en este caso de la clase controladora ***MiningProcessController***.

Un *Controlador* es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. Un ejemplo de la aplicación del patrón controlador se muestra en la Fig. 4. En la misma se presenta la clase ***DiscoveryController*** que es la encargada de manejar los eventos sistémicos que se desencadenan.

```

DiscoveryController
(from Controller::MiningControllers)

<<constructor>> DiscoveryController(in configConnector:ConfigConnector)
SeparateInstances(in listaProcess:List<T1->Process>):List<T1->List<T1->AuditTrailEntry>
LlenarLista(in listaEntrante:List<T1->AuditTrailEntry>):List<T1->AuditTrailEntry>
ExistInListOfActivities(in objeto:AuditTrailEntry, in lista:List<T1->AuditTrailEntry>):bool
ExtractActivities(in listaSecuencias:List<T1->List<T1->AuditTrailEntry>):List<T1->AuditTrailEntry>
CreateMatrixOfRelations(in listaSecuencias:List<T1->List<T1->AuditTrailEntry>):TypeOfRelation[*][*]
GetRelation(in matriz:List<T1->List<T1->AuditTrailEntry>):Dictionary<T1->string,T2->TypeOfRelation>
GetDirectSuccessions(in matriz:List<T1->List<T1->AuditTrailEntry>):Dictionary<T1->string,T2->DiscoverRelation>
ExtractInitialActivities(in listaSecuencias:List<T1->List<T1->AuditTrailEntry>):List<T1->AuditTrailEntry>
ExtractFinalActivities(in listaSecuencias:List<T1->List<T1->AuditTrailEntry>):List<T1->AuditTrailEntry>
ListarActividades_time(in listaAuditTrailEntry:List<T1->AuditTrailEntry>):List<T1->ActivityData>
ListarActividadesString(in listaAuditTrailEntry:List<T1->AuditTrailEntry>):List<T1->string>
    
```

Fig. 4 Ejemplo del uso del patrón Controlador.

2.7 Especificación de clases

Con el objetivo de dar solución a las funcionalidades que debe cumplir el módulo propuesto, se realiza un análisis profundo de estas. Quedan definidas así, las clases que serán utilizadas en el cumplimiento de cada requisito funcional, las cuales se muestran a continuación en el diagrama de clases del diseño.

2.7.1 Diagrama de clases del diseño

El diagrama de clases del diseño representa la interacción de las entidades, los conectores a la base de datos y muestra cómo se comporta el flujo de informaciones en el sistema. A continuación, en la Fig. 5 se muestra un fragmento del diagrama de clases del diseño del módulo de análisis de trazas del SUIN. En el mismo se observan las relaciones existentes entre algunas de las clases del módulo. El diagrama de clases extendido se muestra en el Anexo 5.

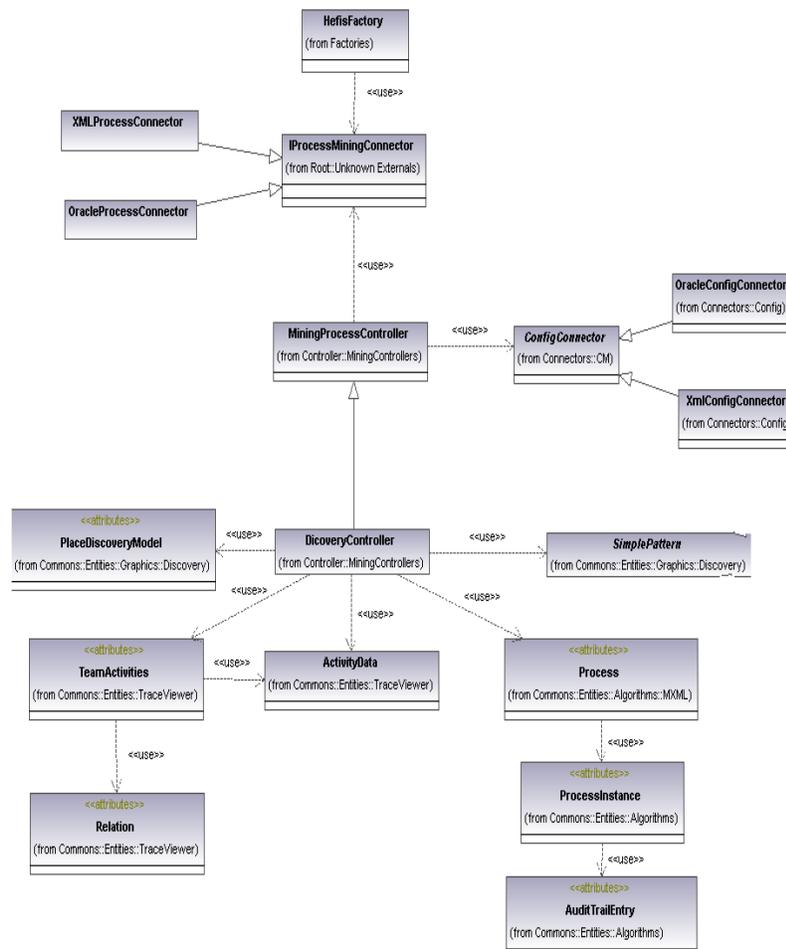


Fig. 5 Fragmento del Diagrama de clases de diseño.

2.7.2 Clases conectoras

Las clases conectoras son las que manejan directamente el acceso a la base de datos. Manipulan entidades de acceso a datos y cuentan con la estructura de la lógica de acceso a datos, permitiendo una independencia total del gestor de base de datos a utilizar de modo que algún cambio en la bases de datos solo genera la actualización de estas clases. A continuación, en la *Tabla 4* se muestra un ejemplo de clase conectora implementada en el módulo.

Nombre	XmlProcessConnector
Descripción	Interfaz que permite la conexión del acceso a datos con el negocio.
Métodos	Descripción
GetProcessById	Método que permite obtener la lista de los procesos dado un id.
GetAllProcess	Método que permite obtener la lista de todos los procesos
GetTraceData	Método que permite obtener la lista de procesos teniendo en cuenta un criterio de búsqueda.

Tabla 4 Descripción de la clase conectora XmlProcessConnector.

2.7.3 Clases gestoras

Las clases gestoras emplean a las clases conectoras haciendo uso de los datos que devuelven y conformando las clases del negocio, lo cual facilita el mantenimiento de las aplicaciones disminuyendo el esfuerzo de actualización en caso de ocurrir cambios en la fuente de datos y además permite la realización de las pruebas a la aplicación mediante implementación de conectores de pruebas que no accedan a datos reales. En la *Tabla 5* que se muestra a continuación se ejemplifica la implementación en el módulo de una clase gestora.

Nombre	MiningProcessController
Descripción	Interfaz que permite el manejo de la clase conectora.
Métodos	Descripción
SearchProcess	Método que devuelve la lista de procesos que coinciden con un criterio de búsqueda.
GetAllProcess	Método que devuelve la lista de todos los procesos.
GetCurrentProcess	Método que devuelve el tiempo que lleva el proceso ejecutándose.
GetOffices	Método para listar las oficinas existentes tanto en los ficheros *.mxml como en la base de datos.
GetUsers	Método para listar los usuarios existentes tanto en los ficheros *.mxml como en la base de datos.
ConvertPath	Método para convertir direcciones de ficheros.
GetProcessInstance	Método que devuelve la lista de instancias de un proceso.
IsOffice	Método que permite conocer si existe una oficina dada en una lista.
IsUser	Método que permite conocer si existe un usuario dado en una lista.

GetServiceTimeData	Método que devuelve el tiempo de servicio de una lista de procesos.
ServiceTime	Método que devuelve el tiempo de servicio de un proceso.
GetProcessAverageTime	Método que devuelve el tiempo promedio de ejecución de un proceso.
GetGestationTime	Método que devuelve el tiempo de gestación de una instancia de un proceso.

Tabla 5 Descripción de la clase gestora MiningProcessController.

2.7.4 Clases entidad

Las clases entidad constituyen una representación de los datos guardados en la base de datos y permite la presencia de los mismos a lo largo del ciclo de vida. Estas son las encargadas de modelar información que posee larga vida y que es a menudo persistente. Ofrecen soporte y manejan actividades específicas. El módulo de análisis de trazas hace uso de la clase *Process* que se utiliza para almacenar los datos referentes a los procesos registrados, la descripción de la misma se muestra en la *Tabla 6*.

Nombre de la Entidad	Process	
Descripción	<i>Entidad que representan los datos de los procesos.</i>	
Atributo	Tipo de dato	Descripción
Id	string	Identificador del proceso
Description	string	Nombre del proceso
Cases	List<ProcessInstance>	Lista de instancias del proceso

Tabla 6 Descripción de la clase entidad Process.

2.8 Modelo de datos

Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Por lo general, permite describir los elementos que intervienen en una realidad o problema dado y la forma en que se relacionan dichos elementos entre sí. (27) A continuación, en la *Fig. 6* se muestra un fragmento del modelo de datos utilizado en el desarrollo del módulo. El modelo de datos extendido se muestra en el *Anexo 6*.

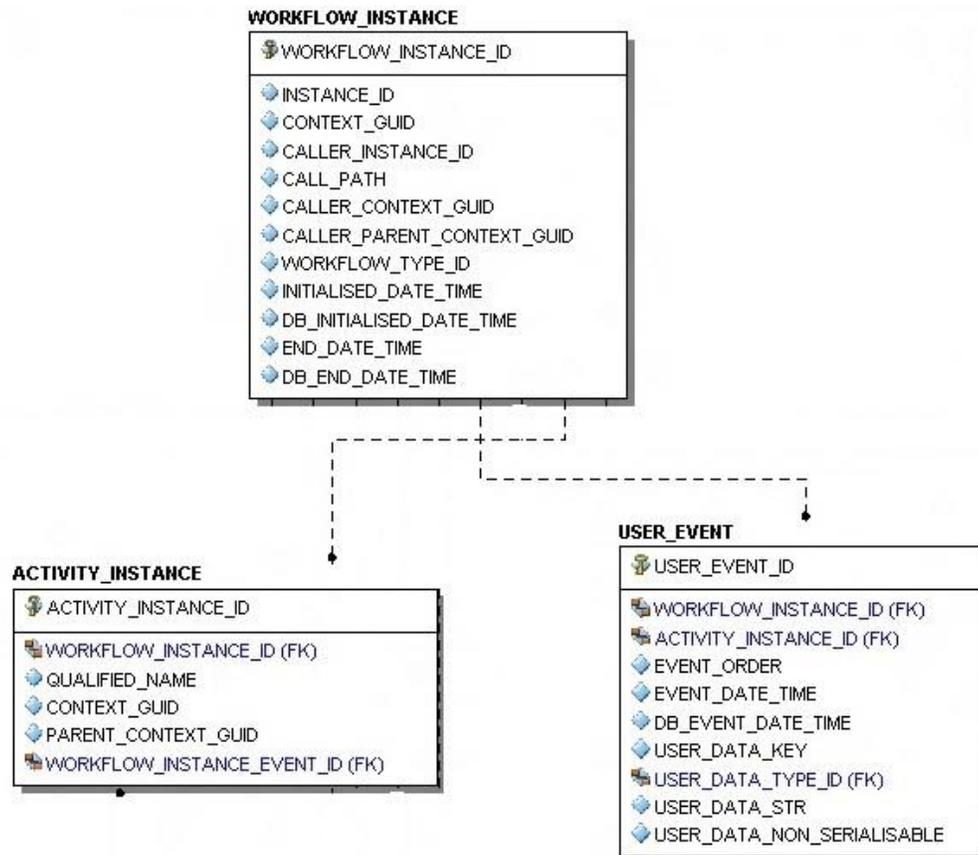


Fig. 6 Fragmento del modelo de datos.

