



UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS

FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN

DEPARTAMENTO DE COMPUTACIÓN

Trabajo para optar por el Título Académico Máster en Ciencia de la Computación

**Herramienta para la aplicación de directrices de
complejidad a modelos de procesos de negocios**

Autor:

Lic. Frank Reyes García

Tutores:

Dra. Isel Moreno Montes de Oca

Dr. Abel Rodríguez Morffi

Santa Clara

Julio 2016

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la de la Maestría en Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma total como parcial y que además no podrá ser presentado en eventos ni publicado sin previa autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos 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 Seminario de
Tecnologías de Programación y Sistemas
de Información

Agradecimientos

A mi tutora por dedicarme gran parte de su tiempo, por todos los caminos que me enseñó y sobre todo por la paciencia que tuvo conmigo.

A mi cuadrilla de 5 inseparables, que juntos somos casi invencibles: Claudia, Addele, Alejandro y Mario Amores.

A Bea y a Erick por ayudarme en todo lo posible para la realización de este trabajo.

A súper Millo por ser el consultor y amigo en todo momento.

A la Dra. Gladys Casas y Dra. Leticia Arco por sus magníficos aportes y por su apoyo incondicional.

A todos los que han trabajado conmigo en el desarrollo de este trabajo: Lisbet, Arturo, Waldo.

A todos que de una forma u otra aportaron en este trabajo

A mi familia que sin ellos no fuera posible nada de esto y por la fuerza que me brindan.

A mi novia que siempre es una luz cuando todo se me pone oscuro.

Dedicatoria

A mi madre por todo su apoyo y cariño incondicional

A mi hermana por ser paciente conmigo en la casa

A Armando por ser la figura paterna que necesito

A mi novia por ser amiga y compañera y estar en todo momento junto a mí.

RESUMEN

En la actualidad, el modelado de procesos de negocios constituye una parte esencial de las organizaciones en la mejora de su eficiencia y efectividad operacionales. Una forma de incrementar la calidad de los modelos de procesos de negocio es a través de la aplicación de directrices prácticas de complejidad. El objetivo de este trabajo consiste en desarrollar una herramienta para evaluar la complejidad de modelos de procesos de negocio BPMN mediante la aplicación de medidas de calidad a archivos de salida de herramientas de modelado, con el fin de ofrecer directrices prácticas que faciliten la mejora de la calidad de dichos modelos. Los principales resultados son la interpretación de los archivos de salida *.bpmn* y *.xpdl* de las herramientas de modelado Bonita y Bizagi respectivamente; el almacenamiento en la estructura de datos grafo la información de los modelos contenida en dichos archivos; el desarrollo de la herramienta SAD_BPMN para evaluar modelos de procesos de negocio BPMN y contribuir a la mejora de la calidad de los mismos.

ABSTRACT

At present, business processes modeling is an essential part of organizations when improving operational efficiency and effectiveness. One way to increase the quality of business process models is through the implementation of practical guidelines related with models complexity. The aim of this study consists in evaluating the complexity of BPMN business processes models by implementing quality measures to output files of modeling tools, in order to provide practical guidelines that facilitate the improvement of the quality of these models. Main results are: the interpretation of *.bpmn* and *.xpdl* files that are provided as results of the modeling process in Bonita and Bizagi tools; the storing of the information contained in the models using graph data structure, the design of SAD_BPMN tool to evaluate BPMN business process models and the contribution to the improvement of business process models quality.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
Capítulo 1 Procesos de negocio, calidad, lenguajes y herramientas de modelado	5
1.1. Procesos de negocio	5
1.2. Modelado de procesos de negocios	6
1.3. Calidad de los modelos de procesos de negocio	6
1.4. Lenguaje para el modelado de procesos de negocio	10
1.4.1. Otras características de BPMN	13
1.5. Herramientas de modelado en BPMN	14
1.6. Interpretación de archivos de salida	18
1.7. Grafos	22
1.7.1. Representaciones para grafos	23
1.7.2. Aplicaciones de los grafos.....	24
1.8. Conclusiones parciales	25
Capítulo 2 Evaluación de modelos de proceso de negocio utilizando teoría de grafos ...	26
2.1. Directrices prácticas	26
2.1.1 Directrices prácticas relacionadas con el tamaño	27
2.1.2 Directrices prácticas relacionadas con la morfología	33
2.2. Extracción de datos de los modelos.....	39
2.3. Otros aspectos de implementación de las directrices prácticas de complejidad	42
2.4. Requisitos funcionales y no funcionales del sistema	46
2.5. Diagrama de casos de uso.....	47

2.6.	Diagramas de secuencia	49
2.7.	Diagrama de paquetes y clases del diseño	49
2.8.	Modelo de componentes	54
2.9.	Conclusiones parciales	54
Capítulo 3 Herramienta para la evaluación de modelos de procesos de negocio SAD_BPMN.....		55
3.1.	Descripción de la herramienta	55
3.2.	Pruebas y evaluación de la herramienta.	59
3.2.1.	Pruebas unitarias	60
3.2.2.	Pruebas de integración	64
3.3.	Análisis de la aceptación de la herramienta SAD_BPMN por parte de los usuarios.....	66
3.4.	Conclusiones parciales	67
CONCLUSIONES		68
RECOMENDACIONES		69
REFERENCIAS BIBLIOGRÁFICAS		70
Anexo 1. Modelo “Gestación y aprobación de PD”.....		79
Anexo 2. Encuesta a los usuarios de la herramienta SAD_BPMN.....		80

LISTA DE FIGURAS

Figura 1.1. Soporte de las directrices por las herramientas.....	17
Figura 1.2. Grado de soporte de las directrices de complejidad.	17
Figura 1.3: Ejemplo de modelo de proceso de negocio	18
Figura 1.4: Estructura del fichero <i>.bpmn</i> generado por Bonita.	20
Figura 1.5: Estructura del fichero <i>.xpdl</i> generado por Bizagi.	22
Figura 2.1: a) Proceso con más de dos evento inicio b) sustitución de varios eventos de inicio por uno.	29
Figura 2.2: a) Proceso con más de dos eventos de fin b) sustitución de varios eventos de fin por uno.	30
Figura 2.3: Modelo de proceso de negocio BPMN (Caso estudio #1).....	32
Figura 2.4 a) Modelo de proceso de negocio no estructurado y b) versión equivalente bien estructurada.	33
Figura 2.5 Proceso oficina de adquisición de libros (Caso estudio #2).	40
Figura 2.6: Seudografo resultante de la transformación del proceso.	41
Figura 2.7: Patrón Estrategia.	42
Figura 2.8 Diagrama de casos de uso de la herramienta.	47
Figura 2.9: Diagrama de secuencia Cargar modelo.	49
Figura 2.10: Diagrama de paquetes de SAD_BPMN.....	50
Figura 2.11: Diagrama del paquete <i>Directrices</i>	51
Figura 2.12 : Diseño de clases del paquete <i>bpmn_data</i>	52
Figura 2.13: Paquete <i>Grafo</i>	53

Figura 2.14: Paquete visual.	53
Figura 2.15 Diagrama de componentes.	54
Figura 3.1: Interfaz gráfica de SAD_BPMN.	56
Figura 3.2: Selección del fichero a evaluar por la herramienta.	56
Figura 3.3: Ventana principal con un modelo seleccionado.	57
Figura 3.4: Selección de directrices a evaluar en el modelo.	58
Figura 3.5: Directrices prácticas de complejidad sugeridas al modelador.	59
Figura 3.6: Prueba del método <i>load</i> para el caso 3.	61
Figura 3.7: Resultado satisfactorio de la prueba <i>testLoad</i>	61
Figura 3.8: Métodos diseñados para las pruebas de la <i>directriz_Problema1</i> y <i>metrica_SN</i>	63
Figura 3.9: Resultado satisfactorio de la prueba <i>directriz_Problema1</i>	63
Figura 3.10: Métodos diseñados para las pruebas de <i>directriz_Problema4</i> y <i>metrica_TNSE</i>	64
Figura 3.11: Resultado satisfactorio de la prueba <i>directriz_Problema4</i>	64
Figura 3.12: Resultado de la prueba de integración de evaluar el modelo.	65

LISTA DE TABLAS

Tabla 2.1 Problemas de tamaño presentes en el modelo Caso estudio #1.	31
Tabla 2.2: Problemas de morfología presentes en el modelo Caso estudio #1.	39
Tabla 2.3: Requisitos no funcionales del sistema.	46
Tabla 2.4 Resumen de los casos de uso de la herramienta.....	48
Tabla 3.1: Correlación entre resultados de la encuesta para las preguntas ocho, nueve y diez.	67

INTRODUCCIÓN

En las últimas décadas, los adelantos tecnológicos han abierto nuevas posibilidades a las organizaciones para realizar sus negocios, lo cual, entre otras causas, provoca que las organizaciones deban ser capaces de reaccionar ágilmente a los cambios e integrar los rápidamente con el fin de aumentar la competitividad del negocio [1-3]. Esto implica acciones de análisis, diseño, implementación, y una readaptación de los procesos organizacionales. Por esta razón, es relevante comunicar, entender y manejar el dominio de la información y los procesos, para comprender el fin y realizar una correcta gestión de requerimientos basada en la comprensión de las actividades de cualquier organización [4]. Esta idea se asume por el concepto de “Gestión de Procesos de Negocio” (BPM), el cual constituye una forma esencial de controlar y gobernar los procesos de negocio para cualquier organización con el objetivo de mejorar su rendimiento empresarial.

Una parte esencial de BPM como concepto de gestión, es el modelado de procesos de negocio, que requiere una abstracción de los procesos del mundo real con el fin de asignarlos a los modelos de procesos. Sólo entonces, los modelos pueden servir mejor a su propósito de modelado y ser útiles para la gestión de procesos de negocio. De ahí que sus objetivos principales son: a) mejorar la comprensión de una situación para que pueda comunicarse entre los inversionistas del negocio y b) usar el proceso como una herramienta para lograr las metas de un proyecto de desarrollo de procesos [5]. Aunque el modelado de procesos de negocio tiene varios años de desarrollo, las investigaciones sobre los aspectos relacionados con su calidad son relativamente recientes [6]. La calidad en el modelado de procesos de negocio puede definirse de acuerdo a [7] como "el cumplimiento de todas las propiedades deseables de un modelo para satisfacer las necesidades de los usuarios del modelo". 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 [8]. Debido a que grandes proyectos en la documentación

de procesos se confían a modeladores noveles y no expertos, una guía adecuada en el modelado es de particular importancia [9].

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.

A pesar de la reconocida importancia de directrices prácticas para el modelado de procesos de negocio [10], las mismas carecen de un soporte automático que permita su evaluación y posterior sugerencia a los modeladores para lograr modelos de procesos de negocio con la mayor calidad posible. Por lo anterior, el **problema de investigación** se enmarca en la ausencia de una herramienta computacional que agrupe directrices de calidad y que facilite la evaluación de modelos de procesos de negocio a partir de estas.

Este trabajo se centra en directrices prácticas que tienen como objetivo mejorar la calidad de los modelos de procesos de negocio en la Notación de Modelado de Procesos de Negocio (*Business Process Modeling Notation*; BPMN) a partir de la disminución de la complejidad de los mismos.

El **objetivo general** de la investigación consiste en desarrollar una herramienta para evaluar la complejidad de modelos de procesos de negocio BPMN mediante la aplicación de medidas de calidad a archivos de salida de herramientas de modelado, con el fin de ofrecer directrices prácticas que faciliten la mejora de la calidad de dichos modelos.

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

1. Seleccionar herramientas de modelado factibles para la interpretación y evaluación de sus archivos de salida que contienen modelos de procesos de negocio.
2. Extraer la información necesaria para la evaluación del modelo de proceso de negocio del archivo de salida que lo contiene.
3. Desarrollar una herramienta para mostrar directrices prácticas de calidad a partir de la evaluación del modelo.

Las **preguntas de investigación** planteadas son:

1. ¿Qué herramientas de modelado ofrecen una salida completa y adecuada para el análisis de los modelos?
2. ¿Qué estructura de datos facilita el almacenamiento y análisis de la información contenida en los archivos de salida sobre los modelos de procesos de negocio?
3. ¿Cómo evaluar los modelos de procesos de negocio para la aplicación directrices prácticas?

El **valor práctico** del trabajo está dado por:

- El sistema SAD_BPMN que dispone de los métodos para evaluar los modelos y a partir de ahí ofrecer las directrices prácticas necesarias a los modeladores. Este sistema facilita a los investigadores y docentes revisar modelos de procesos de negocio tomando como base la integración del conocimiento sobre calidad de los modelos de procesos de negocio que se encuentra en la literatura y obtener una idea de qué acciones llevar a cabo para mejorar la calidad de los mismos.

El **valor metodológico** del trabajo está dado por:

- La herramienta para la evaluación de modelos de procesos de negocio SAD_BPMN ofrece una sistematización de la calidad de los modelos de procesos de negocio lo cual tiene repercusiones docentes pues esta herramienta puede ser utilizada en cursos de pregrado y postgrado que se centren en la enseñanza de esta temática.

La **tesis** está **estructurada** en tres capítulos. En el capítulo 1 se tratan aspectos relacionados con BPM y el modelado en este contexto. Se analizan los principales lenguajes y herramientas de modelado de procesos de negocio, así como una estructura de datos para almacenar y analizar los archivos con información de modelos de procesos de negocio. El capítulo 2 describe los principales diagramas de la herramienta y expone las directrices prácticas de complejidad y medidas asociadas a estas para la evaluación de modelos de procesos de negocio. El capítulo 3 ofrece una descripción de la herramienta SAD_BPMN para la evaluación de modelos de procesos de negocio tomando como base directrices prácticas de complejidad, contribuyendo de este modo a la mejora de la calidad

de modelos de procesos mediante el ofrecimiento de sugerencias a los modeladores y una evaluación de la misma a través de pruebas unitarias y de integridad. Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas y los anexos.

Capítulo 1 Procesos de negocio, calidad, lenguajes y herramientas de modelado

En este capítulo se aborda la gestión de procesos de negocio como punto de partida para el desarrollo de software. Una vez formalizado el concepto proceso de negocio en entornos BPM se presenta el modelado como fase de vital importancia para la gestión de estos, así como la descripción de lenguajes y herramientas para su representación. Se analizan distintos lenguajes para el modelado de procesos de negocios así como un conjunto de herramientas de modelado. Se presentan investigaciones relacionadas con la calidad en el modelado de procesos de negocio. Finalmente se analiza la estructura de los archivos de salida que contienen modelos de procesos de negocio creados por dos herramientas de modelado y se define una estructura de datos para representar el modelo con el objetivo de evaluar la calidad del mismo.

1.1. Procesos de negocio

Para conseguir los objetivos de una empresa, sus actividades se organizan por medio de un conjunto de procesos de negocio, los cuales se pueden identificar a partir de la determinación y estructuración de sus objetivos estratégicos [11]. Se han propuesto varias definiciones para procesos de negocio en la literatura. Según [12], un proceso de negocio consiste en un conjunto de actividades que se desarrollan en coordinación con un ambiente técnico y organizacional. Cada proceso de negocio es ejecutado por una organización pero puede interactuar con otras organizaciones. Por otra parte, [13] plantea que un proceso simplemente es “un conjunto estructurado y medible de actividades diseñadas para producir un producto especificado, para un cliente o mercado específico, que implica un fuerte énfasis en CÓMO se ejecuta el trabajo dentro de la organización, en contraste con el énfasis en el QUÉ, característico de la focalización en el producto”. En [14] se define un proceso de negocio como un conjunto de actividades interrelacionadas, iniciadas en respuesta a un evento, que permite obtener un resultado específico para el cliente del proceso. El evento se hace corresponder con la petición específica del resultado que genera el proceso, y el cliente sería el beneficiario del resultado generado por el proceso de negocio. En [15] se plantea que todo proceso de negocio tiene un objetivo definido, por lo que todo proceso de negocio debe definir cómo la organización consigue alcanzar ese objetivo.

Por su parte, la organización que escribe sobre estándares de flujos de trabajo y BPM, WfMC (*Workflow Management Coalition*), define un proceso de negocio como un conjunto de uno o más procedimientos o actividades que de manera conjunta permiten alcanzar un objetivo de negocio, normalmente dentro del contexto de una estructura organizacional que define roles y relaciones.

De manera general se puede concluir 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. Modelado de procesos de negocios

Un modelo es una abstracción o representación simplificada de un sistema complejo que puede ser real o conceptual. Un modelo se diseña para mostrar ciertas características del sistema que se desea estudiar, predecir, modificar o controlar. Por tanto un modelo incluye determinados aspectos, pero no todos, del sistema que se pretende analizar [11].

Los modelos de procesos de negocio ayudan a formalizar el flujo de trabajo de las organizaciones mediante la descripción de las actividades requeridas para cumplir con los objetivos del negocio. Los modelos de procesos de negocio ayudan a incrementar la eficiencia de los Sistemas de Información (SI), y permiten reducir la complejidad para entender el comportamiento relevante [13].

El objetivo fundamental de un modelo de proceso de negocio es facilitar su comprensión por parte de los involucrados, también proporciona las bases para la formalización del proceso lo cual ayuda a verificar propiedades de interés [14]. Por tanto, un modelo de proceso de negocio pretende describir qué es lo que el proceso o sistema hace, qué o quién lo controla, en qué está trabajando, qué elementos requiere para desarrollar sus funciones y qué produce, y el modelado de estos procesos de negocio es la descripción de dichos modelos mediante lenguajes y herramientas destinadas para ello.

1.3. Calidad de los modelos de procesos de negocio

El mantenimiento y aseguramiento de la calidad de los modelos de procesos de negocios son factores claves [16] debido a que modelos de baja calidad aumentan el esfuerzo de desarrollo o no cumplen con las expectativas de los usuarios. Según [17], un modelo debe

poseer tres nociones de calidad. Primeramente, una correspondencia que establece una representación de un origen natural o artificial. Segundo, solo se representan en el modelo los atributos relevantes del origen y el resto se omite. Es por esto que el modelo ofrece una abstracción en términos de un homomorfismo en un sentido matemático [18]. Tercero, el modelo se usa por el modelador en lugar del origen en un cierto momento o con cierto objetivo. En los últimos años han surgido diversas propuestas para evaluar aspectos de los procesos de negocio a partir del modelo que los representa gráficamente. Algunas de estas propuestas se basan en la noción de que al evaluar el proceso en fases tempranas de desarrollo, es posible obtener información sobre la complejidad del proceso, predecir la probabilidad de error, etc. [19], ya que es más difícil de entender y modificar un modelo con un alto grado de complejidad. Es por ello que con una adecuada medición a nivel conceptual, se pueden obtener modelos más fáciles de entender y esto facilita elegir entre varias alternativas de modelado. De esta manera, una forma de evaluar la calidad del modelo es analizarlo desde el punto de vista estructural. Según [20] la calidad de un producto final depende en gran medida de la precisión durante la etapa de la especificación de requisitos y está centrada en la mejora en las fases tempranas de desarrollo.

Otros estudios sobre la calidad de los modelos de procesos de negocio consisten en la creación de marcos de trabajo para la calidad [20-25]. Los marcos de trabajo intentan producir orden y esclarecimiento a la representación de la calidad en los modelos conceptuales y a la calidad del proceso de modelado conceptual [20], definiendo los niveles de calidad que un determinado modelo debe cumplir [26]. No obstante, solo una pequeña porción de investigadores trabajan en el ámbito de calidad vinculado a los marcos de trabajo [27, 28]. Existen otros trabajos relacionados con la calidad de los modelos de procesos de negocios que investigan medidas de calidad [29], las que consideran aspectos estructurales de un modelo de proceso de negocio. [30]. La calidad interna de los modelos de procesos puede verse desde diversas aristas, y por ende, debe ser cuantificada con más de una medida para obtener la mayor cantidad de información posible de un modelo dado. Los resultados de la evaluación de las medidas son más precisos si se utilizan valores críticos, que puedan indicar cuándo la calidad comienza a declinar. Estos valores límites

se conocen como umbrales, y pueden tratarse como alarmas que ofrecen alertas cuando hay baja calidad.

En [31] se ofrece un amplio conjunto de medidas de calidad, subdividido en cinco partes como se relata a continuación:

Medidas de tamaño: Varios documentos apuntan a “tamaño” como un factor importante para la comprensión de los modelos de procesos de negocio dado que a menudo es relacionado con el número de nodos del modelo de proceso. Estas Medidas ofrecen las cantidades de los diferentes elementos del modelo; se hace un conteo de los elementos (o nodos) del modelo, se pueden contar todos o hacer una selección de los principales y contar solo esos [32]. Un ejemplo de una medida de tamaño es *número de elementos*, que se describe a continuación:

Definición:	Número de elementos [31]	
Medida:	$S_N(G) = N $	(1)
	Donde N es el número de nodos del modelo de proceso. Valor de umbral asociado [33]: 31	

Medidas de densidad: Incluye aquellas medidas que relacionan el número de nodos con el número de arcos del modelo. Ejemplo de medida de densidad es *grado promedio de las compuertas*, que se expone a continuación:

Definición	Grado promedio de las compuertas [31].	
Medida:	$\bar{d}_C(G) = \frac{1}{ C } \sum_{c \in C} d(c)$	(2)
	Valor de umbral según [34]: 3.83	

Medidas de modularidad: En este grupo se incluyen las medidas que cuantifican aquellos aspectos de un modelo de proceso que conciernen la interrelación de los subcomponentes del modelo con el modelo global; pueden dar proporciones del nivel de separación, de secuencialidad, estructura y profundidad del modelo. Como ejemplo de medida de modularidad se presenta la medida *profundidad*:

Definición:	Profundidad [31].	
Medida:	$\Lambda(G) = \max\{\lambda(n) n \in N\}$	(3)
	Donde N es el conjunto de elementos del seudografo G del modelo de proceso, y $\lambda(n)$ es el mínimo entre $\lambda_{in}(n)$ y $\lambda_{out}(n)$. Para calcular $\lambda_{in}(n)$ se define un algoritmo donde para cada elemento n visitado, el	

nuevo valor $\lambda'_{in}(n)$ se actualiza tomando como base los valores de los elementos predecesores visitados y el valor actual de acuerdo a la siguiente regla:

$$\lambda'_{in}(n) = \begin{cases} \max\{\lambda_{in}(n), \lambda_{in}(pre) + 1\} & \text{if } pre \in S \wedge n \notin J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre)\} & \text{if } pre \in S \wedge n \in J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre)\} & \text{if } pre \notin S \wedge n \notin J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre) - 1\} & \text{if } pre \notin S \wedge n \in J \end{cases}$$

De manera análoga se define $\lambda'_{out}(n)$. Valor de umbral según [35]: 1

Medidas de conectividad: Presenta medidas relacionadas con los conectores y su interacción. Constituyen una medida de cómo está la desigualdad y la heterogeneidad de las compuertas en el modelo. Un ejemplo de medida de conectividad es la medida *incongruencia de las compuertas*, que se muestra a continuación:

Definición:	Incongruencia de las compuertas [31].
Medida:	$MM(G) = MM_{or} + MM_{xor} + MM_{and}$ (4)
	Donde $MM_l = \sum_{c \in S_l} d(c) - \sum_{c \in J_l} d(c) $ y l es el tipo de compuerta del pseudografo G del modelo de proceso. Valor de umbral asociado según [35]: 4

Medidas de ciclicidad y concurrencia: Este grupo cuenta con medidas tales como la ciclicidad que brinda la relación del número de nodos en un ciclo con respecto al número de nodos en el modelo y la concurrencia que muestra cómo es el acumulado del grado de salida de las compuertas. Un ejemplo de medida en este subgrupo es la *ciclicidad*, descrita a continuación:

Definición:	Ciclicidad [31].
Medida:	$CYC_N(G) = \frac{ N_c }{ N }$ (5)
	Donde N_c es el número de elementos que pertenecen a ciclos, y N es el número total de elementos del modelo.
	Valor de umbral según [33]: 0.005
Definición:	Número de ciclos en el modelo [36].
Medida:	$S_{Ciclos}(G) = Ciclos $ (6)
	Donde <i>Ciclos</i> es el número de ciclos en el modelo.

