



TRABAJO DE DIPLOMA

Título: “Herramienta para la estimación del esfuerzo de desarrollo de proyectos de software a partir de modelos de procesos de negocio. SDEffort-BPMN”

Autora: Daydeé Padilla León

Tutora: MSc. Yaimara Granados Hondares

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

DIPLOMA THESIS

Title: "Computer tool for estimating the effort of development software projects from business process models. SDEffort-BPMN"

Author: Daydeé Padilla León

Thesis Director: MSc. Yaimara Granados Hondares

Santa Clara, June, 2019
Copyright©UCLV

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

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

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

Teléfonos.: +53 01 42281503-1419

PENSAMIENTO

"El éxito en todos los tipos de organización depende cada vez más del desarrollo de soluciones de software personalizadas. "

Barry Boehm

DEDICATORIA

A mis padres por su apoyo incondicional, por su amor y dedicación.

A mi hermana por guiarme, animarme y estar junto a mí.

A mis sobrinos por quererme y hacerme feliz.

A toda mi familia y amigos.

AGRADECIMIENTOS

A mi familia por su amor y comprensión.

A mi pareja Adrian por su dedicación y apoyo en todo momento.

A mi tutora Msc. Yaimara Granados Hondares por toda la ayuda prestada.

A todos mis compañeros de clases.

A todos los profesores que de una forma u otra tuvieron que ver con mi formación.

RESUMEN

Los modelos de procesos de negocio, como entrada inicial al proceso de desarrollo de software, constituyen una variante novedosa para lograr la estimación temprana del esfuerzo de desarrollo. En la etapa de identificación de la problemática y conformación de pactos cliente-desarrollador, determinar períodos o fechas específicas del producto a desarrollar es fundamental. Es por ello que la presente investigación ofrece una herramienta informática para predecir el esfuerzo de desarrollo de software en horas/hombre, basado en modelos de procesos de negocio, que han sido obtenidos a partir de herramientas de modelado con notación BPMN, en diversos formatos de salida, como BPMN, XPD, BPM y XML. Se obtiene un modelo predictivo del esfuerzo utilizando la técnica regresión lineal múltiple, que utiliza como variable dependiente el esfuerzo y como variables independientes métricas correspondientes a elementos que la propia herramienta extrae de forma automática de dichos modelos de procesos de negocio, con el fin de brindar una estimación del esfuerzo confiable y de valor para el jefe de proyecto o encargado de planificar el cronograma de entrega de un proyecto en cuestión. La herramienta utiliza una arquitectura MVC y como principio central de diseño el patrón *Strategy*, el cual proporciona flexibilidad para incorporar nuevos formatos de salida, cuya interpretación sea de interés. De ahí que posea como contribución adicional la conformación de una base de casos con vista a investigaciones futuras.

ABSTRACT

The business process models, as an initial input to the software development process, constitute a novel variant to achieve the early estimation of the development effort. In the stage of identification of the problem and conformation of client-developer pacts, determining periods or specific dates of the product to be developed is fundamental. That is why this research offers a computer tool to predict the development effort of software in man-hours, based on business process models, which have been obtained from modeling tools with BPMN notation, in various formats output, such as BPMN, XPD, BPM and XML. A predictive model of the effort is obtained using the multiple linear regression technique, which uses as a dependent variable the effort and as independent metric variables corresponding to elements that the tool itself automatically extracts from said business process models, in order to provide an estimate of the reliable effort and value for the project manager or person in charge of planning the delivery schedule of a project in question. The tool uses an MVC architecture and as a central design principle the Strategy pattern, which provides flexibility to incorporate new output formats, whose interpretation is of interest. Hence, it has as an additional contribution the creation of a base of cases with a view to future research.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
RESUMEN	iv
INTRODUCCIÓN	1
Organización del informe	4
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 Estimación del esfuerzo de desarrollo de software.	5
1.1.1 Métodos para la estimación de esfuerzo en proyectos de software.	5
1.1.2 Técnicas de inteligencia artificial para la estimación de esfuerzo en proyectos de software.	9
1.2 Herramientas para estimar el esfuerzo de desarrollo.	12
1.3 Los procesos de negocio como punto de partida para estimar el esfuerzo de desarrollo en proyectos de software.	13
1.3.1 Caracterización de los Modelos de Proceso de Negocio	14
1.3.2 Herramientas para el modelado de procesos de negocio.	14
1.3.3 Formatos de salida de las herramientas de modelado de procesos de negocio.	16
1.3.4 Métricas para la estimación de esfuerzo en modelos de proceso de negocio.	17
1.4 Estimación de esfuerzo basado en modelos de procesos de negocio.	19
1.5 Fundamentación de la metodología utilizada.	21
1.6 Tecnologías a utilizar.	22
1.7 Conclusiones del capítulo	23
CAPÍTULO 2. MODELO DE NEGOCIO Y REQUISITOS	25
2.1 Consideraciones generales del negocio.	25
2.1.1 Modelo de Dominio.	25

2.1.2	Modelo de objeto del negocio.....	26
2.1.3	Trabajadores del Negocio	26
2.2	Especificaciones del sistema.	27
2.2.1	Actores del sistema a automatizar.	27
2.2.2	Requisitos funcionales	28
2.2.3	Requisitos no funcionales	30
2.3	Diagrama de casos de uso del sistema SDEffort-BPMN	31
2.4	Caso de uso significativo	31
2.4.1	Formatos interpretables para el caso de uso Interpretar MPN.....	32
2.4.2	Especificación del Caso de Uso “Interpretar MPN”	33
2.5	Planificación basada en Puntos de Casos de Uso. SDEffort-BPMN	36
2.5.1	Cálculo de los Puntos de Caso de Uso sin ajustar.	36
2.5.2	Estimación del Esfuerzo de Desarrollo.....	40
2.5.3	Estimación del esfuerzo del proyecto	41
1.	Costo de desarrollo del producto.	41
2.6	Conclusiones del capítulo	42
CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA		43
3.1	ARQUITECTURA DEL SISTEMA SDEFFORT-BPMN.	43
3.2	DIAGRAMA DE CLASES DE DISEÑO SDEFFORT-BPMN.....	46
3.3	Principios de diseño SDEffort-BPMN.....	48
3.4	Diagrama de secuencia Caso de Uso “Interpretar MPN”	51
3.5	Diseño de la base de datos SDEffort-BPMN.	51
3.5.1	Modelo lógico de datos.....	52
3.5.2	Modelo físico de datos	52
3.6	Diagrama de componentes integrado al de despliegue SDEffort-BPMN.....	53
3.7	Conclusiones del capítulo	55
CAPÍTULO 4. MODELO DE ESTIMACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN. 56		
4.1	Cálculo del esfuerzo utilizando Regresión Lineal Múltiple.....	56
4.1.1	Base de casos	56
4.1.2	Regresión Lineal Múltiple.	58
4.1.3	Análisis del modelo predictor con Regresión Lineal Múltiple.	60
4.2	Pruebas de Software	63

4.2.1	Planificación de Pruebas para el software SDEffort-BPMN	64
4.2.1.1	Resumen de pruebas de caja blanca realizadas	65
4.2.1.2	Resumen de pruebas de caja negra realizadas.....	66
4.3	Conclusiones del capítulo.	73
CONCLUSIONES		74
REFERENCIAS BIBLIOGRÁFICAS		75
ANEXOS		78
Anexo I. Manual de Usuario.....		78
Anexo II. Resultados de la obtención del modelo de predicción del esfuerzo.		88

ÍNDICE DE FIGURAS

FIGURA 1. FORMATOS DE SALIDA COMUNES EN LAS HERRAMIENTAS DE MODELADO DE PROCESOS DE NEGOCIO DESCRITAS.	17
FIGURA 2. ALGORITMO PARA EL CÁLCULO DE COMPLEJIDAD DE AUTOMATIZACIÓN EN PROCESOS DE NEGOCIO.	19
FIGURA 3: FASES DEL PROCESO UNIFICADO ÁGIL (EDEKI 2013)	22
FIGURA 4. MODELO DE DOMINIO SDEffORT-BPMN.	26
FIGURA 5. MODELO DE OBJETO DEL NEGOCIO SDEffORT-BPMN	26
FIGURA 6. DIAGRAMA DE CASOS DE USO DEL SISTEMA SDEffORT-BPMN.	31
FIGURA 7 TÉCNICA DE ESTIMACIÓN PUNTOS DE CASOS DE USO.	36
FIGURA 8. ESTRUCTURA DEL PATRÓN DE ARQUITECTURA DE SOFTWARE MODELO VISTA CONTROLADOR (MVC)	44
FIGURA 9. PATRÓN DE ARQUITECTURA DE SOFTWARE MODELO VISTA CONTROLADOR (MVC) PARA SDEffORT-BPMN.	45
FIGURA 10. DIAGRAMA DE CLASES DE DISEÑO PARA SDEffORT-BPMN.	48
FIGURA 11. DIAGRAMA DE PERSISTENCIA SDEffORT-BPMN	48
FIGURA 12. DIAGRAMA DE CLASES DEL PATRÓN DE DISEÑO <i>STRATEGY</i>	49
FIGURA 13. IMPLEMENTACIÓN DEL PATRÓN DE DISEÑO <i>STRATEGY</i> EN SDEffORT-BPMN.	50
FIGURA 14. DIAGRAMA DE SECUENCIA CASO DE USO “INTERPRETAR MPN”	51
FIGURA 15. MODELO LÓGICO DE DATOS SDEffORT-BPMN	52
FIGURA 16. MODELO FÍSICO DE DATOS SDEffORT-BPMN	53
FIGURA 17. DIAGRAMA DE COMPONENTES INTEGRADO AL DE DESPLIEGUE SDEffORT-BPMN	54
FIGURA 18. BASE DE CASOS.	57
FIGURA 19. MÉTRICAS CUALITATIVAS Y CUANTITATIVAS.	58
FIGURA 20. FRAGMENTO DE LA TABLA DE CORRELACIÓN	61
FIGURA 21 DESARROLLO DE LA ESTRATEGIA DE PRUEBAS	66
FIGURA 22. TEST CARGAR ARCHIVO.	72
FIGURA 23. TEST ADICIONAR ARCHIVO.	72

ÍNDICE DE TABLA

TABLA 1. HERRAMIENTAS PARA MODELADO DE PROCESOS DE NEGOCIO CON SUS FORMATOS DE SALIDA.....	16
TABLA 2. MÉTRICAS CORRESPONDIENTES AL CONTEO DIRECTO DE ELEMENTOS DEL MODELO. (TOMADA DE GRANADOS HONDARES 2015)	18
TABLA 3. MÉTRICAS GRUPO 2. (TOMADA DE GRANADOS HONDARES 2015).....	18
TABLA 4. TRABAJADORES DEL NEGOCIO SDEffORT-BPMN.....	26
TABLA 5. ACTORES DEL SISTEMA SDEffORT-BPMN	27
TABLA 6. FORMATOS INTERPRETABLES.....	32
TABLA 7. ESPECIFICACIÓN DEL CASO DE USO “INTERPRETAR MPN”	33
TABLA 8 CLASIFICACIÓN DE ACTORES.....	37
TABLA 9 CLASIFICACIÓN DE LOS CASOS DE USO.	37
TABLA 10 CÁLCULO DE LOS FACTORES TÉCNICOS.	38
TABLA 11 CÁLCULO DE LOS FACTORES AMBIENTE	39
TABLA 12. ESTADÍSTICA DE LA BASE DE CASOS.	57
TABLA 13. VARIABLES DE ENTRADA AL MODELO CON SPSS.....	61
TABLA 14 FUNCIONALIDADES EVALUADAS POR MÉTODO CAJA BLANCA	64
TABLA 15 RESUMEN DE PRUEBAS DE CAJA BLANCA REALIZADAS.....	65
TABLA 16 RESUMEN DE PRUEBAS DE CAJA NEGRA REALIZADAS.....	66
TABLA 17. CASO DE PRUEBA PARA EL CASO DE USO “CARGAR ARCHIVO MPN”	67
TABLA 18. RESULTADO DE LA EJECUCIÓN DE LAS CLASES DE ENTRADA.....	67
TABLA 19. CASO DE PRUEBA PARA EL CASO DE USO “CONSULTAR DATOS”	68
TABLA 20. RESULTADO DE LA EJECUCIÓN DE LAS CLASES DE ENTRADA.....	68
TABLA 21. CASO DE PRUEBA PARA EL CASO DE USO “INTERPRETAR MPN”	69
TABLA 22. RESULTADO DE LA EJECUCIÓN DE LAS CLASES DE ENTRADA.....	69
TABLA 23. CASO DE PRUEBA PARA EL CASO DE USO “INTERPRETAR MPN”	70

INTRODUCCIÓN

Una de las actividades más importante en el proceso de desarrollo de proyectos de software es la estimación de elementos como la duración del proyecto, el costo asociado, entre otros; por considerarse la estimación, el intento por determinar cuánto dinero, esfuerzo, recursos y tiempo tomará construir un sistema o producto específico basado en software (Pressman & Ph 2010). En este caso una predicción de cuánto tiempo requerirá y/o los recursos que consumirá el desarrollo y mantenimiento del producto.

En ingeniería de software y gestión de proyectos, las estimaciones se utilizan para desarrollar planes de proyectos, elaborar planificaciones de iteraciones o elaborar presupuestos, y se conoce como una de las diversas actividades contenidas en una de las etapas más importantes en el desarrollo de proyecto, la planificación. Dado que en esta etapa se conoce muy poco acerca de la solución que necesita determinado cliente, la estimación del producto a desarrollar se convierte en algo intuitivo (juicio de experto) y riesgoso, a eso debe sumarse que a causa de las presiones de la competencia y demanda del mercado, es necesario dar una propuesta rápida que incluya tiempos y costos del proyecto. (VILLARRAGA & ZAMUDIO 2017).

Entre de las predicciones más significativas en el contexto actual, se encuentra el cálculo del esfuerzo de desarrollo. Las empresas estiman los esfuerzos en términos de requisitos de personal y las horas de trabajo requeridas para producir el software, esfuerzo requerido por persona/mes, normalmente expresado en términos de Horas/Hombre (HH). Para la estimación de esfuerzos se debe conocer el tamaño del software. Esto lo pueden aportar la experiencia misma de los directores, los datos históricos de la organización, o el tamaño del software se puede convertir en esfuerzos usando alguna formulación estándar.

A lo largo del tiempo, se han propuesto diferentes métodos para estimar el esfuerzo que requiere el desarrollo de proyectos de software. Se menciona, por ejemplo: estimación de

software por juicio de experto, estimación de software por analogía, estimación de software por descomposición, estimación de software por medio de modelos de estimación, entre otros (Pressman & Ph 2010).

Investigaciones recientes han demostrado el uso de diferentes técnicas de Inteligencia Artificial como métodos para estimar el esfuerzo de proyectos de software. En (Ferreira, G., Gálvez, D., Quintero, L. y Antón 2015), (ANTÚNEZ BARBOSA et al. 2016) y (Remón 2017) se analiza por ejemplo: Razonamiento Basado en Casos (RBC), Algoritmos Genéticos (AG), Programación Genética (PG), *Support Vector Regression* (SVR), Redes Neuronales Artificiales (RNA), Sistemas de Inferencia Borrosos (SIB), Lógica Difusa (LD) y Árboles de Decisión (AD) como técnicas viables para obtener el esfuerzo de desarrollo.

Los métodos antes mencionados basan su definición en elementos adquiridos en etapas avanzadas del desarrollo del software, en el mundo actual empresarial se utilizan para la estimación: Líneas de Código Fuente (LOC) con lo que se calcula la complejidad del software dependiendo del tamaño del mismo; Puntos Función (PF), medida de software indirecta que se calcula usando conteos del número de pantallas (en la interfaz de usuario), reportes y componentes que probablemente se requieran para construir la aplicación (Pressman & Ph 2010) y Puntos de Casos de Uso donde se miden características de los Casos de Uso y junto a datos históricos se calcula el esfuerzo requerido para desarrollar el sistema. Sin embargo, los elementos asociados a estas técnicas están presentes en etapas avanzadas del desarrollo del software, por lo que conduce a pronosticar valoraciones tardías.

Por la importancia que tiene estimar elementos como la duración, esfuerzo, recursos, etc. en el desarrollo de proyecto de software, uno de los aspectos fundamentales consiste en estimarlos en etapas bien tempranas de este. En la actualidad, el estudio de los procesos de negocio ha cobrado un auge en el perfeccionamiento de los mismos, las empresas prestan especial atención a ellos, se modelan, se perfeccionan, se automatizan y sirven como base a la creación de aplicaciones informáticas.

En la tesis de maestría “Modelo para estimar el esfuerzo que demanda la automatización de Procesos de Negocio” (Granados Hondares 2015), se definió un modelo predictivo del esfuerzo, basado en valores extraídos de modelos de procesos de negocio que fueron la

entrada a proyectos estudiantiles de prácticas profesionales y trabajos de diploma realizados en la Universidad Central “Marta Abreu” de Las Villas.

A pesar de los esfuerzos por hacer estimaciones certeras en el desarrollo de proyectos de software, según (Standish & Report n.d.) en el año 2017 el índice de éxito de los proyectos fue sólo 16.2% para software completado a tiempo y en el presupuesto estimados, mientras proyectos puestos en duda o considerados fue de 52.7%, y cancelados el 31.1%. Dentro de las causas de fracaso de los proyectos se encuentra la falta de planificación, que representa el 8.1%. Los resultados indican que el 52.7% de proyectos cuestan 189% más de sus estimaciones originales.

La empresa de desarrollo de software cubana no ha estado exenta de esta situación global, se pactan plazos de entregas y fechas finales basados en experiencias documentadas de proyectos anteriores y en algunos, fechas dictadas por organismos superiores. Esto ha traído como consecuencias las siguientes:

- ✓ Incumplimiento de los hitos pactados y cronograma general. Afecta la planificación de la entidad y el prestigio de la misma.
- ✓ Problemas legales por incumplimientos de contratos.
- ✓ Reducción de las utilidades. La empresa asume los plazos extendidos de su presupuesto y no lo recupera con el cobro de los hitos.
- ✓ Sobrecarga del personal. El conjunto de personas es el mismo para comenzar otra tarea planificada al terminar el proyecto actual, lo cual retrasa el proyecto futuro o las personas asumen mayor responsabilidad y carga laboral.

Contar con una estimación certera del esfuerzo de desarrollo de proyectos de software en etapas tempranas, contribuye a que se reduzca la subjetividad y asegura que la entrega del producto sea en el tiempo estipulado, el uso del presupuesto asignado sea eficiente y que cumpla con las funcionalidades requeridas. De ahí que se plantee como problema científico la siguiente interrogante:

¿Cómo estimar el esfuerzo en horas/hombre que demanda el desarrollo de proyectos de software, en etapas tempranas de su ciclo de vida?

Por lo planteado anteriormente se propone el siguiente objetivo general:

Desarrollar una herramienta informática que permita calcular el esfuerzo de desarrollo de proyectos de software en horas/hombre, tomando como base modelos de procesos de negocio.

Del objetivo general se desglosan los siguientes objetivos específicos:

1. Diseñar la herramienta SDEffort-BPMN utilizando estándares de diseño adecuados.
2. Obtener el modelo para estimar el esfuerzo de desarrollo basado en la técnica de regresión lineal múltiple.
3. Implementar la herramienta incorporando el modelo de estimación obtenido.

Organización del informe

El informe de la investigación se estructurará en introducción, capitulario, conclusiones, referencias bibliográficas y anexos.

Capítulo 1 Fundamentación Teórica: se estudian las diferentes técnicas de estimación en proyectos de software e investigaciones de interés para la presente propuesta de tesis, así como los principales conceptos que al tema se refieren.

Capítulo 2 Modelo de negocio y requisitos: se realizan consideraciones generales del negocio, así como especificaciones del sistema, requisitos funcionales, y no funcionales con que debe contar la aplicación, diagrama de casos de uso y paquetes del sistema.

Capítulo 3 Descripción de la solución propuesta: se describe la arquitectura del sistema y se fundamentan los principios de diseño.

Capítulo 4 Modelo de estimación y validación de la propuesta de solución: se especifican características del modelo de estimación de esfuerzo implementado en el software y se realizan pruebas de software como las pruebas de unidad e integridad para garantizar el correcto funcionamiento de la herramienta implementada.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace referencia a diferentes métodos y técnicas que en la actualidad son utilizados para realizar estimaciones en proyectos de software, así como herramientas utilizadas con este fin. Se analizan los procesos de negocio como contenedores de información valiosa para la estimación del esfuerzo, teniendo como premisa que esta información es obtenida en etapas muy tempranas del desarrollo de software. Tomando como base los procesos de negocio se presenta una trayectoria de la evolución que hay tenido este tema en investigaciones precedentes y las herramientas que constituyen antecedentes importantes a la que es objeto en la presente investigación.

1.1 Estimación del esfuerzo de desarrollo de software.

Uno de los factores más importantes en la planificación del proyecto es lograr que la estimación de la duración de las tareas sea lo más realista posible. Los investigadores del tema han presentado, acorde al contexto actual de la época, y la evolución de las tecnologías, diferentes tipos de métodos para la estimación de tiempo y costos de desarrollo de un producto de software.

1.1.1 Métodos para la estimación de esfuerzo en proyectos de software.

Estimación de software por juicio de experto: consiste en entregar la información de levantamiento de requisitos de software (por ejemplo, las minutas de reunión o documento de especificación de requisitos de software) a uno o varios conocedores del desarrollo de software y del área de negocio que se dispone representar en el nuevo sistema. (VILLARRAGA & ZAMUDIO 2017)

Estimación de software por analogía: consiste en comparar el desarrollo de software propuesto con proyectos previos similares. Dicha analogía se basa en experiencias que

están documentadas, por lo cual esta se basa en números documentados, lo que puede funcionar razonablemente bien si el proyecto actual es muy similar a esfuerzos anteriores y otros factores que influyen en el proyecto (por ejemplo, el cliente, condiciones del negocio, entorno de ingeniería de software, fechas límite) son aproximadamente equivalentes. Desafortunadamente, la experiencia pasada no siempre es buen indicador de resultados futuros ya que si existe mucha variación de las tecnologías y funcionalidades de un proyecto a otro será más difícil establecer estimaciones confiables. (VILLARRAGA & ZAMUDIO 2017)

Estimación de software por descomposición: Las técnicas de descomposición tienen un enfoque de “divide y vencerás” para la estimación del proyecto. Al descomponer un proyecto en funciones principales y actividades de ingeniería de software relacionadas, la estimación de costo y esfuerzo puede realizarse en forma escalonada. El enfoque de descomposición se analiza en (Pressman & Ph 2010), desde dos puntos de vista diferentes: descomposición del problema y descomposición del proceso. La estimación usa una o ambas formas de división.

- ✓ **Estimación basada en problema:** se basa en la descomposición del producto en funciones y estimar el tamaño del software. Los proyectos deben agruparse por tamaño de equipo, área de aplicación, complejidad y otros parámetros relevantes. Luego se calculan los promedios de dominio local. Se consideran dos tamaños de software líneas de código (LOC) y los puntos de función (PF). Cuando se usa LOC como la variable de estimación, la descomposición es absolutamente esencial y con frecuencia lleva a considerables niveles de detalle. Mientras mayor sea el grado de partición, es más probable que puedan desarrollarse estimaciones de LOC razonablemente precisas. Para estimaciones PF, se estima cada una de las características del dominio de información (entradas, salidas, archivos de datos, consultas e interfaces externas). Entonces las estimaciones resultantes pueden usarse para inferir un valor PF que pueda ligarse a datos pasados y usarse para generar una estimación. Este tipo de estimación puede basarse en datos históricos y experiencias o intuición, para estimar un valor de tamaño optimista, más probable y pesimista para cada función o conteo para cada valor de dominio de información.