En el presente fragmento se muestra la tabla **Workflow_Instance** que es donde se encuentra toda la información relacionada con las instancias de los procesos, la que se encuentra relacionada con las tablas **Activity_Instance** y **User_Event**. La tabla **Activity_Instance** es la que contiene la información asociada a las actividades de los procesos y posee como llave foránea el identificador de la instancia del proceso con la que se encuentra relacionada. Por su parte, la tabla **User_Event** contiene los eventos lanzados por los usuarios, y como llaves foráneas posee los identificadores de las instancias de actividades y procesos en los que interviene. (28)

Conclusiones parciales

Después de realizar un estudio relacionado con el análisis de los procesos que se ejecutan en el SUIN, se demostró que no existe procedimiento alguno para la ejecución de los análisis de los procesos. Esto impuso la necesidad de diseñar un modelo de dominio para llevar a cabo el desarrollo de la solución propuesta. Se tuvieron en cuenta los propósitos de la solución y con ello, fueron definidos y descritos para un mejor entendimiento los requisitos funcionales y no funcionales del producto. Luego de evaluar el destino y funcionalidades del módulo fueron definidos los roles que interactuarán con el sistema e identificadas las entidades fundamentales atendiendo a su importancia dentro del proceso a desarrollar. Además, fue definida la arquitectura a emplear y se determinó el modelo de datos a utilizar.

Capítulo 3 Implementación y pruebas

Introducción

En el presente capítulo se muestra cómo se ha implementado el módulo. Se describen el diagrama de componentes y el diagrama de despliegue del sistema. Se muestra además, un modelo de implementación donde se detalla la organización y dependencia entre los nodos físicos que conforman los componentes necesarios para el funcionamiento de la aplicación. Además son presentadas las interfaces de usuario definidas para el módulo. En el capítulo también se realiza la validación de la calidad de dicho módulo, utilizando las pruebas de caja blanca y caja negra aplicadas a la solución.

3.1 Estándares de codificación

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema. El mantenimiento del código es la facilidad con que el software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o mejorar el rendimiento. Aunque la legibilidad y el mantenimiento son el resultado de muchos factores, un aspecto del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurar que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código de rutina.

Un patrón de nomenclatura coherente es uno de los elementos más importantes de la previsibilidad y el descubrimiento de una biblioteca de clases administradas. El uso generalizado y la comprensión de estas directrices de denominación deberían eliminar muchas de las preguntas más comunes.

Estilos para la capitalización

Se utilizaron los siguientes tres convenios para la capitalización de los identificadores:

- **Pascal:** La primera letra en el identificador y la primera letra de cada subsiguiente palabra concatenada se capitalizan. Se pueden utilizar los identificadores de *Pascal case* en caso de tres o más caracteres. Por ejemplo: *BackColor (29)*
- **Camello:** La primera letra en el identificador se pone en minúscula y la primera letra de cada subsiguiente palabra concatenada en mayúscula. Por ejemplo: *BackColor (29)*

- **Mayúscula:** Todas las letras en el identificador se capitalizan. Esta convención se utiliza sólo para los identificadores que constan de dos o menos letras. Por ejemplo: *System.IO*, *System.Web.UI* (29)
- **Sensibilidad a mayúsculas:** Para evitar confusiones y garantizar la interoperabilidad entre lenguajes, se siguieron las siguientes reglas sobre el uso de mayúsculas y minúsculas:
 - No utilizar *namespaces*²⁰ que solo se diferencien con en el uso de las mayúsculas.
 - No utilizar funciones con nombres de parámetros que se diferencian solo en el uso de las mayúsculas.
 - No utilizar *namespaces* con nombres de clases que se diferenciaran solo en el uso de las mayúsculas.
 - No crear clases con propiedades que se diferencian solo en el uso de las mayúsculas.
 - No crear clases con métodos que se diferencien solo en el uso de las mayúsculas. (29)
 - Para nombrar las clases no se recomienda usar los nombres de los *namespaces* y otras clases comúnmente usadas, por ejemplo *System*²¹, *Collections*²² o *Forms*²³.
- **Evitando confusión de nombre y tipo:** distintos lenguajes de programación usan diversos términos para declarar los principales tipos de identificadores. Los diseñadores de librerías de clases deben definir una terminología de lenguaje específica. Fueron utilizados nombres que describen a sus identificadores en vez de nombres que describen el tipo de identificador.

Con el objetivo de incrementar la legibilidad del código también se emplearon comentarios en todas las declaraciones de clases y funciones más complejas. También se organizó el código de forma estructurada, en bloques de código, para una mejor lectura del mismo. (30)

3.2 Tratamiento de errores

El diseño de un sistema no solo debe tener en cuenta lo que debe ocurrir, sino que además debe realizar un análisis profundo de las diferentes situaciones que se puedan presentar y que constituyen algún tipo de violación o de situación en particular que provocaría un error dentro del sistema.

²⁰ Define el nombre de un espacio de trabajo que puede ser un componente o proyecto.

²¹ Del inglés: *Sistema*.

²² Del inglés: *Colecciones*.

²³ Del inglés: *Formas*.

Las excepciones son parte del diseño aun cuando la gran mayoría de ellas son identificadas en el proceso de implementación, no obstante es necesario definir con antelación un mecanismo efectivo para su tratamiento. En el análisis de los procesos para lograr una respuesta rápida ante cualquier error se realizó un estudio de errores, para lanzar excepciones en un lenguaje de fácil comprensión para los usuarios. En cada uno de los métodos se evitan, minimizan y tratan los posibles errores. Para el tratamiento de excepciones se definió una interfaz base que permite dejar una traza clara de todas las excepciones lanzadas dentro del sistema.

3.3 Implementación

La implementación es el flujo de trabajo donde se desarrolla el sistema en términos de componentes, los cuales pueden ser ejecutables, ficheros de código fuente, scripts, librerías, etc. En este flujo de trabajo se desarrolla la arquitectura y el sistema como un todo, definiendo y aplicando patrones y estándares para la codificación con el fin de alcanzar una óptima organización del código. Los diagramas de componentes y despliegue conforman el modelo de implementación; el diagrama de componentes describe la organización y dependencia de cada uno de los componentes implementados, representándose la distribución de los mismos en los distintos nodos físicos en los que funcionará la aplicación. (31)

3.3.1 Diagrama de componentes

Los diagramas de componentes se encargan de mostrar las organizaciones y dependencias lógicas que existen entre los componentes de software, sean estos componentes de código fuente, binarios o ejecutables. Este diagrama contiene componentes y relaciones entre ellos que a su vez son utilizados para agrupar los elementos de modelado (32).

En la *Fig. 7* se muestra el diagrama de componentes de la solución propuesta. En el diagrama se muestra la **AdministracionGlobal** que hace uso del **CommonTools** que es el encargado de regir el comportamiento del sistema web y del **UIToolsBox** para la presentación, específicamente las interfaces de usuario.

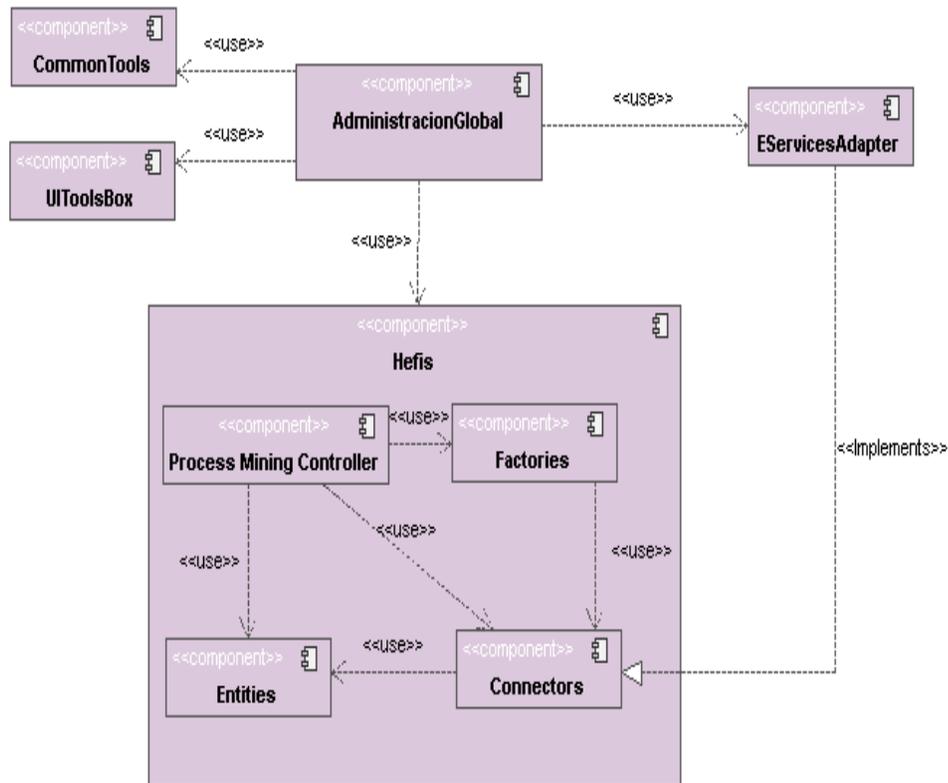


Fig. 7 Diagrama de componentes.

La administración también hace uso del componente **Hefis**, el cual contiene una serie de subcomponentes interrelacionados que permiten dar solución a las funcionalidades requeridas y que es donde se encuentra desarrollado el módulo. El módulo también cuenta con el **EServicesAdapter**, que es el que permite la conexión del sistema con quien tiene la información, a través de los conectores (**Connectors**).

3.3.2 Diagrama de despliegue

“Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos”. La vista de despliegue “representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación”. Un nodo es un recurso de ejecución tal como un servidor, computadora, un dispositivo o memoria. (33)

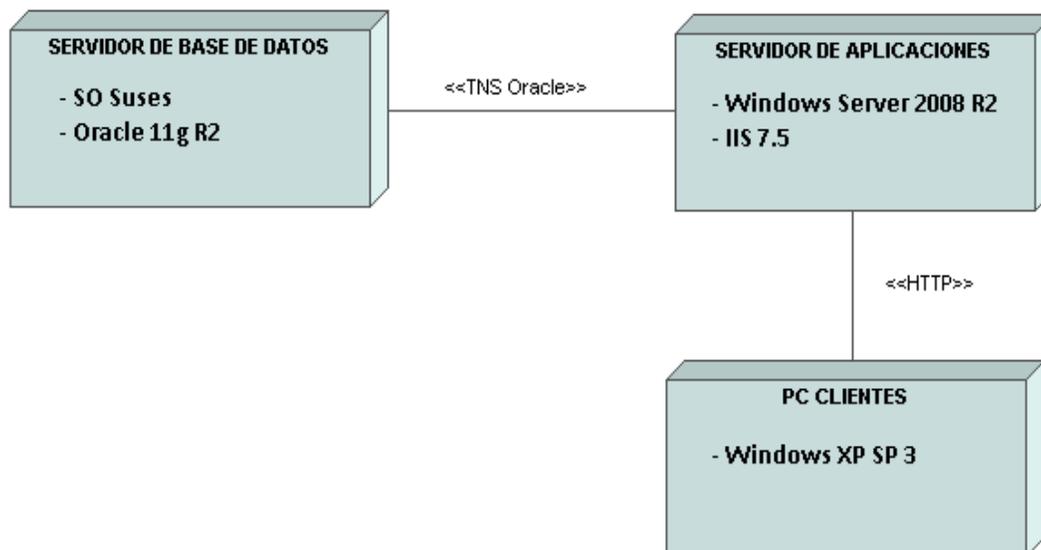


Fig. 8 Diagrama de despliegue.

La Fig. 8 muestra la propuesta de despliegue para el sistema. Esta cuenta con un servidor de base de datos que tiene como sistema operativo (SO) Suses y como SGBD Oracle 11g R2. El mismo tendrá una conexión mediante el protocolo TNS²⁴ para Oracle con el servidor de aplicaciones. Este servidor cuenta con el SO Windows Server 2008 R2, con el Internet Information Server (IIS) en su versión 7.5 y estará conectado mediante un protocolo HTTP²⁵ con las PC²⁶ clientes. Estas estaciones de trabajo contarán con el SO Windows XP Service Pack 3 o superior.

3.4 Prototipos de interfaz del módulo

El diseño de los prototipos de interfaz de usuario tuvo en cuenta la integración del módulo con el módulo de administración del SUIN. Fueron concebidas las interfaces de manera intuitiva y de fácil navegación y entendimiento. A continuación se muestran algunos de los prototipos de interfaz definidos para el módulo de análisis de trazas de procesos del SUIN. El resto de los prototipos pueden ser consultados en el Anexo 4.

