

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Departamento: Base de Datos

TRABAJO DE DIPLOMA

SAD_BPMN v3.0. Herramienta para la evaluación de modelos de procesos de negocio

Autor: Luis Alberto Jiménez Martínez

Tutores del trabajo: Dr. C. Isel Moreno Montes de Oca

MSc. Frank Reyes García

Santa Clara, junio 2018
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419

Declaración Jurada



El que suscribe, **Luis Alberto Jiménez Martínez**, hace constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por la Institución para los fines que estime conveniente, tanto de forma parcial como total y que, además, no podrá ser presentado en eventos, ni publicado sin autorización de la Universidad.

Luis Alberto Jiménez Martínez

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe del Laboratorio

Pensamiento

*“Porque nadie puede saber por ti. Nadie puede crecer por ti. Nadie puede buscar por ti.
Nadie puede hacer por ti lo que tú mismo debes hacer. La existencia no admite
representantes”*

Jorge Bucay

Dedicatoria

A mi abuela Magdita y a mi bisabuela Magda por sus consejos, amor y exigencia.

A mis abuelos por parte padre, por estar presentes en mí, aún sin conocerles.

Agradecimientos

A mis padres, por ser fuente firme y constante de inspiración y apoyo, por su preocupación infinita.

A mi tía Laura y a mi primo Jesús Manuel por darme un amigo incondicional, un tesoro que poseo, el recurso necesario para atravesar estos cinco años: mi laptop.

A mi familia que son mi bastión y soporte, mi guía y mi luz, y por toda la fuerza que me brindan.

A mi novia su enorme paciencia, su cariño y dulzura y por mejorarme como persona.

A mis tutores por su dedicación, enseñanzas y por todas las molestias que les ocasioné.

A todos los que de una forma u otra contribuyeron a este trabajo.

A mis compañeros del aula por acompañarme en este largo viaje de cinco años.

Resumen

Cada día gana mayor importancia el modelado de procesos de negocio para automatizar y documentar procesos de negocio, y a su vez, lograr mayor eficiencia en las organizaciones. SAD_BPMN es una herramienta que permite evaluar modelos de procesos de negocio BPMN tomando como base medidas de calidad y sus umbrales, y luego sugiere directrices de acuerdo al resultado de sus operaciones. Existe la necesidad de perfeccionar SAD_BPMN para brindar mayor usabilidad y soporte de dicha herramienta, cuya última versión es la 2.0. Como resultado del proceso de investigación científica se identifica el desarrollo de nuevas directrices para brindar mayor usabilidad a SAD_BPMN y el perfeccionamiento de las directrices de calidad. Los principales resultados se relacionan con la implementación de nuevas directrices de calidad para los modelos creados; el desarrollo de una nueva versión de SAD_BPMN integrando todas las mejoras realizadas.

Palabras Clave: directrices de calidad, modelado de procesos de negocio

Abstract

Every day it wins bigger importance the modeling of business processes to automate and to document business processes, and in turn, to achieve bigger efficiency in the organizations. SAD_BPMN is a tool that allows to evaluate models of business processes BPMN taking like measured base of quality and its thresholds, and then suggests guidelines according to the result of its operations. Exists the necessity of perfecting SAD_BPMN for bigger usability and support of this tool whose last version is the to toast 2.0. As results of the process of scientific investigation is identified the development of new guidelines to offer bigger usability to SAD_BPMN and the improvement of the guidelines of quality. The main results are related with implementation of new guidelines of quality for the created models; the development of a new version of SAD_BPMN integrating all the carried out improvements.

Keywords: guidelines of quality, business process modeling

Tabla de Contenidos

Introducción	1
Capítulo I. Modelación de procesos de negocio y su calidad	5
1.1 Proceso de negocio	5
1.2 Gestión de procesos de negocio	6
1.3 Modelado de procesos de negocios	8
1.4 Lenguajes para la modelación de procesos de negocio	9
1.4.1 BPMN	10
1.5 Calidad de los modelos de proceso de negocio	15
1.5.1 Herramienta para la evaluación de modelos BPMN	18
1.6 Grafos	20
1.7 Fundamentación del entorno de desarrollo	20
1.8 Conclusiones parciales	22
Capítulo II. Modelo del Negocio y Requisitos	23
2.1 SAD_BPMN v2.0. Análisis de soporte de directrices de calidad	23
2.2 Diagrama de casos de uso del negocio	25
2.3 Actores del negocio y sistema	25
2.4 Definición de los requisitos funcionales y no funcionales	26
2.4.1 Requisitos funcionales	26
2.4.2 Requisitos no funcionales	27
2.5 Diagrama de paquetes y sus relaciones	27
2.6 Diagrama de casos de uso del sistema	28
2.7 Descripción de los casos de uso del sistema	29
2.8 Conclusiones parciales	39

Capítulo III. Descripción de la propuesta de solución	40
3.1 Arquitectura del sistema	40
3.2 Implementación de las directrices de calidad.....	41
3.3 Diagrama de clases del diseño.....	45
3.4 Diagrama de secuencia	47
3.5 Modelo de componentes	49
3.6 Diagrama de despliegue.....	50
3.7 Tratamiento de errores	50
3.8 Conclusiones parciales.....	52
Capítulo IV. Pruebas y análisis de factibilidad.....	53
5.1 Planificación y estimación	53
5.1.1 Planificación basada en uno de los métodos de estimación	53
5.1.2 Estimación de costes mediante COCCOMO I.....	59
5.2 Pruebas unitarias	62
5.2.1 Casos de prueba	63
5.3 Pruebas de integración	67
5.4 Pruebas de caja negra. Diseño de casos de prueba mediante RUP.....	70
5.4.1 Secciones a probar en el Caso de Uso Mostrar modelo de proceso de negocio 70	
5.4.2 Secciones a probar en el Caso de Uso Mostrar sugerencias visuales.....	71
5.5 Conclusiones parciales.....	73
Conclusiones.....	74
Recomendaciones.....	75
Referencias bibliográficas.....	76
Anexos.....	81

Anexo 1. BPMN	81
Anexo 2. Grafos	85
Anexo 3. Análisis de las herramientas	87
Anexo 4. Análisis comparativo de las “directrices obsoletas” con respecto a las “directrices nuevas”	90
Anexo 5. Diagrama de paquetes: auxiliar	92
Anexo 6. Diagrama de paquetes: bpmn_data	92
Anexo 7. Diagrama de paquetes: directrices.....	92
Anexo 8. Diagrama de paquetes: grafo.....	93
Anexo 9. Diagrama de paquetes: ManagementFile	93
Anexo 10. Diagrama de paquetes: TDA	93
Anexo 11. Diagrama de paquetes: usingtreetagger	94
Anexo 12. Diagrama de paquetes: visual.....	94
Anexo 13. Caso de prueba para sección Cargar modelo de proceso de negocio.....	95
Anexo 14. Caso de prueba para sección Calcular métricas de calidad	97
Anexo 15. Caso de prueba para sección Exportar datos	99

Lista de Figura

Figura 1.1 Ciclo de vida BPM (Fuente: (Dumas et al., 2013)).	7
Figura 1.2 Grado de soporte de las directrices	14
Figura 1.3 Grado de soporte de las directrices	14
Figura 1.4 Taxonomía para directrices de calidad de modelado de procesos de negocio (Fuente: (Moreno Montes de Oca 2015))	16
Figura 2.1 Soporte de las métricas en SAD_BPMN v2.0	24
Figura 2.2 Comparación entre las directrices “nuevas” y “actuales”	24
Figura 2.3 Soporte de SAD_BPMN v2.0 para las directrices “nuevas” y “actuales”	25
Figura 2.4 Diagrama de casos de uso del negocio.	25
Figura 2.5 Diagrama de paquetes y sus relaciones	28
Figura 2.6 Diagrama de casos de uso del sistema	29
Figura 3.1 Patrón MVC (Fuente: (Robles 2013))	40
Figura 3.2 Diagrama de clases del diseño CUS Exportar datos	46
Figura 3.3 Diagrama de clases del diseño CUS Mostrar modelo de negocio	46
Figura 3.4 Diagrama de secuencia CUS Exportar datos	47
Figura 3.5 Diagrama de secuencia CUS Mostrar sugerencias visuales.	47
Figura 3.6 Diagrama de secuencia CUS Cargar modelo	48
Figura 3.7 Diagrama de secuencia CUS Mostrar modelo de negocio.	48
Figura 3.8 Diagrama de secuencia CUS Calcular métricas de calidad	49
Figura 3.9 Modelo de componentes SAD_BPMN v3.0	49
Figura 3.10 Diagrama de despliegue SAD_BPMN v3.0	50
Figura 3.11 Notificación de error al intentar cargar un fichero de extensión incorrecta.	50
Figura 3.12 Notificación de error al intentar cargar un modelo XML con formato erróneo	50
Figura 3.13 Notificación de error al intentar cargar un modelo de extensión correcta (.bpmn) pero generado por una herramienta no soportada.	51
Figura 3.14 Notificación de atención generada porque se desea evaluar un modelo y no se ha seleccionado al menos una métrica para ello.	51
Figura 3.15 Notificación de atención generada porque se desea ver sugerencias visuales para el modelo y no se ha seleccionado al menos una métrica para ello.	51

Figura 3.16 Notificación de error al intentar cargar un modelo de extensión correcta (.xpdL) pero generado por una herramienta no soportada.....	51
Figura 4.2 Método diseñado para la prueba de la operación “Flujos de longitud uniforme”	64
Figura 4.3 Resultado satisfactorio de la prueba de la operación “Flujos de longitud uniforme”	64
Figura 4.4 Método modificado de prueba de la operación “Flujos de longitud uniforme” ..	65
Figura 4.5 Resultado satisfactorio de la prueba de la operación “Flujos de longitud uniforme” modificada	65
Figura 4.6 Método diseñado para la prueba de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades.....	66
Figura 4.7 Resultado satisfactorio de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades.....	66
Figura 4.8 Método modificado diseñado para la prueba de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades.....	66
Figura 4.9 Resultado satisfactorio de la operación modificada del cálculo de la razón entre actividades sin etiquetar y el total de actividades.....	67
Figura 4.10 Modelo de proceso de negocio de ejemplo cargado	68
Figura 4.11 Modelo de proceso de negocio de ejemplo evaluado.....	69
Figura 4.12 Visualización del modelo de proceso de negocio cargado.....	69

Lista de Tablas

Tabla 1.1 Clasificación taxonómica de las directrices de calidad para modelos de procesos de negocio (<i>Fuente:</i> (Moreno Montes de Oca 2015)).	17
Tabla 2.1 Actor del sistema SAD_BPMN v2.0.	26
Tabla 2.2 Resumen de los casos de uso del sistema	29
Tabla 2.3 Descripción del caso de uso <i>Cargar modelo de proceso de negocio</i>	30
Tabla 2.4 Descripción del caso de uso <i>Mostrar modelo de proceso de negocio</i>	32
Tabla 2.5 Descripción del caso de uso <i>Exportar datos</i>	34
Tabla 2.6 Descripción del caso de uso <i>Calcular métricas de calidad</i>	37
Tabla 4.1 Factores de peso de los actores	54
Tabla 4.2 Factores de pesos de los casos de uso	55
Tabla 4.3 Factor de complejidad técnica	56
Tabla 4.4 Factor de ambiente	57
Tabla 4.5 Distribución genérica del esfuerzo	58
Tabla 4.6 Distribución real del esfuerzo	58
Tabla 4.7 Conductores de coste	60
Tabla 4.8 Caso de prueba para sección <i>Mostrar modelo de proceso de negocio</i>	70
Tabla 4.9 Escenario a probar para sección <i>Mostrar modelo de proceso de negocio</i>	70
Tabla 4.10 Caso de prueba para sección <i>Mostrar sugerencias visuales</i>	71
Tabla 4.11 Escenario a para sección <i>Cargar modelo de proceso de negocio</i>	72
Tabla 4.12 Escenario a para sección <i>Cargar modelo de proceso de negocio</i>	73

Introducción

La mejora de los procesos corporativos ha sido considerada como una prioridad superior de los directores ejecutivos (CEO) constantemente durante los últimos años. Las organizaciones están cada vez más interesadas en el entendimiento, manejo y mejora de sus procesos portafolios, y en identificar y cuantificar con subcontrata potencial. La colección de herramientas y métodos para alcanzar éstos objetivos se refiere a la Gestión de Procesos de Negocios (BPM) (zur Muehlen & Indulska 2010).

BPM se basa en la observación de que cada producto que una compañía provee al mercado es el resultado de un número de actividades realizadas. Los procesos de negocios son el instrumento clave para organizar esas actividades y mejorar el entendimiento de sus interrelaciones. BPM es influenciada por conceptos y tecnologías de diferentes áreas de administración de negocio y ciencia de la computación. Basado en un reciente trabajo en organización y gestión, BPM tiene sus raíces en la tendencia de orientación de proceso de los 90, de la que emerge una nueva manera de organizar las compañías en la base de procesos de negocio (Weske 2007).

El modelado de procesos de negocio es la fundación de la gestión de procesos de negociación (BPM). Existe un gran número de enfoques de modelación de procesos de negocio en el dominio BPM (He et al. 2010). Muchas técnicas de modelado de procesos de negocios han sido propuestas en las últimas décadas, creando así una demanda por teoría para ayudar en la comparación y evaluación de dichas técnicas (Recker et al. 2009). Un modelo de proceso de negocio consiste en un conjunto de modelos de actividad y la restricción de ejecución entre éstas, según criterio de Weske (2007). Para (Becker et al. 2000), la modelación de procesos se vuelve cada vez más una tarea más importante, no sólo para el propósito de la ingeniería de software, sino también para muchos otros propósitos, además del desarrollo de software. Es por ello que resulta necesario evaluar la calidad de los modelos desde diferentes puntos de vista.

La calidad de un modelo de proceso de negocio (BPM) influencia fuertemente el proceso de negocio desplegado. Esto ha generado diversas investigaciones para proponer métricas para evaluar la calidad de BPM. De hecho, el concepto de métricas de calidad fue inicialmente introducido para examinar la calidad de software (Khelif et al. 2010).

Existen diferentes enfoques para incrementar la calidad de los modelos de procesos de negocio que pueden ser beneficiosos. Con el objetivo de ofrecer una guía práctica a los modeladores, varios estudios han propuesto directrices de calidad que persiguen la introducción directa en la práctica de resultados de investigaciones experimentales para la mejora de la calidad de los modelos de procesos de negocio (Reyes García 2016).

En el laboratorio de base de datos perteneciente al Centro de Investigaciones Informáticas (CII), ubicado la Universidad Central “Marta Abreu” de Las Villas, se realizan estudios vinculados con la calidad y la modelación de procesos de negocios. En este centro se desarrolló SAD_BPMN, (última versión: 2.0): una herramienta basada en Java, la que permite la evaluación de diversos modelos de procesos de negocio mediante la consideración de diferentes medidas de calidad y umbrales.

Se han realizado diferentes versiones de SAD_BPMN, como resultado de aportes realizados por (Reyes García 2016), (Calzada Ricet & Pairol Fuentes 2016) y (Hernández Cárdenas 2016). A pesar del trabajo constante y arduo realizado en SAD_BPMN, aún existen diferentes errores en el soporte de algunas directrices en los modelos, expresados en la limitación en cuanto a la presentación de la evaluación de los modelos y a la carencia de componentes visuales que faciliten la interpretación de la evaluación de las directrices de calidad.

El **problema de investigación** se enmarca en la necesidad de perfeccionar SAD_BPMN para brindar mayor soporte a las directrices de calidad y visualizar los modelos a evaluar.

De ello se deriva el planteamiento del **objetivo general**:

Extender la herramienta SAD_BPMN v2.0, utilizando teoría de grafos para evaluar la calidad de modelos de procesos de negocio generados por herramientas de modelación, que permita ofrecer la visualización del modelo.

Para dar cumplimiento a este objetivo se plantean los siguientes **objetivos específicos**:

1. Evaluar el soporte que ofrece SAD_BPMN al conjunto de directrices de calidad.
2. Incluir en la herramienta la visualización del modelo a evaluar.
3. Implementar la evaluación de nuevas directrices de calidad para modelos creados.
4. Validar el software mediante pruebas de caja negra y caja blanca.
5. Integrar en una nueva versión todas las mejoras realizadas para lograr el completamiento de la herramienta SAD_BPMN.

Para transitar por el proceso investigativo se asumieron las siguientes **preguntas de investigación**:

1. ¿Qué directrices de calidad no se verifican correctamente por SAD_BPMN v2.0?
2. ¿Cómo incluir en la herramienta la visualización del modelo a evaluar?
3. ¿Cómo redefinir las nuevas directrices de calidad para SAD_BPMN?
4. ¿Cómo se puede validar el software?
5. ¿Cómo integrar todas las mejoras realizadas en una versión 3.0 de SAD_BPMN para lograr la evaluación amena de los modelos de procesos de negocio por parte de los usuarios?

Justificación

El desarrollo de la herramienta SAD_BPMN v2.0 en el laboratorio de base de datos de la Universidad Central Marta Abreu de Las Villas, perteneciente al Centro de Investigaciones Informáticas (CII), constituye un antecedente de la presente investigación. La necesidad de implementar nuevas directrices de calidad y de adicionar nuevos componentes gráficos para representar el análisis del modelo, encuentra en la aplicación diseñada una solución científica y práctica que garantiza mayor soporte y visualización de directrices y modelos, respectivamente.

Estructura de la tesis

La tesis está estructurada en cuatro capítulos. En el primer capítulo se definirá la fundamentación teórica necesaria para entender el presente trabajo. En él se describieron conceptos, definiciones, lenguajes y herramientas. En el capítulo II se exponen el modelo del negocio actual, las reglas del negocio a considerar, los actores del negocio, diagramas de casos de uso del negocio y del sistema, trabajadores del negocio, casos de uso del negocio, actores del sistema a automatizar, definición de los requisitos funcionales y no funcionales y los paquetes y sus relaciones. En el tercer capítulo se presentan algunos aspectos de la descripción de SAD_BPMN v3.0 (arquitectura de sistema, tratamiento de errores, etc.) así como el diagrama de paquetes y clases de diseño necesarios para la implementación de las nuevas funcionalidades. En el capítulo IV se detallan todas las pruebas realizadas a la

aplicación y la visualización de los resultados de las mismas. Además de realizó la estimación de costes y la planificación basada en puntos de casos de uso.

Capítulo I. Modelación de procesos de negocio y su calidad

En el presente capítulo se expone la fundamentación teórica necesaria para el entendimiento y desarrollo de la investigación. Se parte de la definición del concepto básico: proceso de negocio, seguidamente se trata el tema de modelado de procesos de negocio, para luego abordar la descripción, lenguajes y herramientas para la representación de dichos lenguajes. Además, se analizan las directrices de calidad para la calidad de modelo de procesos de negocio, destacando su relevancia y presencia en investigaciones precedentes.

1.1 Proceso de negocio

En una organización se maneja cierta cantidad de procesos. Para los procesos de negocio se han realizado diversas definiciones. Según (Smirnov et al. 2012): “un proceso de negocio moderno es un sistema intrínsecamente distribuido: sus actividades son realizadas por varios empleados, en diferentes lugares, usando un conjunto heterogéneo de sistemas de tecnología de la información. Además, un proceso de negocio típicamente cruza los bordes de departamentos organizativos e incluso, compañías.” Un proceso de negocio es definido en (Aguilar-Savén 2004) como la “combinación de un conjunto de actividades dentro de una empresa con una estructura describiendo su orden lógico y dependencia cuyo objetivo es producir un resultado deseado”. Informalmente, por un proceso de negocio se entiende el proceso que es llevado a cabo en una organización en orden para obtener los objetivos de la compañía de negocios (Badica & Fox 2002).

Según criterio de (Rosemann et al. 2008), la flexibilidad de un proceso de negocio es directamente proporcional a la adaptabilidad de la organización a su entorno. Cada proceso de negocio es promulgado por una sola organización, pero él puede interactuar con procesos de negocio realizados por otras organizaciones (Weske 2007). (Lee 1998) describe ciertas características que tiene todo proceso de negocio: “son generalmente cruz-funcionales, de naturaleza horizontal, tendido hacia el vertical usual, estructura jerárquica de compañía, y donde una sola persona no puede tener la responsabilidad del proceso entero.” Los procesos de negocio representan un punto principal de las corporaciones. Éstos tienen impacto directo en lo atractivo que productos y servicios son percibidos por el mercado. Ellos determinan actividades, trabajos y responsabilidades y por esto, forma el trabajo de cada empleado (Dumas et al. 2013).

Un proceso de negocio representa la forma de trabajar las organizaciones, horizontalmente, permitiendo un análisis del flujo de trabajo a través de límites externos e internos (Castela et al. 2002). En (Dumas et al. 2013) se define a un proceso de negocio como “colección de interrelacionados eventos, actividades y puntos de decisión que involucra un número de actores y objetos, y esto colectivamente resulte en un resultado que es valioso a al menos un cliente”. En (Reyes García 2016) se concluye que: “un proceso de negocio es una forma de organizar el trabajo y los recursos para alcanzar objetivos determinados en las organizaciones, a través de un conjunto de actividades que se llevan a cabo en determinado orden.”

1.2 Gestión de procesos de negocio

Actualmente muchas organizaciones se centran, en una manera de proceso orientado, manejar sus negocios. Esta idea es tomada en el concepto de BPM. Las organizaciones apuestan por mejorar el rendimiento de sus negocios, aplicando métodos de BPM (Schrepfer 2010).

Para (Lee 1998), BPM es el medio mediante el cual la organización corporativa aspira a mejorar sus procesos en orden para obtener sus objetivos corporativos. BPM es, a escala total, una manera de gestión que trata sobre lograr la integración de múltiples negocios (Liu & Hou 2010). BPM es el arte y la ciencia de supervisar cómo el trabajo es realizado en una organización para asegurar consistentes resultados y para tomar ventaja de oportunidades de mejora. Importante, BPM no es acerca de mejorar la forma en que las actividades individuales son realizadas. Más bien, es acerca de gestionar cadenas enteras de eventos, actividades y decisiones que en última instancia añaden valor a la organización y sus clientes (Dumas et al. 2013).

BPM también abarca los bienes de las organizaciones como humanos, aplicaciones, documentos, unidades organizacionales, así como otras fuentes de información o sistemas (Schrepfer 2010).

BPM es un campo especial por más de una razón. Primero que todo, BPM es un cruce de múltiples y totalmente diferentes puntos de vista. Directores de negocio están atraídos hacia BPM a causa de su demostrada habilidad para aportar mejoras en el rendimiento organizativo, conformidad regulatoria y calidad del servicio. BPM provee un lenguaje

compartido comunicarse con *stakeholders* de negocio (Dumas et al. 2013). BPM incluye conceptos, métodos y técnicas para soportar el diseño, administración, configuración, promulgación, y análisis de procesos de negocio (Weske 2007).

En (Dumas et al. 2013) se define a BPM como “un conjunto de métodos, técnicos y herramientas para descubrir, analizar, rediseñar, ejecutar y monitorear procesos de negocio”.

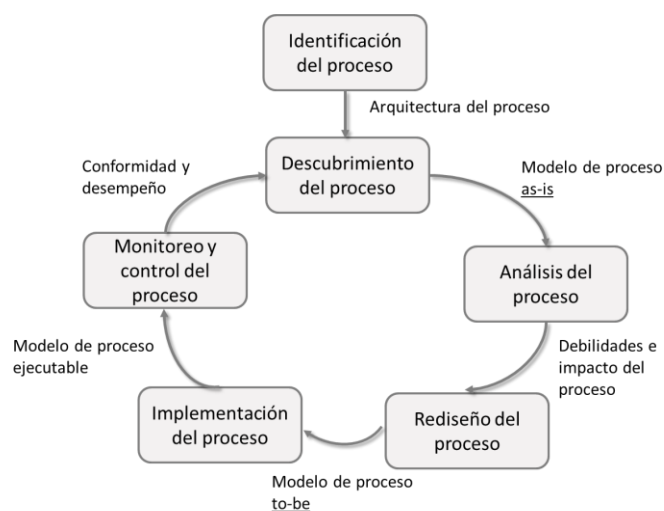


Figura 1.1 Ciclo de vida BPM (Fuente: (Dumas et al., 2013)).

El ciclo de vida BPM (véase figura 1.1) ayuda a entender el rol de la tecnología en BPM. La tecnología en general, y especialmente Tecnología de la Información (IT), es un instrumento clave para mejorar los procesos de negocio (Dumas et al. 2013).

En la etapa de *identificación del proceso* se plantea un problema y seguidamente se identifican, delimitan y relacionan entre sí los procesos relevantes a este problema. La salida de la identificación del proceso es una arquitectura nueva o actualizada que ofrece una visión general de los procesos de la organización y sus interrelaciones.

En la etapa de *descubrimiento del proceso as-is* se documenta el estado actual de los procesos relevantes. En la etapa de *análisis del proceso* se identifican, documentan y miden cuantitativamente los procesos *as-is* usando métricas de desempeño. La fase de *rediseño del proceso* determina los cambios a realizar al mismo que ayudarán a corregir los aspectos que se identificaron en la etapa previa. Con este fin, las opciones de cambios múltiples se analizan y se comparan en términos de las métricas de desempeño seleccionadas. Esto supone que el rediseño del proceso y el análisis del proceso estén estrechamente relacionados; tan pronto como se propongan nuevas opciones de cambios, se analizan utilizando distintas técnicas de

análisis de procesos. Eventualmente, se combinan las opciones de cambio más prometedoras, dando lugar al modelo de proceso rediseñado. La salida de esta fase es el modelo deseado (*to-be*), que sirve de base a la etapa siguiente.

En la etapa de *implementación del proceso* se preparan y desarrollan los cambios requeridos para ir del proceso *as-is* hacia el proceso *to-be*. Esta etapa involucra dos aspectos: la gestión de cambios organizacionales y la automatización del proceso. La primera se refiere al conjunto de actividades que se requieren para cambiar el modo de trabajar de todos los participantes involucrados en el proceso. La segunda, por otra parte, se refiere al desarrollo y despliegue de los sistemas de TI (o versiones mejoradas de los mismos) que respaldan el proceso *to-be*.

Por último, la fase de *monitoreo y control* se tiene una vez que el proceso rediseñado está ejecutándose. Los datos relevantes se coleccionan y analizan para determinar cuán bien se desempeña el mismo con respecto a las métricas y objetivos de desempeño. Se identifican los cuellos de botella, los errores recurrentes o las desviaciones con respecto al comportamiento intencionado del proceso y se emprenden acciones, (Moreno Montes de Oca 2015).

1.3 Modelado de procesos de negocios

Los modelos de procesos de negocio son importantes en varias etapas del ciclo de vida BPM. Existen muchas razones para modelar un proceso. La primera de ellas es simplemente para entender el proceso y para compartir nuestro entendimiento del proceso con las personas involucradas con el proceso diariamente. Además, el modelado de procesos ayuda a entender mejor el proceso e identificar y prevenir problemas (Dumas et al. 2013).

Los modelos de procesos de negocio son un artefacto fundamental para implementar procesos de negocio. Esta implementación puede ser realizada por reglas organizacionales y políticas, pero también puede ser hecha por un sistema de software, usando un sistema de gestión de procesos de negocio (Weske 2007). Profesionales de negocio crean modelos de procesos de negocio en orden para obtener un entendimiento de quién está involucrado y qué recursos son necesarios para la ejecución de un proceso de negocio. Tales modelos de procesos de negocio son requeridos como base de transferencia conocimiento, propósitos de

calidad, regulaciones, comunicación entre internos y externos socios colaborativos, o documentación en general (Born et al. 2009).

El modelado de procesos de negocio es una importante parte de un diseño de sistemas de información. Las técnicas de modelado de procesos de negocio proveen una forma estándar de presentación y comunicación de procesos de negocio (Heidari et al. 2011). Desde que los procesos de negocio son influenciados por su ambiente, el modelado de procesos de negocio inevitablemente hay que considerarlo como un factor y su naturaleza variables (de la Vara et al. 2010). Para (Khlif et al. 2009), el modelado de procesos de negocio es necesario para una empresa que desee evaluar, mejorar, migrar a una plataforma tecnológica diferente, automatizar, y/o documentar sus proceso de negocio.

Los modelos de proceso de negocio son una especie de descripción de procesos de negocio. Con el uso del modelo, estamos disponibles analizar el rendimiento de los procesos, guiar la implementación de procesos, monitorear la ejecución de procesos, habilitar la gestión de procesos (Ning et al. 2004). Para (Schrepfer 2010), el modelado de procesos de negocio es la actividad humana crea modelos de procesos de negocio.