Las medidas ofrecen la noción de características deseables en los modelos de procesos para lograr mayor comprensibilidad de los mismos y menor probabilidad de error desde el punto de vista cuantitativo. Sin embargo, no son lo suficientemente prácticas para servir de guía a los modeladores. 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 [8].

Las directrices prácticas se encuentran clasificadas dentro de dos grupos principales de acuerdo a [37]. Estos son: directrices de complejidad general y directrices de representación visual. Estos grupos a su vez se dividen en directrices de tamaño y de morfología para las directrices de complejidad general, y directrices de estilo de etiquetado y de diseño para las de representación visual. En [37] se proponen 25 directrices prácticas para el modelado de procesos de negocio. Para determinar si un modelo contiene los problemas asociados a las directrices, se usan medidas de calidad. Sin embargo, el cálculo manual de las medidas asociadas a las directrices con vistas a evaluar los modelos de procesos de negocio, es trabajoso y se complejiza a medida que aumenta el tamaño de los modelos a examinar. Es por ello que la existencia de una herramienta para la evaluación de estas medidas asociadas a las directrices es un aspecto de utilidad, sobre todo para la docencia en la enseñanza del modelado de procesos de negocio y para las investigaciones realizadas en esta área. Este trabajo particulariza en las directrices prácticas de complejidad general, con el fin de mejorar la calidad de los modelos de procesos de negocios. Las mismas ofrecen mejoras a los modelos desde el punto de vista tanto semántico como sintáctico en cuanto a la corrección [25, 38-40] de los modelos desde la perspectiva de la presencia de errores de comportamiento.

1.4. Lenguaje para el modelado 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.

Dentro de algunos lenguajes de modelado de proceso de negocio se pueden mencionar:

Redes de Petri: Las redes de Petri fueron creadas por el alemán Carl Adam Petri en 1962. En su tesis doctoral “Comunicación con autómatas”, establece los fundamentos para el desarrollo teórico de los conceptos básicos de las redes de Petri que representan una alternativa para modelar el comportamiento y la estructura de un sistema. Comparada con otros modelos de comportamiento dinámico gráficos, como los diagramas de las máquinas de estados finitos, las redes de Petri ofrecen una forma de expresar procesos que requieren sincronía. Y quizás lo más importante es que las redes de Petri pueden ser analizadas de manera formal y obtener información del comportamiento dinámico del sistema de modelado [41].

Diagramas de Actividades de UML: UML (*Unified Modeling language*) es un lenguaje ampliamente utilizado para modelar y diseñar sistemas que presentan facilidad de uso [42]. Según [43], los diagramas de actividad son los elementos de UML 2.0 utilizados para representar procesos de negocio y flujos de trabajo, y soportan gran variedad de dominios. De acuerdo a [44, 45], en estos diagramas la unidad fundamental de especificación del comportamiento es una acción, que toma un conjunto de entradas y genera un conjunto de salidas, y pueden modificar el estado del sistema. Las acciones pueden invocar una operación, modificar valores de objetos o realizar cálculos sobre sus entradas. También se utiliza el concepto de actividades o acciones [46].

EPC (*Event Process Chan*): ha sido desarrollado dentro del esquema de una Arquitectura de Sistemas de Información Integrados (*ARIS*) y se utiliza en el modelado, diseño, y reingeniería de los procesos de negocio. Esta arquitectura presenta una vista funcional, una vista organizacional y una vista de datos [31, 47]. Se puede utilizar para la configuración de planificaciones empresariales y para la mejora de procesos de negocio [48]. Dentro de sus componentes se encuentran los de tipo función para capturar actividades de un proceso, elementos tipo evento para describir pre- y pos-condiciones de

funciones y conectores lógicos que describen las relaciones lógicas entre los elementos (eventos y funciones) en el flujo de control [49].

IDEF - *Integrated Definition for Function Modelling*: Es una familia de técnicas de modelado que ofrecen una perspectiva integrada para representar y modelar procesos y estructuras de datos. La familia IDEF, consiste en un gran número de técnicas, entre las cuales se destaca IDEF0 e IDEF3, que son aquellas relacionadas con los procesos de negocio, aunque existen otras versiones como IDEF1, IDEF1X, IDEF2, IDEF4 e IDEF5. IDEF0, por una parte, ofrece una metodología para representar las actividades que conforman un sistema o empresa, permitiendo representar qué se persigue obtener, facilitando la representación de la visión estratégica y la comunicación entre usuarios no técnicos. Su elemento principal es la función, o actividad, la cual posee un control, una entrada, un mecanismo y una salida. IDEF3, por otra parte, es una metodología para representar el flujo de trabajo de un proceso, así como sus objetos participantes. Facilita la representación de cómo se lleva a cabo el trabajo así como la comunicación con el propietario mismo del proceso. Los componentes fundamentales que emplea IDEF3 en su representación son: unidad de trabajo, ligas, conexiones y referencias [50].

XPDL (*XML Process Definition Language*) fue creado por la WfMC en el año 2001 y el 3 de octubre del 2005 se liberó la versión 2.0. Su formato de archivo es basado en XML [51], el cual permite intercambiar modelos de procesos de negocios entre herramientas diferentes. Incluye elementos tales como el tamaño y las coordenadas X e Y de un nodo o las líneas que indican el camino a seguir. Los nodos y las líneas tienen atributos que pueden especificar información sobre la ejecución del proceso, tales como roles, descripción de actividades, temporizadores, llamadas a servicios (web o de otro tipo), etc. La versión 2.0 incluye extensiones para poder representar todos los elementos de BPMN.

BPMN (*Business Process Modeling Notation*): Ofrece una notación gráfica estandarizada para el modelado de procesos de negocio. Según la OMG (*Object Management Group*) [52], permite diseñar gráficamente procesos de negocio en su totalidad. La especificación describe dos niveles de abstracción en el proceso de modelado: un nivel detallado que contiene únicamente un conjunto de elementos gráficos utilizados por el analista de negocio y un nivel que establece la correspondencia entre estos elementos y un lenguaje

de ejecución. Uno de los objetivos de BPMN es proporcionar un mecanismo simple para crear modelos de procesos de negocio, pero reduciendo su complejidad.

Este trabajo se enfoca en modelos creados utilizando la notación BPMN, debido a que 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. Además, según [53], BPMN es una notación fácil de usar por los analistas del negocio para documentar y comunicar sus procesos de negocio dentro de las compañías y con los participantes externos del negocio, y posee una amplia riqueza semántica al permitir expresar todos los patrones de flujo de trabajo. Por esto, a continuación se dedica una sección a este lenguaje.

1.4.1. Otras características de BPMN

Esta notación está diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades de una forma clara, completa y eficiente [54]. Su objetivo es soportar la gestión de una forma que sea fácilmente entendida por todos los usuarios que participan en el proceso. Esta forma unificada de representar los procesos, facilita expresar los comportamientos y decisiones en una organización. Además permite que los diagramas y gráficos se conviertan en estructuras que se integran a las tecnologías de información que tiene la empresa para adelantar el proceso. Tales elementos tienen una relación uno a uno con instrucciones en el Lenguaje de Ejecución de Procesos de Negocio (BPEL), lo cual permite generar código ejecutable a partir de un modelo BPMN.

BPMN proporciona cuatro categorías básicas que permiten modelar los procesos de negocios. Estas categorías son de fácil comprensión y reconocimiento. Estas pueden estar sujetas a variaciones y se le puede añadir información adicional para soportar los requerimientos y complejidades cambiando el diseño del diagrama.

Las cuatro categorías fundamentales de BPMN son:

- **Objetos de flujo:** esta categoría incluye eventos, actividades y compuertas. Los eventos consisten en un suceso durante el curso del proceso de negocio. Afectan al flujo de proceso. Normalmente tienen una causa (disparador) o un

impacto (resultado). Dependiendo de cuando afectan al flujo serán eventos iniciales, intermedios o finales. Las actividades son términos genéricos para el trabajo que realiza una compañía. Pueden ser atómicas (tareas) o compuestas (sub-procesos). Las compuertas, por último, se utilizan para controlar la convergencia o divergencia de flujos. Representan una decisión para mezclar o unir caminos [55].

- **Objetos conectivos:** consisten en las líneas que conectan los elementos del diagrama a través de flujos de secuencia (conectan elementos del mismo pool), flujos de mensajes (conectan elementos de diferentes pools), y asociaciones (ayudan a la esclarecer las entradas y salidas) [55].
- **Contenedores:** estos componentes se usan para segmentar el diagrama en áreas que contienen una sucesión de actividades relacionadas por los flujos de secuencia con el objetivo de ilustrar capacidades funcionales distintas o responsabilidades. BPMN da soporte a dos construcciones principales, estas son:
 - **Pool:** actúa como contenedor gráfico para segmentar un conjunto de actividades procedentes de otros pools y pueden representar varias cosas dentro de una organización. Las actividades separadas por ellos se consideran procesos [56].
 - **Lanes:** Es una sub-partición dentro de un pool que se extiende a lo largo de este. Se utilizan para organizar y categorizar actividades [57].

Artefactos: Los artefactos son elementos que ayudan a notar el modelo, de modo que el lector pueda entender mejor el contexto de las actividades y acciones que se representan en el mismo.

Este trabajo se enfoca a los elementos de las categorías objetos de flujo, objetos conectivos y contenedores.

1.5. Herramientas de modelado en 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.

Para la realización de este trabajo se analizó un grupo de herramientas en su mayoría de código abierto y de libre acceso en la red. Estas permiten el modelado de los procesos bajo los términos de Licencia Pública General [58], que ha tomado mucho auge entre los modeladores de negocio [59, 60]. Algunas de las herramientas que se analizaron fueron:

- **Aris Express:** es una herramienta flexible y agradable para la vista de cualquier usuario. Como principal ventaja posee la posibilidad de efectuar modelamientos, ya que no restringe las conexiones y posee los diferentes tipos de tareas requeridos. Como desventaja presenta el hecho de que el modelo es más difícil de entender para los usuarios por el hecho de no permitir nombres en las conexiones. Sólo pueden agregarse anotaciones de texto, lo cual hace que no se cumpla la especificación BPMN [61].
- **Camunda:** es una aplicación para la edición de diagramas de procesos BPMN. Es muy fácil de usar, lo que significa que los analistas de negocios la pueden utilizar para desarrollar sus propios diagramas. Además del modelado visual, Camunda también permite editar todas las propiedades que son necesarias para la ejecución técnica. Desde la misma se puede trabajar directamente sobre archivos XML y BPMN, los desarrolladores la pueden combinar fácilmente con su IDE preferido (por ejemplo, Eclipse, Netbeans) [62].
- **Bonita:** es una herramienta de código abierto que permite el modelado, gestión de procesos de negocio y flujos de trabajo y ha tomado mucho auge entre los modeladores de negocio. Según un estudio realizado por [54] resultó ser la mejor de las herramientas de código abierto para modelar procesos de negocio usando la notación BPMN. Permite al usuario crear y modificar gráficamente los procesos de negocio con el estándar BPMN. También admite conectar procesos a otras piezas del sistema de información para generar una aplicación de negocios autónoma accesible como formulario web. Además, el usuario puede trabajar con procesos diseñados con otros estándares y tecnologías tales como XPDL o jBPM [63].

- **Signavio:** para trabajar con Signavio se hace online, lo que evita consumo de disco, permite el modelado de procesos de forma multilingüe y mantener los atributos de proceso y los procesos de enlace jerárquicamente. Todos los modelos de proceso se guardan en un repositorio central de proceso. Combina las ventajas de una solución de modelado profesional con las ventajas de un servicio en la nube flexible [64].
- **BizAgi Process Modeler:** modela procesos de negocio de un modo comprensible para los usuarios finales, para que luego sea automatizada, enganchándolos con una arquitectura de servicio (a través de servicios Web). Tiene la capacidad de simular el comportamiento de un sistema, medirlo y optimizarlo [4].
- **Visual Paradigm:** Es una herramienta CASE (*Computer Aided Software Engineering*) aplicable en todo el ciclo de vida del desarrollo de software. Soporta UML (*Unified Modeling Language*) y BPMN. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación además ofrece una función de animación de procesos de negocio [65].

Para la realización de este trabajo se utilizaron las herramientas de modelado de procesos de negocios, Bonita y Bizagi. Para ello se tuvo en cuenta el soporte que brindan estas herramientas a las directrices prácticas. Este soporte se determinó a partir de un estudio realizado en conjunto entre la Universidad K. Leuven y la Universidad Central “Marta Abreu” de Las Villas [66]. En el mismo se determina el soporte de las herramientas de modelado a un conjunto de directrices. Para lograrlo, se crearon 50 modelos BPMN que incumplían cada directriz y se introdujeron en las herramientas. De esta forma se pudo concluir que el 21% de casos revelaron algún tipo de soporte lo que indica un soporte individual de las directrices relativamente bajo. La mayoría de los desarrolladores de herramientas conocen las directrices pero existen grandes diferencias entre las herramientas en cuanto al soporte de las directrices (como se muestra en la Figura 1.1) donde es posible observar que las herramientas Bizagi y Bonita carecen de soporte para una parte significativa de las directrices de complejidad.

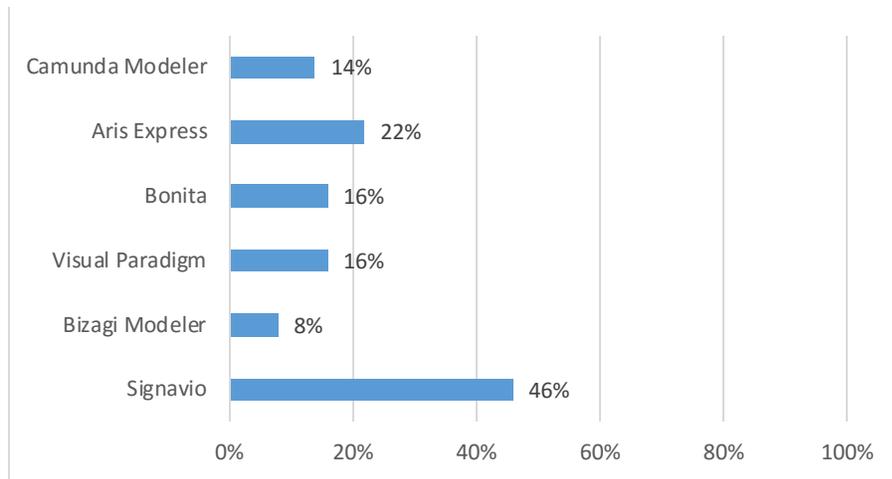


Figura 1.1. Soporte de las directrices por las herramientas.

La Figura 1.2 muestra una representación del grado de soporte por categoría por las herramientas antes mencionadas.

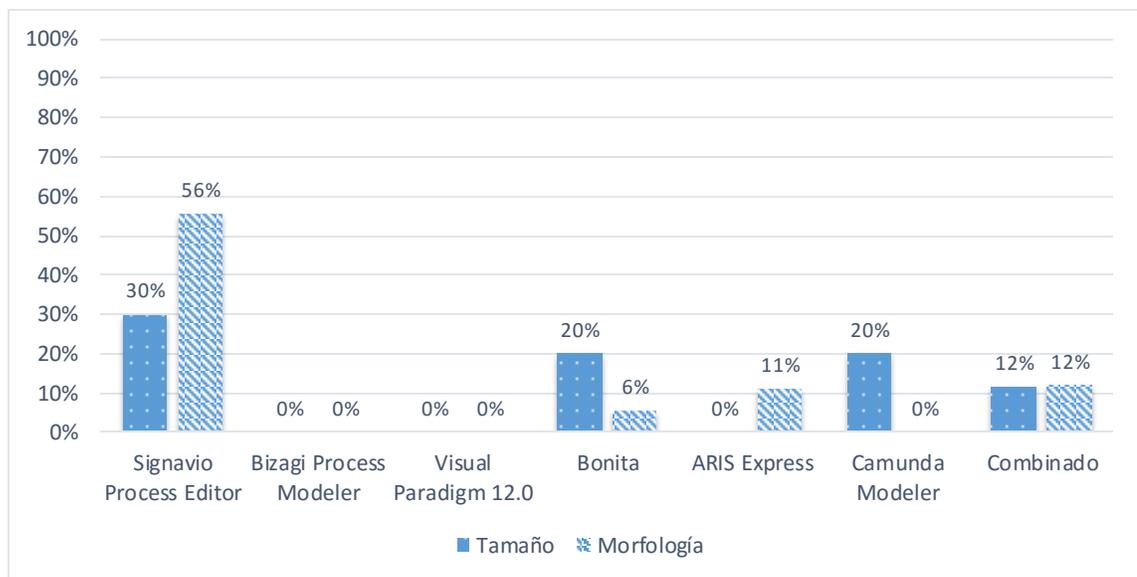


Figura 1.2. Grado de soporte de las directrices de complejidad.

Es posible observar como las herramientas seleccionadas no se soportan del todo las directrices de tamaño y morfología.

También se tuvo en cuenta el formato de exportación de las herramientas el cual se encuentra directamente ligado con la posibilidad de ejecutar procesos en donde se determina la facilidad para exportar la información. Ambas herramientas exportan la información del modelo a varios tipos de ficheros. Los formatos escogidos para este

trabajo fueron *.bpmn* de Bonita y *.xpdl* de Bizagi. Estos formatos son representados como ficheros XML donde se almacena toda la información del “dibujo” del seudografo que representa la definición del proceso. Por esta razón, incluye elementos tales como el tamaño y las coordenadas X e Y de un nodo o las líneas que indican el camino a seguir. Los nodos y las líneas tienen atributos que pueden especificar información sobre la ejecución del proceso, tales como roles, descripción de actividades, temporizadores, llamadas a servicios (web o de otro tipo), etc.

1.6. Interpretación de archivos de salida

Como se menciona en la sección anterior, se evalúan como parte de este trabajo ficheros exportados obtenidos a partir del modelado en Bizagi y Bonita con extensión *.bpmn* y *.xpdl*. En la Figura 1.3 se muestra un ejemplo de modelo de proceso de negocio el cual está compuesto por tres tareas, dos eventos (uno de inicio y otro de salida) y dos compuertas (una de división de flujo y otra de unión de flujo).

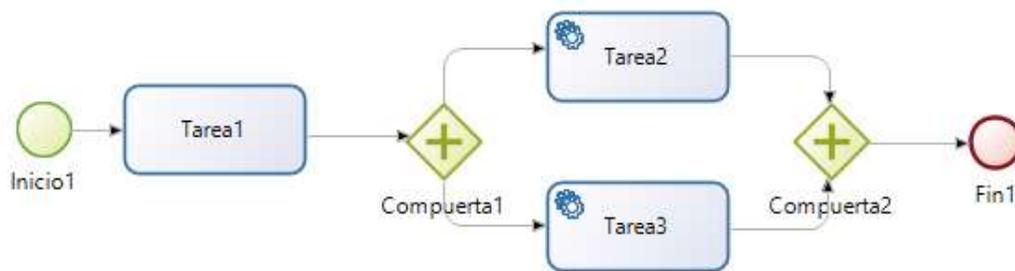


Figura 1.3: Ejemplo de modelo de proceso de negocio

El archivo *.bpmn* generado por la herramienta Bonita se divide en tres secciones principales:

1. Información de la herramienta: Muestra información detallada de la versión de la herramienta utilizada.
2. Información del proceso: Se muestra información detallada del proceso en sí, como el nombre del proceso y un identificador del mismo. Además en esta sección se describen todos los componentes involucrados en el modelado del proceso de negocio, los cuales se representan como elementos del propio fichero XML:

- Pool: Contiene el nombre del Pool y un identificador, así como otros datos propios del Pool del Lane: Se muestra el nombre y un identificador, así como datos propios de los Lanes.
 - Eventos: Se visualiza el tipo de evento, es decir, evento de inicio, intermedio o de fin. Además un identificador característico para cada uno y el nombre.
 - Tareas: Se representa el tipo de tarea, un identificador propio y el nombre, así como el tipo de tarea representada.
 - Compuertas: Se muestra el tipo de compuerta, un identificador y el nombre, y el identificador a cada conector asociado a dicha compuerta.
 - Conectores: Las conexiones entre los objetos de flujo de BPMN (eventos, actividades y compuertas) se representan en el archivo XML, con el encabezado `model:flowNodeRef`. En la descripción de cada conector se muestra el tipo de conector y un identificador, así como el identificador del elemento donde comienza la conexión (*source*) y el identificador del elemento donde finaliza dicha conexión (*target*).
3. Información de estilos y notación del diagrama: contiene información detallada de los estilos de cada elemento del diagrama (forma, dimensiones, colores, tipo de letra para el etiquetado) y las coordenadas en píxeles donde está ubicado cada elemento en el área de representación del esquema.

La Figura 1.4 muestra la estructura general del fichero *.bpmn* generado por la herramienta Bonita. En esta figura se aprecian las 3 secciones anteriormente descritas.

```

<?xml version="1.0" encoding="UTF-8"?>
<model:definitions Información de la herramienta>
  <model:process Información del pool 1>
    <model:laneSet Informacion de los Lane>
      <model:lane Descripción del Lane 1">
        <model:flowNodeRef>Identificador del flujo 1</model:flowNodeRef>
        .....
      </model:lane>
      <model:flowNodeRef>Identificador del flujo N</model:flowNodeRef>
      </model:lane>
      <model:lane Descripción del Lane N">
        <model:flowNodeRef>Identificador del flujo 1</model:flowNodeRef>
        .....
        <model:flowNodeRef>Identificador del flujo N</model:flowNodeRef>
      </model:lane>
    </model:laneSet Fin de 1>
    <model:Element Descripción del objeto de flujo 1/>
    .....
    <model:Element Descripción del objeto de flujo N/>
    <model:sequenceFlow Descripción del flujo de secuencia 1"/>
    .....
    <model:sequenceFlow Descripción del flujo de secuencia N"/>
  </model:process>
  .....
  <model:process Información del pool N>
    <model:laneSet Información de los Lane>
      <model:lane Descripción del Lane 1">
        <model:flowNodeRef>Identificador del flujo 1</model:flowNodeRef>
        .....
      </model:lane>
      <model:flowNodeRef>Identificador del flujo N</model:flowNodeRef>
      </model:lane>
      <model:lane Descripción del Lane N">
        <model:flowNodeRef>Identificador del flujo 1</model:flowNodeRef>
        .....
        <model:flowNodeRef>Identificador del flujo N</model:flowNodeRef>
      </model:lane>
    </model:laneSet Fin de 1>
    <model:Element Descripción del objeto de flujo 1/>
    .....
    <model:Element Descripción del objeto de flujo N/>
    <model:sequenceFlow Descripción del flujo de secuencia 1"/>
    .....
    <model:sequenceFlow Descripción del flujo de secuencia N"/>
  </model:process>
  <di:BPMNDiagram Información del diagrama>
    <di:BPMNPlane >
      <di:BPMNShape>Coordenadas, alto y ancho del Objeto de flujo 1 </di:BPMNShape>
      .....
      <di:BPMNShape>Coordenadas, alto y ancho del Objeto de flujo 1 </di:BPMNShape>
      <di:BPMNEdge>Coordenadas de los flujos de secuencia 1</di:BPMNEdge>
      .....
      <di:BPMNEdge>Coordenadas de los flujos de secuencia N</di:BPMNEdge>
      <di:BPMNLabelStyle >Información de estilos y notación del diagrama</di:BPMNLabelSt:
    </di:BPMNPlane>
  </di:BPMNDiagram>
</model:definitions>

```

Figura 1.4: Estructura del fichero *.bpmn* generado por Bonita.