- ✓ **Estimación basada en proceso:** es la técnica más común para estimar un proyecto. Utilizando el proceso se identifica un conjunto pequeño de actividades de trabajo o tareas de trabajo y se estima el esfuerzo requerido para llevar a cabo cada tarea. comienza con un delineado de las funciones de software obtenidas del ámbito del proyecto. Para cada función debe realizarse una serie de actividades de marco conceptual. Una vez fusionadas las funciones del problema y las actividades de proceso, se estima el esfuerzo (por ejemplo, persona-mes) que se requerirá para lograr cada actividad de proceso de software para cada función del software.

Estimación de software por modelos empíricos: Los modelos de estimación empírica pueden usarse para complementar las técnicas de descomposición y ofrecer un enfoque de estimación potencialmente valioso por derecho propio. Un modelo de estimación para software de computadora usa fórmulas empíricamente inferidas para predecir el esfuerzo como una función de LOC o PF. El modelo debe probarse aplicando los datos recopilados de los proyectos completados, alimentando los datos en el modelo y luego comparando los resultados reales con los predichos. Si la concordancia es pobre, el modelo debe afinarse y volverse a probar antes de poder usarse. (Pressman & Ph 2010) Se caracterizan los modelos COCOMO, COCOMOII, Puntos de función y Caso de Uso.

Puntos Función (1970 Allan J. Albrecht): Se define como la medición del tamaño del software mediante la cuantificación de la funcionalidad proporcionada para el usuario, basada sólo en diseño lógico y las especificaciones funcionales. Mide la funcionalidad que el usuario solicita y recibe, y permite calcular el desarrollo del software, los tipos y el tamaño de este, independiente de la tecnología a utilizar. Para realizar estimaciones usando este método se deben: Identificar y clasificar los requisitos, determinar el factor de complejidad y obtener los Puntos de función. La metodología de Puntos Función involucra 5 categorías funcionales de un software:

- Entradas: comandos que aceptará el software
- Salidas: tipo de información que el sistema puede generar
- Consultas: diferente tipo de información que el usuario puede solicitar al sistema
- Archivos: cantidad de archivos que se puede acceder al mismo tiempo
- Interfaces: cantidad de enlaces con otros sistemas

Modelo COCOMO 81 (1981 por Barry Boehm): El modelo original COCOMO, es el preferido para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir. Este método ha tenido varias versiones: COCOMO 81, ADA COCOMO, COCOMO II. El Modelo Constructivo de Costos (o COCOMO) es una técnica de estimación matemática con base empírica basada en costos de software, que incluye tres sub modelos:

- Básico, que calcula el esfuerzo y el costo del desarrollo de software, en función del tamaño del software expresado en LOC;
- Intermedio que calcula el esfuerzo del desarrollo de software en función del tamaño y de un conjunto de variables denominadas conductores de costo, que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto;
- Detallado que Incorpora todas las características del modelo Intermedio y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase de desarrollo de software.

Los modelos de COCOMO están definidos para tres tipos de proyectos de software:

- Modo orgánico, son proyectos pequeños y sencillos en los que trabajan equipos reducidos, con experiencia en la aplicación, sobre un conjunto de requerimientos poco rígidos.
- Modo semi acoplado, son proyectos de software intermedios en los que equipos, con variados niveles de experiencia, deben satisfacer requisitos poco o medianamente rígidos.
- Modo empotrado, proyectos que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringidas.

Estimación con casos de uso (1993 por Gustav Karner): Utiliza los actores y casos de uso para calcular el esfuerzo que significa desarrollarlos. Se establece el nivel dentro de la jerarquía estructural, se determina la longitud promedio (en páginas) de cada caso de uso, se define el tipo de software (por ejemplo, tiempo real, empresarial, ingeniería/científico, web, incrustado) y se considera una arquitectura burda para el sistema. Una vez establecidas dichas características pueden usarse datos empíricos para establecer el número estimado de LOC o PF por caso de uso (por cada nivel de la jerarquía). Consta de cuatro

etapas en las que se desarrollan los siguientes cálculos: Factor de peso de los actores sin ajustar (UAW), Factor de peso de los casos de uso sin ajustar (UUCW), Puntos de casos de uso ajustados (UCP) y esfuerzo horas/hombre.

Modelo COCOMO II (1997 por Barry Boehm): Es el más actualizado y cada año se le agregan mejoras. Está formado por tres modelos: el modelo de Composición de Aplicación, indicado para proyectos contruidos con herramientas modernas de generación de interfaces gráficas para usuario; generalmente es utilizado en la etapa de prototipado; el modelo de Diseño Anticipado, se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto, en estas etapas, al carecer de información, se utiliza de Puntos Función para estimar tamaño y el uso de un número reducido de factores de costo, este modelo aplica un ajuste sobre el esfuerzo nominal utilizando siete factores de costo; y el modelo Post-Arquitectura, adaptándose al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, este modelo utiliza: Puntos Función y/o Líneas de Código Fuente para estimar tamaño. Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, entre otros, que tienen injerencia en los costos, cinco factores que determinan un exponente, que incorpora al modelo el concepto de deseconomía y economía de escala. Reemplazando los modos Orgánico, Semiacoplado y Empotrado del modelo COCOMO 81.

1.1.2 Técnicas de inteligencia artificial para la estimación de esfuerzo en proyectos de software.

Los modelos basados en técnicas de Inteligencia Artificial, han sido utilizados para mejorar la precisión de las estimaciones. Estos modelos se fundamentan en el uso de datos recogidos en proyectos anteriores en los que se realizaran estimaciones y aplicación de diferentes técnicas de extracción de conocimiento, con el objetivo de realizar estimaciones de manera más eficiente, eficaz y, si fuera posible, con mayor precisión. En (Ferreira, G., Gálvez, D., Quintero, L. y Antón 2015) se realiza un análisis de las diferentes técnicas de Inteligencia Artificial utilizadas para la estimación de esfuerzo en proyectos de software.

Razonamiento Basado en Casos (RBC): principio fundamental es el razonamiento por analogía, y un proceso cíclico compuesto fundamentalmente por cuatro estados:

1. Recuperar el proyecto más similar (caso).

2. Reusar este proyecto para intentar resolver el problema.
3. Revisar la solución sugerida si es necesario.
4. Retener la solución y el nuevo problema como un nuevo proyecto.

Evita problemas asociados con la adquisición y codificación de conocimiento, soporta mejor la colaboración con los usuarios, hace frente a dominios pobremente ya que las soluciones están basadas en lo que realmente pasó al contrario de modelos hipotéticos.

Redes Neuronales Artificiales (RNA): tienen la habilidad de modelar complejas relaciones no lineales y características que las hacen atractivas para resolver tareas de reconocimiento de patrones sin tener que construir un modelo explícito del sistema. En un sentido amplio, la red neuronal es un modelo debido a que la topología de la red, así como las funciones de transferencia de los nodos son usualmente formuladas para coincidir con el problema actual. Pueden conducir a desarrollar modelos que pueden ser precisos a través de un ejemplo dado, pero fallan cuando las condiciones cambian. Carecen de la capacidad de explicación o justificación de la estimación realizada y no proveen un ambiente para una adaptación directa por el usuario de los datos, es decir una re-calibración del modelo. Para poder incorporar un nuevo caso al conjunto de casos válidos del propio modelo, necesitan ser reentrenadas, generando un modelo nuevo.

Árboles de decisión (AD). AD es una técnica que construye un modelo en forma de árbol, dividiendo recursivamente el conjunto de datos hasta que un criterio de parada es satisfecho. Esta división se realiza con el propósito de alcanzar la máxima homogeneidad posible, relativa a la variable de salida o dependiente, entre los ejemplos que alcanzan el nodo. Todos los nodos en el árbol, menos los terminales, especifican una condición basada en una de las variables que tienen influencia en la variable dependiente. Luego de que el árbol es generado, el mismo se puede usar para realizar predicciones siguiendo un camino a través del árbol de acuerdo con los valores específicos de las variables del nuevo caso.

Redes bayesianas (RB): es una técnica de modelado para tratar interrelaciones causales basadas en inferencias bayesianas. La red es representada como un grafo acíclico con nodos para cada variable imprecisa y aristas dirigidas para cada interrelación entre las variables. Por cada nodo o variable existe una tabla de probabilidades del nodo. Las tablas definen las interrelaciones y las imprecisiones de las variables y sus valores usualmente son discretos con un número fijo de estados. Para cada estado, la tabla muestra la probabilidad que tiene

la variable. Si existen nodos padres; por ejemplo, un nodo que influye sobre un nodo actual, las probabilidades se definen en dependencia del estado de los nodos padres. El proceso de construcción de una red bayesiana sugiere una serie de pasos importantes que no son triviales:

1. Identificación de las principales variables que serán modeladas, representándolas como nodos.
2. Construcción de la topología de la red.
3. Definición de la tabla de probabilidades del nodo.

Support Vector Regression (SVR): El algoritmo *Support Vector (SV)* es una generalización no lineal del algoritmo *Generalized Portrait* desarrollado en Rusia en los años sesenta. En su forma actual fue largamente desarrollado en AT&T Bell Laboratories en la década del 90 del pasado siglo. En poco tiempo los clasificadores SV se volvieron competitivos con los mejores sistemas de Reconocimiento Óptico de Caracteres (*Optical Character Recognition - OCR*), pero también en regresión, tomando el nombre de SVR, fueron obtenidos excelentes resultados rápidamente.

Algoritmos genéticos (AG): fueron desarrollados como una técnica alternativa, basada en la teoría de la selección natural, para afrontar problemas generales de optimización con largos espacios de búsqueda. Ellos tienen la ventaja de que no necesitan ningún conocimiento anterior, opinión de un experto o lógica relacionada con el problema en particular a resolver. Los AG ocasionalmente pueden producir la solución óptima, pero para la mayoría de los problemas con grandes espacios de búsqueda, una buena aproximación al óptimo es la salida más probable. El proceso básico de los AG es el siguiente:

1. Generación aleatoria de una familia de soluciones.
2. Creación de una nueva familia de soluciones partiendo de la previa, aplicando los operadores genéticos a los cromosomas o pares de cromosomas más adecuados de la población anterior.
3. Repetición del paso 2 hasta que cualquiera de los cromosomas converja a la solución, o cuando se haya producido un número de iteraciones anteriormente especificado.

La mejor solución de la última población es tomada como la mejor aproximación al óptimo del problema en cuestión. El proceso completo normalmente se repite un número de veces

con diferentes poblaciones iniciales aleatorias y se escoge la mejor de las soluciones obtenidas.

Programación genética (PG): es una extensión de los AG que elimina la restricción de que un cromosoma sea una cadena binaria de tamaño restringido. Generalmente en PG el cromosoma es un tipo de programa que es ejecutado para obtener los resultados requeridos. Una forma simple de estos programas es un árbol binario que contiene operando y operadores, es decir cada solución es una expresión algebraica que puede ser evaluada.

1.2 Herramientas para estimar el esfuerzo de desarrollo.

El objetivo de las herramientas de estimación de esfuerzo y costo es brindar a un equipo de proyecto estimaciones del esfuerzo requerido, duración del proyecto y costo, de manera que aborde las características específicas del proyecto a mano y el entorno donde se construirá el proyecto. En general, las herramientas de estimación de costo usan una base de datos histórica inferida de proyectos locales y datos recopilados a través de la industria, y un modelo empírico (por ejemplo, COCOMO II) que se utiliza para derivar estimaciones de esfuerzo, duración y costo. Las características del proyecto y el entorno de desarrollo son entradas y la herramienta proporciona un rango de salidas de estimación.

Herramientas representativas(Pressman & Ph 2010):

- Costar, desarrollada por Softstar Systems (www.softstarsystems.com), usa el modelo COCOMO II para desarrollar estimaciones de software.
- CostXpert, desarrollada por Cost Xpert Group, Inc. (www.costxpert.com) integra múltiples modelos de estimación y una base de datos histórica de proyectos.
- Estimate Professional, desarrollada por el Software Productivity Centre, Inc. (www.spc.com), se basa en el COCOMO II y en el modelo SLIM.
- Knowledge Plan, desarrollado por Software Productivity Research (www.spr.com), usa entrada de puntos de función como el principal controlador para un paquete de estimación completo.
- Price S, desarrollada por Price Systems (www.pricesystems.com), es una de las herramientas de estimación más antiguas y de más uso para proyectos de desarrollo de software a gran escala.

- SEER/SEM, desarrollada por Galorath, Inc. (www.galorath.com), proporciona una amplia capacidad de estimación, análisis de sensibilidad, valoración de riesgo y otras características.
- SLIM-Estimate, desarrollada por QSM (www.qsm.com), se apoya en una exhaustiva “base de conocimiento industrial” para proporcionar una “comprobación de sanidad” para estimaciones que se infieren usando datos locales.

1.3 Los procesos de negocio como punto de partida para estimar el esfuerzo de desarrollo en proyectos de software.

En los proyectos de software, la actividad de planeación incluye un conjunto de prácticas administrativas y técnicas que permiten que el equipo de desarrollo defina un mapa mientras avanza hacia su meta estratégica y sus objetivos tácticos (Pressman & Ph 2010). La planificación de proyectos de software abarca cinco grandes actividades: **estimación**, calendarización, análisis de riesgos, planificación de gestión de la calidad y planificación de gestión del cambio. Su objetivo es proporcionar un marco conceptual que permita hacer estimaciones razonables, que deben intentar definir los escenarios de mejor caso y peor caso, de modo que los resultados del proyecto puedan acotarse. En la actualidad predecir el esfuerzo que demandará el desarrollo de un producto constituye una prioridad en el mundo empresarial, y son diversos los enfoques que existen. De manera general los métodos para estimar esfuerzo, requieren información con la que no se cuenta en la etapa de inicio o conceptualización del proyecto. En esta etapa es común utilizar los procesos de negocio de las empresas para las cuales se desarrolla el producto, con el fin de entender el funcionamiento de las mismas y descubrir las funcionalidades que implementará el futuro sistema, de ahí que el desarrollo basado en procesos de negocio sea hoy una solución prometedora.

La gestión de procesos de negocio, se constituye en un conjunto de actividades, cuyo propósito es apoyar el diseño, la administración, la configuración, y el análisis de procesos de negocio. Es por ello que, con el paso del tiempo, se han investigado y aportado taxonomía, reglas, e incluso herramientas que propician una mayor calidad de modelación, lo que pudiera precisar cuán complejo de automatizar pudiera ser un proyecto, sin embargo, esto no ayuda en la determinación de cuánto tiempo puede tardar la realización del mismo.

1.3.1 Caracterización de los Modelos de Proceso de Negocio

Un modelo de proceso de negocio es una representación abstracta (gráfica) de los procesos de una empresa u organización, que muestra principalmente cómo y por quien son realizadas las actividades que generan valor para la organización. A través de estos se pueden apreciar los actores involucrados en el proceso, las actividades operativas distinguibles, cuales son ejecutables y por quien, sus entradas y salidas y la secuencia en que ocurren; los recursos que son consumidos y los eventos que dirigen el proceso.(Bandomo Toledo 2014) Para el modelado existen notaciones y metodologías que buscan crear un vínculo cercano entre la realidad del negocio y lo que se pretende con el producto de software.

Business Process Management (BPM) es una metodología integrada para diseñar, dirigir y mejorar procesos de una empresa, en aras de aumentar su productividad. BPM comprende la automatización de procesos de la empresa y la integración de aplicaciones, permite el diseño, la ejecución, el monitoreo y el análisis de los procesos, y la integración con sistemas externos. Puede integrar aplicaciones existentes y apoyar el cambio de los procesos. BPM es una colaboración entre personas de negocio y personal de Tecnologías de Información para fomentar procesos de negocio efectivos, ágiles y transparentes(Giraldo et al. 2017).

Business Process Model and Notation (BPMN) es una notación entendida por todos los usuarios del negocio, desde el que crea los borradores iniciales de los procesos, hasta los desarrolladores técnicos responsables de poner en práctica la tecnología que plasmará esos procesos, y finalmente al agente comercial que manejará y supervisará dichos procesos. BPMN crea un puente estandarizado para el abismo entre el diseño de proceso comercial y los procesos de implementación en sistemas de software. En 2007 la sigla BPMN cambia de *Business Process Modeling Notation* a *Business Process Model and Notation*, ahora siendo la versión BPMN 2.0. La adopción de esta especificación mejora la capacidad los analistas de negocio para desarrollar, comunicar y entender los modelos de procesos de negocios(Giraldo et al. 2017).

1.3.2 Herramientas para el modelado de procesos de negocio.

Los procesos de negocio, pueden ser modelados en diferentes herramientas, lo que trae consigo un sinnúmero de formatos de salida. Las herramientas que se analizan a

continuación son algunas de las más relevantes en el mercado para la gestión de los procesos, todas fueron probadas y analizadas para la investigación.

BizAgi v2.7.0.2: posee la facilidad para efectuar modelamientos, ya que no restringe las conexiones y posee los diferentes tipos de tareas requeridos, diferentes tipos de eventos y de puertas de enlaces. Es un software gratuito, fácil de instalar. Tiene la posibilidad de ejecutar procesos, permitiendo publicar como modelos de información (PDF), modelo de información (Word), *Web*, *Sharepoint* y *Wiki*. Ofrece módulos de Diagramas BPMN. Tiene posibilidad de validar diagramas.

Bonita Open Solution 7.1.3: contiene diferentes tipos de tareas, eventos y *gateways*, admite agregar texto en las conexiones, haciendo más entendible el diagrama. Es un software gratuito. Para el proceso de instalación es necesario tener previamente tener *Java Runtime Environment* (JRE) y *Java Development Kit* (JDK). La herramienta se puede encontrar en la página: <http://www.bonitasoft.com/>. Ofrece módulos que combina tres soluciones en una: un estudio innovador para procesos de modelación, un proceso motor de BPM y *workflow*, y un avance de usuario de interface. Cuenta con mensajes de error y advertencias que aparecen cuando una tarea no es configurada apropiadamente o falta algún dato.

HEFLO: Es gratuita para modelado y documentación, en la nube, sin instalación o mantenimiento. Es muy rápida y fácil de usar. Usa la notación BPMN 2.0. Todos los diagramas de procesos se almacenan en la nube lo que es muy importante porque los usuarios pueden acceder desde cualquier lugar sus modelos se exportan como imagen y presenta control de versiones como Git, Subversion entre otros. Se accede vía web mediante la dirección: <http://app.heflo.com>. Integra al equipo de trabajo, accediendo a los trabajos mediante la nube, con cualquier dispositivo y es capaz de validar los diagramas.

Intalio BPMS 7. (2013): Tiene la capacidad para nombrar las conexiones y efectuar modificaciones de tamaños en los elementos. Su principal desventaja es la forma de usar los conectores, ya que está limitado a solo hacerse de izquierda a derecha. Es una herramienta gratuita, aunque se requiere de un registro en la página previo a la descarga: <http://www.intaliodesigner.software.informer.com/>. Ofrece módulos como creación e integración de formularios AJAX, la gestión de flujo de trabajo y la supervisión de procesos y gestión de la interacción humana con el marco de trabajo de administración

extensible. Tiene la posibilidad de validar diagramas con reglas lógicas, visuales y de ejecución de procesos.

Enterprise Architect 13.5: no es una herramienta de uso trivial en cuanto al manejo del pool, los lanes, las tareas y de los conectores. Es necesario que el modelador tenga un buen manejo de la herramienta ya que de otro modo puede tardar gran cantidad de tiempo en la diagramación. No es un software gratuito, pero existe la edición RTM y Retail. Ofrece módulos como modelos de desarrollo de software, diagramas de procesos de negocio, mapas conceptuales. Modelos y diagramas generales bajo UML.

Signavio Process Editor 7.0: para trabajar con Signavio se hace online con una versión gratuita de 30 días. Su principal desventaja es la ausencia de eventos intermedios y de diferentes tipos de tareas, haciendo que la lectura del diagrama por parte del usuario se torne algo confusa. Se encuentra en <http://www.signavio.com/es>. Ofrece módulos como *QuickModel*, BPMN 2.0 procesos de modelación, simulación de procesos, modelo de repositorios, funciones de colaboración y publicación de procesos de documentación. Cuenta con un validador de diagramas el cual es mostrado a la hora de guardar el modelo.

1.3.3 Formatos de salida de las herramientas de modelado de procesos de negocio.

Tabla 1. Herramientas para modelado de procesos de negocio con sus formatos de salida.

Herramienta	Licenciamiento	Formato de salida y exportación.	Capacidad de validación.	BPMN 2.0
BizAgi	Gratuito	bpm(BizAgi), XPD, Visio, PNG, JPG, BMP, SVG	Si	Si
Bonita Open Solution	Gratuito	bos , BPMN 2.0, PDF, JPG, JPEG, PNG, BMP, GIF y SVG	Si	Si
HEFLO	Gratuito(web)	BPMN 2.0	Si	Si
Intalio	Gratuita(requiere registro en la página oficial)	XML, herramientas SOA, bpm herramientas Intalio y Java Script	Si	Si
Signavio Process Editor 7.0	Gratuito	SGX, BPMN2.0, XML, PNG, SVG y PDF	Si	Si
Enterprise Architect 13.5	No Gratuito	xml y csv	Si	Si

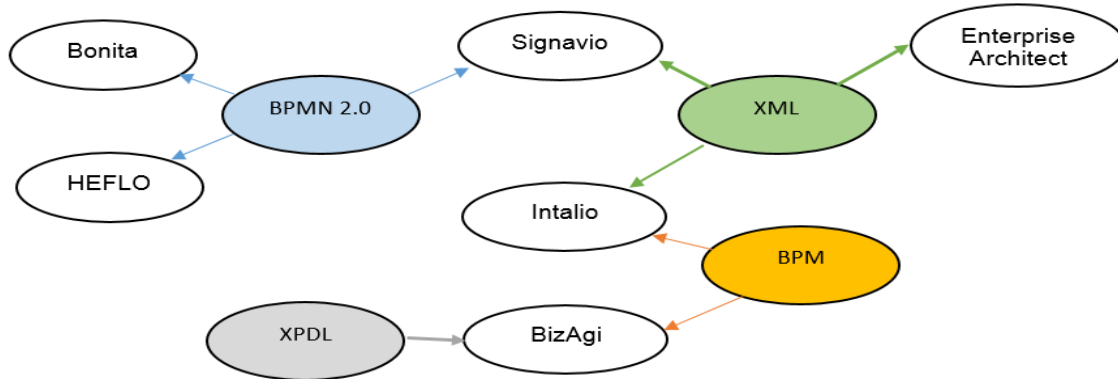


Figura 1. Formatos de salida comunes en las herramientas de modelado de procesos de negocio descritas.

Cada una de estas herramientas, aunque poseen peculiaridades, se basan en representar de forma lo más homogénea y real posible los procesos de negocio, usando para ello elementos de la notación BPMN, que se ha convertido en universal para el modelado de procesos de negocios. Esta puede ser la fuente para definir métricas a partir de elementos contenidos en los diagramas de procesos de negocio que sirvan al proceso de estimación y planificación en proyectos de software.

1.3.4 Métricas para la estimación de esfuerzo en modelos de proceso de negocio.

Existe un gran volumen de literatura dedicado a la definición y discusión de numerosas métricas relacionadas con los procesos de negocio, derivadas usualmente de métricas de software. En la tesis de maestría “Modelo para estimar el esfuerzo que demanda la automatización de Procesos de Negocio”(Granados Hondares 2015), tesis tomada como antecedentes para la presente investigación, se proponen como uno de sus objetivos analizar la influencia que tienen diversas métricas asociadas a procesos de negocio en el esfuerzo de desarrollo del mismo y definir un conjunto de métricas.

