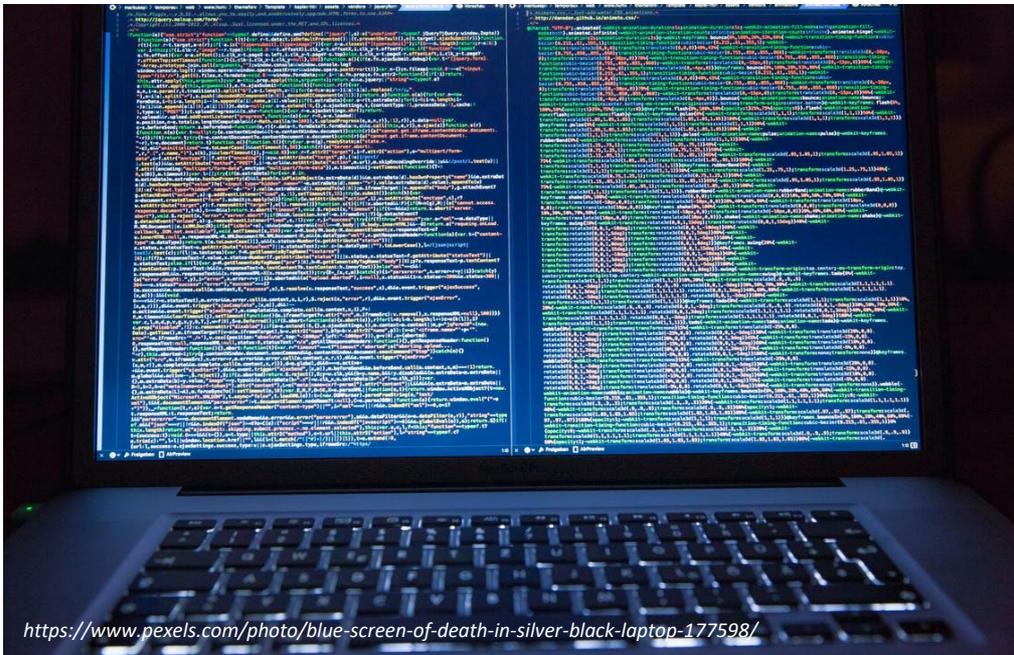


# Análisis de herramientas software para el modelado de procesos de software



**Álvaro Augusto Muñoz Franco.** Estudiante del pregrado en Administración de Sistemas Informáticos de la Universidad Nacional de Colombia, sede Manizales. **Correo electrónico:** aamuñozf@unal.edu.co

**Leonardo Bermón Angarita.** PhD. en Ingeniería Informática, Mg. en Informática. Profesor del pregrado en Administración de Sistemas Informáticos. Director del departamento de Informática y Computación de la Universidad Nacional de Colombia, sede Manizales. **Correo electrónico:** lbermona@unal.edu.co

---

## Cómo citar este artículo

Franco Muñoz, A.A. & Bermón Angarita, L. (2014) Análisis de herramientas software para el modelado de procesos de software. NOVUM, (4), p.p 38-56.

## Resumen

El proceso de desarrollo de software es un proceso no estandarizado a través del cual las organizaciones persiguen un único objetivo que es crear herramientas y aplicaciones software con mayor rapidez, pero con una mayor calidad. Por lo tanto, las organizaciones se han enfocado en seleccionar los modelos de ciclo de vida de desarrollo existentes y adaptarlos al desarrollo de sus procesos organizacionales. Para llevar a cabo los diferentes procesos que componen la infraestructura del desarrollo de software, se ha vuelto de gran importancia para las organizaciones el uso de herramientas que permitan llevar a cabo su modelado. En el presente trabajo se muestran herramientas de este tipo, se hace una comparación de éstas y propone la creación de una herramienta que ofrezca mayores beneficios al momento de modelar un proyecto, teniendo en cuenta las ventajas que ofrecen las herramientas analizadas.

**Palabras clave:** Proceso software; Modelo de procesos; Herramientas software.

## Abstract

The software development process is a non-standardized process through which organizations pursue a unique goal that is to create faster software tools and applications but with superior quality. Therefore, organizations have focused on selecting existing development lifecycle models and adapting them to the development of their organizational processes. To carry out the various processes that make up the software development infrastructure, it has become of great importance for organizations to use the tools that enable them to carry out their model. In this work, the tools for process modeling are described, a comparison of these and proposes is made and the creation of a tool that offers greater benefits to the model of a project, taking into account the advantages offered by the tools analyzed. **Keywords:** Software process; Process model; Software tool.

## 1. Introducción

Un proceso de software puede definirse como la ejecución de un conjunto de actividades que se encuentran relacionadas y que al llevarse a cabo de acuerdo a los planes establecidos conllevan a la obtención de un producto de software, que cumple con los requerimientos del cliente y satisface la necesidad o necesidades que provocaron la decisión de su creación (Sutton, 2000) (Fuggetta, A. y Di Nitto, 2014).