Los archivos *.xpd* generados por la herramienta Bizagi muestran una estructura similar al formato explicado anteriormente. El archivo se divide en tres partes fundamentales: una primera parte donde se expone la información de la herramienta como autor, versión y hora de creado el modelo. En la segunda sección se muestra la información referente al proceso. Se exhiben todos los pool, así como los lanes pertenecientes a cada uno. Tanto

los lane como los pool contienen una descripción detallada de sus principales características: identificador, nombre, coordenadas y estilo de etiquetas. En este formato no existe una sección correspondiente al estilo y coordenadas de cada elemento del proceso debido a que esta información aparece comprendida en la definición de cada elemento. Como tercera sección tenemos lo referente a los componentes de flujo del proceso, las tareas, eventos y compuertas son representados a través de etiquetas *activity* las cuales diferencian el tipo de elemento al cual hace referencia a través de nodos hijos. El formato *.xpd* almacena la información para mostrar gráficamente los flujos de secuencia dentro del nodo *<Transitions>* el cual adiciona nodos *<Transition>* por cada flujo de secuencia que contiene el proceso. Los nodos *Transition* contienen un identificador así como los atributos *From* y *To* para indicar el origen y destino del mismo.

En la Figura 1.5 se muestra la estructura general del fichero *.xpd* generado por Bizagi descrito anteriormente.

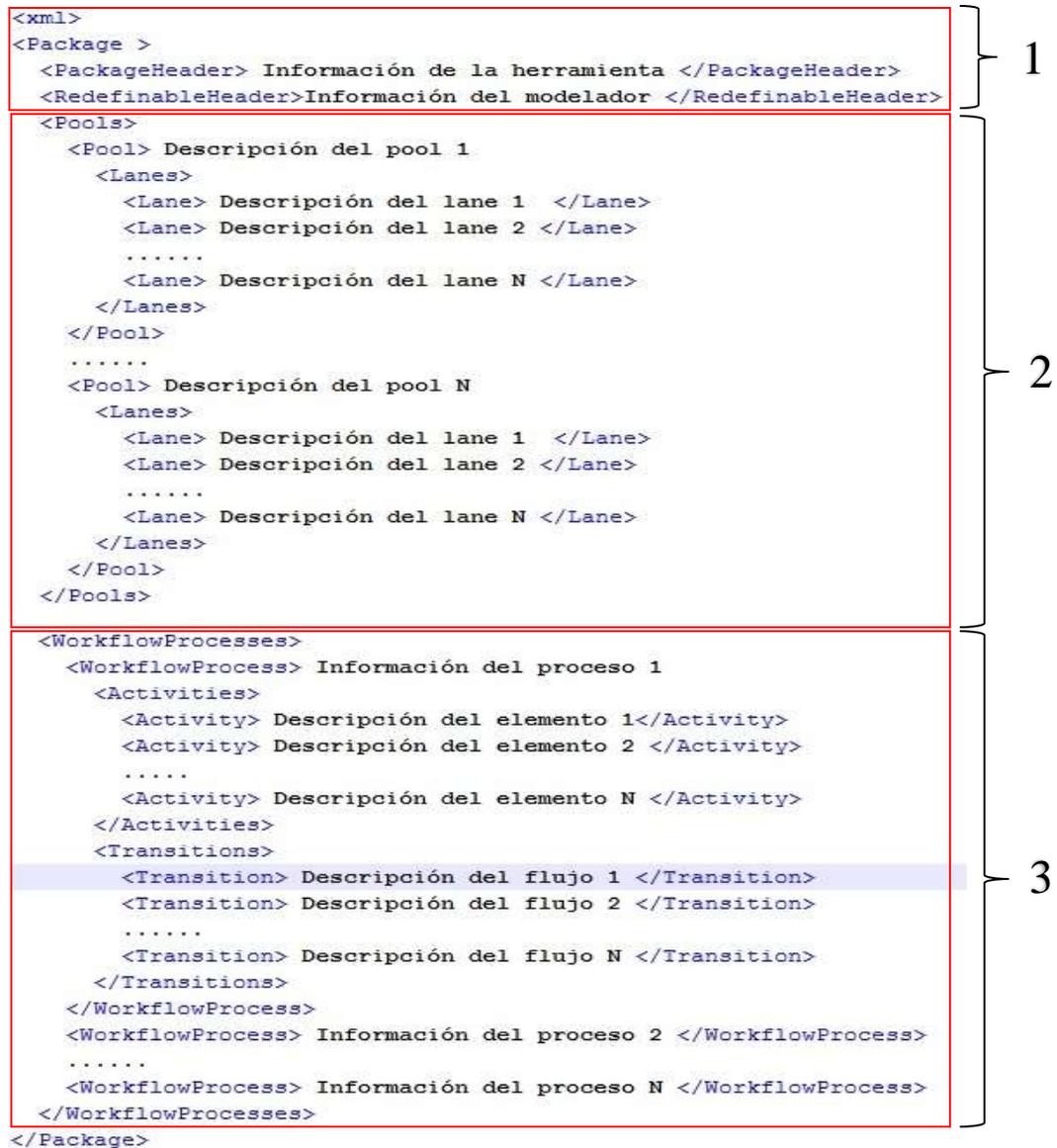


Figura 1.5: Estructura del fichero .xpd generado por Bizagi.

1.7. 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) [67].

A continuación se presenta un conjunto de definiciones y notaciones que se asumen en este trabajo:

- Un grafo $G = G(V, A)$ está formado por un conjunto no vacío de vértices V y un conjunto de aristas A . Cada arista es un par $\langle v, w \rangle$, donde $v, w \in V$. En ocasiones, las aristas tienen una tercera componente, denominada peso o costo [54].
- 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 pseudografo $G=(V, E, f)$ es una terna, donde $V \neq \emptyset$ es el conjunto de nodos, E es el conjunto de aristas y f una función definida como $f: E \rightarrow V \times V \cup \{u, v\}; u \in V, v \in V$.

1.7.1. Representaciones para grafos

Existen diferentes formas de representar grafos para el trabajo con los mismos. La estructura de datos usada depende de las características del grafo y el algoritmo destinado a su manipulación. Entre las estructuras más sencillas y usadas se encuentran las listas y las matrices [68], aunque frecuentemente se usa una combinación de ambas. Las listas son preferidas en grafos dispersos porque tienen un eficiente uso de la memoria. Por otro lado, las matrices proveen acceso rápido, pero pueden consumir grandes cantidades de memoria.

Estructura de lista.

- Lista de incidencia: Las aristas se representan con un vector de pares donde cada par representa una de las aristas. Estos pares son ordenados si el grafo es dirigido.
- Lista de adyacencia: Cada vértice tiene una lista de vértices los cuales son adyacentes a él. Esto causa redundancia en un grafo no dirigido, ya que A existe en la lista de adyacencia de B y viceversa. Las listas de adyacencia facilitan las búsquedas más rápidas, aunque poseen un costo extra de almacenamiento.

Estructuras matriciales

- Matriz de incidencia: El grafo está representado por una matriz de aristas (A), por vértices (V), donde $[A, V]$ contiene la información de la arista (1 - conectado, 0 - no conectado)
- Matriz de adyacencia: El grafo está representado por una matriz cuadrada M de tamaño n^2 , donde n es el número de vértices. Si hay una arista entre un vértice “ x ” y un vértice “ y ”, entonces el elemento $m_{x,y}$ es 1, de lo contrario, es 0.

1.7.2. Aplicaciones de los 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.

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 *.xpdll* 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 seudografo. En los capítulos posteriores se detalla con mayor profundidad lo expuesto anteriormente.

1.8. Conclusiones parciales

La calidad de los modelos de procesos de negocio es un aspecto importante del modelado. Una forma de evaluar la calidad de los modelos es a través de medidas de calidad asociadas a directrices prácticas de calidad. Sin embargo, no existe una herramienta que posibilite la evaluación de los modelos usando directrices prácticas como base. En este capítulo se analizaron varias herramientas de modelado de las cuales se obtuvieron archivos de salida que permiten la evaluación de los modelos de esta forma. Además se seleccionaron los grafos como estructura de datos para almacenar la información de los modelos con el objetivo de evaluar los mismos como se persigue en este trabajo.

Capítulo 2 Evaluación de modelos de proceso de negocio utilizando teoría de grafos

Este capítulo describen las directrices prácticas de morfología a evaluar en los modelos de proceso de negocio, así como el prototipo de software desarrollado exponiendo algunos diagramas creados durante las fases de análisis y diseño de la herramienta. Además se describe el funcionamiento de la misma.

2.1. Directrices prácticas

El estándar BPMN proporciona a las organizaciones la capacidad de comprensión de sus procesos internos de negocio en una notación gráfica y la capacidad de comunicar sus procedimientos de manera estándar. Además, define todos los elementos que conforman un modelo de procesos de negocio. Sin embargo, el uso del estándar no garantiza que los procesos se modelen de forma clara y comprensible. La forma en que los modeladores interpretan las condiciones de negocio y cómo definen su estructura, es crucial para asegurar que se entienden correctamente y que estén libres de errores [69]. Para ello, el estándar ofrece un conjunto de reglas que ayudan a los modeladores a comprender mejor los modelos de proceso de negocios y auxilian a que el modelo esté sintácticamente bien representado. Estas reglas están relacionadas con los principales elementos de modelado de los procesos de negocio. Existen reglas relacionadas con los eventos. Una de estas plantea que los eventos finales deben estar referenciados por uno o varios flujos de secuencia. Plantea además que se deben utilizar eventos finales por separado para identificar cuándo un proceso terminó con éxito y cuándo no. Otro conjunto de reglas se refieren a las compuertas y su forma de representación planteando que las mismas son graficadas como rombos trazados por una línea fina. De forma general estas reglas están orientadas a la forma de representación de los componentes en el diagrama, y a la manera específica en que deben aparecer los componentes en el modelo en cuanto a color, tamaño, líneas de trazado y texto que aparece en el mismo.

Estas reglas no están relacionadas con la calidad de los modelos de procesos de negocio en general, sino que están orientadas a la calidad sintáctica de los mismos. Es por eso que en [37] se propone un conjunto de directrices aplicables a los modelos de procesos de negocio que permiten obtener modelos con mayor calidad pragmática, empírica y semántica, logrando modelos lo más real posible del proceso en cuestión. Sin embargo al

no contar con una herramienta que permita evaluar la calidad de los modelos de procesos de negocio y con el objetivo de automatizar dicha evaluación se seleccionan un conjunto de medidas asociadas a las directrices para su implementación en la herramienta SAD_BPMN. En [37] se presenta un patrón que tiene como objetivo formular las directrices de forma homogénea. Para la automatización de la evaluación de las directrices en este trabajo se toman los siguientes elementos del patrón:

- Problema: características cuya ocurrencia denota la infracción de la directriz.
- Directriz unificada: la propuesta de este trabajo para la formulación de la directriz para BPMN de forma tal que resuma todas las directrices relacionadas encontradas en la literatura en una sola directriz.
- Medidas asociadas: listan las medidas estructurales para cuantificar la ocurrencia de los problemas en los modelos.

A continuación se presentan las directrices de tamaño siguiendo el formato mencionado anteriormente y explicando en algunos casos la interpretación de las mismas.

2.1.1 Directrices prácticas relacionadas con el tamaño

Algunos estudios ofrecen directrices relacionadas con el tamaño de los modelos debido a que los modelos de mayor tamaño son más difíciles de mantener, de comprender y tienen mayor probabilidad de que poseer errores [8, 33, 70].

Problema 1 (*Alto número de elementos*): El modelo contiene un alto número de elementos como compuertas de división/unión, actividades y eventos.

Directriz unificada: Evite los modelos con más de 31 elementos.

Medida asociada:

Definición:	Número de elementos [31]	
Medida:	$S_N(G) = N $	(1)
	Donde N es el número de nodos del pseudografo G del modelo de proceso.	
	Valor de umbral asociado [33]: 31	

Al representar un modelo de proceso de negocio este puede quedar demasiado grande por lo que la comprensión del mismo puede resultar una tarea compleja [71]. Esta directriz sugiere usar tan pocos elementos como sea posible por nivel de modelo de proceso, y en particular descomponer un modelo de proceso si este tiene más de 31 objetos de flujo. Esto permite reducir la presencia de errores y mejorar la calidad pragmática, empírica y sintáctica del modelo.

Problema 2 (*Alto número de eventos*): el modelo contiene más de siete eventos.

Directriz unificada: Evite los modelos con más de siete eventos.

Medida asociada:

Definición:	Número de eventos [31].	
Medida:	$S_E(G) = E $	(2)
	Donde E es el número de eventos del pseudografo G del modelo de proceso. Valor de umbral según [35]: 7	

Problema 3 (*Alto número de eventos de inicio*): El modelo contiene más de dos eventos de inicio.

Directriz unificada: No utilice más de dos eventos de inicio.

Medida asociada:

Definición:	Número de eventos de inicio [31].	
Medida:	$S_{E_S}(G) = E_S $	(3)
	Donde E_S es el número de eventos de inicio del pseudografo G del modelo de proceso. Valor de umbral según [33]: 2.5	

Problema 4 (*Ausencia de eventos de inicio*): Los modelos no poseen evento de inicio.

Directriz unificada: No omita el evento de inicio.

Medidas asociadas: Se asocia la medida (3).

Los problemas 3 y 4 pueden unificarse en una misma directriz práctica “no omita el evento de inicio y utilice dos como máximo”.

Problema 5 (*Alto número de eventos de fin*): El modelo contiene más de dos eventos de fin.

Directriz unificada: No utilice más de dos eventos de fin.

Medida asociada:

Definición:	Número de eventos de fin [31].	
Medida:	$S_{E_F}(G) = E_F $	(4)
	Donde E_F es el número de eventos de fin del pseudografo G del modelo de proceso. Valor de umbral según [33]: 2.5	

Problema 6 (*Ausencia de eventos de fin*): Los modelos no poseen eventos de fin.

Directriz unificada: No omita el evento de fin.

Medida asociada: Se asocia la medida (4).

Problema 7 (*Alto número de eventos intermedios*): Los modelos poseen un alto número de eventos intermedios.

Directriz unificada: Evite los modelos con un alto número de eventos intermedios.

Medida asociada:

Definición:	Número de eventos intermedios [31].	
Medida:	$S_{E_{int}}(G) = E_{int} $	(5)
	Donde $ E_{int} $ es el número de eventos intermedios del pseudografo G del modelo de proceso. No se cuenta con un valor de umbral para esta medida.	

Las directrices relacionadas con los problemas del tres al siete evalúan la presencia de los eventos en los modelos. Según [35] varias investigaciones han demostrado que los modelos con más de siete eventos son difíciles de comprender. Algunos autores recomiendan usar solo un evento de inicio y un evento fin, esto disminuye la probabilidad de errores. Aunque la presencia de dos eventos de tipo fin e inicio respectivamente es correcto. En la Figura 2.1 se muestra un modelo que contiene más de dos eventos de inicio y la solución de este problema de acuerdo a la directriz 3. En la Figura 2.2 se muestra lo mismo para el caso de un modelo con más de 2 eventos de fin. Estas directrices plantean que la presencia de eventos según la cantidad especificada anteriormente permite una mejor comprensión del modelo y disminuye la probabilidad de error.

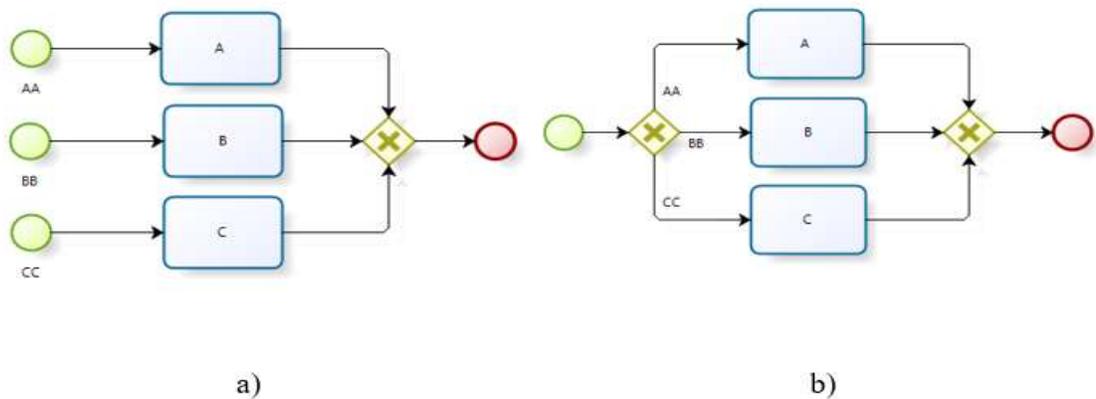


Figura 2.1: a) Proceso con más de dos evento inicio b) sustitución de varios eventos de inicio por uno.

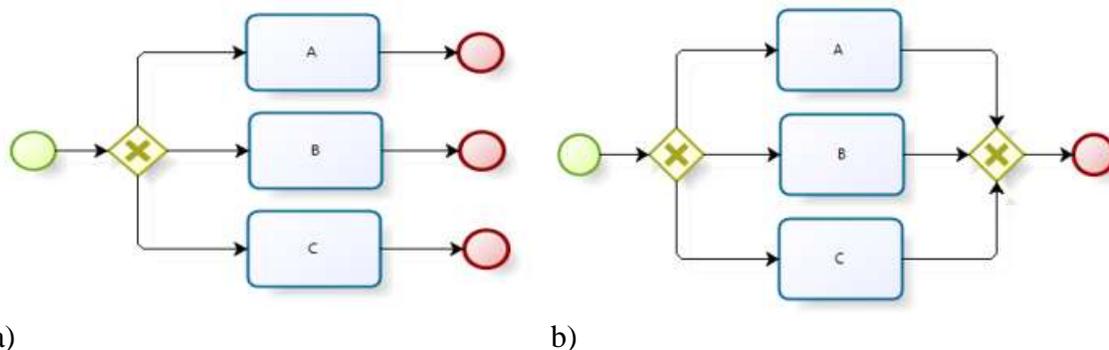


Figura 2.2: a) Proceso con más de dos eventos de fin b) sustitución de varios eventos de fin por uno.

Problema 8 (*Alto número de flujos de secuencia*): El modelo contiene más de 34 flujos de secuencia.

Directriz unificada: Evite los modelos con más de 34 flujos de secuencia.

Medida asociada:

Definición:	Número de flujos de secuencia [31].	
Medida:	$S_A(G) = A $	(6)
	Donde A es el número de flujos de secuencia del pseudografo G del modelo de proceso. Valor de umbral según [72]: 34	

El modelo de proceso puede contener un alto número de arcos. Si estos modelos se descomponen en subprocesos la comprensión del modelo se incrementa debido a que el número de arcos se divide entre el proceso y el subproceso mejorando así la calidad el modelo.

Problema 9 (*Alto número de compuertas*): El modelo contiene más de 12 compuertas de control de flujo.

Directriz unificada: Evite los modelos con más de 12 compuertas.

Medida asociada:

Definición:	Número de compuertas [31].	
Medida:	$S_C(G) = C $	(7)
	Donde C es el número de compuertas del pseudografo G del modelo de proceso. Valor de umbral según [34]: 12	

Según [73] el alto número de compuertas puede reducirse pues las compuertas que unen caminos paralelos e inmediatamente preceden un evento fin pueden eliminarse.

Problema 10 (*Alto número de actividades*): El modelo contiene más de 31 actividades.

Directriz unificada: Evite los modelos con más de 31 actividades.

Definición:	Número de actividades [31].	
Medida:	$S_F(G) = F $	(8)
	Donde F es el número de actividades del seudografo G del modelo de proceso. Valor de umbral según [28]: 31	

Un alto número de actividades incrementa la posibilidad de error en el modelo por lo que se recomienda descomponer el modelo como se plantea en el **Problema 1** con el objetivo de mejorar la calidad del mismo.

Para la aplicación de las directrices prácticas de tamaño se toma como ejemplo el modelo presentado en la Figura 2.3 el cual corresponde a un proceso de denuncia en las oficinas del MININT. En el Caso de estudio #1 se muestra el resultado de aplicar estas directrices al modelo. Se muestra el problema relacionado con las directrices cuyas medidas asociadas sobrepasan el umbral establecido para las mismas.

Tabla 2.1 Problemas de tamaño presentes en el modelo Caso estudio #1.