El modelado de proceso de negocio tiene dos enfoques: el *as-is* y el *to-be*. El enfoque *as-is* es ajustado para ilustrar el negocio actualmente. El enfoque *to-be* es ajustado para la simulación de cambios a ser implementados en el futuro. El modelado de proceso de éstos dos enfoques no es el mismo (Castela et al. 2002). Un modelo de proceso es usualmente expresada por medio de notación gráfica, que debe ser capaz capturar toda la información relevante desde el modelo y adicionalmente debe facilitar el análisis del proceso de modelado por verificación estática y/o simulación dinámica (Badica & Fox 2002).

1.4 Lenguajes para la modelación de procesos de negocio

Al modelar un proceso se busca desarrollar una descripción lo más exacta posible de este, así como de las actividades y demás elementos que lo conforman. Es una actividad mediante la cual un proceso se representa o describe usando lenguajes apropiados que faciliten la comunicación de la representación, su documentación, la comprensión del proceso y la visualización de las interrelaciones que existen entre las distintas actividades que lo definen. Los lenguajes notacionales permiten representar gráficamente un modelo de proceso. Estos lenguajes definen su conjunto de reglas semánticas y sintácticas que establecen los elementos

y las relaciones que son parte de los procesos. El modelado de procesos de negocios es también conocido como especificación de flujos de trabajo. Para modelar un flujo de trabajo, es necesario hacer uso de un lenguaje o herramienta de modelado (Reyes García 2016).

Analistas de procesos de negocio usan lenguajes de gráfico como *Event-driven Process Chain* (EPC), diagramas de actividad UML, **Redes de Petri**, **BPEL**, **IDEF0**, **BPMN** o **YAWL**, (Gruhn & Laue 2008). Aunque existen otros más, como los modelos de procesos de negocio poco estructurados, analizados en (Gruhn & Laue 2008).

A continuación, se exponen elementos fundamentales de BPMN por ser este el estándar, por ser el centro de estudio de SAD_BPMN y puesto que es una notación universal.

1.4.1 BPMN

BPMN es una notación para el modelado de procesos de negocios desarrollada por la BPMP una organización incluye compañías como: Intalio, SAP, Sun, y Versata, siendo una agrupación que tiene dentro de sus objetivos principales el crear una notación estándar para el modelado de procesos de negocio. Provee una notación gráfica para expresar los procesos de negocio en un diagrama de procesos de negocio. Tiene como objetivo principal servir como soporte para la gestión por procesos, como una notación que pueda ser entendida fácilmente desde los analistas que crean los bocetos iniciales del proceso, los desarrolladores técnicos responsables de implementar la tecnología que ejecutará estos procesos, hasta las personas que los ejecutan y aquellas que llevarán a cabo el monitoreo y supervisión de los procesos. En otras palabras, esta notación crea un enlace entre las etapas de diseño e implementación. Dentro de todos estos lenguajes y notaciones el más integrador es BPMN, para su construcción sus creadores, los miembros del grupo de trabajo para la notación de BPMP, revisaron y analizaron diferentes notaciones existentes tomando de ellas las mejores ideas consolidándolas en una notación estándar. Entre las notaciones y metodologías revisadas están: Diagrama de Actividad de UML, IDEF, ebXML BPSS, Diagrama ADF, RosettaNet, LOVeM y EPCs entre otras (Rodríguez Andrés 2008).

Las cuatro categorías fundamentales de BPMN para (Calzada Ricet & Pairol Fuentes 2016) son:

- Objetos de flujo: Eventos, Actividades, Compuertas
- Objetos Conectivos: Flujo de Secuencia, Flujo de Mensajes, Asociación

- Contenedores: Piscina, carriles (acrónimo del inglés *Pool, Lane*)
- Artefactos: Objetos de Datos, Grupos, Anotaciones

BPMN fue diseñado para permitir a los modeladores y las herramientas de modelado un poco de flexibilidad al extender la notación básica y habilitar un contexto apropiado adicional según una situación específica, como para un mercado vertical. Se puede añadir cualquier número de artefactos a un diagrama como sea apropiado para un contexto de proceso de negocio específico (Calzada Ricet & Pairol Fuentes 2016).

De acuerdo a (White & Miers 2009), “En BPMN, los procesos de negocio involucran la captura de una secuencia ordenada de las actividades e información de apoyo. Modelar un proceso de negocio implica representar cómo una empresa realiza sus objetivos centrales; los objetivos por sí mismos son importantes, pero por el momento no son capturados por la notación. Con BPMN, sólo los procesos son modelados”.

Este trabajo es enfocado hacia los modelos desarrollados en la notación BPMN, pues tal y como señala (Reyes García 2016): “este es el estándar propuesto por la OMG para la captura de procesos de negocio. También, por su amplio uso tanto en la academia como en la industria, así como por la gran popularidad entre analistas de negocio, vendedores de herramientas (62 según la OMG), practicantes, y usuarios”.

Según (Calzada Ricet & Pairol Fuentes 2016), la intención de BPMN es servir como lenguaje de enlace entre el diseño y la implementación de los procesos de negocio, así como ofrecer soporte solamente para los conceptos aplicables a los procesos de negocio.

BPMN es importante además porque:

- Presenta amplias posibilidades de consolidarse en un estándar internacional de modelado de procesos.
- Es independiente de la metodología de modelado de procesos.
- Es un lenguaje formal que describe los pasos de una lógica de negocios.
- Permite el entendimiento generalizado de los procesos dentro de la organización.
- Facilita la comunicación.

Se puede encontrar mayor especificación de BPMN en el anexo 1.

Herramientas para la modelación con BPMN

Con el aumento del uso de la gestión de procesos por parte de las empresas, cada vez se requiere de mejores herramientas para el análisis de los diferentes procesos que describen todas las tareas y roles de los empleados y socios de negocio, así como las aplicaciones de software que los soportan y los flujos de trabajo tanto internos como externos. Todo esto es esencial para alcanzar las metas de negocio de una compañía (Reyes García 2016).

Hoy en día las empresas se enfrentan a diversos problemas relacionados con los procesos de negocio. En donde se observa ausencia de participación de los responsables del negocio sólo por el hecho de que estos no entienden el funcionamiento del sistema. Es allí donde surgen las herramientas de software BPMN que facilitan el entendimiento de los procesos de negocio en una organización por parte de diferentes usuarios involucrados en el proceso de software como el cliente y los desarrolladores. Cuando se habla de herramientas de apoyo de BPM, se habla principalmente de BPMN (*Business Process Modeling Notation*), enfocándose a la administración de los procesos (Mesa et al. 2014).

Las características de estas herramientas son diversas, desde código abierto o privado hasta comerciales o públicas, utilizadas en nuestro centro o no, herramientas modeladoras con validador o no, herramientas gráficas, entre otras. La mayoría de las aplicaciones comerciales se le evaluaron los modelos en su versión de prueba (trial, en inglés) o en su versión de comunidad (*community*, por sus siglas en inglés).

En (Jiménez Martínez & Reyes García 2018) se analizaron varias de éstas herramientas de acuerdo a las mismas directrices de calidad, éstos resultados son referenciados en el presente trabajo junto a otros análisis similares que se realizaron en (Reyes García 2016) y (Moreno Montes de Oca 2015).

Para determinar la calidad y soporte de directrices de cada una de las herramientas mencionadas anteriormente, y caracterizadas en cada uno de sus correspondientes anexos, se realizaron diferentes modelos erróneos de ejemplo respectivos a cada directriz existente de la tabla 1.1. Los datos colectados de los trabajos citados con respecto al análisis de las herramientas y los de elaboración propia se pueden observar en las figuras 1.2 y 1.3.

En dichas figuras es necesario destacar varios datos y elementos a tener en cuenta. En la columna *Combinado* se muestra el porciento total de las herramientas (9,7%). La

herramienta Signavio es la de mayor grado de soporte mientras ofrecen 0% las herramientas *Dia*, *yEd*, *Edraw Max* y *Software Ideas Modeler*. Signavio, cumple con menos de la mitad de las métricas (46%), lo que motiva la necesidad de tener una herramienta que permita evaluar los modelos teniendo en cuenta las directrices de calidad (véase anexo 4).

Los datos correspondientes a las herramientas *Aris Express*, *Signavio* y *Visual Paradigm* son extraídos de (Reyes García 2016) y (Moreno Montes de Oca 2015), en el caso de las herramientas restantes son el resultado de la elaboración propia del autor del presente trabajo.

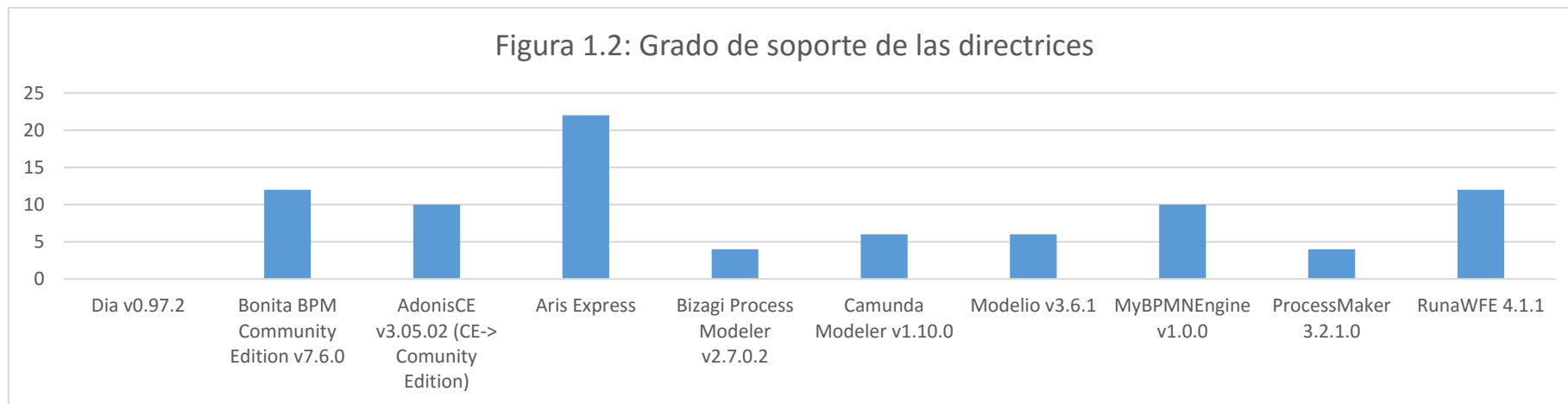


Figura 1.2 Grado de soporte de las directrices

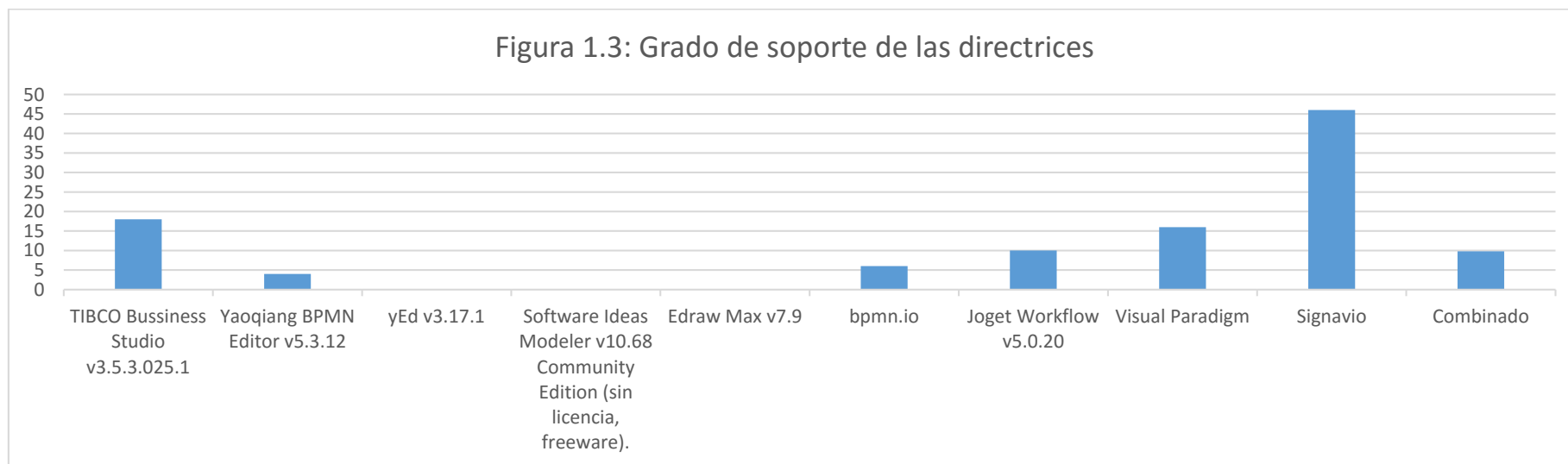


Figura 1.3 Grado de soporte de las directrices

1.5 Calidad de los modelos de proceso de negocio

La calidad de un modelo de proceso de negocio influencia grandemente la actividad deseada, según (Khlif et al. 2009). Un problema notorio relacionado con la calidad de los modelos es la baja competencia de modelado que pueden poseer modeladores casuales en proyectos de documentación de procesos (Mendling et al. 2010).

Es necesario evaluar la calidad de los modelos de procesos desde diferentes puntos de vista. Esto es aún tan importante como el número creciente de diferentes usuarios finales, diferentes propósitos y la capacidad de diferentes técnicas de modelados y herramientas de modelados guían a una mayor complejidad de los modelos de información (Becker et al. 2000). La calidad de un proceso de negocio naturalmente afecta la calidad del negocio a quien pertenece desde que entrega algunos tipos de resultados. Diferentes áreas son investigadas en orden para extraer y adaptar dimensiones de calidad para hacer más apropiado y aplicable para un proceso de negocio (Heidari et al. 2011). Es necesario ser capaces de evaluar la calidad de un modelo de proceso de negocio, lo que requiere un conjunto de métricas de calidad. La mayoría de los trabajos propuestos para evaluar los modelos de procesos de negocio tratan con la calidad adaptando métricas de software. Esto es posible, porque los productos de software y los software de procesos de negocio son bastante similares (Khlif et al. 2009).

En (Mendling et al. 2009) se define “la necesidad de ver la calidad de modelado como un proceso que ocurre en un determinado contexto” y además se aporta un marco de trabajo para garantizar la calidad de los modelos de procesos de negocio. Los factores de calidad son agrupados dentro de dimensiones de calidad, cada dimensión representa una faceta de calidad de procesos de negocio. Un factor de calidad representa un aspecto particular de una dimensión dada. Un factor de calidad es asociado con un conjunto de métricas permitidas para efectivamente evaluarlo (Heidari et al. 2011). Para (Schrepfer 2010), la mayoría de los aspectos de calidad respecto al proceso de modelado son puestos y documentados durante la fase de inicialización, pero deberían ser monitoreadas durante el proyecto.

Un concepto sumamente importante en el tema de la calidad del modelado de procesos de negocio es el de directriz práctica. Según (Mendling et al. 2010), una *directriz práctica* en el contexto de modelado de procesos de negocio constituye una buena práctica o consejo explícito que ofrece una guía sobre cómo un modelo de proceso puede mejorarse y cuál

alternativa de un conjunto de representaciones de comportamiento equivalente debe preferirse, que sirve como una importante herramienta de transferencia de conocimiento desde la academia hacia la práctica del modelado.

Se clasificaron en una taxonomía en (Moreno Montes de Oca 2015) como se muestra en la figura 1.4.

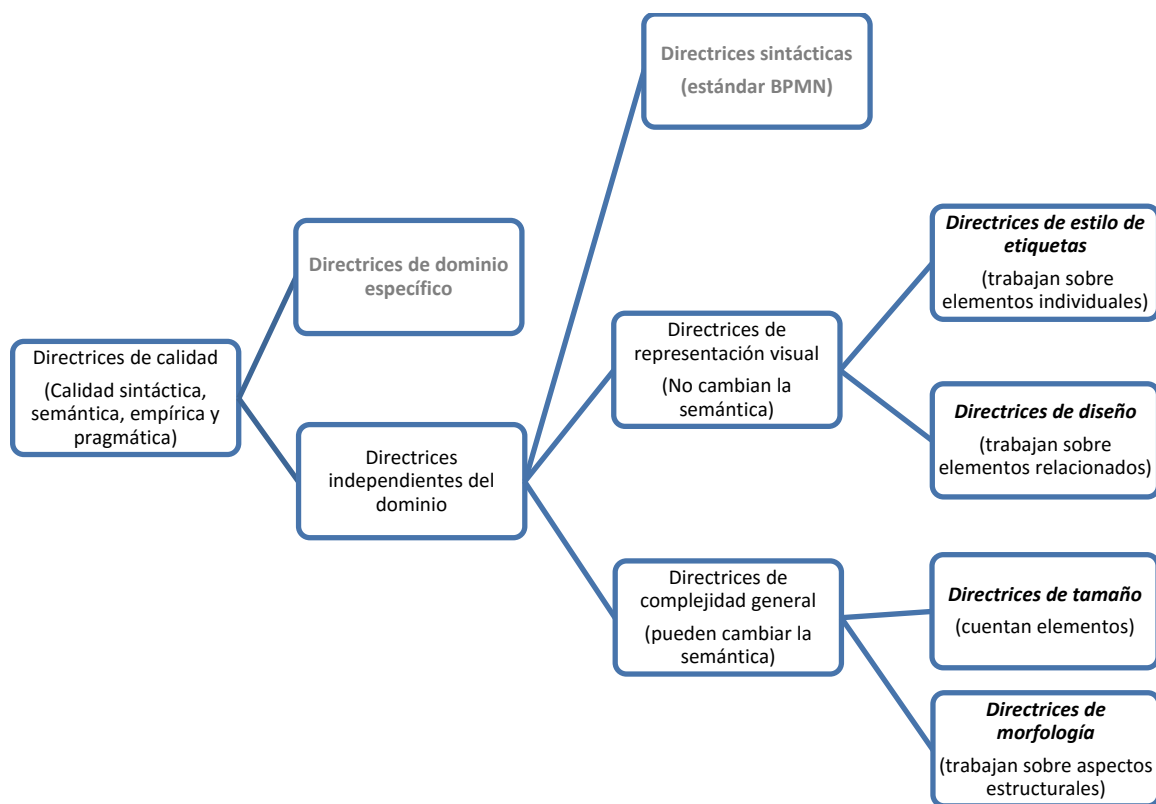


Figura 1.4 Taxonomía para directrices de calidad de modelado de procesos de negocio (Fuente: (Moreno Montes de Oca 2015))

Tabla 1.1 Clasificación taxonómica de las directrices de calidad para modelos de procesos de negocio (*Fuente:* (Moreno Montes de Oca 2015)).

Directrices de Calidad			
Representación visual		Complejidad general	
Diseño	Estilo de etiquetado	Morfología	Tamaño
P27: Alto número de cruces de flujos de secuencia P28: Curvas en los flujos de secuencia P29: No perpendicularidad P30: Solapamiento de flujos de secuencia P31: Solapamiento de elementos P32: Solapamiento de flujos de secuencia y elementos P33: Flujo de secuencia inconsistente P34: Longitud variada de flujos de secuencia P35: Tamaño no uniforme de los elementos P36: Modelo cuadrado P37: No simetría P38: Área de dibujo extensa P39: Lejanía de elementos relacionados P40: Apariencia no uniforme P41: Diagrama desordenado	P42: Las etiquetas no siguen el estilo verbo-sustantivo P43: Tipos de tareas no especificados P44: Etiquetas largas P45: Eventos sin etiquetar P46: Compuertas sin etiquetar P47: Actividades sin etiquetar P48: Flujo de secuencia de salida sin etiquetar P49: Pool sin etiquetar P50: Etiquetas enviar/recibir no están reservadas para estos elementos	P11: Falta de estructuración en el modelo P12: Alto número de flujos de secuencia por compuerta P13: Alto número de flujos de secuencias por eventos P14: Múltiples entradas y salidas en la misma compuerta P15: Múltiples ciclos P16: Ciclos no estructurados P17: Ciclos mal formados P18: Alto nivel de paralelismo P19: Paralelismo incorrecto P20: Alto nivel de profundidad P21: Elementos duplicados P22: Elementos innecesarios P23: Diámetro largo P24: Alta diversidad de compuertas P25: Uso de compuertas de división de tipo <i>or</i> P26: Alto grado de complejidad	P1: Alto número de elementos P2: Alto número de eventos P3: Alto número de eventos de inicio P4: Ausencia de eventos de inicio P5: Alto número de eventos de fin P6: Ausencia de eventos de fin P7: Alto número de eventos intermedios P8: Alto número de flujos de secuencia P9: Alto número de compuertas P10: Alto número de actividades
Directrices de Calidad			
Representación visual		Complejidad general	
Diseño	Estilo de etiquetado	Morfología	Tamaño

<p>P27: Alto número de cruces de flujos de secuencia</p> <p>P28: Curvas en los flujos de secuencia</p> <p>P29: No perpendicularidad</p> <p>P30: Solapamiento de flujos de secuencia</p> <p>P31: Solapamiento de elementos</p> <p>P32: Solapamiento de flujos de secuencia y elementos</p> <p>P33: Flujo de secuencia inconsistente</p> <p>P34: Longitud variada de flujos de secuencia</p> <p>P35: Tamaño no uniforme de los elementos</p> <p>P36: Modelo cuadrado</p> <p>P37: No simetría</p> <p>P38: Área de dibujo extensa</p> <p>P39: Lejanía de elementos relacionados</p> <p>P40: Apariencia no uniforme</p> <p>P41: Diagrama desordenado</p>	<p>P42: Las etiquetas no siguen el estilo verbo-sustantivo</p> <p>P43: Tipos de tareas no especificados</p> <p>P44: Etiquetas largas</p> <p>P45: Eventos sin etiquetar</p> <p>P46: Compuertas sin etiquetar</p> <p>P47: Actividades sin etiquetar</p> <p>P48: Flujo de secuencia de salida sin etiquetar</p> <p>P49: Pool sin etiquetar</p> <p>P50: Etiquetas enviar/recibir no están reservadas para estos elementos</p>	<p>P11: Falta de estructuración en el modelo</p> <p>P12: Alto número de flujos de secuencia por compuerta</p> <p>P13: Alto número de flujos de secuencias por eventos</p> <p>P14: Múltiples entradas y salidas en la misma compuerta</p> <p>P15: Múltiples ciclos</p> <p>P16: Ciclos no estructurados</p> <p>P17: Ciclos mal formados</p> <p>P18: Alto nivel de paralelismo</p> <p>P19: Paralelismo incorrecto</p> <p>P20: Alto nivel de profundidad</p> <p>P21: Elementos duplicados</p> <p>P22: Elementos innecesarios</p> <p>P23: Diámetro largo</p> <p>P24: Alta diversidad de compuertas</p> <p>P25: Uso de compuertas de división de tipo <i>or</i></p> <p>P26: Alto grado de complejidad</p>	<p>P1: Alto número de elementos</p> <p>P2: Alto número de eventos</p> <p>P3: Alto número de eventos de inicio</p> <p>P4: Ausencia de eventos de inicio</p> <p>P5: Alto número de eventos de fin</p> <p>P6: Ausencia de eventos de fin</p> <p>P7: Alto número de eventos intermedios</p> <p>P8: Alto número de flujos de secuencia</p> <p>P9: Alto número de compuertas</p> <p>P10: Alto número de actividades</p>
--	--	---	---

1.5.1 Herramienta para la evaluación de modelos BPMN

Según (Calzada Ricet & Pairol Fuentes 2016), SAD_BPMN v1.0 es un sistema para la evaluación de modelos BPMN que permite recomendar a los usuarios cómo mejorar la calidad de sus modelos de procesos de negocio interpretando los archivos de salida .bpmn de la herramienta de modelado Bonita. El sistema está desarrollado completamente en JAVA, su código es libre y multiplataforma. Solo requiere que el sistema operativo tenga instalado Java

Runtime Enviroment (JRE). Su diseño permite que sea extensible, flexible y reutilizable con facilidad.

La herramienta ofrece la posibilidad de analizar todas las directrices en el modelo BPMN o una parte de ellas, según desee el modelador. Al evaluar el modelo se calculan las métricas de calidad asociadas a las directrices y se determina si el modelo posee los problemas asociados de acuerdo al patrón propuesto. Para las directrices cuyas métricas no tienen umbrales asociados se muestran los valores de las mismas y los usuarios tienen noción de lo que ocurre en sus modelos respecto a las directrices (Moreno Montes de Oca 2015).

Las versiones 1.0 y 1.1 fueron fruto de la labor conjunta de los trabajos (Moreno Montes de Oca 2015), (Hernández Cárdenas 2016) y (Reyes García 2016).

En esta herramienta se encuentran implementados dos módulos para el análisis de los modelos procesos de negocio que incluyen el módulo para la evaluación de directrices de complejidad general, y el módulo para la evaluación de directrices de representación visual (Calzada Ricet & Pairol Fuentes 2016).

La versión 2.0 desarrollada por (Calzada Ricet & Pairol Fuentes 2016), implementó las lecturas modelos de procesos de negocio desarrollados en Bizagi mediante el análisis de la representación de los elementos de salida BPMN en ficheros .xpdL y un módulo para realizar propuestas de cambios a los modelos de forma visual a partir de su evaluación, conocido como la funcionalidad *Sugerencia*.

Como parte de la investigación que se ha desarrollado en el presente trabajo, se realizó un exhaustivo análisis de la última versión de SAD_BPMN (versión 2.0). Este análisis ratificó las funcionalidades que anteriormente se han expuesto, pero brindó también algunos errores que contiene la aplicación. Entre los errores se encontraron:

- Incorrecta validación de errores y excepciones, como por ejemplo no se valida cuando se carga un modelo con extensión incorrecta, o cuando teniendo extensión correcta haya sido generado por una de las herramientas soportadas.
- Divisiones entre 0 durante el cálculo de las métricas de calidad.
- Errores en el árbol de directrices en la interfaz gráfica.

- Documento generado de exportar a ficheros Excel, tenía un estilo incorrecto, y en muchos casos por el incorrecto tamaño del ancho de las columnas se perdía información, para la cual había que manualmente agrandar los anchos de dichas columnas.
- Ausencia de gráficos que permitan resumir de forma visual los problemas en los modelos.
- Existencia de directrices sin programar como: pool sin etiquetar, longitud variada de flujos de secuencia, solapamiento de flujos de secuencia, solapamiento de elementos y solapamiento de flujos de secuencia y elementos.

1.6 Grafos

Los grafos son entidades matemáticas que permiten representar relaciones entre objetos. Pueden definirse como un conjunto no vacío de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas) (West 2001).

Los grafos también son útiles en el modelado de procesos de negocio, pues un modelo de proceso de negocio puede ser analizado como un grafo dirigido, donde los objetos de flujo son los vértices del grafo y los objetos conectivos son las aristas. Debido a que los modelos de procesos de negocio pueden contener aristas paralelas y bucles una representación más exacta es a través de pseudografos dirigidos. De esta forma se puede realizar un análisis de los elementos BPMN en los ficheros *.bpmn* y *.xpdL* generados por Bonita y Bizagi respectivamente, realizando un cotejo entre estos elementos y la estructura de un pseudografo dirigido. Una vez realizado el mapeo, el cálculo de las directrices de calidad se puede efectuar, mediante recorridos en el pseudografo (Reyes García 2016). Puede encontrar una mejor descripción acerca de esta temática en el anexo 7.

1.7 Fundamentación del entorno de desarrollo

El lenguaje de programación seleccionado es *Java* puesto que es un lenguaje de programación robusto, simple, orientado a objetos, seguro, portable, de alto rendimiento, distribuido, dinámico, interpretado, de arquitectura neutral y multihilo.

Como Entorno de Desarrollo Integrado (IDE) se seleccionó a *NetBeans*, pues como se menciona en (Velasco 2016), “es uno de los IDE de programación más utilizados tanto por

aficionados como por profesionales para programar en distintos lenguajes de programación, especialmente en Java (y en *Android*, antes de la llegada de *Android Studio*). Este IDE, desarrollado y mantenido por *Oracle*, es totalmente gratuito y cuenta con un desarrollo libre, por lo que cualquiera puede participar en el desarrollo y aportar ideas para mejorar el proyecto con las constantes actualizaciones”.

Entre las dependencias de terceros utilizadas se encuentran:

- **Jgraphx** es un componente gráfico de Java ligero y rico en características, y su sucesor es **Jgraph**. Provee el diseño de 2D automático y el direccionamiento para diagramas. Objetos y relaciones pueden ser mostradas en cualquier Swing UI y aplicarle zoom.