²⁴ Por sus siglas en inglés: *Transparent Network Substrate*.

²⁵ Del inglés: Hypertext Transfer Protocol.

²⁶ Del inglés: *Personal Computer*.



Fig. 9 Prototipo de interfaz para la selección de la fuente de datos.

En la Fig. 9 se muestra el prototipo de interfaz definido para la selección de la fuente de datos. En el mismo, se le da la posibilidad de seleccionar desde dónde desea cargar los datos de los procesos. En el caso de los ficheros *.mxml, son listados los ficheros con los que ya se han trabajado anteriormente. Brinda además la posibilidad de agregar otros ficheros proporcionando la ruta física. En el otro caso, son listadas las bases de datos disponibles para cargar los datos.

A continuación, en la Fig. 10 se muestra la interfaz definida para realizar la búsqueda de los procesos desde un fichero *.mxml. En ella al usuario se le muestra la ubicación del fichero con el que está trabajando y se le muestran los campos para realizar la búsqueda de los procesos. Posee además el área donde se mostrarán los resultados de la búsqueda, las operaciones de minería a realizar y el área para mostrar datos de los procesos, instancias y actividades. En ella también se puede encontrar la barra de navegación para navegar dentro del proceso o la instancia seleccionada.

Fichero

Nombre de fichero:

Seleccionar usuario

Usuario:

Buscar proceso

Nombre del proceso:
 Fecha inicio:
 Fecha fin:
 Oficina:

Carné de Identidad	Ver estados
Carné de Identidad	Ver estados

> Carné de Identidad
> asdasd-dasd2d-d2dasd-2d
>

Minería de proceso

- Descubrimiento
- Chequeos de conformidad
- Extensión

Datos de elemento seleccionado

Nombre: Carné de Identidad
 Actividades: 6
 Fecha inicio: 10/25/2011 10:46:50
 Fecha fin: 10/25/2011 10:46:51
 Período de gestación: 00:03:01
 Tiempo de servicio: 00:01:10

Fig. 10 Prototipo de interfaz para buscar procesos desde ficheros MXML.

3.5 Pruebas

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requisitos especificados, se observan los resultados y son registrados, para luego llevar a cabo una evaluación del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del sistema y representa una revisión final de las especificaciones del diseño y de la codificación. Las pruebas pueden ser clasificadas en pruebas de caja blanca y pruebas de caja negra. Las pruebas de caja blanca

Visual Studio Team System Test incluye un conjunto de herramientas de prueba perfectamente integradas con Visual Studio, que no sólo funcionan en su propio marco de pruebas, sino también en un marco más amplio de herramientas de ciclo de vida del software. *Test Edition* permite crear, administrar, editar y ejecutar pruebas, así como obtener y almacenar los resultados de las pruebas. Incluye varios tipos de pruebas, como manuales, unitarias, de web y de carga. Las pruebas se ejecutan mediante el IDE²⁷ de Visual Studio. Además, puede ejecutar grupos de pruebas o cualquier prueba individual que no sea el tipo de prueba manual desde una línea de comando. (34)

3.5.1 Pruebas de caja blanca

Pruebas unitarias

Las pruebas unitarias representan una forma de probar un fragmento de código por lo que tiene gran importancia para los desarrolladores y el equipo de pruebas. Las pruebas unitarias de Microsoft son una herramienta muy importante para el personal de pruebas, sobre todo para los desarrolladores. Aunque el tamaño de la "unidad" que se prueba puede variar, la herramienta para pruebas del Visual Studio 2010 permite generar pruebas unitarias específicas para métodos y funcionalidades. (35)

En la *Fig. 11* se muestra una de las pruebas unitarias aplicadas. El resto de las pruebas unitarias aplicadas se pueden observar en el *Anexo 2*.

```
/// <summary>
///A test for ConvertPath
///</summary>
[TestMethod()]
public void ConvertPathTest()
{
    var configConnector = new XmlConfigConnector{ Path = "D:\\Ivancito\\TESIS\\Implementación\\TESIS\\Modulo Ivan\\SUIIN_10\\Administracion\\V
    var target = new MiningController(configConnector);|
    string path = "D:\\Ivancito\\TESIS\\Implementación\\TESIS\\Modulo Ivan\\SUIIN_10\\Administracion\\Administracion\\Dev\\source\\Administrac
    string expected = "D:/Ivancito/TESIS/Implementación/TESIS/Modulo Ivan/SUIIN_10/Administracion/Administracion/Dev/source/AdministracionGlot
    string actual;
    actual = target.ConvertPath(path);
    Assert.AreEqual(expected, actual);
}
```

Fig. 11 Prueba unitaria aplicada al método ConvertPath.

²⁷ Por sus siglas en inglés: *Integrated Development Environment*.

El resultado satisfactorio alcanzado por un sistema ante la aplicación de las pruebas funcionales y de código realizadas, es una garantía de qué tan bien está desarrollado el software en correspondencia con el cumplimiento de los requerimientos especificados. En la Fig. 12 se muestran los resultados obtenidos de la aplicación de las pruebas unitarias a algunos métodos que implementan parte de alguna funcionalidad requerida.

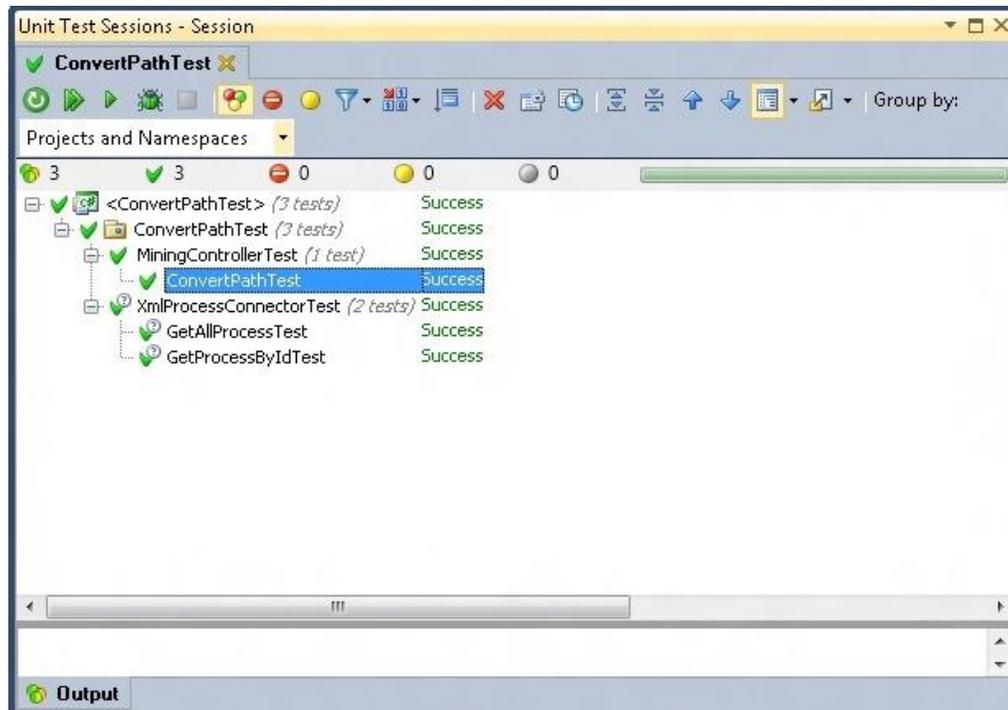


Fig. 12 Resultados de las pruebas unitarias.

3.5.2. Pruebas de caja negra

Las pruebas de sistema (pruebas de caja negra) se refiere a aquellas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (36) A continuación se muestra la descripción del

CPRF²⁸ 1 Extraer datos de los sistemas fuente. El resto de los casos de prueba descritos pueden observarse en el Anexo 3.

CPRF 1 Extraer datos de los sistemas fuente.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1. Extraer datos de los sistemas fuente.	1.1 Seleccionar fuente de datos.	El sistema carga los datos desde la fuente de datos seleccionada.	1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.2 Seleccionar fuente de datos sin seleccionar fichero.	El sistema muestra un mensaje indicando que no ha seleccionado ningún fichero.	1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.3 Seleccionar fuente de datos sin seleccionar base de datos.	El sistema muestra un mensaje indicando que no ha seleccionado ninguna base de datos.	1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.4 Agregar nuevo fichero *.mxml a la lista de ficheros.	El sistema agrega el nombre del fichero a la lista de ficheros *.mxml.	1. Minería de procesos. 2. Minar proceso. 3. Agregar.
	1.5 Introducir dirección de archivo *.mxml no válida.	El sistema muestra un mensaje de error indicando que el fichero debe existir en la dirección.	1. Minería de procesos. 2. Minar proceso. 3. Agregar.
	1.6 Introducir dirección de archivo *.mxml que ya existe en la lista.	El sistema muestra un mensaje de error indicando que el fichero ya existe en la lista.	1. Minería de procesos. 2. Minar proceso. 3. Agregar.

²⁸ Caso de Prueba del Requisito Funcional.

	1.7 Cancelar selección realizada.	El sistema muestra nuevamente la página sin selección.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Cancelar
--	-----------------------------------	--	---

Tabla 7 Descripción del CPRF 1 Extraer datos de los sistemas fuente.

La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software, al ejecutar dicho caso para detectar un posible fallo en el programa.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
1.4	Clases Válidas	Dirección del archivo: "D:\escritorio\file.xml"	Se agrega el archivo a la lista.	Se activa una estación de trabajo.	
1.5	Clases Inválidas	Dirección del archivo: "D:\escritorio\"	Se muestra un mensaje de error indicando que el fichero debe existir en la dirección.	Se muestra un mensaje de error indicando que el fichero debe existir en la dirección.	
1.7	Clases Inválidas	Dirección del archivo: "D:\escritorio\file.xml"	Se muestra un mensaje de error indicando que el fichero ya existe.	Se muestra un mensaje de error indicando que el fichero ya existe.	

Tabla 8 Resultado de la ejecución del CPRF 1 Extraer datos de los sistemas fuente.

La aplicación de las pruebas a la versión funcional del módulo de análisis de trazas de procesos en su primera versión, ha arrojado una serie de no conformidades que han sido eliminadas durante las iteraciones por las que ha transitado la aplicación. (Ver Fig. 13 Resultados de las iteraciones)



Fig. 13 Resultados de las iteraciones.

3.6 Beneficios del módulo

Con la implementación del módulo de análisis de trazas de procesos del Sistema Único de Identificación Nacional de la población de la República de Cuba se obtendrán beneficios que serán palpables en cuanto a costo en tiempo de respuesta y nivel de automatización. La aplicación desarrollada presenta características tales como:

- Realiza el descubrimiento de los procesos ejecutados por el SUIN empleando técnicas de minería de procesos.
- Incorpora además la detección de cuellos de botella y las predicciones de tiempo de los procesos que se encuentran en ejecución.
- Toma las informaciones históricas y parciales almacenadas tanto en ficheros *.mxml como en las tablas de una base de datos.

Conclusiones parciales

En el presente capítulo se llega a la solución final arribando a las siguientes conclusiones:

- Empleo de los estándares de codificación permitió se obtuviera como resultado un código más limpio, legible y organizado.
- El tratamiento de las excepciones identificadas permitió se interceptaran a tiempo los posibles errores que pudieran ser presentados tanto por la aplicación como por el usuario.

- El diagrama de despliegue permitió el uso de la aplicación teniendo en cuenta los requisitos no funcionales definidos.
- La realización de las pruebas permitió obtener una versión de la solución que cumple con todas las funcionalidades requeridas, independientemente de que en la 3ra iteración hayan sido identificadas 2 no conformidades.

Conclusiones generales

Con la realización del presente trabajo de diploma se obtuvo la primera versión del Módulo de Análisis de Trazas de Procesos del Sistema Único de Identificación Nacional de la República de Cuba. Esta aplicación garantizará conocer a profundidad las ejecuciones de los procesos de negocio que se realizan en el sistema. Con ella se estará aportando un mecanismo para la detección de cuellos de botella y la realización de predicciones de tiempo en los procesos.