Problema	Medida asociada	Valor	Umbral
<i>Alto número de elementos</i>	Número de elementos: $S_N(G) = N $	41	31
<i>Alto número de eventos de inicio</i>	Número de eventos de inicio: $S_{E_S}(G) = E_S $	3	2.5
<i>Ausencia de eventos de fin</i>	Número de eventos de fin: $S_{E_E}(G) = E_E $	0	
<i>Alto número de flujos de secuencia</i>	Número de flujos de secuencia: $S_A(G) = A $	42	34
<i>Alto número de actividades</i>	Número de actividades: $S_F(G) = F $	33	31

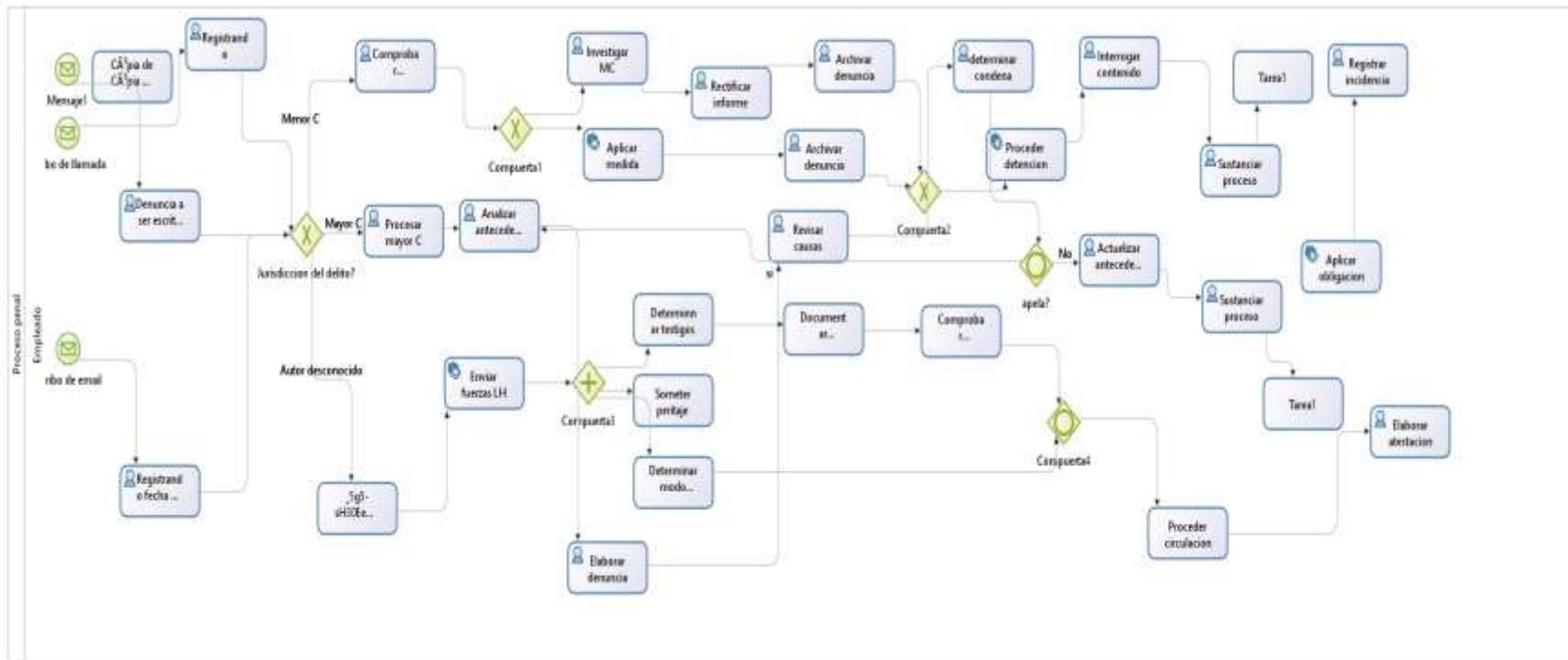


Figura 2.3: Modelo de proceso de negocio BPMN (Caso estudio #1).

2.1.2 Directrices prácticas relacionadas con la morfología

Existen algunas directrices prácticas cuyo objetivo es la mejora de la corrección semántica de los modelos, así como de la calidad pragmática y empírica.

Problema 11 (*Falta de estructuración en el modelo*): La suma de las incongruencias para cada tipo de compuerta está por encima de cuatro.

Directriz Unificada: Para cada nodo de decisión establecer un nodo de unión del mismo tipo.

Medida asociada:

Definición:	Incongruencia de las compuertas [31].
Medida:	$MM(G) = MM_{or} + MM_{xor} + MM_{and}$ (9)
	Donde $MM_l = \sum_{c \in S_l} d(c) - \sum_{c \in J_l} d(c) $ y l es el tipo de compuerta del pseudografo G del modelo de proceso. Valor de umbral asociado según [35]: 4

Los modelos estructurados pueden verse como fórmulas con paréntesis balanceados, es decir, cada paréntesis que abre tiene su correspondiente paréntesis que cierra del mismo tipo. Esta directriz enfatiza la importancia que posee la estructuración del modelo respecto a sus compuertas. Para mantener el modelo bien estructurado se aplica la topología propuesta por [74] donde a cada compuerta divisoria de un tipo se le hace corresponder otra compuerta de unión de igual tipo. En la *Figura 2.4* se muestra un modelo mal estructurado y otro como resultado de aplicar la directriz.

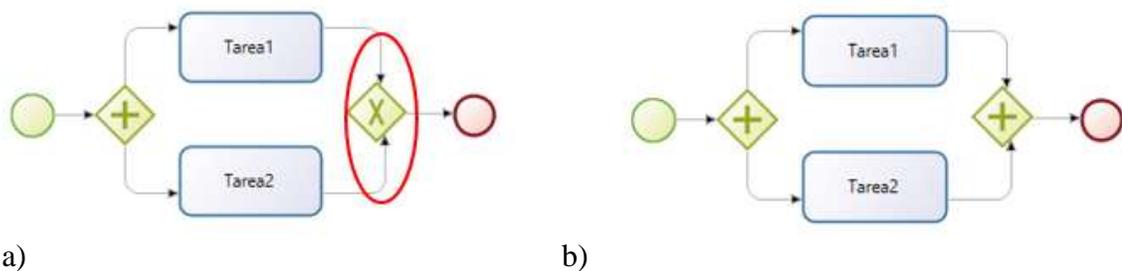


Figura 2.4 a) Modelo de proceso de negocio no estructurado y b) versión equivalente bien estructurada.

Problema 12 (*Alto número de flujos de secuencia por compuerta*): Las compuertas del modelo contienen un alto número de flujos de secuencia de entrada y/o salida.

Directriz unificada: No use más de cinco flujos de secuencia en cada compuerta.

Medidas asociadas:

Definición:	Grado promedio de las compuertas [31].	
Medida:	$\bar{d}_C(G) = \frac{1}{ C } \sum_{c \in C} d(c)$	(10)
	Valor de umbral según [34]: 3.83	
Definición:	Grado máximo de las compuertas [31].	
Medida:	$\hat{d}_C(G) = \max\{d(c) c \in C\}$	(11)
	En (10) y en (11), C es el conjunto de compuertas del pseudografo G del modelo de proceso, y $d(c)$ es el número de flujos de secuencia a los que se conecta; es decir, la suma del semigrado inicial (número de flujos de secuencia de salida) y del semigrado final (número de flujos de secuencia de entrada) de c . Valor de umbral para (11) según [34]: 5	

Problema 13 (*Alto número de flujos de secuencias por eventos*): Hay más de cuatro flujos de secuencia procedentes de eventos.

Directriz unificada: Utilice a lo sumo cuatro flujos de secuencia de salida en los eventos.

Medida asociada:

Definición:	Número de flujos de secuencia procedentes de eventos [20].	
Medida:	$NSFE = \sum_{e \in E} d_{out}(e)$	(12)
	Donde E es el conjunto de eventos del pseudografo G del modelo de proceso y $d_{out}(e)$ es el semigrado inicial (número de flujos de secuencia de salida) de e . Valor de umbral según [35]: 4	

Problema 14 (*Múltiples entradas y salidas en la misma compuerta*): El modelo contiene compuertas con más de un flujo de entrada y más de un flujo de salida a la vez, es decir, con comportamientos de unión y división.

Directriz unificada: No combine más de un flujo de entrada con más de un flujo de salida en una misma compuerta.

Medidas propuestas:

Definición:	Número de compuertas que combinan flujos de entrada y salida.	
Medida:	$S_{SJC}(G) = SJC $	(13)
Definición:	Razón de compuertas que combinan flujos de entrada y salida con la cantidad total de compuertas.	
Medida:	$SJC_C(G) = \frac{ SJC }{ C }$	(14)
	Donde $SJC \in S \cap J$, es decir, es el número de compuertas que combinan flujos de entrada y salida (comportamiento de unión y división) en el modelo de proceso y C es el número de compuertas del modelo del pseudografo G del modelo de proceso. Además, S es el conjunto de compuertas de división y J es el conjunto de compuertas de unión.	

Mientras mayor sea la cantidad de flujos de entrada o salida de un elemento del modelo más difícil será la comprensión del mismo. Esto puede resultar en problemas de sincronización, ejemplo de esto es cuando varios flujos se ejecutan en paralelo y se unen a través de una compuerta exclusiva [75]. También, pueden producir *deadlock* si se sincronizan utilizando una compuerta de unión paralela [33]. Se deben sincronizar con elementos de unión inclusivos lo cual contribuye a la estructuración del modelo, pero que a su vez son complejas de implementar de forma general [31, 76]. Las directrices 12, 13, 14 analizan estos problemas con el fin de mejorar la calidad del modelo.

Problema 15 (*Múltiples ciclos*): El modelo contiene múltiples ciclos.

Directriz unificada: Evite los ciclos en sus modelos de procesos.

Medidas asociadas:

Definición:	Ciclicidad [31].	
Medida:	$CYC_N(G) = \frac{ N_C }{ N }$	(15)
	Donde N_C es el número de elementos que pertenecen a ciclos, y N es el número total de elementos del modelo.	
	Valor de umbral según [33]: 0.005	
Definición:	Número de ciclos en el modelo [36].	
Medida:	$S_{Ciclos}(G) = Ciclos $	(16)
	Donde <i>Ciclos</i> es el número de ciclos en el modelo.	

Problema 16 (*Ciclos no estructurados*): El modelo contiene ciclos con múltiples puntos de salida, es decir, ciclos no estructurados.

Directriz unificada: Evite los ciclos no estructurados.

Medidas propuestas:

Definición:	Número de ciclos no estructurados.	
Medida:	$S_{CiclosNE}(G) = CiclosNE $	(17)
Definición:	Razón entre el número de ciclos no estructurados y el número de ciclos del modelo.	
Medida:	$CiclosNE_{Ciclos}(G) = \frac{ CiclosNE }{ Ciclos }$	(18)
	Donde <i>CiclosNE</i> es el número de ciclos que poseen más de un punto de salida y <i>Ciclos</i> es el número de ciclos del modelo.	

Problema 17 (*Ciclos mal formados*): La conexión de regreso del ciclo no es una compuerta de división de tipo *xor* o no regresa a una compuerta de unión de tipo *xor*.

Directriz unificada: Use una compuerta de división de tipo *xor* para retornar en un ciclo, y una compuerta de unión de tipo *xor* para el comienzo del ciclo.

Medidas propuestas:

Definición:	Número de ciclos mal formados.	
Medida:	$S_{CMF}(G) = CMF $	(19)
Definición:	Razón entre el número de ciclos mal formados y el número de ciclos.	
Medida:	$CMF_{Ciclos}(G) = \frac{ CMF }{ Ciclos }$	(20)
	Donde <i>CMF</i> es el número de ciclos con compuertas de entrada o salida que no son de tipo <i>xor</i> y <i>Ciclos</i> es el número de ciclos del modelo.	

Las directrices 15, 16 y 17 tratan el problema de los ciclos en el modelo. Los ciclos son fragmentos del modelo que pueden repetirse n-veces, convirtiendo a los modelos en diagramas inherentemente no estructurados [77] que son más difíciles de comprender y de mantener [33]. La falta de estructuración a su vez causa alta probabilidad de error, menor comprensión y menor capacidad de modificación de los mismos. Las directrices antes mencionadas proponen soluciones para la mejora del modelo y evitar la aparición de estos ciclos. De ser necesaria la construcción de ciclos se debe asegurar que se mantengan una correcta estructuración.

Problema 18 (*Alto nivel de paralelismo*): la suma de los grados de salida de las compuertas paralelas (de tipo *and*) y de las compuertas inclusivas (de tipo *or*) es mayor que 8.

Directriz unificada: Evite altos niveles de paralelismo.

Medida asociada:

Definición:	Suma de los grados de salida de las compuertas de división de tipo <i>and</i> o de tipo <i>or</i> menos uno [31].	
Medida:	$TS(G) = \sum_{c \in C_{or} \cup C_{and}} d_{out}(c) - 1$	(21)
	Donde $d_{out}(c)$ es el grado de salida de las compuertas del seudografo <i>G</i> del modelo de proceso. Valor de umbral según [33]: 7.5	

Problema 19 (*Paralelismo incorrecto*): Los caminos paralelos no se sincronizan ni alcanzan eventos finales.

Directriz unificada: Cada camino paralelo debe sincronizarse o alcanzar un evento de fin.

Medida propuesta:

Definición:	Número de caminos paralelos que no se sincronizan.	
Medida:	$S_{CNS}(G) = CNS $	(22)

Donde CNS es el número de caminos que surgen en compuertas de división paralelas que no se sincronizan y no alcanzan un evento de fin.

Problema 20 (*Alto nivel de profundidad*): El modelo contiene bloques estructurados profundamente anidados.

Directriz unificada: Evite anidamientos profundos de bloques estructurados.

Medidas asociadas:

Definición: Profundidad [31].

Medida: $\Lambda(G) = \max\{\lambda(n) | n \in N\}$ (23)

Donde N es el conjunto de elementos del pseudografo G del modelo de proceso, y $\lambda(n)$ es el mínimo entre $\lambda_{in}(n)$ y $\lambda_{out}(n)$. Para calcular $\lambda_{in}(n)$ se define un algoritmo donde para cada elemento n visitado, el nuevo valor $\lambda'_{in}(n)$ se actualiza tomando como base los valores de los elementos predecesores visitados y el valor actual de acuerdo a la siguiente regla:

$$\lambda'_{in}(n) = \begin{cases} \max\{\lambda_{in}(n), \lambda_{in}(pre) + 1\} & \text{if } pre \in S \wedge n \notin J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre)\} & \text{if } pre \in S \wedge n \in J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre)\} & \text{if } pre \notin S \wedge n \notin J \\ \max\{\lambda_{in}(n), \lambda_{in}(pre) - 1\} & \text{if } pre \notin S \wedge n \in J \end{cases}$$

De manera análoga se define $\lambda'_{out}(n)$. Valor de umbral según [35]: 1

Problema 21 (*Elementos duplicados*): El modelo contiene elementos o fragmentos duplicados que capturan la misma lógica de control de flujo.

Directriz unificada: Evite el uso de elementos y fragmentos duplicados.

Medidas propuestas:

Definición: Número de actividades y eventos duplicados.

Medida: $S_{AED}(G) = |AED|$ (24)

Donde AED es el número de actividades y eventos que tienen igual tipo e igual etiqueta y que aparecen repetidos en el modelo.

Definición: Número de fragmentos duplicados.

Medida: $S_{FD}(G) = |FD|$ (25)

Donde FD es el número de fragmentos duplicados que presentan la misma lógica de control de flujo y que están compuestos por elementos idénticos en el modelo.

Problema 22 (*Diámetro largo*): El modelo contiene un camino largo desde el inicio hasta el fin.

Directriz unificada: Mantenga el camino desde el evento de inicio hasta el evento final tan corto como sea posible.

Medida asociada:

Definición:	Diámetro [31].
Medida:	$diam(G) = \max\{\delta(u, v) u, v \in V\}$ (26)
	Donde $\delta(u, v)$ es la distancia geodésica [31] entre los elementos del pseudografo G del modelo de proceso. Umbral según [28]: 12.2

Problema 23 (*Alta diversidad de compuertas*): El modelo contiene compuertas de diversos tipos.

Directriz unificada: Minimice la diversidad de las compuertas utilizadas.

Medida asociada:

Definición:	Heterogeneidad de las compuertas [31].
Medida:	$GH(G) = - \sum_{l \in \{and, or, xor\}} p(l) * \log_3 p(l)$ (27)
	Donde $p(l) = C_l / C $, C_l es la cantidad de compuertas de un tipo determinado y C es la cantidad de compuertas del pseudografo G del modelo de proceso. Valor de umbral según [34]: 0.79

Problema 24 (*Uso de compuertas de división de tipo or*): El modelo contiene compuertas de división de tipo *or*.

Directriz unificada: Evite el uso de compuertas de división de tipo *or*.

Medida asociada:

Definición:	Número de compuertas de división de tipo <i>or</i> [31].
Medida:	$S_{or}(G) = S_{or} $ (28)
	Donde S_{or} es la cantidad de compuertas de división de tipo <i>or</i> del pseudografo G del modelo de proceso. Valor de umbral según [33]: 0.5

Problema 25 (*Alto grado de complejidad*): El modelo contiene alta complejidad.

Directriz unificada: Seleccione la alternativa más sencilla al modelar.

Medidas asociadas: Se asocian las medidas (9) a (28).

En ocasiones las soluciones propuestas a través de modelos de proceso de negocio son muy complejas debido al problema que se quiere representar. En ocasiones las personas añaden complejidad sin sentido al modelo, esto debe de evitarse pues perjudica la comprensión del modelo y aumenta la presencia de errores.

En la Tabla 2.2 se muestra el resultado de aplicar las directrices de morfología al modelo Caso Estudio #1 presentado en la Figura 2.3.

Tabla 2.2: Problemas de morfología presentes en el modelo Caso estudio #1.

Problema	Medida asociada	Valor	Umbral
<i>Alto número de flujos de secuencia por compuerta</i>	Grado promedio de las compuertas $\bar{d}_c(G) = \frac{1}{ C } \sum_{c \in C} d(c)$	3.66	3.83
	Grado máximo de las compuertas $\hat{d}_c(G) = \max\{d(c) c \in C\}$	6	5
<i>Múltiples entradas y salidas en la misma compuerta</i>	Número de compuertas que combinan flujos de entrada y salida: $S_{SJC}(G) = SJC $	1	0
	Razón de compuertas que combinan flujos de entrada y salida con la cantidad total de compuertas: $SJC_c(G) = \frac{ SJC }{ C }$	0.17	0
<i>Múltiples ciclos</i>	Número de ciclos en el modelo: $S_{Ciclos}(G) = Ciclos $	1	0
	Ciclicidad: $CYC_N(G) = \frac{ N_C }{ N }$	0.12	0.005
<i>Ciclos mal formados</i>	Número de ciclos mal formados: $S_{CMF}(G) = CMF $	1	0
	Razón entre el número de ciclos mal formados y el número de ciclos: $CMF_{Ciclos}(G) = \frac{ CMF }{ Ciclos }$	1	0
<i>Elementos duplicados</i>	Número de actividades y eventos duplicados: $S_{AED}(G) = AED $	3	0
	Número de fragmentos duplicados: $S_{FD}(G) = FD $	1	0
<i>Uso de compuertas de división de tipo OR</i>	Número de compuertas de división de tipo or: $S_{S_{or}}(G) = S_{or} $	1	0.5
<i>Alto grado de complejidad</i>	Se asocian las medidas de la (9) a la (29)		

2.2. Extracción de datos de los modelos

Para el almacenamiento de datos de los modelos de procesos de negocio es necesario utilizar una estructura de datos que permita generar información, donde se puedan realizar

distintos análisis con vistas a mejorar la calidad de los mismos. Como se menciona en el capítulo anterior, con el uso de pseudografos es posible expresar procesos y sistemas que requieren sincronización para su análisis de manera general y para la obtención de información del comportamiento dinámico de estos. En este trabajo se hace uso de pseudografos dirigidos que se implementan a través de listas de adyacencia. Esto permite una adecuada representación de los modelos de procesos de negocios además de poder almacenar información adicional de las aristas.

Para la automatización de la evaluación de los modelos de proceso de negocio BPMN tomando como base directrices de calidad, lo primero que se realiza es el pase de información del proceso a un pseudografo dirigido. En el mismo cada componente del proceso se corresponde con un vértice del pseudografo y las transiciones son las aristas que unen a su vez los vértices del mismo. Los nombres de los vértices se corresponden con el identificador de los componentes, de esta forma se logra que todos los componentes queden representados por vértices. También cada vértice almacena el tipo de componente que representa y el nombre del mismo para de esta forma poder diferenciarlos al evaluar.

Para ilustrar el proceso de extracción de datos de un modelo de proceso de negocio se utiliza el ejemplo que se muestra en la Figura 2.5. Este modelo describe el proceso de adquisición de libros en una universidad (Caso estudio #2). El modelo está compuesto por varias tareas, eventos y compuertas. La Figura 2.6 muestra el pseudografo correspondiente al modelo proporcionado en el Caso estudio #2.

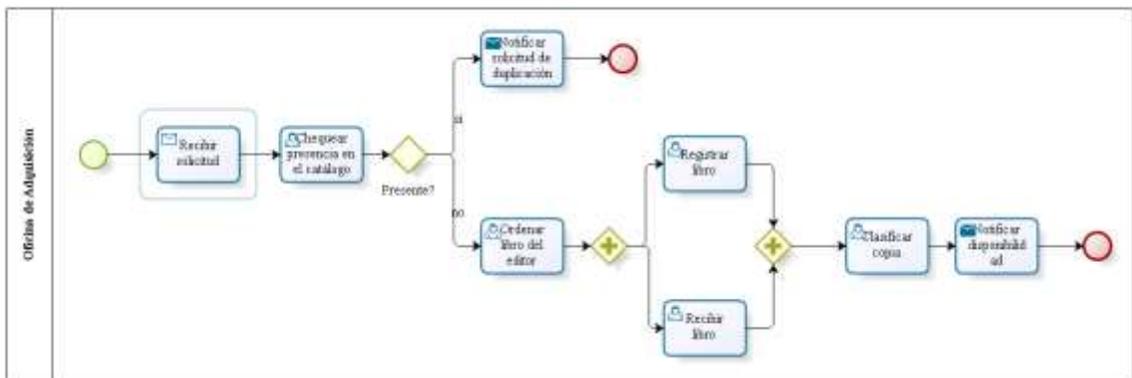


Figura 2.5 Proceso oficina de adquisición de libros (Caso estudio #2).

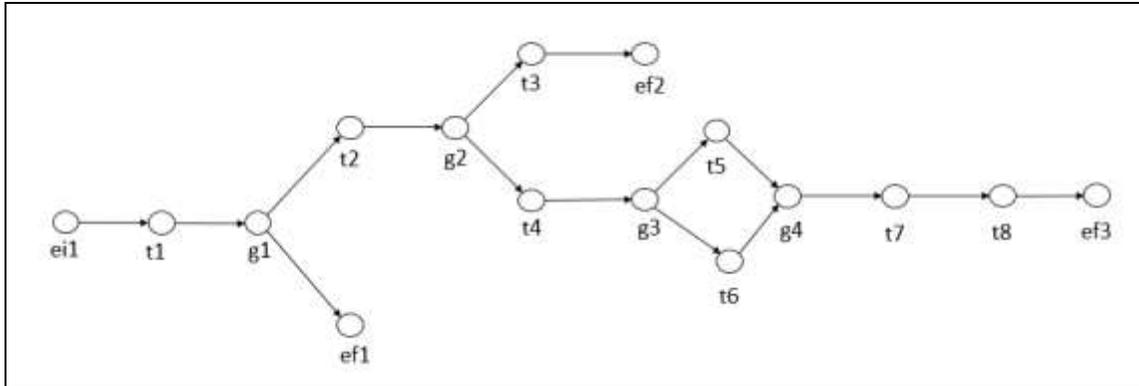


Figura 2.6: Seudografo resultante de la transformación del proceso.

Como se puede observar en la Figura 2.6 cada elemento del proceso se representa por un vértice en el seudografo y se garantiza así que toda la información del modelo se extraiga hacia el seudografo para su posterior evaluación. El seudografo resultante de la extracción de los datos del modelo a partir de los archivos de salida de las herramientas de modelado es común tanto para los modelos creados por la herramienta Bizagi como por la herramienta Bonita.