En la ejecución de las diferentes fases del proceso de Ingeniería de Software, intervienen diversos elementos que posibilitan el alcance de los objetivos del desarrollo, entre ellos se destaca por su gran importancia, el uso de diferentes herramientas software, que sirven de apoyo a la infraestructura del proceso, brindando la posibilidad de realizar el modelado de los diferentes procesos que requiere la Ingeniería de Software (Fugetta, 2000). Éstas consideradas tecnologías de software posibilitan el desarrollo de un producto software específico, para los diferentes ámbitos empresariales.

Al hablar del proceso de desarrollo de software no se hace referencia a la existencia de un único proceso, ya que existen diversos procesos que involucran una serie de metodologías

que las organizaciones van adaptando, sino al proceso como tal que cada organización ha definido para sí misma (Wieczorek et al., 2014).

Más adelante, en este trabajo se describirán los modelos de ciclo de vida del proceso de desarrollo, se mostrarán las herramientas software de apoyo a la infraestructura del proceso y la propuesta preliminar para la creación de una nueva herramienta de este tipo, pero mucho más completa.

## 2. Planteamiento del problema

Las organizaciones de acuerdo a sus necesidades definen su propio proceso de desarrollo de software. En base a ello, suele suceder que muchas de ellas al definir un proceso de desarrollo, pueden incurrir con el paso del tiempo en el uso de metodologías ya obsoletas y pueden no aprovechar los beneficios que le ofrecen metodologías más actuales. Así mismo en el uso de las herramientas software que intervienen en el proceso, puede presentar el mismo inconveniente en las organizaciones.

En la actualidad son diversos los modelos de ciclo de vida de desarrollo de software, metodologías y herramientas que pueden ser usados por las organizaciones para llevar a

cabo el desarrollo de sus productos, debido a la existencia de múltiples ciclos de vida, metodologías y herramientas. Para las organizaciones se crea la necesidad de lograr un proceso más efectivo, que lleve al cumplimiento de los requisitos del producto, pero también que cumpla con las expectativas del negocio.

Con el presente trabajo se pretende realizar un bosquejo del proceso del desarrollo de software, los modelos de procesos más utilizados y se van a explorar algunas herramientas de apoyo al modelado de procesos en cuanto a la definición de la infraestructura del proceso del desarrollo de software, con el fin de determinar qué bondades brindan las herramientas analizadas y qué desventajas pueden encontrarse al modelar un proceso. Por último, se hará una propuesta para la creación de una herramienta software que ofrezca mejores bondades para el apoyo a la infraestructura del desarrollo de software.

### **3. Marco conceptual**

El marco conceptual del trabajo realizado se basa en modelos de procesos de desarrollo de software, infraestructura del proceso de

software y herramientas de soporte al proceso.

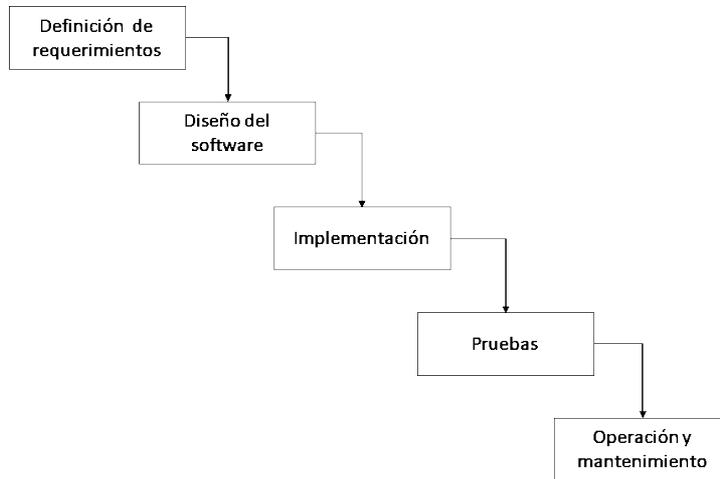
#### **3.1. Modelos de procesos de desarrollo de software**

En la actualidad existen diferentes modelos de procesos de acuerdo a las necesidades de la organización, pueden ser adaptados y ofrecen un “orden” para llevar a cabo el desarrollo de software, un orden en cuanto a que establecen una serie de pasos a seguir, pero que no se establecen como un estándar. Los principales modelos de procesos de software son los siguientes.

##### **3.1.1. Modelo en cascada**

Como lo establece Sommerville, “el modelo en cascada toma las actividades fundamentales de los procesos de especificación, desarrollo, validación y evolución y, luego, los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de software, implementación, pruebas y operación y mantenimiento.” (Sommerville, 2011, p. 29), como se observa en la Figura 1.

**Figura 1. Modelo en cascada**



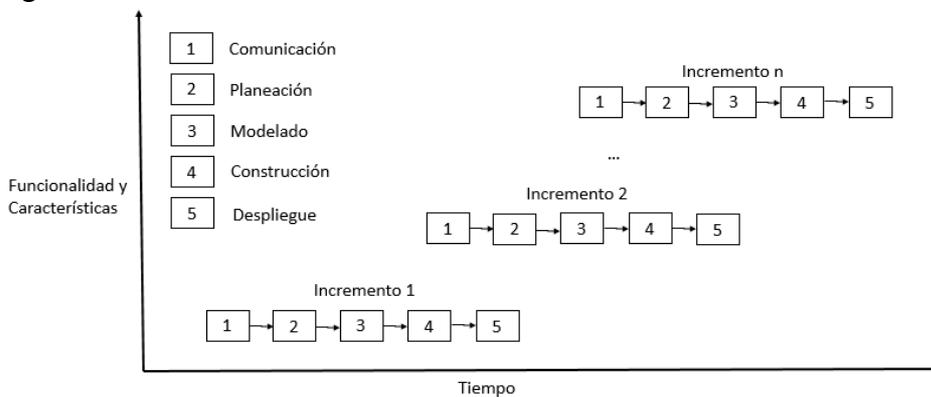
**Fuente.** Elaboración propia.