Este plugin de terceros nos permitirá modelar gráficamente el modelo de proceso de negocio a evaluar, e incluso, a mostrar una sugerencia al usuario.

- **JDom**. Según (Feregrino 2012), esta es una biblioteca para Java que facilita el uso y manipulación de documentos XML en memoria y de manera persistente. Similar al DOM del W3C, con la potente diferencia de que este se refuerza con las características de Java.

Esto nos permitirá trabajar con los documentos xpdl y bmpn exportados desde diferentes herramientas y exportar diferentes datos u objetos a ficheros xml.

- **iText**. Es una librería muy potente y fácil de usar para generar reportes desde Java (Guerrero 2015). Según (Falla Aroche 2009), con iText puedes generar archivos PDF de forma rápida, soportado por la tecnología Java y con una licencia MPL/LGPL.

iText permitirá exportar hacia documentos pdf información que el usuario desee reportar.

- **jxl**. Permite leer, escribir, crear y modificar hojas en un libro de trabajo Excel (.xls), no soporta formato .xlsx.
- **jfreechart**. Según (Villamarín Rodríguez 2010), JFreeChart es una librería la cual nos permite hacer gráficos de una forma muy sencilla, esta librería posee todo tipo de gráficos, desde tortas, líneas, barras, etc.
- **OpenCSV**. Para (Gracia 2014), es una librería Java que permite trabajar con CSVs de forma muy sencilla, tanto para leer como para escribirlos, (<http://opencsv.sourceforge.net>).

1.8 Conclusiones parciales

Desde sus inicios, los procesos de negocio despertaron gran interés por su aplicación para las organizaciones, donde mediante la práctica ganó su investigación y profundización adquirió importancia y se llegó a la gestión de procesos de negocio y modelado de los mismos; gestión soportada por el importante tema de la calidad de los procesos de negocio. Para evaluar dicha calidad se elaboraron un conjunto de directrices de calidad, las cuales se evaluaron en un conjunto de herramientas de modelado de procesos de negocio y se realizó un análisis a cada una de las mismas con respecto a los resultados. Los resultados de ésta evaluación denotan el poco soporte de las herramientas evaluadas y la necesidad de crear una herramienta de evaluación para los modelos generados. SAD_BPMN es una aplicación desktop desarrollada en lenguaje Java que permite evaluar modelos procesos de negocio teniendo como parámetros directrices de calidad; ésta herramienta posee errores de soporte y usabilidad, y se pretende añadirle nuevas funcionalidades.

Capítulo II. Modelo del Negocio y Requisitos

En el presente capítulo se exponen el modelo del negocio actual, las reglas del negocio a considerar, los actores del negocio, diagramas de casos de uso del negocio y del sistema, trabajadores del negocio, casos de uso del negocio, actores del sistema a automatizar, definición de los requisitos funcionales y no funcionales y los paquetes y sus relaciones. Todo lo anterior apoya la solución propuesta.

2.1 SAD_BPMN v2.0. Análisis de soporte de directrices de calidad

En (Calzada Ricet & Pairol Fuentes 2016) se hizo un análisis de SAD_BPMN v1.1, mediante el cual planificaron su trabajo para eliminar debilidades que se solucionarían con SAD_BPMN v2.0. En la figura 2.1 se muestra el soporte de SAD_BPMN v2.0 a las directrices prácticas de calidad. Como se puede observar en las matrices de complejidad no existen métricas mal programadas, hay 27 correctamente programadas (93%), pero existen 2 sin programar (7%), éstas son: paralelismo incorrecto (debido a la complejidad de programación) y elementos innecesarios (puesto que encuentra de manera explícita en la literatura las posibles descripciones de un elemento innecesario en modelos BPMN). Mientras que en el grupo taxonómico de representación visual existen 12 métricas bien programadas (63%), 5 sin programar (26%) y 2 mal programadas (11%). Las directrices prácticas mal programadas fueron área de dibujo y lejanía de elementos relacionados, siendo las no programadas: pool sin etiquetar, longitud variada de flujos de secuencia, solapamiento de flujos de secuencia, solapamiento de elementos y solapamiento de flujos de secuencia y elementos. El soporte se analizó teniendo como base las directrices prácticas de calidad descritas en (Moreno Montes de Oca 2015).

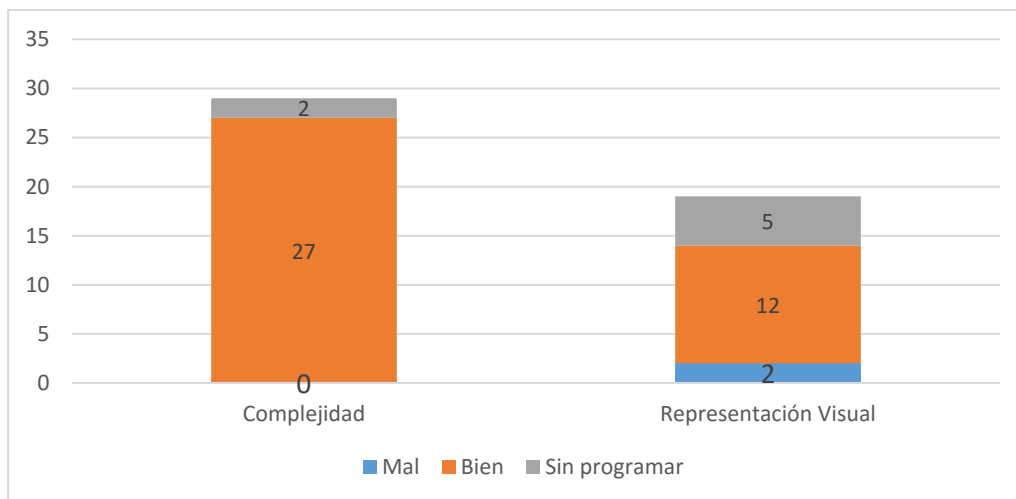


Figura 2.1 Soporte de las métricas en SAD_BPMN v2.0

Realizando un análisis comparativo entre las métricas brindadas en (Moreno Montes de Oca 2015), a las cuales llamaremos “directrices actuales” y (Moreno Montes de Oca & Snoeck 2017), las cuales se denotan como “directrices nuevas”, se obtiene la figura 2.2.

Como nos interesa mantener las “directrices actuales” que no se encuentran en las “directrices nuevas”, como directrices que se mantienen para brindar mayor usabilidad y soporte a SAD_BPMN v3.0, es por ello que en la figura 2.2 se representan como 37 las “directrices actuales” que se mantienen, sumando las 10 métricas mencionadas anteriormente con las 27 métricas que se mantienen y las 10 métricas conformadas. En el anexo 4 se puede encontrar mayor información con respecto a todo lo abordado anteriormente.

Haciendo un resumen de las figuras 2.1 y 2.2 se obtiene la figura 2.3, la cual generaliza el soporte de directrices prácticas de calidad de SAD_BPMN v2.0 y se obtiene el total de 58 directrices entre “nuevas” y “actuales”.

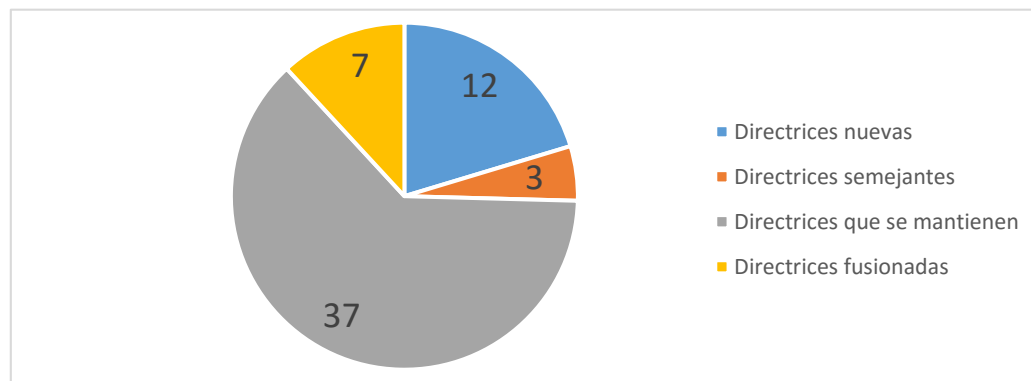


Figura 2.2 Comparación entre las directrices “nuevas” y “actuales”

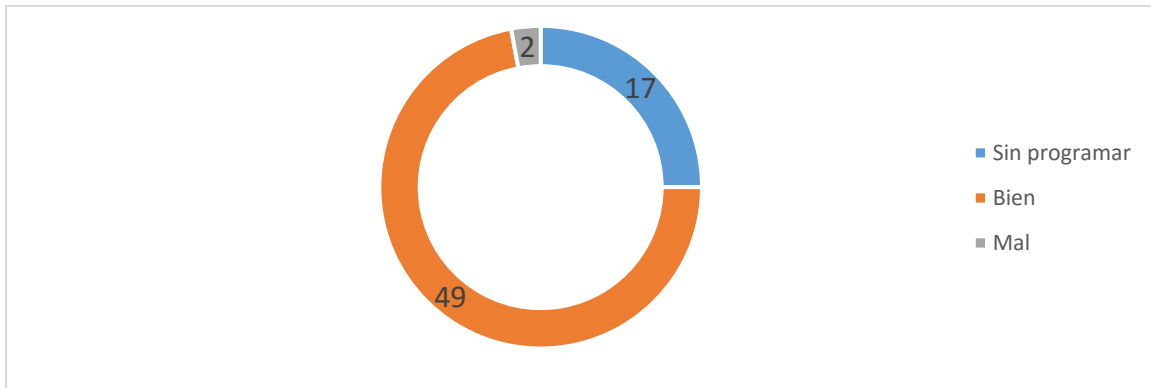


Figura 2.3 Soporte de SAD_BPMN v2.0 para las directrices “nuevas” y “actuales”

Las 3 métricas semejantes y las 7 métricas fusionadas se incluyeron en las directrices que se encuentran en correcto estado en la figura 2.3, es por ello que el total es 49 (39 que se encontraban de las “directrices actuales” en buen estado sumado a las 10 que se mencionaban anteriormente).

2.2 Diagrama de casos de uso del negocio

Según (Craig Larman 1999), “Los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican e incluyen tácitamente los requerimientos en las historias que narran”.

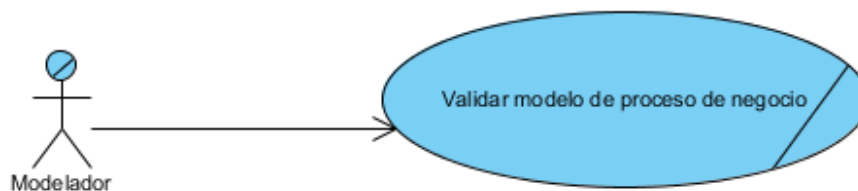



Figura 2.4 Diagrama de casos de uso del negocio

El modelador es el actor del modelo del negocio ya que es la persona capacitada para modelar un proceso de negocio y determinar los errores que en dichos modelos se puedan encontrar.

2.3 Actores del negocio y sistema

En el caso de este proyecto ambos actores son los mismos y juegan el mismo rol. Según (Jacobson et al. 2000), “un actor juega un papel por cada caso de uso con el que colabora”, y se añade: “los actores representan terceros fuera del sistema que colaboran con el sistema”.

Tabla 2.1 Actor del sistema SAD_BPMN v2.0

Actor	Descripción
	Persona que necesita validar un modelo de proceso de negocio. En el caso del sistema puede validar un modelo generado en Bonita o Bizagi.

2.4 Definición de los requisitos funcionales y no funcionales

Para (Craig Larman 1999), “los requerimientos son una descripción de las necesidades o deseos de un producto. La meta primaria de la fase de requerimientos es identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo. El reto consiste en definirlos de manera inequívoca, de modo que se detecten los riesgos y no se presenten sorpresas al momento de entregar el producto”.

De acuerdo a (Jacobson et al. 2000), “el propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre qué debe y qué no debe hacer el sistema”.

Los requisitos identificados y definidos para SAD_BPMN v3.0 son:

2.4.1 Requisitos funcionales

RF1: Cargar modelo de proceso de negocio.

RF2: Calcular métricas de calidad.

RF3: Mostrar sugerencias visuales.

RF4: Exportar datos.

RF5: Mostrar modelo de proceso de negocio.

2.4.2 Requisitos no funcionales

RNF1: *Requerimientos de apariencia o interfaz externa*. El sistema cuenta con una interfaz amigable asequible a los usuarios y responde al cumplimiento de los requisitos planteados.

RNF2: *Requerimientos de usabilidad*. Este nuevo sistema podrá ser utilizado por personal con poco conocimiento informático ya que el mismo brinda documentación de ayuda.

RNF3: *Software*. JRE 1.7 o superior

RNF4: *Hardware y capacidad* (configuración mínima). 128 megabytes de memoria RAM, 18 megabytes de espacio en disco para el ejecutable de la instalación, 1 megabyte extra para guardar los ficheros

RNF5: *Requisitos de soporte*. Pruebas (se realizaron pruebas de interfaz).

RNF6: *Requisitos Legales*.

- Registro legal del software.
- Se registrará por las normas ISO todo el sistema.

2.5 Diagrama de paquetes y sus relaciones

En (Craig Larman 1999) se destaca “El lenguaje UML ofrece el mecanismo paquete que permite describir los grupos de elementos o subsistemas. Un paquete es un conjunto de cualquier tipo de elementos de un modelo: clases, casos de uso, diagramas de colaboración u otros paquetes (los anidados). Un sistema puede examinarse íntegramente dentro del ámbito de un solo paquete de alto nivel: el paquete Sistema. El paquete define un espacio de un nombre anidado, de modo que los elementos del mismo nombre pueden duplicarse dentro de varios paquetes. (...) Los paquetes nos permiten describir la arquitectura de un sistema con la notación UML”. Además, se define al diagrama de paquetes como la representación de agrupamientos lógicos mediante los diagramas de un paquete de UML.

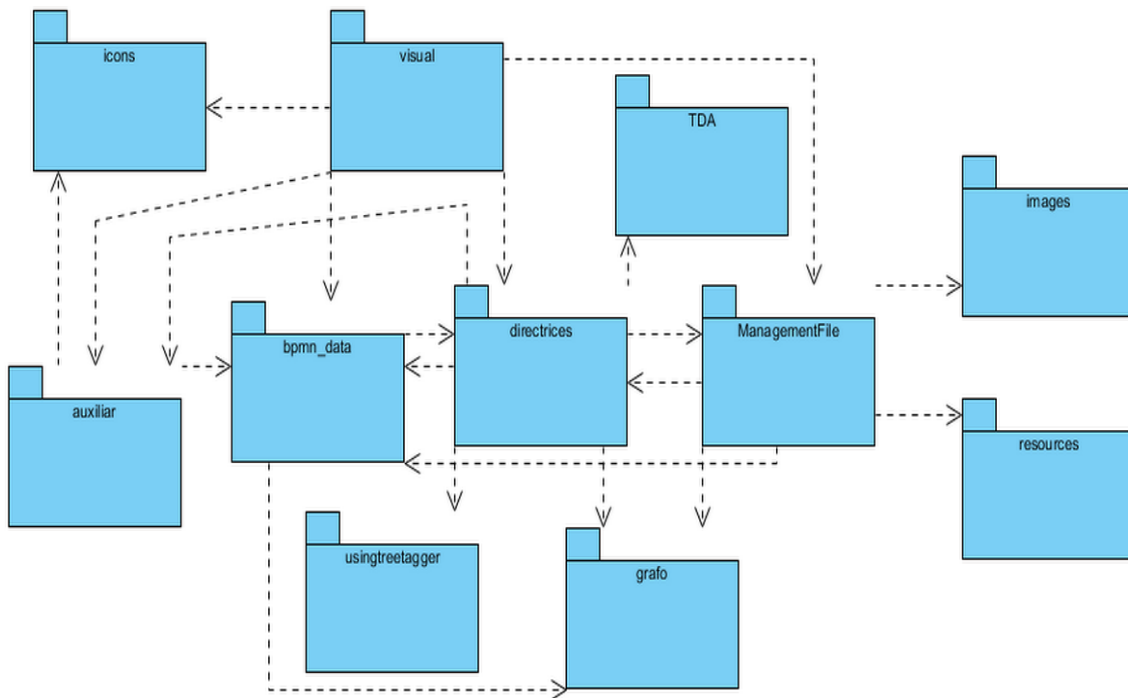


Figura 2.5 Diagrama de paquetes y sus relaciones

2.6 Diagrama de casos de uso del sistema

Los casos de uso del sistema se caracterizan en (Craig Larman 1999) cuando se expresa: “Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre éstos y los casos de uso. Estos últimos se muestran en óvalos y los actores son figuras estilizadas. Hay líneas de comunicaciones entre los casos y los actores; las flechas indican el flujo de la información o el estímulo. El diagrama tiene por objeto ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan.”

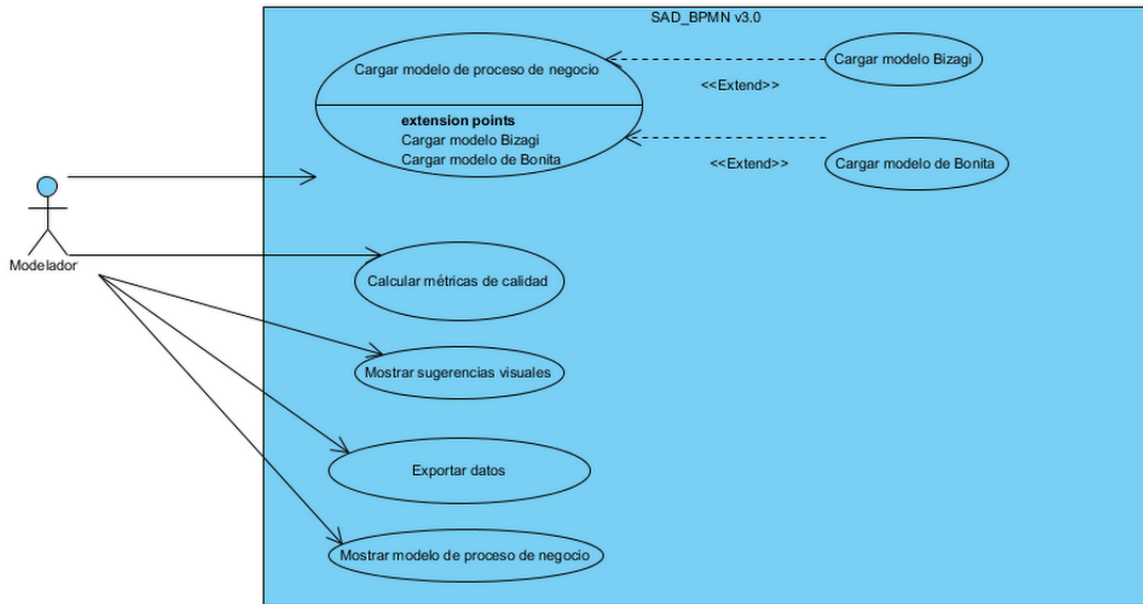


Figura 2.6 Diagrama de casos de uso del sistema

2.7 Descripción de los casos de uso del sistema

En (Calzada Ricet & Pairol Fuentes 2016), autoras de SAD_BPMN v2.0, se brinda una tabla resumen de los casos de usos del sistema, donde podemos encontrar a: *Calcular métricas de calidad*, *Cargar modelo de proceso de negocio* y *Mostrar sugerencias visuales*. Es necesario explicar que el caso de uso de dicha fuente *Guardar consejos* cambia tanto en nombre como en descripción y declaración, en SAD_BPMN v3.0 este caso de uso es *Exportar datos*. A continuación, en la tabla 2.2 se describen resúmenes de este último caso de uso mencionado y del nuevo caso de uso, introducido en SAD_BPMN v3.0, *Mostrar modelo de proceso de negocio*.

Tabla 2.2 Resumen de los casos de uso del sistema

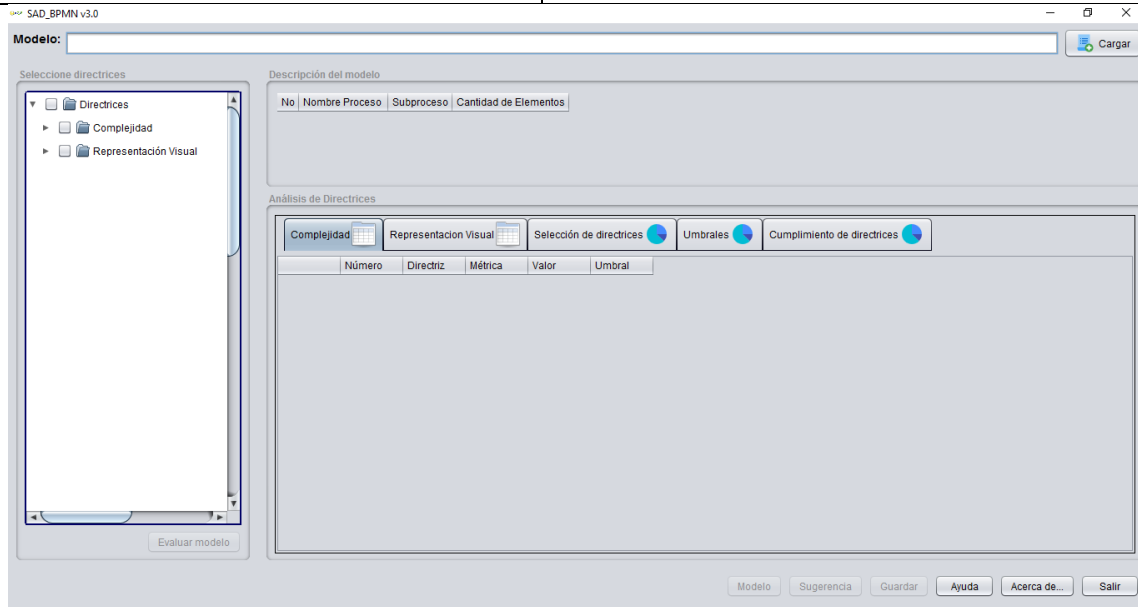
Caso de uso del sistema	Descripción
Exportar datos	Este caso de uso responde a la necesidad que tiene el modelador (actor del sistema) de exportar los datos brindados en las tablas de análisis de las métricas.
Mostrar modelo de proceso de negocio	El modelador elige en el sistema la opción de visualizar modelo, lo que permitirá que se pueda observar el modelo que está siendo evaluado.

A continuación, se muestran en las tablas 2.3, 2.4, 2.5, 2.6 y 2.7 las descripciones de los casos de uso más significativos. Descripciones más detalladas que las brindadas en la tabla 2.2.

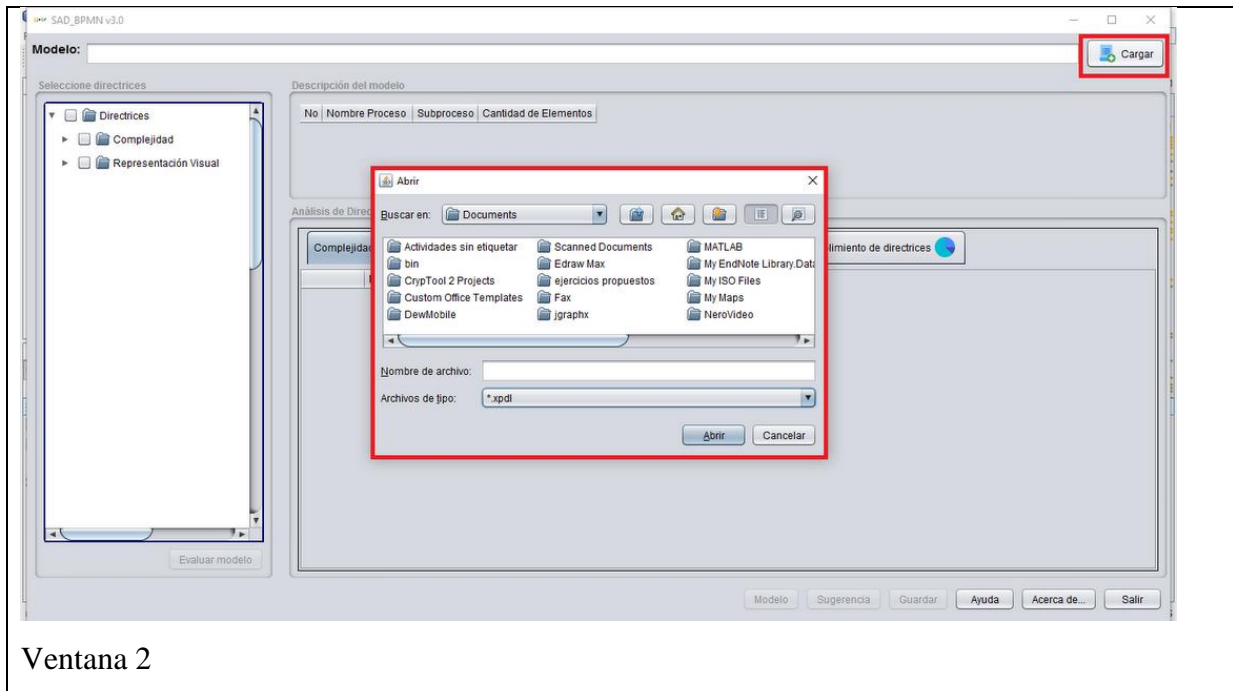
Tabla 2.3 Descripción del caso de uso *Cargar modelo de proceso de negocio*

Caso de uso del sistema	Cargar modelo de proceso de negocio
Actores	Modelador
Propósito	Cargar modelo de proceso de negocio
Resumen	El caso de uso comienza cuando el modelador desea validar un modelo de proceso de negocio en formato .xpd o .bpmn, y para ello debe seleccionar la opción abrir y cargar el modelo.
Responsabilidades	Cargar modelo de proceso de negocio
Casos de uso asociados	-
Requisitos especiales	-
Precondiciones	-
Descripción	
Ventana principal	
Cargar	
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema
1. El modelador selecciona la opción de cargar modelo.	2. El sistema muestra el cuadro de diálogo que le permite cargar el archivo.
3. Se selecciona la extensión del fichero que desea cargar.	
4. El modelador seleccionar el archivo y da la opción abrir.	
	5. Validar archivo.
	6. Cargar modelo de archivo.