Para la extracción de los datos de los modelos se implementó el patrón de diseño Estrategia. Este patrón de diseño permite realizar un cambio de estrategia en tiempo de ejecución sustituyendo el objeto que se encargará de implementarla [78], por lo que el usuario puede elegir en tiempo de ejecución que tipo de archivo desea analizar, *.xpd* o *.bpmn* generados por las herramientas Bizagi y Bonita respectivamente. En la herramienta SAD_BPMN este patrón se implementa a través de la clase *AbstractModel* la cual es la interfaz común para todos los algoritmos del patrón y que es usada por la clase *SAD* para invocar la estrategia correcta. Las clases *LoadBizagi* y *LoadBonita* son las encargadas de implementar el algoritmo utilizando la interfaz definida por la estrategia. Por último la clase *SAD_BPMN* es la encargada de elegir cuál estrategia aplicar para extraer los datos del modelo. La Figura 2.7 muestra el diagrama de clases correspondiente al patrón Estrategia implementado en la herramienta SAD_BPMN.

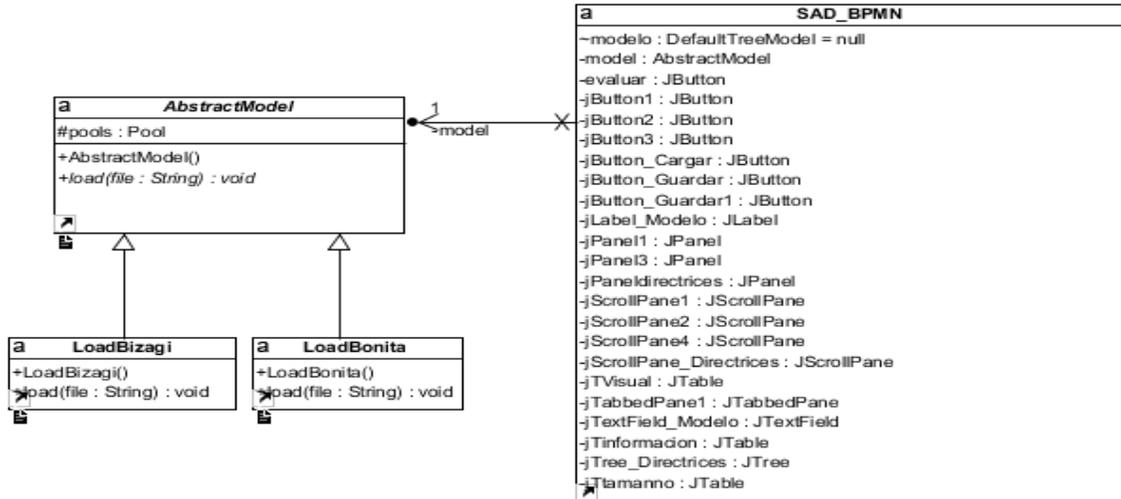


Figura 2.7: Patrón Estrategia.

2.3. Otros aspectos de implementación de las directrices prácticas de complejidad

Los algoritmos pueden ser expresados de muchas maneras, a través del lenguaje natural, pseudocódigo, diagramas de flujo y lenguajes de programación, entre otros [79]. Para la implementación de las directrices de prácticas de morfología se diseñaron algoritmos con el objetivo de calcular las medidas propuestas asociadas con cada directriz.

Los algoritmos propuestos para el cálculo de las medidas asociadas a las directrices de tamaño presentan una complejidad lineal. Esto se debe en parte a la representación del pseudografo que es a través de listas de adyacencia como se menciona en el Capítulo 1, y a que las medidas asociadas a las directrices de tamaño tienen como funcionalidad contar elementos en el modelo. A continuación se presenta el algoritmo asociado al problema *Alto número de eventos*, el cual se evalúa a través de la medida $(S_E(G))$.

Algoritmo 1: medida_SE(Pool p), calcula la cantidad de eventos del proceso, recibe como parámetro el pool, el cual contiene el seudografo con toda la información del modelo. Este algoritmo devuelve un entero que indica la cantidad de eventos del modelo.

Los pasos del algoritmo son:

```

Paso 1.  c->0
Paso 2.  Grafo g->p.getGrafo()
Paso 3.  Para i<-0 hasta cantElementosGrafo() incrementar i
           hacer
               si(elementoGrafoi es evento) entonces
                   incrementar c
               fin si
           fin para
    
```

A continuación se detallan los pasos del algoritmo.

```

Paso 1.  Se inicializa la variable c en cero. La variable c es la
           encargada de acumular la cantidad de eventos.
Paso 2.  Se almacena en g el seudografo asociado al proceso para su
           análisis.
Paso 3.  Se recorre la tabla de vértices del seudografo, preguntando
           si cada vértice es un evento. Se incrementa la variable c de ser
           cierta la interrogante.
    
```

Este algoritmo presenta una complejidad lineal debido a que solo es necesario recorrer una vez la tabla de vértices del seudografo para chequear los elementos que son eventos. Es decir, solo se ejecutarán n comparaciones, por lo que su complejidad es un $O(n)$. De igual manera pasa con los demás algoritmos asociados a las directrices de tamaño.

Alguno de los algoritmos diseñados para las medidas asociadas a las directrices de morfología presentan una complejidad mayor. Ejemplo de esto es el algoritmo propuesto para la detección de elementos duplicados utilizando la medida *Número de actividades y eventos duplicados*. A continuación se plantea el algoritmo:

Algoritmo 2: medida_SAED(Pool p), detecta la cantidad de elementos duplicados que existen en el modelo. Recibe como parámetro un pool que contiene el pseudografo con los elementos a analizar. Devuelve una lista con los elementos duplicados.

Los pasos del algoritmo son:

```

Paso 1.  c->0
Paso 2.  Grafo g->p.getGrafo()
Paso 3.  para i->0 hasta cantElementosGrafo() incrementar i
           hacer
           si(elementoGrafoi no ha sido visitado) entonces
             marcar visitado i
             para j->i+1 hasta cantElemenGrafo(), incrementar j
             hacer
               si(elementoGrafoj no ha sido visitado) entonces
                 si(elementoGrafoi es igual elementoGrafoj)
                   entonces
                     incrementar c
                     marcar visitado j
                 fin si
             fin para
           fin para
           fin para
           fin para

```

A continuación se detallan los pasos del algoritmo.

Paso 1. Se inicializa la variable **c** en cero. La variable **c** es la encargada de acumular la cantidad de eventos.

Paso 2. Se almacena en **g** el grafo asociado al proceso para su análisis.

Paso 3. Se selecciona un vértice del grafo. Si este no ha sido visitado, se marca como tal y se compara con el resto de los vértices. Se marcan los vértices que tengan igual nombre e igual tipo. De coincidir alguno se incrementa **c** en 1.

El **Algoritmo 2** presenta en el peor de los casos una complejidad cuadrática $O(n^2)$. Debido a que realiza una cantidad de **nXn** comparaciones. Esto se debe a que puede ocurrir que en el modelo no existan elementos duplicados y se analicen todos los elementos del pseudografo. Algunas directrices involucran medidas que para su cálculo es necesario hacer un recorrido completo del pseudografo. Cualquier algoritmo de recorrido de pseudografo consiste en visitar un nodo del pseudografo y luego ir visitando los nodos conectados a él. Este principio se aplica recursivamente comenzando desde un nodo inicial cualquiera del pseudografo. Dos algoritmos de recorrido de pseudografo son el recorrido en profundidad y a lo ancho. Ambos algoritmos han sido estudiados con anterioridad y se les conoce su orden de complejidad en tiempo y todos los beneficios de aplicarlos. Un grupo de directrices a evaluar en los modelos de procesos de negocio analizan la presencia de ciclos

en el diagrama, su estructuración y la cantidad de elementos que los conforman. En [80] se propone un algoritmo donde se detectan ciclos en un seudografo el cual tiene una complejidad de $O(n^3m^2)$. Para la detección de ciclos en el modelo se modifica el algoritmo de recorrido en profundidad de un grafo DFS (*DepthFirst Search*) almacenando los caminos con ciclos para su posterior análisis. La medida *Ciclicidad* calcula la razón entre la cantidad de elementos que pertenecen a ciclos entre la cantidad de ciclos. A continuación se presenta el algoritmo asociado al cálculo de dicha medida.

Algoritmo 3: medida_CYC(Pool p), calcula la razón entre la cantidad de elementos que pertenecen a ciclos entre la cantidad de ciclos. Recibe como parámetro un pool que contiene el seudografo con los elementos a analizar. Devuelve un valor real.

Los pasos del algoritmo son:

```

Paso 1. Grafo g->p.getGrafo()
Paso 2. ArrayList<String> elementos <- new ArrayList<String>()
Paso 3. ArrayList<String> caminos <- caminos_con_ciclo(g)
Paso 4. para k->0 hasta cantCiclos() incrementar k
          hacer
            para i->0 hasta cantCaminos()-1 incrementar i
              hacer
                para j->1+1 hasta cantCaminos() incrementar j
                  hacer
                    si(caminos[i] es igual a caminos[j]) entonces
                      adicionarElementos(elementos);
                    fin si
                fin para
            fin para
          fin para

```

Paso 5. Calcular razón

A continuación se detallan los pasos del algoritmo.

Paso 1. Se almacena en g el seudografo asociado al proceso para su análisis.

Paso 2. Se inicializa la variable elementos. La variable elementos es la encargada de acumular los elementos de ciclos.

Paso 3. La variable caminos recibe los caminos con ciclos del seudografo.

Paso 4. Se recorren los caminos con ciclos, se selecciona el un vértice del camino, se compara con los demás hasta que coincida. Se almacenan los vértices comprendidos entre el primer vértice y la posición donde aparece de nuevo en el camino.

Paso 5. Se calcula la razón entre la cantidad de elementos en ciclos entre la cantidad de ciclos.

El **Algoritmo 3** presenta una complejidad de $O(n^3)$ debido a que es necesario comparar los elementos de los caminos con ciclos para detectar cuales pertenecen a los ciclos en el seudografo.

2.4. Requisitos funcionales y no funcionales del sistema

Los requisitos para una herramienta de software se utilizan para comunicar de forma precisa las funciones que debe proveer la misma. Estos determinan lo que hará la herramienta y definen las restricciones de su implementación y operación [81]. Por esto la ingeniería de requisitos es una actividad esencialmente de interacción con los interesados en el software [82]. Los requisitos de software se clasifican en funcionales y no funcionales.

Los requisitos funcionales son declaraciones de los servicios que el software debe proveer mientras que los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas [81]. El objetivo principal de la herramienta es la evaluación de modelos de procesos de negocio tomando como base directrices de calidad con el fin de mejorar la calidad de los mismos. Para ello se definen los siguientes requisitos funcionales y no funcionales:

Requisitos funcionales

RF1: Cargar modelo de proceso de negocio diseñado por la herramienta de modelado Bizagi o Bonita

RF2: Calcular medidas de calidad

RF4: Evaluar modelo de proceso de negocio

RF3: Guardar directrices en un archivo con formato *.xls*

Requisitos no funcionales

Tabla 2.3: Requisitos no funcionales del sistema.

RNF#	Nombre	Descripción
RNF1	Interfaz	La interfaz debe ser sencilla y fácil de usar
RNF2	Hardware	Para garantizar el correcto funcionamiento de la aplicación, se necesita una computadora 128 MB de memoria RAM, 7 MB de espacio en disco para el código fuente, 1 MB de

		espacio en disco para el ejecutable de la instalación, 1 MB extra para guardar los ficheros
RNF3	Software	Para el correcto funcionamiento de la herramienta se necesita tener instalado la versión 1.7 o superior de JRE. Puede funcionar en todas las versiones de sistema operativo Windows.

2.5. Diagrama de casos de uso

El diagrama de casos de uso del sistema es un artefacto gráfico que contiene todos los requisitos funcionales que deberá cumplir el sistema y una parte estimable de los requisitos no funcionales que deben lograrse como propiedades. En él se representan los actores, los casos de uso del sistema y las asociaciones que existen entre actores y casos de uso [83]. En la Figura 2.8 se muestra el diagrama de casos de uso de la herramienta propuesta.

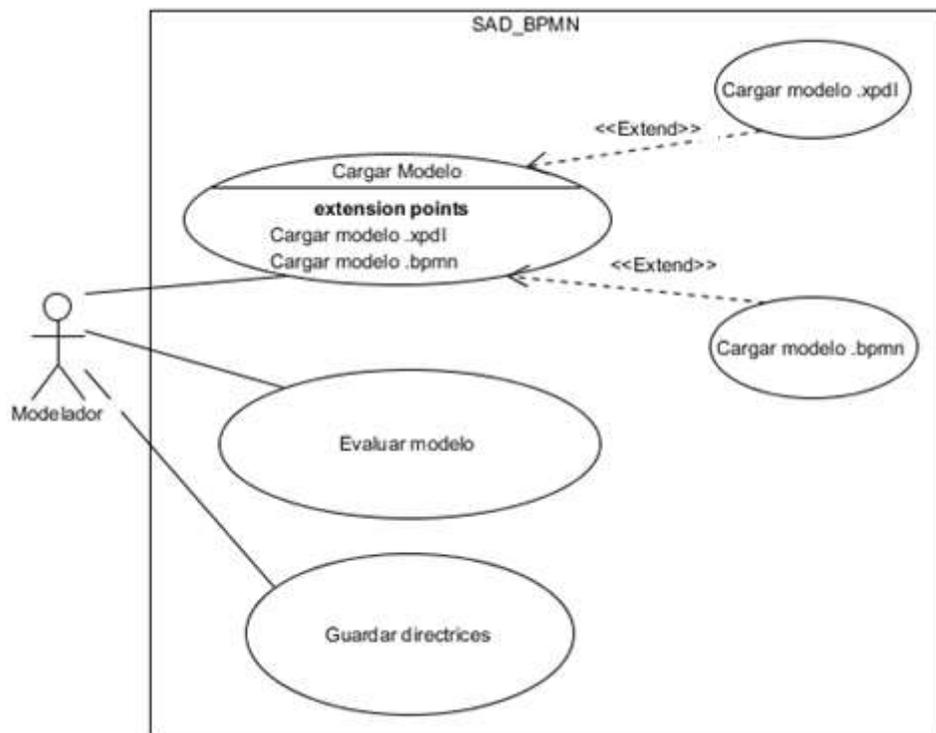


Figura 2.8 Diagrama de casos de uso de la herramienta.

El sistema presenta un único actor el que interactúa con los tres casos de usos representados en la Figura 2.8, este es el encargado de realizar la carga del modelo, evaluar el mismo a través de las directrices prácticas de calidad y exportar los resultados del proceso hacia un fichero EXCEL. La Tabla 2.4 muestra la descripción de los casos de uso de la herramienta.

Tabla 2.4 Resumen de los casos de uso de la herramienta.

Casos de uso de la herramienta	Descripción
Cargar modelo	El caso de uso se inicia cuando el modelador desea cargar un modelo de proceso de negocio contenido en un archivo creado en las herramientas Bizagi y Bonita con extensión <i>.xpd</i> l o <i>.bpmn</i> con el fin de evaluar el modelo a través de las directrices.
Evaluar modelo	El modelador elige en la herramienta las directrices que desea aplicar a su modelo, selecciona la opción evaluar modelo y obtiene sugerencias de posibles cambios para mejorar la calidad del mismo.
Guardar consejos	El caso de uso comienza cuando el modelador desea guardar los resultados obtenidos luego de aplicar las directrices de calidad, selecciona la opción en el sistema y guarda la información en un documento Excel.

Los casos de uso más significativos que presenta la herramienta SAD_BPMN son: cargar modelo y evaluar modelo.

2.6. Diagramas de secuencia

Los diagramas de secuencia forman una parte esencial de la investigación para interpretar un software. Estos muestran gráficamente los eventos que fluyen de los actores de la herramienta [84]. En la Figura 2.9 se representa el diagrama de secuencia asociado al caso de uso *Cargar modelo*. El diagrama de secuencia comienza con la selección del modelador de sobre qué tipo de archivo desea cargar en la herramienta para su posterior evaluación. En dependencia del tipo de archivo a analizar se procesan los datos de modelo.

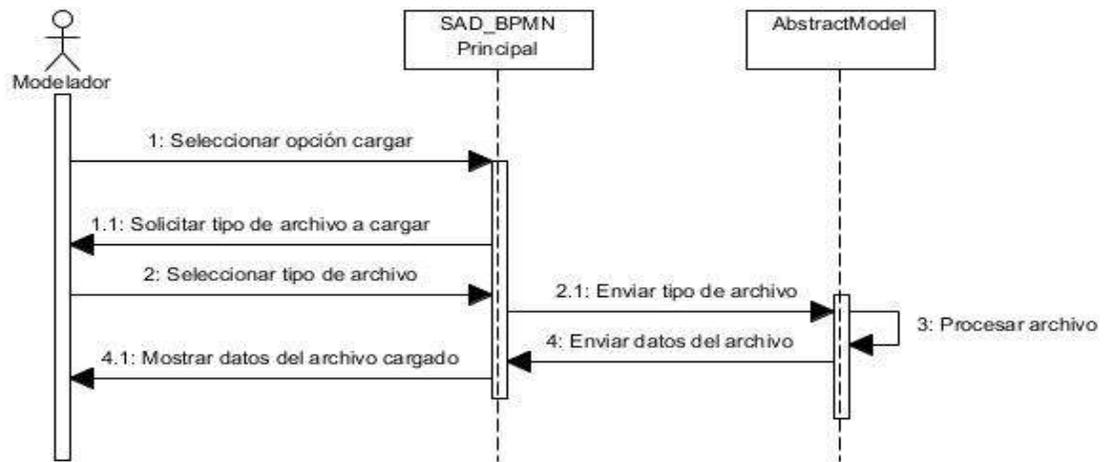


Figura 2.9: Diagrama de secuencia Cargar modelo.

2.7. Diagrama de paquetes y clases del diseño

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Se puede decir que existe dependencia entre varios paquetes cuando un elemento de un paquete requiere de otro elemento que pertenece a un paquete distinto. En la Figura 2.10 se muestra el diagrama de paquetes correspondiente a la herramienta propuesta en este trabajo, presentando las relaciones que existen entre los diferentes paquetes.

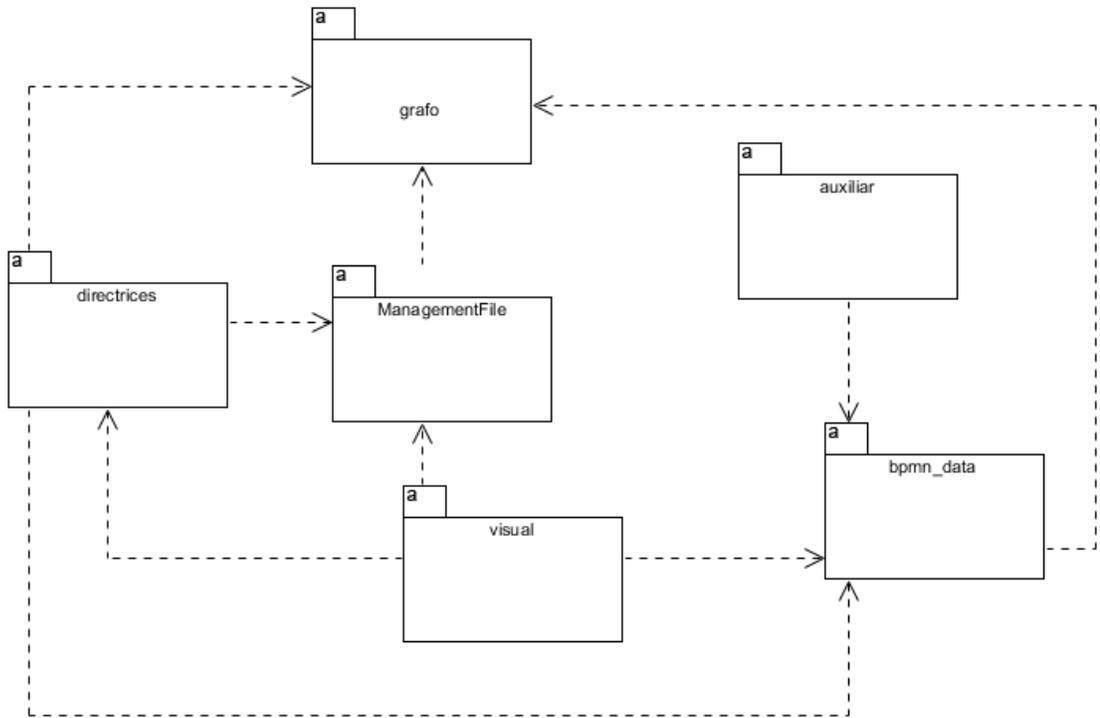


Figura 2.10: Diagrama de paquetes de SAD_BPMN.

Por otra parte, el diagrama de clases de un diseño incluye información como: clases, asociaciones, atributos, operaciones, constantes, métodos, navegabilidad y dependencias. Según [83], estos diagramas describen la realización física de los casos de uso, sirven de abstracción en la implementación del sistema y son utilizados como una entrada fundamental de las actividades de implementación. A continuación se describe una selección de los paquetes que conforman la herramienta SAD_BPMN.

El paquete *directrices* contiene las clases *Tamanno* y *XMLStructure*. La clase *Tamanno* es la encargada de implementar las directrices de calidad además de calcular todas las mediadas asociadas a estas directrices. Por otra parte la clase *XMLStructure* brinda un conjunto de métodos necesarios para la extracción de los datos de modelo. En la Figura 2.11 se muestra el paquete *directrices* con el diagrama de clases asociado al mismo.

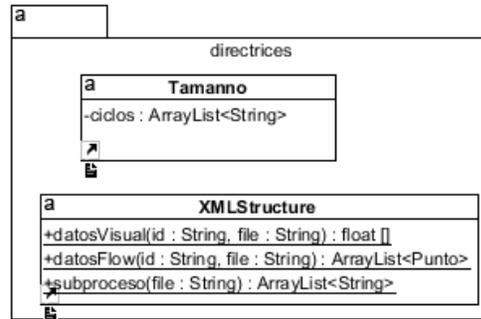


Figura 2.11: Diagrama del paquete *Directrices*.

En el paquete *bpmn_data* se implementa un conjunto de clases encargadas de almacenar la información de todos los componentes gráficos del modelo. Se diseña la clase jerárquica *Elements* donde se definen los elementos comunes de los componentes del diagrama como identificador, nombre y puntos. Esta clase es heredada por las clases *Pool*, *Gateways*, *Event*, *Task* y *Lane*. Las mismas son las encargadas de almacenar la información de los componentes del modelo respectivamente. En este paquete también se implementa una clase *Punto* que brinda un formato para guardar las coordenadas de los componentes del proceso. Además, se implementó la clase *Constant* la cual contiene un conjunto de constantes que hacen referencia a los componentes del estándar BPMN. El objetivo de esta clase es unificar bajo una misma constante los componentes de los de BPMN debido a que las herramientas de modelado Bizagi y Bonita expresan de forma distinta en los ficheros generados los componentes del modelo. En la Figura 2.12 se puede observar el diseño de las clases pertenecientes al paquete *bpmn_data*.

