



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN  
TRABAJO FINAL DE GRADUACIÓN

**MODELO DEL PROCESO DE DESARROLLO DE SOFTWARE  
CON ENFOQUE EN EL PROCESO DE NEGOCIO**

Autores:

**Fernanda Magdalena Leguizamón**

**Valeria Cecilia Acosta**

Profesor Guía:

**Ing. Mabel Sosa**

**-2013-**

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

**TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE  
INFORMACIÓN**

# **MODELO DEL PROCESO DE DESARROLLO DE SOFTWARE CON ENFOQUE EN EL PROCESO DE NEGOCIO**

**Autores:**

.....  
Fernanda Magdalena Leguizamón

.....  
Valeria Cecilia Acosta

**Profesor Guía:**

.....  
Ing. Mabel del Valle Sosa

Aprobado el día..... del mes de ..... del año 2013

por el Tribunal integrado por:

.....  
.....

Santiago del Estero – Argentina

## DEDICATORIAS

*A Dios*

*por acompañarme en este camino.*

*A mis padres y hermanas*

*por el apoyo incondicional de siempre.*

*A mis amigos y compañeros*

*por apoyarme y ayudarme en cada etapa,*

*dar-me su contención en los momentos difíciles*

*y su aliento para terminar de realizar esta etapa.*

### Cecilia Acosta

*A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar.*

*A mis padres por estar en los momentos importantes de mi vida, su paciencia, sus consejos, por velar por mi bienestar y educación siendo mi apoyo en todo momento.*

*A mis hermanos Juan, Matías y Lourdes, que con su amor me enseñaron a salir adelante.*

*A mis sobrinos Leonardo y Benjamín por darle sentido a mi vida.*

*A mis amigos por estar a mi lado en las buenas y malas.*

### Fernanda Leguizamón

# AGRADECIMIENTOS

En este último escalón de una de nuestra meta compartida queremos agradecer por todo el apoyo brindado y la confianza depositada en nosotras.

Un agradecimiento especial y nuestra profesora guía Ing. Mabel Sosa por la dedicación, paciencia, consejos y sobre todo por el tiempo brindado. Gracias Profe!.

A nuestro compañero y amigo Alejandro Oscar Autalan por su ayuda y su colaboración con la programación en phyton.

Gracias a toda nuestra familia, compañeros y amigos por el apoyo incondicional y por estar siempre alentando.

Gracias Totales.

**Fernanda Leguizamón – Cecilia Acosta**

---

RESUMEN.....	1
INTRODUCCIÓN.....	2
<b>CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA, OBJETIVOS Y ALCANCE DEL TRABAJO</b>	
I.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.....	4
I.2. ANTECEDENTES.....	5
I.3. JUSTIFICACIÓN DE LA PROPUESTA.....	6
I.4.OBJETIVOS.....	7
I.5. ALCANCE DEL TRABAJO .....	8
<b>CAPÍTULO II. MARCOS REFERENCIALES</b>	
II.1. MARCO TEÓRICO.....	9
II.1.1. Proceso de Desarrollo de Software.....	9
II.1.2. Proceso de Negocio.....	12
II.1.3. Reglas de Proceso de Negocio.....	12
II.1.4. Modelo.....	14
II.1.5. Metamodelo.....	14
II.1.6. Modelo de Procesos de Negocio.....	14
II.2. MARCO METODOLÓGICO.....	16
II.2.1. Notación de Modelado de Proceso de Negocio (BPMN).....	16
II.2.2. AToM3.....	16
<b>CAPÍTULO III. DESARROLLO DEL TRABAJO</b>	
III.1. INTRODUCCIÓN.....	18
III.2. PROCEDIMIENTO DE CONSTRUCCIÓN DEL METAMODELO.....	19
III.2.1. Análisis de la Notación BPMN.....	19
III.2.1.1 <i>Características Semánticas de los Elementos de BPMN</i> .....	20
III.2.1.2 <i>Aspectos Sintácticos de la Notación BPMN</i> .....	31