Las métricas definidas en (Granados Hondares 2015) se agrupan de acuerdo a si corresponden directamente con algún elemento de la notación BPMN o no y se presentan dos grupos de métricas en correspondencia a esta característica. En el primer grupo estarán aquellas métricas que respondan al conteo directo de elementos del modelo y en el segundo grupo, aquellas que no tienen correspondencia directa con elementos del modelo pero que pudieran aportar indicadores de interés.

En la Tabla 2 se presentan las métricas correspondientes al primer grupo (GRUPO I) mencionado anteriormente. Tomando en cuenta que la notación BPMN se compone de

cuatro categorías de elementos centrales, con los cuales es posible definir modelos de procesos de negocio: Objetos de Flujo, Objetos de Conexión, Swimlanes y Artefactos, se identificaron dentro de la categoría de Objetos de Flujo ocho métricas relativas a los elementos de la notación: Evento, Actividad y Nodo. En la categoría de Objetos de Conexión por su parte, se definieron tres métricas referentes al elemento: Flujo de secuencia. En cuanto a Swimlanes está compuesto por dos métricas relativas a Carriles y Participantes. Por último, en la categoría Artefactos, se identificó una métrica referente a los Objetos de Dato.

Tabla 2. Métricas correspondientes al conteo directo de elementos del modelo. (Tomada de Granados Hondares 2015)

<i>Elemento BPMN</i>	<i>Clasificación</i>	<i>Métrica</i>	<i>Definición de la Métrica</i>
Evento	Evento de inicio	NTEI	Número total de eventos de inicio
	Evento Intermedio	NTEIn	Número Total de Eventos Intermedios
	Evento final	NTEF	Número Total de Eventos Fin
	Total de Eventos	NTE	NTEI+ NTEIn+ NTEF
Actividad	Tareas	NTT	Número Total de Tareas
	Subprocesos	NTS	Número Total de Subprocesos
	Total de actividades	NTA	NTT+ NTS
Nodo	Nodos Decisión/Unión	NTCO	Número total de nodos decisión/unión
Flujos de secuencia	Entre actividades	NFSA	Número total de flujos de secuencia entre actividades
	Procedentes de un nodo	NFSN	Número total de flujos de secuencia procedentes de un nodo
	Total de flujos	NTFS	Número total de flujos de secuencia
Carriles		NTC	Número total de carriles
Participantes		NTP	Número total de participantes
Objetos de dato		NTOD	Número total de objetos de dato

La Tabla 3 muestra las métricas correspondientes al segundo grupo definido (GRUPO II).

Tabla 3. Métricas grupo 2. (Tomada de Granados Hondares 2015)

<i>Nombre</i>	<i>Métrica</i>	<i>Definición de la Métrica</i>
Nivel de conectividad entre actividades	NCA	NTA/ NFSA
Coefficiente de complejidad	CCOM	NTFS/(NTA+ NTCO)
Complejidad de automatización	CAPN	Descrita más adelante

Algoritmo para el cálculo de CAPN.(Granados Hondares 2015) referenciado de (Bandomo Toledo 2014)

```

Si NTCO <=2 y NTC <=5 y NTOD <=2 y NTP <=1 y NFSA <=5
  Entonces CAPN = “Baja”
Si no
  Si 3< NTC <=5 y NTOD <=2
    Entonces CAPN = “Media”
  Si no
    Si NTC <=3 y NTOD <=3 y NTCO > 3,
      Entonces CAPN = “Media”.
    Si no
      Si NTC >4 y NTOD >1
        Entonces CAPN = “Alta”.
      Fin Si
    Fin Si
  Fin Si
Fin Si

```

Figura 2. Algoritmo para el cálculo de Complejidad de Automatización en Procesos de Negocio.

1.4 Estimación de esfuerzo basado en modelos de procesos de negocio.

Elvira Rolón en su tesis doctoral “Medidas para asegurar la calidad de los procesos de negocio” (Rolón Aguilar 2009) realiza una propuesta clara y abarcadora de métricas debidamente fundamentadas, obtenidas a partir de modelos de procesos de negocio con notación BPMN, la cual incluye un conjunto de experimentos que validan la influencia de estas métricas en la entendibilidad y mantenibilidad de los procesos de negocio, con el fin de evaluar los procesos de negocio desde el punto de vista de su complejidad estructural, en la búsqueda de mejorar la calidad de los modelos de procesos de negocio, así como su mantenibilidad y usabilidad.

Años más tarde (O’Farril Fernández 2012) propone un procedimiento para la industria cubana del software, basado en métricas, que evalúa la complejidad de informatizar los procesos de negocio, cuyas métricas en algunos casos son comunes con las planteadas en (Rolón Aguilar 2009). En dicho procedimiento se realiza la evaluación de la complejidad de un modelo de proceso de negocio de acuerdo a tres criterios, Complejidad Alta, Complejidad Media y Complejidad Baja.

En (Bandomo Toledo 2014) a partir de la investigación de (O’Farril Fernández 2012) y su definición de métricas que inciden directamente en informatización de los procesos de negocio, se realizó una predicción de la complejidad de estos, utilizando técnicas de inteligencia artificial. Esta predicción aportaba información similar al procedimiento

anterior, clasificando la complejidad de informatización atendiendo a los mismos criterios anteriormente especificados.

En la tesis de maestría “Modelo para estimar el esfuerzo que demanda la automatización de Procesos de Negocio” (Granados Hondares 2015) a partir de la utilización de un método empírico, se definió un modelo utilizando Regresión Lineal Simple, obteniendo así una ecuación que permite estimar el esfuerzo de desarrollo en proyectos de software. La investigación se enfoca en la estimación en Modelos de procesos de negocio, utilizando como punto de partida los propios modelos. Se utilizó como base de casos para la investigación, valores extraídos de modelos de procesos de negocio que fueron la entrada a proyectos estudiantiles de prácticas profesionales y trabajos de diploma realizados en la Universidad Central “Marta Abreu” de Las Villas, lo que permitió que fuera ideal para comparar el acierto del modelo propuesto.

Dichas investigaciones dieron como resultado un conjunto de herramientas que manipulan modelos de procesos de negocio.

La herramienta denominada INTERVAR (Esquivel Ariz 2014) fue diseñada e implementada con el objetivo extraer los valores correspondientes a las métricas involucradas en el modelo obtenido en (Bandomo Toledo 2014) y calcular a partir de estas la complejidad de informatización de procesos de negocio, utilizando como base su diagrama o representación gráfica. Dicha herramienta basaba la obtención de las métricas (O’Farrill Fernández 2012) en archivos con formato XPDL, (lenguaje basado en XML, formato estándar de intercambio entre herramientas de modelado de procesos de negocio). De esta forma las métricas eran obtenidas de forma única sin dar margen a errores de interpretación.

Posteriormente en 2015 se rediseñó la herramienta informática realizada en (Esquivel Ariz 2014) manteniendo su propósito en cuanto a la extracción de los elementos de los modelos, pero esta vez calculando el esfuerzo de automatización de los proyectos de software y ampliando sus funcionalidades. Así surgió IEEMPON Sistema para la Interpretación y Estimación del Esfuerzo de Modelos de Procesos de Negocio (Granados Hondares 2015). En esta versión se extendieron los formatos de los archivos a interpretar y el conjunto de métricas se amplió teniendo en cuenta la propuesta de dicha investigación.

1.5 Fundamentación de la metodología utilizada.

Se elige para el cumplimiento de los objetivos propuestos: Proceso Unificado Ágil (AUP) por considerarse dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental. El Proceso Unificado Ágil adopta una filosofía “en serie para lo grande” e “iterativa para lo pequeño” a fin de construir sistemas basados en computadora. Al adoptar las actividades en fase clásicas del Proceso Unificado (PU) —concepción, elaboración, construcción y transición—, el AUP brinda un revestimiento en serie (por ejemplo, una secuencia lineal de actividades de ingeniería de software) que permite que el equipo visualice el flujo general del proceso de un proyecto de software. Sin embargo, dentro de cada actividad, el equipo repite con objeto de alcanzar la agilidad y entregar tan rápido como sea posible incrementos de software significativos a los usuarios finales. Cada iteración del AUP aborda las actividades siguientes:(Edeki 2013)

- **Modelado.** Se crean representaciones de UML de los dominios del negocio y el problema. No obstante, para conservar la agilidad, estos modelos deben ser “sólo suficientemente buenos” para permitir que el equipo avance.
- **Implementación.** Los modelos se traducen a código fuente.
- **Pruebas.** Igual que con la XP, el equipo diseña y ejecuta una serie de pruebas para detectar errores y garantizar que el código fuente cumple sus requerimientos.
- **Despliegue.** Se centra en la entrega de un incremento de software y en la obtención de retroalimentación de los usuarios finales.
- **Configuración y administración del proyecto.** En el contexto del AUP, la administración de la configuración incluye la administración del cambio y el riesgo, y el control de cualesquiera productos del trabajo persistentes que produzca el equipo. La administración del proyecto da seguimiento y controla el avance del equipo y coordina sus actividades.
- **Administración del ambiente.** La administración del ambiente coordina una infraestructura del proceso que incluye estándares, herramientas y otra tecnología de apoyo de la que dispone el equipo.

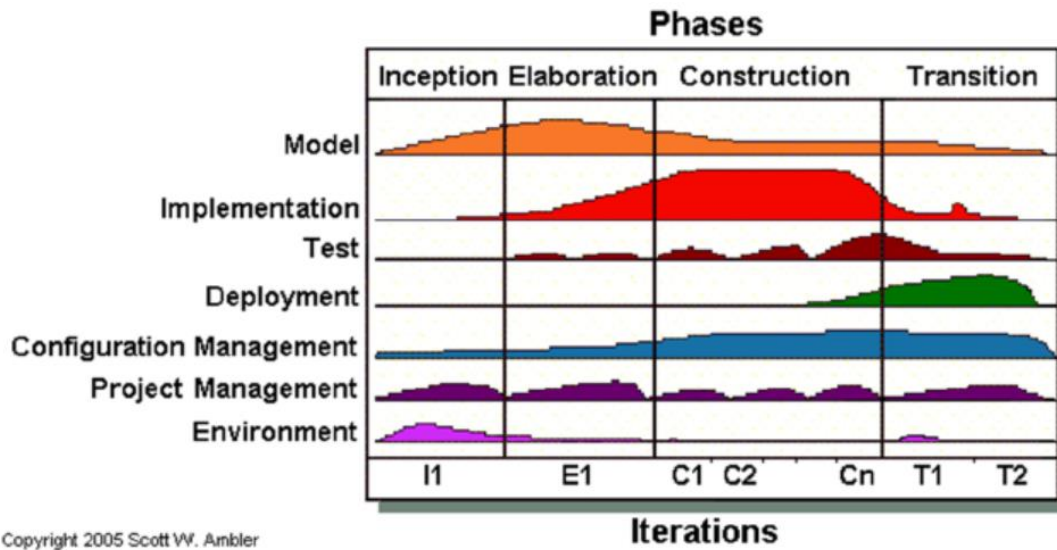


Figura 3: Fases del Proceso Unificado Ágil (Edeki 2013)

1.6 Tecnologías a utilizar.

Para la obtención del modelo predictor de esfuerzo y desarrollo del software es necesario tener en cuenta qué herramientas utilizar para poder dar cumplimiento de manera efectiva a los objetivos planteados. Para una mejor comprensión se categorizan dichas herramientas de la siguiente manera:

- Desarrollo: se seleccionó como lenguaje de programación Java, (Gutiérrez Acea 2018) por ser un lenguaje de alto nivel orientado a Objetos el cual ofrece amplias posibilidades para el desarrollo de aplicaciones y, como Entorno de Desarrollo Integrado (IDE, de sus siglas en inglés) que permite el desarrollo de proyectos en Java, NetBeans fue el seleccionado como ambiente de programación con jdk 1.8.0. y JVM (en inglés Java Virtual Machine) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.
- Modelado: VisualParadigm, (Ferrerías Bustamante 2018) es una herramienta CASE: Ingeniería de Software Asistida por Computación para la modelación de todo el proceso de desarrollo del software y UML como lenguaje de modelado que soporta dicha herramienta. Utiliza un enfoque Orientado a Objetos.

- Gestión de la información: SQLite, (Hernández 2015) ya que no necesita ser instalado para poder ejecutarlo, simplemente se necesita el archivo binario de ejecución, y los comandos de consulta, de inserción, y de creación son exactamente idénticos a los de SQL. Como IDE para manejar las consultas, SQLiteMan porque es portable y driver JDBC para poder conectar Java con SQLite: `sqlite-jdbc-3.8.11.2.jar`.
- Obtención del modelo: se utiliza IBM SPSS, (Patricia et al. 2019) programa para análisis estadístico de datos, que consta de un módulo de base y módulos anexos que se actualizan constantemente con procedimientos estadísticos.
- Pruebas: JUnit, (Design & Institutions 2018) para desarrollo de pruebas al código. Constituye un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. JUnit es también un medio de controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

1.7 Conclusiones del capítulo

Existen diferentes métodos de estimación de proyectos de software, entre los que se encuentran, juicio de expertos, analogías, puntos de función y técnicas de Inteligencia Artificial, entre otros, los cuales utilizan para estimar, métricas aplicables al proceso de desarrollo del software. Según el estudio realizado, los métodos estudiados basan la estimación, en elementos que se obtienen únicamente en etapas avanzadas del proceso de desarrollo de software, lo cual limita una estimación temprana de la duración del mismo, trayendo como consecuencias incumplimiento de contratos.

Los modelos de procesos de negocio, como entrada inicial al proceso de desarrollo de software, constituyen una variante novedosa para lograr la estimación temprana, mencionada anteriormente. Múltiples y variadas son las herramientas para modelar procesos de negocio, a partir del estudio de los formatos de salidas de algunas de ellas (las más utilizadas en la actualidad), se identificaron elementos que influyen en el cálculo del esfuerzo en proyectos de software.

Se seleccionan tanto la metodología como la tecnología a utilizar en el desarrollo de una solución informática que propone tomar como base los modelos de procesos de negocio para el cálculo del esfuerzo (Horas/Hombre).

CAPÍTULO 2. MODELO DE NEGOCIO Y REQUISITOS

En este capítulo se realizan consideraciones generales del negocio, evidenciando la situación actual del mismo. Se describen los actores del sistema a automatizar, así como especificaciones del sistema, requisitos funcionales, y no funcionales con que debe contar la aplicación, diagrama de casos de uso y paquetes del sistema. Se especifican casos de uso como interpretar y calcular esfuerzo por su importancia.

2.1 Consideraciones generales del negocio.

En la etapa de la identificación de la problemática y conformación de pactos cliente-desarrollador, determinar periodos o fechas específicas del producto a desarrollar es fundamental. El personal especializado dentro del equipo de proyecto se encarga de estudiar factores como complejidad y tamaño de proyecto. Teniendo en cuenta que en esta etapa se cuenta solamente con los procesos de negocio que desarrolla la entidad o cliente que demanda en software, el tamaño y la complejidad del software a desarrollar, podría estar dada por la complejidad de dichos procesos. Se definen como actividades dentro del negocio:

1. Analizar los modelos de procesos de negocio.
2. Determinar el esfuerzo que demanda el desarrollo de producto.

2.1.1 Modelo de Dominio.

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, se representa con un conjunto de diagramas de clases en los que no se define ninguna operación (García Peñalvo & García Holgado 2018). La imagen muestra el modelo de dominio donde un Archivo, que representa el diagrama del proceso de negocio, el cual se relaciona directamente con ModeloPN o sea en un

modelo de proceso de negocio, que contiene la información procesada extraída del diagrama. Esta información es procedente de múltiples procesos de negocio, lo cual hace necesario relacionar un conjunto de procesos de negocio modelados en ListadoPN

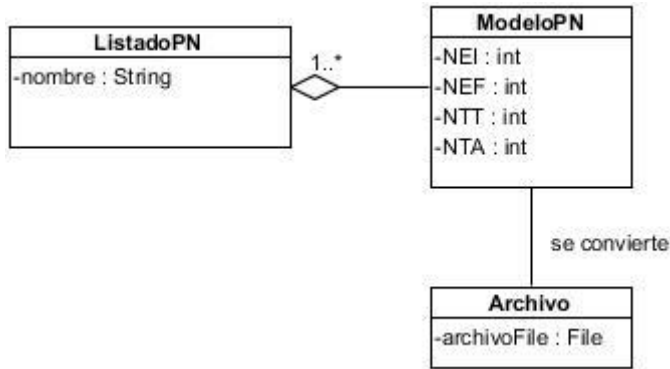


Figura 4. Modelo de dominio SDEffort-BPMN.

2.1.2 Modelo de objeto del negocio

El modelo de objeto del negocio es un diagrama que representa un conjunto de objetos y sus relaciones en un momento concreto. Gráficamente, un diagrama de objetos es una colección de nodos y arcos, (García Peñalvo & García Holgado 2018).

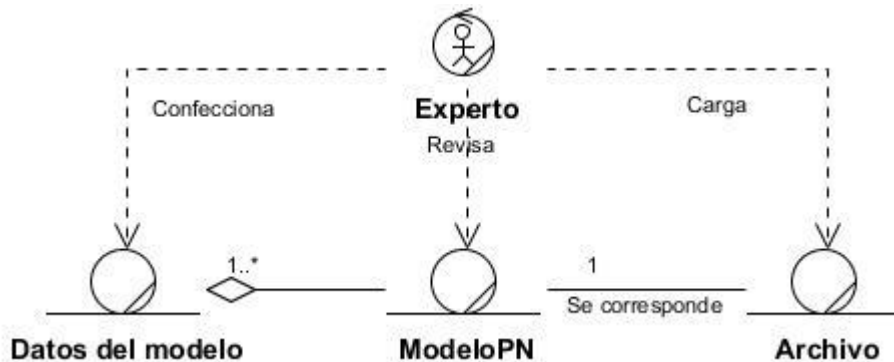


Figura 5. Modelo de objeto del negocio SDEffort-BPMN

2.1.3 Trabajadores del Negocio

Los trabajadores del negocio representan un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades. (E.V.A. 2017)

En la siguiente Tabla se observa el trabajador del negocio actual:

Tabla 4. Trabajadores del Negocio SDEffort-BPMN

Trabajador del Negocio	Descripción
------------------------	-------------

Experto	Es el miembro del proyecto de desarrollo de software encargado de establecer la duración y por ende fecha de terminación y entrega del producto a desarrollar. Dependiendo de las características del equipo de desarrollo, este rol puede ser ocupado por el planificador, el jefe de proyecto, entre otros.
----------------	---

2.2 Especificaciones del sistema.

El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requerimientos. La meta de la ingeniería de requerimientos (IR) es entregar una especificación de requisitos de software correcta y completa (García Peñalvo & García Holgado 2018). Según (Pressman & Ph 2010) la Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software.

2.2.1 Actores del sistema a automatizar.

Se le llama actor a algo externo al sistema en discusión que interactúa con éste, (García Peñalvo & García Holgado 2018).

Tabla 5. Actores del Sistema SDEffort-BPMN

Actor del Sistema	Descripción
Experto	Es el miembro del proyecto de desarrollo de software encargado de establecer la duración y por ende fecha de terminación y entrega del producto a desarrollar. Dependiendo de las características del equipo de desarrollo, este rol puede ser ocupado por el planificador, el jefe de proyecto, entre otros.

Un requisito es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requisitos se utilizan como datos de entrada en la etapa de diseño del producto. Establecen qué debe hacer el sistema, pero no cómo hacerlo.(Pressman & Ph 2010)

2.2.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares y de cómo se comportará en situaciones particulares. Los requisitos funcionales de un sistema describen la funcionalidad o los servicios que se espera que este proveerá.(ROMERO ESPINOZA 2018)

A continuación, se muestra la definición de los mismos:

RF1: “Gestionar Archivo MPN”.

El sistema debe brindar la posibilidad de cargar los archivos para así poder realizar las acciones deseadas.

RF1.1: Cargar Archivo MPN

El sistema debe permitir al usuario cargar archivos con formatos XPDL, BPMN, BPM, XML para su posterior interpretación.

RF1.2: Eliminar Archivo MPN

El sistema debe brindar la posibilidad de eliminar de la lista de archivo a interpretar los archivos deseados.

RF1.3: Interpretar Archivo MPN

El sistema debe brindar la posibilidad de interpretar los archivos cargados y mostrar el esfuerzo estimado.

RF1.4: Obtener esfuerzo

El sistema debe brindar la posibilidad de obtener el esfuerzo por el método que el usuario seleccione.

RF1.5: Obtener por Regresión Lineal Simple

El sistema debe brindar la posibilidad de obtener el esfuerzo por la técnica regresión lineal simple.

RF1.6: Obtener por Regresión Lineal Múltiple

El sistema debe brindar la posibilidad de obtener el esfuerzo por la técnica regresión lineal múltiple.

RF1.7: Actualizar lista de archivos recientes

El sistema debe brindar información actualizada del total de archivos recientes.

RF1.8: Actualizar lista de archivos almacenados

El sistema debe brindar información actualizada del total de archivos almacenados.

RF2: “Exportar datos”.

El sistema debe brindar la posibilidad de exportar los datos evaluados por ambas técnicas y por la que el usuario seleccione.

RF2.1: Exportar PDF

El sistema debe permitir al usuario generar un informe con los datos de las métricas utilizadas y el esfuerzo obtenido.

RF2.2: Exportar Excel

El sistema debe brindar la posibilidad de generar un documento Excel con los datos de los modelos interpretados.

RF3: “Consultar”

El sistema debe brindar la posibilidad de listar los datos evaluados.

RF4: “Generar estadísticas”.

El sistema debe brindar la posibilidad de generar gráficos para la mejor interpretación los datos evaluados.

2.2.3 Requisitos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema, como pueden ser la seguridad y la fiabilidad del mismo. Estos requerimientos no solo hacen referencia al sistema que se va a desarrollar, también consideran algunas restricciones, es decir, se encargan de definir aspectos como, por ejemplo, qué estándares de calidad se deben seguir en el desarrollo del sistema, (García Peñalvo & García Holgado 2018).

A continuación, se muestra la definición de los mismos:

“Requisitos de Software”:

RNF1: Java Runtime Environment (JRE) versión 7.0 o superior.

RNF2: Lector de PDF Adobe Reader o similares.

RNF3: Lector de Excel.

RNF4: Sistema operativo Windows 95 o superior.

“Requisitos de Hardware”:

RNF5: Memoria RAM igual a 512 MB o superior.

RNF6: Espacio en disco igual a 1.5 M o superior.

“Requisitos de Portabilidad”:

RNF7: El sistema debe ser multiplataforma.

“Requisitos de Interfaz”:

RNF8: La interfaz contará con botones y menús desplegables que faciliten su utilización, esta debe ser simple de usar por los usuarios.

“Ayuda y documentación”:

RNF9: Proporcionar ayuda al usuario para comprender fácilmente su modo de uso.

2.3 Diagrama de casos de uso del sistema SDEffort-BPMN

Los diagramas de caso de uso de sistema contienen los casos de uso más importante desde el punto de vista de la arquitectura del sistema. (Molina et al. 2018)

El siguiente diagrama de casos de uso representa las clases más significativas desde la perspectiva del diseño, que posteriormente fueron implementadas en el sistema. La Figura 6 muestra el diagrama de casos de uso del sistema con el fin ilustrar de manera general las funcionalidades de la herramienta.