Durante la ejecución del trabajo:

- El estudio de las técnicas, algoritmos y soluciones existentes, permitió se detectara la necesidad de crear un nuevo software que dé solución a la problemática planteada.
- La metodología, técnicas y herramientas utilizadas, permitieron el desarrollo de una aplicación acorde con los estándares y procedimientos requeridos por el cliente.
- El modelo de dominio y la arquitectura propuesta, permitieron sentar las bases para dar respuesta a los requerimientos funcionales y no funcionales definidos para la aplicación.
- El desarrollo del módulo para el análisis de trazas utilizando técnicas de minería de procesos, permite la detección de cuellos de botella, así como la realización de predicciones de tiempo en los procesos de negocio previamente ejecutados en el SUIN.

Recomendaciones

Luego de haber desarrollado una primera versión del Módulo de análisis de trazas de procesos se recomienda:

- Hacer uso de otras tecnologías, librerías o frameworks que permitan tener mayor control sobre el graficado del modelo.
- Incluir en el módulo la realización de chequeos de conformidades y el descubrimiento de los modelos de redes sociales y organizacionales.
- Incluir en el módulo la realización de perfiles de cuellos de botella, de manera tal que el usuario pueda definir los tiempos de las actividades en dependencia del tipo de actividad.

Bibliografía

1. **Bauta Camejo, René R. y Torres Galvez, Ariel.** *Herramienta para el Registro y Monitoreo de Trazas en el Sistema de Gestión Integrar CEDRUX*. Universidad de las Ciencias Informáticas. Ciudad de la Habana : s.n., 2010. Tesis.
2. **van der Aalst, Wil M.P.** *Process Mining Discovery, Conformance and Enhancement of Business Processes*. Schleidon : Springer, 2010.
3. **2000, ISO.** *Norma Internacional ISO 9001 - Sistemas de gestión de la calidad – Requisitos*. [ed.] Secretaría Central de ISO. Ginebra : s.n., 2005.
4. **Barros, Oscar.** *Reingeniería de Procesos de negocio*. Santiago de Chile : Dolmen, 1994.
5. **Davenport, Thomas.** *Process Innovation*. s.l. : Harvard Business School Press, 1993.
6. **University, Oxford.** *Oxford Dictionary*.
7. **van der Aalst, Wil, y otros, y otros.** *Manifiesto sobre Minería de Procesos*. 2012.
8. **MEGA International, 1994 - 2012.** *Mega. Mega*. [En línea] [Citado el: 2 de Mayo de 2012.] <http://www.mega.com/en/c/product>.
9. **Castillo, Paola Alejandra Alvarado.** *BONITA SOFT: Gestor de procesos de negocios BPM*. Universidad Nacional de Colombia. Bogotá : s.n., 2011.
10. **ARIS Process Performance Manager.** *ARIS Process Performance Manager*. [En línea] http://www.softwareag.com/corporate/products/aris_platform/aris_controlling/aris_process_performance/overview/default.asp.
11. **Intel® Software Network.** *Intel® Software Network*. [En línea] Intel®. [Citado el: 18 de Enero de 2012.] <http://software.intel.com/en-us/articles/intel-trace-analyzer/>.
12. **ProM Process Mining Workbench.** [En línea] <http://www.promtools.org/prom6/>.
13. **Upload and share.** [En línea] [Citado el: 22 de enero de 2011.] www.slideshare.net/deboradipiano/3metodologas-y-templates-para-tfs.
14. **Martínez, Claudia.** *La Web del programador*. [En línea] 14 de julio de 2008. [Citado el: 17 de Enero de 2011.] <http://www.lawebdelprogramador.com/news/>.
15. **Booch, G., Jacobson, I. y Rumbaugh, J.** *The UML specification documents*. 1997.

16. **Larman, Craig.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos.* México : PRENTICE HALL, 1999. 970-17-0261-1.
17. Mozilla Developer Network. [En línea]
https://developer.mozilla.org/es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript.
18. Centro de desarrollo de .NET Framework. *Centro de desarrollo de .NET Framework.* [En línea]
<http://msdn.microsoft.com/es-es/netframework/aa496123>.
19. Microsoft .NET Framework 4 (instalador web). *Microsoft .NET Framework 4 (instalador web).* [En línea]
<http://www.microsoft.com/downloads/es-es/details.aspx?familyid=9cfb2d51-5ff4-4491-b0e5-b386f32c0992>.
20. **Microsoft.** LINQ (Language-Integrated Query). *LINQ (Language-Integrated Query).* [En línea] [Citado el: 9 de Diciembre de 2011.] <http://msdn.microsoft.com/es-es/library/bb397926.aspx>.
21. *Oracle 11g R1 Features.* 2009.
22. **Corporation, Microsoft.** MSDN. [En línea] [Citado el: 22 de enero de 2011.]
<http://www.microsoft.com/spain/visualstudio/products/2010-editions/premium..>
23. **Cuéllar, Javier Fernández.** *CALIDAD Y SEGURIDAD A NIVEL DE FILAS EN BBDD ORACLE.* Universidad Carlos III de Madrid : s.n., 2009.
24. [En línea] 17 de enero de 2011. [Citado el: 9 de 12 de 2011.] <http://www.altova.com/umodel.html>.
25. **Microsoft.** MSDN Magazine. *MSDN Magazine.* [En línea] <http://msdn.microsoft.com/es-es/magazine/ee236415.aspx>.
26. **Pressman, Roger S.** *Ingeniería del Software: Un enfoque práctico.* México DF : Editorial McGraw Hill, 2006.
27. Definiciones. *Definiciones.* [En línea] [Citado el: 31 de marzo de 2011.] <http://definicion.de/modelo-de-datos/>.
28. **Proyecto Identificación, Inmigración y Extranjería de la República de Cuba.** *MODELO DE DATOS - BISON.* Centro de Identificación y Seguridad Digital (CISED), Universidad de las Ciencias Informáticas. La Habana : s.n., 2012.
29. **Proyecto Identidad Cuba.** *Estándares de codificación.* Ciudad de la Habana : s.n., 2009.
30. **Blanco, Reynier y Salazar, Humberto.** *Documento de Arquitectura.* Ciudad de la Habana : s.n., 2010.
31. **Turossi, AUS Gustavo.** Scribd. [En línea] 2008. <http://es.scribd.com/doc/51305810/50/Implementacion>.
32. **Microsoft.** MSDN. [En línea] 2010. [Citado el: 8 de 1 de 2012.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.

33. **Jacobson, Ivar.** *El proceso unificado de desarrollo de Software.* s.l. : Addison-Wesley Iberoamericana España, S.A., 2000.
34. **Grupo de Aseguramiento de la Calidad Proyecto Identificación, Migración y Extranjería.** *Manual de Pruebas utilizando Microsoft Visual Studio Team System.* Habana : s.n., 2010.
35. **Microsoft.** MSDN. [En línea] 2011. [Citado el: 12 de 03 de 2012.] [http://msdn.microsoft.com/es-es/library/ms182215\(2v=vs.180\).aspx](http://msdn.microsoft.com/es-es/library/ms182215(2v=vs.180).aspx).
36. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico.*
37. **van der Aalst, Wil M.P., y otros, y otros.** *Process Mining: A Two-Step Approach using Transition Systems and Regions.* 2006. pág. 1.
38. **Microsoft.** Microsoft Visual Studio Team Foundation Server 2010. *Microsoft Visual Studio Team Foundation Server 2010.* [En línea] <http://www.microsoft.com/downloads/es-es/details.aspx?FamilyID=3660cacf-f077-44d3-a9d9-97e801da2035>.
39. **Garlan, David y Shawn, Mary.** *An Introduction to Software Architecture.*
40. **Cabrera., Irving Deither Cao.** *XCAD, Generador de Capas de Acceso a Datos.* Habana. Cuba : s.n., 2008.
41. **Microsoft.** MSDN Online. *ADO.NET Entity Framework.* [En línea] <http://msdn.microsoft.com/es-es/library/bb399572.aspx>.
42. —. MSDN Library . *MSDN Library.* [En línea] <http://www.msdn.microsoft.com/en-us..>
43. **Bob Bryla, Kevin Loney.** *Oracle Database 11g DBA Handbook (Osborne ORACLE Press Series).*
44. **WCF, MSDN.** MSDN Library. [En línea] <http://msdn.microsoft.com/es-es/library/ms731082.aspx>.
45. **Pimentel, Luis Alberto y Rivero, Iósev Pérez.** *ArBaWeb: ARQUITECTURA BASE SOBRE LA WEB.* s.l. : UCI, junio del 2007.
46. **Microsoft.** *Microsoft Application Architecture Guide 2nd Edition.* 2009.
47. **Reinier Sotes, Sandy Moore.** *Propuesta de integración para los servicios web del Sistema de Emisión de Documentos de Identificación del Centro de Identificación y Seguridad Digital.* Ciudad Habana : Uci, Mayo 2009.
48. *La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real.* **Microsoft Corporation.** 2006.
49. **Líderes del grupo de Tecnología SOA de Accenture.** *Arquitectura Orientada a Servicios.* s.l. : Un estudio publicado por el Centro de Alto Rendimiento de Accenture (CAR), 2008.

50. **Barco, Antonio.** Arquitectura Orientada a Servicios (SOA) . [En línea] 9 de Junio de 2006.
<http://arquitecturaorientadaaservicios.blogspot.com/>.
51. **Gartner, Inc.** [En línea] 18 de Abril de 2011. <http://procesamientoeventos.blogspot.com/2011/04/definicion-de-bam.html>.
52. **Microsoft.** Características de WCF. [En línea] 2011. <http://msdn.microsoft.com/es-es/library/ms731082.aspx>.
53. —. Información general sobre seguridad. *MSDN*. [En línea] 2011. <http://msdn.microsoft.com/es-es/library/ms735093.aspx>.
54. **M.Crispino.** Petroglifo. [En línea] 14 de septiembre de 2001. [Citado el: 5 de enero de 2011.]
<http://www.concepto.com.uy/petrocsharp/hwnver03.aspx?1,N,0,0,101>. 10.
55. **W3 Schools.** w3schools.com. [En línea] 2010. <http://www.w3schools.com/soap/default.asp>.
56. **Miranda, Emmerson.** Desarrollo Java. [En línea] 27 de Julio de 2009. [Citado el: 4 de enero de 2011.]
<http://www.emmersonmiranda.net/2009/07/que-es-un-esb.html>.
57. **L., Ing. Maria Eugenia Arevalo.** Maria Eugenia Arevalo Blog. [En línea] 7 de enero de 2010.
<http://arevalomaria.wordpress.com/2010/01/07/ejemplo-de-documento-vision-y-alcance/>.
58. **Microsoft.** MSDN. *Introducing Windows Communication Foundation in .NET Framework 4*. [En línea] 2010.
<http://msdn.microsoft.com/en-us/library/ee958158.aspx>.
59. **Proyecto Identidad Cuba. UCI.** *Arquitectura de Software*. [Documento] 2010.
60. —. *Proyecto Técnico. DIE*. [Documento] 2009.
61. —. *Proyecto Técnico. DIR*. [Documento] 2009.
62. *Utilización del patrón Modelo - Vista - Controlador (MVC) en el diseño de software educativo.* **Febe, Ángel Ricardo.** Ciudad de la Habana : s.n., 2006.
63. **Rojas, José Fabricio.** Desarrollo Alter.Netivo. *Arquitectura MVC*. [En línea] 22 de abril de 2009. [Citado el: 12 de marzo de 2011.] [HTTP://DEVSOFTX.WORDPRESS.COM/2009/04/22/QUE-ES-ASPNET-MVC/](http://DEVSOFTX.WORDPRESS.COM/2009/04/22/QUE-ES-ASPNET-MVC/).
64. **Creative Commons.** LibrosWeb. *La Arquitectura MVC*. [En línea]
http://www.librosweb.es/jobeeet/capitulo4/la_arquitectura_mvc.html.
65. **Oracle Corporation.** PL / SQL. [En línea] 2010. [Citado el: 11 de enero de 2011.]
http://www.oracle.com/technology/tech/pl_sql.