III.2.1.3 <i>Relaciones entre los Elementos Notacionales</i> .....	35
III.2.1.4 <i>Asociación entre Elementos Notacionales y Símbolos Gráficos que lo Representan</i> .....	39
III.2.2 <b>Definición del Metamodelo Conceptual</b> .....	40
III.2.3. <b>Implementación del Metamodelo BPMN</b> .....	42
III.2.4. <b>Verificación del Metamodelo en AToM<sup>3</sup></b> .....	43
<b>III.3. MODELADO DEL PROCESO SOFTWARE CON BPMN</b> .....	44
III.3.1. <b>Comprobación del Proceso del Software Obtenido</b> .....	48
III.3.1.1. <i>Definición de Reglas Gramaticales</i> .....	48
III.3.1.2. <i>Aplicación de las Reglas Gramaticales</i> .....	54
<b>CONCLUSIONES</b> .....	60
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	61
ANEXOS A.....	65
ANEXOS B.....	68
ANEXOS C.....	79
ANEXOS D.....	101
ANEXOS E.....	107

El modelado de los procesos de negocio ha adquirido actualmente gran relevancia en la Ingeniería del Software, por lo que se requiere de métodos para la construcción de modelos donde se identifiquen con precisión tanto los elementos de software como los de negocio. El objetivo que se propone en este trabajo es la modelación del proceso de software considerando el enfoque proceso de negocio. Para la modelación se utilizará la notación BPMN (Business Process Modeling Notation; Notación para el Modelado de Procesos de Negocio) orientada al negocio. El cumplimiento del objetivo definido permitirá determinar la viabilidad de modelar el proceso de software guiándose por aspectos de negocio. Además en este trabajo se creará el Metamodelo de BPMN y se lo implementará en el entorno de la herramienta AToM<sup>3</sup>, los que servirán de soporte a la implementación y comprobación del modelo de proceso software con enfoque en el proceso de negocio obtenido.

### **Palabras claves**

Proceso de Software, Procesos de Negocio, Metamodelo, BPMN.

El proceso software es el conjunto de actividades necesarias para producir un sistema software, ejecutadas por un conjunto de recursos humanos organizados según una estructura organizativa concreta y contando con un soporte de herramientas técnico-conceptuales [2]. El proceso software representa el flujo de subprocesos y actividades realizados en pos de un objetivo.

Un proceso de negocio es un conjunto de actividades que tomando una o más entradas crean una salida que tiene valor para un cliente. Los procesos de negocio representan el flujo de trabajo y de información a través del negocio, definidos en un contexto lógico y temporal. Los objetivos centrales de un modelo de proceso de negocio son [25]: a) mejorar el entendimiento de una situación y comunicarla entre los diversos stakeholders y, b) utilizarlos como una herramienta para alcanzar las metas de los proyectos organizacionales.

Los procesos software y los procesos de negocio presentan ciertas similitudes, siendo la más común de ellas el que ambos tratan de capturar las principales características de un grupo de actividades parcialmente ordenadas que son llevadas a cabo para lograr una meta específica. Ahora bien, mientras que la meta de un proceso software es obtener un producto software [2], el de un proceso de negocio es obtener como resultado un producto o servicio beneficioso para los clientes u otros usuarios afectados por el proceso [5].

Un proceso puede ser correctamente comprendido si es representado por modelos adecuados. Para una adecuada descripción del proceso es necesario integrar diferentes aspectos en un mismo modelo: *qué hace, quién lo hace, cuándo y dónde, cómo y por qué se hace*. Es decir, el modelo debe ser capaz de explicitar los elementos de un proceso: tareas, eventos, mensajes, roles, etc. En consecuencia, la tarea de modelado del proceso debe permitir la elaboración de modelos comprensibles por los usuarios, y también la detección y corrección de errores antes de la etapa de construcción e implantación. En otras palabras, esta tarea debe posibilitar una evaluación temprana de las propiedades de calidad y la mantenibilidad de los modelos conceptuales.