	7. Habilitar opciones de visualizar modelo y evaluar métricas.
Flujos alternativos	
	6. Mostrar cuadro de diálogo notificando al usuario que el archivo no posee la extensión de fichero correcta.
	6. Mostrar cuadro de diálogo notificando al usuario que el archivo no fue generado por las herramientas Bonita o Bizagi.
Post condiciones	El sistema cargó correctamente el archivo.
Pantallas asociadas	



Ventana 1

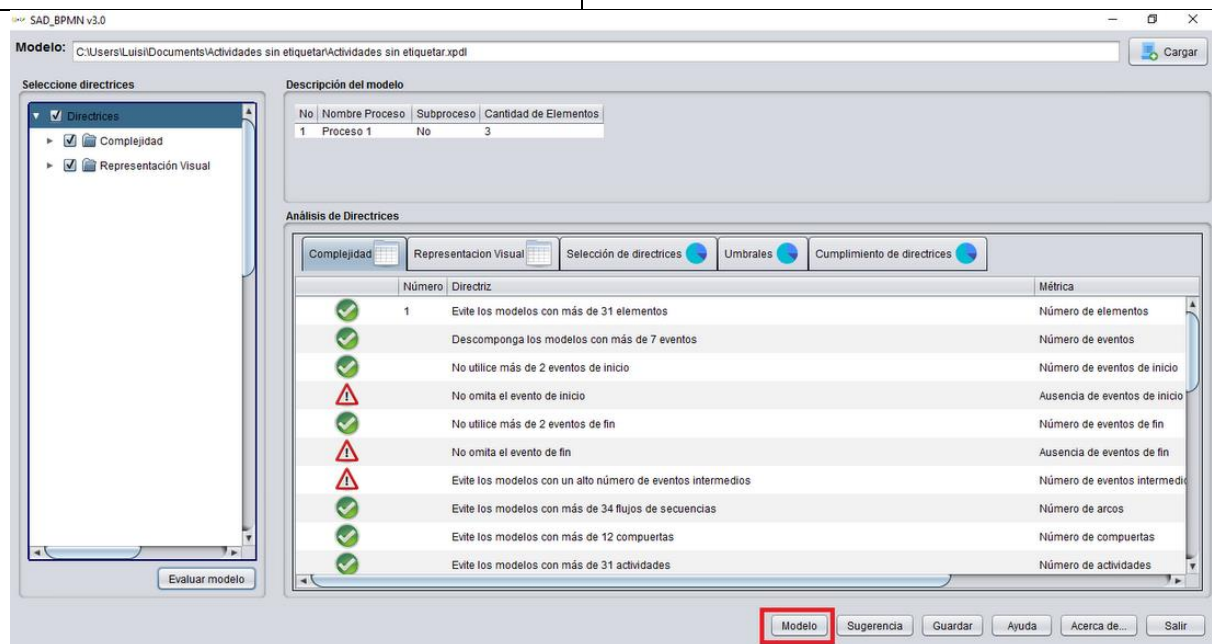


Ventana 2

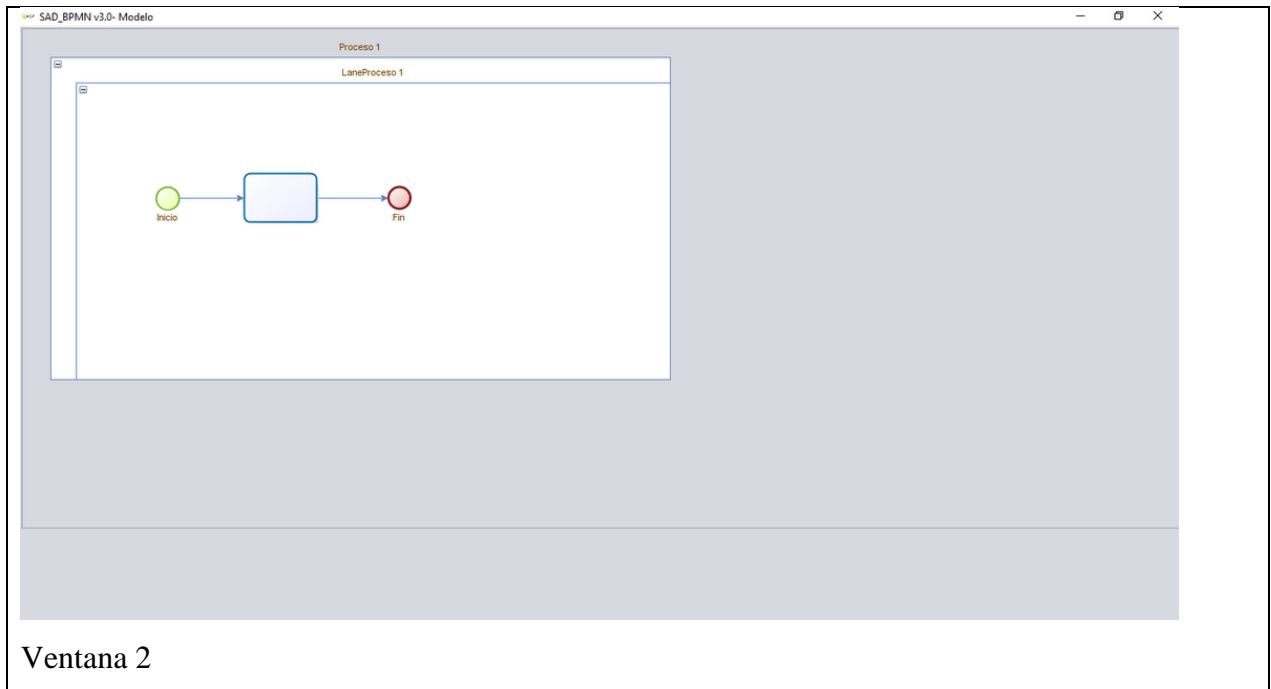
Tabla 2.4 Descripción del caso de uso *Mostrar modelo de proceso de negocio*

Caso de uso del sistema	Mostrar modelo de proceso de negocio
Actores	Modelador
Propósito	Mostrar modelo de proceso de negocio
Resumen	El modelador elige en el sistema la opción de visualizar modelo, lo que permitirá que se pueda observar el modelo cargado.
Responsabilidades	Mostrar modelo de proceso de negocio
Casos de uso asociados	-
Requisitos especiales	-
Precondiciones	Cargar modelo
Descripción	
Ventana principal	
Modelo	
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema

1. El modelador selecciona la opción de visualizar modelo de proceso de negocio.	2. El sistema muestra una ventana con el modelo de procesos de negocio.
Flujos alternativos	
Post condiciones	El sistema mostró el modelo de proceso de negocio correctamente.
Pantallas asociadas	



Ventana 1

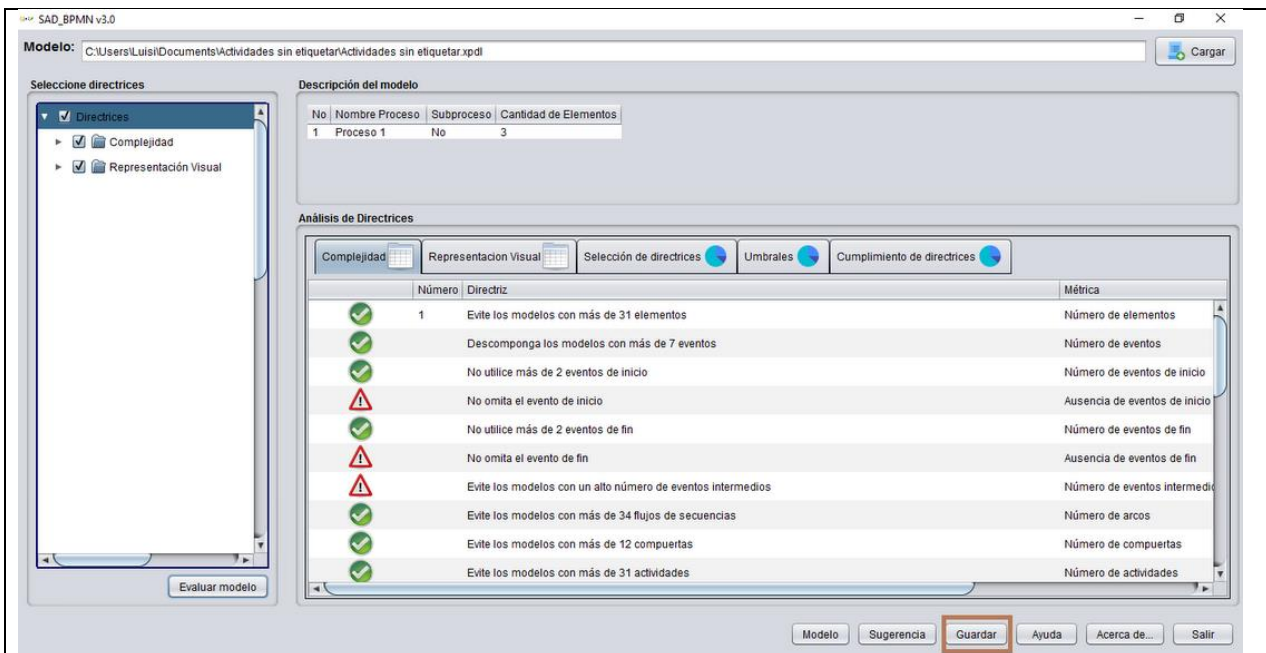


Ventana 2

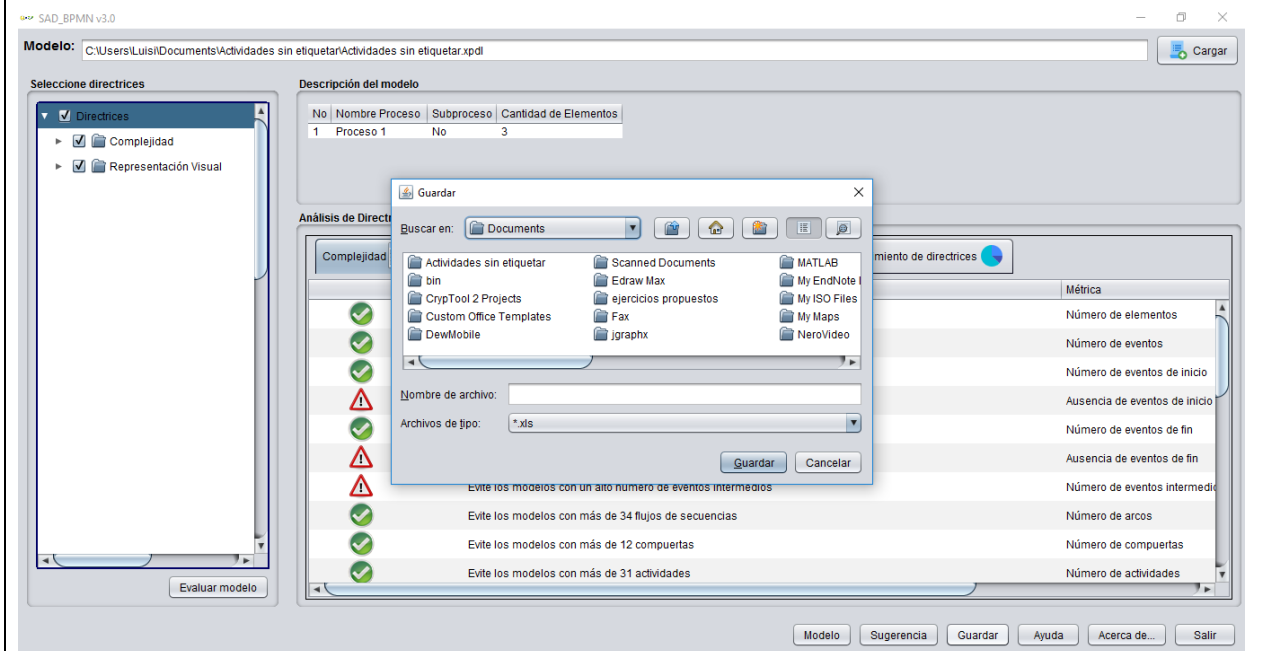
Tabla 2.5 Descripción del caso de uso *Exportar datos*

Caso de uso del sistema	Exportar datos
Actores	Modelador
Propósito	Exportar datos
Resumen	Este caso de uso responde a la necesidad que tiene el modelador (actor del sistema) de exportar los datos brindados en las tablas de análisis de las métricas.
Responsabilidades	Exportar datos
Casos de uso asociados	-
Requisitos especiales	-
Precondiciones	Cargar modelo Calcular métricas
Descripción	
Ventana principal	
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema

<p>1. El modelador selecciona la opción de guardar datos.</p> <p>4. Aceptar notificación.</p> <p>6. Elegir la opción que permita abrir el archivo.</p>	<p>2. El sistema muestra una ventana (cuadro de diálogo) para permitir al modelador que seleccione el nombre del archivo que contendrá los datos.</p> <p>3. Notificar al usuario que se ha exportado el archivo satisfactoriamente.</p> <p>5. Mostrar cuadro de diálogo para preguntar al usuario si desea abrir el archivo generado.</p> <p>7. Abrir archivo.</p>
Flujos alternativos	
<p>4. Aceptar notificación.</p>	<p>3. Notificar al usuario que no se ha exportado el archivo satisfactoriamente porque ocurrió un error.</p>
<p>6. Elegir la opción que no permita abrir el archivo, cancelar abrir archivo generado.</p>	
<p>Post condiciones</p>	<p>El sistema exportó los datos del análisis de las métricas correctamente.</p>
<p>Pantallas asociadas</p>	



Ventana 1



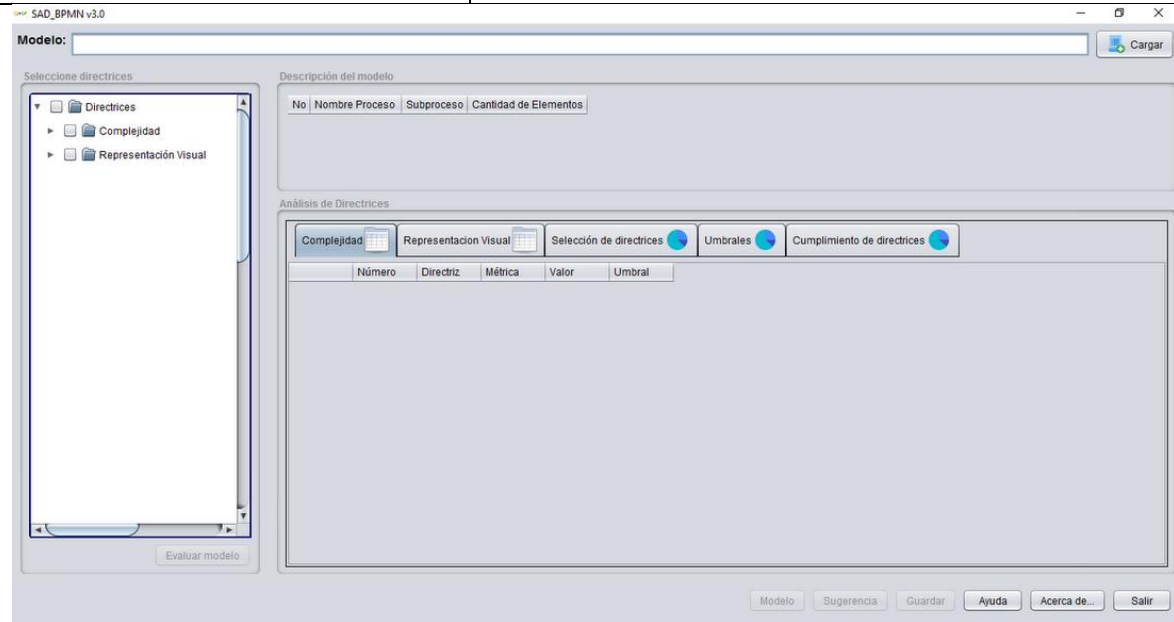
Ventana 2

Tabla 2.6 Descripción del caso de uso *Calcular métricas de calidad*

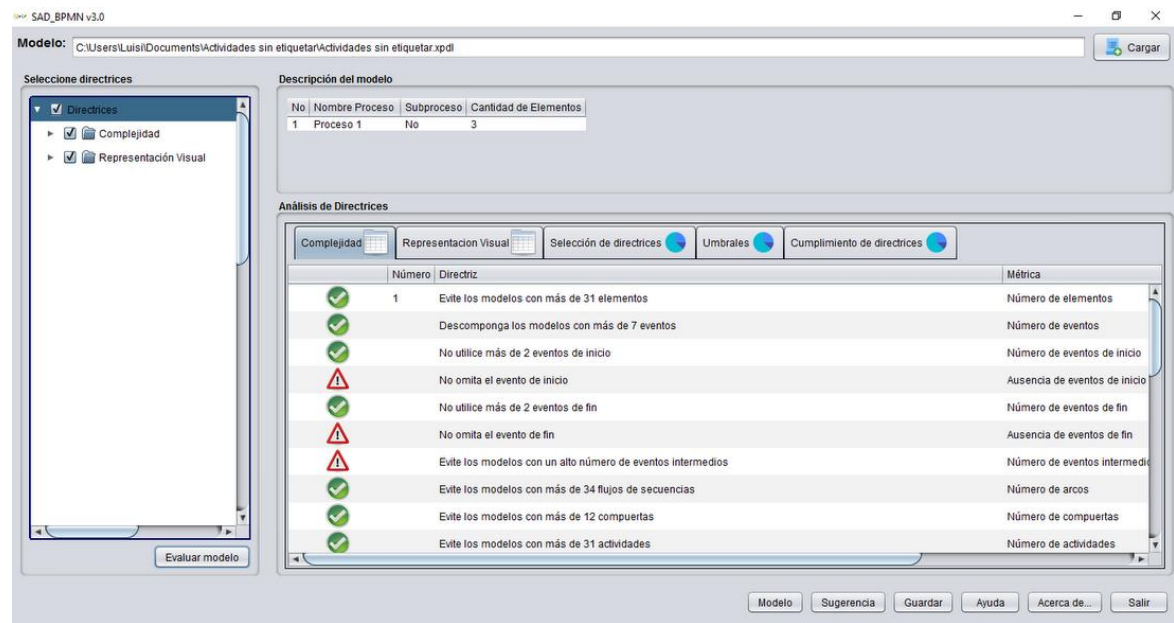
Caso de uso del sistema	Calcular métricas de calidad
Actores	Modelador
Propósito	Calcular métricas de calidad
Resumen	El modelador elige en el sistema la métrica que desea aplicar a su modelo, selecciona la opción evaluar modelo y obtiene sugerencias de posibles cambios.
Responsabilidades	Calcular métricas de calidad
Casos de uso asociados	-
Requisitos especiales	-
Precondiciones	Cargar modelo
Descripción	
Ventana principal	
Flujo normal de los eventos	
Acción del actor	Respuesta del sistema
1. El modelador selecciona las métricas que desea evaluar. 2. Selecciona la opción de evaluar.	3. Validar que se haya seleccionada al menos una métrica a evaluar. 4. Calcular métricas. 5. Mostrar datos en tablas y gráficos. 6. Habilitar opciones de sugerencia y guardar datos.
Flujos alternativos	
6. Aceptar notificación.	5. Notificar al usuario que no se ha seleccionado ninguna métrica de calidad.

Post condiciones	El sistema calculó correctamente las métricas de calidad seleccionadas.
------------------	---

Pantallas asociadas	
---------------------	--



Ventana 1



Ventana 2

2.8 Conclusiones parciales

En este capítulo se definió todo lo relacionado a los temas de modelo de negocio y requisito. Se describe a SAD_BPMN v3.0 mediante los diagramas de casos de uso, de los actores y de los requisitos tanto funcionales como no funcionales. Todo lo anterior contribuye a la documentación de la aplicación.

Capítulo III. Descripción de la propuesta de solución

El software SAD_BPMN proporciona una herramienta para la evaluación de modelos de procesos de negocio en BPMN, tomando como base directrices prácticas de calidad relacionadas con el tamaño y la morfología de los modelos. En el presente capítulo se presentan algunos aspectos de la descripción de SAD_BPMN v3.0 así como el diagrama de paquetes y clases de diseño necesarios para la implementación de las nuevas funcionalidades.

3.1 Arquitectura del sistema

La arquitectura de software se conoce por ser el conjunto de formas y guías generales, que indican la estructura, funcionamiento e interacción entre las partes del software. Se le conoce por Arquitectura lógica y se considera el diseño de más alto nivel de la estructura de un sistema, por lo que describe diversos aspectos del software.

Para la realización del presente software se eligió como arquitectura de software al patrón Modelo-Vista-Controlador (MVC, por sus siglas en inglés). De acuerdo a (Álvarez 2014), MVC es útil para cualquier desarrollo en el que intervengan interfaces de usuario y considera que “su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, *Model, Views & Controllers*”.

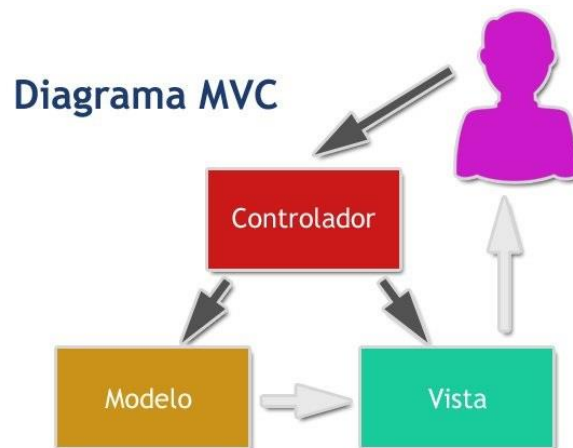


Figura 3.1 Patrón MVC (Fuente: (Robles 2013))

Según (Robles 2013), “El patrón de diseño (de software) MVC se encarga de separar la lógica de negocio de la interfaz de usuario y es el más utilizado en aplicaciones web, framework, etc, ya que facilita la funcionalidad, mantenibilidad, y escalabilidad del sistema, de forma

cómoda y sencilla, a la vez que ayuda no mezclar lenguajes de programación en el mismo código”.

En la figura 2.5 se observa la estructura de los paquetes del sistema, en la cual no se observan las tres capas clásicas del patrón MVC, esto no quiere decir que por ende el patrón no se cumpla o no sea el correctamente seleccionado, pues en cada uno de los paquetes o en la unión de varios o todos como sistema se complementan de tal forma que siempre se cumple con la arquitectura de software elegida.

3.2 Implementación de las directrices de calidad

En esta herramienta se han implementado, tal y como se explica en el capítulo II, directrices con problemas y sin programar de SAD_BPMN v2.0 y las “directrices nuevas”. A continuación, se proponen dos algoritmos, cada uno relacionado por cada tipo de directriz mencionado anteriormente. Ambos algoritmos relacionados con directrices de representación visual. El primer algoritmo responde a la directriz “*Longitud variada de flujos de secuencia*”, mientras el segundo corresponde a la directriz “*Maximizar el número de objetos conectores dibujados ortogonalmente*”.

Algoritmo 1: getLDList (Pool p), lista todos los flujos de secuencia (arcos) del modelo de longitud no uniforme, recibe como parámetro el pool que se desea analizar. Este algoritmo devuelve una lista con los flujos de secuencia de longitud no uniforme, es decir, lista los flujos de secuencia que no poseen la longitud que más se manifieste en los flujos de secuencia del modelo.

Los pasos del algoritmo son:

Paso 1. distancia<-0

Paso 2. moda<-0

Paso 3. ArrayList<Punto> edges<-p.getFlow()

Paso 4. ArrayList< Flow > flujostemp<-ArrayList<Flow>()

Paso 5. ArrayList< Flow > flujos<-ArrayList<Flow>()

Paso 6. ArrayList< Double > distancias<-ArrayList<Double>()

Paso 7. para i<-0 hasta totalEdges incrementar i

```

        hacer
            distancia<-0
            ArrayList<Punto>puntos
edges.get(i).getPuntos()
            para j<-0 hasta totalPuntos incrementar j
                hacer
                    si (j>0) entonces
                        Punto p2<-puntos.get(j)
                        Punto p1<- puntos.get(j-1)
                        distancia+=longitudRecta(p1,p2)
                    fin si
                fin para
                distancias.add(distancia)
                flujostemp.add(edges.get(i))
            fin para

```

Paso 8. List<Double> distanciascopy <- distancias.subList(0, distancias.size())

Paso 9. getModal(distanciascopy, 0, nulo)

Paso 10. si (totalDistancascopy > 1) entonces

```

    moda <- distanciascopy.get(0)
    si (distanciascopy.get(1)==1) entonces
        flujos.add(sublista edges (0, totalEdges-1))
    sino
        para cada Flow f en edges hacer
            si (longitudFlujo(f)!=moda) entonces
                flujos.add(f)
            fin si
        fin para
    fin si
fin si

```

Paso 11. Retornar flujos

A continuación, se detallan los pasos del algoritmo:

Paso 1. Se inicializa la variable **distancia** en cero. La variable **distancia** es la encargada de almacenar el cálculo de las distancias de cada uno de los flujos de secuencia.

Paso 2. Se inicializa la variable **moda** en cero. La variable **moda** es la encargada de almacenar el cálculo de la moda de cada una de las distancias de los flujos de secuencia.

Paso 3. La variable **edges** recibe la lista de todos los arcos del modelo.

Paso 4. Se instancia una lista de flujos, **flujostemp**. En esta variable se almacenarán los arcos que contenga un proceso. Es una copia de **flujos**.

Paso 5. Se instancia una lista de flujos, **flujos**. En esta variable se almacenarán los arcos que contenga un proceso.

Paso 6. Se instancia una lista de double (numéricos), **distancias**. En esta variable se almacenarán las distancias de cada uno de los arcos.

Paso 7. En este paso se recorren los flujos de secuencia del modelo, se le calcula la distancia a cada uno de los mismos al recorrer cada uno de los puntos que posee, la distancia se adiciona al listado de distancias **distancias**. Se adiciona al flujo que se le calculó la distancia a **flujostemp**. De esta forma, al recorrer cada flujo en **flujostemp**, a cada flujo *i* se le asocia una distancia *i* en **distancias**.

para $i < 0$ hasta totalEdges incrementar *i*.

Paso 8. Se crea una copia de **distancias** en una nueva lista nombrada **distanciacy**.

Paso 9. Se calcula la moda de las distancias, es decir, se determina la distancia que más se repite.

Paso 10. Si el valor modal mayor fue 1, es porque no existió uniformidad del tamaño de los flujos de secuencia por lo que se devuelven todos los flujos de secuencia excepto 1, ¿por qué

excepto 1?, porque al menos uno de todos éstos será uniforme desde el punto de vista del modelo o para el modelador.

Si la cantidad de elementos con la moda es mayor que 1, entonces existe cierto nivel de uniformidad de la longitud de los flujos de secuencia, ya sea un nivel parcial o total. Entonces, se buscan los flujos que no presenten como distancia la moda (no uniformidad) y se añaden al listado que se obtendrá como resultado.

Paso 11. Se devuelve el listado de flujos de tamaño

Algoritmo 2: orthogonallyConnectingObjects (Pool p) calcula la cantidad de arcos que están ortogonalmente modelados, recibe como parámetro el pool que se desea analizar. Este algoritmo devuelve la cantidad de objetos ortogonalmente conectados.

Los pasos del algoritmo son:

Paso 1: count<-0

Paso 2: si (p!=nulo y p.getFlow()!=nulo) entonces

Paso 2.1: para cada Flow f en p.getFlow()

 si (f.getPuntos()!=nulo && totalPuntosF>=2)

entonces

 flag<-true

 para i<-0 hasta totalPuntos incrementar i

 hacer

 Punto before<-nulo

 si (i==0) entonces

 before<-f.getPuntos().get(i)

 sino

 Punto now<- f.getPuntos().get(i)

 before<-f.getPuntos().get(i-1)

 si (before.getX() != now.getX() &&

before.getY() != now.getY())

 flag<-false

```

        fin si
    fin si
    fin para
    si (flag==true) entonces
        count<-count+1
    fin si
    fin si
    fin para
    fin si

```

Paso 3: retornar count

A continuación, se detallan los pasos del algoritmo:

Paso 1: Se inicializa y declara la variable **count** con valor 0. Esta variable será contador de los flujos que se correspondan con las características que se buscan.

Paso 2: Si existe un proceso. Validar que exista algún proceso.

Paso 2.1: Recorrer los arcos del proceso, a su vez recorrer los puntos de cada uno de éstos y hallar si el flujo es ortogonalmente dibujado o no.

Paso 3: Devolver cantidad de objetos conectores (arcos) ortogonalmente dibujados.

El **Algoritmo 2** presenta en el peor de los casos una complejidad cuadrática **O (n²)**, debido a que realiza una cantidad de **nxn** comparaciones. Esto se debe a la necesidad de comparar los puntos de cada flujo de secuencia.

3.3 Diagrama de clases del diseño

Según (Jacobson et al. 2000), “Una clase de diseño es una abstracción sin costuras de una clase o construcción similar en la implementación del sistema”. Donde a menudo esta clase aparece como un estereotipo sin costuras que se corresponde con una construcción en el lenguaje de programación dado.

Para el diagrama de casos de uso *Cargar modelo* el diagrama de clases de diseño se encuentra en (Calzada Ricet & Pairol Fuentes 2016). En las figuras 3.2 y 3.3 se muestran los diagramas de clases del diseño relacionados a cada uno de los casos de uso de sistema correspondientes.