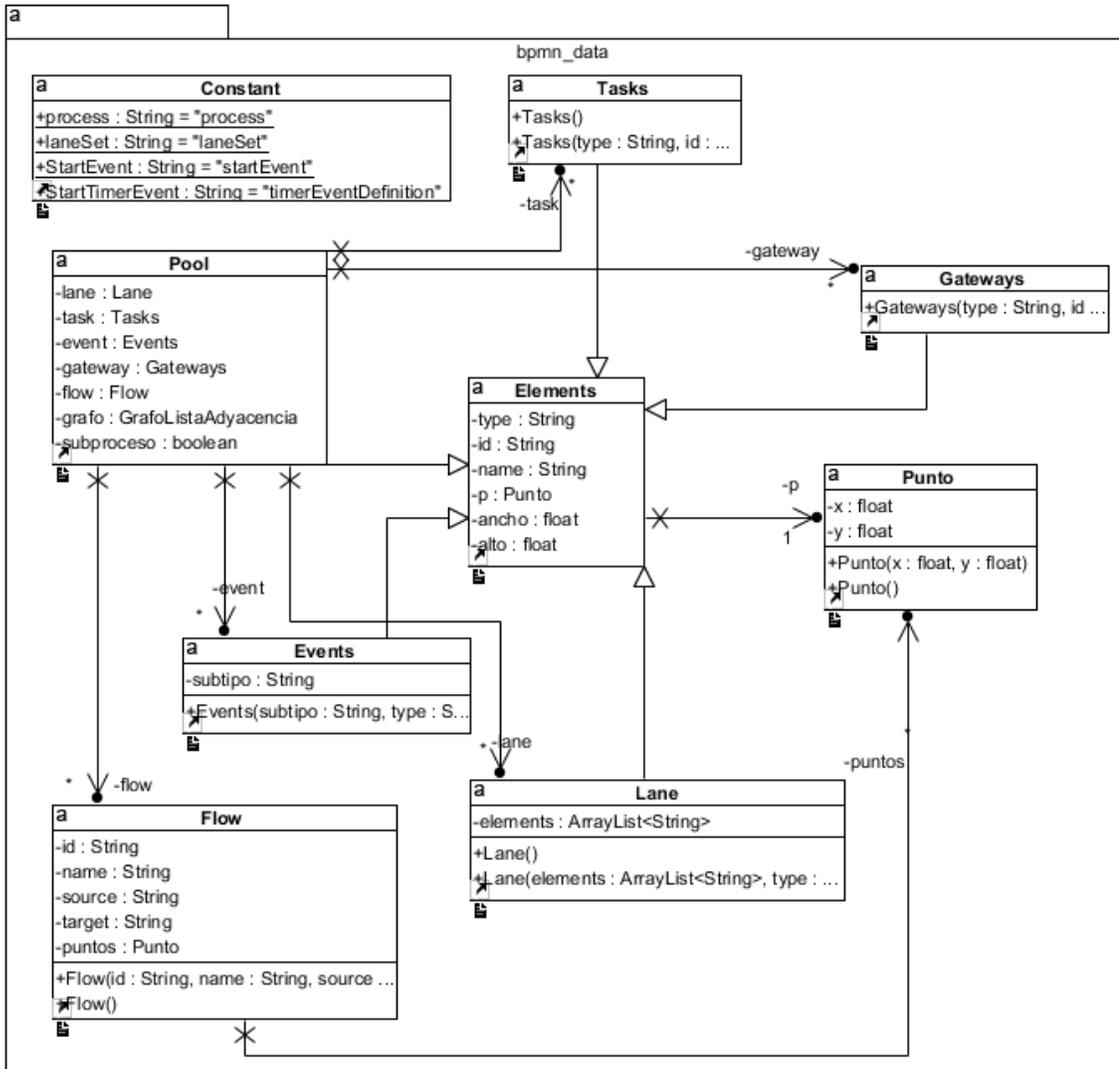


Figura 2.12 : Diseño de clases del paquete *bpmn_data*.

ManagementFile es el paquete donde se implementan las clases encargadas de la extracción de los datos del modelo como se explica en epígrafe 2.2. Esto se logra haciendo uso del patrón de diseño Estrategia como se explica en dicho epígrafe. El paquete pseudografo contiene las clases Vértice, Arista y GrafoListaAdyacencia. Como se dijo anteriormente, la implementación del pseudografo se realiza a través de listas de adyacencias. Hay además otras clases que son necesarias para completar dicho paquete. Existe un paquete *visual* donde se implementan las clases relacionadas con el diseño visual de la herramienta. Además se diseña una estructura de datos *JTree* personalizada a través de las clases *Render*, *Plantilla* y *Editor* con el objetivo de construir un árbol con las directrices seleccionadas en este trabajo permitiendo al usuario un mejor uso de las

directrices. También se implementa la clase SAD la cual es la encargada de controlar la ejecución de las directrices de calidad sobre el modelo. A continuación se muestran los paquetes grafo y visual en las Figura 2.13 y Figura 2.14 respectivamente.

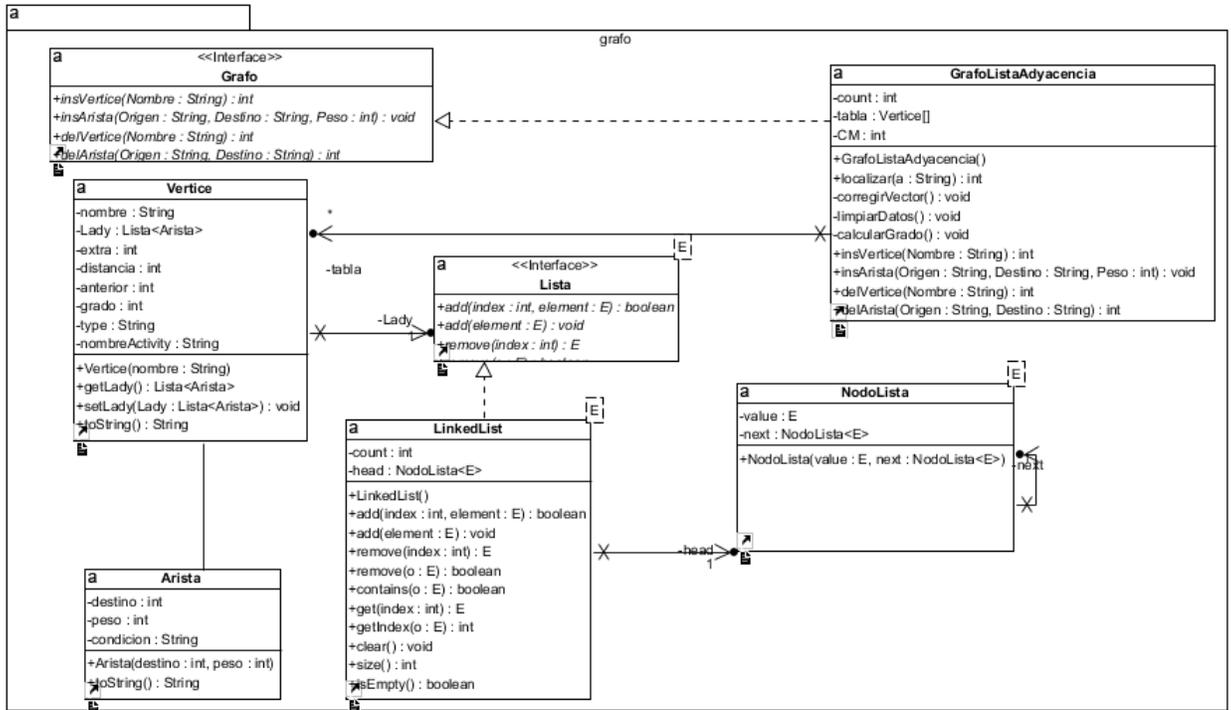


Figura 2.13: Paquete Grafo.

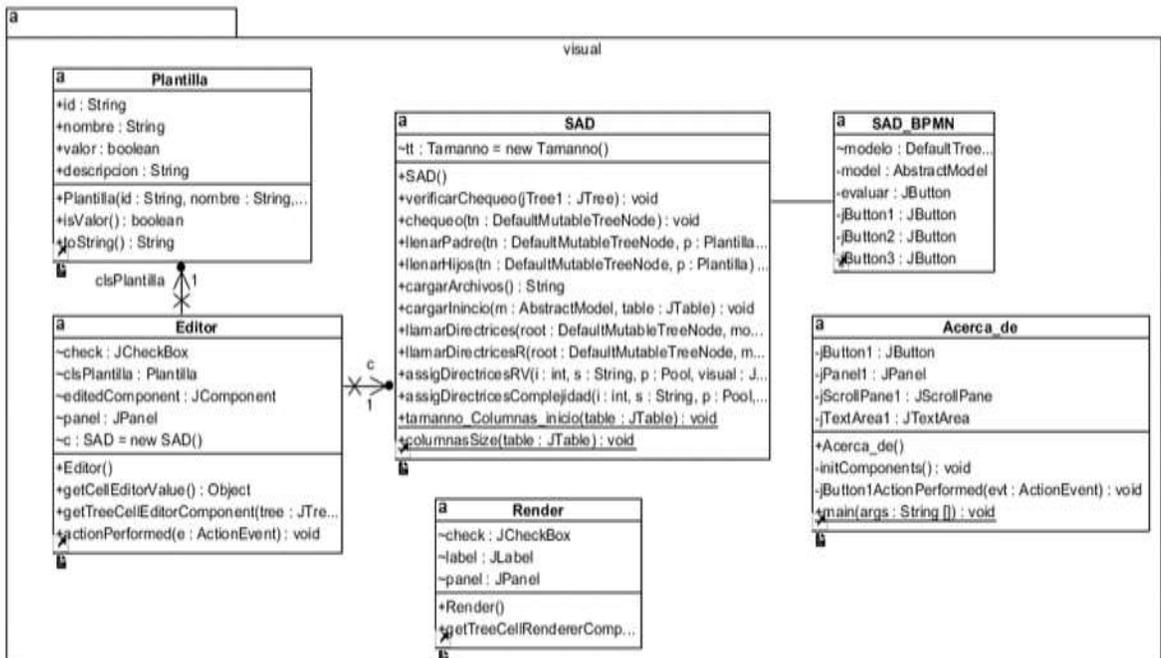


Figura 2.14: Paquete visual.

2.8. Modelo de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. El diagrama de componentes asociado a SAD_BPMN se muestra a continuación en la Figura 2.15

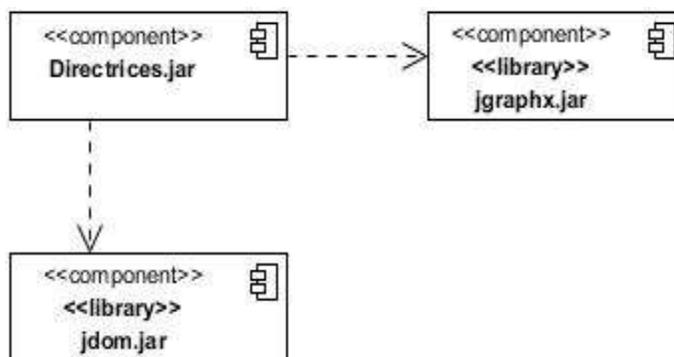


Figura 2.15 Diagrama de componentes.

2.9. Conclusiones parciales

En este capítulo se presentaron las directrices prácticas que se usan como referente para la evaluación de modelos de procesos de negocio BPMN. Se analizaron los archivos de salida de las herramientas de modelado de procesos de negocio Bizagi y Bonita para la extracción de los datos del modelo haciendo uso del patrón de diseño Estrategia. Se almacenó la información de los modelos de procesos de negocio en la estructura de datos grafo para su posterior evaluación a través de las directrices de calidad haciendo uso de las medidas asociadas a las mismas. Se mostró una perspectiva de la herramienta SAD_BPMN haciendo uso de diagramas de clase, diagramas de secuencia y diagramas de componentes.

Capítulo 3 Herramienta para la evaluación de modelos de procesos de negocio SAD_BPMN

El software SAD_BPMN proporciona un sistema para la evaluación de modelos de procesos de negocio tomando como base directrices prácticas de calidad. Permite al usuario obtener una información sobre los problemas relacionados con las directrices. Para ello evalúa un conjunto de medidas de calidad asociadas a las directrices en el modelo BPMN, contrasta los valores obtenidos con umbrales y de acuerdo a este resultado muestra las directrices prácticas como una ayuda a los modeladores de procesos de negocio. Para ello interpreta los archivos de salida de las herramientas Bonita y Bizagi con extensiones *.bpmn* y *.xpd* respectivamente. Estos archivos de salida se basan en *xml*, por lo que el trabajo en la herramienta se dirige fundamentalmente a la interpretación de este tipo de archivo.

3.1. Descripción de la herramienta

La interfaz visual de la aplicación, como se muestra en la Figura 3.1, se puede dividir en cuatro partes fundamentales:

1. Carga del modelo: es donde se selecciona el archivo que contiene el modelo de proceso de negocio que se pretende evaluar.
2. Contenedor de directrices a evaluar en el modelo: incluye las directrices de complejidad general que pueden sugerirse a partir de la evaluación de los modelos de procesos de negocio, separadas en dos grupos fundamentales, las directrices de tamaño y las directrices de morfología.
3. Panel de muestra de resultados de la evaluación: sugiere las directrices prácticas que el modelador debe aplicar a sus modelos de procesos de negocio evaluados para incrementar su calidad.
4. Sección de exportación de los resultados hacia un fichero Excel y visualización en un fichero Word de la descripción detallada de las directrices prácticas como ayuda al modelador.

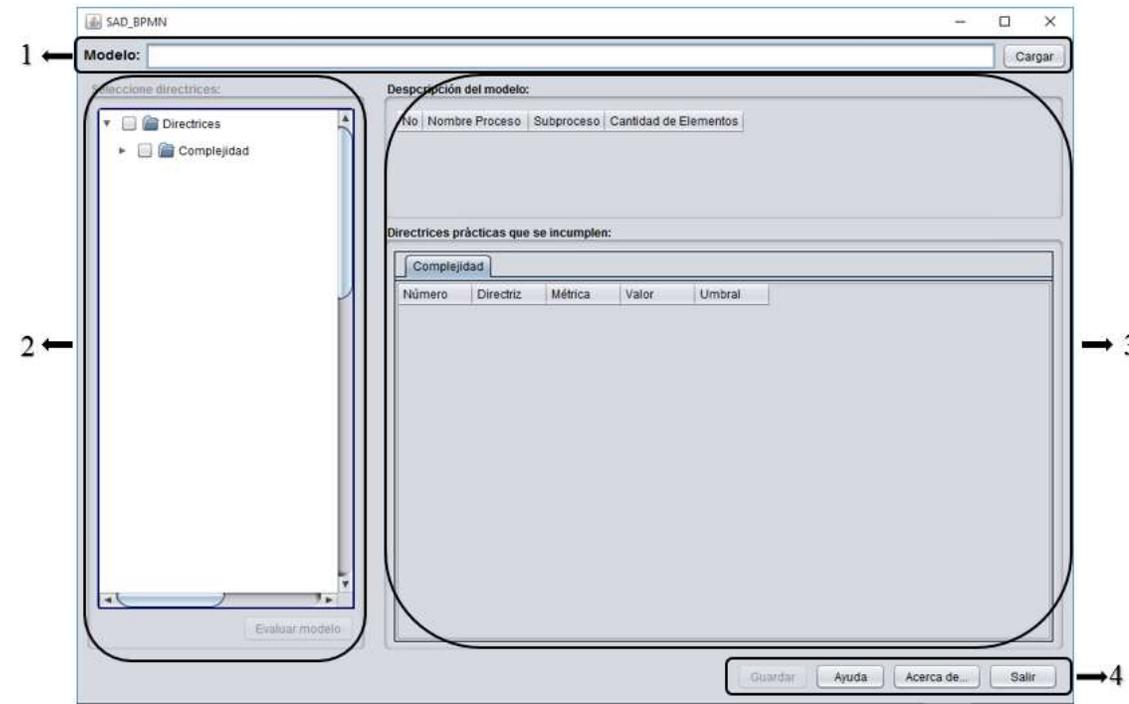


Figura 3.1: Interfaz gráfica de SAD_BPMN.

Como muestra la Figura 3.1, además de la selección de las directrices a aplicar, lo primero que se debe realizar en el software es el modelo a ser procesado. Como puede observarse, cuando aún no se ha cargado un modelo el botón *Evaluar modelo* no se encuentra habilitado. Este se habilita cuando se realice la carga del archivo que contiene el modelo. Para la carga del archivo el usuario puede escoger el tipo de archivo que desea cargar, que puede ser de tipo *.bpmn* o *.xpdl*, como se muestra en la Figura 3.2.

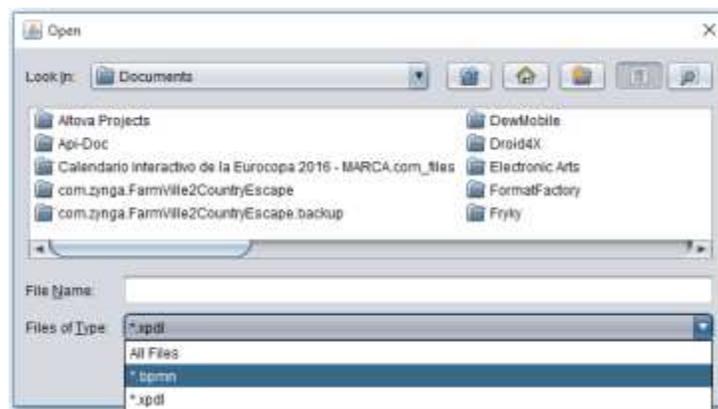


Figura 3.2: Selección del fichero a evaluar por la herramienta.

Cuando el usuario selecciona el modelo que desea procesar, pasa a seleccionar las directrices que desea aplicar al mismo. Aquí las directrices se encuentran clasificadas en

directrices de complejidad, las cuales incluyen las directrices de tamaño y las directrices de morfología. La Figura 3.3 muestra la ventana principal luego de seleccionar un modelo de proceso de negocio. Como puede observarse, la sección “Descripción del modelo” muestra información preliminar sobre el mismo, como su nombre lo indica. Aquí se le asigna un número a cada proceso que será evaluado y se muestra si es un subproceso o no, además de la cantidad total de elementos que posee.

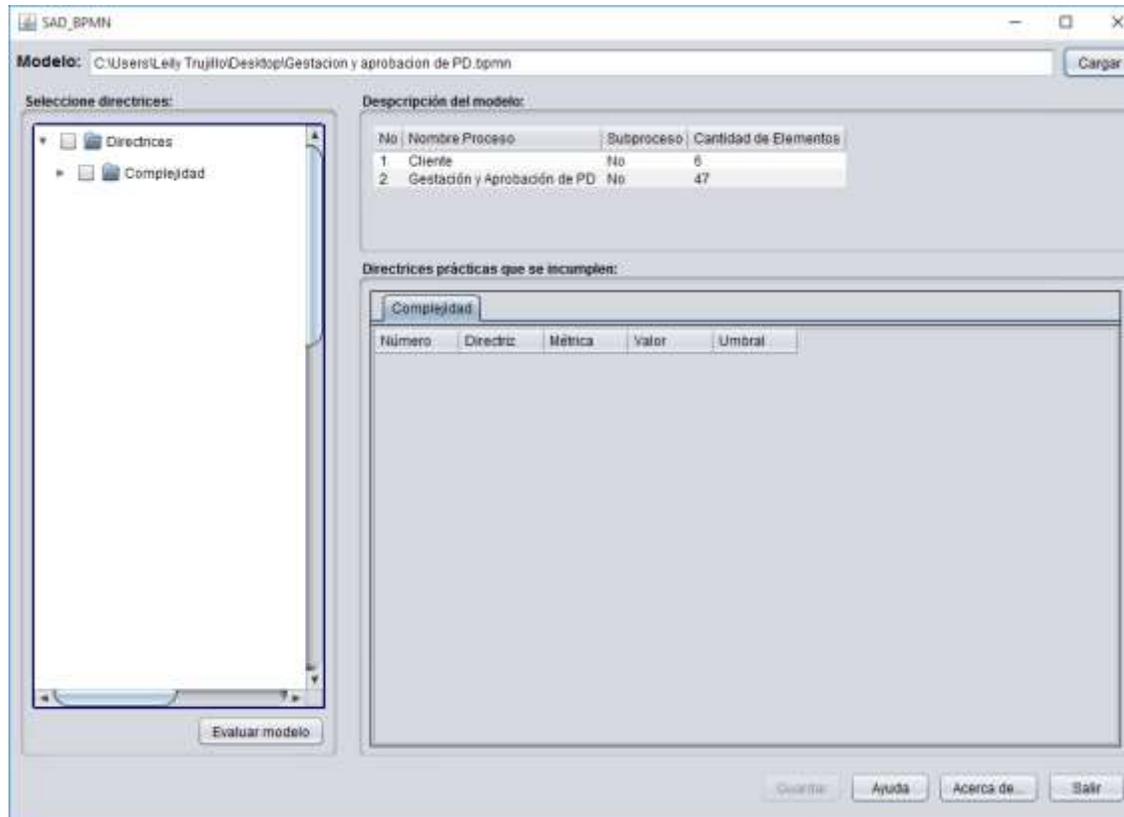


Figura 3.3: Ventana principal con un modelo seleccionado.

Después de seleccionar el modelo que se quiere evaluar, el usuario debe escoger las directrices que va a aplicar al mismo. Aquí puede seleccionar de una vez todas las directrices, o puede seleccionarlas de acuerdo a los diferentes tipos en los que estas se clasifican, como se muestra en la Figura 3.4. Todos los modelos pueden poseer problemas relacionados con estos tipos de directrices y a través de SAD_BPMN estos problemas pueden ser detectados y corregidos tras la aplicación de las mismas.