66. —. Oracle Data Provider. [En línea] 2010. [Citado el: 12 de marzo de 2011.] <http://www.oracle.com/technology/obe/hol08/dotnet>.
67. —. Oracle. [En línea] 2010. [Citado el: 12 de enero de 2011.] <http://www.oracle.com/technology/products/database/oracle11g>.
68. **CALISOFT**. *Acta de liberación del subsistema SPDI*. Cuba : s.n., 2011.
69. *Oracle-Warehouse-Builder-11g-Data-Quality-Option.pdf*. 2009.
70. Masadelante. [En línea] [Citado el: 12 de Enero de 2012.] <http://www.masadelante.com/faqs/base-de-datos>.
71. **Altova Coporation**. Altova. [En línea] <http://www.altova.com/list/xmlschema-dev/201001>.
72. Monografias. [En línea] [Citado el: 11 de Enero de 2012.] <http://www.monografias.com/trabajos56/sistemas-bases-de-datos/sistemas-bases-de-datos.shtml>.
73. **Microsoft**. MSDN. *Visual Studio 2010*. [En línea] 2011. [Citado el: 2011 de 01 de 2012.] <http://msdn.microsoft.com/es-es/library/6x6bk1f4.aspx>.
74. —. MSDN. [En línea] 2011. [Citado el: 12 de 03 de 2012.] <http://msdn.microsoft.com/es-es/library/ms735093.aspx>.
75. —. MSDN. [En línea] 2011. [Citado el: 12 de 03 de 2012.] [http://msdn.microsoft.com/es-es/library/ms182215\(2=vs180\).aspx](http://msdn.microsoft.com/es-es/library/ms182215(2=vs180).aspx).
76. —. MSDN. [En línea] [Citado el: 5 de Enero de 2012.] [http://msdn.microsoft.com/es-es/library/aa546827\(BTS.10\).aspx](http://msdn.microsoft.com/es-es/library/aa546827(BTS.10).aspx).

Anexos

Anexo 1: Descripción de requisitos funcionales

RF 1 Extraer datos de los sistemas fuente.

Propósito	Extraer los datos de los sistemas fuente.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
Descripción	<p>1.1. Permitir al usuario seleccionar la fuente de datos:</p> <p style="padding-left: 40px;">a) Ficheros MXML.</p> <p style="padding-left: 40px;">b) Base de datos.</p> <p style="padding-left: 40px;">1.1.1. Cargar datos desde un fichero *.mxml: si selecciona Ficheros MXML.</p> <p style="padding-left: 40px;">1.1.2. Cargar datos desde una base de datos: si selecciona Base de datos.</p> <p>1.2. Mostar lista de ficheros MXML.</p> <p>1.3. Mostrar lista de base de datos disponibles.</p> <p>1.4. Validar selección de fuente de datos.</p> <p>1.5. Permitir al usuario agregar nuevo fichero *.mxml a la lista.</p> <p style="padding-left: 40px;">a) Dirección del archivo.</p> <p style="padding-left: 40px;">1.5.1. Agregar nombre de fichero a la lista: si selecciona Agregar ver 1.6.</p> <p>1.6. Validar dirección y nombre de fichero *.mxml.</p> <p>1.7. Agregar nuevo fichero *.mxml a la lista.</p> <p style="padding-left: 40px;">1.7.1. Agregar nombre de fichero a la lista: si selecciona Agregar.</p> <p style="padding-left: 40px;">1.7.2. Mostrar mensaje de error: si la dirección del archivo es incorrecta o el fichero ya existe en la lista.</p> <p>1.8. Cargar datos desde el archivo *.mxml o la base de datos seleccionada.</p> <p style="padding-left: 40px;">1.8.1. Cargar datos: si la selección se ha realizado correctamente.</p> <p style="padding-left: 40px;">1.8.2. Mostrar mensaje de error: si no se ha seleccionado ninguna fuente de datos.</p> <p style="padding-left: 40px;">1.8.3. Mostrar mensaje de error: si no se ha seleccionado ningún fichero *.mxml o base de</p>	

	<p>datos.</p> <p>1.9. Mostrar página sin selección: si selecciona Cancelar.</p>
Validaciones	<ol style="list-style-type: none">1. Debe haberse seleccionado la fuente de datos y el fichero o base de datos desde donde se cargarán los datos.2. Para agregar un nuevo archivo, el archivo debe existir y no puede encontrarse en la lista de archivos.
Prototipo	

Tabla 9 Descripción del RF 1 Extraer datos de los sistemas fuente.

RF 2 Transformar datos.

Propósito	Transformar los datos de los sistemas fuente.	
Roles	-	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • Debe haberse seleccionado la fuente de datos. • Deben haberse seleccionado el archivo o la base de datos a utilizar. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
Descripción	<p>2.1. Transformar los datos del sistema fuente seleccionado.</p> <p>2.1.1. Transformar a entidad <i>Process</i> los datos de los procesos extraídos desde fichero *.mxml: si selecciona un fichero *.mxml.</p> <p>2.1.2. Transformar a entidad <i>Process</i> los datos de los procesos que se encuentran en tablas de base de datos: si selecciona una base de datos.</p>	
Validaciones	1. Debe haberse seleccionado la fuente de datos y el fichero o base de datos desde donde se cargarán los datos.	
Prototipo	-	

Tabla 10 Descripción del RF 2 Transformar datos.

RF 3 Buscar procesos según criterio de búsqueda.

Propósito	Mostrar los procesos que se han ejecutado o se encuentran en ejecución tanto los almacenados en ficheros <i>.mxml</i> como los almacenados en la Base de datos Nacional, teniendo en cuenta una fecha de inicio y una fecha de fin, un usuario, oficina o tipo de proceso.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse seleccionado la fuente de datos. • Deben haberse registrados los procesos en el fichero <i>.mxml</i>. • Deben existir procesos en ejecución o deben haberse ejecutado con anterioridad. • El administrador debe estar autenticado. • Deben haberse seleccionado la fecha inicio y la fecha fin. • Deben haberse seleccionado la oficina, el usuario o el tipo de proceso. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
Descripción	<p>3.1. Permitir que el usuario introduzca los datos para la búsqueda de los procesos:</p> <ol style="list-style-type: none"> Usuario Fecha inicio Fecha fin Oficina Tipo de proceso <p>3.1.1. Buscar procesos correspondientes: si selecciona Buscar: ver 3.3</p> <p>3.2. Validar que los campos Fecha Inicio y Fecha Fin no se encuentren vacíos o estén correctamente.</p> <p>3.3. Mostar lista de procesos coincidentes con los criterios de búsqueda y los datos generales del usuario.</p> <p>3.3.1. Mostrar en la barra de navegación el texto "Sin resultados": si no existen resultados coincidentes.</p> <p>3.4. Mostrar lista de las instancias del proceso seleccionado.</p> <p>3.5. Mostrar lista de las actividades de la instancia de proceso.</p> <p>3.6. Permitir al usuario retroceder a los resultados iniciales de la búsqueda y a las</p>	

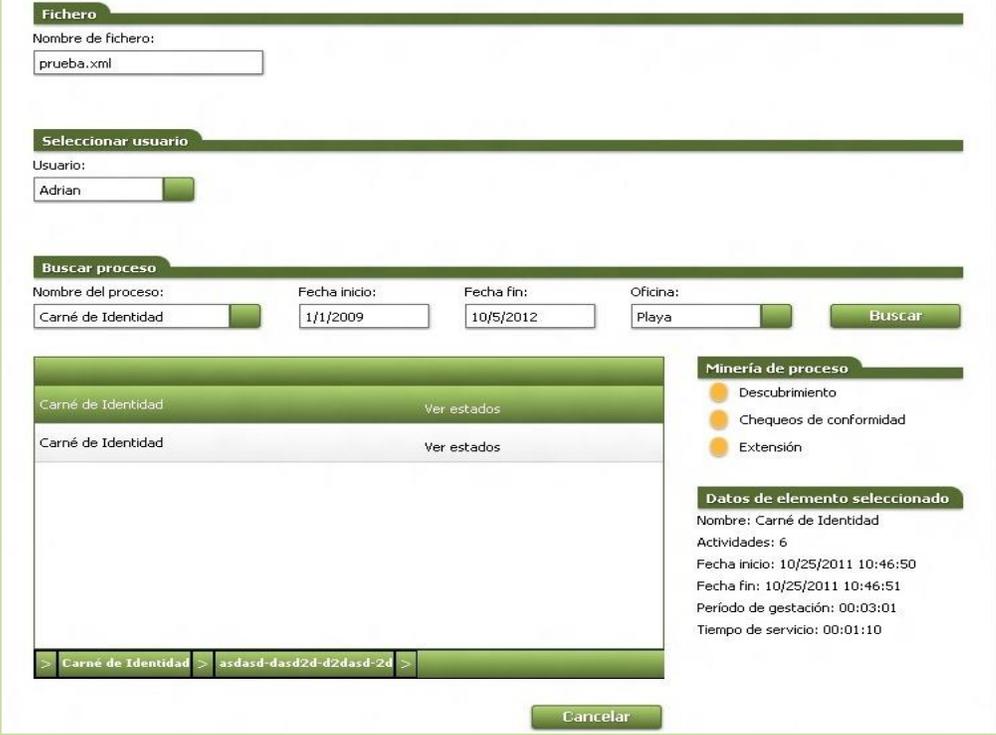
	<p>instancias de procesos del proceso seleccionado.</p> <p>3.6.1. Mostrar resultados iniciales de la búsqueda: si selecciona atrás.</p> <p>3.6.2. Mostrar lista de instancias del proceso: si selecciona el nombre del proceso en la barra de navegación.</p> <p>3.6.3. Mostrar lista de instancias del proceso: si selecciona el Id de la instancia en la barra de navegación.</p> <p>3.7. Mostrar página para seleccionar fuente de datos: si selecciona Cancelar.</p>
Validaciones	<p>1. Los campos Fecha Inicio y Fecha Fin no deben estar vacíos.</p> <p>2. La fecha de inicio no puede ser mayor que la fecha de fin.</p>
Prototipo	

Tabla 11 Descripción del RF 3 Buscar procesos según criterio de búsqueda.

RF 4 Mostrar datos de los procesos, instancias de procesos y actividades.

Propósito	Mostrar datos generales de los procesos, instancias de procesos y actividades.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar autenticado en el sistema. • Debe haberse seleccionado la fuente de datos. • Deben haberse seleccionado el archivo o la base de datos a utilizar. • Debe haberse seleccionado un proceso, instancia de proceso o actividad. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	<p>4.1. Mostrar datos del proceso seleccionado:</p> <ol style="list-style-type: none"> Nombre. Casos. <p>4.2. Mostrar datos de la instancia de proceso (Caso) seleccionada:</p> <ol style="list-style-type: none"> Nombre. Actividades. Fecha Inicio. Fecha Fin. <p>4.3. Mostrar datos de la actividad seleccionada:</p> <ol style="list-style-type: none"> Nombre. Evento. Tiempo. Usuario. Oficina. 	

Validaciones		
Prototipo	GestionarUsuarios	Minería de proceso <input type="radio"/> Descubrimiento <input type="radio"/> Chequeos de conformidad <input type="radio"/> Extensión
	configuration1	
	configuration1	
	AccionesParalelas	Datos de elemento seleccionado Nombre: configuration1 Evento: Closed Tiempo: 10/25/2011 10:46:51 Usuario: Adrian
	FlujoNormal	
	FlujoTiempo	
	> Carné de Identidad > asdasd-dasd2d-d2dasd-2d > configuration1	

Tabla 12 Descripción del RF 4 Mostrar datos de los procesos, instancias de procesos y actividades.

RF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.