Para cubrir los diferentes aspectos de negocio se han formulado diversas arquitecturas, frameworks y metamodelos, cada uno orientado hacia un aspecto en particular y en algunos casos integrando los distintos aspectos [5][7][38]. También hay diferentes lenguajes para el modelado de procesos de negocio, cada uno con sus reglas propias y específicas, tales como BPMN [8], EPC (Procesos impulsados por Eventos en Cadena), XPD (Lenguaje de Definición de Procesos XML), Diagrama de Actividad de UML (Lenguaje Unificado de Modelado). Entre estos lenguajes, la notación BPMN es la que mejor cubre los aspectos de procesos de negocio tales como calidad, soporte de la complejidad, gestión e integración. BPMN es un lenguaje o notación gráfica, adoptado como estándar por OMG [8], que permite diseñar y gestionar procesos de negocio con *diagramas de procesos de negocio* (DPN) y actualmente es de gran interés para el ámbito empresarial, usada tanto por analistas de negocio como por analistas de sistemas.

Este trabajo se centra en la importancia de abordar el proceso de desarrollo de software desde una perspectiva de negocio, y se considera la relevancia de realizar un modelado que abarque tanto aspectos de software como de negocio. El modelado del proceso de software requiere de métodos formales que permitan capturar, representar y evaluar eventos, entradas, salidas y recursos asociados al proceso de negocio. Es decir, métodos que

faciliten la construcción de modelos trazables donde se conecte en forma directa los elementos de diseño con los elementos de negocio a través de conectores de implementación, para definir más claramente el alcance del sistema y la forma de su implementación. Modelar el proceso de negocio es una parte esencial de cualquier proceso de desarrollo de software.

Considerando lo expuesto, se plantea:

- a) Modelar el proceso de software desde una perspectiva de negocios; y
- b) Construir el metamodelo BPMN e implementarlo en la herramienta AToM<sup>3</sup> [4].

El primer objetivo se orienta a determinar la viabilidad de modelar el proceso de software guiándose de aspectos de negocios, estos modelos pueden ser manipulados y simulados en su comportamiento, para lo cual se define la semántica operacional de los modelos del formalismo BPMN mediante gramática de grafos. Esta simulación facilitara la comprobación de los modelos de los procesos diseñados. El segundo, tiende a proporcionar una herramienta actualizada con el metamodelo BPMN, lo que permitirá la implementación y validación del proceso de software con enfoque en el negocio, es decir, permitirá la representación de los modelos de procesos que cubran las cuatro perspectivas: *funcional* (indica cuáles elementos del proceso se están realizando) y de *comportamiento* (representa cuándo los elementos del proceso se realizan, por ejemplo, secuencialidad, y bajo qué condiciones, por ejemplo loops, iteración, etc.); *informacional* (representa las entidades informacionales producidas o manipuladas por un proceso, por ejemplo datos, artefactos y objetos) y *organizacional* (indica dónde y por quién son realizados los elementos del proceso en una organización, por ejemplo una unidad organizacional, una persona, rol o un recurso automatizado).

Este trabajo se estructura del siguiente modo. En el *capítulo 1*, se presenta el planteamiento del problema, objetivos y alcance del trabajo; en el *capítulo 2* se exponen los marcos referenciales relacionados con los aspectos teóricos y metodológicos; en el *capítulo 3* se presenta el desarrollo de la propuesta, se describe el metamodelo BPMN explicando cada una de las tareas realizadas hasta su construcción, se definen las reglas de negocio y se presentan los resultados de la comprobación del modelo de Proceso de Software orientado al negocio alcanzado. Finalmente se presentan las conclusiones del trabajo relacionadas con el modelado del Proceso Software con enfoque en el Proceso de Negocio.

**PLANTEAMIENTO DEL PROBLEMA, OBJETIVOS  
Y ALCANCE DEL TRABAJO**

---

**I.1. PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA**

El modelado de proceso software tiene como objetivo principal guiar, controlar y gestionar las actividades del proceso; y el objetivo del modelado de proceso de negocio es producir una descripción de la realidad que permita entenderla y eventualmente modificarla con el propósito de incorporar mejoras organizacionales; sin embargo presentan ciertas similitudes, ambas tratan de capturar las principales características de un grupo de actividades parcialmente ordenadas que son llevadas a cabo para lograr una meta específica. Ahora bien, mientras que la meta de un proceso software es obtener un producto software [1], el de un proceso de negocio es obtener como resultado un producto o servicio beneficioso para los clientes u otros usuarios afectados por el proceso [8].