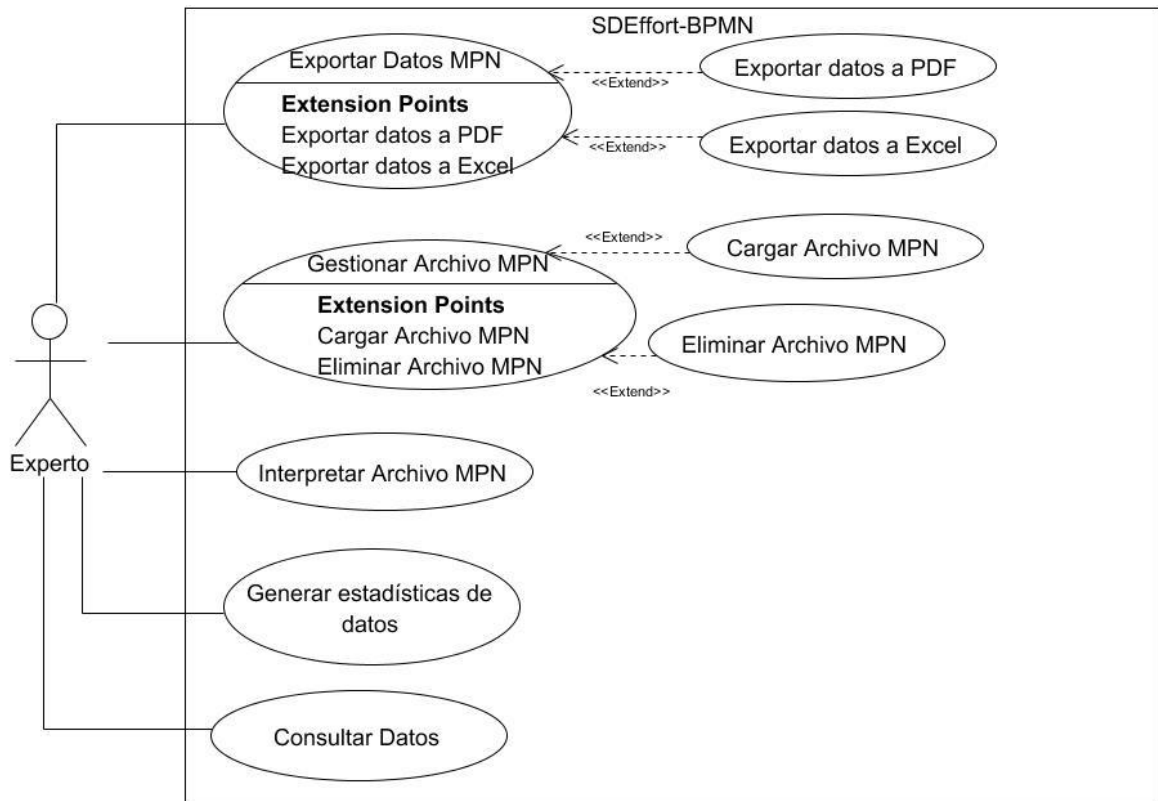


Figura 6. Diagrama de casos de uso del sistema SDEffort-BPMN.

2.4 Caso de uso significativo

En este sistema el Caso de Uso significativo es el siguiente:

- **Interpretar MPN:** Este caso de uso permite extraer información de los modelos de proceso de negocio (diagrama), correspondiente a los valores de cada una de las métricas que permiten el cálculo del esfuerzo.

A continuación, se muestra la descripción detallada de esta funcionalidad.

2.4.1 Formatos interpretables para el caso de uso Interpretar MPN.

Para seleccionar qué formatos se interpretarían de forma inicial se realizó un estudio de varias herramientas de modelado basadas en la notación BPMN, dando como resultado la selección realizada en el capítulo 1.

A continuación se muestran algunas etiquetas que presentan los formatos de salida .bpm, .xml, .bpnm, .xpdl de las herramientas estudiadas y que permitirán obtener los valores de las métricas propuestas por (Granados Hondares 2015) y analizadas en la presente investigación.

Tabla 6. Formatos Interpretables

<i>Etiquetas formato .xpdl</i>	<i>Etiquetas formato .bpnm</i>
<pre> <Pools> <Pool id> <Lanes> < lane id /> </Lanes> </Pool> </Pools> <Workflowprocess> <Activities> <Activity> <Event> Start, Tasks, Route, Intermediate, Ends </Event> </Activity> </Activities> </Workflowprocess> </pre>	<pre> <model: laneSet id> <model: lane id="" name="" /> <model: lane id="" name="" /> </model: laneSet> <model: userTask /> <model: startEvent /> <model: serviceTask /> <model: parallelGateway /> <model: serviceTask /> <model: endEvent /> <model: sequenceFlow /> </pre>
<i>Etiquetas formato .bpm</i>	<i>Etiquetas formato .xml</i>
<pre> <Package> <Pools> <Pool> <Lanes/> </Pool> </Pools> <Artifacts /> <WorkflowProcesses> <WorkflowProcesses> <ActivitySets> <ActivitySet> <Associations /> <Artifacts /> <Activities /> <Transitions /> </ActivitySet> </ActivitySets> </WorkflowProcesses> </Package> </pre>	<pre> <Pools> <Pool > <Lanes /> </Pool> </Pools> <WorkflowProcesses> <Workflowprocess > <Activities> <Activity > <Event> <StartEvent Trigger="None" /> </Event> <Task /> </Activity> </Workflowprocess> </WorkflowProcesses> </pre>

<pre> <Activity> <Implementation> Task </Implementation> </Activity> <Activity> <Route GatewayType /> </Activity> <Transitions> <Transition from="" To="" /> </Transitions> </Activities> </WorkflowProcesses> </WorkflowProcesses> </Package> </pre>	
---	--

2.4.2 Especificación del Caso de Uso “Interpretar MPN”

- El experto selecciona la técnica estadística para evaluar los archivos que desea interpretar y el sistema muestra el resultado de la evaluación y permite al usuario conocer detalles de misma.

Tabla 7. Especificación del Caso de Uso “Interpretar MPN”

Caso de uso del sistema	Interpretar MPN
Actor	Experto
Propósito	Interpretar los modelos de procesos de negocio con el fin de obtener el esfuerzo que demanda la automatización de proyecto de software.
Resumen	Inicia cuando el especialista selecciona la técnica estadística con que desea evaluar, selecciona la opción “Interpretar” del menú. Finaliza cuando se muestran una tabla con los resultados de la interpretación junto al esfuerzo calculado de cada proceso seleccionado a interpretar
Responsabilidades	Obtener el esfuerzo que demanda la automatización de proyectos de software.
Casos de uso asociados	Cargar MPN
Precondiciones	Cargar al menos un archivo, Conexión a la base de datos
Post-condiciones	Se almacenan los datos.
Interfaces de usuario caso de uso “Interpretar”	

Software Development Esfort

Archivo Reportes Ayuda

Interpretar Archivo MPN

Archivos Recientes Archivos Repetidos

- Acopiar 78(productos.bpmn
- Acopio.bpmn
- Complejo escultorico.xpdl
- Ferrocarril.xml
- Gestión de Información Digital.bpmn
- New Model.bpm
- Prueba 2.bpm
- Realizar trabajo periodístico .bpmn
- Solicitar reactivos.bpmn
- Telecentros de Santa Clara.bpmn
- camunda.bpmn
- organización ACME.bpmn

Archivos Almacenados Archivos Recientes

Formatos: .xpdl: 1
.bpmn: 8
.bpm: 2
.xml: 1

Total: 12

Técnica de evaluación

☐ Regresión Lineal Simple ☒ Regresión Lineal Múltiple

Buscar por: Nombre

Página Inicio x

SDEffort-BPMN

Cargar Archivo Ayuda

Técnica de evaluación

☐ Regresión Lineal Simple ☒ Regresión Lineal Múltiple

Página Inicio x Regresión Lineal Múltiple x

Nombre de Proyecto	Coeeficiente de complejidad	Complejidad de automatización	Esfuerzo(Horas/Hombre)
Acopiar78(productos.bpmn	1.19	NO CALCULADA	46.89
Acopio.bpmn	1.31	MEDIA	22.3
Complejo escultorico.xpdl	1.3	NO CALCULADA	14.57
Ferrocarril.xml	1.3	NO CALCULADA	14.57
Gestión de Información Digital....	1.5	BAJA	6.83
New Model.bpm	1.3	NO CALCULADA	14.57
Prueba 2.bpm	1.3	NO CALCULADA	14.57
Realizar trabajo periodístico .bp...	1.22	BAJA	14.73
Solicitar reactivos.bpmn	1.16	NO CALCULADA	19.48
Telecentros de Santa Clara.bpmn	1.08	NO CALCULADA	40.03
camunda.bpmn	1.5	BAJA	6.83
organización ACME.bpmn	1.3	NO CALCULADA	15.44

Software Development Effort

Archivo Reportes Ayuda

Archivos Recientes Archivos Repetidos

- Acopiar7&(productos.bpmn
- Acopio.bpmn
- Complejo escultorico.xpdl
- Ferrocarril.xml
- Gestión de Información Digital....
- New Model.bpm
- Prueba 2.bpm
- Realizar trabajo periodístico .bp...
- Solicitar reactivos.bpmn
- Telecentros de Santa Clara.bpmn
- camunda.bpmn
- organización ACME.bpmn

Archivos Almacenados Archivos Recientes

Formatos: .xpdl: 1
.bpmn: 8
.bpm: 2
.xml: 1

Total: 12

Técnica de evaluación

☐ Regresión Lineal Simple ☒ Regresión Lineal Múltiple

Buscar por: Nombre

Página Inicio x Regresión Lineal Múltiple x

Nombre de Proyecto	Coefficiente de complejidad	Complejidad de automatización	Esfuerzo(Horas/Hombre)
Acopiar7&(productos.bpmn	1.19	NO CALCULADA	46.89
Acopio.bpmn	1.31	MEDIA	22.3
Complejo escultorico.xpdl	1.3	NO CALCULADA	14.57
Ferrocarril.xml	1.3	NO CALCULADA	14.57
Gestión de Información Digital....	1.5	BAJA	6.83
New Model.bpm	1.3	NO CALCULADA	14.57
Prueba 2.bpm	1.3	NO CALCULADA	14.57
Realizar trabajo periodístico .bp...	1.22	BAJA	14.73
Solicitar reactivos.bpmn	1.16	NO CALCULADA	19.48
Telecentros de Santa Clara.bpmn	1.08	NO CALCULADA	40.03
camunda.bpmn	1.5	BAJA	6.83
organización ACME.bpmn	1.3	NO CALCULADA	15.44

Detalles del Modelo: Acopiar7&(productos.bpmn

Número Total de Eventos de Inicio: 1
Número Total de Eventos de Intermedios: 0
Número Total de Eventos de Final: 2
Número Total de Eventos: 3
Número Total de Tareas: 27
Número Total de Subprocesos: 0
Número Total de Actividades: 27
Número Total de Puertas: 4
Número de Flujos de Secuencia entre tareas: 7
Número de Flujos de Secuencia procedentes de un evento: 0
Número Total de Flujos de Secuencia: 37
Número Total de Lanes: 7
Número Total de Pools: 1
Número Total de Objetos de dato: 0
Nivel de Conectividad entre Actividades: 1.37

Flujo normal de eventos

Flujo normal	Acción del actor	Respuesta del sistema
	1. El Especialista da clic en la opción "Interpretar".	2. Sistema define los tipos de archivos cargados y cantidad 3. Sistema identifica la técnica de evaluación seleccionada 4. Sistema interpreta los archivos por la técnica seleccionada 5. Sistema devuelve tabla con valores obtenidos de la interpretación de archivos

	6. Especialista da clic sobre el archivo que desea conocer detalles	7. Sistema muestra detalles de las métricas utilizadas para obtener esfuerzo y complejidad en cada archivos interpretado que el usuario seleccione.
--	---	---

2.5 Planificación basada en Puntos de Casos de Uso. SDEffort-BPMN

El método de estimación de puntos de casos de uso es factible de aplicar cuando se ha realizado un análisis de requisito, se han establecido las principales funcionalidades y se ha presentado el modelo de casos de uso del sistema. La estimación mediante el análisis de Puntos de Casos de Uso, consiste en un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para que luego a partir de esos factores se pueda contabilizar el tiempo total estimado para el proyecto en cuestión, (García Peñalvo & García Holgado 2018). En este epígrafe se realizará el proceso para el cálculo de costo y tiempo del proyecto utilizando este método.

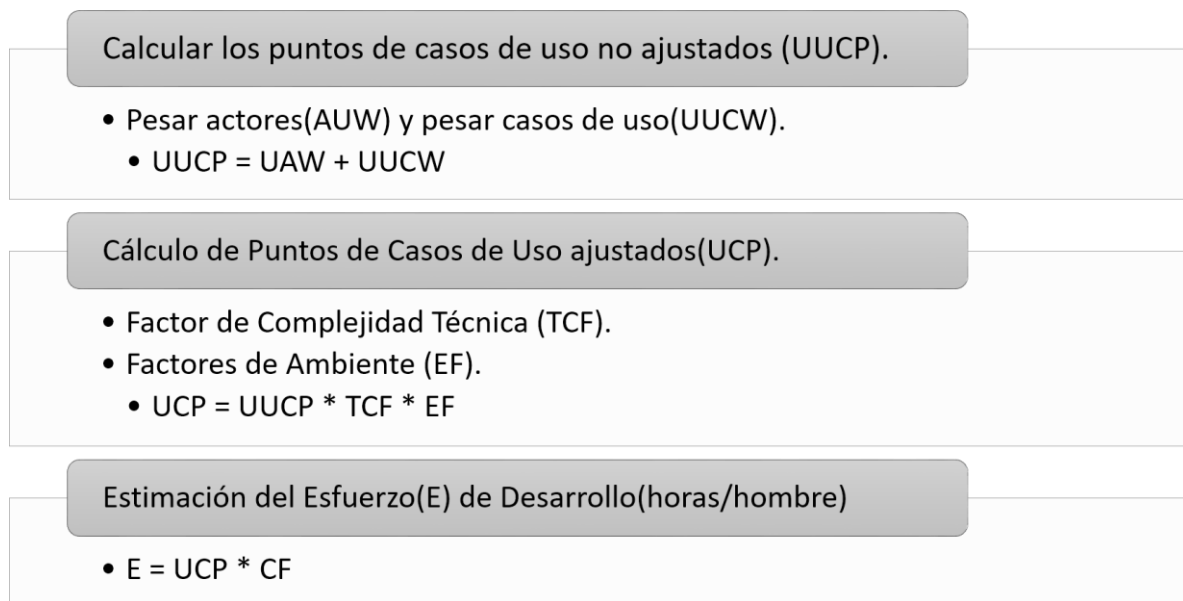


Figura 7 Técnica de estimación Puntos de Casos de Uso

2.5.1 Cálculo de los Puntos de Caso de Uso sin ajustar.

Para calcular los puntos de casos de uso no ajustados (UUCP) se define un peso tanto para actores (UAW) como para casos de uso (UUCW) y se suma el resultado de esos pesos.

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 35$$

$$UUCP = 38$$

Donde,

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

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

Factor de Peso de los Actores sin ajustar (UAW).

Para obtener el valor del factor de peso de los actores, se debe calcular la cantidad de actores existentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo el tipo de actor, pudiendo ser éste: simple, medio y complejo; así como también teniendo en cuenta la forma que el mismo interactúa con el sistema, asignando el peso correspondiente asociado a dicha interacción. Se calcula mediante la ecuación: $UAW = \sum (Cantidad\ TipoActor_i * Peso_i)$

Tabla 8 Clasificación de Actores.

<i>Tipo de Actor</i>	<i>Descripción</i>	<i>Peso</i>	<i>Cantidad de Tipo de Actores</i>	<i>Resultado</i>
Simple	Otro sistema externo, interactúa con el sistema a desarrollar mediante una interfaz de programación definida y conocida, (API, <i>Application Programming Interface</i>)	1	0	0
Medio	Otro sistema externo que interactúa a través de protocolo (conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de la red).	2	0	0
Complejo	Un usuario físico que interactúa a través de una interfaz gráfica de usuario.	3	1	3
			Total	3

$$UAW = \sum (Cantidad\ TipoActor_i * Peso_i)$$

$$UAW = 3$$

Factor de Peso de los Casos de Uso sin ajustar(UUCW).

Para obtener el valor del factor de Peso de los casos de uso, se calcula la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos.

Tabla 9 Clasificación de los Casos de Uso.

<i>Tipo de Caso de Uso</i>	<i>Descripción</i>	<i>Peso</i>	<i>Cantidad de Tipo de Casos de Uso</i>	<i>Resultado</i>
Simple	El caso de Uso contiene hasta 3 transacciones	5	7	35
Medio	El caso de Uso contiene de 4 a 7 transacciones	10	0	
Complejo	El caso de uso tiene más de 7 transacciones	15	0	
		Total		35

$$UUCW = \sum (\text{Tipo de Caso de Uso}_i * \text{Peso}_i)$$

$$UUCW = 85$$

Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar ese valor mediante factores de ajuste, tanto técnicos como de ambiente, haciendo uso de la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

$$UCP = 38 * 0.88 * 0.6$$

$$UCP = 20.06$$

Donde,

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de Complejidad Técnica (TCF).

El coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 al 5, donde:

Un valor de 0 significa que el factor es irrelevante.

Un valor de 3 significa que el factor es promedio.

Un valor de 5 significa que el factor es esencial.

Tabla 10 Cálculo de los Factores Técnicos.

<i>Factor</i>	<i>Descripción</i>	<i>Peso</i>	<i>Valor Asignado</i>
---------------	--------------------	-------------	-----------------------

F1	Sistema Distribuido	2	0
F2	Tiempo de Respuesta y Desempeño	1	2
F3	Eficiencia respecto al usuario final	1	4
F4	Procesamiento interno complejo	1	2
F5	Código reutilizable en otras aplicaciones	1	0
F6	Facilidad en la instalación	0,5	3
F7	Usabilidad (Fácil de usar)	0,5	5
F8	Portabilidad	2	5
F9	Facilidad en mantener	1	4
F10	Accesos simultáneos (concurrentes)	1	0
F11	Incluye objetivos especiales de seguridad	1	0
F12	Provee acceso directo a terceros	1	0
F13	Se requiere facilidades especiales de entrenamiento a usuarios	1	2
Total		28	

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

$$TCF = 0.6 + 0.01 * (28)$$

$$TCF = 0.88$$

Factores de Ambiente (EF).

Además de tener en cuenta los factores técnicos para el ajuste de los UUCP (Puntos de Casos de Uso no ajustados), se contabilizan los factores de ambiente. De manera similar al cálculo de los TCF, a cada factor de ambiente definido en la siguiente tabla, se la asignan valores entre el 0 y el 5.

Tabla 11 Cálculo de los Factores Ambiente

<i>Factor</i>	<i>Descripción</i>	<i>Peso</i>	<i>Valor Asignado</i>
E1	Familiarizado con el proceso de desarrollo (RUP)	1,5	5

E2	Experiencia en la aplicación	0,5	5
E3	Experiencia en orientación a objetos	1	5
E4	Capacidades de análisis	0,5	5
E5	Motivación	1	5
E6	Requisitos estables	2	4
E7	Trabajadores a tiempo parcial	-1	0
E8	Lenguaje complejo	-1	4
Total			26.5

$FA = 1.4 - 0.03 \times \Sigma (\text{Peso } i \times \text{Valor asignado})$

$FA = 1.4 - 0.03 \times 26.5$

$FA = 0.6$

2.5.2 Estimación del Esfuerzo de Desarrollo.

El esfuerzo en horas-hombre se calcula aplicando la siguiente ecuación:

$E = UCP * CF$

$E = 20.06 * 20$

$E = 401 \text{ horas-hombre}$

$E = 3.3 \text{ meses/hombre}$

Donde,

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

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

Desde el punto de vista de complejidad de los algoritmos, la aplicación que se desarrolla no se considera compleja, dado que basa su función central en la extracción de datos a partir de diagramas con notación BPMN. Cuenta con un único actor: “Experto”, encargado del procesamiento de los modelos de procesos de negocio y la obtención del esfuerzo de automatización de proyectos de software. El equipo de desarrollo es pequeño, cuenta con una persona que ocupa los diferentes roles desempeñados en cada uno de los flujos de trabajo de la metodología seleccionada, AUP. En consecuencia, con las características antes mencionadas, los valores asignados para el cálculo del esfuerzo de desarrollo dan como resultado 401 horas/hombre. Este valor se puede traducir a aproximadamente 3.3 meses/hombre dedicada a la implementación del sistema en cuestión, teniendo en cuenta la

capacidad y desempeño laboral y previendo una utilización del tiempo diario promedio de 5 horas, sin contar fines de semana y días festivos.

2.5.3 Estimación del esfuerzo del proyecto

Para la estimación del cronograma completo del proyecto se presenta a continuación la distribución en términos de porcentos de dedicación a cada una de las actividades del proyecto. Se decidió tomar como referencia los flujos de trabajo en los que se estructuran la metodología de desarrollo seleccionada: modelo, implementación y pruebas. No se incluirán los flujos correspondientes a Desarrollo, Gestión de Configuración, Gestión del Proyecto y el Ambiente por las características propias del mismo. Se incluyen en esta distribución una parte esencial de la investigación, relacionada con el estudio, procesamiento de datos y obtención del modelo predictivo del esfuerzo.

Actividades	Tiempo asignado (porcentaje)	Tiempo asignado (horas)
Modelo	10	100.2
Implementación	40	401
Prueba	10	100.2
Obtención del modelo	20	200.2
Documentación	20	200.2
Tiempo total empleado		1002.5

$$E_{total} = E / 0.4$$

$$E_{total} = 401 / 0.4$$

$$E_{total} = 1002.5(h/h)$$

$$E_{total} = 8 \text{ (meses/ hombre)}$$

Donde,

E_{total}: Esfuerzo Total

De acuerdo a la técnica de estimación utilizada, se planifica un cronograma de trabajo para aproximadamente 8 meses de trabajo comenzando a trabajar en septiembre y concluyendo en mayo, teniendo en cuenta periodos no productivos.

1. Costo de desarrollo del producto.

De acuerdo a la tarifa horaria planificada por DESOFT 11.73 pesos cubanos, considerándose esta por conocimientos previos de esta, se estima el costo por concepto de

salario, teniendo en cuenta que es costo de un producto de software conlleva otros gastos adicionales que pueden ser considerados.

Se calcula mediante la siguiente ecuación:

$$C = E_{\text{total}} * CH * CHH$$

$$C = 1002.5 * 1 * 11.73$$

$$C = \$ 11759$$

Donde,

CH: Cantidad de Hombres

CHH: Costo por Hombre Hora

Finalmente concluimos que el costo total es de **\$ 11759**.

2.6 Conclusiones del capítulo

Se realizan las consideraciones pertinentes sobre el negocio donde se identifica como trabajador del negocio al experto quien interactúa de forma directa en el proceso de interpretación y obtención de esfuerzo, quien de la misma forma se convierte en el actor del sistema a automatizar. Se definen los requisitos funcionales y no funcionales del sistema y se describe como caso de uso más significativo Interpretar MPN.

CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el capítulo se describe la arquitectura del sistema, se especifica el diagrama de clases de diseño y diagrama de secuencia de los Casos de Uso significativos. Se fundamenta los principios de diseño, mostrando el diseño de la base de datos tanto el modelo conceptual como el modelo físico de datos. Se presenta el modelo de componentes y diagrama de despliegue.