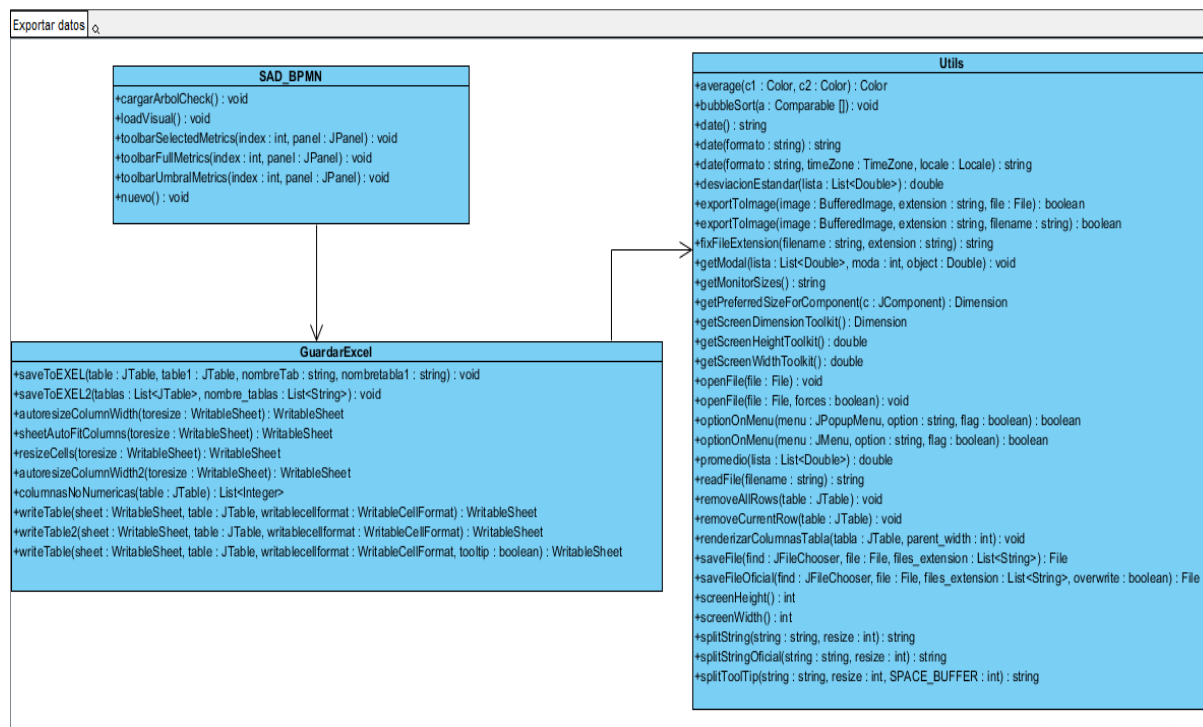


Figura 3.2 Diagrama de clases del diseño CUS Exportar datos

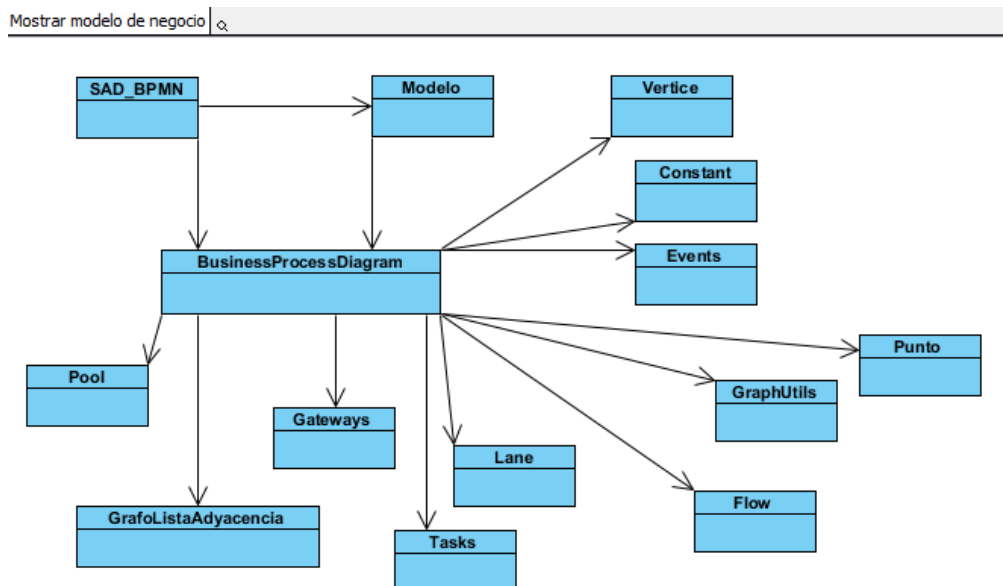


Figura 3.3 Diagrama de clases del diseño CUS Mostrar modelo de negocio

3.4 Diagrama de secuencia

De acuerdo a (Craig Larman 1999), “El diagrama de la secuencia de un sistema muestra gráficamente los eventos que fluyen de los actores al sistema. En este capítulo describiremos la forma de elaborarlos. La creación de los diagramas de la secuencia de un sistema forma parte de la investigación para conocer el sistema; se incluye, pues, dentro del modelo de análisis”.

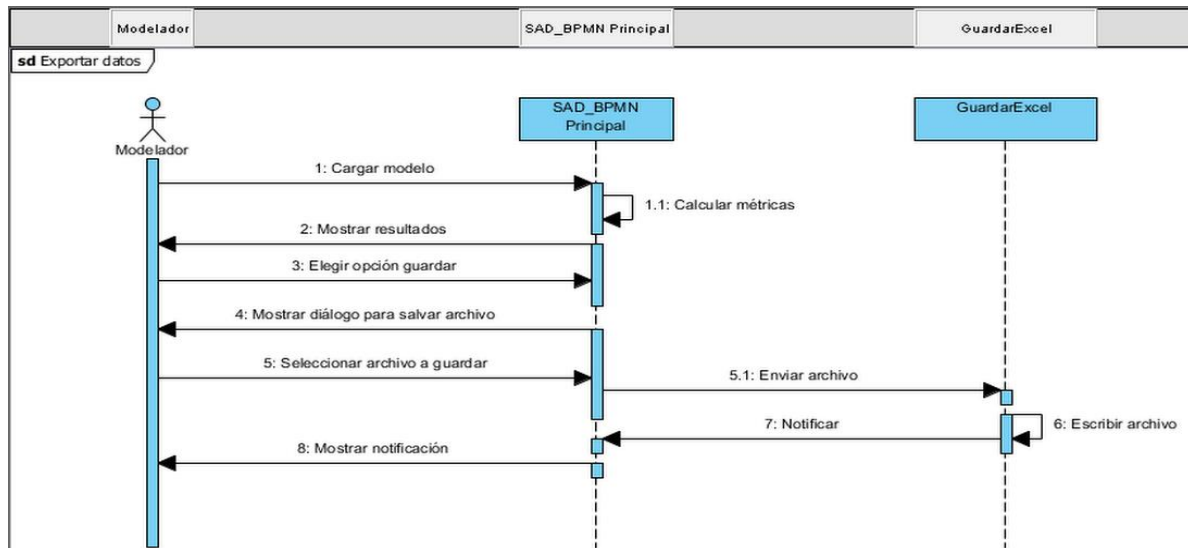


Figura 3.4 Diagrama de secuencia CUS Exportar datos

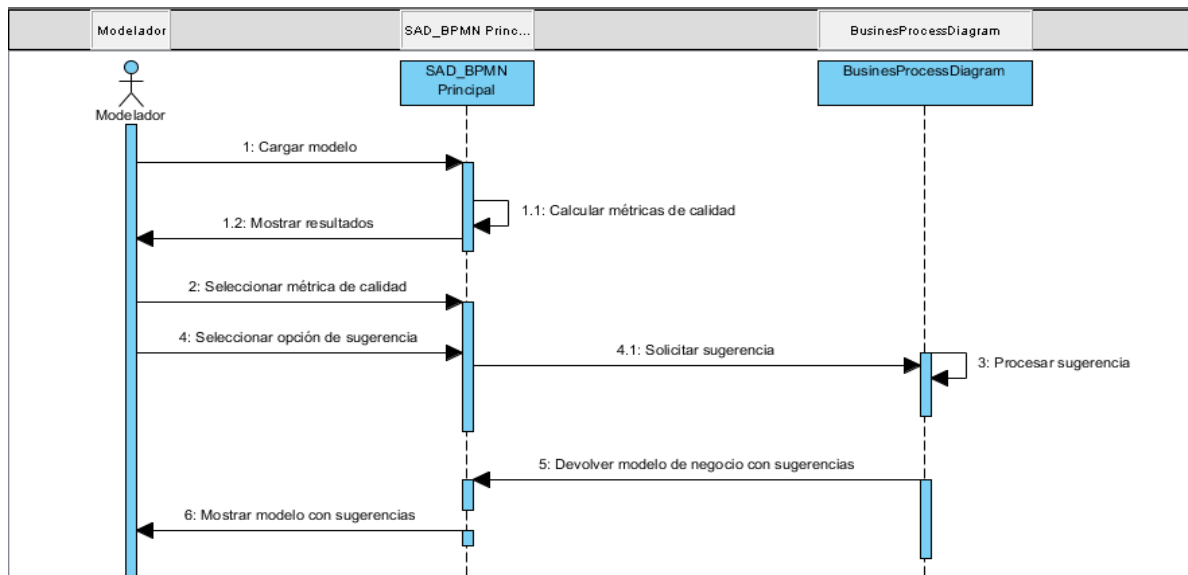


Figura 3.5 Diagrama de secuencia CUS Mostrar sugerencias visuales

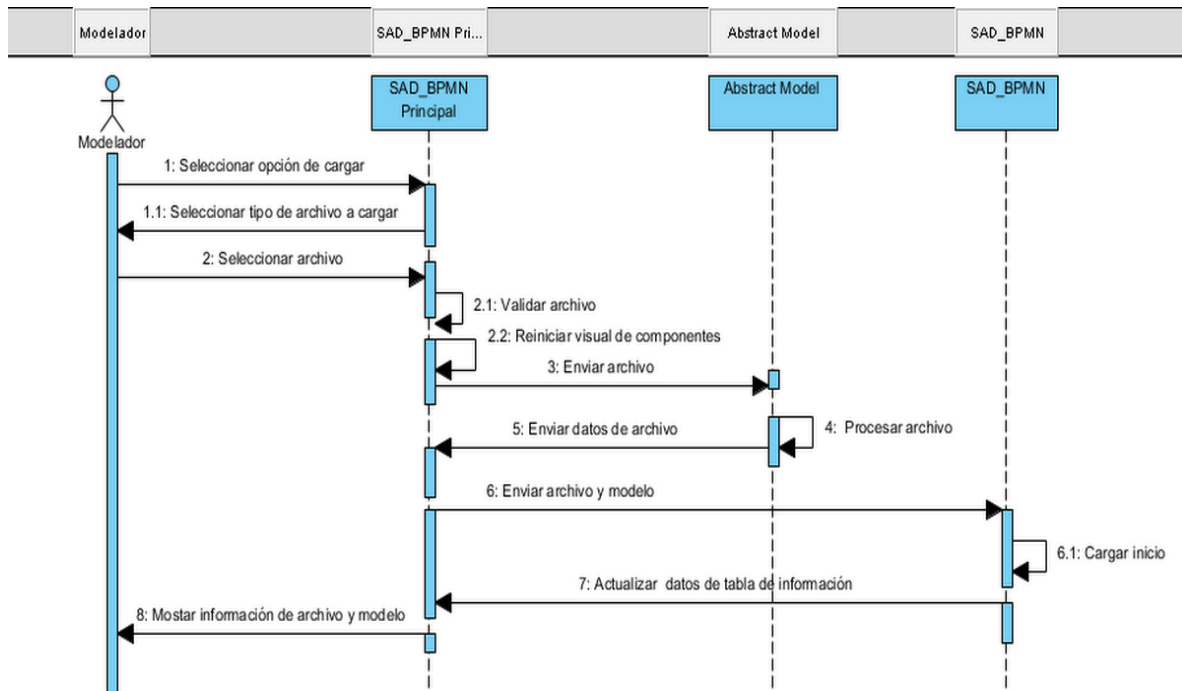


Figura 3.6 Diagrama de secuencia CUS Cargar modelo

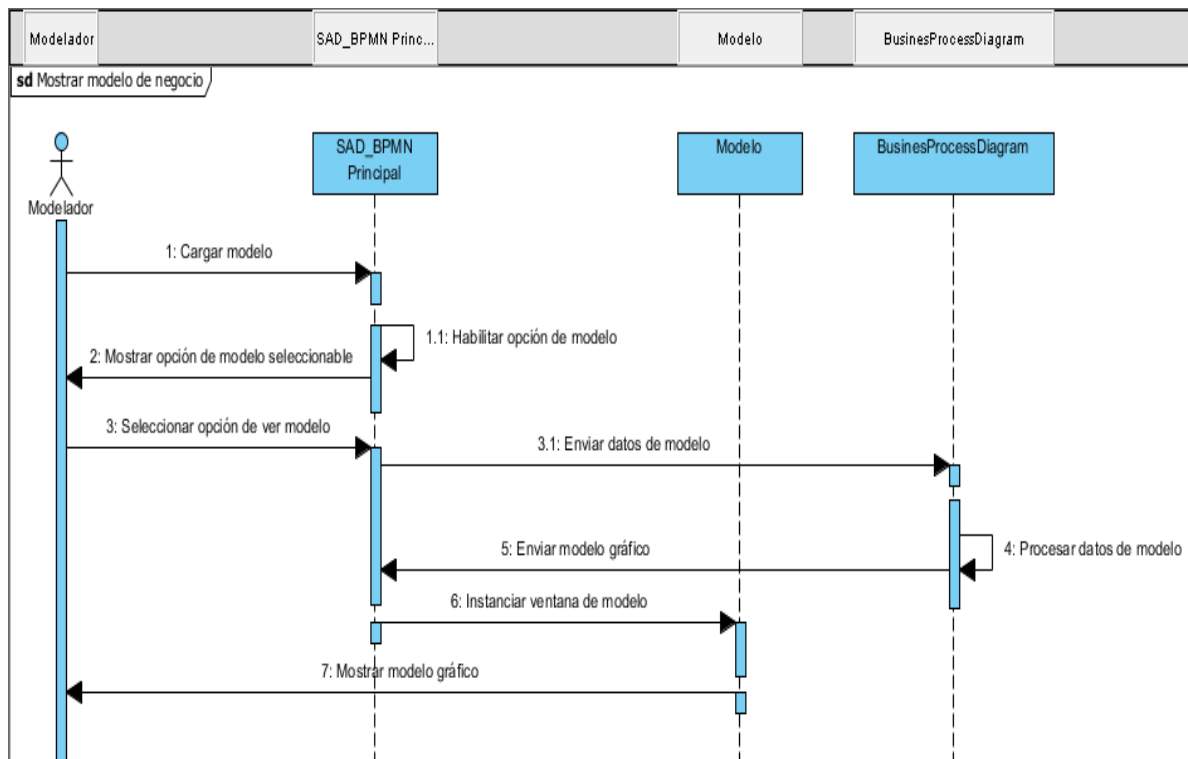


Figura 3.7 Diagrama de secuencia CUS Mostrar modelo de negocio

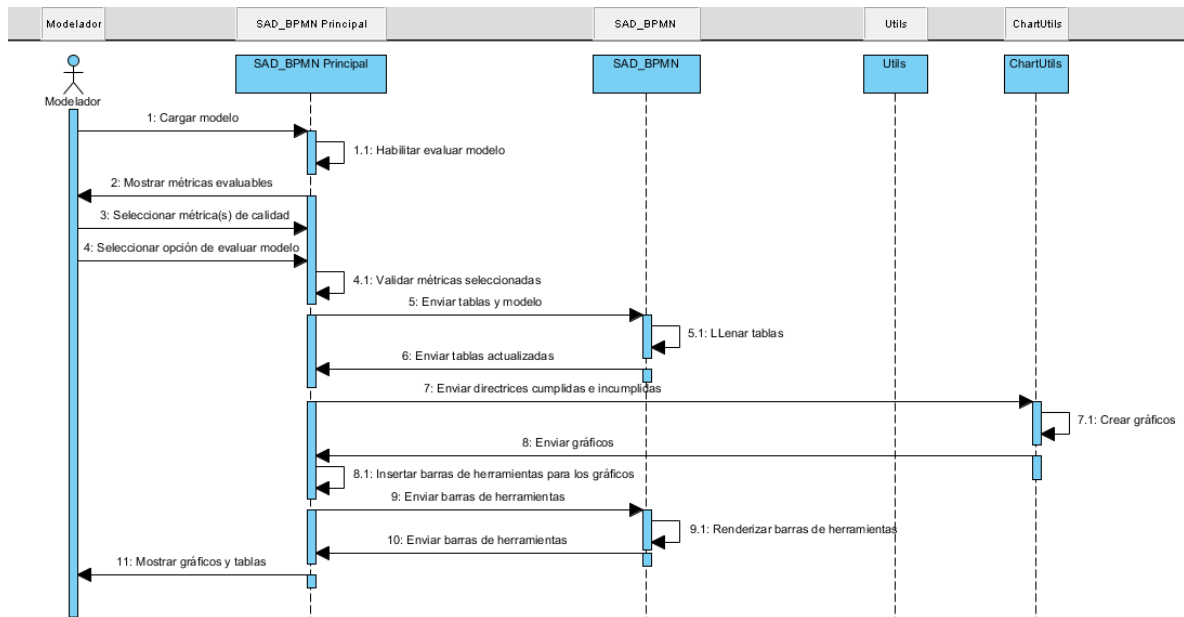


Figura 3.8 Diagrama de secuencia CUS Calcular métricas de calidad

3.5 Modelo de componentes

Para (Craig Larman 1999), “Los diagramas de componentes muestran las dependencias del compilador y del “*runtime*” entre los componentes del software; por ejemplo, los archivos del código fuente y los DLL.”

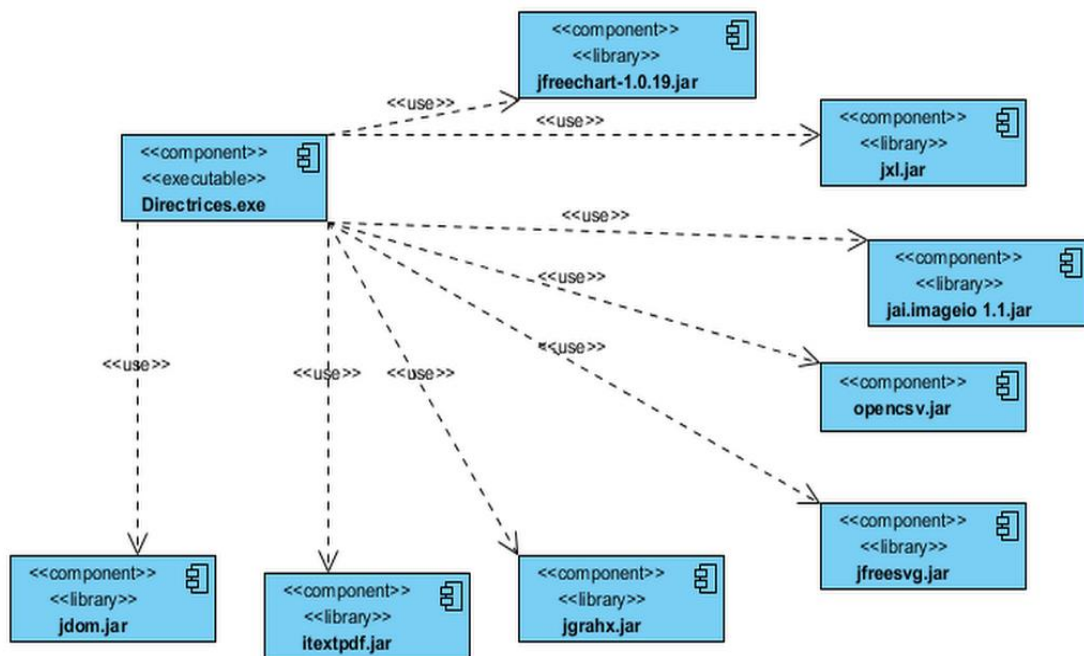


Figura 3.9 Modelo de componentes SAD_BPMN v3.0

3.6 Diagrama de despliegue

Según (Craig Larman 1999), “Los diagramas de despliegue muestran a los nodos procesadores la distribución de los procesos y de los componentes”.

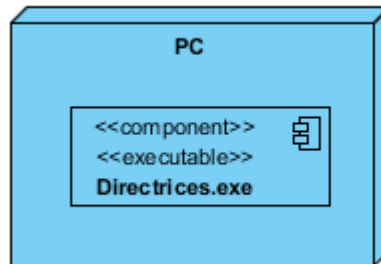


Figura 3.10 Diagrama de despliegue SAD_BPMN v3.0

3.7 Tratamiento de errores

La colección y análisis de los errores es la única manera de mejorar es medir cómo se está haciendo algo. El aseguramiento de la calidad del software reúne y analiza errores y datos acerca de los defectos para entender mejor cómo se cometen los errores y qué actividades de la ingeniería de software son más apropiadas para eliminarlos (Pressman 2010).

En SAD_BPMN v3 se capturan los errores y se notifica al usuario. A continuación, se muestran algunas figuras de ejemplo que muestran lo anterior.

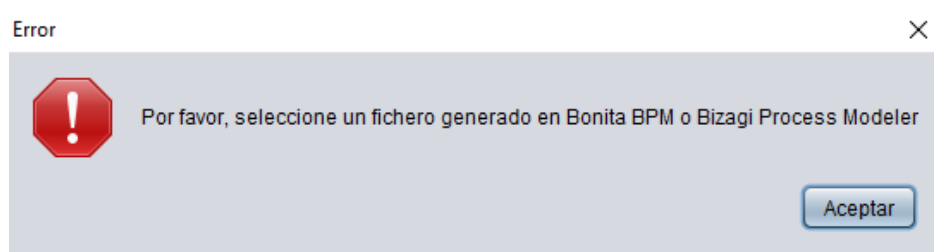


Figura 3.11 Notificación de error al intentar cargar un fichero de extensión incorrecta

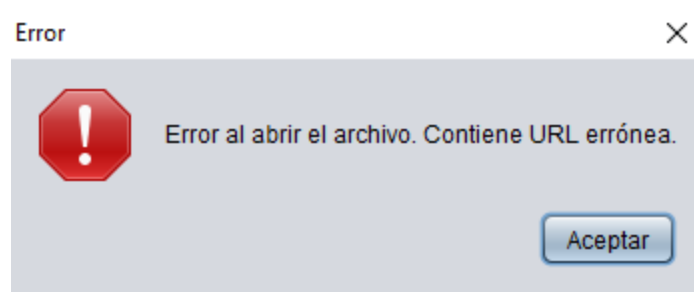


Figura 3.12 Notificación de error al intentar cargar un modelo XML con formato erróneo

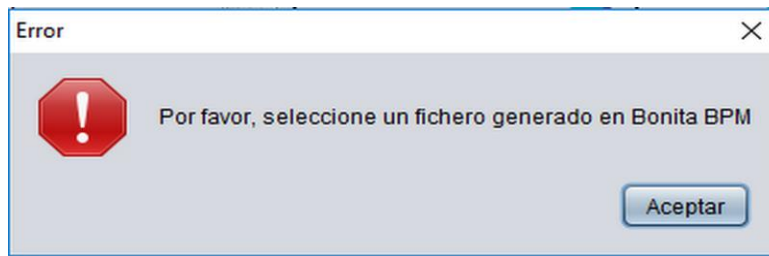


Figura 3.13 Notificación de error al intentar cargar un modelo de extensión correcta (.bpmn) pero generado por una herramienta no soportada

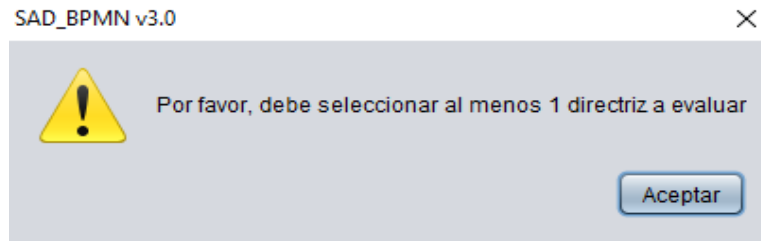


Figura 3.14 Notificación de atención generada porque se desea evaluar un modelo y no se ha seleccionado al menos una métrica para ello

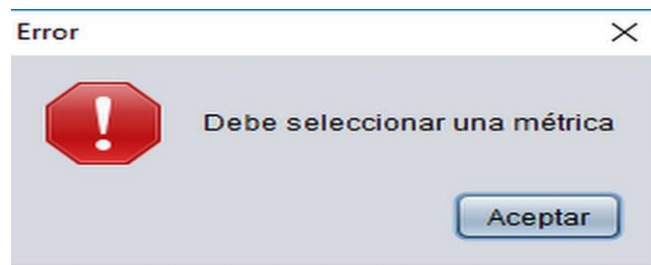


Figura 3.15 Notificación de atención generada porque se desea ver sugerencias visuales para el modelo y no se ha seleccionado al menos una métrica para ello

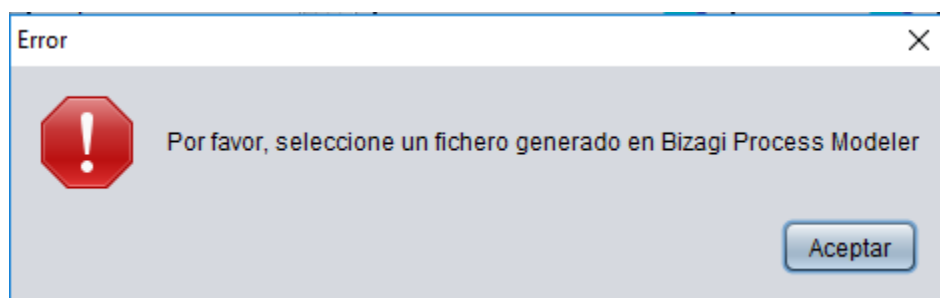


Figura 3.16 Notificación de error al intentar cargar un modelo de extensión correcta (.xpd) pero generado por una herramienta no soportada

En SAD_BPMN v3.0 cuando ocurra un error se abortará la función que se esté realizando en el instante de tiempo, mientras que en el caso de los mensajes de advertencia evitarán realizar la función que se desee hasta que no se tengan las condiciones debidas.

El tratamiento de errores en esta herramienta también se manifiesta con la captura y validación de diversas excepciones, algunas de estas excepciones son:

- `NullPointerException`: Para validar cuando se encuentren valores o parámetros nulos, y, por ende, para no realizar operaciones sobre los mismos.
- `ArrayOutOfBoundsException`: Esta excepción se captura para no recorrer mediante índices estructuras de datos (arreglos, listas enlazadas, listas de adyacencias, matrices, etc.) o elementos visuales tal como tablas, que no posean índices porque no contiene información o elementos.
- `IOException`: La captura de este tipo de excepción nos permitirá capturar errores que se produzcan con la entrada y salida de ficheros, tal como para los casos de uso *Cargar modelo* y *Exportar datos*.
- `MalformedURLException`: Los modelos que se cargan son archivos con una estructura XML personalizada por las herramientas con que se generó. Internamente estos modelos pueden contener URL erróneas. Con la validación de este tipo de error se puede definir una estrategia para enfrentarlo.
- `ParserConfigurationException`: Al validar este error definiremos la forma de actuar con respecto a los modelos que no tienen bien definida su estructura XML.

3.8 Conclusiones parciales

SAD_BPMN v3.0 es una herramienta basada en Java que permite la evaluación de modelos de procesos de negocios generados por diversas herramientas de modelación de procesos de negocio. Como se ha podido apreciar en el presente capítulo SAD_BPMN posee a MVC como arquitectura de sistema y fuerte validación de errores, la cual se activa y muestra mensajes automáticamente.

Capítulo IV. Pruebas y análisis de factibilidad

En el presente capítulo se presentan las pruebas y estimación de esfuerzo. Realizar pruebas a SAD_BPMN v3.0 le dará un mayor soporte y pondrá su rendimiento en una fase de determinar si existen dudas.

4.1 Planificación y estimación

“El objetivo de la planificación del proyecto de software es proporcionar un marco conceptual que permita al gerente hacer estimaciones razonables de recursos, costo y calendario. Además, las estimaciones deben intentar definir los escenarios de mejor caso y peor caso, de modo que los resultados del proyecto puedan acotarse. Aunque hay un grado inherente de incertidumbre, el equipo de software se embarca en un plan que se haya establecido como consecuencia de dichas tareas. Por tanto, el plan debe adaptarse y actualizarse conforme avanza el proyecto. En las siguientes secciones se estudia cada una de las acciones asociadas con la planificación del proyecto de software” (Pressman 2010).

4.1.1 Planificación basada en uno de los métodos de estimación

Este método utiliza los casos de estudio como dato de entrada para calcular el esfuerzo en horas/hombre (hh), donde luego de realizado el cálculo se puede llegar a otros datos.

- 1. Calcular los Puntos de Casos de Uso sin Ajustar (UUCP). Este valor se calcula a partir de la siguiente expresión:**

$$UUCP = UAW + UUCW \text{ donde,}$$

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

- a) Determinar el factor de peso de los actores sin ajustar (UAW).**

El factor de peso de los actores sin ajustar consiste en la evaluación de la complejidad de los actores con los que tendrá que interactuar el sistema. Este puntaje se calcula determinando si cada actor es una persona u otro sistema, y la forma en la que este interactúa con el caso de uso y la cantidad de actores de cada tipo como se muestra a continuación en la Tabla 4.1.