Las circunstancias de mercado de la última década han conducido, tanto a los desarrolladores de software como a los analistas de negocios y a las organizaciones en general, a tener que centrarse en sus procesos como uno de los aspectos más relevantes para prosperar y sobrevivir [33]. Esta situación ha incrementado la necesidad de analizar, evaluar, medir y mejorar los procesos, tanto de software como de negocio [37].

A partir de esto se infiere la necesidad de producir modelos precisos y legibles para los usuarios, que faciliten la evaluación de la complejidad estructural y dinámica de los modelos conceptuales obtenidos.

Tanto el modelado del proceso de negocio como del proceso software requiere de herramientas y métodos formales de especificación, que permitan dar soporte al análisis, diseño y comprobación del proceso y, en particular, determinar con precisión la corrección de sus desarrollos y mantenibilidad de los modelos [22][24]. Para realizar estas especificaciones, existen diferentes herramientas [15][27][46], sin embargo, se necesita de un formalismo capaz de modelar y representar de forma completa, explícita y gráfica los aspectos de la dinámica del proceso, como así también el análisis y comprobación automática de las propiedades del sistema y los aspectos del negocio [41].

Actualmente, el modelado de negocios es uno de los procesos técnicos más importantes. Su objetivo es entender bien el problema, antes de abordar su solución; es decir, modelar primero el sistema de negocios donde habrá de operar la aplicación, antes de identificar, analizar y especificar los requisitos que esta aplicación deberá satisfacer. El modelado de negocios es esencial para la comprensión y evolución de una empresa y de sus sistemas de información, sin embargo falta la integración del campo del software con los del negocio y existe un gap en la relación entre analistas de negocio y los desarrolladores de aplicaciones de software. [13].

## I.2. ANTECEDENTES

En este trabajo mencionamos los siguientes antecedentes relevantes para el desarrollo del mismo:

El trabajo de Andrea Delgado [13] plantea la centralización de las aplicaciones informáticas en los procesos del negocio que realiza la organización, permitiendo reaccionar ágilmente a los cambios de los negocios, a la vez a los cambios en las tecnologías, reutilizando elementos de software tanto de diseño como de ejecución. Los estándares de BPMN (Business Process Modeling Notation), BPML (Business Process Modeling Language) y las herramientas BPMS (Business Process Modeling Systems) pueden proveer las bases para cerrar la brecha entre el modelado de los procesos del negocio y su implementación. Con este trabajo se intenta integrar dos visiones para que la organización pueda realizar su negocio: la visión del negocio centrada en especificar y mejorar sus procesos mediante análisis del negocio; y la visión de TI (Information technology) centrada en informatizar dichos procesos evolucionando en la tecnología y metodologías de desarrollo de software. En general, esta conjunción ha sido compleja y problemática sin alcanzar una visión común del negocio por ambas partes. Sin embargo las organizaciones son cada vez más dependientes de sus sistemas informáticos, cuentan con diversidad de sistemas que tienen entre sí dependencias complejas donde estos sistemas han ido creciendo en forma separada y heterogénea. Los avances en tecnología y los cambios en los requerimientos del negocio se retroalimentan y deben ser gestionados. Para integrar estas visiones, se hace necesario cambiar la forma en que se relacionan el Negocio y su informatización, permitiendo que los procesos sean definidos y gestionados por quienes tienen ese conocimiento, y la informatización sea realizada a partir de dichas definiciones y pueda ser cambiada según los cambios de la tecnología sin afectar esta definición. De esta forma se minimiza el impacto de los cambios en los procesos en la implementación de los mismos. El enfoque de diseño SOA (Service Oriented Architecture) propone integrar el enfoque de BPM (Business Process Modeling) con el desarrollo orientado a servicios, y el enfoque de desarrollo MDA (Model Driven Architecture) propone aportes a la automatización del desarrollo.