3.1 ARQUITECTURA DEL SISTEMA SDEFFORT-BPMN.

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. (Pressman & Ph 2010)

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución.(Rojas et al. 2018)

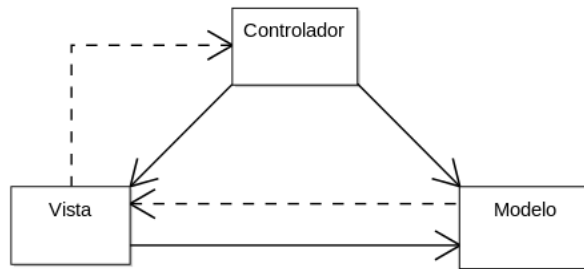


Figura 8. Estructura del patrón de arquitectura de software Modelo Vista Controlador (MVC)

Se asume el patrón de arquitectura de software Modelo Vista Controlador (MVC), donde se propone la construcción de tres componentes; el paquete “Modelo” compuesto por las clases, `AbstractProcesarArchivo`, `IProcesarArchivo` y `IntetpretarArchivo`, clases que trazan la estrategia a seguir para el diseño del software y el paquete “Modelo.derivadas” que conforman las estrategias concretas. El paquete “Principal” compuesto por la clase `Principal` que contiene la interfaz visual, o sea contiene los componentes que interactúan con el usuario y que mantiene relación de dependencia con el paquete “Controlador”. El paquete “Controlador”, compuesto por la clase `GestionArchivo()`, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

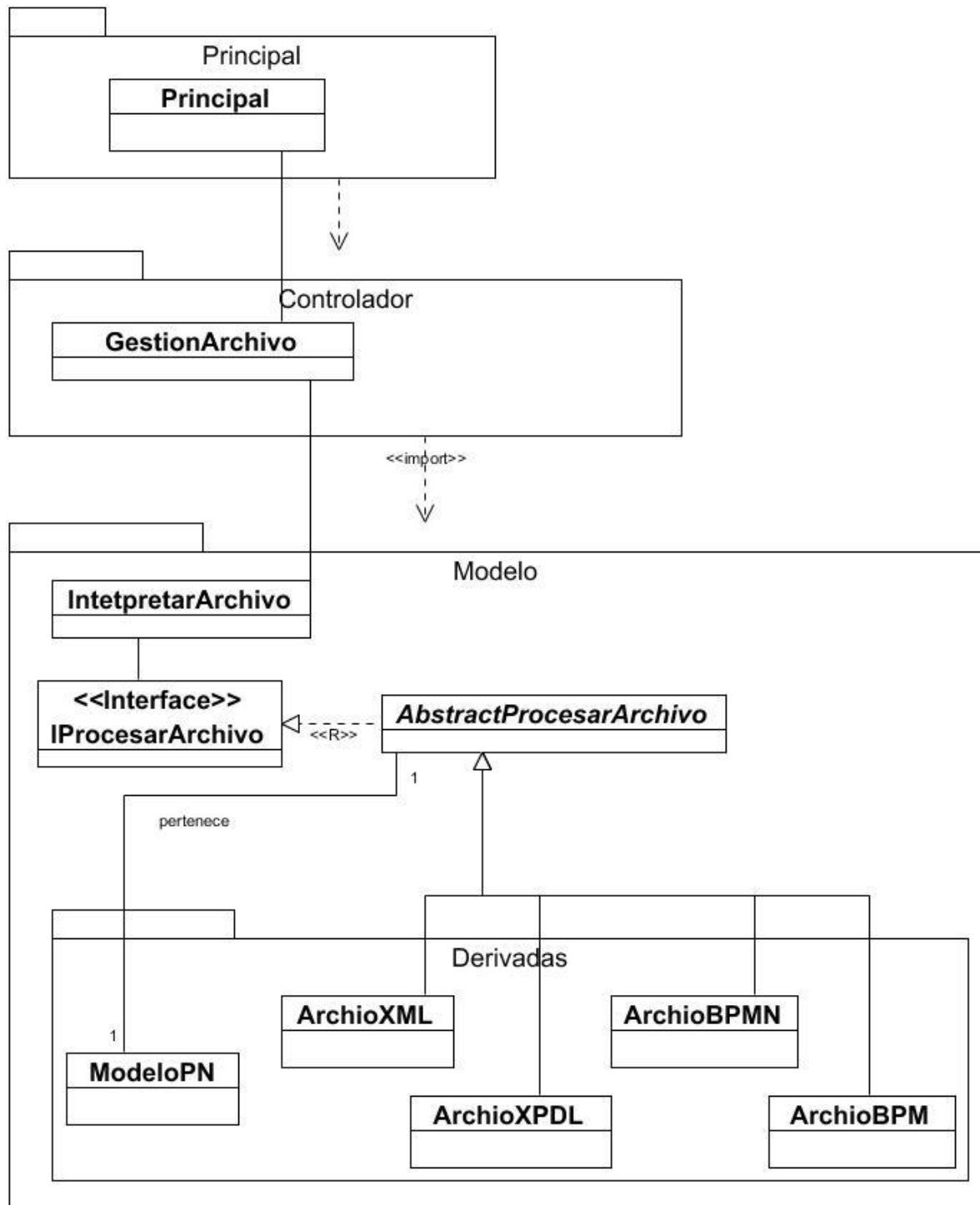


Figura 9. Patrón de arquitectura de software Modelo Vista Controlador (MVC) para SDEffort-BPMN.

3.2 DIAGRAMA DE CLASES DE DISEÑO SDEFFORT-BPMN.

El diagrama de clases de diseño está compuesto por elementos identificados como clase y relaciones, que al unirlos conforman los esquemas o diagramas que representarán un modelo de clases. Estos diagramas resultan de la composición de los casos de uso. El diagrama de clases debe expresar detalles de los objetos que existen en el área de enfoque, en él se incluyen atributos y métodos de cada una de las clases, (García Peñalvo & García Holgado 2018).

El siguiente diagrama de clase de diseño muestra la arquitectura definida en el desarrollo de la herramienta SDEffort-BPMN, así como el uso del patrón de diseño.

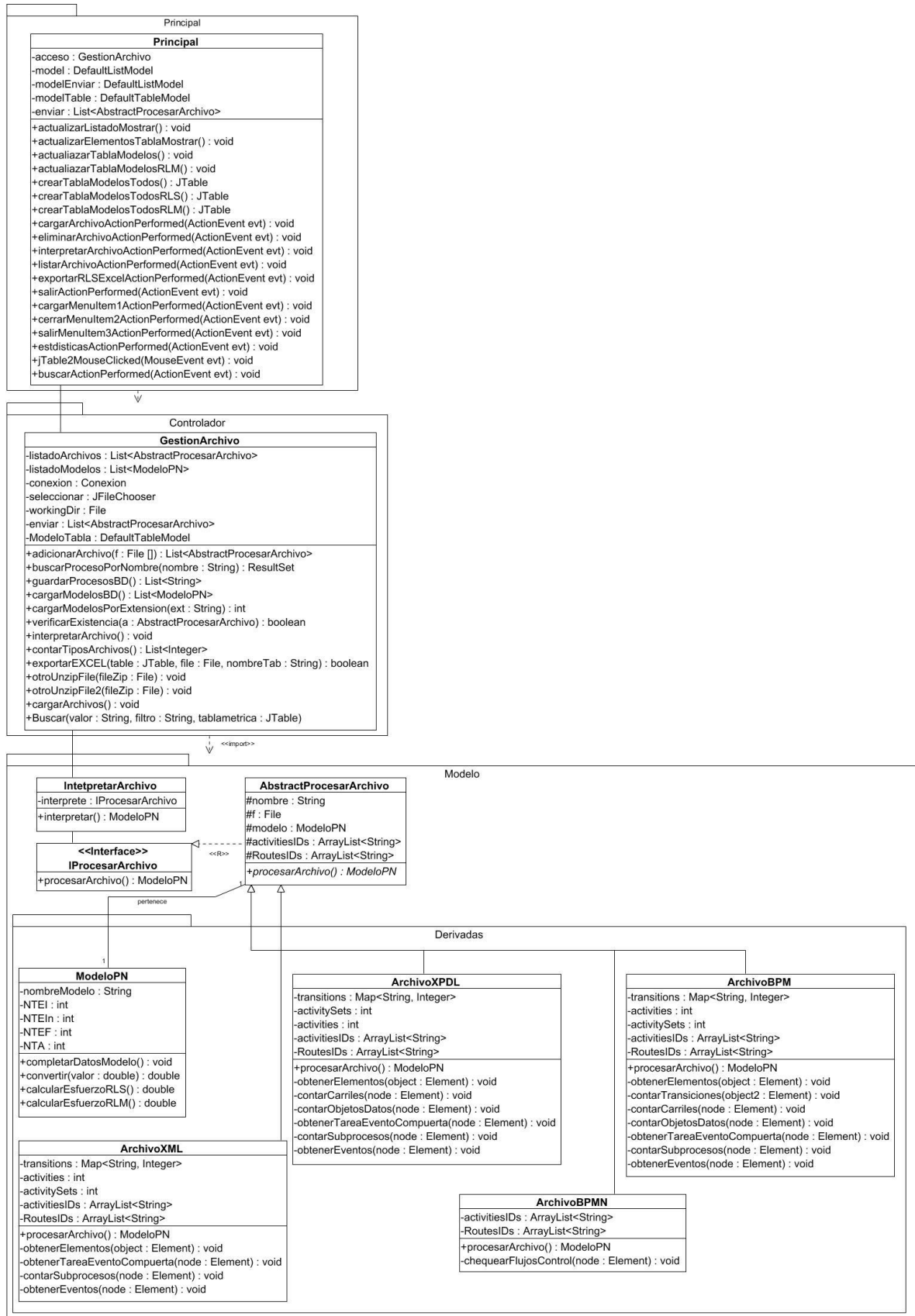
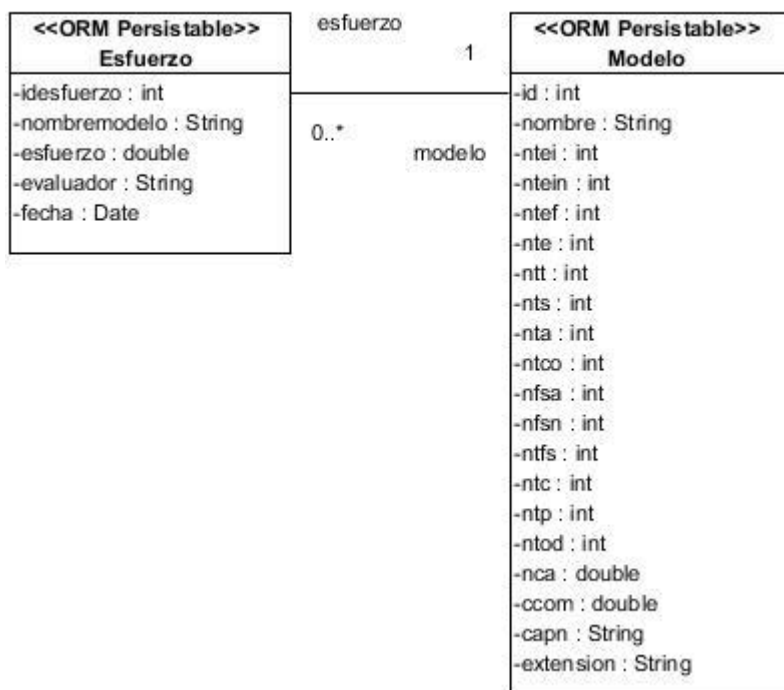


Figura 10. Diagrama de clases de diseño para SDEffort-BPMN

En el diagrama de clases del sistema se muestran detalles de las clases sus relaciones y métodos con los que interactúan. El paquete de clases “Derivadas” contiene la implementación de la interpretación de cada uno de los archivos analizados en el capítulo 1, formatos más utilizados en el mundo empresarial para exportar modelos de procesos de negocio, los que se analizan e interpretan para realizar el cálculo del esfuerzo de desarrollo de proyectos a partir de la representación gráfica de los modelos de negocio.

La persistencia de una clase está definida por la propiedad de los objetos de trascender su estado en el tiempo y el espacio: una clase persistente existirá durante la ejecución de un programa, y deberá sobrevivir incluso a la eliminación o colapso del mismo. En la siguiente figura se muestra la vista del diagrama de clases persistentes para las clases de almacenamiento de datos:

**Figura 11. Diagrama de persistencia SDEffort-BPMN**

3.3 Principios de diseño SDEffort-BPMN.

Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del diseño dentro de un contexto específico y entre “fuerzas” que afectan la manera en la

que se aplica y en la que se utiliza dicho patrón. El objetivo de cada patrón de diseño es proporcionar una descripción que permita a un diseñador determinar 1) si el patrón es aplicable al trabajo en cuestión, 2) si puede volverse a usar (con lo que se ahorra tiempo de diseño) y 3) si sirve como guía para desarrollar un patrón distinto en funciones o estructura (Pressman & Ph 2010).

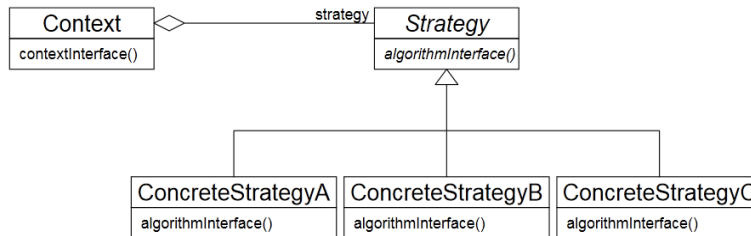


Figura 12. Diagrama de clases del patrón de diseño *Strategy*.

Se utiliza el patrón *Strategy* como patrón de diseño para el desarrollo de software ya que se clasifica como patrón de comportamiento, porque determina cómo se debe realizar el intercambio de mensajes entre diferentes objetos para resolver una tarea. El patrón estrategia permite mantener un conjunto de formatos a interpretar cada uno con diferentes algoritmos ya que cada una de las estrategias que se diseña tendrá que sobrescribir el método **ProcesarArchivo()** y proveer un algoritmo concreto para dicha estrategia. Relaciona las clases de forma que con el uso de la interfaz y la agregación de la clase abstracta se crea una estrategia para la interpretación de los diferentes formatos que extienden de la clase abstracta, lo que brinda al sistema adaptabilidad en el momento de incluir nuevos tipos de archivo que surjan como resultado de investigaciones futuras. Bastará solo crear una nueva clase con el nombre del archivo que redefina el método "ProcesarArchivo", sin que esto genere cambios en el código de la aplicación.

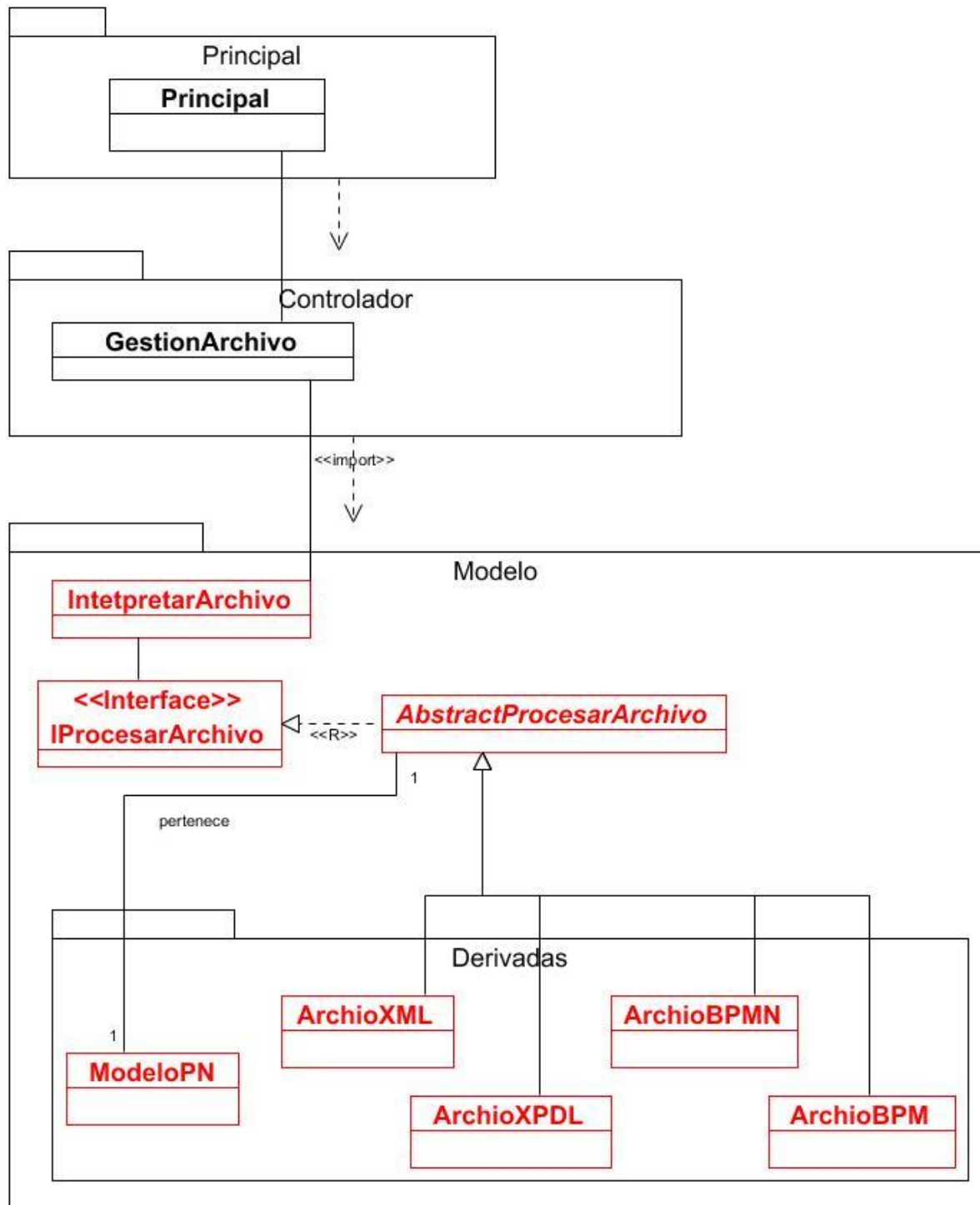


Figura 13. Implementación del patrón de diseño *Strategy* en SDEffort-BPMN.

3.4 Diagrama de secuencia Caso de Uso “Interpretar MPN”

Los Diagramas de Secuencia representan una interacción entre objetos de manera secuencial en el tiempo. Muestra la participación de objetos en la interacción entre sus “líneas de vida” (desde que se instancia el objeto) y los mensajes que ellos organizadamente intercambian en el tiempo, (García Peñalvo & García Holgado 2018).

El siguiente diagrama de secuencia muestra la interacción del experto con el sistema en el caso de uso específico “Interpretar MPN”, señalando los objetos y las clases que intervienen en la interpretación de los archivos.

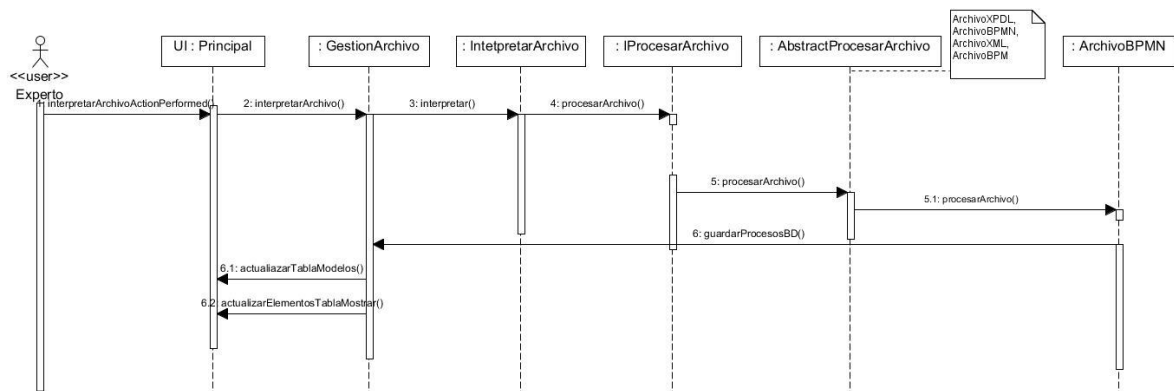


Figura 14. Diagrama de secuencia Caso de Uso “Interpretar MPN”.

3.5 Diseño de la base de datos SDEffort-BPMN.

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos de un sistema de información. Estos modelos de datos están basados en una percepción del mundo real que consta de una colección de objetos básicos, llamado entidades, y de relaciones entre esos objetos, (García Peñalvo & García Holgado 2018).

Se muestra a continuación el modelo conceptual y físico de los datos, donde se describe la relación entre las tablas de la base de datos Interpretacion.db. La tabla modelo contiene los atributos del modelo que se desea predecir el esfuerzo y la tabla esfuerzo, el esfuerzo obtenido por cada técnica de evaluación utilizada. Por tanto, un modelo puede estar relacionado con uno o varios esfuerzos, mientras un esfuerzo está asociado a un único modelo.

3.5.1 Modelo lógico de datos

Un modelo conceptual de datos identifica las relaciones de más alto nivel entre las diferentes entidades. Un modelo de datos lógico contiene más detalle que un modelo conceptual ya que se definen las entidades transaccionales y operativas más detalladas. El modelo lógico es independiente de la tecnología en la que se implementará (Rondón et al. 2009).

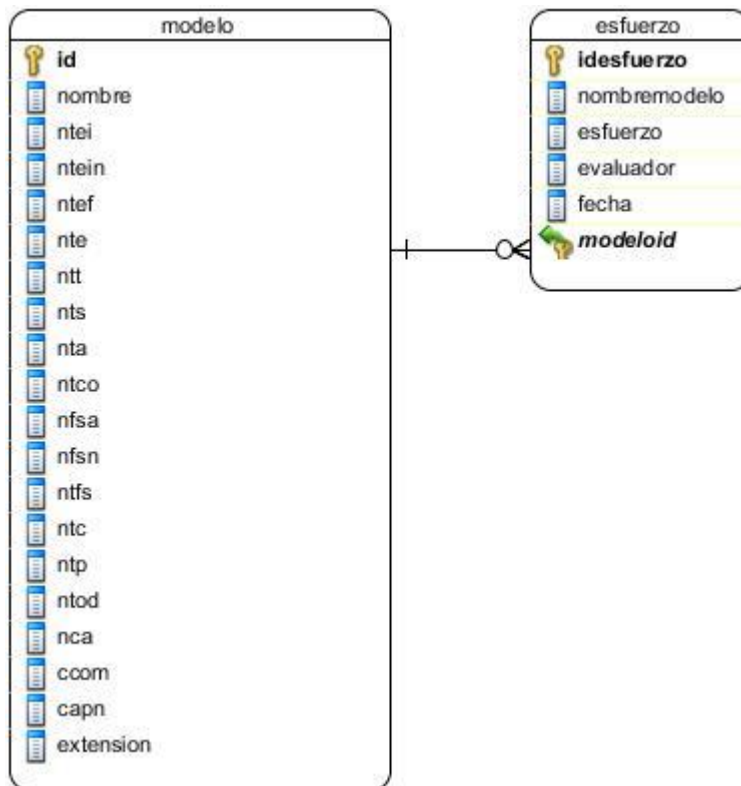


Figura 15. Modelo lógico de datos SDEffort-BPMN.

3.5.2 Modelo físico de datos

Uno o más modelos físicos pueden desarrollarse a partir de cada modelo lógico. El modelo físico debe mostrar los suficientes detalles tecnológicos para producir e implementar la base de datos en cuestión, (García Peñalvo & García Holgado 2018).