Tabla 4.1 Factores de peso de los actores

Tipo de actor	Descripción	Factor de peso	Número de actores	Resultado
Simple	Otro sistema que interactúa con el sistema mediante una interfaz de programación.	1	0	0
Promedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1	3
Total				3

Finalmente, hay que contar el número de actores que hay en el sistema, multiplicar cada tipo por su factor de peso y sumar esos productos para obtener el total (AUW). De forma tal que $UAW = 3$

b) Determinar el factor de peso en los casos de uso sin ajustar (UUCW).

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómicas, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Tabla 4.2 Factores de pesos de los casos de uso

Tipo de caso de uso	Descripción	Factor de peso	No. de transacciones por CUS	Resultado
Simple	1-3 Transacciones	5	1	5
Promedio	4-7 Transacciones	10	1	10
Complejo	Mayor de 8	15	4	60
Total				75

De manera que $UUCW = 75$

Por tanto:

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 75$$

$$UUCP = 78$$

2. Calcular los Puntos de Casos de Uso ajustados.

Luego de calcular los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente expresión:

$$UCP = UUCP \times TCF \times EF$$

Dónde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

a) Determinar el factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Tabla 4.3 Factor de complejidad técnica

Número de factor	Descripción	Peso	Valor
T1	Sistema distribuido	2	0
T2	Tiempo de respuesta	1	0
T3	Eficiencia por el usuario	1	0
T4	Proceso interno complejo	1	5
T5	Reusabilidad	1	4
T6	Facilidad de instalación	0.5	3
T7	Facilidad de uso	0.5	3
T8	Portabilidad	0	2
T9	Facilidad de cambio	1	2
T10	Concurrencia	1	1
T11	Objetivos especiales de seguridad	1	0
T12	Acceso directo a terceras partes	1	0
T13	Facilidades especiales de entrenamiento a usuarios finales	1	0

Total

20

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\text{TCF} = 0.6 + 0.01 * \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$\text{TCF} = 0.6 + 0.01 * 20$$

$$\text{TCF} = 0.8$$

b) Determinación del factor ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

Tabla 4.4 Factor de ambiente

No. del factor	Descripción	Peso	Valor	Factor
E1	Familiaridad con el modelo del proyecto usado.	1.5	1	1.5
E2	Experiencia en la aplicación	0.5	1	0.5
E3	Experiencia en orientación a objetos	1	2	2
E4	Capacidad de analista líder	0.5	0	0
E5	Motivación	1	2	2
E6	Estabilidad de los requisitos	2	1	2
E7	Personal media jornada	-1	0	0
E8	Dificultad del lenguaje de programación	-1	4	-4

El factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 4$$

$$EF = 1.28$$

De manera que el Cálculo de los Puntos de Casos de Uso Ajustados es como sigue:

$$UCP = UUCP * TCF * EF$$

$$UCP = 78 * 0.8 * 1.28$$

$$UCP = 79.872$$

Cálculo del esfuerzo

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Dónde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de casos de uso ajustados

CF: Factor de conversión (20 horas-hombre por defecto)

$$E = 79.872 * 20$$

$E = 1597,44$ horas-hombre

Para la obtención de una estimación más exacta de la duración del proyecto, se hace necesario agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las restantes actividades que se llevaron a cabo durante el desarrollo del software; así la distribución del esfuerzo entre dichas actividades está dada por la siguiente aproximación:

Tabla 4.5 Distribución genérica del esfuerzo

Actividad	Porcentaje
Análisis	10%
Diseño	10%
Programación	40%
Pruebas	20%
Sobrecarga(otras actividades)	20%

Con este criterio y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Tabla 4.6 Distribución real del esfuerzo

Actividad	Horas/hombres
Análisis	399.36
Diseño	399.36
Programación	1597.44
Pruebas	798.72
Sobrecarga (otras actividades)	798.72
Total	3993.6

Cálculo del esfuerzo total

$E_{Total} = 3993.6$ horas /hombre

Cálculo del tiempo de desarrollo

$T_{Desarrollo} = E_{Total}/CH_{Total}$ CH_{Total} : Cantidad de hombres = 1

$T_{Desarrollo} = 3993.6$ horas

Considerando que se trabajan 8 horas diarias:

$T_{Desarrollo} = T_{Desarrollo}/8$ horas por/día

$$T_{\text{Desarrollo}} = 3993.6/8$$

$$T_{\text{Desarrollo}} = 499.2 \text{ días aproximadamente}$$

Cálculo del costo

$$\text{CostoTotal} = E_{\text{Total}} * 2 * TH \quad TH: \text{Tarifa horaria} = \$2.403$$

$$\text{CostoTotal} = 3993.6 * 2 * 2.40$$

$$\text{CostoTotal} = \$ 19169.28$$

4.1.2 Estimación de costes mediante COCCOMO I

COCOMO viene por sus siglas en inglés de *Constructive Cost Model* (Modelo constructivo de costes). Se caracteriza por ser una jerarquía de modelos de estimación de costes de software. Fue desarrollado por B. W. Boehm en 1981 detallándolo profundamente en su libro "*Software Engineering Economics*".

De acuerdo a (Rene 2015), “Es un modelo que permite estimar el costo, el esfuerzo, y el horario en la planificación de un nuevo proyecto de desarrollo de software. Este modelo empírico se obtuvo recopilando datos de grandes proyectos, los cuales fueron analizados para descubrir las fórmulas que mejor se ajustaban; estas fórmulas relacionan el tamaño del sistema y del producto, factores del proyecto y del equipo con el esfuerzo necesario para desarrollar el sistema”.

Tipo de Proyecto de Software: Orgánico

Orgánico: proyectos relativamente sencillos, menores de 50 KDLC (Kilo-líneas de código) líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.

Modelo de Cocomo elegido:

Modelo intermedio: Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.

Fórmulas.

$$\text{Esfuerzo (E)} = a \text{ KLDC } e * \text{FAE (persona x mes)} \quad (4.1)$$

$$\text{Tiempo de duración del desarrollo (T)} = c \text{ Esfuerzo } d \text{ (meses)} \quad (4.2)$$

$$\text{Personal (P)} = E/T \text{ (personas)} \quad (4.3)$$

Donde:

- ▶ E es el esfuerzo requerido por el proyecto, en persona-mes
- ▶ T es el tiempo requerido por el proyecto, en meses
- ▶ P es el número de personas requerido por el proyecto.
- ▶ a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo.

- ▶ KLDC es la cantidad de líneas de código, en miles.
- ▶ FAE Es un multiplicador que depende de 15 atributos.

LDC: 24 715

El tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, pero el proyecto es medianamente complejo, por lo consiguiente se usará el modelo intermedio, con los coeficientes del modo orgánico siguientes: el tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, pero el proyecto es medianamente complejo, por lo consiguiente se usará el modelo intermedio, con los coeficientes del modo orgánico siguientes:

a= 3,2 e= 1,05 c= 2,5 d= 0,38

Tabla 4.7 Conductores de coste

Conductores de coste	Valoración					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	-
Tamaño de la base de datos	-	0,94	1.00	1,08	1,16	-
Complejidad del producto	0,70	0,85	1.00	1,15	1,30	1,65
Restricciones de tiempo de ejecución	-	-	1.00	1,11	1,30	1,66
Restricciones del almacenamiento principal	-	-	1.00	1,06	1,21	1,56
Volatilidad de la máquina virtual	-	0,87	1.00	1,15	1,30	-
Tiempo de respuesta del ordenador	-	0,87	1.00	1,07	1,15	-
Capacidad del analista	1,46	1,19	1.00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82	-
Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70	-
Experiencia en S.O. utilizado	1,21	1,10	1.00	0,90	-	-
Experiencia en el lenguaje de programación	1,14	1,07	1.00	0,95	-	-
Prácticas de programación modernas	1,24	1,10	1.00	0,91	0,82	-
Utilización de herramientas software	1,24	1,10	1.00	0,91	0,83	-
Limitaciones de planificación del proyecto	1,23	1,08	1.00	1,04	1,10	-

$$\text{FAE} = 1,15 * 1,15 * 1,11 * 1 * 1 * 1,07 * 0,86 * 0,82 * 0,7 * 1 * 0,95 * 1 * 0,91 * 1,08 = 0,72$$

Justificación de los valores:

Atributos de software

- **Fiabilidad requerida del software:** *Si se produce un fallo por mala validación del modelo o en la carga y/o tratamiento del modelo (Valoración Alta).*
- **Tamaño de la base de datos:** *Nuestro producto no requiere base de datos (Valoración Muy bajo).*
- **Complejidad del producto:** *La aplicación va a realizar cálculos semi complejos (Valoración Alta).*

Atributos de hardware

- **Restricciones del tiempo de ejecución:** *En los requerimientos se exige alto rendimiento (Valoración Alta).*
- **Restricciones del almacenamiento principal:** *No hay restricciones al respecto (Valoración Nominal).*
- **Volatilidad de la máquina virtual:** *Se usarán sistemas de la “Familia Windows” y utilizar máquina virtual de Java (Valoración Nominal).*
- **Tiempo de respuesta del ordenador:** *Deberá ser interactivo con el usuario (Valoración Alta).*

Atributos del personal

- **Capacidad del analista:** *Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta).*
- **Experiencia en la aplicación:** *Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).*
- **Capacidad de los programadores:** *Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).*
- **Experiencia en S.O. utilizado:** *Con Windows 2000 Professional la experiencia es a nivel usuario (Valoración Nominal).*
- **Experiencia en el lenguaje de programación:** *Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).*

Atributos del proyecto

- **Prácticas de programación modernas:** *Se usarán prácticas de programación mayormente convencional (Valoración Nominal).*
- **Utilización de herramientas software:** *Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).*
- **Limitaciones de planificación del proyecto:** *Existen pocos límites de planificación. (Valoración Baja).*

Cálculo de estimación.

- ❖ *Cálculo del esfuerzo del desarrollo:*

$$E = a \text{ KLDC } e * \text{FAE} = 3,2 * (24,715^{1,05}) * 0,72 \approx 66,85 \text{ personas /mes.}$$

- ❖ *Cálculo tiempo de desarrollo:*

$$T = c \text{ Esfuerzo } d = 2,5 * (66,85^{0,38}) \approx 12,4 \text{ meses}$$

- ❖ *Productividad:*

$$PR = \text{LDC/Esfuerzo} = 24\ 715 / 66,85 = 369,71 \text{ LDC/personas mes}$$

- ❖ *Personal promedio:*

$$P = E / T = 66,85 / 12,4 = 5,4 \text{ personas}$$

Según estas cifras será necesario un equipo de 5 personas trabajando alrededor de 12 meses (1 año), por lo que considerando que el trabajo se ha realizado por 1 persona en 4 meses, se puede decir que se han superado las expectativas y objetivos a un punto de sobrecumplimiento.

4.2 Pruebas unitarias

Para (Pressman 2010), “La prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software: un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos”.

Según (Parra Conejero 2016), “La idea de utilizar un framework de pruebas unitarias es comprobar que nuestro código no fallará en determinados casos creando clases y métodos que apliquen la lógica que hemos programado, de esta forma, nos aseguramos que nuestras aplicaciones funcionarán sin problemas en la prácticamente todos los casos”. SAD_BPMN v3.0 efectuará las pruebas unitarias mediante la librería JUnit, el cual es el entorno que nos

permitirá probar cada una de las clases. Además, vale destacar que las pruebas unitarias son conocidas también como prueba de unidad.

4.2.1 Casos de prueba

Los conceptos fundamentales a manejar es el caso de prueba (test case), y la suite de prueba (test suite). De acuerdo a (Alicante 2014), “Los casos de prueba son clases o módulos que disponen de métodos para probar los métodos de una clase o módulo concreta/o. Así, para cada clase que quisiéramos probar definiríamos su correspondiente clase de caso de prueba. Mediante las suites podemos organizar los casos de prueba, de forma que cada suite agrupa los casos de prueba de módulos que están funcionalmente relacionados. Las pruebas que se van construyendo se estructuran así en forma de árbol, de modo que las hojas son los casos de prueba, y podemos ejecutar cualquier subárbol (suite).” Los métodos de prueba en estos casos son testeados para cuando no se cargue y no se tenga ningún modelo, por lo que constantemente se estará testeando excepciones como *NullPointerException*, *ArrayOutOfBoundsException*, *FileNotFoundException* y NaN.

Tanto para las pruebas que a continuación se detallan como para la prueba de integración, se realiza teniendo como caso de estudio:

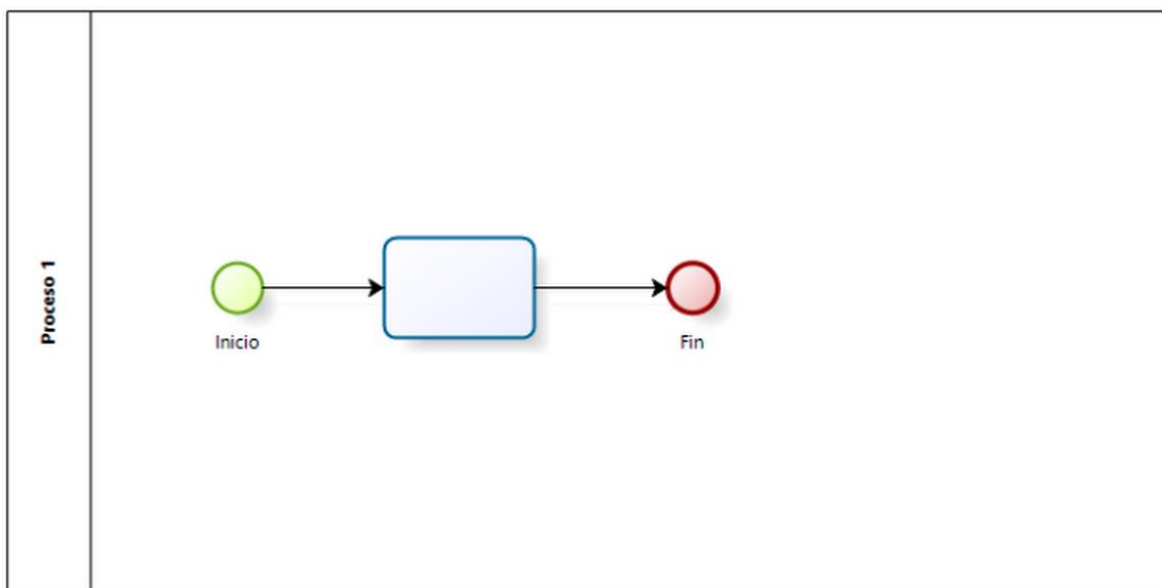


Figura 4.1 Modelo de proceso de negocio de ejemplo

- Flujos de longitud uniforme

El caso de prueba pertenece a la directriz práctica de mismo nombre, esta operación recibe como parámetro el componente gráfico del modelo que analiza (*mxGraphComponent*) y devuelve los flujos de secuencia de longitud uniforme en una lista (*List<mxCell>*). En caso de que el parámetro sea nulo se devolverá una lista vacía, creada por un constructor por defecto. En la figura 4.2 se presentan los métodos de prueba para esta directriz y las medidas asociadas a la misma.

```
/**
 * Test of flujosLongitudUniforme method, of class BusinessProcessDiagram.
 */
@Test
public void testFlujosLongitudUniforme() {
    System.out.println("flujosLongitudUniforme");
    mxGraphComponent graphComponent = null;
    BusinessProcessDiagram instance = new BusinessProcessDiagram();
    List<mxCell> expectedResult = new ArrayList<>();
    List<mxCell> result = instance.flujosLongitudUniforme(graphComponent);
    assertEquals(expectedResult, result);
    // TODO review the generated test code and remove the default call to fail.
    fail("The test case is a prototype.");
}
```

Figura 4.2 Método diseñado para la prueba de la operación “Flujos de longitud uniforme”

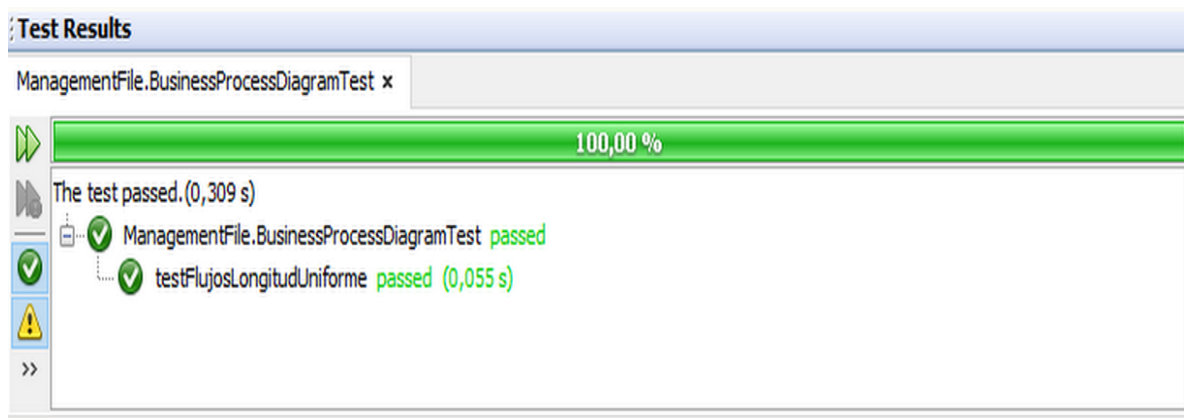


Figura 4.3 Resultado satisfactorio de la prueba de la operación “Flujos de longitud uniforme”

Como se puede observar en la figura 4.3 se obtuvo un resultado satisfactorio de la prueba, esto se debió a que se cumplieron los elementos instanciados y los esperados.

Para el caso de estudio (figura 4.1) se modifica los resultados esperados en la prueba, tal y como se muestra en la figura 4.4. Como resultado se obtiene la figura 4.5

```

/**
 * Test of flujosDeLongitudUniforme method, of class RepresentacionVisual.
 */
@Test
public void testFlujosDeLongitudUniforme() {
    System.out.println("flujosDeLongitudUniforme");
    Pool p = null;
    RepresentacionVisual instance = new RepresentacionVisual();
    List<Flow> expectedResult = null;
    List<Flow> result = instance.flujosDeLongitudUniforme(p);
    assertEquals(2, result.size());
    // TODO review the generated test code and remove the default call to :
    fail("The test case is a prototype.");
}

```

Figura 4.4 Método modificado de prueba de la operación “Flujos de longitud uniforme”

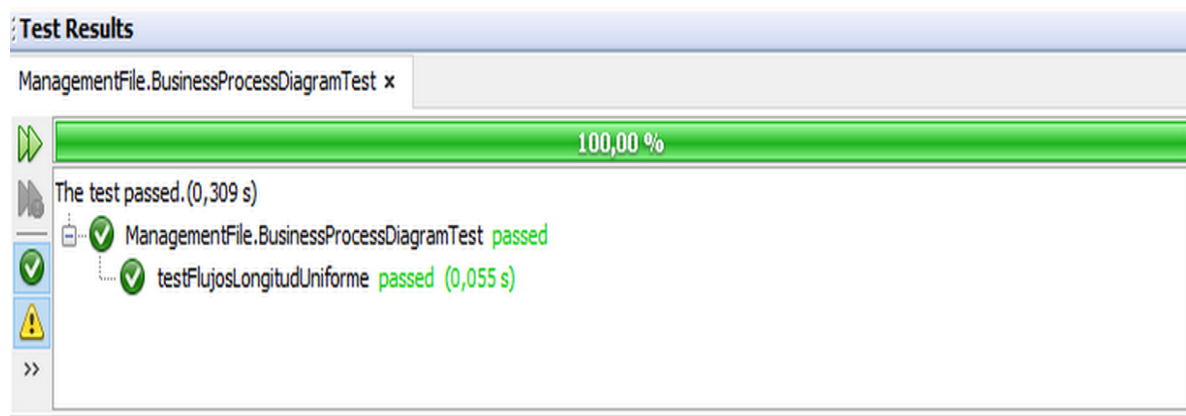


Figura 4.5 Resultado satisfactorio de la prueba de la operación “Flujos de longitud uniforme” modificada

- Razón entre actividades sin etiquetar y el total de actividades

Este método de prueba se utiliza para validar el cálculo de la razón entre actividades sin etiquetar y el total de actividades. La operación originalmente puede recibir un grafo (*mxGraph*) o un componente gráfico (*mxGraphComponent*) y devuelve un entero (*int*) que representa la razón anteriormente mencionada. En el cálculo de la razón se puede dar diversas excepciones como *NullPointerException* (elementos nulos) o NaN, esta última cuando el divisor del cálculo de la razón es 0 y se indefiniría la operación de división. En el caso de errores de primer tipo se devolverá 0, en el segundo caso de tipo de error se devolverá solamente el dividendo de la división.

```

/**
 * Test of getFSEF method, of class BusinessProcessDiagram.
 */
@Test
public void testGetFSEF_mxGraphComponent() {
    System.out.println("getFSEF");
    mxGraphComponent graphComponent = null;
    BusinessProcessDiagram instance = new BusinessProcessDiagram();
    int expResult = 0;
    int result = instance.getFSEF(graphComponent);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    fail("The test case is a prototype.");
}

```

Figura 4.6 Método diseñado para la prueba de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades

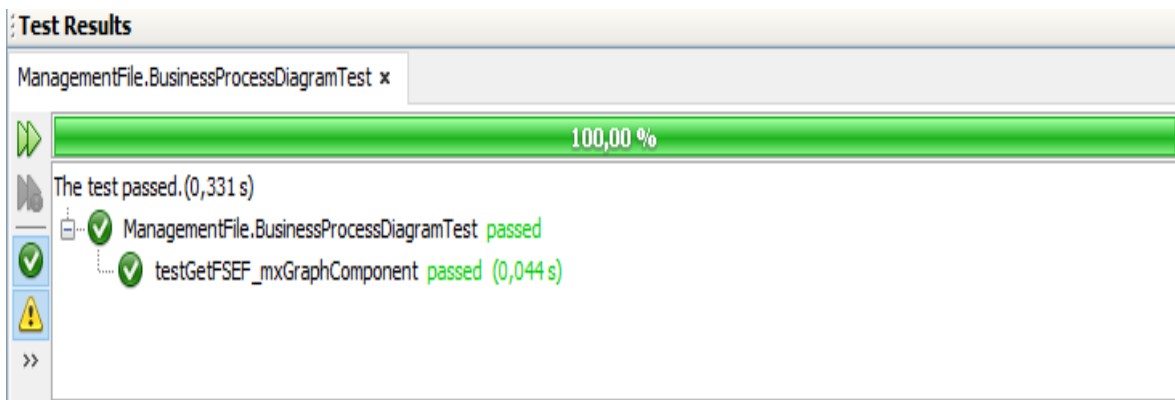


Figura 4.7 Resultado satisfactorio de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades

Para el caso de estudio (figura 4.1) se modifica los resultados esperados en la prueba, tal y como se muestra en la figura 4.8. Como resultado se obtiene la figura 4.9.

```

/**
 * Test of getFSEF method, of class BusinessProcessDiagram.
 */
@Test
public void testGetFSEF_mxGraphComponent() {
    System.out.println("getFSEF");
    mxGraphComponent graphComponent = null;
    BusinessProcessDiagram instance = new BusinessProcessDiagram();
    int expResult = 1;
    int result = instance.getFSEF(graph);
    assertEquals(expResult, result);
    // TODO review the generated test code and remove the default call to fail.
    fail("The test case is a prototype.");
}

```

Figura 4.8 Método modificado diseñado para la prueba de la operación del cálculo de la razón entre actividades sin etiquetar y el total de actividades

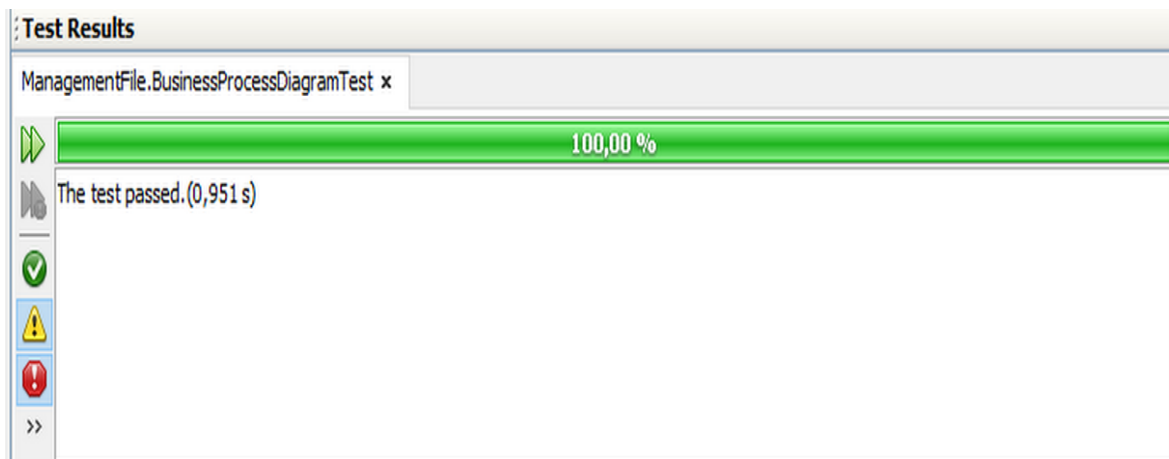


Figura 4.9 Resultado satisfactorio de la operación modificada del cálculo de la razón entre actividades sin etiquetar y el total de actividades

4.3 Pruebas de integración

Según (Pressman 2010), “Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño”.

De acuerdo a (Sommerville 2002) “Las pruebas de integración se desarrollan a partir de la especificación del sistema y dan inicio tan pronto como estén disponibles versiones utilizables de algunos componentes del sistema. La principal dificultad que surge en las pruebas de integraciones localizar los errores que se descubren durante el proceso. Existen interacciones complejas entre los componentes del sistema y cuando se descubre una salida anómala, es difícil encontrarla fuente del error. Para hacer más fácil la localización de errores, siempre se utiliza un enfoque incremental para la integración y prueba del sistema. De forma inicial, se debe integrar una configuración mínima del sistema y probar dicho sistema. Luego se agregan componentes a ésta configuración mínima y se prueba después de que se agrega cada incremento”.

Para ejecutar la prueba de integración se evaluará el modelo de proceso de negocio simple como el que se muestra en la figura 4.1.

La prueba inicia con la carga del modelo, visualizando el camino absoluto del archivo que contiene el modelo, el (los) nombre(s) del (los) proceso(s) que contiene el modelo, la cantidad

de elementos y la clasificación de los procesos con respecto a que si es subproceso o no. Lo anterior se muestra en la figura 4.6.