### 3.1.2. Desarrollo incremental

A diferencia del modelo en cascada, el desarrollo incremental se utiliza para procesos de desarrollo de software, en los cuales a pesar de tener los requisitos iniciales bien definidos, no se sugiere un proceso de desarrollo únicamente lineal, por el contrario, al

analizar de forma global se excluye una linealidad pura (Tilloo, 2013), ver Figura 2. En este modelo se encuentran inmersos el Modelo Incremental y el modelo DRA (Desarrollo Rápido de Aplicaciones).

**Figura 2. Desarrollo incremental**



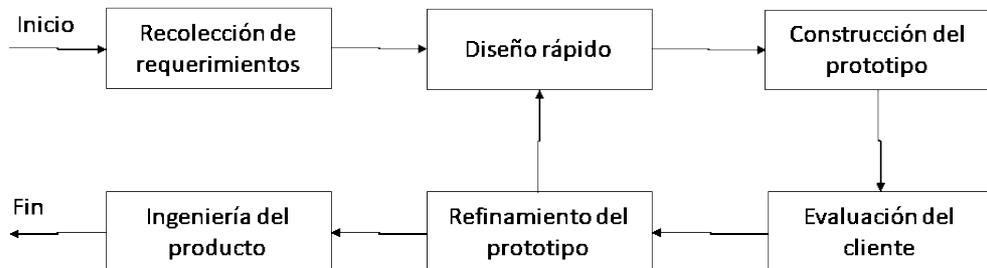
**Fuente.** Elaboración propia.

### 3.1.3. Desarrollo evolutivo

En este modelo se reconoce que los sistemas evolucionan con el paso del tiempo, por tanto, requieren de modelos que se adapten a dicha evolución ya que “los requisitos de los negocios y productos, a menudo cambian conforme se realiza el desarrollo; por lo tanto, la ruta lineal que conduce a un producto final no será real” (Pressman, 2005, p. 54). Por

lo tanto, se requiere un modelo que permita a los desarrolladores, a través del tiempo, lograr que el sistema evolucione y hacer entregas cada vez más completas de los sistemas. Entre los modelos de desarrollo evolutivo se encuentran el de Construcción de Prototipos y el Modelo en Espiral. En la Figura 3 se presenta el modelo de construcción de prototipos.

**Figura 3. Construcción de prototipos**



**Fuente.** Elaboración propia.

- Modelo de Construcción de Prototipos (Carr y Verner, 1997): Se inicia con la recolección de los principales requerimientos funcionales. Continúa con un rápido diseño y un desarrollo del prototipo. El prototipo es evaluado por los usuarios que pueden probarlo y proporcionar retroalimentación y experiencia en tiempo real. Los desarrolladores modifican el prototipo hasta que no haya cambios substanciales o actualizaciones, y el cliente y usuarios estén satisfechos.
- Modelo en Espiral (Boehm, 1987): En lugar de presentar las fases de desarrollo de manera secuencial, las presenta en forma de espiral. La espiral está dividida en varias regiones de tareas: Planificación, Análisis de riesgo, Ingeniería y Evaluación del cliente.

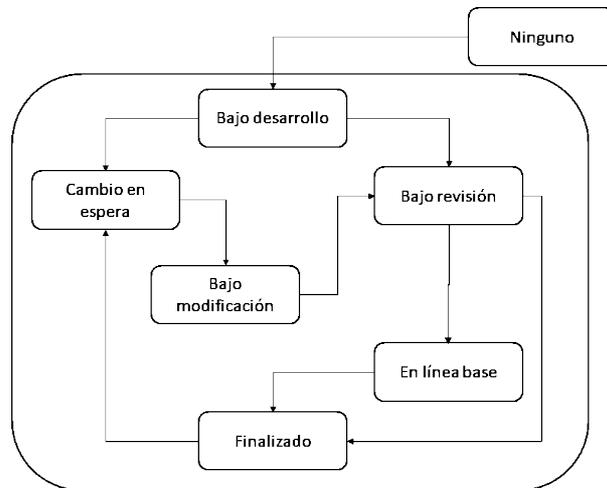
### 3.1.4. El Modelo de desarrollo concurrente

Este modelo puede ser aplicado a cualquier tipo de desarrollo de software, ya que consiste en la definición de un conjunto de estados, que son asignados a las diferentes

actividades o tareas del proceso de desarrollo, dichos estados determinan el cómo se van desarrollando las actividades del proceso, es decir, se puede saber cuándo una actividad ya ha finalizado, si requiere modificaciones o si está en proceso

(Davis y Sitaram, 1994), como se observa en la Figura 4. El cambio de los diferentes estados depende de la revisión de las diferentes fases del proceso de desarrollo y de las actividades.