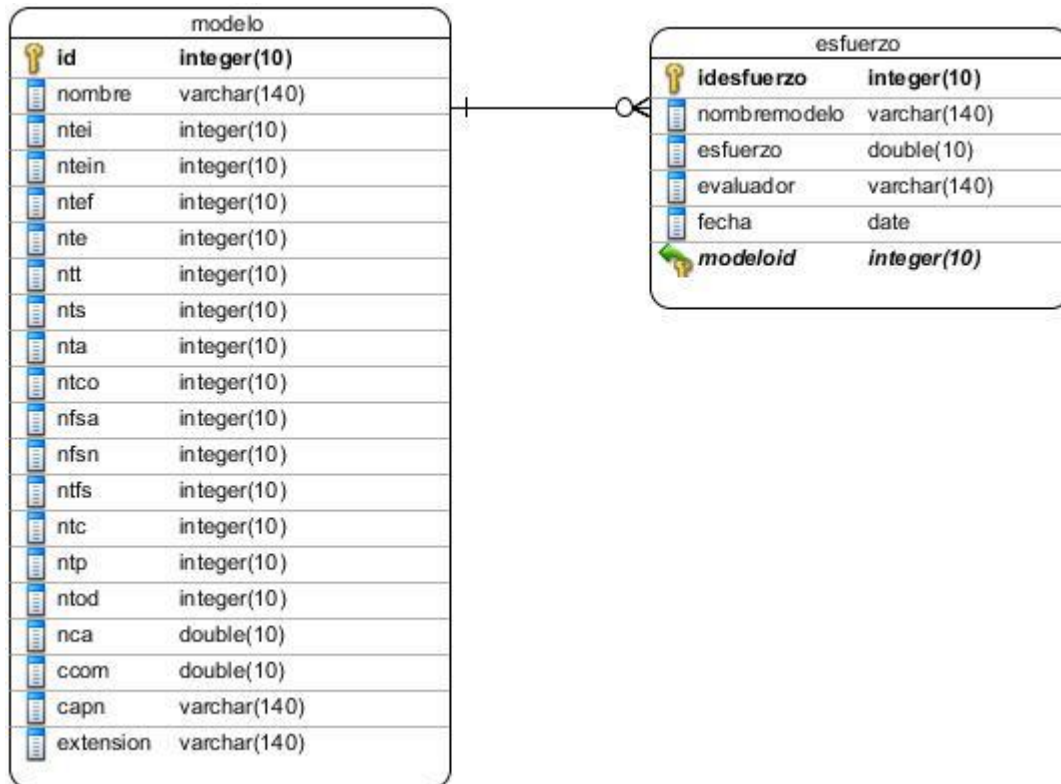


Figura 16. Modelo físico de datos SDEffort-BPMN

3.6 Diagrama de componentes integrado al de despliegue SDEffort-BPMN.

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: Dispositivos, Procesadores, Memoria. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo Código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente (Rojas et al. 2018).

La herramienta SDEffort-BPMN propuesta será usada en un solo ordenador y solo por el experto, con una memoria RAM superior a 512 MB y un espacio en disco superior a 1.5 MB. Para el correcto funcionamiento de la misma deben estar presentes un conjunto de componentes. A continuación, se describen brevemente.

- jdom: es una biblioteca de código abierto para manipulaciones de datos XML optimizados para Java.
- jxl: permite a los leer, escribir, crear y modificar hojas en un libro de Excel (.xls) en tiempo de ejecución.
- Itextpdf: permite, usando Java, editar o transformar contenido a PDF.
- Zip4j: biblioteca de código abierto de java para manejar archivos Zip.
- Sqlite: es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración.
- JFreeChart: es un marco de software open source para el lenguaje de programación java, el cual permite a creación de gráficos complejos de forma muy simple.

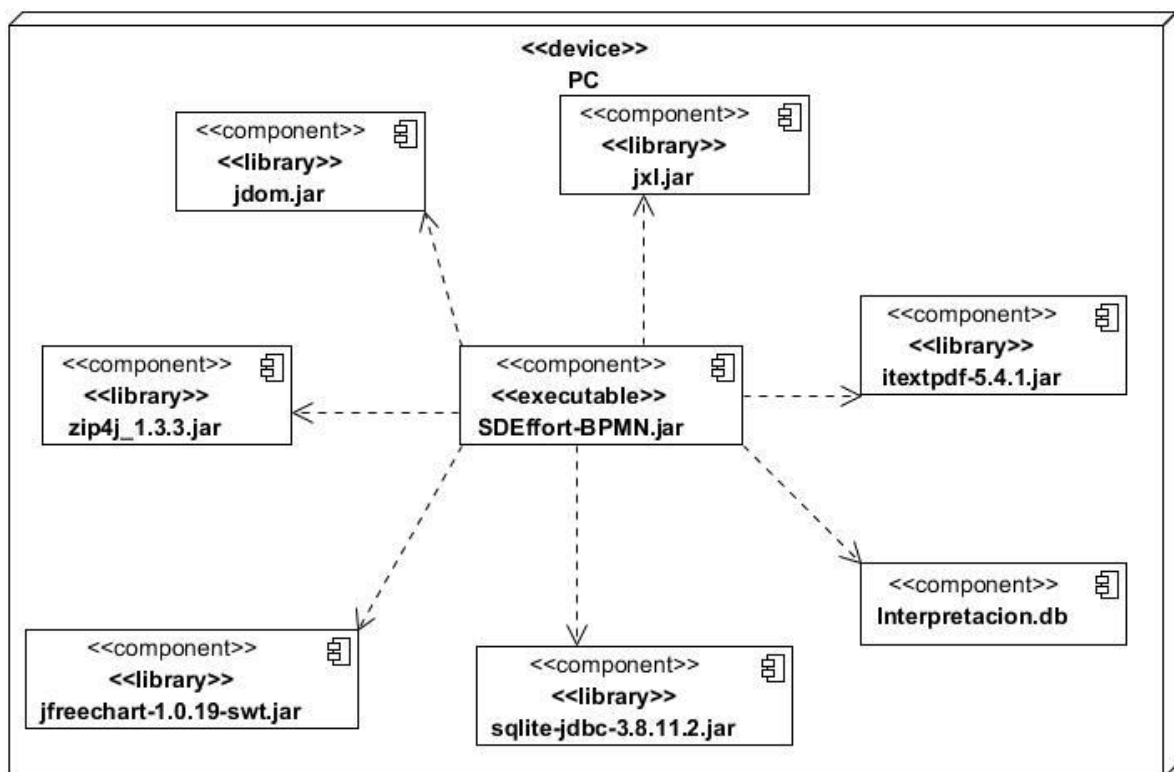


Figura 17. Diagrama de componentes integrado al de despliegue SDEffort-BPMN

3.7 Conclusiones del capítulo

El uso de Modelo Vista Controlador (MVC) en la arquitectura del sistema, brinda limpieza, claridad y robustez al software, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo.

Por su parte la implementación del patrón de diseño *Strategy* posibilita crear una estrategia para la interpretación de los diferentes archivos, brindando al sistema adaptabilidad en el momento de incluir nuevas extensiones, como formatos de salida de otras herramientas para el modelado de procesos de negocio, que surjan como resultado de investigaciones futuras.

CAPÍTULO 4. MODELO DE ESTIMACIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.

En este capítulo se especifican características del modelo de estimación de esfuerzo implementado en el software y se realizan pruebas de software como las pruebas de unidad e integridad para garantizar el correcto funcionamiento de la herramienta implementada.

4.1 Cálculo del esfuerzo utilizando Regresión Lineal Múltiple.

El esfuerzo es la medida o cantidad de trabajo que un equipo de desarrolladores debe aplicar en determinada tarea o etapa para lograr un objetivo en común, ya sean objetivos específicos o generales. El esfuerzo se puede reconciliar con la complejidad, refiriéndose esta al estado de eventos o cosas, que tienen múltiples interconexiones y una estructura altamente complicada, dado que mientras más complejo es el producto a desarrollar mayor esfuerzo de desarrollo requerirá.

En el capítulo 1 se hace referencia al cálculo de la complejidad de desarrollo de software propuesta por (Bandomo Toledo 2014) y referenciado por su importancia por (Granados Hondares 2015) en la que se logra captar un vínculo entre la complejidad y el esfuerzo de desarrollo a partir de modelos de procesos de negocio fundamentalmente en formatos de salida XPDL y BPMN. Esta última es tomada como antecedente a la presente investigación, por lo que se analiza la base de casos de la misma.

4.1.1 Base de casos

La investigación antecedente propone una base de casos bien conformada por trabajos de diplomas, donde por lineamientos se hace énfasis especial en los procesos de negocio como punto de partida al sistema a desarrollar y en la realización del estudio de factibilidad del

software. De este estudio debe quedar especificado el tiempo estimado de implementación requerido obteniendo finalmente un esfuerzo “confiable”.

Durante los cursos, 2011-2012, 2012-2013 y 2013-2014 en (Granados Hondares 2015) se realizó una revisión detallada de los informes de trabajos de diploma, decantando aquellos cuyos procesos de negocio no quedaran modelados o explicados de forma clara y detallada, y cuyo estudio de factibilidad no estuviera lo más completo posible. Se revisaron un total de setenta y tres trabajos de diploma, de los cuales se obtuvieron ochenta modelos de procesos de negocio. De esta revisión resultó la siguiente tabla:

Tabla 12. Estadística de la base de casos.

<i>Curso</i>	<i>Trabajos de Diplomas</i>	<i>Cantidad de MPN</i>
2011-2012	31	48
2012-2013	27	19
2013-2014	15	13
Total	73	80

Se obtienen los valores correspondientes a cada una de las métricas propuestas en dicha investigación, obteniendo el valor del esfuerzo previamente obtenido.

NOMBRE	NTEI	NTEIn	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN	NTFS	NTC	NTP	NTOD	NCA	CCOM	CAPN	ESFUERZO
AccederEstadistica_Elisa_12-13	1	0	1	2	3	0	3	0	2	0	4	1	1	0	.20	1,33	BAJA	6
AdicionarSolicitud_Félix_11-12	1	0	1	2	13	1	14	4	6	9	25	2	1	0	2,33	1,38	MEDIA	26
AdministrarEstadistica_Elisa_12-13	1	0	1	2	8	0	8	2	4	4	13	1	1	0	2,00	1,30	BAJA	15
AprobarSolicitudEntrenador_Airina_13-14	1	0	1	2	3	0	3	0	2	0	4	2	1	0	1,50	1,33	BAJA	6
AtenderPaciente_Liz_11-12	1	0	1	2	6	0	6	1	4	2	10	2	1	0	1,50	1,42	BAJA	12
AutenticarUsuario_Adrian11-12	1	0	1	2	5	0	5	3	2	5	12	2	1	0	2,50	1,50	NO CA	11
BuscarPorCategoria_Ernesto_12-13	1	0	1	2	5	0	5	0	4	0	6	2	1	0	1,25	1,20	BAJA	9
CancelarProcesoReemplazo_Ariel_11-12	1	0	1	2	5	0	5	1	3	2	8	2	1	0	1,66	1,33	BAJA	10
ComentarTema_Greisy_11-12	1	0	1	2	5	0	5	2	2	4	10	1	1	0	2,50	1,42	BAJA	11
ConexiónPerdida_Ariel_11-12	1	0	1	2	6	0	6	1	4	2	9	2	1	0	1,50	1,28	BAJA	12
ConsultarColección_Yealenys_11-12	1	0	1	2	6	0	6	0	5	0	7	2	1	0	1,20	1,16	BAJA	11
ConsultarContenido_Ariel_11-12	1	0	1	2	4	0	4	3	0	6	11	1	1	0	.00	1,57	NO CA	10
ConsultarDocumentos_Yaniel_11-12	1	0	1	2	5	0	5	3	1	6	12	2	1	0	5,00	1,50	NO CA	11
ConsultarMateriales_Adrian11-12	1	0	1	2	5	0	5	3	2	6	12	2	1	0	2,50	1,50	NO CA	11
ConsultarMateriales_Rachel_12-13	1	0	1	2	3	0	3	2	0	4	8	2	1	0	.00	1,60	BAJA	8
ConsultarPaciente_Gretel_12-13	1	0	1	2	7	0	7	1	5	2	10	1	1	0	1,40	1,25	BAJA	13
Cotización de planes turísticos_BPMN2	1	1	2	4	6	1	7	3	1	6	15	2	1	0	7,00	1,50	NO CA	15
CrearColección_Yaniel_11-12	1	0	1	2	17	0	17	0	16	0	18	2	1	0	1,06	1,05	NO CA	28
CrearNuevaColección_Yealenys_11-12	1	0	1	2	7	0	7	2	4	4	12	2	1	0	1,75	1,33	BAJA	14
DescargarArticulo_CarlosRivas_11-12	1	0	1	2	3	0	3	2	0	4	8	1	1	0	.00	1,60	BAJA	8
EditarPaciente_Liz_11-12	1	0	1	2	5	0	5	2	3	4	10	2	1	0	1,66	1,42	BAJA	11
EliminarColecciónObsoleta_Yaniel_11-12	1	0	1	2	4	0	4	3	0	6	11	1	1	0	.00	1,57	NO CA	10
EliminarColección_Yealenys_11-12	1	0	2	3	6	0	6	1	3	2	9	2	1	0	2,00	1,28	BAJA	12

Figura 18. Base de casos.

De esta forma la investigación que sirve de antecedente deja una base de casos sólida, con resultados confiables, para la utilización de la misma en otros proyectos como el que el presente trabajo de diploma trata.

4.1.2 Regresión Lineal Múltiple.

Los factores que intervienen en un experimento pueden ser cuantitativos o cualitativos. Un factor cuantitativo es aquel cuyos niveles pueden asociarse con puntos en una escala numérica. Los factores cualitativos son aquellos cuyos niveles no pueden ordenarse por magnitud (Carrasquilla-batista et al. 2016). En el caso especial de la obtención del esfuerzo, se tiene una métrica definida en el capítulo 1, que se considera cualitativo como es Complejidad de Automatización de procesos de negocio (CAPN). Las variables relacionadas con los elementos obtenidos del modelo son cuantitativas: NTEI, NTEIN, NTEF, NTE, NTT, NTS, NTA, NTCO, NFSA, NFSN, NTFS, NTC, NTP, NTOD, NCA, CCOM. Además, la variable de interés o salida (*Esfuerzo*) se mide en horas/hombre (h/h), por lo tanto, también es cuantitativa.

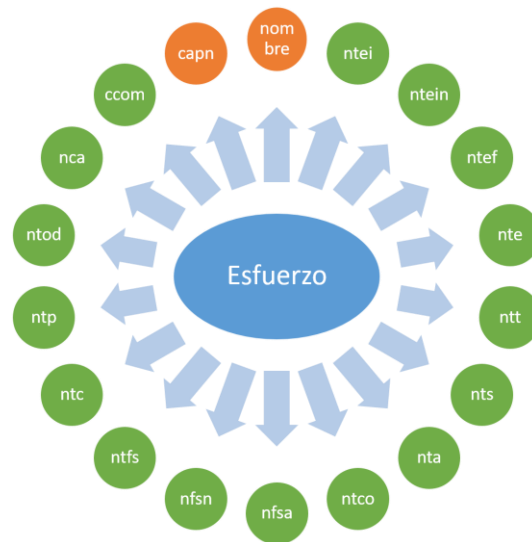


Figura 19. Métricas cualitativas y cuantitativas.

Se conoce como regresión múltiple el cálculo de la ecuación correspondiente de un plano que mejor describe la relación entre la respuesta y las variables que la explica. Dicha ecuación representa el plano que mejor se ajusta a los puntos en un gráfico de dispersión, siendo la respuesta la variable que se mide. Si se asocia con el esfuerzo de desarrollo de proyectos de software se establece este entonces como dependiente. Los factores que

influyen en esta respuesta pueden ser una cualquiera de las variables cuantitativas mencionadas anteriormente, las cuales son denominadas independientes. En la regresión múltiple se tienen k variables explicativas o variables predictoras. A partir de la regresión lineal es posible hacer predicciones sobre la respuesta con base en valores de las variables predictoras.

La estructura del modelo de regresión múltiple es la siguiente:

$$y = f(X_1, \dots, X_k) + \epsilon$$

Donde,

- y es la variable explicada, dependiente o respuesta.
- X_1, \dots, X_k son las variables explicativas, regresores o variables independientes.
- $y = f(X_1, \dots, X_k) + \epsilon$ es la parte determinista del modelo.
- ϵ representa el error aleatorio. Contiene el efecto sobre y de todas las variables distintas de X_1, \dots, X_k .

El modelo de regresión lineal múltiple tiene la forma:

$$y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + \epsilon$$

El modelo de regresión lineal múltiple se utiliza cuando:

- La variable dependiente, Y , depende linealmente de cada una de las variables explicativas, X_1, \dots, X_k .
- Un regresor no basta para explicar suficientemente la variabilidad de y .

Hipótesis del modelo de regresión lineal múltiple

Generalizando, al ajustar un modelo de regresión lineal múltiple, se supondrá que se verifican las siguientes hipótesis:

1. Linealidad: los valores de la variable dependiente están generados, Fijados los valores X_{1i}, \dots, X_{ki} de las variables X_1, \dots, X_k , por el siguiente modelo lineal:

$$y_i = \beta_0 + \beta_1 x_{1i} + \dots + \beta_k x_{ki} + e_i$$

2. Normalidad: la distribución de la perturbación aleatoria tiene distribución normal. Cada error $e_i \approx N(0, \sigma^2)$.
3. Independencia: las perturbaciones aleatorias son independientes entre sí. Cualquier par de errores e_i y e_j son independientes.
4. Las variables explicativas son, algebraicamente, linealmente independientes.

5. El número de datos es mayor o igual que $k + 2$.

Prueba de hipótesis en el modelo de regresión múltiple

En la prueba de hipótesis en el modelo de regresión múltiple, se establecen comparaciones de hipótesis nula y alternativa:

- Hipótesis nula: $H_0: \beta_1 = \beta_2 = \dots = \beta_k = 0$
- Hipótesis alternativa: H_1 : Existe algún β_i con $i = 1, \dots, k$ tal que $\beta_i \neq 0$

La aceptación de la hipótesis nula de la Prueba de Hipótesis, puede ser debida a:

- Independencia de la variable explicada (respuesta) frente a todas las variables predictoras /regresoras.
- Dependencia no lineal de la variable respuesta respecto de alguna variable predictora.

El rechazo de la hipótesis nula significa que la variable explicada depende linealmente de alguno de los regresores.

4.1.3 Análisis del modelo predictor con Regresión Lineal Múltiple.

Para la obtención de la ecuación predictora de esfuerzo se utiliza el software IBM SPSS Statistics. Se realiza un análisis de regresión lineal donde se introducen las variables cuantitativas correspondientes a las métricas de esfuerzo, calculadas y guardadas sus valores reales en la base de casos analizada al principio del capítulo. Se selecciona el método STEPWISE o “paso a paso” para el análisis de las variables dependientes y como variable independiente la variable ESFUERZO.

Como variables influyentes o variables de entradas en el modelo se obtienen tres variables, luego del análisis de Correlación Pearson (medida de la asociación lineal entre dos variables, los valores del coeficiente de correlación van de -1 a 1, los valores mayores indican que la relación es más estrecha), donde se identifican como variables no influyentes: NTEI, NTOD dado su coeficiente de correlación igual a cero. Por otro lado, las variables más influyentes para este modelo serán: NTA, NTCO y NTC, de las cuales la de mayor coeficiente de correlación es NTA.

A continuación, se muestra un fragmento de la tabla de correlación generada por el SPSS. Ver tabla de correlación en los anexos.

		ESFUERZO	NTEI	NTEIN	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN
Pearson Correlation	ESFUERZO	1.000	.	-.061	.506	.391	.990	.149	.992	.235	.873	.341
	NTEI	.	1.000

Figura 20. Fragmento de la tabla de correlación

La siguiente tabla indica las variables implicadas y el método elegido.

Tabla 13. Variables de entrada al modelo con SPSS

Variables Entered/Removed ^a			
Model	Variables Entered	Variables Removed	Method
1	NTA		Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
2	NTCO		Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).
3	NTC		Stepwise (Criteria: Probability-of-F-to-enter <= .050, Probability-of-F-to-remove >= .100).

a. Dependent Variable: ESFUERZO

Se describen las variables a utilizar en el modelo.

Tabla 10. Descripción de los elementos del modelo

Descriptive Statistics			
	Mean	Std. Deviation	N
ESFUERZO	13.5771	6.78225	75
NTA	6.96	4.196	75
NTCO	1.68	1.199	75
NTC	1.64	.536	75

De ese análisis se obtiene entonces las posibles modelos regresiones. A continuación, aparece el resumen, especificando los valores de R y R cuadrado (corregido = 1,00 o no = 1,00 siendo iguales, indica que el no corregido estima bien). La tabla indica que la variable Esfuerzo queda explicada en un 100% por las variables explicativas según el modelo lineal considerado.

Tabla 11. Modelos posibles.

Model Summary ^d				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	1.000 ^a	1.000	1.000	.00254

a. Predictors: (Constant), NTA, NTCO, NTC

d. Dependent Variable: ESFUERZO

Para la presente investigación se usa el modelo definido por (c) dado el interés por ser el de menor error de estimación estándar. Por lo tanto, se describe las variables a utilizar en el modelo.

Tabla 12. Correlación de las variables involucradas.

		Correlations			
		ESFUERZO	NTA	NTCO	NTC
Pearson Correlation	ESFUERZO	1.000	.992	.235	-.028
	NTA	.992	1.000	.113	.006
	NTCO	.235	.113	1.000	-.266
	NTC	-.028	.006	-.266	1.000
Sig. (1-tailed)	ESFUERZO	.	.000	.021	.406
	NTA	.000	.	.167	.481
	NTCO	.021	.167	.	.011
	NTC	.406	.481	.011	.
N	ESFUERZO	75	75	75	75
	NTA	75	75	75	75
	NTCO	75	75	75	75
	NTC	75	75	75	75

La tabla ANOVA presenta el análisis de la varianza del test global, indicando la significación $P = 0,000$, siendo mayor que las expectativas de la Hipótesis nula menor del 0.0001, lo que nos lleva a rechazar la hipótesis nula y concluir que las variables están asociadas linealmente con la variable dependiente.

Tabla 13. Tabla ANOVA

		ANOVA ^a			
Model		Sum of Squares	df	Mean Square	Sig.
1	Regression	3403.920	3	1134.640	176193842.441
	Residual	.000	71	.000	.000 ^d
	Total	3403.920	74		

Por tanto, el modelo de regresión queda de la siguiente forma:

Tabla 14. Tabla de coeficientes

Coefficients ^a					
Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
3 (Constant)	1.390	.001		1125.609	.000
NTA	1.581	.000	.978	22330.855	.000
NTCO	.705	.000	.125	2742.976	.000
NTC	-.001	.001	.000	-2.447	.017

a. Dependent Variable: ESFUERZO

Se determina el modelo predictor de esfuerzo como:

$$\text{ESFUERZO} = 1.390 + 1.581 * \text{NTA} + 0.705 * \text{NTCO} - 0.001 * \text{NTC}$$

4.2 Pruebas de Software

La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación (Pressman & Ph 2010). Unas de las vías más importantes para determinar el estado de la calidad de un producto de software es el proceso de pruebas. Estas están dirigidas a componentes del sistema en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos. Estas permiten detectar la presencia de errores que pudieran generar las entradas o salidas de datos y comportamientos inapropiados durante su ejecución.

Pruebas de Caja Negra: También llamadas funcionales y basadas en especificaciones pretenden examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Al ejecutar las pruebas de Caja Negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable (Eduardo et al. 2015).

Pruebas de Caja Blanca: También llamadas estructurales o de cobertura lógica pretenden investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba. En las pruebas de Caja Blanca se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa (Eduardo et al. 2015).

4.2.1 Planificación de Pruebas para el software SDEffort-BPMN

Las tablas 15 y 16, respectivamente, muestran la estrategia de pruebas seguida en la implementación de la herramienta. Para caja blanca se planifican 4 iteraciones acordes a las fases de desarrollo de la metodología de desarrollo asumida, por lo que las pruebas son realizadas en su mayoría en la fase de Construcción. Teniendo en cuenta un total de 24 métodos a probar por su significación para el correcto funcionamiento del software. Para las pruebas de caja negra o estructural con métodos correspondientes a los casos de uso del sistema, se utiliza clase de partición de equivalencia en el diseño de casos de pruebas alfa y beta.