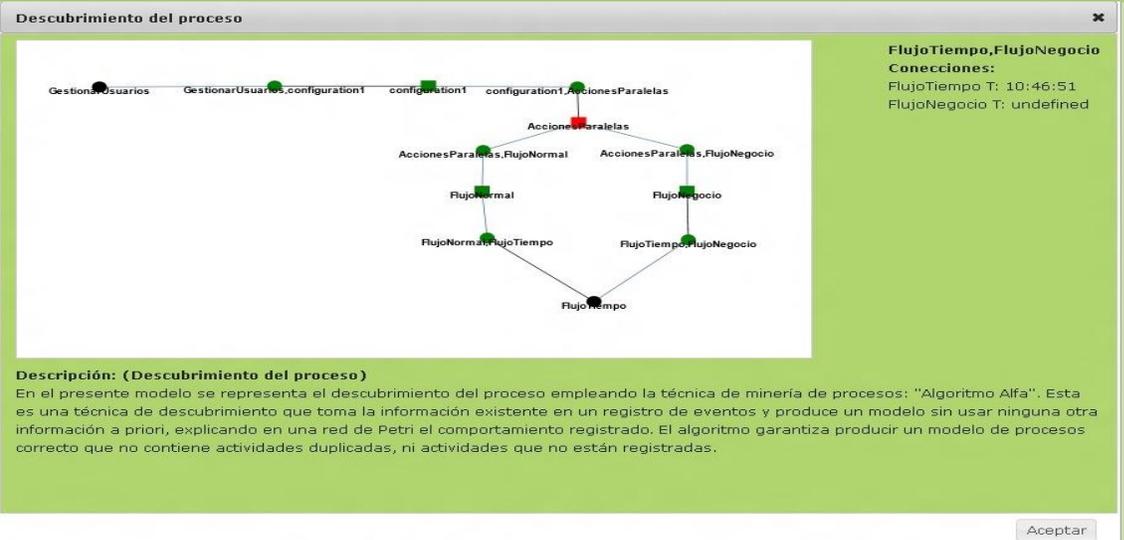
Propósito	Mostrar modelo del descubrimiento del proceso.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> Debe haberse seleccionado el proceso a descubrir. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	PlaceDiscoveryModel	adjacencies data id name
	TeamActivities	Activities Relations
Descripción	<p>5.1. Permitir al usuario seleccionar el proceso que desea descubrir.</p> <p>5.2. Mostrar opciones de descubrimiento: si selecciona Descubrimiento.</p> <p>5.3. Mostrar modelo del proceso: si selecciona Descubrir proceso.</p>	
Validaciones	-	
Prototipo	 <p>FlujoTiempo, FlujoNegocio Conecciones: FlujoTiempo T: 10:46:51 FlujoNegocio T: undefined</p> <p>Descripción: (Descubrimiento del proceso) En el presente modelo se representa el descubrimiento del proceso empleando la técnica de minería de procesos: "Algoritmo Alfa". Esta es una técnica de descubrimiento que toma la información existente en un registro de eventos y produce un modelo sin usar ninguna otra información a priori, explicando en una red de Petri el comportamiento registrado. El algoritmo garantiza producir un modelo de procesos correcto que no contiene actividades duplicadas, ni actividades que no están registradas.</p> <p>Aceptar</p>	

Tabla 13 Descripción del RF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.

RF 6 Analizar rendimiento del proceso.

Propósito	Realizar el análisis de rendimiento del proceso.	
Roles	-	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse seleccionado el proceso a analizar. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	6.1. Realizar análisis de rendimiento del proceso seleccionado.	
Validaciones	-	
Prototipo	-	

Tabla 14 Descripción del RF 6 Analizar rendimiento del proceso.

RF 7 Mostrar resultados del análisis de rendimiento.

Propósito	Mostrar resultados del análisis de rendimiento del proceso.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> Debe haberse seleccionado el proceso a descubrir. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	7.1. Mostrar los datos del análisis de rendimiento: <ol style="list-style-type: none"> Período de Gestación. Tiempo de Servicio. 	
Validaciones	-	
Prototipo		

Tabla 15 Descripción del RF 7 Mostrar resultados del análisis de rendimiento.

RF 8 Mostrar cuellos de botella.

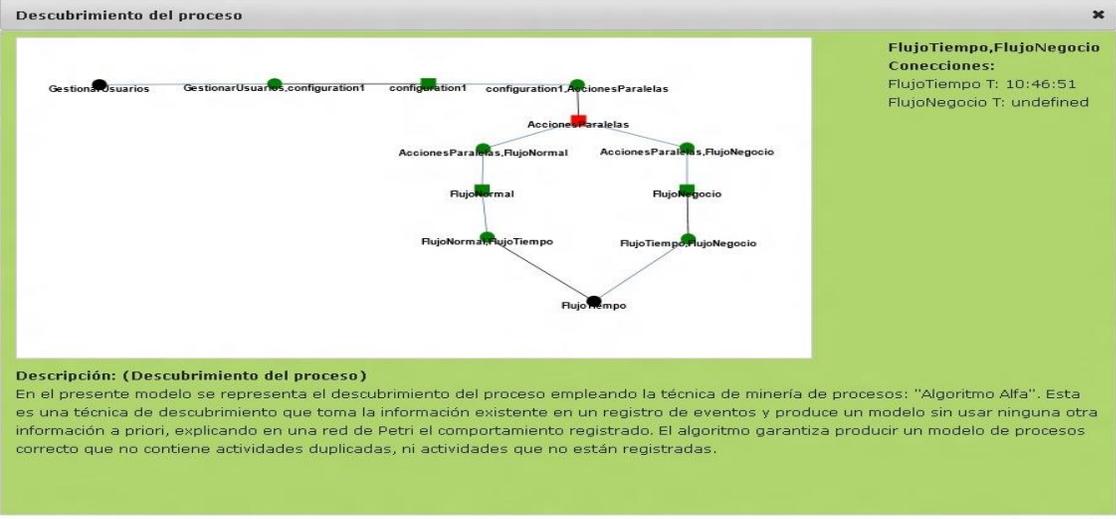
Propósito	Mostrar cuellos de botella en el proceso.	
Roles	Auditor de trazas	
Precondiciones	<ul style="list-style-type: none"> Debe haberse seleccionado el proceso a descubrir. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	8.1. Mostrar tiempo de la actividad: si se posiciona sobre una actividad. 8.2. Resaltar las actividades que más demoran (cuellos de botella).	
Validaciones	-	
Prototipo	 <p>Descripción: (Descubrimiento del proceso) En el presente modelo se representa el descubrimiento del proceso empleando la técnica de minería de procesos: "Algoritmo Alfa". Esta es una técnica de descubrimiento que toma la información existente en un registro de eventos y produce un modelo sin usar ninguna otra información a priori, explicando en una red de Petri el comportamiento registrado. El algoritmo garantiza producir un modelo de procesos correcto que no contiene actividades duplicadas, ni actividades que no están registradas.</p>	

Tabla 16 Descripción del RF 8 Mostrar cuellos de botella.

RF 9 Analizar operacionalmente el proceso.

Propósito	Realizar análisis operacional al proceso.	
Roles	-	
Precondiciones	<ul style="list-style-type: none"> • Debe haberse seleccionado el proceso a descubrir. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	9.1. Determinar el tiempo total de ejecución del proceso. 9.2. Determinar el tiempo promedio de ejecución de las instancias del proceso.	
Validaciones	-	
Prototipo		

Tabla 17 Descripción del RF 9 Analizar operacionalmente el proceso.

RF 10 Mostrar predicciones de tiempo.

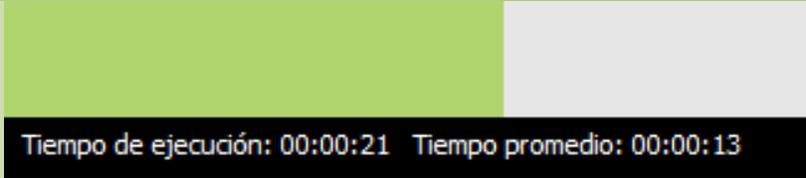
Propósito	Mostrar predicciones de tiempo de ejecución de los procesos.	
Roles	Funcionarios.	
Precondiciones	<ul style="list-style-type: none"> • Debe haber iniciado algún proceso. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
	ProcessInstance	Id AuditTrailEntry
	AuditTrailEntry	Data WorkflowModelElement EventType Originator Timestamp
Descripción	10.1. Mostrar tiempo de ejecución del proceso: si inicia la ejecución de algún proceso. 10.2. Mostrar tiempo promedio de ejecución del proceso iniciado: si inicia la ejecución de algún proceso.	
Validaciones	-	
Prototipo		

Tabla 18 Descripción del RF 10 Mostrar predicciones de tiempo.

RF 11 Generar fichero *.mxml.

Propósito	Generar fichero *.mxml con las trazas de los procesos ejecutados en un rango de tiempo.	
Roles	-	
Precondiciones	<ul style="list-style-type: none"> El usuario debe estar autenticado en el sistema. 	
Conceptos tratados	Entidad	Atributos
	Process	processId processName processInstanceList
Descripción	<p>11.1. Permitir que el usuario introduzca los datos del fichero a generar:</p> <ol style="list-style-type: none"> Nombre de fichero. Fecha inicio. Fecha fin. Dirección de destino. <p>11.2. Generar fichero *.mxml: Si selecciona Generar.</p> <p>11.3. Validar que los campos Fecha Inicio y Fecha Fin no se encuentren vacíos o estén correctamente.</p> <p>11.4. Mostrar página para generar fichero *.mxml: si selecciona Cancelar.</p>	
Validaciones	1. Los campos Fecha Inicio y Fecha Fin no deben estar vacíos.	
Prototipo		

Tabla 19 Descripción del RF 11 Generar fichero *.mxml.

Anexo 2: Pruebas unitarias

```

/// <summary>
///A test for GetAllProcess
///</summary>
[TestMethod()]
public void GetAllProcessTest()
{
    var configConnector = new XmlConfigConnector { Path = "D:\\Ivancito\\TESIS\\Implementación\\TESIS\\Modulo Ivan\\SUIIN_10\\Administracion\\"
    var target = new XmlProcessConnector(configConnector);

    List<Process> actual;
    actual = target.GetAllProcess();

    Assert.AreEqual(actual.Count, 1);
}

```

Fig. 14 Prueba unitaria aplicada al método GetAllProcess.

```

/// <summary>
///A test for GetProcessById
///</summary>
[TestMethod()]
public void GetProcessByIdTest()
{
    string id = "Carné Identidad";
    var configConnector = new XmlConfigConnector { Path = "D:\\Ivancito\\TESIS\\Implementación\\TESIS\\Modulo Ivan\\SUIIN_10\\Administracion\\"
    var target = new XmlProcessConnector(configConnector);

    List<Process> actual;
    actual = target.GetProcessById(id);
    Assert.AreEqual(actual.Count,1);
}

```

Fig. 15 Prueba unitaria aplicada al método GetProcessById.

Anexo 3: Casos de prueba

CPRF 1 Extraer datos de los sistemas fuente.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
1. Extraer datos de los sistemas fuente.	1.1 Seleccionar fuente de datos.	El sistema carga los datos desde la fuente de datos seleccionada.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.2 Seleccionar fuente de datos sin seleccionar fichero.	El sistema muestra un mensaje indicando que no ha seleccionado ningún fichero.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.3 Seleccionar fuente de datos sin seleccionar base de datos.	El sistema muestra un mensaje indicando que no ha seleccionado ninguna base de datos.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Aceptar.
	1.4 Agregar nuevo fichero *.mxml a la lista de ficheros.	El sistema agrega el nombre del fichero a la lista de ficheros *.mxml.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Agregar.
	1.5 Introducir dirección de archivo *.mxml no válida.	El sistema muestra un mensaje de error indicando que el fichero debe existir en la dirección.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Agregar.
	1.6 Introducir dirección de archivo *.mxml que ya existe en la lista.	El sistema muestra un mensaje de error indicando que el fichero ya existe en la lista.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Agregar.

	1.7 Cancelar selección realizada.	El sistema muestra nuevamente la página sin selección.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Cancelar
--	-----------------------------------	--	---

Tabla 20 Descripción del CPRF 1 Extraer datos de los sistemas fuente.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
1.4	Clases Válidas	Dirección del archivo: "D:\escritorio\file.xml"	Se agrega el archivo a la lista.	Se activa una estación de trabajo.	
1.5	Clases Inválidas	Dirección del archivo: "D:\escritorio\"	Se muestra un mensaje de error indicando que el fichero debe existir en la dirección.	Se muestra un mensaje de error indicando que el fichero debe existir en la dirección.	
1.7	Clases Inválidas	Dirección del archivo: "D:\escritorio\file.xml"	Se muestra un mensaje de error indicando que el fichero ya existe.	Se muestra un mensaje de error indicando que el fichero ya existe.	

Tabla 21 Resultado de la ejecución del CPRF 1 Extraer datos de los sistemas fuente.