Otro de los antecedentes encontrados con respecto a nuestro problema, es el trabajo publicado por la OMG [49] quien propone una alternativa para lograr la automatización de procesos de desarrollo de software definidos con SPEM (*Software Process Engineering Metamodel*). La idea es emplear motores workflow que son utilizados para automatizar procesos de negocio. Para lograr esta automatización se define una transformación del metamodelo SPEM a un metamodelo de BPMN (Business Process Modeling Notation) por medio del lenguaje Relations que forma parte de QVT (Query/Views/Transformations). La especificación BPMN resultante podrá ser transformada a un lenguaje estándar para la implementación de procesos workflow. Con esto se logra fundamentalmente la automatización de cualquier proceso de desarrollo de software especificados bajo SPEM a través de su transformación a proceso workflow estándar.

Otro trabajo [6] propone las características que deberían poseer una las herramientas de soporte al proceso de desarrollo guiado por procesos de negocio. En este trabajo se plantea la construcción de una herramienta que posea dichas características pero que vaya más allá, comprendiendo características tales como, la integración de las tareas de la gestión del proceso de desarrollo con las tareas propias del proceso en sí. De esta manera una herramienta con dichas particularidades contribuiría a la agilización del proceso, disminuyendo los costes y mejorando la calidad del proceso y del producto.

En el trabajo realizado por José Luis De la Vara González, David Anes Alcolea y Juan Sánchez Díaz [10], presentan una aproximación que utiliza una especificación en la forma de modelo de metas, construida mediante heurísticas en base a modelos de procesos en la forma de BPMN. A partir del modelo de metas, mediante un proceso de refinamiento y de etiquetado de metas, se obtiene un modelo de requisitos en la forma de casos de uso. La especificación así obtenida permite reflejar de manera más cercana las necesidades del negocio y asegura el alineamiento de las mismas con el futuro sistema de información.

En [12] se propone una guía metodológica para ayudar a los analistas de sistema a cumplir esta necesidad mediante la integración de dos propuestas de ingeniería de requisitos. Primero, se capturan y especifican los requisitos funcionales de un sistema de información a partir de los procesos de negocio de una organización. Después, se especifican los flujos de información de los requisitos funcionales. Finalmente, se especifican los requisitos de datos a partir de los flujos de información siguiendo un conjunto de heurísticas.

Por último en [28] se presentan la aplicación del modelado de procesos de negocio en un proyecto de interoperabilidad entre un sistema de información de un hospital, teniendo en cuenta la integración del modelado de procesos con el diseño de sistemas de información, y la capacidad de los modelos para aportar información en el desarrollo de los componentes basados en un estándar de interoperabilidad para la comunicación entre sistemas en entornos de salud.

*A partir de la exploración bibliográfica se detectan algunos antecedentes como los mencionados anteriormente que incorporan al proceso de desarrollo de software ciertos aspectos de negocio, sin embargo se observa que persiste la necesidad de contar con un modelo formal y probado que represente de forma explícita y gráfica los aspectos de la dinámica del proceso de software alineado a los aspectos del negocio. En este trabajo se tomarán en cuenta los procesos de negocio del “desarrollo de software”, es decir, el proceso de software visto como “proceso de negocio” o desde el enfoque de proceso de negocio. Por otra parte no se han detectado herramientas de automatización gráficas que soporten al metamodelado BPMN necesario para la implementación y comprobación del modelo de proceso software orientado al negocio, obtenido en una primera etapa del trabajo.*

### **I.3.JUSTIFICACIÓN DE LA PROPUESTA**

El modelado del negocio es la técnica por excelencia para alinear los desarrollos con las metas y objetivos de las empresas e instituciones. Si se realiza de tal forma en que el modelo quede consensuado entre los grupos interesados, las posibilidades de éxito del proyecto aumentarán en forma muy importante. El modelado de negocios, y más específicamente el modelado de procesos de negocio, es la forma idónea para comunicarnos con los usuarios de todos los niveles.