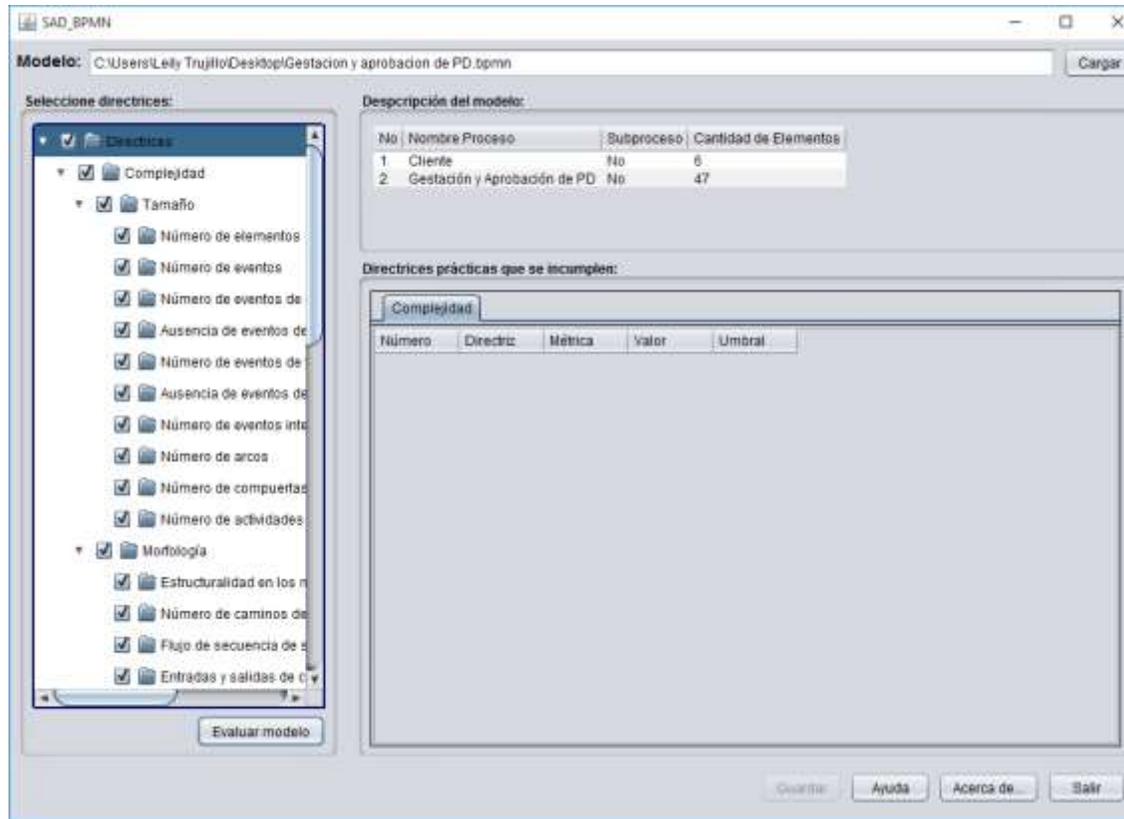


Figura 3.4: Selección de directrices a evaluar en el modelo.

Al pulsar el botón *Evaluar modelo* el usuario podrá ver en la sección “Directrices prácticas que se incumpelen”. Aquí se agrupan aquellas directrices prácticas que se sugiere aplicar al modelo para mejorar su calidad después de su evaluación. Además, se muestra el valor de la medida asociada a la directriz y los umbrales de las mismas cuyo sobrepaso indica el problema relacionado con la directriz en el modelo. En la Figura 3.5 se muestran las directrices de complejidad sugeridas al modelador tras la evaluación del modelo cuyo diagrama se muestra en el anexo 1.

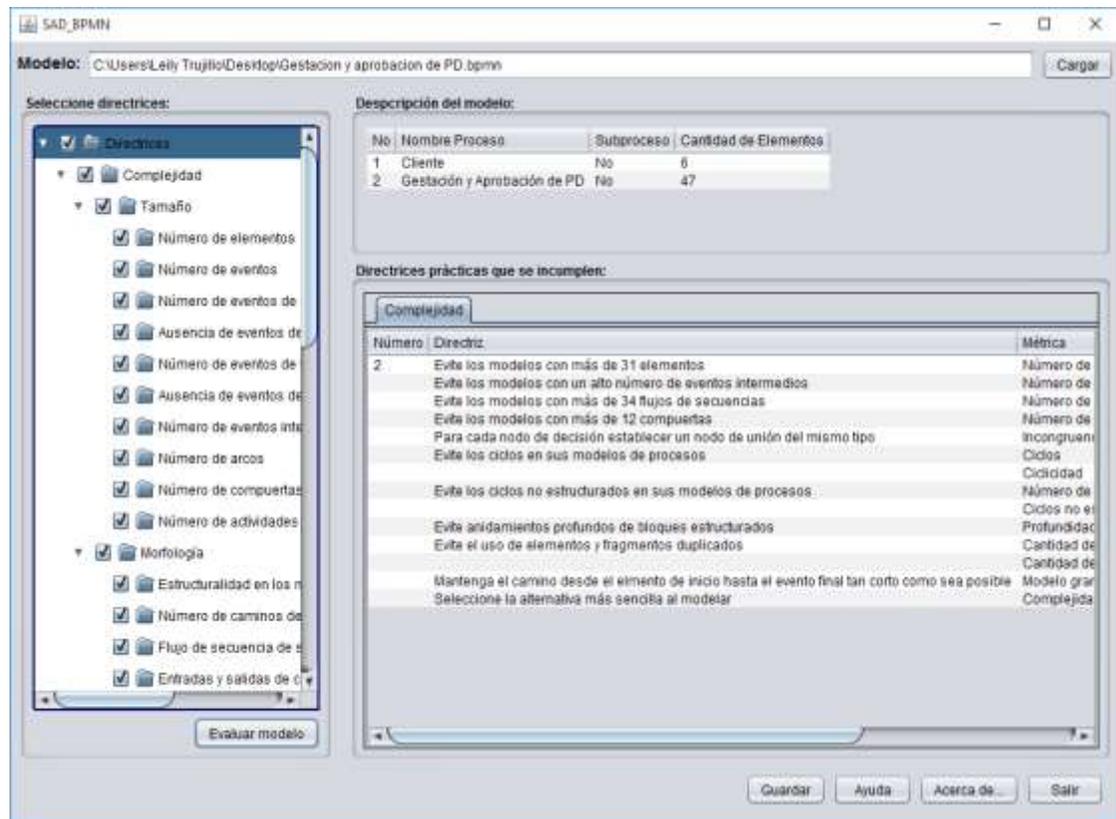


Figura 3.5: Directrices prácticas de complejidad sugeridas al modelador.

El botón Guardar permite crear un documento *.xls* con las directrices prácticas sugeridas que el modelador puede consultar posteriormente e imprimir. Este botón se activa luego de que se haya realizado la evaluación de un modelo. El botón Ayuda, por otra parte, abre un archivo *.pdf* que incluye información de todas las directrices prácticas, así como posibles acciones a realizar para aplicarlas en forma de patrones, la motivación de su uso y las medidas asociadas para la evaluación. El botón Acerca de permite ver los autores de la aplicación así como la versión actual del mismo. Finalmente, el botón Salir posibilita cerrar la aplicación.

3.2. Pruebas y evaluación de la herramienta.

Para comprobar el correcto funcionamiento de SAD_BPMN resulta necesario realizar un conjunto de pruebas que demuestren su calidad. Esta validación se puede llevar a cabo a partir de una serie de pruebas que demuestren la conformidad con los requisitos del software.

3.2.1. Pruebas unitarias

Las pruebas unitarias consisten en el proceso de probar individualmente subprogramas, subrutinas, clases, métodos o procedimientos en un software. Específicamente, en lugar de probar inicialmente el programa en su conjunto, la prueba se centra primero en los bloques de construcción más pequeños del programa [85].

De acuerdo a [86], estas pruebas están centradas en el procesamiento lógico interno y en las estructuras de datos dentro de un componente. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos estén cubiertos, y que ellos funcionen como se espera. Dichas pruebas requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o del código programado.

Casos de prueba.

Los casos de prueba son clases o módulos que deben ser configurados para probar los métodos de una clase. Así, para cada clase que se desea probar se define su correspondiente caso de prueba. De esta forma, se construye un programa que sirve para probar los diferentes módulos de la herramienta. Existen diferentes técnicas para el diseño de pruebas como son el camino básico y las pruebas de decisión-condición. Según lo planteado en [87] el método del camino básico permite al diseñador de casos de pruebas obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

Como caso de prueba se toma el modelo presentado en la Figura 2.3, que representa un proceso de denuncia en el MININT. De forma manual se calculan las medidas asociadas a las directrices al modelo. A continuación se diseñan los casos de prueba para un conjunto de métodos de la herramienta.

- Método *load()*

Este método es el encargado de la carga del modelo y recibe como parámetro el fichero a analizar, aquí se trabaja con ficheros con extensión *.bpmn* o *.xpdl*. Para esto se hace uso de la biblioteca *jdom* de código abierto para Java que permite manipular ficheros *XML*. La idea del método es procesar el documento *XML* del que se obtiene un objeto *Document*

con el cual se trabaja. Luego se recorre el objeto creado y se pregunta por el tipo de atributo para crear las tareas, los eventos y las compuertas. Para cada uno de estos elementos se toma el identificador y el nombre, luego se adiciona un vértice nuevo al seudografo que representa dicho elemento, con su nombre. Para este método se definen tres casos de prueba:

Caso 1: No se selecciona ningún fichero a cargar.

Caso 2: Se selecciona un fichero con extensión *.bpmn* o *.xpdl*.

Caso 3: Se selecciona un fichero con otra extensión.

El método que se muestra a continuación en la Figura 3.6 fuerza la ejecución del caso 3 el cual lanza una excepción al intentar realizar la carga de un fichero con formato diferente a *.bpmn* o *.xpdl*.

```

/**
 * Test of load method, of class LoadBonita.
 */
@Test
public void testLoad() {
    System.out.println("load");
    LoadBonita instance = new LoadBonita();
    try {
        instance.load("prueba.pdf");
        fail("Falla si no se lanza ninguna excepcion");
    } catch (IOException | JDOMException | SAXException | ParserConfigurationException ex) {
        Logger.getLogger(LoadBonitaTest.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Figura 3.6: Prueba del método *load* para el caso 3.

Como se puede observar en la Figura 3.7 la prueba se ejecuta de forma satisfactoria pues se lanza la excepción cuando se intenta extraer los datos de un fichero con un formato inválido evitando así que se ejecute el método *fail*.

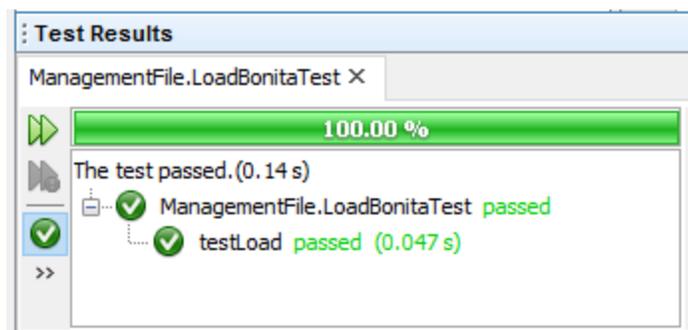


Figura 3.7: Resultado satisfactorio de la prueba *testLoad*.

El siguiente conjunto de pruebas está dirigido al cálculo de medidas asociadas a las directrices de complejidad junto con la evaluación de las mismas. Estas medidas están implementadas en la clase *Tamanno* perteneciente al paquete *directrices*. A continuación se describen dos métodos para la evaluación de un subconjunto de las directrices, siendo estas evaluadas a través del cálculo de las medidas de calidad asociadas a ellas.

Método 1: *directriz_Problema1*

Los números de los problemas se hacen corresponder con los mencionados en el epígrafe 2.1, en este caso el problema 1 evalúa la cantidad de elementos en el modelo, la cual debe de ser menor que 31 según el umbral asociado. Dicho problema tiene asociada la medida *número de elementos* ($S_N(G) = |N|$). Se espera que el resultado del cálculo de la medida sea 42 por lo que se incumple la directriz dando como salida valor true. En la

```

/**
 * Test of directriz_Problema1 method, of class Tamanno.
 */
@Test
public void testDirectriz_Problema1() {
    Tamanno instance = new Tamanno();
    boolean expectedResult = true;
    boolean result = instance.directriz_Problema1(p);
    assertEquals(expectedResult, result);
    System.out.println("directriz_Problema1");
}

/**
 * Test of metrica_Sn method, of class Tamanno.
 */
@Test
public void testMetrica_Sn() {
    Tamanno instance = new Tamanno();
    int expectedResult = 42;
    int result = instance.metrica_Sn(p);
    assertEquals(expectedResult, result);
    System.out.println("metrica_Sn");
    // TODO review the generated test code and remove the default call to fail.
}

```

Figura 3.8: se presentan los métodos de prueba para esta directriz y la medida asociada a ella.

```

/**
 * Test of directriz_Problema1 method, of class Tamanno.
 */
@Test
public void testDirectriz_Problema1() {
    Tamanno instance = new Tamanno();
    boolean expectedResult = true;
    boolean result = instance.directriz_Problema1(p);
    assertEquals(expectedResult, result);
    System.out.println("directriz_Problema1");
}

/**
 * Test of metrica_Sn method, of class Tamanno.
 */
@Test
public void testMetrica_Sn() {
    Tamanno instance = new Tamanno();
    int expectedResult = 42;
    int result = instance.metrica_Sn(p);
    assertEquals(expectedResult, result);
    System.out.println("metrica_Sn");
    // TODO review the generated test code and remove the default call to fail.
}

```

Figura 3.8: Métodos diseñados para las pruebas de la *directriz_Problema1* y *metrica_SN*.

Al ejecutar la prueba el valor esperado coincide con el resultado por lo que la prueba es exitosa. En la Figura 3.9 se muestra el resultado satisfactorio de la ejecución de la prueba para la *directriz_Problema1* y el cálculo de la medida asociada a ella.

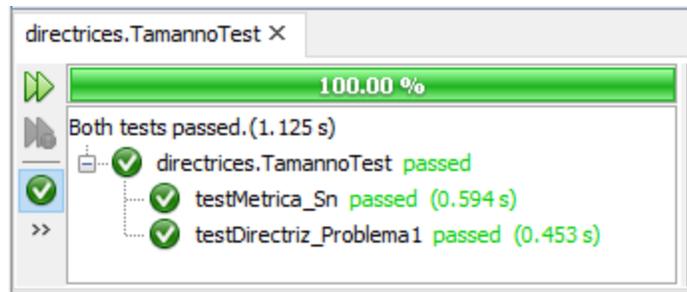


Figura 3.9: Resultado satisfactorio de la prueba *directriz_Problema1*.

Método 2 *directriz_Problema4*

Esta directriz es la encargada de analizar la ausencia de eventos de inicio en el modelo, es decir, evalúa si existen o no estos eventos. La medida asociada a dicha directriz es $S_{E_S}(G) = |E_S|$, la cual calcula la cantidad de eventos de inicio en el modelo. La directriz se lanza si el modelo no contiene eventos de inicio por lo que el valor esperado para este caso es *false*. En la Figura 3.10 se observa la implementación de los métodos asociados a las pruebas de la directriz antes mencionada y su medida asociada, donde se aprecia que

el valor esperado al calcular la medida es 3 por lo que la directriz se muestra en la herramienta al obtener *false* como resultado.

```

/**
 * Test of directriz_Problema4 method, of class Tamanno.
 */
@Test
public void testDirectriz_Problema4() {
    System.out.println("directriz_Problema4");
    Tamanno instance = new Tamanno();
    boolean result = instance.directriz_Problema4(p);
    assertFalse(result);
}