CPRF 3 Buscar procesos según criterio de búsqueda.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
3. Buscar procesos según criterio de búsqueda.	3.1 Seleccionar criterio de búsqueda para datos de fichero *.mxml.	El sistema muestra el nombre del o los procesos encontrados que coinciden con el criterio de búsqueda.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Ficheros MXML. 4. Aceptar 5. Buscar
	3.2 Seleccionar criterio de búsqueda para datos de la base de datos.	El sistema muestra el nombre del o los procesos encontrados que coinciden con el criterio de búsqueda.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar usuario. 6. Aceptar. 7. Buscar
	3.3 Seleccionar criterio de búsqueda con campos requeridos vacíos para datos de fichero *.mxml.	El sistema muestra un mensaje de error indicando que el campo vacío es requerido.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Ficheros MXML. 4. Aceptar 5. Buscar
	3.4 Seleccionar criterio de búsqueda con campos requeridos vacíos para datos de la base de datos.	El sistema muestra un mensaje de error indicando que el campo vacío es requerido.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar
	3.5 Seleccionar criterio de búsqueda con la fecha Inicio Mayor que la Fecha Fin para datos de	El sistema muestra un mensaje de error indicando que la fecha inicio no puede ser mayor que la fecha fin.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Ficheros MXML. 4. Aceptar 5. Buscar

	fichero *mxml.		
	3.6 Seleccionar criterio de búsqueda con la fecha Inicio Mayor que la Fecha Fin para datos de la base de datos.	El sistema muestra un mensaje de error indicando que la fecha inicio no puede ser mayor que la fecha fin.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar.
	3.7 Seleccionar cancelar los trabajos de búsqueda.	El sistema muestra la página sin datos.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Cancelar.

Tabla 22 Descripción del CPRF 3 Buscar procesos según criterio de búsqueda.

ID del Escenario	Tipos de Clases	Clases	Resultado Esperado	Resultado de la Prueba	Observaciones
3.1	Clases Válidas	Usuario: "Adrian" Nombre del proceso: "Carné de Identidad" Fecha inicio: "1/1/2009" Fecha fin: "1/5/2012" Oficina: "Playa"	Se listan los procesos que coinciden con el criterio de búsqueda.	Se listan los procesos que coinciden con el criterio de búsqueda.	
3.3	Clases Inválidas	Usuario: "Adrian" Nombre del proceso: "Carné de Identidad" Fecha inicio: "" Fecha fin: "" Oficina: "Playa"	Se muestra un mensaje de error indicando que los campos vacíos son requeridos.	Se muestra un mensaje de error indicando que los campos vacíos son requeridos.	
3.4	Clases Inválidas	Usuario: "Adrian" Nombre del proceso: "Carné de Identidad" Fecha inicio: "" Fecha fin: "" Oficina: "Playa"	Se muestra un mensaje de error indicando que los campos vacíos son requeridos.	Se muestra un mensaje de error indicando que los campos vacíos son requeridos.	
3.5	Clases Inválidas	Usuario: "Adrian" Nombre del proceso: "Carné de Identidad" Fecha inicio: "1/5/2012" Fecha fin: "1/1/2009" Oficina: "Playa"	Se muestra un mensaje de error indicando que la fecha inicio debe ser menor que la fecha fin.	Se muestra un mensaje de error indicando que la fecha inicio debe ser menor que la fecha fin.	
3.6	Clases Inválidas	Usuario: "Adrian" Nombre del proceso: "Carné de Identidad" Fecha inicio: "1/5/2012" Fecha fin: "1/1/2009" Oficina: "Playa"	Se muestra un mensaje de error indicando que la fecha inicio debe ser menor que la fecha fin.	Se muestra un mensaje de error indicando que la fecha inicio debe ser menor que la fecha fin.	

Tabla 23 Resultado de la ejecución del CPRF 3 Buscar procesos según criterio de búsqueda.

CPRF 4 Mostrar datos de los procesos, instancias de procesos actividades.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
4. Mostrar datos de los procesos.	4.1 Seleccionar proceso de la búsqueda en fichero *.mxml.	El sistema muestra los datos del proceso seleccionado.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Ficheros MXML. 4. Aceptar 5. Buscar 6. Proceso.
	4.2 Seleccionar proceso de la búsqueda en base de datos.	El sistema muestra los datos del proceso seleccionado.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar 5. Buscar 6. Proceso.
	4.3 Seleccionar instancia de proceso de la búsqueda en fichero *.mxml.	El sistema muestra los datos de la instancia de proceso seleccionada.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Ficheros MXML. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Instancia de proceso.
	4.4 Seleccionar instancia de proceso de la búsqueda en base de datos.	El sistema muestra los datos de la instancia de proceso seleccionada.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Instancia de proceso.
	4.5 Seleccionar actividad de la instancia de la	El sistema muestra los datos de la actividad de la instancia de proceso seleccionada.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Fichero MXML.

	búsqueda en *.mxml.		<ol style="list-style-type: none"> 4. Aceptar. 5. Buscar. 6. Proceso. 7. Instancia de proceso. 8. Actividad.
	4.6 Seleccionar actividad de la instancia de la búsqueda en base de datos.	El sistema muestra los datos de la actividad de la instancia de proceso seleccionada.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Instancia de proceso. 8. Actividad.

Tabla 24 Descripción del CPRF 4 Mostrar datos de los procesos, instancias de procesos actividades.

CPRF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
5. Mostrar modelo del descubrimiento del proceso.	5.1 Seleccionar proceso de la búsqueda en fichero *.mxml.	El sistema muestra el modelo de descubrimiento del proceso seleccionado.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Fichero MXML. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.
	5.2 Seleccionar proceso de la búsqueda en la base de datos.	El sistema muestra el modelo de descubrimiento del proceso seleccionado.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso.

			<ol style="list-style-type: none"> 7. Descubrimiento. 8. Descubrir proceso.
	5.3 Seleccionar descubrir proceso sin seleccionar proceso.	El sistema muestra un mensaje de error indicando que debe seleccionar el proceso a descubrir.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.

Tabla 25 Descripción del CPRF 5 Mostrar modelo del descubrimiento del proceso usando el algoritmo alfa.

CPRF 7 Mostrar resultados del análisis de rendimiento.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
7. Mostrar resultado del análisis de rendimiento.	7.1 Seleccionar actividad del modelo descubierto de los datos del fichero *.mxml.	El sistema muestra los tiempos de las actividades.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Fichero MXML. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.
	7.2 Seleccionar actividad del modelo descubierto de los datos de la base de datos.	El sistema muestra los tiempos de las actividades.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.

Tabla 26 Descripción del CPRF 7 Mostrar resultados del análisis de rendimiento.

CPRF 8 Mostrar cuellos de botella.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
8. Mostrar cuellos de botella.	8.1 Descubrir proceso seleccionado de la búsqueda en fichero *.mxml.	El sistema resalta en el modelo las actividades con mayor tiempo.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Fichero MXML. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.
	8.2 Descubrir proceso seleccionado de la búsqueda en la base de datos.	El sistema resalta en el modelo las actividades con mayor tiempo.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Proceso. 7. Descubrimiento. 8. Descubrir proceso.

*Tabla 27 Descripción del CPRF 8 Mostrar cuellos de botella.***CPRF 10 Mostrar predicciones de tiempo.**

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
10. Predicciones de tiempo.	10.1 Seleccionar proceso de la búsqueda en la base de datos.	El sistema muestra el tiempo restante aproximado para que culmine el proceso.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Base de datos. 4. Aceptar. 5. Buscar. 6. Ver estado.

Tabla 28 Descripción del CPRF 10 Mostrar predicciones de tiempo.

CPRF 11 Generar fichero *.mxml.

Nombre del Flujo	Escenarios del flujo	Descripción de la funcionalidad	Flujo central
11. Generar *.mxml.	11.1 Entrar los datos del fichero *.mxml a generar.	El sistema muestra un mensaje que notifica que se ha generado el fichero correctamente.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Generar fichero MXML. 4. Generar.
	11.2 Entrar los datos del fichero *.mxml a generar con campos requeridos vacíos.	El sistema muestra un mensaje de error indicando que el campo vacío es requerido.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Generar fichero MXML. 4. Generar.
	11.3 Seleccionar cancelar la generación del fichero *.mxml.	El sistema muestra la página sin datos.	<ol style="list-style-type: none"> 1. Minería de procesos. 2. Minar proceso. 3. Generar fichero MXML. 4. Cancelar.

Tabla 29 Descripción del CPRF 11 Generar fichero *.mxml

Anexo 4: Prototipos de interfaz del módulo

Selección de la fuente de datos

Ficheros MXML Recargar

Archivo MXML: test.xml
Archivo MXML: FicheroPrueba.xml
Archivo MXML: prueba.xml
Archivo MXML: L5.xml

Dirección del archivo:
 Agregar

Base de datos Recargar

Base de Datos: SUIN DB
Base de Datos: SIEC DB
Base de Datos: UCI DB
Base de Datos: CENTRAL DB

Aceptar Cancelar

Fig. 16 Prototipo de interfaz para la selección de la fuente de datos.

En la Fig. 16 se muestra el prototipo de interfaz definido para la selección de la fuente de datos. En el mismo, se le da la posibilidad de seleccionar desde dónde desea cargar los datos de los procesos. En el caso de los ficheros MXML, son listados los ficheros con los que ya se han trabajado anteriormente. Brinda además la posibilidad de agregar otros ficheros proporcionando la ruta física. En el otro caso, son listadas las bases de datos disponibles para cargar los datos.

Fichero

Nombre de fichero:

Seleccionar usuario

Usuario:

Buscar proceso

Nombre del proceso: Fecha inicio: Fecha fin: Oficina:

Minería de proceso	
<input type="radio"/>	Descubrimiento
<input type="radio"/>	Chequeos de conformidad
<input type="radio"/>	Extensión

Datos de elemento seleccionado

Nombre: Carné de Identidad
 Actividades: 6
 Fecha inicio: 10/25/2011 10:46:50
 Fecha fin: 10/25/2011 10:46:51
 Período de gestación: 00:03:01
 Tiempo de servicio: 00:01:10

> Carné de Identidad > asdasd-dasd2d-d2dasd-2d >

Fig. 17 Prototipo de interfaz para buscar procesos desde ficheros MXML.

En la Fig. 17 se muestra la interfaz definida para realizar la búsqueda de los procesos desde un fichero *.mxml. En ella al usuario se le muestra la ubicación del fichero con el que está trabajando y se le muestran los campos para realizar la búsqueda de los procesos. Posee además el área donde se mostrarán los resultados de la búsqueda, las operaciones de minería a realizar y el área para mostrar datos de los procesos, instancias y actividades.



Fig. 18 Prototipo de interfaz para mostrar datos de procesos, instancias de proceso y actividades seleccionadas.

En la Fig. 18 se muestra la interfaz en la que se mostrarán los datos de los procesos, instancias de procesos y actividades seleccionadas. En la sección datos de elemento seleccionado se le mostrarán al usuario datos generales y específicos de cualquiera de los elementos que seleccione; dígame un proceso, instancia o actividad.

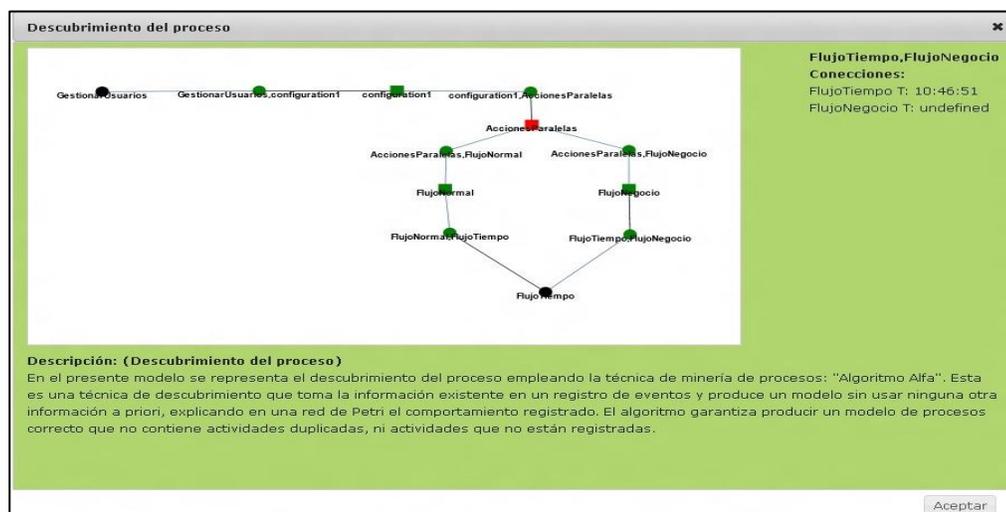


Fig. 19 Prototipo de interfaz para mostrar el modelo de descubrimiento del proceso seleccionado.