El objetivo es comprender a la organización en su conjunto y establecer claramente una serie de fases que ordenen las actividades a llevar a cabo de manera permanente, para alcanzar el propósito de contar con un ciclo de mejora continua de procesos capaces de absorber los cambios que propone la realidad.

Las principales razones que justifican el trabajo son:

- La posibilidad de controlar la evolución del proceso de desarrollo de software orientado al negocio, desde modelos abstractos y la detección de errores en etapas tempranas del proceso tendiendo a una mejor calidad del mismo.
- La importancia de actualizar una herramienta Open Source (Código Abierto) como AToM<sup>3</sup> que facilite el modelado de los procesos de negocio soportado en BPMN.
- La importancia de contar con modelos de altos niveles de abstracción con la posibilidad de generar nuevos modelos en forma automática a partir de los modelos definidos y aplicando reglas de transformación entre dichos modelos para su ejecución o evaluación.
- Permitir especificar la funcionalidad del proceso de desarrollo de software separadamente de los detalles de la plataforma donde será implementado. Es decir, lograr centrarse en la lógica de la aplicación y no en los detalles propios de la plataforma e implementación.
- Posibilita la construcción de modelos que contemplan la captura de eventos, las entradas, los recursos y las salidas más importantes vinculadas con el proceso de negocio.
- Se puede iniciar la construcción de formalismos de modelos desde cero, lo cual implica la posibilidad de correcciones (adición o borrado de elementos, por ejemplo) en modelos que cambian constantemente de versión o en otros que presentan modificaciones sutiles frecuentemente.
- Facilidad gráfica de expresión del formalismo de los diferentes modelos, lo que reduce la complejidad en la comprensión de los mismos.
- Cada día están teniendo más importancia los procesos de negocio y por extensión las herramientas que nos sirven para modelar, simular, supervisar y gestionar ese tipo de procesos.
- El metamodelo de BPMN que permitirá definir formalismos que posibiliten la creación de un nuevo metamodelo para modelar los diferentes patrones de comportamiento de una determinada notación (Diagrama de clases, diagramas E-R, Redes de Petri, etc).

#### **I.4.OBJETIVOS**

Los objetivos generales que se definen para el logro del trabajo son:

- Complementar el modelado del proceso de software mediante la incorporación de aspectos de negocio.
- Contribuir a una mejor comunicación entre analistas de negocio y desarrolladores de software.

Los objetivos específicos definidos son:

- Proporcionar un metamodelo de la notación orientada al modelado de negocio con BPMN.
- Actualizar la herramienta de AToM<sup>3</sup> con la incorporación del formalismo BPMN.
- Proveer un modelo de proceso de desarrollo de software desde una perspectiva de Negocio.

Los objetivos metodológicos son:

- Definir un procedimiento para el desarrollo de metamodelos.
- Definir un procedimiento para comprobar la correctitud del metamodelo.

## **I.5. ALCANCE DEL TRABAJO**

Para cumplir con los objetivos definidos en el trabajo se realizarán las siguientes actividades:

1. Se actualizará la herramienta AToM<sup>3</sup>, con la incorporación del metamodelo de BPMN, siguiendo las reglas de metamodelado del entorno de AToM<sup>3</sup>.
  - El metamodelo a desarrollar considera los principales elementos de la notación BPMN y se tendrá en cuenta las variantes más importantes de cada uno de estos elementos notacionales.
  - El metamodelado BPMN permitirá modelar los diferentes patrones de comportamiento de los procesos o workflows. Por ejemplo ciclos, secuencias, bucles, etc.
2. Se modelará el proceso de desarrollo de software con un enfoque de negocio, para lo cual se emplea la notación BPMN (de OMG). De la actividad de modelado se obtiene un modelo semántico del proceso de desarrollo del software orientado al negocio.
3. Una vez obtenido el modelo de proceso de software en BPMN se lo implementa en AToM<sup>3</sup> actualizado para su posterior verificación.
4. Para cumplir con este paso, el modelo original del proceso de software será traducido a un modelo equivalente. Este modelo equivalente será descrito con un formalismo de gramática de grafos, para su posterior ejecución en AToM<sup>3</sup>, lo que permitirá analizar el comportamiento, comprobar la correctitud del mismo y el cumplimiento de las reglas de negocio especificadas previamente.