/**
 * Test of metrica_TNSE method, of class Tamanno.
 */
@Test
public void testMetrica_TNSE() {
    System.out.println("metrica_TNSE");
    Tamanno instance = new Tamanno();
    int expectedResult = 3;
    int result = instance.metrica_TNSE(p);
    assertEquals(expectedResult, result);
}

```

Figura 3.10: Métodos diseñados para las pruebas de *directriz_Problema4* y *metrica_TNSE*. Como se observa en la Figura 3.11 la prueba es exitosa ya que se se obtienen los valores esperados.

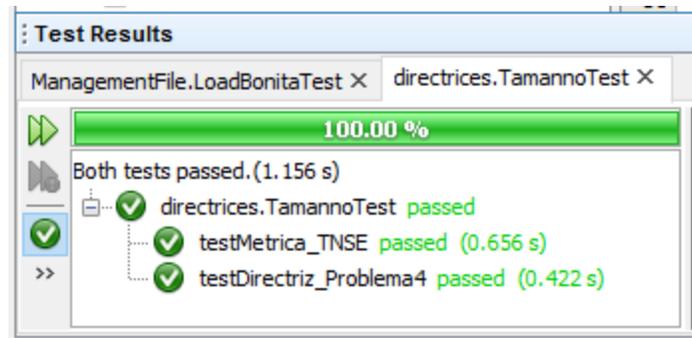


Figura 3.11: Resultado satisfactorio de la prueba *directriz_Problema4*.

3.2.2. Pruebas de integración

Las pruebas de integración prueban de manera combinada los módulos que componen el software. Dichos módulos se van acoplando progresivamente en conjuntos. Luego de cada

acople, se prueba la correcta interacción entre los módulos. Una vez que se haya verificado que el conjunto funciona de acuerdo con lo previsto, se le suma un nuevo módulo y se vuelven a realizar pruebas. Así, el conjunto va creciendo y se prueba con cada agregado hasta completar la aplicación. Durante la integración, las técnicas que más se utilizan son las de caja negra, aunque se pueden llevar a cabo algunas pruebas de caja blanca para asegurar que se cubren los principales flujos de comunicación entre las unidades [87].

Para ejecutar la prueba de integración se evalúa el modelo de proceso de negocio utilizado en la sección anterior. La prueba inicia con la carga del modelo, visualizando el nombre del proceso, la cantidad de elementos y si es un subprocesso o no. Luego se seleccionan las directrices que se desean evaluar en el modelo, se evalúa el mismo y se muestran los datos en la aplicación. En la Figura 3.12 se muestra el resultado de la ejecución de dicha prueba.

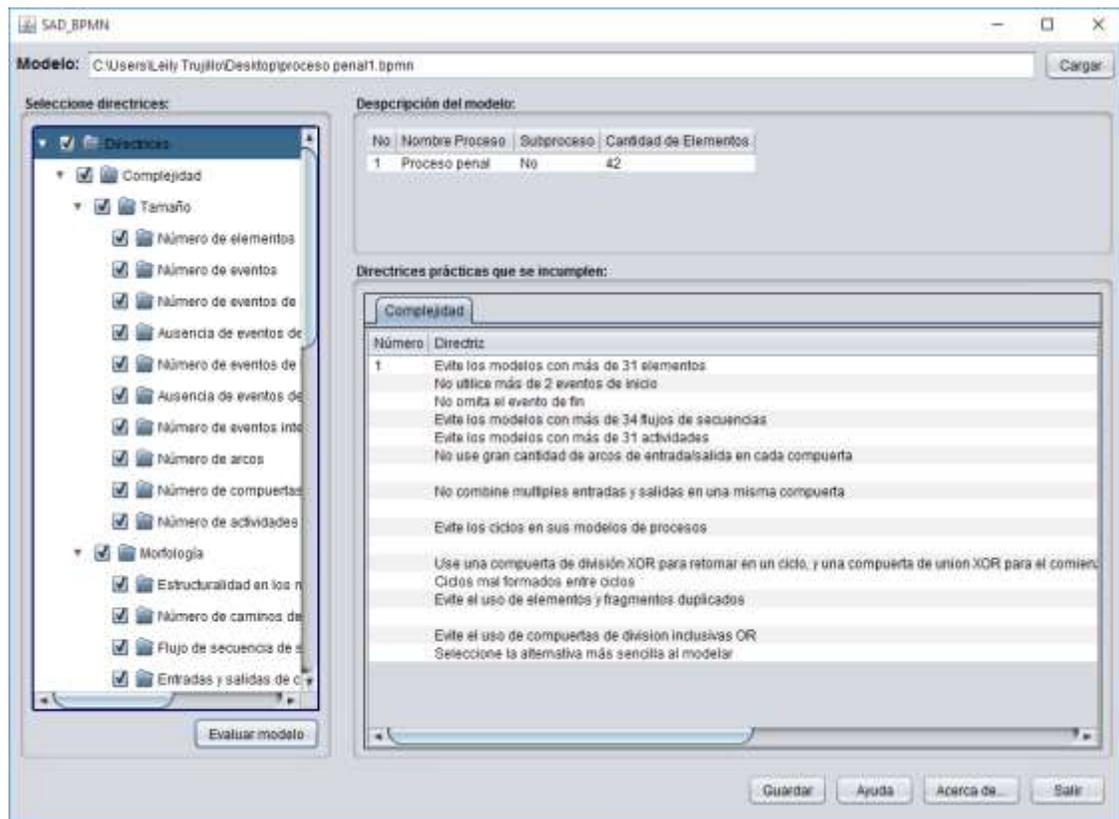


Figura 3.12: Resultado de la prueba de integración de evaluar el modelo.

En la sección 1 se muestra el resultado de cargar un modelo donde se visualizan los datos del modelo. La sección 2 muestra el resultado de evaluar las directrices en dicho modelo.

3.3. Análisis de la aceptación de la herramienta SAD_BPMN por parte de los usuarios.

Al finalizar el desarrollo de una herramienta computacional, es importante alcanzar dos niveles de calidad relacionados con su evaluación: es en la evaluación: retroalimentación verbal, por escrito o almacenando historias para sacar conclusiones sobre el uso de los resultados obtenidos, y realización de encuestas globales que contengan cuestionarios a usuarios sobre percepción, uso y valor de los resultados [88, 89]. En la presente investigación se seleccionó la encuesta a los usuarios del sistema como una vía de evaluar la aceptación de la herramienta por los usuarios. En la encuesta se incluyen preguntas que están enfocadas en la aplicación y en los usuarios, ya que es objetivo de la misma conocer la satisfacción de los usuarios con los resultados obtenidos. Se verifica que la herramienta sea flexible, esto indica si el grado de ajuste y configuraciones dependiendo de la necesidad de cada usuario es adecuado. Otro aspecto importante es la transparencia de la herramienta, en la medida que es entendible y sus salidas adecuadas para cada entrada. También se mide la usabilidad, sobre la cual pueden influir los aspectos del diseño de la misma [90].

Los participantes en este estudio fueron 19 estudiantes de la carrera de Ingeniería Informática de la Universidad Central “Marta Abreu” de Las Villas. Con el objetivo de preparar a los estudiantes en la tarea de modelado, estos fueron entrenados en BPMN usando la herramienta de modelado Bonita. Primeramente, se les pidió a los participantes que modelaran un proceso de negocio de tamaño medio que fue extraído de sistemas reales usando BPMN. Posteriormente, se les dio un entrenamiento explicándoles un conjunto representativo de directrices prácticas de complejidad. Seguidamente, se les pidió a los participantes que consideraran la posible aplicación de las directrices a los modelos de procesos de negocio obtenidos haciendo uso de la herramienta SAD_BPMN o de forma manual. En la encuesta las preguntas de la 12 a la 16 están dirigidas al funcionamiento de la herramienta a través de la evaluación de la flexibilidad, usabilidad, rendimiento y capacidad de soporte de la misma. El 100% de los encuestados respondieron que habían utilizado al menos una vez la herramienta para la evaluación de los modelos. Al interpretar los resultados el 100% de los encuestados coinciden que se hace de una manera muy fácil

por lo que la familiarización con la herramienta fue muy rápida. Por otra parte, el 84.21% respondió que la herramienta ofrece una ayuda útil para la evaluación de los modelos.

Para evaluar si existe o no interrelación entre las diferentes preguntas relacionadas con la utilidad de la herramienta, se calculó el coeficiente de correlación de Pearson. La Tabla 3.1 muestra la correlación que existe entre las variables y su significación.

Tabla 3.1: Correlación entre resultados de la encuesta para las preguntas ocho, nueve y diez.

	Pearson	Pregunta 9	Pregunta 10
Pregunta 8	Coef. Correlación	.896(**)	.774(**)
	Significación	.000	.000

La tabla muestra la correlación positiva que existe entre la toma de decisiones y los nuevos enfoques e ideas de modelado con los resultados obtenidos de evaluar el modelo.

3.4. Conclusiones parciales

La herramienta SAD_BPMN posibilita la evaluación de modelos a través de directrices prácticas de morfología con el objetivo de mejorar la calidad de los modelos de procesos de negocio. En este capítulo se presentó una descripción detallada de la herramienta. Además como parte de su validación se realizó un conjunto de pruebas unitarias con el objetivo de evaluar el funcionamiento de las directrices de calidad y las medidas asociadas a ellas. También se realizaron pruebas de integridad para comprobar la correcta comunicación entre los módulos de la herramienta. Los resultados de las encuestas muestran que la herramienta fue aceptada satisfactoriamente por los usuarios facilitando la evaluación de los modelos de procesos de negocio.

CONCLUSIONES

Como resultado de esta investigación, se ha arribado a las conclusiones siguientes:

1. Se seleccionaron las herramientas de modelado de procesos de negocio Bizagi y Bonita para la interpretación y evaluación de sus archivos de salida.
2. Se almacenó la información contenida en los archivos de salida generados por herramientas de modelado, utilizando pseudografos como estructura de datos, para su posterior análisis.
3. Se desarrolló una herramienta para la evaluación de modelos de procesos de negocio tomando como base directrices prácticas de complejidad haciendo uso de medidas de calidad asociadas a las mismas.

RECOMENDACIONES

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

- Adicionar un módulo a la herramienta SAD_BPMN que permita la evaluación de los modelos de procesos de negocio a través de las directrices de representación visual.
- Extender la herramienta SAD_BPMN para que permita la interpretación de archivos de salida exportados por otras herramientas de modelado de proceso de negocio.

REFERENCIAS BIBLIOGRÁFICAS

1. Laudon, K.C. and J.P. Laudon, *Management information systems*. Vol. 8. 2011: Prentice Hall New Jersey.
2. Rainer, R.K., et al., *Introduction to information systems: Supporting and transforming business*. 2013: John Wiley & Sons.
3. Stair, R. and G. Reynolds, *Fundamentals of information systems*. 2012: Cengage Learning.
4. Bizagi. *Bizagi Process Modeler User's Guide*. 2015 March 2015]; Available from: <http://help.bizagi.com/processmodeler/es/>.
5. Multamäki, M., *Objective-driven planning of business process modeling*. Department of Industrial Engineering and Management, Helsinki University of Technology, 2002.
6. Mendling, J., H.A. Reijers, and J. Recker, *Activity labeling in process modeling: Empirical insights and recommendations*. *Information Systems*, 2010. **35**(4): p. 467-482.
7. Bandara, W., G.G. Gable, and M. Rosemann, *Factors and measures of business process modelling: model building through a multiple case study*. *European Journal of Information Systems*, 2005. **14**(4): p. 347-360.
8. Mendling, J., H.A. Reijers, and W.M.P. van der Aalst, *Seven process modeling guidelines (7PMG)*. *Information and Software Technology*, 2010. **52**(2): p. 127-136.
9. Mendling, J., *Detection and Prediction of Errors in EPC Business Process Models*. 2007, Vienna University of Economics and Business Administration.
10. Silver, B. and B. Richard, *BPMN method and style*. Vol. 2. 2009: Cody-Cassidy Press Aptos.

11. Miskin, C., *Taxonomies*. Legal Information Management, 2002. **2**(1): p. 16.
12. Wohed, P., et al., *Pattern-based Analysis of BPMN*. aprox 2006.
13. Davenport, T.H., *Process innovation: reengineering work through information technology*. 2013: Harvard Business Press.
14. van Der Aalst, W.M.P., et al., *Workflow Patterns*. 2002.
15. Marshall, C., *Enterprise Modeling with UML. Designing Successful Software through Business Analysis*. 2000.
16. Leopold, H., S. Smirnov, and J. Mendling, *On the refactoring of activity labels in business process models*. Information System, 2012. **37**(5): p. 443-459.
17. Mendling, J., *Detection and Prediction of Errors in EPC Business Process Models*, in *Institute of Information Systems and New Media*. 2007, Vienna University of Economics and Business Administration (WU Wien): Austria.
18. Kühne, T., *Matters of (meta-) modeling*. Software and Systems Modeling, 2006. **5**(4): p. 369–385.
19. Moody, D.L., *Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions*. Data & Knowledge Engineering, 2005. **55**(3): p. 243-276.
20. Lindland, O.I., G. Sindre, and A. Solvberg, *Understanding quality in conceptual modeling*. IEEE Software 1994. **11**(2): p. 42-49.
21. Krogstie, J., G. Sindre, and H. Jørgensen, *Process models representing knowledge for action: a revised quality framework*. European Journal of Information Systems, 2006. **15**(1): p. 91-102.
22. Reijers, H., J. Mendling, and J. Recker, *Business process quality management*, in *Handbook on Business Process Management 2010*, Springer p. 167-185.
23. Nelson, H.J., et al., *A conceptual modeling quality framework*. Software Quality Journal, 2012. **20**: p. 201-228.

24. Schütte, R. and T. Rotthowe, *The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models*, in *Proceedings of the 17th International Conference on Conceptual Modeling*, T. Wang Ling, S. Ram, and M.-L. Lee, Editors. 1998, Springer: Singapore. p. 240–254.
25. Becker, J., M. Rosemann, and C. von Uthmann, *Guidelines of business process modeling*, in *Business Process Management*, W. van der Aalst, J. Desel, and A. Oberweis, Editors. 2000, Springer Berlin / Heidelberg. p. 241-262.
26. Pérez-Santiago, J., *Directrices Prácticas para la Modelación Conceptual de Procesos del Negocio*. 2014, Universidad Central "Marta Abreu" de las Villas. p. 1-84.
27. Sánchez-González, L., *Toward a quality framework for business process models*. *International Journal of Cooperative Information Systems*, 2013. **22**(1): p. 35.
28. Sánchez-González, L., et al., *A case study about the improvement of business process models driven by indicators*. *Software and Systems Modeling* [online], 2015. **2015**(July): p. 1-30.
29. Sánchez-González, L., et al. *Towards thresholds of control flow complexity measures for BPMN models*. in *Proceedings of the 2011 ACM symposium on Applied computing*. 2011. ACM.
30. Sánchez-González, L., GARCÍA, F., MENDLING, J., RUIZ, F. & PIATTINI, M., *Prediction of Business Process Model Quality Based on Structural Metrics*, in *Proceedings 29th International Conference on Conceptual Modeling - ER 2010*. 2010.
31. Mendling, J., *Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness*. 2008: Springer-Verlag Berlin Heidelberg.
32. Gruhn, V.L., R., *Complexity metrics for business process models*. *9th International Conference on Business Information Systems (BIS 2006)*. 2006: Klagenfurt.

33. Mendling, J., et al., *Thresholds for error probability measures of business process models*. Journal of Systems and Software, 2012. **85**(5): p. 1188-1197.
34. Sánchez-González, L., et al., *Quality indicators for business process models from a gateway complexity perspective*. Information and Software Technology, 2012. **54**(11): p. 1159-1174.
35. Sánchez-González, L., et al., *Improving Quality of Business Process Models*, in *Evaluation of Novel Approaches to Software Engineering, ENASE 2011*, L.A. Maciaszek and K. Zhang, Editors. 2013, Springer: Beijing, China. p. 130-144.
36. Nissen, M.E., *Redesigning reengineering through measurement-driven inference*. MIS Quarterly, 1998. **22**(4): p. 509–534.
37. Moreno-Montes de Oca, I. and M. Snoeck, *Pragmatic guidelines for Business Process Modeling*. 2015: KU Leuven -FEB -Management Information Systems Group.
38. Qi, Y.-d., N. Qu, and X.-f. Xie, *Towards a preliminary ontology for conceptual model quality evaluating*, in *Proceedings of the International Conference on Web Information Systems and Mining (WISM) (Volume 1)*, F. Lee-Wang, et al., Editors. 2010, Springer-Verlag Berlin Heidelberg: Sanya, China. p. 329–334.
39. Mohagheghi, P., V. Dehlen, and T. Neple, *Definitions and approaches to model quality in model-based software development – A review of literature*. Information and Software Technology, 2009. **51**(12): p. 1646-1669.
40. Rittgen, P., *Quality and perceived usefulness of process models*, in *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010, ACM New York, USA: Sierre, Switzerland. p. 65-72.
41. Daneva, M. *A best practice based approach to CASE-tool selection*. in *4th IEEE International Software Engineering Standards Symposium and Forum (ISESS'99), "Best Software Practices for the Internet Age"*. 1999. IEEE.
42. OMG, *UML Superstructure Specification version 2.0*, O.M. Group, Editor. 2004: Massachusetts, USA.

43. Rodríguez, A.F.-M., E., Piattini, M., *Capturing Security requirements in Business Processes Through a UML 2.0 Activity Diagrams Profile*. Lecture Notes in Computing Science, 2006. **4231**: p. 32-42.
44. Wohed, P.R., N. van, A. W. M. P., Dumas, M., Hofstede, A. H. M., *Pattern Based Analysis of the Control-Flow Perspective of UML Diagrams*. ACM Transactions on Programming Languages and System, 2005. **3716**(63-78).
45. Russell, N., van der Aalst, W. M., ter Hofstede, A. H., and Wohed, P., *On the suitability of UML 2.0 activity diagrams for business process modeling*, in *Proceeding of the 3rd Asia-Pacific Conference on Conceptual Modelling*. 2006.
46. Wohed, P.R., N. van, A. W. M. P., Dumas, M., Hofstede, A. H. M., *On the Suitability of BPMN for Business Process Modeling*. Lecture Notes in Computing Science, 2006. **4102**(161-176).
47. Korherr, B.L., B., *Extending the EPC and the BPMN with Business Process Goals and performance Measures*, in *International Conference on Enterprise Information Systems (ICEIS)*. 2003: Madeira, Portugal. p. 287-294.
48. Hommes, L.J., *The evaluation of business process modeling techniques*. 2004: TU Delft, Delft University of Technology.
49. Laue, R. and J. Mendling, *Structuredness and its significance for correctness of process models*. Information Systems and E-Business Management, 2010. **8**(3): p. 287-307.
50. Kikuchi, Y., et al., *Analysis and Modeling of Information Required for Process Assessment on Environment, Health, and Safety by IDEF0 and UML*, in *11th International Symposium on Process Systems Engineering, Pts a and B*, I.A. Karimi and R. Srinivasan, Editors. 2012. p. 1392-1396.
51. Geiger, M. and G. Wirtz, *Detecting Interoperability and Correctness Issues in BPMN 2.0 Process Models*. Zeus, 2011: p. 41-44.
52. White, S., *Business Process Modeling Notation (BPMN) Specification-(2006)*. OMG.

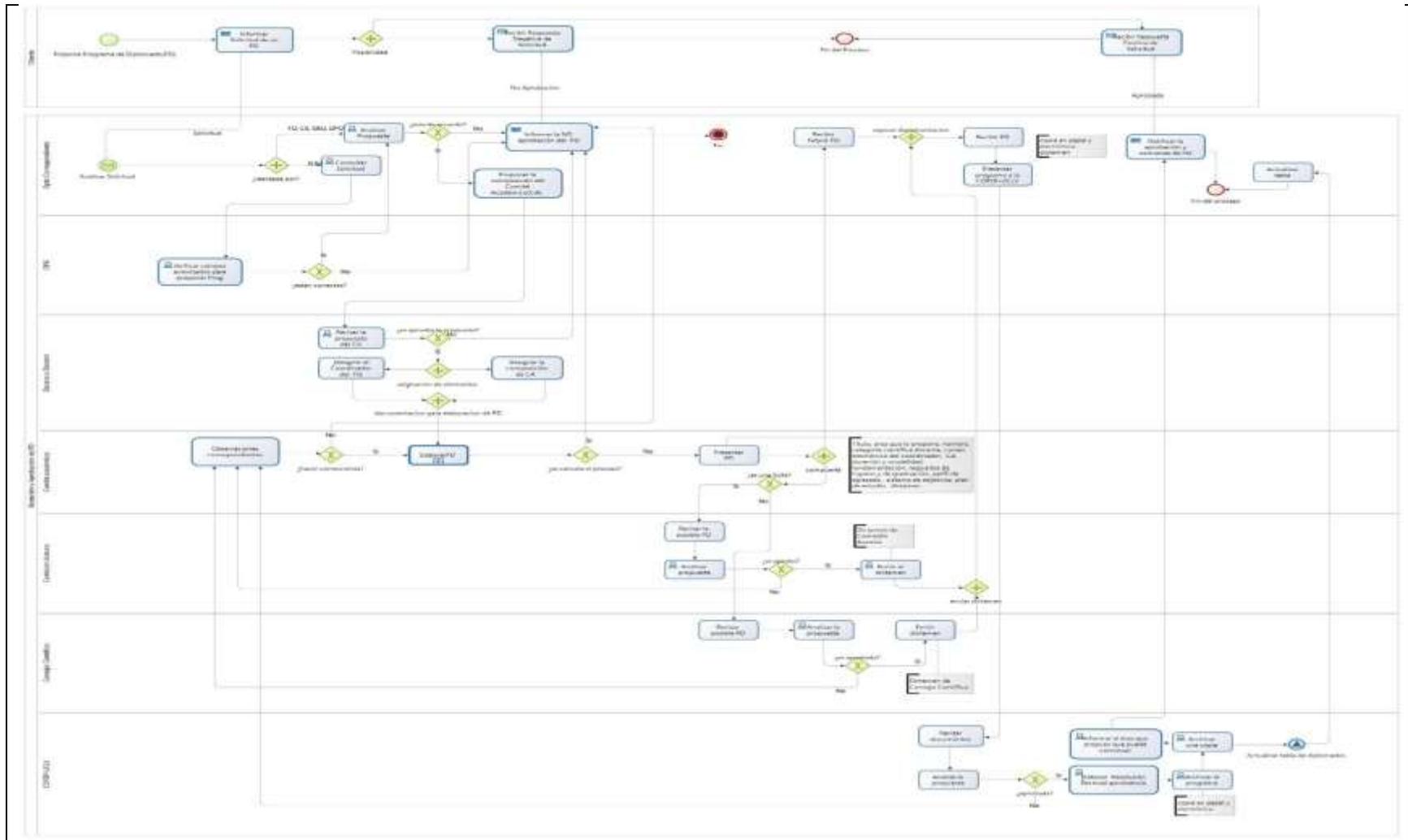
53. Wohed, P., et al., *On the suitability of BPMN for business process modelling*, in *Business Process Management, Proceedings*, S.F.J.L.S.A. Dustdar, Editor. 2006. p. 161-176.
54. Chinosi, M. and A. Trombetta, *BPMN: An introduction to the standard*. Computer Standards & Interfaces, 2012. **34**(1): p. 124-134.
55. White, S.A., *Introduction to BPMN*. IBM Corporation (2004). 2013.
56. Brambilla, M., P. Fraternali, and C. Vaca. *BPMN and design patterns for engineering social BPM solutions*. in *International Conference on Business Process Management*. 2011. Springer.
57. Ciuciu, I., et al., *Semantic support for security-annotated business process models*, in *Enterprise, Business-Process and Information Systems Modeling*. 2011, Springer. p. 284-298.
58. Stallman, R., *Gnu general public license*. Free Software Foundation, Inc., Tech. Rep, 1991.
59. Harmon, P. and C. Wolf. *Business Process Trends; Business Process Modeling Survey*. 2011 [cited 2015 July]; Available from: http://www.bptrends.com/bpt/wp-content/surveys/Process_Modeling_Survey-Dec_11_FINAL.pdf.
60. BPMN-Forum. *BPMN tool related posts and discussions*. 2015 [cited 2015 July]; Available from: <http://bpmnforum.com/bpmn-tools>.
61. Krumnow, S. and G. Decker, *A concept for spreadsheet-based process modeling*, in *Business Process Modeling Notation*. 2010, Springer. p. 63-77.
62. Kluza, K., et al. *Opportunities for Business Process semantization in open-source process execution environments*. in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. 2015. IEEE.
63. Castillo, P.A.A., *BONITA SOFT: Gestor de procesos de negocios BPM*. 2011, Bogotá.

64. Signavio. *Guidelines by convention: Signavio Best Practice*. 2014 [cited 2015 2015]; Available from: <http://academic.signavio.com>.
65. Wang, M. and L. Lu. *A transformation method from UML statechart to Petri nets*. in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*. 2012. IEEE.
66. Snoeck, M., et al. *Testing a Selection of BPMN Tools for Their Support of Modelling Guidelines*. in *IFIP Working Conference on The Practice of Enterprise Modeling*. 2015. Springer.
67. West, D.B., *Introduction to graph theory*. Vol. 2. 2001: Prentice hall Upper Saddle River.
68. Biggs, N.L., *Matemática discreta*. 1994: Vicens Vives.
69. Mendling, J., M. Strembeck, and J. Recker, *Factors of process model comprehension-Findings from a series of experiments*. *Decision Support Systems*, 2012. **53**(1): p. 195-206.
70. Weber, B., et al., *Refactoring large process model repositories*. *Computers in Industry*, 2011. **62**(5): p. 467-486.
71. Dumas, M., et al., *Fundamentals of Business Process Management*. 2012: Springer Heidelberg New York Dordrecht London.
72. Sánchez-González, L., et al., *Quality Assessment of Business Process Models Based on Thresholds*, in *On the Move to Meaningful Internet Systems*, R. Meersman, T. Dillon, and P. Herrero, Editors. 2010. p. 78-95.
73. Silver, B., *BPMN: Method and Style*. 2009: Cody-Cassidy Press.
74. Polyvyanyy, A., L. García-Bañuelos, and M. Dumas, *Structuring acyclic process models*, in *Business Process Management*. 2010, Springer. p. 276-293.
75. Koehler, J. and J. Vanhatalo, *Process anti-patterns: How to avoid the common traps of business process modeling*, in *Tech. Rep. Report RZ-3678*. 2007, IBM Zurich Reseach Lab.

76. Kindler, E., *On the semantics of EPCs: Resolving the vicious circle*. Data & Knowledge Engineering, 2006. **56**: p. 23-40.
77. Dumas, M., et al., *Understanding Business Process Models: The Costs and Benefits of Structuredness*, in *Advanced Information Systems Engineering*, J. Ralyté, et al., Editors. 2012, Springer Berlin Heidelberg. p. 31-46.
78. Eric Freeman, E.F., Bert Bates, Kathy Sierra, *Head first design patterns*. 2008.
79. Carrasco, A.P., J.Á.V. Iturbide, and F.A. Martínez, *La representación de algoritmos diseñados bajo la técnica "divide y vencerás"*. Indagatio Didactica, 2011. **3**(3).
80. García González, C.E., *Enfoque sistémico a la modelación de datos, base para el desarrollo de una herramienta CASE*, in *Departamento de Ciencia de la Computación*. 2010, Universidad Central "Marta Abreu" de Las Villas. p. 123.
81. Sommerville, I., *Ingeniería de software*. 6ta ed. 2002.
82. Cristiá, M., *Introducción a la Ingeniería de Requisitos*. 2011.
83. Ivar Jacobson, G.B., James Rumbaugh, *El proceso unificado de desarrollo de software*. 1ra ed. 2000.
84. Larman, C., *UML y Patrones: una introducción al análisis y diseño a objetos y al proceso unificado*. 2003, Madrid: Pearson Educación.
85. Myers, G.J., C. Sandler, and T. Badgett, *THE ART OF SOFTWARE TESTING*. Tercera ed. 2012.
86. Gómez, D., D. Jústiz, and M. Delgado, *Unit Tests of Software in a University Environment*. Computación y Sistemas, 2013. **17**: p. 69-77
87. Pressman, R.S., *Software Engineering. A Practitioner's Approach*. Seven ed. 2010.
88. Mertins, K., P. Heisig, and J. Vorbeck, *Knowledge management: concepts and best practices*. 2003: Springer Science & Business Media.

89. Kohavi, R., et al., *Web mining*. Data Mining and Knowledge Discovery, 2002. **6**(1): p. 5-8.
90. Nielsen, J. and H. Loranger, *Usabilidad: prioridad en el diseño Web*.

Anexo 1. Modelo “Gestión y aprobación de PD”.



Anexo 2. Encuesta a los usuarios de la herramienta SAD_BPMN

Encuesta a los usuarios de la herramienta SAD_BPMN

Esta encuesta se realiza con el propósito de conocer las opiniones que se tienen acerca de la herramienta SAD_BPMN y así poder evaluar la misma siguiendo los criterios de sus usuarios. Los objetivos de la evaluación son investigar y explorar acerca de la calidad y utilidad de los resultados alcanzados con la herramienta. Las principales variables de medición son la eficacia, eficiencia, conformidad, confiabilidad y robustez; aunque algunos elementos de usabilidad, como facilidad de uso, utilidad de uso e intención de uso serán medidos. Por ello le pediría que fuera tan amable de contestar las preguntas que contribuyen a medir las variables y objetivos trazados. No le tomará más de 10 minutos.

Le agradecemos que conteste este cuestionario con la mayor sinceridad posible, después de haber probado el sistema con varios modelos de procesos de negocio. No hay respuestas incorrectas ni correctas.

Las preguntas con variantes de respuestas encabezadas con "O" permiten una única respuesta y aquellas con variantes de respuestas encabezadas con "☒" son de selección múltiple.

1. Ha utilizado el sistema:

Frecuentemente Algunas veces Casi nunca

2. ¿Con qué tipos de modelos ha trabajado?

Menos de 15 elementos Entre 15 y 30 elementos Más de 30 elementos

3. Lenguajes de modelado que ud. conoce:

4. La familiarización con la herramienta fue:

Fácil Relativamente fácil Aceptable Relativamente difícil Difícil

5. ¿Se encuentra satisfecho con la evaluación obtenida de las directrices?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

6. ¿Cuáles son los problemas que más desea evaluar en sus modelos?

7. La presentación de los resultados de la evaluación del modelo le permiten:

Alta interpretabilidad Adecuada interpretabilidad Baja interpretabilidad

8. ¿Descubre conocimiento con los resultados de la evaluación de sus modelos?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

9. ¿Los resultados de la evaluación contribuyen a la toma de decisiones para mejorar la calidad del modelo?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

10. ¿Los resultados le muestran nuevos enfoques e ideas del modelo de proceso de negocio que ha evaluado?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

11. ¿La evaluación basada en directrices le facilita el análisis de los modelos de procesos?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

12. Interpretar los resultados de la herramienta es:

Muy fácil Fácil Posible Difícil Muy difícil

13. La herramienta ofrece una ayuda:

Muy útil Adecuada Poco útil

14. ¿El ambiente de la herramienta le permite utilizarla con facilidad?

Definitivamente sí En alguna medida Definitivamente no

15. ¿Pretende continuar usando la herramienta para evaluar modelos de procesos de negocio?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

16. ¿La herramienta le permite una mayor productividad en el modelado de procesos de negocio?

Definitivamente sí Casi siempre Algunas veces Casi nunca Definitivamente no

PROCESOS DE NEGOCIO, CALIDAD, LENGUAJES Y
HERRAMIENTAS DE MODELADO

2

EVALUACIÓN DE MODELOS DE PROCESO DE NEGOCIO UTILIZANDO TEORÍA DE GRAFOS

3

HERRAMIENTA PARA LA EVALUACIÓN DE MODELOS
DE PROCESOS DE NEGOCIO SAD_BPMN