En la Fig. 19 se muestra la interfaz definida para mostrar el modelo de descubrimiento del proceso. En ella al usuario se le muestra el modelo del proceso y una breve descripción del descubrimiento usando el algoritmo alfa.



Fig. 20 Prototipo de interfaz para mostrar los resultados del análisis de rendimiento.

En la Fig. 20 se muestra la interfaz definida para mostrar los resultados del análisis de rendimiento. Cuando se trate de la selección de una instancia de proceso, en la sección de datos de elemento seleccionado se le mostrarán el período de gestación y el tiempo de servicio de la instancia seleccionada.

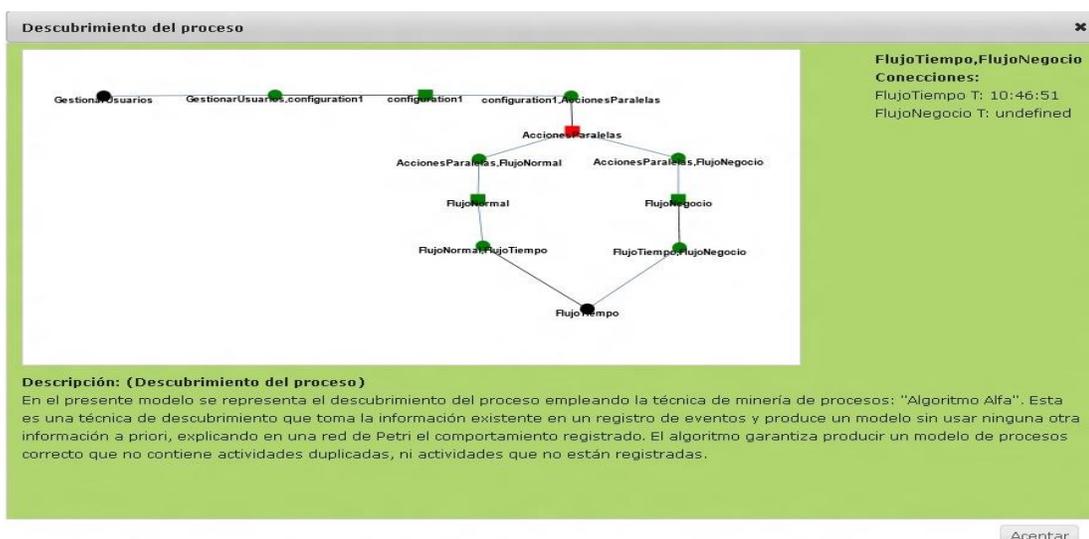


Fig. 21 Prototipo de interfaz para mostrar cuellos de botella.

En la *Fig. 21* se muestra la interfaz definida para mostrar los cuellos de botella. En el propio modelo del descubrimiento del proceso serán resaltadas en color rojo las transiciones (actividades) que mayor tiempo presenten, determinando así, los cuellos de botella.

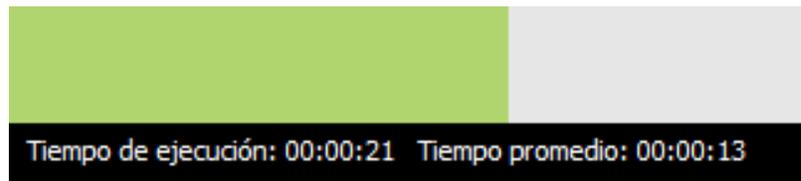


Fig. 22 Prototipo de interfaz para mostrar predicciones de tiempo.

En la *Fig. 22* se muestra la interfaz definida para mostrar las predicciones de tiempo. Desde el momento en que el usuario inicie un proceso, en la parte inferior izquierda se mostrará el tiempo de ejecución del mismo y el tiempo promedio de ejecución, teniendo en cuenta.

Anexo 5: Diagrama de clases del diseño

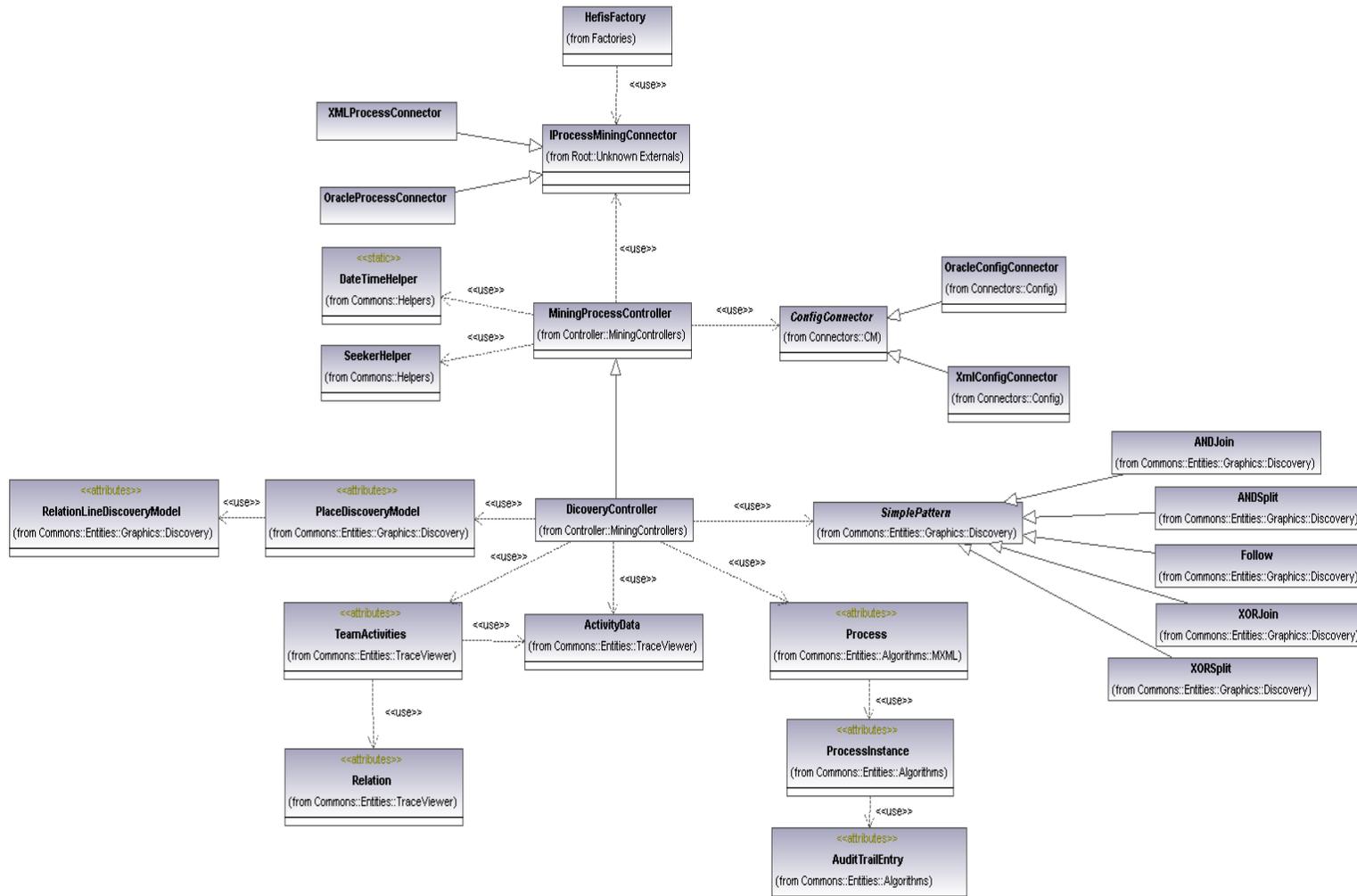


Fig. 23 Diagrama de clases del diseño extendido.

Anexo 6: Modelo de datos

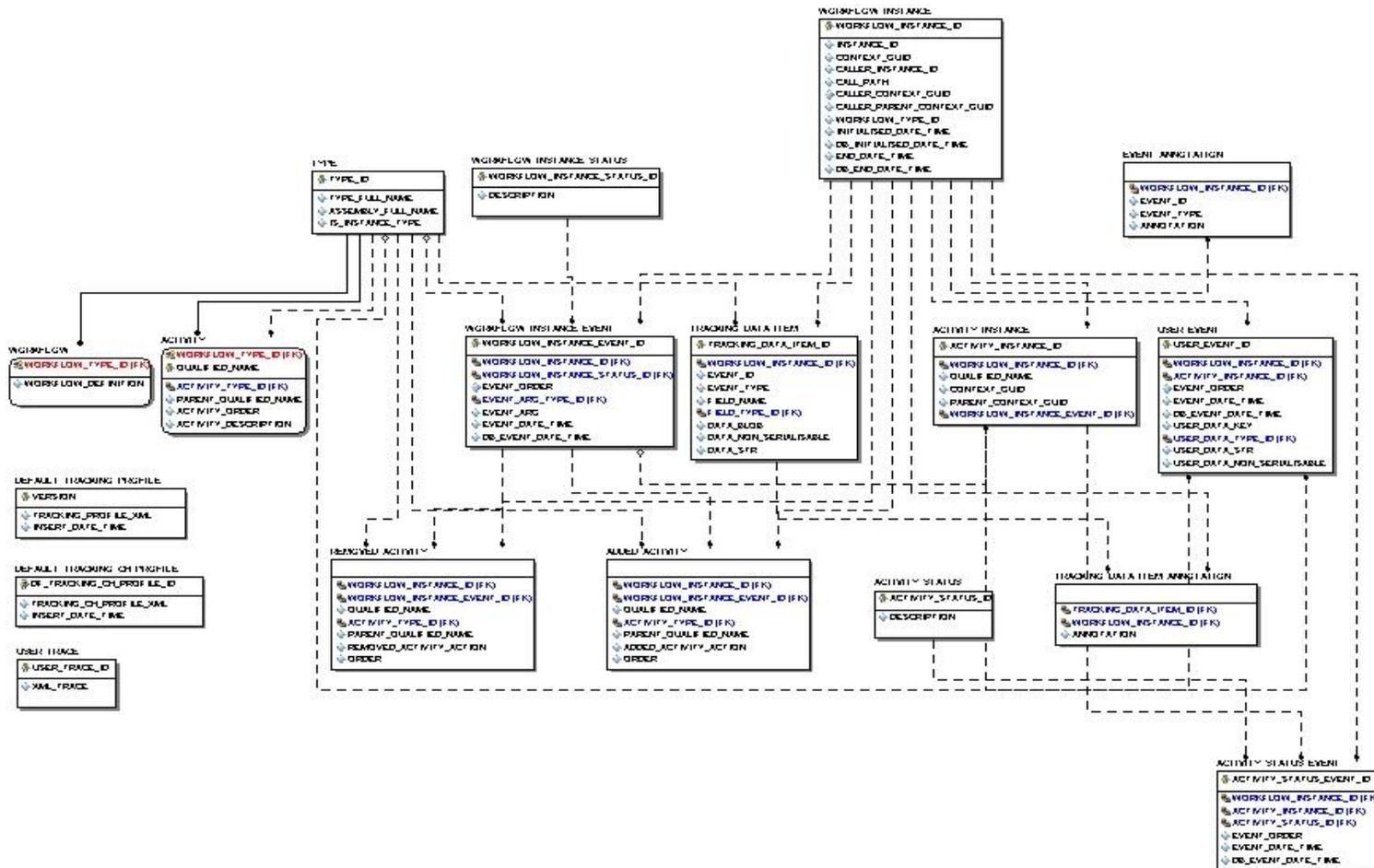


Fig. 24 Modelo de datos extendido.