**Figura 4. Modelo de desarrollo concurrente**



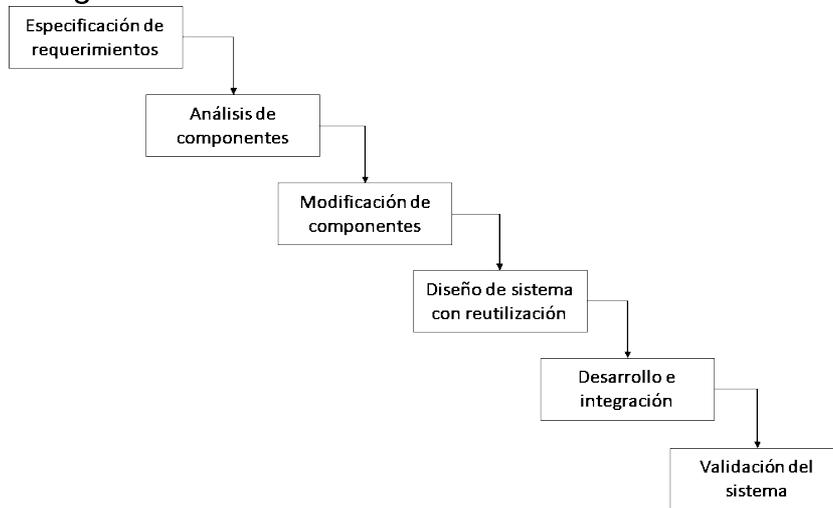
Fuente. Elaboración propia.

### 3.1.5. Desarrollo basado en componentes

La reutilización de componentes es una metodología que consiste en tomar componentes o partes de los mismos de proyectos anteriores, para usar sus diseños o códigos que son similares a los que requiere el desarrollo actual y haciéndoles modificaciones y ajustes, se adaptan al sistema (Crnkovic y Larsson, 2001).

Se lleva a cabo en cuatro etapas análisis de componentes, modificación de requerimientos, diseño del sistema de reutilización y desarrollo e integración, como se observa en la Figura 5. Estas etapas relacionadas con la reutilización de componentes y dos etapas adicionales, una al inicio de especificación de requisitos y otra al final, para validar el sistema.

**Figura 5. Ingeniería de software orientada a la reutilización**



**Fuente.** Elaboración propia.

### 3.2. Infraestructura del proceso de software

La infraestructura del proceso proporciona definiciones y políticas para interpretar y aplicar los procesos, y descripciones de los procedimientos a ser utilizados para implementar el proceso. Adicionalmente, proporciona herramientas, entrenamiento y los miembros del personal a quienes se le han sido asignados responsabilidades para establecer y mantener la infraestructura (SwEBoK, 2014).

### 3.3. Herramientas para el proceso de software

Este tipo de herramientas soportan notaciones para definir, implementar y gestionar procesos de software y modelos de ciclos de vida de software. Incluyen editores para notaciones como: diagramas de flujo de datos,

diagramas de estados, BPMN, redes de Petri y diagramas de actividad UML (SwEBoK, 2014). Algunas herramientas de proceso de software permiten diferentes tipos de análisis y simulaciones (por ejemplo, simulación de eventos discretos).

## 4. Metodología

Para la realización del presente trabajo se llevó a cabo la exploración de la información relacionada con el proceso de la Ingeniería de Software, más específicamente con los modelos del proceso y la infraestructura del mismo, con el fin de realizar un estado del arte de dicha información.

Teniendo en cuenta lo anterior, se realizó la búsqueda de herramientas software que apoyaran el modelado

de procesos en la ingeniería de software, con el objetivo de describir las herramientas en términos de ventajas y desventajas y, por último, hacer una propuesta de implementación de una herramienta que integre las diferentes ventajas identificadas, para ofrecer un apoyo más completo al modelado de procesos de software.

La búsqueda de herramientas se realizó utilizando motores de búsqueda con las siguientes palabras claves: *“software process modeling and (“software” or “tool”)*. Los criterios de selección de las herramientas de modelado de proceso se basaron en los resultados de la consulta de las páginas web de cada herramienta, la lectura de sus características y funcionalidades, descarga e instalación de las herramientas y análisis de su ejecución.