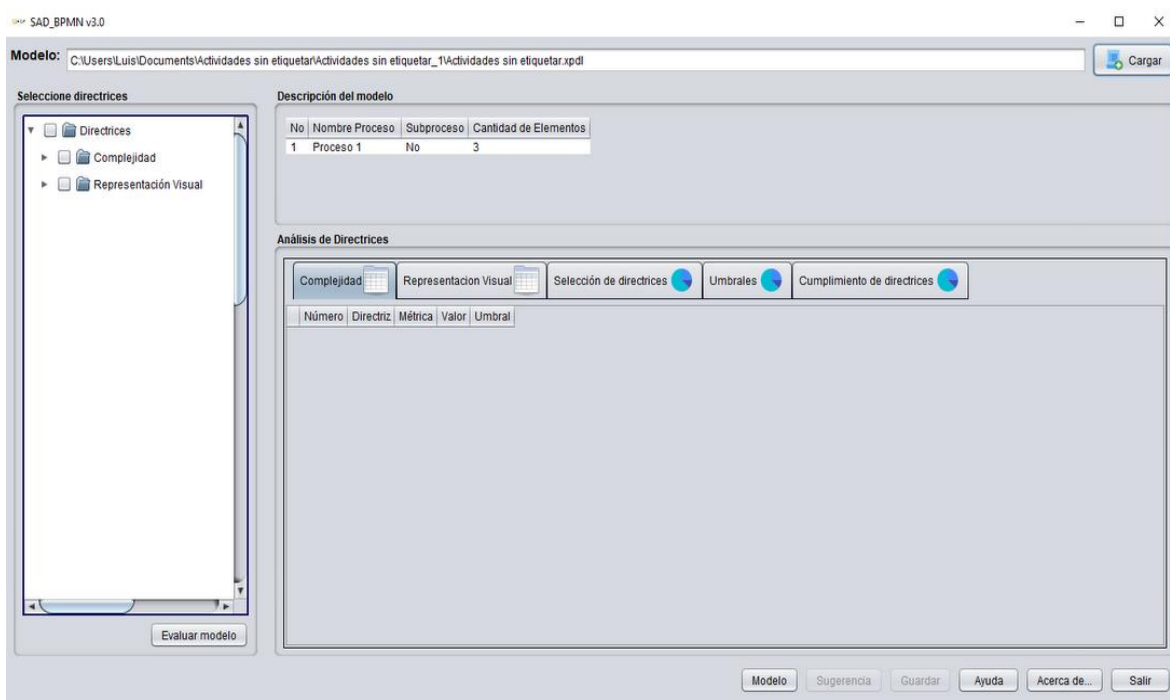


Figura 4.10 Modelo de proceso de negocio de ejemplo cargado

Luego se seleccionan las *directrices que se desean evaluar* en el modelo, se evalúa el mismo y se muestran los datos en la aplicación. Esto se muestran en la siguiente figura:

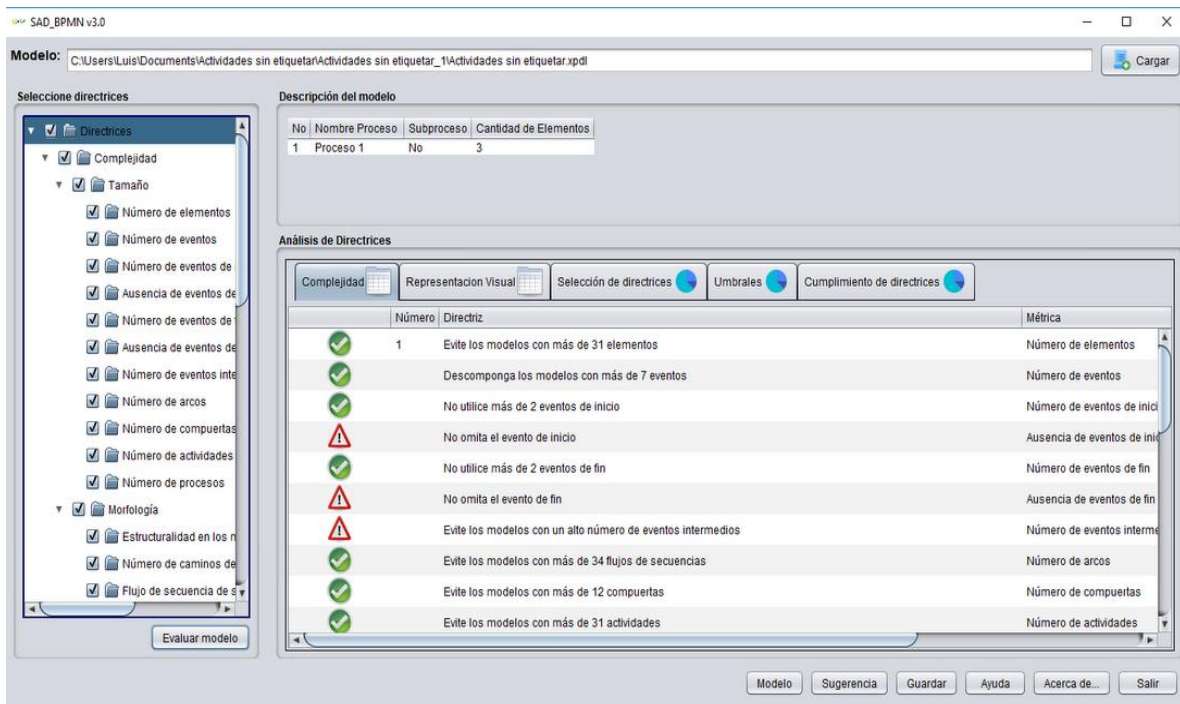


Figura 4.11 Modelo de proceso de negocio de ejemplo evaluado

Es necesario mencionar que una vez cargado el modelo, el mismo se puede visualizar, esta funcionalidad es totalmente nueva en SAD_BPMN 3.0, y, a su vez, constituye uno de los casos de usos significativos. Para el caso de estudio el resultado de desear visualizar el modelo cargado se muestra en la figura que se muestra a continuación:

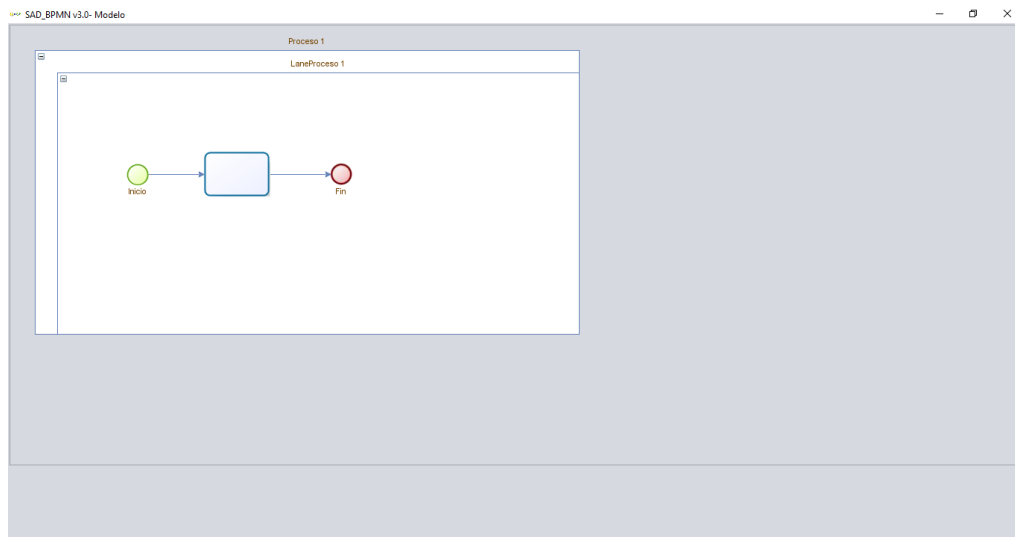


Figura 2.12 Visualización del modelo de proceso de negocio cargado

4.4 Pruebas de caja negra. Diseño de casos de prueba mediante RUP.

“La prueba de caja negra se refiere a las pruebas que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software”, (Pressman 2010).

4.4.1 Secciones a probar en el Caso de Uso Mostrar modelo de proceso de negocio

Tabla 4.8 Caso de prueba para sección Mostrar modelo de proceso de negocio

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Modelar proceso de negocio	EC 1.1: Flujo Normal.	El caso de uso se inicia cuando el modelador (usuario) desea visualizar el modelo de proceso de negocio cargado. Como resultado se muestra una nueva venta con dicho modelo.	1. El modelador carga el modelo de proceso de negocio. 2. El sistema habilita la opción Modelo en la parte inferior derecha de la interfaz. 3. El modelador selecciona la opción habilitada anteriormente. 4. El sistema muestra una nueva interfaz que contiene el modelo de proceso de negocio representado gráficamente.

Escenarios a probar en el caso de uso Mostrar modelo de proceso de negocio

Tabla 4.9 Escenario a probar para sección *Mostrar modelo de proceso de negocio*

Id del escenario	Escenario	Variable 3 <i>Extensión del archivo</i>	Variable 4 <i>Estructura del modelo</i>	Resultado del Sistema	Resultado de la Prueba
EC1	Cargar modelo	Válido	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Válido	Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando.	
		Inválido	Válido		
		Inválido	Inválido		

4.4.2 Secciones a probar en el Caso de Uso Mostrar sugerencias visuales

Tabla 4.10 Caso de prueba para sección *Mostrar sugerencias visuales*

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Mostrar sugerencias visuales	EC 1.1: Flujo Normal.	El caso de uso se inicia cuando el modelador (usuario) desea mostrar las sugerencias visuales con respecto a ciertas directrices. Como resultado se muestra una nueva vista el modelo cargado y la señalización de las sugerencias existentes.	<ol style="list-style-type: none"> 1. El modelador carga el modelo de proceso de negocio. 2. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 3. El modelador selecciona las directrices que desea evaluar. 4. Se selecciona la opción Evaluar por parte del modelador. 5. El sistema calcula las métricas, muestra los resultados en los gráficos y tablas y se habilita las opciones Sugerencia y Guardar. 6. En algunas de las tablas el modelador selecciona las directrices con respecto a las cuales desea visualizar las sugerencias. 7. El sistema muestra el modelo y las sugerencias con respecto a las directrices seleccionadas.

	EC 1.2: Flujos alternativos Directrices no seleccionadas	El modelador desea visualizar las sugerencias para un modelo de proceso de negocio, pero al hacerlo no selecciona ninguna de las directrices.	<ol style="list-style-type: none"> 1. El modelador carga el modelo de proceso de negocio. 2. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 3. El modelador selecciona las directrices que desea evaluar. 4. Se selecciona la opción Evaluar por parte del modelador. 5. El sistema calcula las métricas y muestras los resultados en los gráficos y tablas. 6. El modelador no selecciona las directrices con respecto a las cuales desea visualizar las sugerencias en alguna de las tablas.
--	---	---	--

Escenarios a probar en el caso de uso Mostrar sugerencias visuales

Tabla 4.11 Escenario a para sección Cargar modelo de proceso de negocio

Id del escenario	Escenario	Variable 3 <i>Extensión del archivo</i>	Variable 4 <i>Estructura del modelo</i>	Resultado del Sistema	Resultado de la Prueba
EC1	Cargar modelo	Válido	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Válido	Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando.	
		Inválido	Válido		
		Inválido	Inválido		

Tabla 4.12 Escenario a para sección *Cargar modelo de proceso de negocio*

Id del escenario	Escenario	Variable 3 <i>Selección de directrices</i>	Resultado del Sistema	Resultado de la Prueba
EC2	Calcular métricas de calidad	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando y muestra una animación en el área de selección de directrices.	

Como se puede apreciar, las secciones y los escenarios asociados a cada una de éstas mostradas anteriormente no abarcan todos los casos de usos de SAD_BPMN 3.0, véase los anexos 13, 14 y 15 que contienen las pruebas relacionadas a este epígrafe con respecto a los restantes casos de usos.

4.5 Conclusiones parciales

SAD_BPMN es una herramienta de evaluación y validación de modelos de procesos de negocio, que, desde sus inicios, se le realizó una estimación de costes y una planificación desde diversos puntos de vistas. Además, para garantizar la calidad y el punto de partida para la entrega de ésta aplicación, se le realizaron varias pruebas (de caja blanca y de caja negra), las cuales contribuyen a la calidad y obtención de rendimiento en SAD_BPMN. La aplicación cumple con los requisitos funcionales validando los errores que se puedan encontrar.

Conclusiones

1. Se realizó la evaluación del soporte que ofrece SAD_BPMN al conjunto de directrices de calidad, determinando los déficits en cada uno de los casos.
2. Se logró desarrollar la visualización de modelo a evaluar utilizando teoría de grafos.
3. Fue implementada la evaluación de nuevas directrices de calidad para los modelos creados, lo que permitirá la evaluación de las mismas. Esto se logró con la elaboración e incorporación de cinco nuevas directrices, una perteneciente al grupo taxonómico *Complejidad general* (Número de procesos) y cuatro de *Representación visual* (Maximizar el número de objetos conectores dibujados ortogonalmente, Considerar el uso de particiones, ejemplo: pools y lanes, Pool sin etiquetar y Longitud variada de flujos de secuencia).
4. Se logró una satisfactoria validación del software mediante pruebas de caja negra y caja blanca. Entre las pruebas realizadas se encuentran pruebas de unidad, de integración y casos de prueba.
5. Se integraron en una nueva versión todas las mejoras y nuevas funcionalidades como la inclusión de la ayuda de la herramienta y resúmenes de la evaluación mediante gráficos, lográndose el completamiento de la herramienta SAD_BPMN.

Recomendaciones

Derivadas del estudio realizado, así como de las conclusiones generales emanadas del mismo, se recomienda:

- Continuar con la extensión de SAD_BPMN para garantizar su soporte y mantenimiento.
- Implementar una funcionalidad que permita corregir los errores que se encuentren en los modelos de manera automática o semiautomática.

Referencias bibliográficas

- Aguilar-Savén, R.S., 2004. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2), pp.129–149. Available at: <http://www.sciencedirect.com/science/article/pii/S0925527303001026>.
- Alicante, U. de, 2014. Casos de prueba JUnit. *Dept. Ciencia de la Computación e IA*. Available at: <http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion04-apuntes.html> [Accessed October 6, 2017].
- Allen Weiss, M., 2004. *Estructuras de datos en Java III*. A. Otero, ed., La Habana, Cuba: Editorial Félix Varela.
- Álvarez, M.Á., 2014. Qué es MVC. *DesarrolloWeb.com*. Available at: <http://www.desarrolloweb.com/articulos/que-es-mvc.html> [Accessed May 14, 2018].
- Badica, C. & Fox, C., 2002. Modelling and verification of business processes. *Proceedings of the IASTED International Conference Applied Simulation and Modelling*.
- Becker, J., Rosemann, M. & von Uthmann, C., 2000. Guidelines of business process modeling. In W. van der Aalst, J. Desel, & A. Oberweis, eds. *Business Process Management*. Springer Berlin / Heidelberg, pp. 241–262. Available at: http://dx.doi.org/10.1007/3-540-45594-9_3.
- Born, M., Kirchner, J. & Mueller, J.P., 2009. Context-driven business process modelling. In O. Camp & H. S., eds. *Advanced Technologies and Techniques for Enterprise Information Systems (11th International Conference on Enterprise Information Systems (ICEIS 2009))*. Milan, ITALY: INSTICC Press, pp. 17–26.
- Calzada Ricet, A. & Pairol Fuentes, R., 2016. *Herramienta para la evaluación de modelos de procesos de negocio SAD_BPMN v2.0*. Universidad Central “Marta Abreu” de Las Villas.
- Casquero Torres, W.A. et al., 2014. Estándares para el Modelado de Procesos de Negocios. *SlideShare*. Available at: <https://www.slideshare.net/CsarHernnPatricioPeralta/estndares-para-el-modelado-de-procesos-de-negocios> [Accessed January 18, 2018].
- Castela, N. et al., 2002. Survey, analysis and validation of information for business process modeling. *ICEIS 2002. Proceedings of the Fourth International Conference on Enterprise Information Systems*, 2, pp.803–806. Available at: <Go.
- Craig Larman, 1999. *UML y Patrones I*. P. E. R. Vekquez, ed., México: PRENTICEHALL.
- Dumas, M. et al., 2013. *Fundamentals of business process management*, Springer-Verlag Berlin Heidelberg.

- Falla Aroche, S., 2009. Librerías para generar archivos .PDF.
<https://www.facebook.com/maestrosdelweb>. Available at:
<http://www.maestrosdelweb.com/librerias-para-generar-archivos-pdf/> [Accessed February 16, 2016].
- Feregrino, A., 2012. JDOM, Fácil manipulacion de XML en Java.
<http://blog.fferegrino.org>. Available at: <http://fferegrino.blogspot.com/2012/03/estas-una-biblioteca-para-java-que.html> [Accessed October 20, 2017].
- Gracia, L.M., 2014. OpenCSV: trabajando con CSV de forma sencilla con Java. *Un poco de Java y +*. Available at: <https://unpocodejava.com/2014/12/31/opencsv-trabajando-con-csv-de-forma-sencilla-con-java/> [Accessed February 5, 2018].
- Gruhn, V. & Laue, R., 2008. Good and Bad Excuses for Unstructured Business Process Models. In *EuroPlop*. UVK - Universitaetsverlag Konstanz, pp. 279–292.
- Guerrero, T., 2015. Dato Java Ejemplo de un reporte pdf con iText desde Java.
<http://datojava.blogspot.com/>. Available at:
<http://datojava.blogspot.com/2015/08/itextPDFJavaEjemplo.html> [Accessed February 16, 2016].
- He, G. et al., 2010. Business Process Modeling: A Survey. In Y. X. Yang, ed. *Proceedings of Annual Conference of China Institute of Communications*. pp. 172–178.
- Heidari, F., Loucopoulos, P. & Kedad, Z., 2011. A Quality-Oriented Business Process Meta-Model. In J. Barjis, T. Eldabi, & A. Gupta, eds. *Enterprise and Organizational Modeling and Simulation*. pp. 85–99.
- Hernández Cárdenas, L., 2016. *Medidas para la representación visual de modelos de procesos de negocio*. Universidad Central “Marta Abreu” de Las Villas.
- Jacobson, I., Booch, G. & Rumbaugh, J., 2000. *El proceso Unificado de Software* Edición en. Pearson Educación S.A, ed., Madrid, España: Addison Wesley.
- Jiménez Martínez, L.A. & Reyes García, F., 2018. Directrices prácticas de modelación de procesos de negocio en herramientas informáticas. , p.9.
- Khelif, W. et al., 2009. Quality metrics for business process modeling. In R. Revetria, B. Mladenov, & N. Mastorakis, eds. *Acs'09: Proceedings of the 9th Wseas International Conference on Applied Computer Science*. pp. 195–200.
- Khelif, W., Zaaboub, N. & Ben-Abdallah, H., 2010. Coupling metrics for business process modeling. *WSEAS Transactions on Computers*, 9(1), p.11.

- De la Vara, J.L. et al., 2010. COMPRO: A Methodological Approach for Business Process Contextualisation. In R. Meersman, T. Dillon, & P. Herrero, eds. *On the Move to Meaningful Internet Systems: Otm 2010, Pt I*. pp. 132–149.
- Lee, R.G., 1998. Business process management: a review and evaluation. *Business Process Management Journal*, 4(3), p.214. Available at: <http://sfx.libis.be/sfxlcl3?sid=google&auinit=RG&aulast=Lee&atitle=Business>.
- Liu, R. & Hou, H., 2010. Business process metrics modeling based on model-driven. *2010 IEEE International Conference on Software Engineering and Service Sciences (ICSESS 2010)*.
- Mendling, J., Recker, J. & Reijers, H.A., 2009. *PROCESS MODELING QUALITY: A FRAMEWORK AND RESEARCH AGENDA*,
- Mendling, J., Reijers, H.A. & van der Aalst, W.M.P., 2010. Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52(2), pp.127–136.
- Mesa, A., Lochmuller, C. & Tabares, M.S., 2014. Comparativo entre herramientas BPMN. *Revista Soluciones de Postgrado EIA*, pp.95–108.
- Moreno Montes de Oca, I., 2015. *Patrón y clasificación taxonómica para directrices prácticas en modelos de procesos de negocio*. Universidad Central “Marta Abreu” de Las Villas.
- Moreno Montes de Oca, I. & Snoeck, M., 2017. A taxonomy of quality guidelines for business process modeling. *ACM Computing Surveys*.
- Zur Muehlen, M. & Indulska, M., 2010. Modeling languages for business processes and business rules: A representational analysis. *Information Systems*, 35(4), pp.379–390. Available at: <http://www.sciencedirect.com/science/article/pii/S0306437909000118>.
- Ning, K., Chen, Y. & O’Sullivan, D., 2004. *Rationality validation of business process model by simulation method* J. Chen, ed., Beijing, CHINA.
- Parra Conejero, I., 2016. Introducción a las pruebas unitarias con PHPUnit - Uno de piera. *Uno de piera*. Available at: <https://www.uno-de-piera.com/introduccion-a-las-pruebas-unitarias-con-phpunit/> [Accessed November 15, 2016].
- Pressman, R.S., 2010. *Ingeniería de software. Un enfoque práctico* Séptima ed. P. R. Vázquez, ed., México, D. F.: University of Connecticut.
- Recker, J. et al., 2009. Business Process Modeling- A Comparative Analysis. *Journal of the Association for Information Systems*, 10(4), pp.333–363.

- Rene, J., 2015. COCOMO II. Available at: <http://adminprojsoft.blogspot.com/2015/11/cocomo-ii.html> [Accessed May 14, 2018].
- Reyes García, F., 2016. *Herramienta para la aplicación de directrices de complejidad a modelos de procesos de negocios*. Universidad Central “Marta Abreu” De Las Villas.
- Robles, V., 2013. MVC (Modelo Vista Controlador) en PHP nativo. <https://www.facebook.com/Víctor-Robles-WEB-146046699278162/>. Available at: <https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/> [Accessed May 14, 2018].
- Rodríguez Andrés, R., 2008. Lenguajes, notaciones y herramientas para el modelado y análisis... <https://www.facebook.com/gestiopolis>. Available at: <https://www.gestiopolis.com/lenguajes-notaciones-herramientas-modelado-analisis-procesos/> [Accessed January 18, 2018].
- Rosemann, M., Recker, J.C. & Flender, C., 2008. Contextualisation of business processes. *International Journal of Business Process Integration and Management*, 3(1), pp.47–60.
- Schrepfer, M., 2010. *Modeling guidelines for business process models*. HUMBOLDT-UNIVERSITÄT ZU BERLIN.
- Smirnov, S. et al., 2012. Business process model abstraction: a definition, catalog, and survey. *Distrib. Parallel Databases*, 30(1), pp.63–99.
- Sommerville, I., 2002. Pruebas del software. In J. L. V. Chavarria, ed. *Ingeniería de Software*. México: Pearson Educación, p. 712.
- Velasco, R., 2016. Llega el nuevo NetBeans 8.2 con soporte completo para PHP7. *RedesZone*. Available at: <https://www.redeszone.net/2016/10/03/llega-nuevo-netbeans-8-2-soporte-completo-php7/> [Accessed January 31, 2018].
- Villamarín Rodríguez, R., 2010. Uso de JFreeChart. *MyJavaZone*. Available at: <http://www.myjavazone.com/2010/11/uso-de-jfreechart.html>.
- Weske, M., 2007. *Business Process Management: Concepts, Languages, Architectures*, Springer-Verlag Berlin Heidelberg.
- West, D.B., 2001. *Introduction to graph theory*, Prentice hall Upper Saddle River.
- White, S.A. & Miers, D., 2009. *Modelado BPMN. Comprendiendo y utilizando BPMN*. Edición Di., Lighthouse Point, Florida, USA: Future Strategies Inc. Available at: www.futstrat.com.

Wikipedia, Business Process Model and Notation - Wikipedia, la enciclopedia libre.
Wikipedia. Available at:
https://es.wikipedia.org/wiki/Business_Process_Model_and_Notation [Accessed April 11, 2017].

Anexos

Anexo 1. BPMN

BPMN Notación para el Modelado de Procesos de Negocio BPMN (*Business Process Modeling Notation*) es la nomenclatura estándar para el modelado de procesos de negocios. Fue diseñado como una notación de tipo diagrama de flujo robusto, fácil de usar y completamente independiente de la implementación. Su nomenclatura remite a conceptos propios de la programación: intercambio de mensajes, condicionales, ciclos, manejo de excepciones, flujos en paralelo, estados y eventos. El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (*stakeholders*). Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores (responsables de implementar los procesos) y los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos). En síntesis, BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Se define como una notación gráfica que describe la lógica de los pasos en un proceso de negocio. Es un lenguaje formal que permite modelar, simular y, eventualmente, ejecutar un proceso de negocio. Proporciona un método normalizado para representar procesos de negocio. Es legible, entendible y de poca complejidad. Propone un lenguaje común entre los usuarios de negocio y los técnicos. Facilita la diagramación de los procesos de negocio. Determina y define los requerimientos del sistema (Casquero Torres et al. 2014).

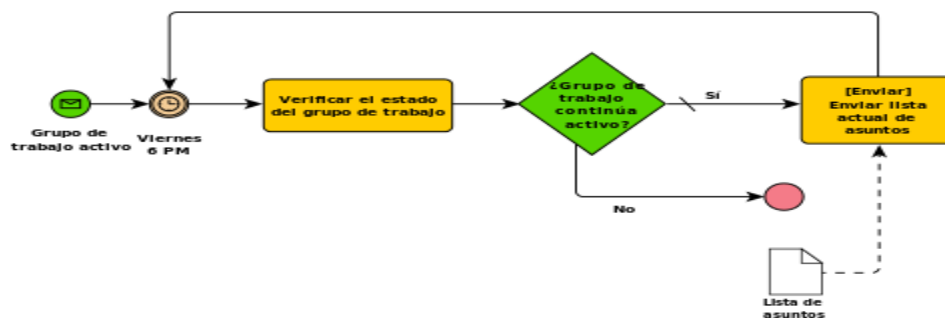


Figura A.1: Ejemplo de proceso con flujo normal utilizando BPMN (Fuente: (Wikipedia n.d.))



Figura A.2: BPMN logo.

BPMN ha venido evolucionando a lo largo del tiempo desde su surgimiento en el año 2001. Su desarrollado estuvo a cargo de la organización BPM Initiative, pasando posteriormente a manos de OMG después de la fusión de las dos organizaciones en el año 2005. Su versión actual en el año 2011 es la 2.0. Elementos El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores sea fácil entender el flujo y el proceso. Objetos de flujo: Eventos, Actividades, Decisión (Gateways). Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación, Swimlanes (Carriles), Pool, Lane. Artefactos: Objetos de Datos, Grupo, Anotación; (Casquero Torres et al. 2014).

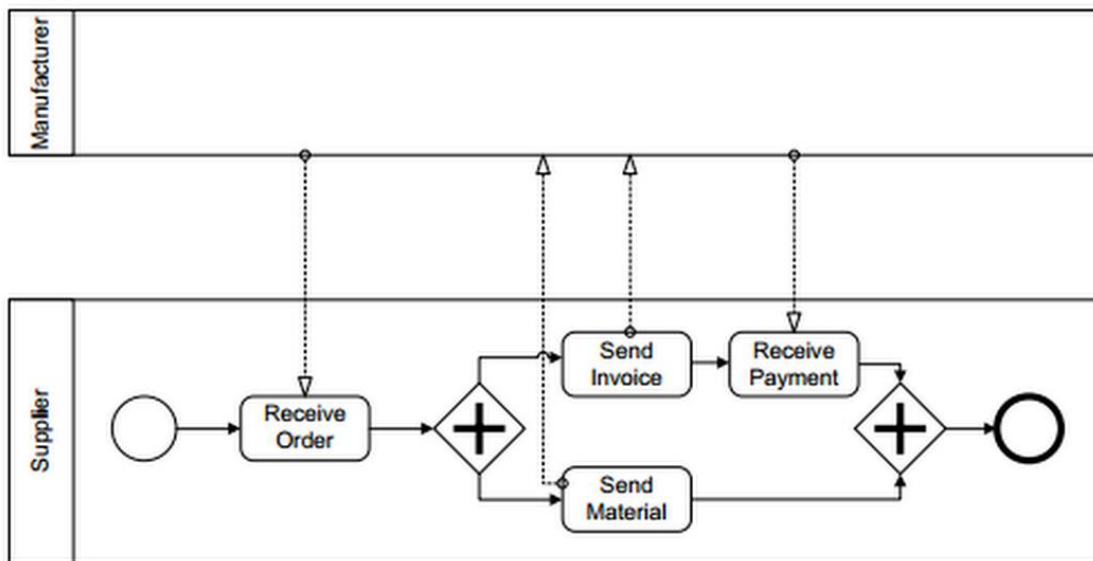


Figura A.3: Ejemplo de procesos de negocio interactuando (Fuente: (Weske 2007))

BPMN define un Diagrama de Procesos de Negocio basado en la técnica de Diagramas de Flujo y presenta tres tipos de modelado de procesos, los cuales son: proceso de negocio interno, proceso de negocio abstracto (externo) y proceso de negocio colaborativo. Según

(White & Miers 2009), originalmente BPMN no especificaba un formato de almacenamiento, permitiendo que una franja más amplia de proveedores adoptaran el estándar, a pesar de limitar la portabilidad de los modelos.

BPMN 2.0 - Business Process Model and Notation

<http://bpmb.de/poster>

Traducido por Ildelfonso Montero, Luciano García-Bañuelos, Marlon Dumas

Actividades



Tarea
Una Tarea es una unidad de trabajo, el trabajo a realizar. Cuando aparece con el símbolo indica un Subproceso, una actividad que puede ser refinada.

Transacción
Una Transacción es un conjunto de actividades relacionadas lógicamente, advirtiéndose a un protocolo transaccional particular.

Subproceso de Evento
Un Subproceso de Evento se situa en el interior de otro subproceso. Este se activa en la ocurrencia del evento de inicio especificado y mientras el proceso que lo contiene permanece también activo. El subproceso de evento puede interrumpir o no al proceso que lo contiene.

Actividad de Llamada
Una Actividad de Llamada es una referencia a un Subproceso o Tarea definido de forma global, que se reutiliza en el proceso actual.