Tabla 14 Funcionalidades evaluadas por método Caja Blanca

Identificador	<i>Funcionalidades</i>
1	Cargar Archivos
2	Adicionar Archivo
3	Contar Tipos de Archivos
4	Verificar Existencia
5	Eliminar Archivo
6	Eliminar Modelo
7	Procesar Archivo
8	Guardar Procesos en BD
9	Actualizar Tabla Modelos
10	Actualizar Tabla Modelos RLM

11	Buscar Por Nombre
12	Buscar Por Extensión
13	Cargar Modelos de la BD
14	Cargar Modelos de la BD por RLS
15	Cargar Modelos de la BD por RLM
16	Buscar Nombre por RLS
17	Buscar Nombre por RLM
18	Create PDF
19	Exportar a EXCEL.
20	Exportar a EXCEL evaluados por RLS
21	Exportar a EXCEL evaluados por RLM
22	Crear Grafico Esfuerzo vs Complejidad
23	Crear Grafico Complejidad
24	Crear Grafico Esfuerzo

4.2.1.1 Resumen de pruebas de caja blanca realizadas

La tabla 15 muestra la cantidad de pruebas realizadas en cada iteración del ciclo de desarrollo, las respectivas funcionalidades, evaluadas en cada caso, haciendo mención por el identificador definido en la tabla 14.

Tabla 15 Resumen de pruebas de caja blanca realizadas

<i>Iteraciones</i>	<i>Cantidad de pruebas realizadas</i>	<i>Funcionalidades(identificador)</i>
1	6	1, 2, 3, 4, 5, 6.
2	17	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17.
3	15	7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.

4	8	17, 18, 19, 20, 21, 22, 23, 24.
---	---	---------------------------------

4.2.1.2 Resumen de pruebas de caja negra realizadas

La tabla 16 muestra la cantidad de pruebas realizadas en cada una de las fases de la metodología asumida y los casos de usos asociados.

Tabla 16 Resumen de pruebas de caja negra realizadas

<i>Iteraciones</i>	<i>Cantidad de pruebas realizadas</i>	<i>Casos de uso Asociados</i>
1	2	Cargar Archivo MPN
		Eliminar Archivo MPN
2	2	Interpretar Archivo MPN
		Consultar Datos
3	3	Exportar datos a PDF
		Exportar datos a Excel
		Generar Estadísticas de datos
4	7	Gestionar Archivo MPN

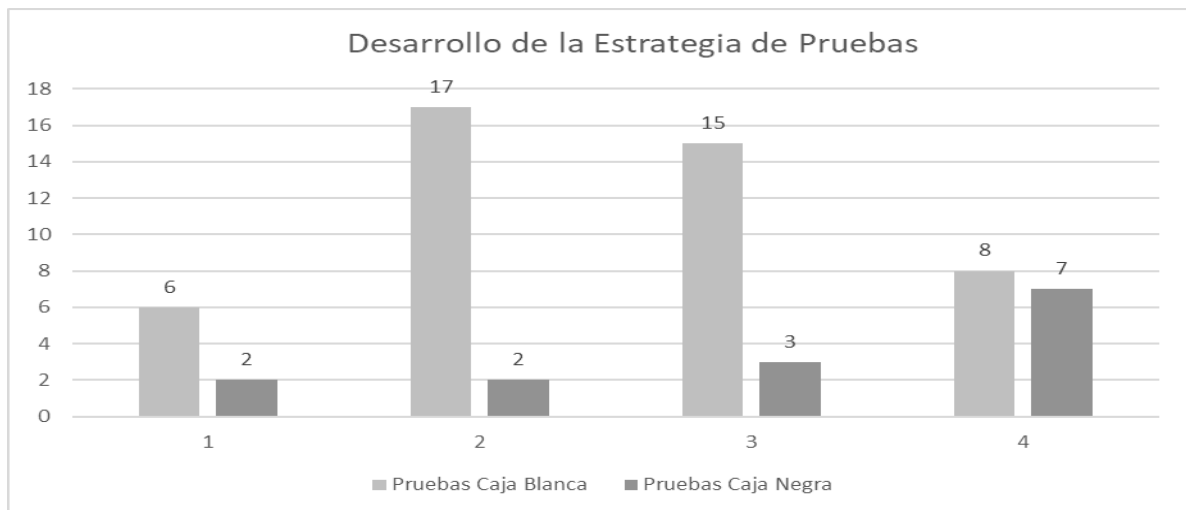


Figura 21 Desarrollo de la Estrategia de Pruebas

4.2.2 Pruebas de Caja negra. Diseño de los casos de prueba.

Se muestra a continuación los casos de pruebas realizados correspondientes a diferentes casos de uso con el fin de obtener las clases válidas y no válidas y la respuesta del sistema ante estas.

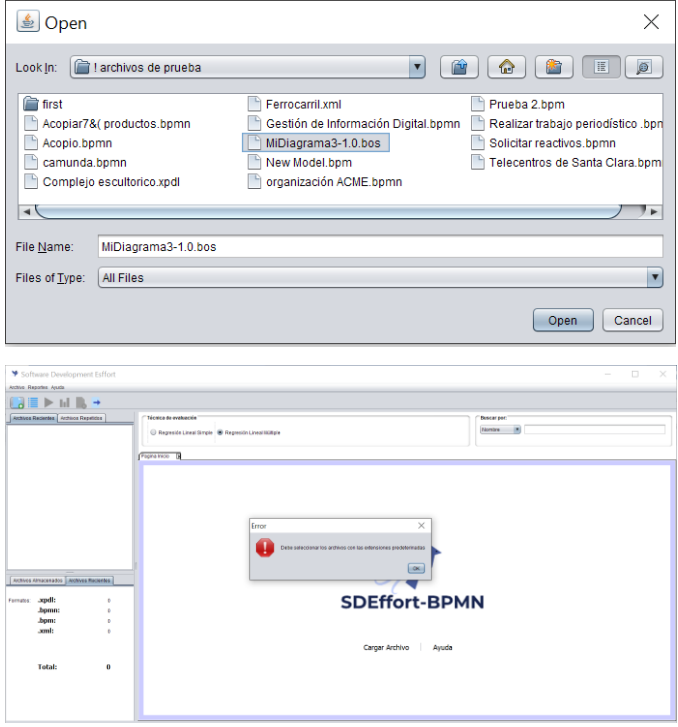
Caso de prueba para el Caso de Uso “Cargar Archivo MPN”

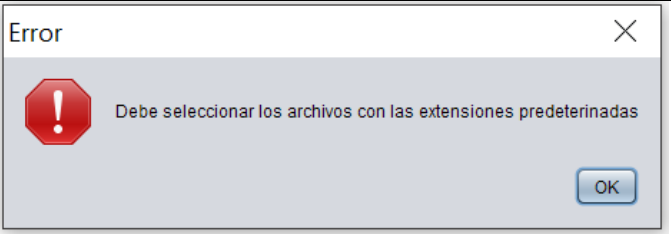
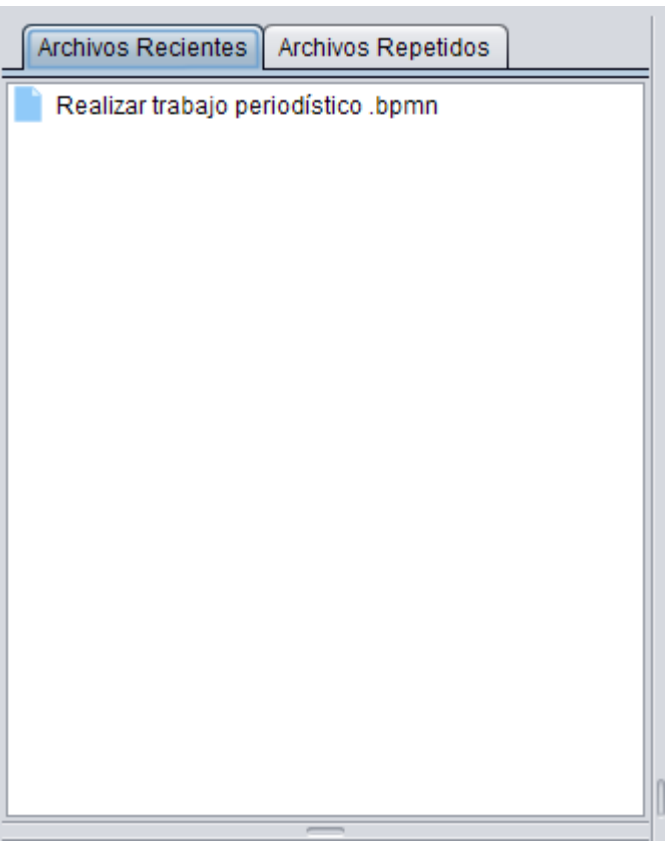
Tabla 17. Caso de prueba para el Caso de Uso “Cargar Archivo MPN”

Condición de entrada	Clases válidas	Clases No válidas
Nombre de archivo	1. Campo alfanumérico.	2. Nulo (en blanco).
Tipo de archivo	3. bpmn 4. bpm 5. xpdl 6. xml	7. Otra extensión.

Resultado de la ejecución de las clases de entrada de datos propuesta:

Tabla 18. Resultado de la ejecución de las clases de entrada

Casos de pruebas	Clases	Resultado
<p>Nombre de archivo: <u>Archivo</u></p> <p>Tipo de archivo: <u>bos</u></p>	<p>1</p> <p>7</p>	

		 <p>Error</p> <p>Debe seleccionar los archivos con las extensiones predeterminadas</p> <p>OK</p>
<p>Nombre de archivo:</p> <p><u>Realizar trabajo</u> <u>periodístico</u></p> <p>Tipo de archivo: <u>bpmn</u></p>	<p>1</p> <p>3</p>	 <p>Archivos Recientes Archivos Repetidos</p> <p>Realizar trabajo periodístico .bpmn</p>

Caso de prueba para el Caso de Uso “Consultar Datos”

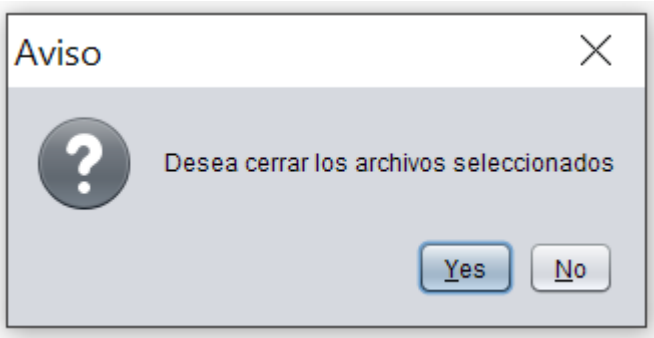

Tabla 19. Caso de prueba para el Caso de Uso “Consultar Datos”

Condición de entrada	Clases válidas	Clases No válidas
Archivo	1. ModeloPN	2. Nulo.

Resultado de la ejecución de las clases de entrada de datos propuesta:

Tabla 20. Resultado de la ejecución de las clases de entrada

Casos de pruebas	Clases	Resultado
	es	

Archivo: <u>Acopio.bpmn</u>	1	
Archivo: _____	2	

Caso de prueba para el Caso de Uso “Interpretar MPN”

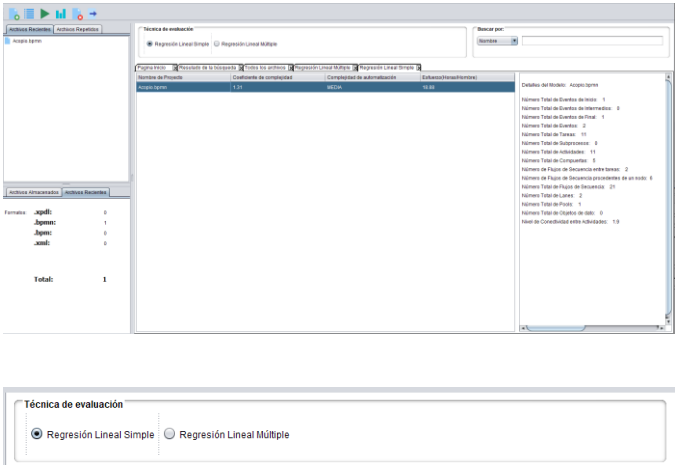
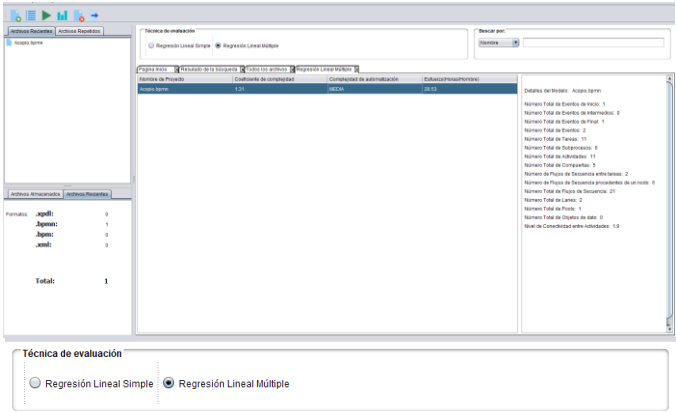
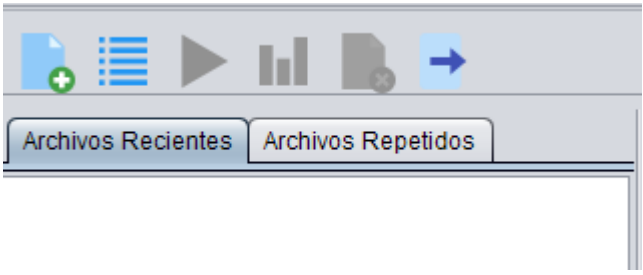
Tabla 23. Caso de prueba para el Caso de Uso “Interpretar MPN”

Condición de entrada	Clases válidas	Clases No válidas
Archivo	1. ModeloPN	2. Nulo.
Tipo de Regresión	3. Regresión Lineal Simple 4. Regresión Lineal Múltiple	

Resultado de la ejecución de las clases de entrada de datos propuesta:

Tabla 22. Resultado de la ejecución de las clases de entrada

Casos de pruebas	Clases	Resultado
	es	

<p>Archivo: <u>Acopio.bpmn</u></p> <p>Tipo de Regresión: <u>Regresión Lineal Simple</u></p>	<p>1</p> <p>3</p>	
<p>Archivo: <u>Acopio.bpmn</u></p> <p>Tipo de Regresión: <u>Regresión Lineal Múltiple</u></p>	<p>1</p> <p>4</p>	
<p>Archivo: _____</p>	<p>2</p>	

4.2.3 Pruebas de Caja blanca

De igual forma se realizaron pruebas a todas las funcionalidades del sistema en diferentes iteraciones de su desarrollo. A continuación, se presentan algunos de los resultados obtenidos.

Caso de prueba para el método “cargarArchivo()” correspondiente al caso de uso “Cargar Archivo MPN”

Se le realiza el test al método “cargarArchivo()”, donde se verifica la capacidad del sistema para entrar archivos, se muestra con el método *assertNotNull(object)* que comprueba que el objeto obtenido es nulo o no.

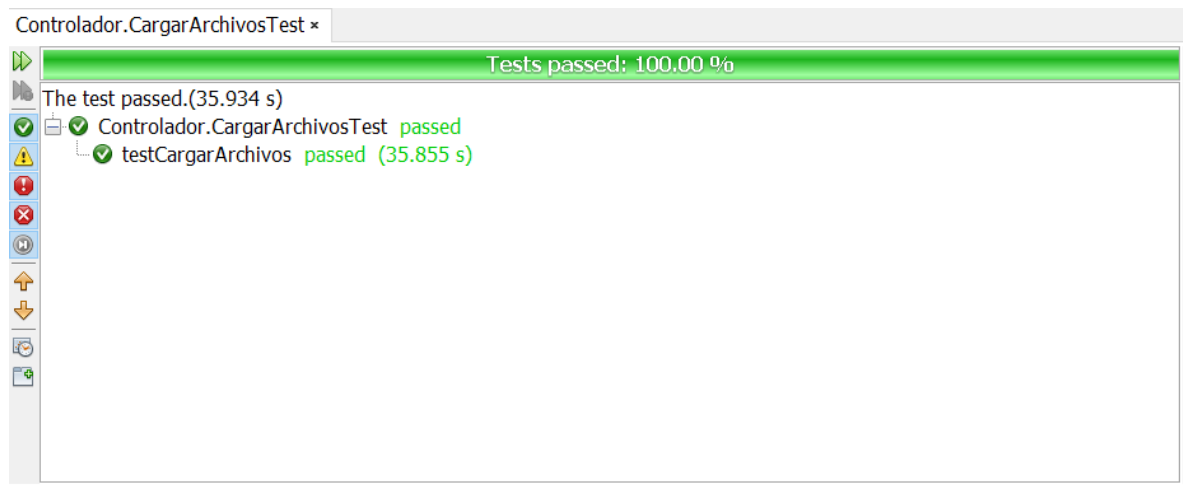


Figura 22. Test Cargar Archivo.

Caso de prueba para el método “adicionarArchivo()” relacionado al caso de uso “Interpretar Archivo MPN”

Se le realiza el test al método “adicionarArchivo” donde el programa debe ser capaz de obtener un archivo, determinar qué tipo de archivo es y devolver un listado de archivo. Se utiliza el método *assertEquals(valor esperado, valor obtenido)* demostrando que el sistema dado un juego de datos es capaz de devolver el resultado correcto. Se muestra mediante la función *println()* el resultado esperado y el resultado que devuelve el método de la clase *GestionArchivo()*, coincidiendo ambos resultados y por tanto el test pasa satisfactoriamente.

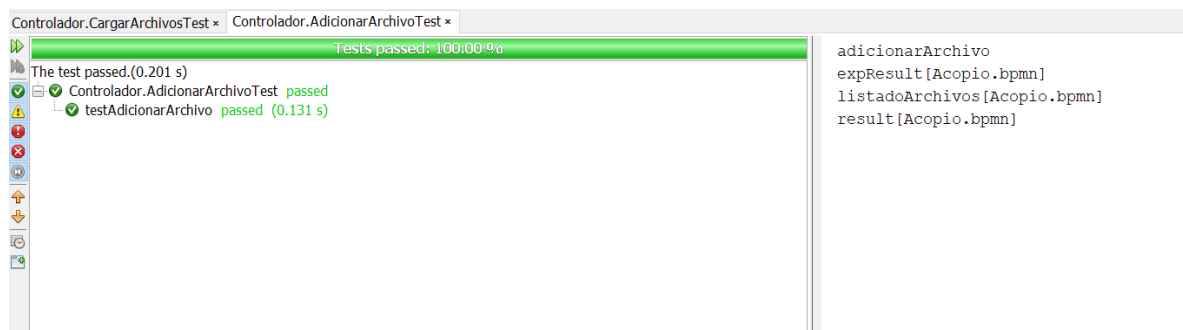


Figura 23. Test Adicionar Archivo.

4.3 Conclusiones del capítulo.

Se analiza el modelo a utilizar en la confección de la herramienta y se focaliza teniendo por resultado una ecuación de regresión lineal múltiple para la obtención del esfuerzo de desarrollo de proyectos de software. Con la planificación de las pruebas realizadas al software se cubrió en un 100% las funcionalidades del mismo. Se utilizaron técnicas de caja blanca y caja negra, las pruebas de caja blanca se realizaron con la herramienta JUnit obteniendo resultados satisfactorios.

CONCLUSIONES

1. Se diseña la herramienta SDEffort-BPMN utilizando como arquitectura del sistema: Modelo Vista Controlador (MVC), brindando limpieza, claridad y robustez al software. Como patrón de diseño Strategy se crea una estrategia para la interpretación de los diferentes archivos, brindando al sistema adaptabilidad.
2. La obtención del modelo predictivo del esfuerzo da como resultado una ecuación basada en las tres métricas que más influyentes de forma conjunta y lineal sobre el esfuerzo: NTC, NTA y NTCO.
3. La implementación del sistema SDEffort-BPMN permitió la incorporación del Nuevo modelo de estimación del esfuerzo analizado, la interpretación de nuevos formatos de salida de herramientas de modelado y la generación de estadísticas graficas importantes para la toma de decisions por parte de los expertos.
4. La estrategia de prueba utilizada permitió la comprobación de cada una de las funcionalidades del software desde diferentes perspectivas y mostrando resultados satisfactorios.

REFERENCIAS BIBLIOGRÁFICAS

- ANTÚNEZ BARBOSA, T. et al., 2016. Estimación de costos de desarrollo, caso de estudio: Sistema de Gestión de calidad del Reactor TRIGA Mark III. *Revista Cubana de Ciencias Informáticas*, 10(1), pp.215–228. Available at: [http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path\[\]=1092](http://rcci.uci.cu/index.php?journal=rcci&page=article&op=view&path[]=1092).
- Bandomo Toledo, L., 2014. Procedimiento para evaluar el nivel de complejidad de los procesos de negocio a partir de su representación gráfica.
- Carrasquilla-batista, A., Chacón-rodríguez, A. & Núñez-, K., 2016. Regresión lineal simple y múltiple : aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal Simple and multiple regression : application in the prediction of natural variables related to microalgae growing process.
- Design, S. & Institutions, H.E., 2018. Revista Internacional de Investigación e Innovación Tecnológica. , (35), pp.1–20.
- E.V.A., U., 2017. Flujo de Trabajo Modelo del Negocio - EcuRed.
- Edeki, C., 2013. Agile Unified Process. *International Journal of Computer Science and Mobile Applications*, 1, pp.13–17.
- Eduardo, M. et al., 2015. Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2 A formal model for the functional test of software to achieve maturity integrated level 2 Modelo formal de provas funcionais de software para alcançar o Nível de Madureza Integrado 2.
- Esquivel Ariz, M., 2014. INTERVAR. Sistema para la interpretación de archivos XPDL correspondientes a procesos de negocio modelados con notación gráfica BPMN.
- Ferreira, G., Gálvez, D., Quintero, L. y Antón, J., 2015. ESFUERZO EN PROYECTOS DE DESARROLLO DE SOFTWARE ARTIFICIAL INTELLIGENCE TECHNIQUES

- FOR ESTIMATING THE Resumen Palabras claves : Keywords : *Iberoamerican Journal of Project Management*, 6(1), pp.1–22.
- Ferreras Bustamante, D.A.F., 2018. Aplicación para la obtención y procesado de datos de vehículo eléctrico.
- García Peñalvo, F.J. & García Holgado, A., 2018. Modelo de dominio.
- Giraldo, J., Jovani, J. & Tabares, S., 2017. Modelo para optimizar el proceso de gestión de negocio combinando minería de procesos con inteligencia de negocios desde almacenes de datos. *Revista Espacios*, 38(Nº 02), pp.1–10. Available at: <http://www.revistaespacios.com/a17v38n02/17380209>.
- Granados Hondares, Y., 2015. 7-7-15Modelo para estimar el esfuerzo que demanda la automatización de Procesos de Negocio.
- Gutiérrez Acea, Y. de la C., 2018. Mercado de datos: Análisis de tendencias para la actividad de desarrollo- producción UCI. , pp.61–65.
- Hernández, A.P., 2015. Propuesta de tutorial interactivo en línea para apoyar la enseñanza del lenguaje de consulta estructurado. , (112), pp.1658–1675.
- Molina, J.G. et al., 2018. De los Procesos del Negocio a los Casos de Uso.
- O’Farrill Fernández, L., 2012. Procedimiento para evaluar el nivel de complejidad de la informatización de los procesos de negocio a partir de su representación gráfica. , p.92.
- Patricia, I. et al., 2019. Phonological consciousness and lecturing of third grade students of primary basic education *. , 11(1), pp.165–174.
- Pressman, R.S. & Ph, D., 2010. *Ingeniería del software* Séptima ed.,
- Remón, C.A., 2017. Estimación de Esfuerzo en el Desarrollo de Software a partir de Especificación de Requerimientos.
- Rojas, E.M. et al., 2018. Sistema de Gestión Académica a través del desarrollo de Modelo-Vista-Controlador. , pp.1083–1094.
- Rolón Aguilar, E., 2009. Medidas para Asegurar la Calidad de los Modelos de Proceso de Negocio.
- ROMERO ESPINOZA, O.O., 2018. GESTIÓN DE PROYECTOS DE TECNOLOGÍAS.
- Rondón, Y.Q., Domínguez, L.C. & Berenguer, A.D., 2009. Diseño de la Base de Datos para Sistemas de Digitalización y Gestión de Medias. ,

8(15), pp.17–25.