## 5. Herramientas de modelado de procesos analizadas

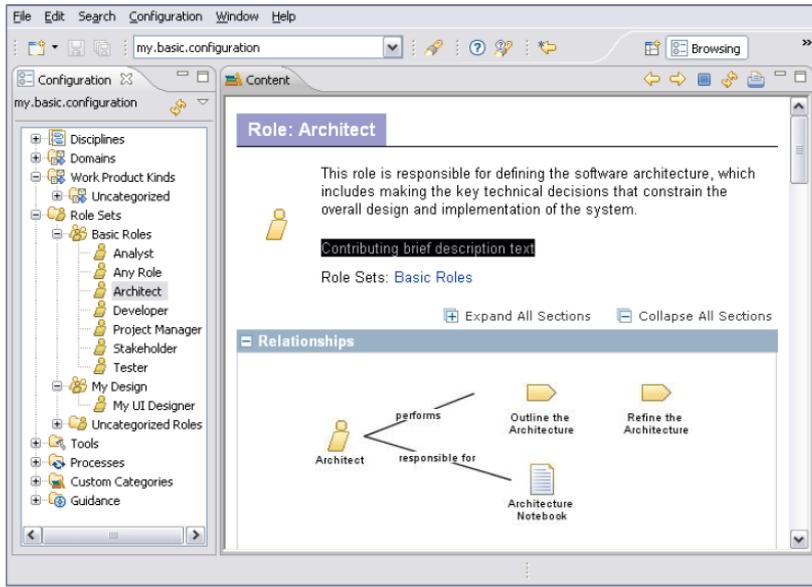
Las herramientas seleccionadas fueron Eclipse Process Framework Composer (EPF Composer), Genexus Business Process Modeler, Cameo Business Modeler, Bizagi y Enterprise Architect.

### 5.1. Eclipse Process Framework Composer- EPF Composer

EPF Composer (EPF, 2017) es una herramienta de modelado de procesos que permite:

- Modelar procesos en los cuales se pueden definir los roles, las tareas, los productos de trabajo y las guías de desarrollo.
- Crear paquetes de trabajo que en un proceso de desarrollo son considerados hitos, por cada paquete de trabajo se define el conjunto de roles, tareas, productos de trabajo y las guías.
- Generar un documento en HTML en el cual se muestra la documentación del proceso modelado por medio de una página web, en donde también se visualizan los diagramas de actividad.
- Establecer las relaciones de los roles, con las tareas, los productos de trabajo y las guías.
- Crear un diagrama de actividad por cada proceso modelado, en donde por defecto, se cargan las tareas involucradas en éste (ver Figura 6).
- Definir cuál o cuáles son las actividades predecesoras de cada actividad.

Figura 6. Modelo de procesos desarrollado con EPF Composer



Fuente. Elaboración propia.

Algunas desventajas identificadas son:

- No permite calendarizar las actividades.
- No tiene opción de poder definir un tiempo de ejecución de las actividades.
- No brinda opción de notificaciones.

## 5.2. Genexus Business Process Modeler

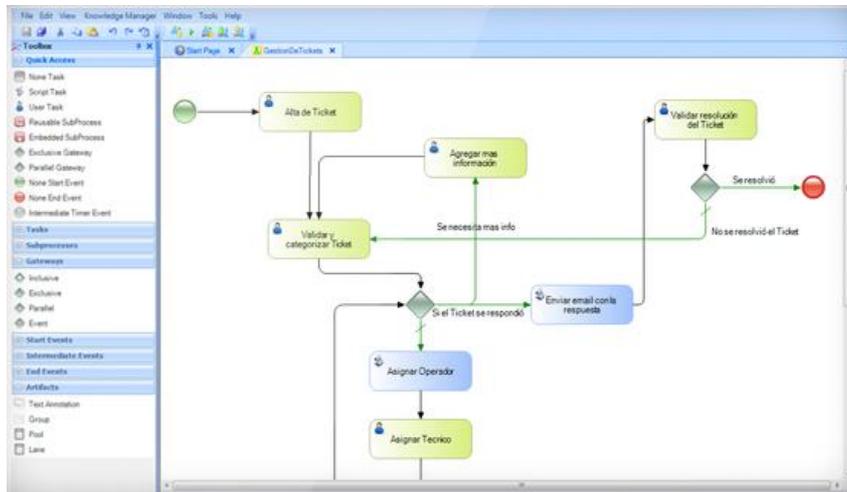
Genexus Business Process Modeler (Genexus, 2017) es una herramienta de modelado de procesos de negocio que permite:

- Modelar los procesos de forma gráfica, es decir, a través de los

diagramas de actividad (ver Figura 7).

- Crear las tareas, crear los usuarios (roles/responsables), artefactos de trabajo o productos de trabajo.
- Crear y cargar documentos de soporte al proceso.
- Crear la documentación de las guías o documentos de apoyo al proceso.
- Permitir asociar a los procesos las tareas correspondientes.
- Crear calendarios de ejecución que se asocian a las tareas.
- Programar notificaciones para el proceso.

**Figura 7. Modelo de procesos con Genexus Business Process Modeler**



**Fuente.** Elaboración propia.

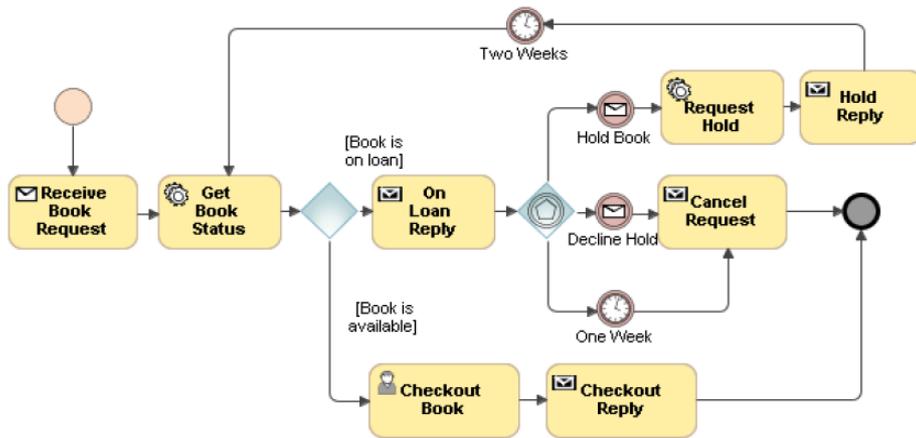
Algunas desventajas identificadas son:

- No permite hacer la documentación del mismo.
- No permite calendarizar las actividades.
- No permite definir tiempo de ejecución de las actividades.
- No provee la opción de dar notificaciones de ningún tipo.
- No permite establecer los predecesores de las actividades del proyecto.
- Crear los paquetes necesarios para el modelado del proceso (ver Figura 8) y la documentación antes de realizar la diagramación.
- Crear los diagramas de actividad o de proceso.
- Trabajar de forma paralela en la documentación y la creación del diagrama de actividad, es decir que se puede documentar a medida que se realiza la gráfica.
- Insertar roles, subprocessos o actividades, tareas y guías.
- Colocar las relaciones entre tareas, roles y actividades.
- Asignar tiempo de ejecución a las actividades.
- Colocar hipervínculos a las guías de apoyo al proceso.

### 5.3. Cameo Business Modeler

Cameo Business Modeler (Cameo, 2017) es un software de modelado de procesos que permite:

**Figura 8. Modelo de procesos desarrollado con Cameo Business Modeler**



**Fuente.** Elaboración propia.

Algunas desventajas identificadas son:

- No permite programar un calendario de las actividades.
- No permite establecer los predecesores de las actividades del proyecto.
- No permite colocar notificaciones.

#### 5.4. Bizagi Modeler

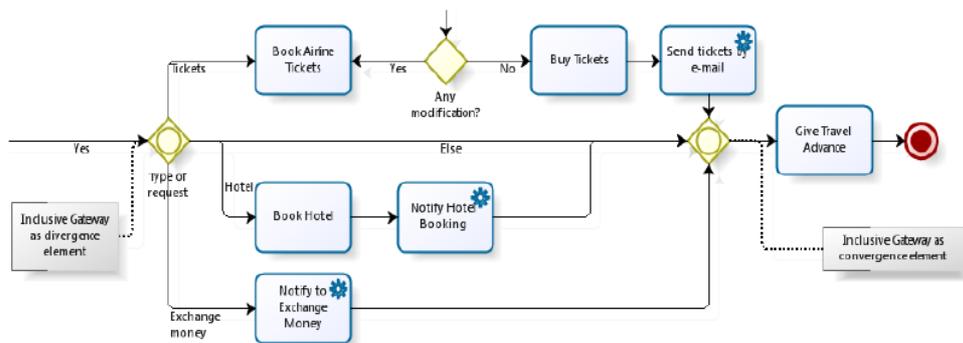
Bizagi Modeler (Bizagi, 2017) es una herramienta que permite:

- Realizar el modelado a través del diagrama de procesos, como se observa en la Figura 9.
- Insertar o crear los recursos como roles o entidades con su descripción.
- Insertar las tareas en el diagrama y posteriormente en sus propiedades se hace la edición de nombre,

descripción y se asocian los recursos.

- Relacionar tareas con roles.
- Insertar instancias de subprocessos, que en este caso son cada una de las actividades a desarrollar.
- Adicionar objetos de datos, es decir, documentos, datos, guías y otros objetos usados durante el proceso de desarrollo, se pueden cargar los archivos correspondientes.
- Simular la ejecución del proceso, una vez finalizada la ejecución brinda una opción de mostrar los resultados, los cuales pueden ser descargados en una plantilla de Excel.
- Adicional se puede publicar el proceso en un documento en formato Word, PDF, web o como una wiki, esta última se debe cargar en un servidor.

**Figura 9. Modelo de procesos desarrollado con Bizagi Modeler**



**Fuente.** Elaboración propia.

Algunas desventajas identificadas son:

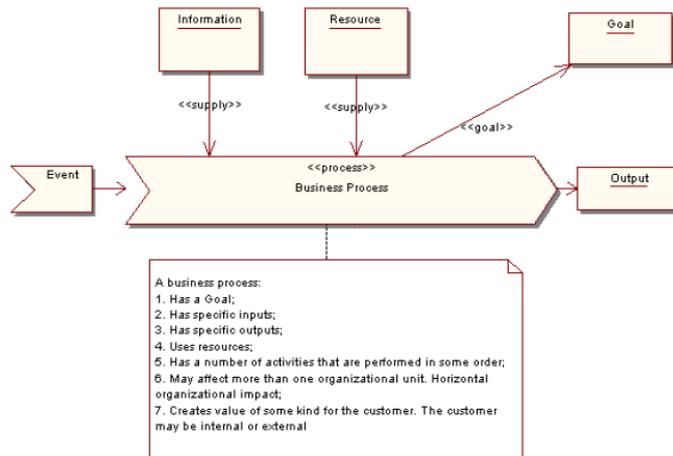
- No permite crear la documentación del proceso antes de realizar el diagrama de proceso o de actividad.
- No permite crear una relación directa de actividades con las tareas, los roles y guías, sino que estas relaciones se crean por defecto al insertarse gráficamente la tarea dentro de la actividad o subproceso, por lo cual en este caso podría incurrirse en errores de asignación de las relaciones.
- Se debe crear una actividad o subproceso por cada rol que esté relacionado con la misma.
- No permite establecer los predecesores de las actividades del proyecto de desarrollo de software.