Marcador de Actividad

Los marcadores especifican el comportamiento particular de las actividades durante su ejecución:



Tipos de Tarea

Los tipos especifican la naturaleza de la tarea que se desea llevar a cabo:



Compuertas



Exclusiva
En un punto de bifurcación, selecciona exactamente un flujo de secuencia de entre las alternativas existentes. En un punto de convergencia, la compuerta espera a que un flujo incidente complete para activar el flujo saliente.

Basada en Eventos
Esta compuerta siempre será seguida por eventos o tareas de recepción, y sólo activará un flujo saliente dependiendo del evento que ocurra en primer lugar.

Paralela
En un punto de bifurcación, todos los caminos salientes serán activados simultáneamente. En un punto de convergencia, la compuerta espera a que todos los flujos incidentes completen antes de activar el flujo saliente.

Inclusiva
En un punto de bifurcación, al menos un flujo es activado. En un punto de convergencia, espera a todos los flujos que fueron activados para activar el siguiente.

Exclusiva Basada en Eventos
En la ocurrencia de uno de los eventos subsecuentes se crea una nueva instancia del proceso.

Compleja
Comportamiento complejo de convergencia/bifurcación no capturado por el resto de compuertas.

Conversaciones

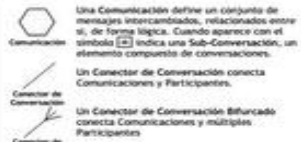
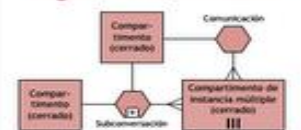


Diagrama de Conversación



Coreografías



Diagrama de Coreografía

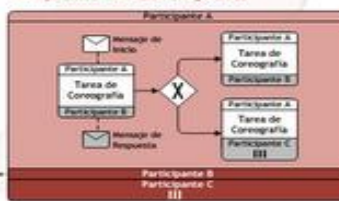
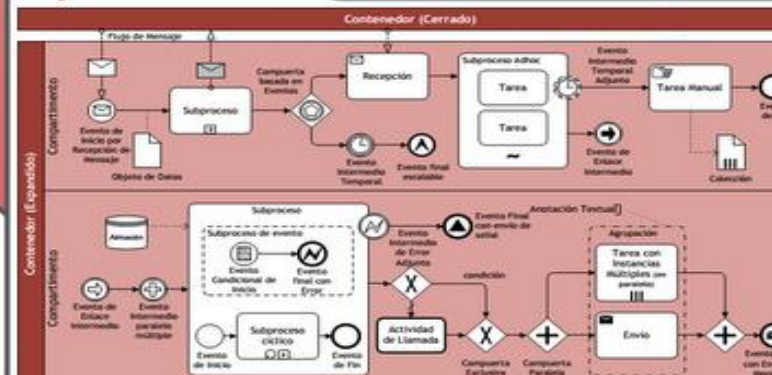
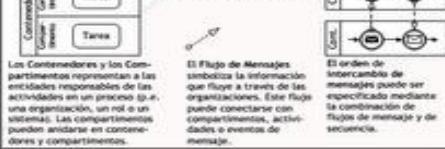


Diagrama de Colaboración



Contenedores



Eventos

Inicio Intermedios Fin

	Inicio	Intermedios	Fin
Evento de Inicio			
Evento Intermedio			
Evento Final			
Evento de Inicio de Subproceso			
Evento de Fin de Subproceso			
Captura			
Adaptación			
Adaptación de No Interrupción			
Lanzamiento			

Simple: Eventos sin especificar. Indican puntos de inicio, de fin y situaciones intermedias.

Message: Recepción y envío de mensajes.

Temporal: Puntos en el tiempo, lapsos, límites (timeouts). Pueden ser eventos únicos o cíclicos.

Escalable: Cambio a un nivel más alto de responsabilidad.

Condición: Reacción a cambios en las condiciones de negocio o integración de reglas de negocio.

Enlace: Conexiones fuera de página. Dos conectores de enlace equivalen a un flujo de secuencia.

Error: Captura y lanzamiento de errores conocidos con nombre.

Cancellation: Reacción a la cancelación de una transacción/ solicitud de cancelación.

Compensación: Mensaje/ solicitud de compensación.

Signal: Interrupción de señales entre procesos. Una señal puede ser capturada varias veces.

Multiples: Captura uno de un conjunto de eventos. Lanza todos los eventos definidos.

Paralelo Multiples: Captura todos los eventos de un conjunto de eventos en paralelo.

Terminación: Terminación inmediata del proceso.

Datos



Un **Dato de Entrada** o **Input** es una entrada externa a todo el proceso. Puede ser leído por una actividad.

Un **Dato de Salida** o **Output** es una variable disponible como resultado del proceso.

Un **Dato de Tipo Objeto** representa información que fluye a través del proceso tales como documentos, correos electrónicos o cartas.

Una **Colección de Objetos de Datos** representa una colección de información, p.e. una lista de artículos.

Un **Almacén** es un lugar donde el proceso puede leer o escribir datos, p.e. una base de datos. La información en un almacén persiste más allá de la vida de la instancia del proceso.

Un **Mensaje** es utilizado para representar el contenido de una comunicación entre dos participantes.

Figura A.4: BPMN 2.0

Anexo 2. Grafos

Gracias a la teoría de grafos se pueden resolver diversos problemas como por ejemplo la síntesis de circuitos secuenciales, contadores o sistemas de apertura. Se utiliza para diferentes áreas: dibujo computacional, y en todas las áreas de Ingeniería. Los grafos se utilizan también para modelar trayectos como el de una línea de autobús a través de las calles de una ciudad, en el que es posible obtener caminos óptimos para el trayecto aplicando diversos algoritmos. La teoría de grafos también ha servido de inspiración para las ciencias sociales, en especial para desarrollar un concepto no metafórico de red social que sustituye los nodos por los actores sociales y verifica la posición, centralidad e importancia de cada actor dentro de la red. Esta medida permite cuantificar y abstraer relaciones complejas, de manera que la estructura social puede representarse gráficamente. Por ejemplo, una red social puede representar la estructura de poder dentro de una sociedad al identificar los vínculos (aristas), su dirección e intensidad y da idea de la manera en que el poder se transmite y a quiénes, (Reyes García 2016).

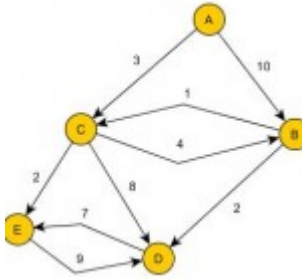


Figura A.5: Grafo

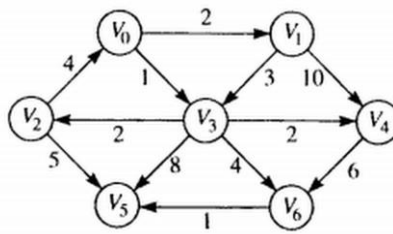


Figura A.6: Grafo dirigido (Fuente: (Allen Weiss 2004))

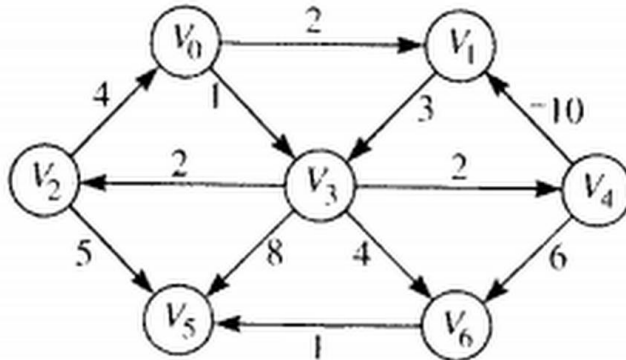


Figura A.7: Grafo dirigido con ciclo de coste negativo (Fuente: (Allen Weiss 2004))

En (Reyes García 2016) se presenta un conjunto de definiciones y notaciones que se asumen, algunas de éstas son:

- Un grafo puede ser dirigido o no dirigido. A un grafo dirigido se le llama digrafo.
- Los vértices se denominan también nodos o puntos y son objetos que pueden tener nombre y otras propiedades.
- Las aristas se denominan también arcos y son conexiones entre dos vértices y representan relaciones entre los objetos.
- Un seudografo $G = (V, E, f)$ es una terna, donde $V \neq \emptyset$ es el conjunto de nodos, E es el conjunto de aristas y f una función definida como $f: E \rightarrow V \times V \cup \{ \{u, v\}; u \in V, v \in V \}$.

Anexo 3. Análisis de las herramientas

Tabla TA.1 Análisis de las herramientas

Herramientas	Directrices cumplidas	Directrices incumplidas	Directrices de tamaño cumplidas	Directrices de morfología cumplidas	Directrices de diseño visual cumplidas	Directrices de estilo de etiquetas cumplidas
Dia (v0.97.2)	0	50	0/10	0/16	0/15	0/9
Bonita BPM Community Edition v7.1.3	6 (<i>P47, P45, P46, P49, P21, P31</i>)	44	0/10	1/16 (<i>P21</i>)	1/15 (<i>P31</i>)	4/9 (<i>P45, P46, P47, P49</i>)
AdonisCE 3.05.02 (CE-> Comunity Edition)	5 (<i>P19, P21, P17, P3, P8</i>)	45	2/10 (<i>P3, P8</i>)	3/16 (<i>P17, P19, P21</i>)	0/15	0/9
Bizagi Process Modeler 64 bit v2.7.0.2	2 (<i>P3, P17</i>)	48	1/10 (<i>P3</i>)	1/16 (<i>P17</i>)	0/15	0/9
Camunda Modeler v1.10.0	3 (<i>P30, P31, P32</i>)	47	0/10	0/16	3/15 (<i>P30, P31, P32</i>)	0/9

Modelio v3.6.1	3 (<i>P4, P5, P6</i>)	47	3/10 (<i>P4, P5, P6</i>)	0/16	0/15	0/9
MyBPMNEngine v1.0.0	5 (<i>P30, P31, P32, P47 P49</i>)	45	0/10	0/16	3/15 (<i>P30, P31, P32</i>)	2/9 (<i>P47, P49</i>)
ProcessMaker 3.2.1.0	2 (<i>P13, P21</i>)	48	0/10	2/16 (<i>P13, P21</i>)	0/15	0/9
RunaWFE 4.1.1	6 (<i>P47, P16, P31, P19, P49, P3</i>)	44	1/10 (<i>P3</i>)	2/16 (<i>P16, P19</i>)	1/15 (<i>P31</i>)	2/9 (<i>P47, P49</i>)
TIBCO Bussiness Studio v3.5.3.025.1	9 (<i>P45, P46, P47, P3, P15, P16, P17, P21, P31</i>)	41	1/10 (<i>P3</i>)	4/16 (<i>P15, P16, P17, P21</i>)	1/15 (<i>P31</i>)	3/9 (<i>P45, P46, P47</i>)
Yaoqiang BPMN Editor v5.3.12	2 (<i>P4, P6</i>)	48	2/10 (<i>P4, P6</i>)	0/16	0/15	0/9
yEd v3.17.1	0	50	0/10	0/16	0/15	0/9
Software Ideas Modeler v10.68 Community Edition (sin	0	50	0/10	0/16	0/15	0/9

licencia, freeware).						
Edraw Max v7.9	0	50	0/10	0/16	0/15	0/9
bpmn.io	3 (<i>P30, P31, P32</i>)	47	0/10	0/16	3/15 (<i>P30, P31, P32</i>)	0/9
Joget Workflow v5.0.20	5 (<i>P47, P49, P19, P40, P8</i>)	45	1/10 (<i>P8</i>)	1/16 (<i>P19</i>)	1/15 (<i>P40</i>)	2/9 (<i>P47, P49</i>)
Aris Express*	11	39	0/10	6/16	-	-
Visual Paradigm*	8	42	0/10	0/16	-	-
Signavio*	23	27	3/10	9/16	-	-

* Los datos correspondientes son extraídos de (Reyes García 2016) y (Moreno Montes de Oca 2015).

Anexo 4. Análisis comparativo de las “directrices obsoletas” con respecto a las “directrices nuevas”

Tabla TA.2 Análisis comparativo de las directrices obsoletas con respecto a las directrices nuevas

Directrices de Calidad			
Representación visual		Complejidad general	
Diseño	Estilo de etiquetado	Morfología	Tamaño
<p>P27: Alto número de cruces de flujos de secuencia (se mantiene)</p> <p>P28: Curvas en los flujos de secuencia (se mantiene)</p> <p>P29: No perpendicularidad</p> <p>P30: Solapamiento de flujos de secuencia (se fusiona junto a P31 y P32)</p> <p>P31: Solapamiento de elementos (se fusiona junto a P30 y P32)</p> <p>P32: Solapamiento de flujos de secuencia y elementos (se fusiona junto a P31 y P30)</p> <p>P33: Flujo de secuencia inconsistente (se mantiene)</p>	<p>P42: Las etiquetas no siguen el estilo verbo-sustantivo (se mantiene)</p> <p>P43: Tipos de tareas no especificados</p> <p>P44: Etiquetas largas (se mantiene en una nueva directriz y tiene otra nueva directriz semejante mínimamente)</p> <p>P45: Eventos sin etiquetar</p> <p>P46: Compuertas sin etiquetar</p> <p>P47: Actividades sin etiquetar</p> <p>P48: Flujo de secuencia de salida sin etiquetar</p> <p>P49: Pool sin etiquetar</p> <p>P50: Etiquetas enviar/recibir no están reservadas para estos elementos</p>	<p>P11: Falta de estructuración en el modelo (se mantiene)</p> <p>P12: Alto número de flujos de secuencia por compuerta (posee una nueva directriz semejante en parte)</p> <p>P13: Alto número de flujos de secuencias por eventos (se mantiene)</p> <p>P14: Múltiples entradas y salidas en la misma compuerta (se mantiene)</p> <p>P15: Múltiples ciclos (se mantiene)</p> <p>P16: Ciclos no estructurados (se mantiene)</p> <p>P17: Ciclos mal formados</p> <p>P18: Alto nivel de paralelismo (se mantiene)</p> <p>P19: Paralelismo incorrecto</p>	<p>P1: Alto número de elementos (se mantiene)</p> <p>P2: Alto número de eventos (se mantiene)</p> <p>P3: Alto número de eventos de inicio (se fusiona con P5 para formar una nueva métrica)</p> <p>P4: Ausencia de eventos de inicio (se fusiona con P6 para formar una nueva métrica)</p> <p>P5: Alto número de eventos de fin (se fusiona con P3 para formar una nueva métrica)</p> <p>P6: Ausencia de eventos de fin (se fusiona con P4 para formar una nueva métrica)</p> <p>P7: Alto número de eventos intermedios (se mantiene)</p> <p>P8: Alto número de flujos de secuencia (se mantiene)</p> <p>P9: Alto número de compuertas (se mantiene)</p> <p>P10: Alto número de actividades (cambia)</p>

<p>P34: Longitud variada de flujos de secuencia</p> <p>P35: Tamaño no uniforme de los elementos (se mantiene)</p> <p>P36: Modelo cuadrado (se mantiene)</p> <p>P37: No simetría (se mantiene)</p> <p>P38: Área de dibujo extensa (se mantiene)</p> <p>P39: Lejanía de elementos relacionados (se mantiene)</p> <p>P40: Apariencia no uniforme (tiene una nueva directriz semejante)</p> <p>P41: Diagrama desordenado (se mantiene)</p>		<p>P20: Alto nivel de profundidad (se mantiene)</p> <p>P21: Elementos duplicados (se mantiene)</p> <p>P22: Elementos innecesarios (se mantiene)</p> <p>P23: Diámetro largo (Se mantiene, lo que cambia es el umbral de 12.2 a 7.92)</p> <p>P24: Alta diversidad de compuertas (se mantiene)</p> <p>P25: Uso de compuertas de división de tipo <i>or</i> (se mantiene)</p> <p>P26: Alto grado de complejidad</p>	
--	--	--	--

Anexo 5. Diagrama de paquetes: auxiliar

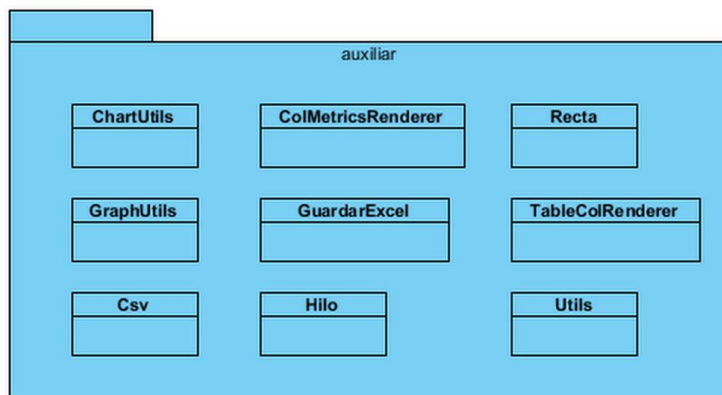


Figura A.8: Paquete auxiliar

Anexo 6. Diagrama de paquetes: bpmn_data

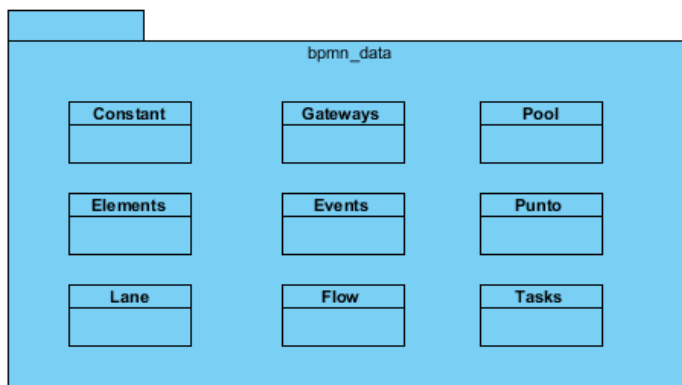


Figura A.9: Paquete bpmn_data

Anexo 7. Diagrama de paquetes: directrices

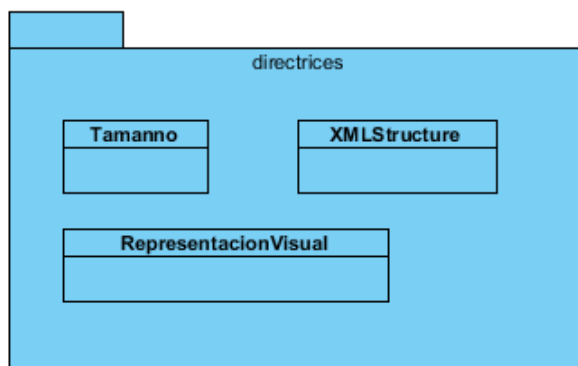


Figura A.10: Paquete directrices

Anexo 8. Diagrama de paquetes: grafo

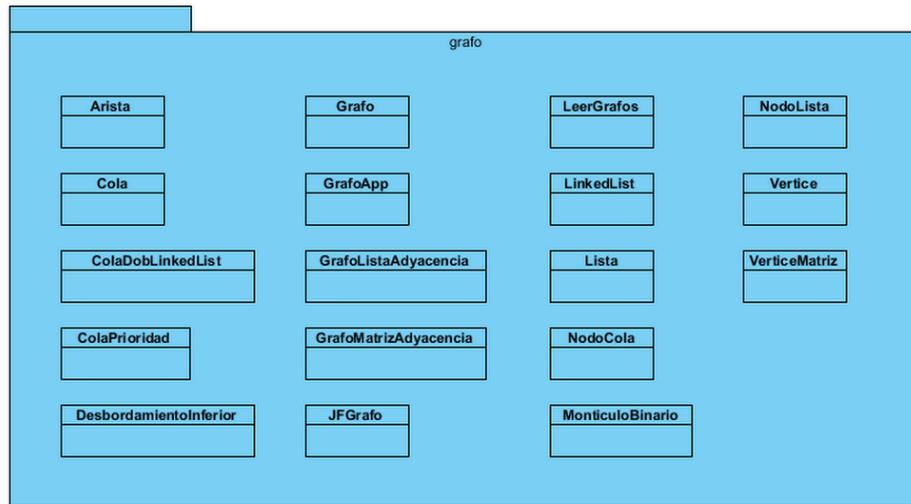


Figura A.11: Paquete grafo

Anexo 9. Diagrama de paquetes: ManagementFile

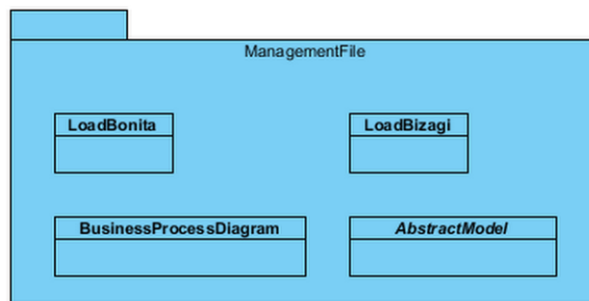


Figura A.12: Paquete ManagementFile

Anexo 10. Diagrama de paquetes: TDA

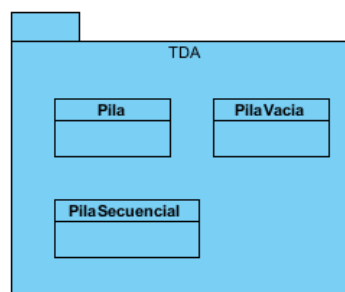


Figura A.13: Paquete TDA

Anexo 11. Diagrama de paquetes: usingtreetagger

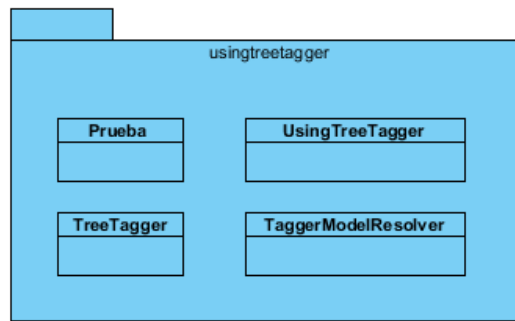


Figura A.14: Paquete usingtreetagger

Anexo 12. Diagrama de paquetes: visual

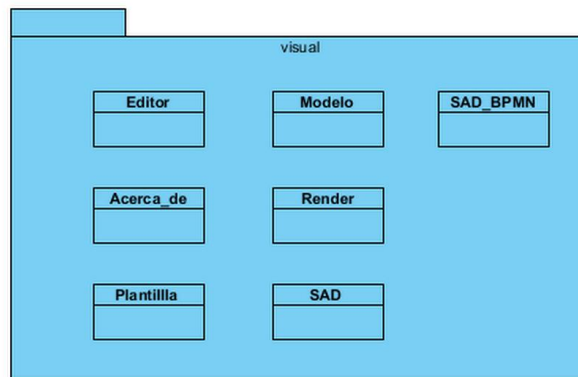


Figura A.15: Paquete visual

Anexo 13. Caso de prueba para sección Cargar modelo de proceso de negocio

Tabla TA.3 Caso de prueba para sección *Cargar modelo de proceso de negocio*

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Cargar proceso de negocio	EC 1.1: Flujo Normal.	El caso de uso se inicia cuando el modelador (usuario) desea cargar un modelo de proceso de negocio generado en alguna herramienta de modelación de éste tipo de proceso. Como resultado se habrá cargado correctamente el modelo, se habilitarán varias opciones y se mostrarán los datos sobre dicho modelo.	<p>7. El usuario selecciona en la parte superior izquierda de la interfaz la opción Cargar.</p> <p>8. El sistema muestra un cuadro de diálogo para que el modelador seleccione el modelo a cargar.</p> <p>9. El modelador selecciona el a a cargar.</p> <p>10. El sistema habilita las opciones que dependen de que se haya cargado un modelo de proceso de negocio.</p> <p>11. Se muestran los datos del proceso de negocio en una tabla.</p>

	<p>EC 1.2: Flujos alternativos</p> <p>Extensión errónea</p>	<p>El modelador desea cargar un modelo de proceso de negocio pero al hacerlo intenta cargar un archivo de extensión incorrecta.</p>	<ol style="list-style-type: none"> 1. El usuario selecciona en la parte superior izquierda de la interfaz la opción Cargar. 2. El sistema muestra un cuadro de diálogo para que el modelador seleccione el modelo a cargar. 3. El modelador selecciona el archivo a cargar. 4. El sistema muestra un mensaje de error notificando que el modelo no ha sido generado por una herramienta soportada.
	<p>Estructura del modelo errónea</p>	<p>El modelador desea cargar un modelo de proceso de negocio, pero al hacerlo intenta cargar un archivo de extensión correcta, pero estructura errónea.</p>	<ol style="list-style-type: none"> 1. El usuario selecciona en la parte superior izquierda de la interfaz la opción Cargar. 2. El sistema muestra un cuadro de diálogo para que el modelador seleccione el modelo a cargar. 3. El modelador selecciona el archivo a cargar. 4. El sistema muestra un mensaje de error notificando que el modelo no ha sido generado por una herramienta soportada o posee URL errónea.

Anexo 14. Caso de prueba para sección Calcular métricas de calidad

Tabla TA.4 Caso de prueba para sección *Calcular métricas de calidad*

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Calcular métricas de calidad	EC 1.1: Flujo Normal.	El caso de uso se inicia cuando el modelador (usuario) ha cargado correctamente un modelo de proceso de negocio y se impulsa a evaluar dicho modelo teniendo como base directrices prácticas. El resultado de este proceso será la visualización del cálculo de las métricas en gráficos y tablas.	1. El modelador carga el modelo de proceso de negocio. 2. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 3. El modelador selecciona las directrices que desea evaluar. 4. Se selecciona la opción Evaluar por parte del modelador. 5. El sistema calcula las métricas y muestra los resultados en los gráficos y tablas.
	EC 1.2: Flujo alternativo Directrices no seleccionadas	El modelador pretende evaluar el modelo de proceso de negocio, pero una vez cargado el modelo pretende evaluar el modelo sin seleccionar alguna directriz.	1. El modelador carga el modelo de proceso de negocio. 2. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 3. El modelador no selecciona alguna de las directrices seleccionables. 4. Se selecciona la opción Evaluar por parte del modelador.

Escenarios a probar en el caso de uso Calcular métricas de calidad

Tabla TA.5 Escenarios a para sección *Calcular métricas de calidad*

Id del escenario	Escenario	Variable 3 <i>Extensión del archivo</i>	Variable 4 <i>Estructura del modelo</i>	Resultado del Sistema	Resultado de la Prueba
EC1	Cargar modelo	Válido	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Válido	Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando.	
		Inválido	Válido		
		Inválido	Inválido		

Anexo 15. Caso de prueba para sección Exportar datos

Tabla TA.6 Caso de prueba para sección *Exportar datos*

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: Exportar datos	EC 1.1: Flujo Normal.	El caso de uso se inicia cuando el modelador (usuario) ha cargado correctamente un modelo de proceso de negocio y se impulsa a evaluar dicho modelo teniendo como base directrices prácticas. El resultado de este proceso será la visualización del cálculo de las métricas en gráficos y tablas.	6. El modelador carga el modelo de proceso de negocio. 7. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 8. El modelador selecciona las directrices que desea evaluar. 9. Se selecciona la opción Evaluar por parte del modelador. 10. El sistema calcula las métricas y muestras los resultados en los gráficos y tablas.
	EC 1.2: Flujo alternativo Directrices no seleccionadas	El modelador pretende evaluar el modelo de proceso de negocio, pero una vez cargado el modelo pretende evaluar el modelo sin seleccionar alguna directriz.	5. El modelador carga el modelo de proceso de negocio. 6. El sistema habilita la funcionalidad de las opciones que requieren que se encuentre cargado algún modelo de proceso de negocio. 7. El modelador no selecciona alguna de las directrices seleccionables. 8. Se selecciona la opción Evaluar por parte del modelador.

Escenarios a probar en el caso de uso Exportar datos

Tabla TA.7 Escenarios a para sección *Exportar datos*

Id del escenario	Escenario	Variable 3 <i>Extensión del archivo</i>	Variable 4 <i>Estructura del modelo</i>	Resultado del Sistema	Resultado de la Prueba
EC1	Cargar modelo	Válido	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Válido	Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando.	
		Inválido	Válido		
		Inválido	Inválido		

Tabla TA.8 Escenario a para sección *Exportar datos*

Id del escenario	Escenario	Variable 3 <i>Selección de directrices</i>	Resultado del Sistema	Resultado de la Prueba
EC2	Calcular métricas de calidad	Válido	El sistema habilita las funcionalidades que necesitan del modelo.	Prueba superada exitosamente.
		Inválido	El sistema lanza un mensaje de alerta indicando el tipo de error que se está manifestando y muestra una animación en el área de selección de directrices.	