Standish, T. & Report, G., The Standish Group Report.

VILLARRAGA, A.E. & ZAMUDIO, N.Y.M., 2017. SISTEMA TELEMATICO PARA
LA ESTIMACION DE ESFUERZO DE PROYECTOS DE SOFTWARE.

ANEXOS

Anexo I. Manual de Usuario.

Introducción

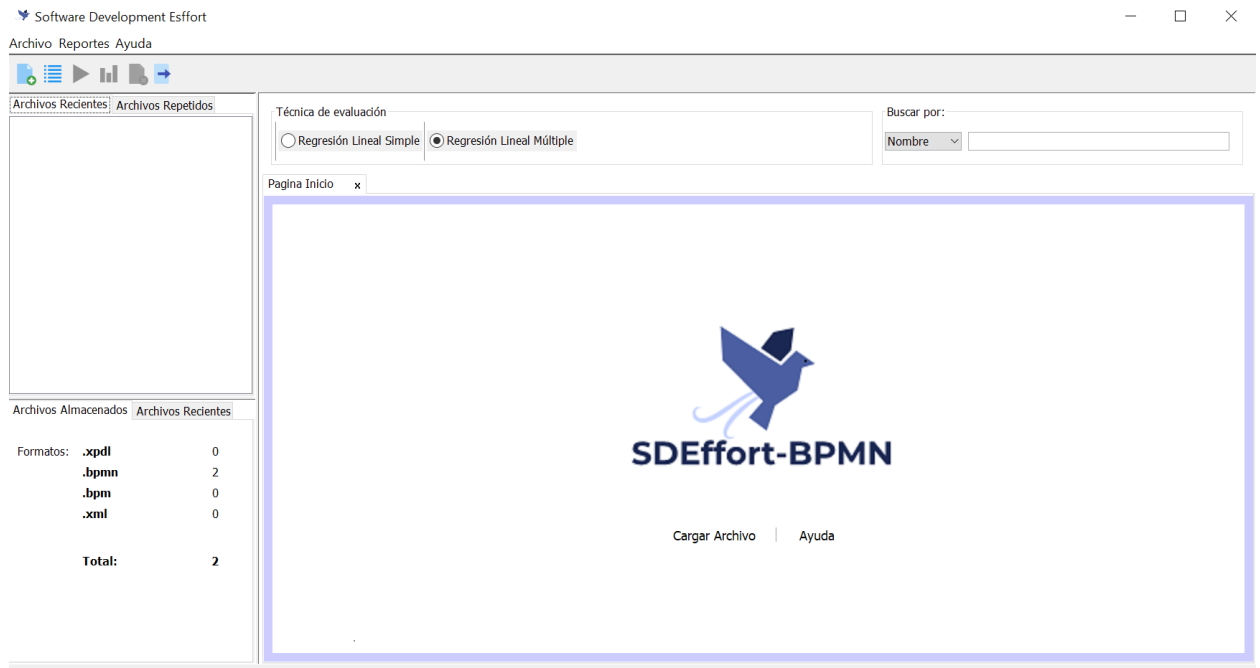
SDEffort-BPMN es una herramienta basada en modelos de procesos de negocio, como entrada inicial al proceso de desarrollo de software, por lo que constituye una variante novedosa para lograr la estimación temprana del esfuerzo de desarrollo.

SDEffort-BPMN tiene como objetivo ofrecer una predicción del esfuerzo de desarrollo de software, en una etapa temprana de su desarrollo, utilizando técnicas de regresión lineal múltiple y regresión lineal simple, con el fin de brindar opciones confiables y que se puedan acercar más a la predicción que el experto en cada caso pudiera requerir, mediante la extracción de métricas del esfuerzo de desarrollo y a partir de la medición de las mismas.



SDEffort-BPMN

La herramienta está compuesta por varios paneles, dando claridad en su manejo y permitiendo tener todos a la vista del usuario.



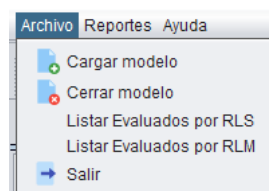
Menú

Cuenta con una barra de menú en la cual se conciben las funcionalidades de la aplicación, de forma que el usuario puede acceder a través de estas a las acciones que desea realizar dentro de

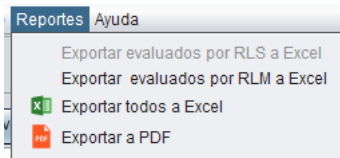


la misma.

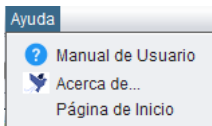
El menú Archivo brinda opciones descargar archivos, cerrar los archivos, listar solo los archivos evaluados por las diferentes técnicas que brinda la herramienta y por ultimo abandonar la herramienta.



El menú Reportes permite obtener los archivos evaluados en forma de Excel, tanto los evaluados por cada técnica, como el conjunto de evaluados en general. Brinda un reporte realizado en PDF donde se exponen las características de las técnicas de evaluación y el resultado de los datos evaluados en forma de tabla.



El menú Ayuda muestra información de cómo usar la herramienta, portando un manual de usuario e información de los desarrolladores. También y como parte de accesibilidad para el usuario se



brinda el link a la página de Inicio.

Barra de herramientas y funcionalidades

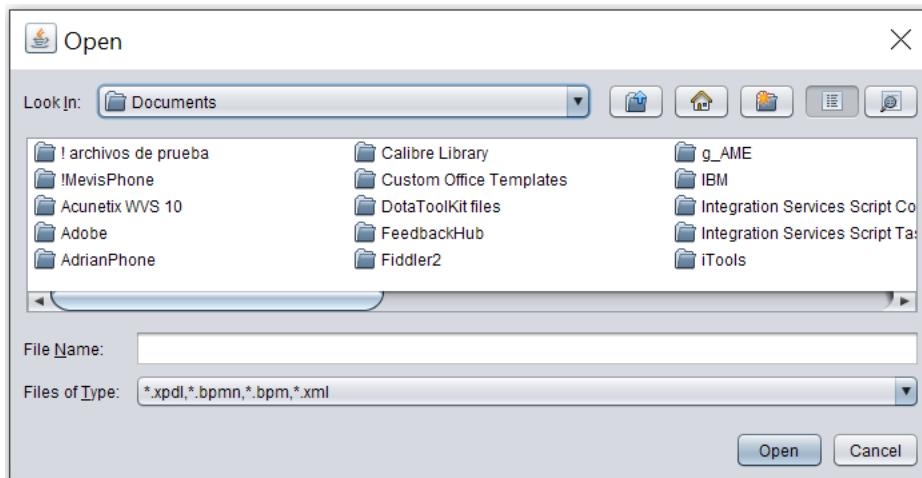
La barra de herramientas tiene acceso las principales funcionalidades de la aplicación



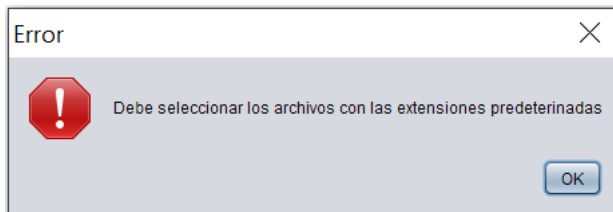
- Cargar Archivo
- Listar Archivos evaluados
- Evaluar Archivos
- Ver estadística de los archivos evaluados
- Cerrar Archivos (no es eliminar los archivos evaluados)
- Abandonar la herramienta

Cargar Archivos

Se cargan los archivos a evaluar, siendo estos de tipo predefinido. Se actualizan las estadísticas de los archivos.



Si se trata de cargar un archivo que no es de los predeterminados se lanza un error.



De forma contraria se cargan los archivos satisfactoriamente, y se actualizan las estadísticas, siendo los Archivos recientes los cargados y los Archivos Almacenados los existentes en la base de datos.

Archivos Almacenados	Archivos Recientes
Formatos:	.xpdI 1
	.bpmn 3
	.bpm 0
	.xml 0
Total:	4

Archivos Recientes	Archivos Repetidos
Acopio.bpmn	
Acopiar7&(productos.bpmn	
Complejo escultorico.xpdI	
Ferrocarril.xml	
Gestión de Información Digital.bpmn	
New Model.bpm	
Prueba 2.bpm	
Realizar trabajo periodístico .bpmn	
Solicitar reactivos.bpmn	
Telecentros de Santa Clara.bpmn	
camunda.bpmn	
organización ACME.bpmn	

Archivos Almacenados	Archivos Recientes
Formatos:	.xpdI: 1
	.bpmn: 8
	.bpm: 2
	.xml: 1
Total:	12

Listar Archivos

Se listan los archivos evaluados y guardados en la base de datos, de igual forma desde los menús Listar Evaluados RLS y Listar Evaluados RLM se puede acceder a los datos evaluados por las distintas técnicas.

Página Inicio

☒ Regresión Lineal Múltiple

☒ Regresión Lineal Simple

☒ Evaluados por Regresión Simple

☒ Evaluados por Regresión Múltiple


Nombre	NTEI	NTEIn	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN	NTFS	NTC	NTP	NTOD	NCA	CCOM	CAPN	Esfuerzo...	Evaludor
Acopiar7...	1	0	2	3	27	0	27	4	7	3	37	7	1	0	1.37	1.19	NO CAL...	42.06	RLM
Acopio	1	0	1	2	11	0	11	5	2	6	21	2	1	0	1.9	1.31	MEDIA	18.88	RLM
Complej...	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
Ferrocarril	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
Gestión ...	1	0	1	2	3	0	3	1	2	2	6	2	1	0	2.0	1.5	BAJA	6.89	RLM
New Mo...	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
Prueba 2	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
Realizar ...	1	0	1	2	8	0	8	1	5	1	11	4	1	0	1.37	1.22	BAJA	13.94	RLM
Solicitar ...	1	0	1	2	11	0	11	1	7	2	14	3	1	0	1.27	1.16	NO CAL...	19.46	RLM
Telecent...	1	0	2	3	24	0	24	1	11	2	27	3	1	0	1.12	1.08	NO CAL...	39.39	RLM
camunda	1	0	2	3	3	0	3	1	0	2	6	0	1	0	0.0	1.5	BAJA	6.38	RLM
organiza...	2	0	2	4	7	1	8	2	2	3	13	2	2	0	1.62	1.3	NO CAL...	14.09	RLM

Eliminar Permanentemente

Solo desde la ventana que se accede desde el botón Listar de la barra de herramienta se puede eliminar los archivos permanentemente de la base de datos. Para ello si existen elementos en la base de datos se selecciona el o los elementos de la tabla y se elimina, de lo contrario si no existen en la base de datos se le hace saber al usuario.

Página Inicio	<input checked="" type="checkbox"/>	Todos los archivos	<input checked="" type="checkbox"/>																
Nombre	NTEI	NTEIn	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN	NTFS	NTC	NTP	NTOD	NCA	CCOM	CAPN	Esfuerzo...	Evalúador
Acopio	1	0	1	2	11	0	11	5	2	6	21	2	1	0	1.9	1.31	MEDIA	18.88	RLM
Acopiar7...	1	0	2	3	27	0	27	4	7	3	37	7	1	0	1.37	1.19	NO CAL...	42.06	RLM
Complej...	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
camunda	1	0	2	3	3	0	3	1	0	2	6	0	1	0	0.0	1.5	BAJA	6.38	RLM


Seleccione los archivos que desea eliminar de la base de datos

 Eliminar


Página Inicio ☒ Todos los archivos ☒

Nombre	NTEI	NTEIn	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN	NTFS	NTC	NTP	NTOD	NCA	CCOM	CAPN	Esfuerzo...	Evaluador
Acopio	1	0	1	2	11	0	11	5	2	6	21	2	1	0	1.9	1.31	MEDIA	18.88	RLM
Acopiar7...	1	0	2	3	27	0	27	4	7	3	37	7	1	0	1.37	1.19	NO CAL...	42.06	RLM
Complej...	1	0	2	3	6	1	7	3	0	5	13	2	2	0	0.0	1.3	NO CAL...	13.17	RLM
camunda	1	0	2	3	3	0	3	1	0	2	6	0	1	0	0.0	1.5	BAJA	6.38	RLM

Aviso ✕

 Desea eliminar permanentemente los archivos de la base de datos

Seleccione los archivos que desea eliminar de la base de datos

 Eliminar

Página Inicio ☒ Todos los archivos ☒

Nombre	NTEI	NTEIn	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA	NFSN	NTFS	NTC	NTP	NTOD	NCA	CCOM	CAPN	Esfuerzo...	Evaluador
--------	------	-------	------	-----	-----	-----	-----	------	------	------	------	-----	-----	------	-----	------	------	-------------	-----------

No existen elementos para mostrar

Evaluar Archivos

Para evaluar deben estar cargados con anterioridad los archivos y se debe tener en cuenta el tipo de técnica de evaluación, que por defecto se muestra seleccionada la técnica de Regresión Lineal Múltiple.

Archivos RecientesArchivos Repetidos

Acopiar7&(productos.bpmn
Acopio.bpmn
Complejo escultorico.xpd
Ferrocaril.xml
Gestión de Información Digital.bpmn
New Model.bpm
Prueba 2.bpm
Realizar trabajo periodístico .bpmn
Solicitar reactivos.bpmn
Telecentros de Santa Clara.bpmn
camunda.bpmn
organización ACME.bpmn

Archivos AlmacenadosArchivos Recientes

Formatos: **.xpd:** 1
.bpmn: 8
.bpm: 2
.xml: 1

Total: 12

Técnica de evaluación

☐ Regresión Lineal Simple
☒ Regresión Lineal Múltiple

Buscar por:

Nombre

Página Inicio

☒ Regresión Lineal Múltiple
☒

Nombre de Proyecto	Coefficiente de complejidad	Complejidad de automatización	Esfuerzo(Horas/Hombre)
Acopiar7&(productos.bpmn	1.19	NO CALCULADA	42.06
Acopio.bpmn	1.31	MEDIA	18.88
Complejo escultorico.xpd	1.3	NO CALCULADA	13.17
Ferrocaril.xml	1.3	NO CALCULADA	13.17
Gestión de Información Digital.bpmn	1.5	BAJA	6.89
New Model.bpm	1.3	NO CALCULADA	13.17
Prueba 2.bpm	1.3	NO CALCULADA	13.17
Realizar trabajo periodístico .bpmn	1.22	BAJA	13.94
Solicitar reactivos.bpmn	1.16	NO CALCULADA	19.46
Telecentros de Santa Clara.bpmn	1.08	NO CALCULADA	39.39
camunda.bpmn	1.5	BAJA	6.38
organización ACME.bpmn	1.3	NO CALCULADA	14.09

Se puede conocer detalles de los modelos evaluados seleccionando el modelo y se observan en el panel lateral a la tabla.

Página Inicio

☒ Regresión Lineal Múltiple
☒

Nombre de Proyecto	Coefficiente de complejidad	Complejidad de automatización	Esfuerzo(Horas/Hombre)
Acopiar7&(productos.bpmn	1.19	NO CALCULADA	42.06
Acopio.bpmn	1.31	MEDIA	18.88
Complejo escultorico.xpd	1.3	NO CALCULADA	13.17
Ferrocaril.xml	1.3	NO CALCULADA	13.17
Gestión de Información Digital.bpmn	1.5	BAJA	6.89
New Model.bpm	1.3	NO CALCULADA	13.17
Prueba 2.bpm	1.3	NO CALCULADA	13.17
Realizar trabajo periodístico .bpmn	1.22	BAJA	13.94
Solicitar reactivos.bpmn	1.16	NO CALCULADA	19.46
Telecentros de Santa Clara.bpmn	1.08	NO CALCULADA	39.39
camunda.bpmn	1.5	BAJA	6.38
organización ACME.bpmn	1.3	NO CALCULADA	14.09

Detalles del Modelo: Acopio.bpmn

Número Total de Eventos de Inicio: 1
Número Total de Eventos de Intermedios: 0
Número Total de Eventos de Final: 1
Número Total de Eventos: 2
Número Total de Tareas: 11
Número Total de Subprocesos: 0
Número Total de Actividades: 11
Número Total de Puertas: 5
Número de Flujos de Secuencia entre tareas: 2
Número de Flujos de Secuencia procedentes de un nodo: 6
Número Total de Flujos de Secuencia: 21
Número Total de Lanes: 2
Número Total de Pools: 1
Número Total de Objetos de dato: 0
Nivel de Conectividad entre Actividades: 1.9

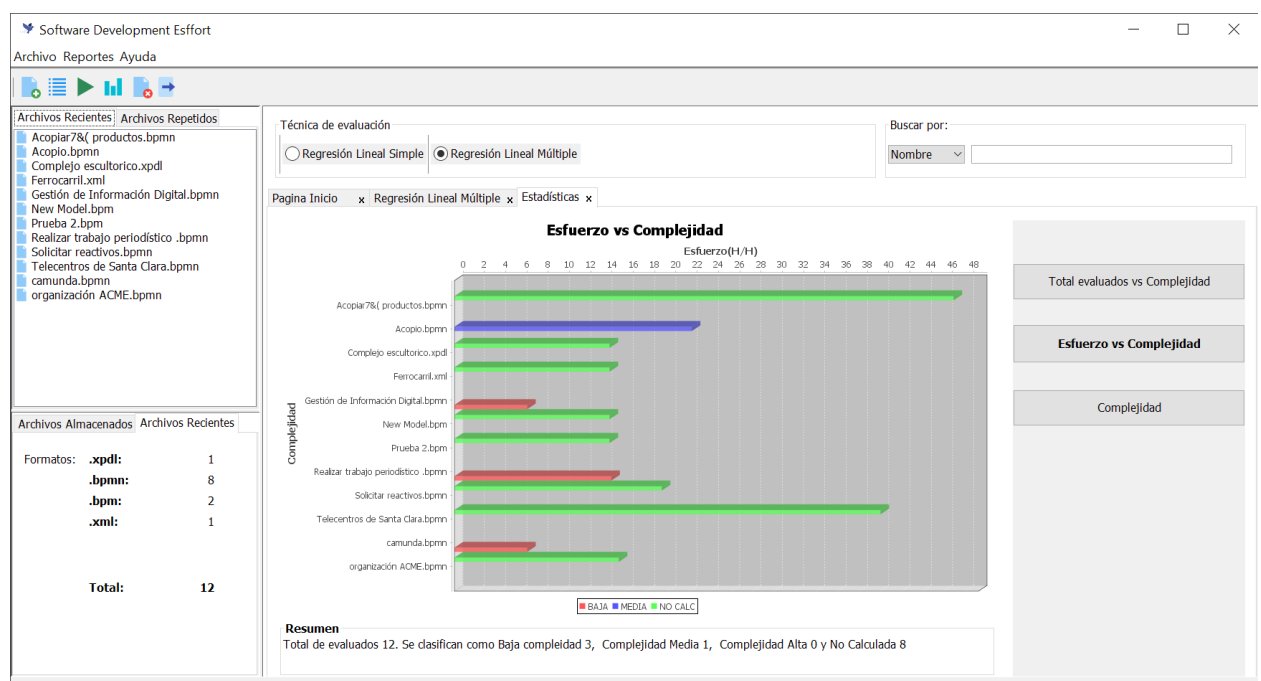
Estadísticas

Las estadísticas que se muestran se refieren a los datos evaluados en el momento y no a datos evaluados y guardados en la base de datos con anterioridad. Las relaciones de las gráficas muestran Esfuerzo evaluado en comparación con otras métricas influyentes en los modelos.



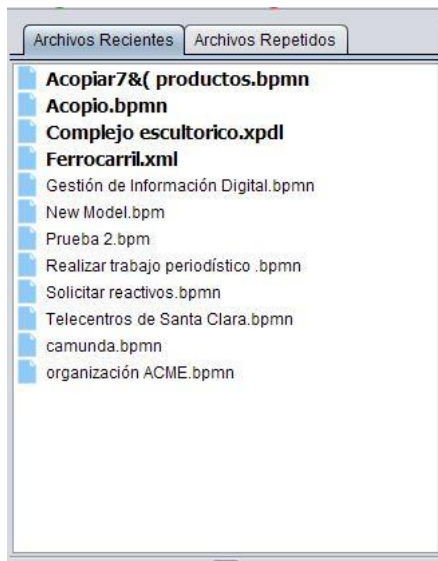
Solo está disponible la opción de luego de evaluar al menos un archive, de lo contrario se muestra la opción deshabilitada.

Además, existen varios gráficos de barra que muestran estadísticas en las diferentes consideraciones que el usuario como experto puede necesitar y un resumen de los datos evaluados.



Cerrar Archivos

Para cerrar los archivos con los que no se desean trabajar una vez cargados se seleccionan y se eliminan de la lista de archivos interpretables. Esto no quiere decir que se eliminan físicamente.



Anexo II. Resultados de la obtención del modelo de predicción del esfuerzo.

Tabla de correlación de Pearson para las variables cuantitativas.

		Correlations									
		ESFUE RZO	NTEI	NTEIN	NTEF	NTE	NTT	NTS	NTA	NTCO	NFSA
Pearson Correlation	ESFUER	1.000	.	-.061	.506	.391	.990	.149	.992	.235	.873
	ZO										
	NTEI	.	1.000
	NTEIN	-.061	.	1.000	-.017	.543	-.064	.099	-.059	-.022	-.063
	NTEF	.506	.	-.017	1.000	.830	.513	.066	.514	.022	.418
	NTE	.391	.	.543	.830	1.000	.395	.111	.399	.006	.315
	NTT	.990	.	-.064	.513	.395	1.000	.070	.999	.100	.929
	NTS	.149	.	.099	.066	.111	.070	1.000	.116	.283	-.094
	NTA	.992	.	-.059	.514	.399	.999	.116	1.000	.113	.921
	NTCO	.235	.	-.022	.022	.006	.100	.283	.113	1.000	-.224
	NFSA	.873	.	-.063	.418	.315	.929	-.094	.921	-.224	1.000
	NFSN	.341	.	-.029	.168	.125	.215	.217	.224	.973	-.108
	NTFS	.938	.	.009	.550	.467	.890	.183	.895	.507	.686
	NTC	-.028	.	.037	-.138	-.095	.005	.010	.006	-.266	.117
	NTP	.017	.	-.038	-.025	-.042	-.026	.569	.001	.129	-.122
	NTOD
	NCA	.048	.	.120	.096	.148	-.003	.238	.008	.320	-.135
	CCOM	-.392	.	.384	.000	.214	-.448	-.096	-.450	.386	-.541
Sig. (1-tailed)	ESFUER	.	.000	.303	.000	.000	.000	.101	.000	.021	.000
	ZO										
	NTEI	.000	.	.000	.000	.000	.000	.000	.000	.000	.000
	NTEIN	.303	.000	.	.441	.000	.292	.198	.307	.426	.296
	NTEF	.000	.000	.441	.	.000	.000	.287	.000	.427	.000
	NTE	.000	.000	.000	.000	.	.000	.172	.000	.479	.003
	NTT	.000	.000	.292	.000	.000	.	.277	.000	.197	.000
	NTS	.101	.000	.198	.287	.172	.277	.	.160	.007	.211
	NTA	.000	.000	.307	.000	.000	.000	.160	.	.167	.000
	NTCO	.021	.000	.426	.427	.479	.197	.007	.167	.	.027
	NFSA	.000	.000	.296	.000	.003	.000	.211	.000	.027	.

[illegible]