### 5.5. Enterprise Architect

Enterprise Architect (Enterprise, 2017) es una herramienta de modelado de procesos que permite:

- Crear paquetes para hacer la definición de las tareas, los paquetes de trabajo, los roles, y las actividades.
- Modelar los procesos a través de los diagramas de actividad, como se observa en la Figura 10.
- Establecer las relaciones de los roles, con las tareas, los productos de trabajo y las guías.
- Crear los diagramas de actividad y de procesos.
- La documentación se desarrolla una vez el proceso esté diagramado

**Figura 10. Modelo de procesos desarrollado con Enterprise Architect.**



**Fuente.** Elaboración propia.

Algunas desventajas identificadas son:

- No permite la creación de la documentación del proyecto antes de diagramar.
- No permite adicionar notificaciones.
- No permite crear un calendario de las actividades.
- No genera ningún tipo de archivo con la documentación.

## 6. Comparativa de herramientas

En la Tabla 1 se presenta una comparativa de las herramientas de modelado de procesos estudiadas. Las notaciones de las herramientas están

basadas fundamentalmente en SPEM, BPMN y diagramas de actividad UML. La integración con otras herramientas es muy amplia pero cada herramienta ofrece una integración específica no compartida por otras herramientas. Todas las herramientas modelan procesos genéricos de desarrollo de software, únicamente Enterprise Architect modela artefactos de desarrollo. La mayoría de herramientas genera documentación del proceso en HTML. La única herramienta que ofrece funcionalidades para la ejecución del proceso es Bizagi. Por último, el tipo de licencia es bastante heterogéneo con licencias tanto propietarias como libres.

**Tabla 1. Comparativa de herramientas de modelado de procesos.**

Herramienta	Notación	Integración	Procesos	Documentación	Ejecución	Licencia
EPF Composer	SPEM 2.0	Eclipse	Procesos genéricos	HTML	No	Open source y Propietaria
Genexus Business Process Modeler	BPMN	Con GeneXus BPM Suite	Procesos genéricos	HTML	No	Open source
Cameo Business Modeler	BPMN y diagramas de actividad UML	SOA, Enterprise Architect	Procesos genéricos	HTML, Word	No	Propietaria
Bizagi	BPMN	Servicios Web	Procesos genéricos	Word, PDF, HTML, wiki	Si	Open source, excepto la ejecución
Enterprise Architect	UML, BPMN y SysML	NIEM (National Information Exchange Model)	Procesos técnicos	HTML	No	Propietaria

Fuente. Elaboración propia.

## 7. Descripción de la herramienta propuesta

Los elementos de un proceso que hacen parte de la herramienta que se propone son los siguientes, y están basados en la notación de Lai (1993):

1. Actividad: Es aquello que va a pasar en un proyecto, que puede estar relacionado con diferentes aspectos del proyecto y que es asociado a un equipo de trabajo.
2. Secuencia: Es el orden en el cual se hace la ejecución de las actividades.
3. Modelo de proceso: Hace parte del proceso, un modelo instanciado que permite analizar el proceso que

se está ejecutando y predecir así su comportamiento.

4. Recurso: Los recursos necesarios para llevar a cabo el proceso. El modelo de proceso determina la demanda total del recurso.
5. Control: Es un mecanismo para lograr que la ejecución del proyecto y que sus diversos componentes se lleven a cabo como se espera.
6. Política: Se considera una norma establecida a seguir para el logro del objetivo del proyecto.
7. Organización: Es la estructura de conformación del equipo de trabajo.

En la herramienta propuesta, el modelado de un nuevo proyecto incluirá la creación por defecto de

estos 7 elementos base para el modelado de un proceso, Los 7 elementos representan cada uno, un paquete, en donde se hará la inserción de los elementos pertinentes.

De acuerdo a la información recopilada, se identifican las siguientes propiedades que una herramienta software para modelado de procesos debe tener (Pfleeger, 2002, p. 78).

Facilitar la comprensión y la comunicación humanas: Esta propiedad se refiere a la forma en la cual el proceso debe ser modelado para que sea entendido por las diferentes personas que interactúan en el mismo. La herramienta que se va a desarrollar debe ofrecer una guía de entrenamiento y tutoriales que ayuden a la comprensión del proceso de modelado que se va a llevar a cabo.

Dar soporte para la mejora del proceso: Una de las búsquedas de la Ingeniería de Software es la mejora del proceso de desarrollo de forma continua, la herramienta debe permitir la reutilización de componentes o parte de los mismos para el desarrollo de los procesos. También la herramienta tendrá disponible una biblioteca de componentes software que pueden ayudar en los diferentes aspectos y fases del proceso de

desarrollo de software, biblioteca que el equipo de trabajo irá nutriendo de acuerdo a su experiencia con las mismas y para que la herramienta que se propone haga la recomendación de éstas en la ejecución de las diferentes etapas del proceso.

Dar soporte para la gestión del proceso: La herramienta a desarrollar debe permitir un monitoreo y supervisión continuos de la ejecución del proceso, es decir, debe identificar los puntos clave del proceso con el fin de que el equipo de trabajo se centre en éstos e identifique posibles mejoras en su gestión.

Brindar una guía automatizada en la realización del proceso: Con base a las técnicas en que estará basada la herramienta, permitirá definir la infraestructura necesaria para representar el ambiente de desarrollo o parte de éste, de acuerdo al proceso modelado, lo cual servirá para guiar el proceso, brindar sugerencias y guardar las partes reutilizables del proceso.

Dar soporte para la ejecución automatizada de procesos: Es la propiedad que permite que el proceso o gran parte del mismo pueda ser automatizado, que la información sea respaldada y tener reglas definidas para la conservación de la integridad

del proceso. Esta propiedad debe ser integrada a la herramienta, la cual permitirá ejecutar el proceso y generar un mapa documental de la información, con su estructura definida de acuerdo al modelado realizado y las representaciones gráficas pertinentes. Este procedimiento dará la opción de generar un documento en diferentes extensiones con el proceso documentado.

## 8. Conclusiones

El modelado de procesos de software es un área de investigación importante que permite a las empresas desarrolladoras de software definir e implementar sus procesos propios de software. Se requieren de herramientas que faciliten el modelado de procesos. En el mercado se encuentran numerosas herramientas de este tipo, cada una con sus propias ventajas y desventajas.

Este trabajo presenta un análisis de procesos de software y de herramientas que permiten el modelado de dichos procesos. Además, una colección de características que debe poseer una herramienta de modelado es presentada, donde es muy importante que ofrezca, además de las

capacidades de modelado, poder gestionar el proceso, guiar y automatizar su ejecución, reutilizar procesos existentes y por último mejorarlo con base en su monitoreo.

La descripción de la herramienta propuesta es una primera aproximación para iniciar el desarrollo de la misma, no pretende ser un análisis de una futura herramienta si no un punto de partida para la realización de un estudio de viabilidad de la misma.

El trabajo futuro consistirá en el desarrollo de los modelos de análisis de requisitos y el modelo de diseño de dicha herramienta.

## Referencias

- Boehm, B. (1987). A spiral model of software development and enhancement, *Software Engineering Project Management*, pp. 128-142.
- Bizagi Modeler.  
<https://www.bizagi.com/es>
- Cameo Business Modeler.  
<https://www.nomagic.com/products/cameo-business-modeler>
- Carr, M. y Verner, J. (1997). *Prototyping and software development approaches*.

- Department of Information Systems, Hong Kong: City University of Hong Kong.
- Crnkovic, I. y Larsson, M. (2001). Component-Based Software Engineering-New Paradigm of Software Development. Malardalen University, Sweden.
- Davis, A., y P. Sitaram, «A Concurrent Process Model for Software Development», Software Engineering Notes, ACM Press, vol. 19, n.º 2, abril 1994, pp. 38- 51.
- EPF Composer.  
[https://eclipse.org/epf/downloads/tool/tool\\_downloads.php](https://eclipse.org/epf/downloads/tool/tool_downloads.php).
- Enterprise Architect.  
<http://www.sparxsystems.com.au/products/ea/>
- Fuggetta, A. (2000). Software Process: A Roadmap, 22nd International Conference on Software Engineering (ICSE' 2000), Future of Software. Engineering Track, Limerick (Irlanda), ACM.
- Fuggetta, A. y Di Nitto, E. Software Process. Proceedings of the on Future of Software Engineering FOSE 2014, Hyderabad, India – May 31 - June 07, ACM New York, NY, 2014, pp. 1-12.
- Genexus.  
<https://www.genexus.com/producos/bpm-business-process-modeler?es>
- Lai, R. Ch. T. (1993). Process Definition and Process Modeling Methods. Virginia Center of Excellence for Software Reuse and Technology Transfer.
- Pfleeger, S. L. (2002). Ingeniería de software Teoría y Práctica (Primera Edición ed.). Buenos Aires, Republica de Argentina: Pearson Education S.A.
- Pressman, R. S. (2005). Ingeniería de software un Enfoque Práctica (Sexta Edición ed.). México, México: McGraw-Hill/Interamericana Editores, S.A. de C.V.
- Sommerville, I. (2011). Ingeniería de Software (Novena Edición ed.). México: Pearsón Educación.
- Sutton, S. M. (2000). The Role of Process in a software startup, IEEE Software, July/August 2000.
- SwEBok. (2014). Guide to the Software Engineering, Body of Knowledge, Version 3.0. SWEBOK. Editors Pierre Bourque, École de technologie supérieure (ÉTS) Richard E. (Dick) Fairley, Software and Systems Engineering Associates (S2EA). IEEE Computer Society.

Tilloo, R. (2013). What is incremental model in software engineering? Retrieved from <http://www.technotrice.com/incremental-model-in-software-engineering>

Wieczorek, M., Vos, D. y Bons, H. (2014). Systems and Software Quality: The next step for industrialization. Springer Publishing Company.