## II.1. MARCO TEÓRICO

A continuación, se presenta los conceptos y marco teórico en lo que se fundamenta el presente trabajo.

### II.1.1 Proceso de Desarrollo de Software

#### Definición de proceso

En un sentido muy general, “un proceso define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo” [19], constituyendo así un marco de trabajo de las tareas a realizar [35]. En términos concretos, “los procesos son unas series de pasos que involucran actividades, restricciones y recursos que producen una determinada salida esperada” [30]. Cabe destacar que estas actividades no se seleccionan aleatoriamente, sino que las mismas se organizan y relacionan con la salida a producir (representa a los objetivos por cumplimentar) [30]. La salida a producir puede ser vista como el valor agregado que genera el proceso.

De lo expuesto, se concluye que la importancia de implementar procesos radica en que los mismos imponen consistencia y estructuras sobre el conjunto de las actividades, permitiendo así capturar experiencias, examinar, comprender, controlar y mejorar las actividades que éste abarca [30].

#### Proceso de Desarrollo de Software.

El crear software involucra una serie de pasos y actividades que se especifican en un proceso cuya finalidad es su desarrollo y mantenimiento. Las diversas actividades, métodos y prácticas que se especifican en el proceso, deben estar asociadas a la producción de software [40]. Pero debido a la característica distintiva del software, la maleabilidad, también se deben detallar en la especificación del proceso de desarrollo de software, las estrategias que acompañan la evolución de éste (gestionar el cambio).

Cabe preguntarse cuáles son las tareas o actividades que integran a los procesos de desarrollo de software. [40] especifica cuatro actividades que son comunes a todos los procesos de desarrollo:

- *Especificación del Software*; se engloban todas las actividades orientadas a definir la funcionalidad y las restricciones del software para su operación.
- *Desarrollo del Software*; involucra todas las actividades dedicadas a la producción del software, según lo especificado en la actividad anterior.
- *Validación del Software*; esta actividad se centra en asegurar que el sistema proporcione lo que el cliente quiere.
- *Evolución del Software*; involucra a las actividades que permiten gestionar el cambio.

Las actividades descritas integran una de las categorías (desarrollo) que se especifican para los procesos denominados primarios [36]. Cabe destacar que los procesos primarios deben ser complementados con los procesos de soporte (los cuales dan apoyo a otros procesos realizando alguna función especial) y organizacionales (estos establecen, controlan y mejoran los otros procesos) [36].

El proceso de desarrollo de software implica el “conjunto de acciones, tareas y procedimientos involucrados en producir un sistema de software a través de su ciclo de vida” [20]. En este proceso se detallan las tareas y actividades que se realizan en cada fase, como así también las actividades que son continuas [20]. Cuando el proceso implica la construcción de algún producto, se suele referir al proceso como su ciclo de vida. Es por este motivo, que al proceso de desarrollo de software suele denominarse comúnmente ciclo de vida del software, debido a que describe la vida de un producto desde su concepción hasta su implementación, entrega, utilización y mantenimiento.

El proceso de desarrollo de software también puede ser visto como la interacción entre tres componentes; las políticas de la empresa, los mecanismos y estructuras [29]. Estos componentes están fuertemente interrelacionados. El seleccionar uno de estos componentes, tiene como consecuencia implicaciones o limitaciones para los otros dos [29]. Otro aspecto importante a tener en cuenta, es lo planteado por Yourdon [47], el cual sugiere que un sistema se modela en tres dimensiones: sus funciones, sus datos y su dinámica. Si bien, se pueden enumerar aún más definiciones y actividades sobre los procesos de desarrollo de software, la implementación de los mismos trae consigo orden y claridad en las actividades de creación del software.

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que suponga los requisitos del cliente. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas [40]. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. A continuación se explican algunas particularidades asociadas al desarrollo de software y que influyen en su proceso de construcción.

Un producto software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos software similar. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado en producto sino también durante el proceso de desarrollo.

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software. A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos [40].

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos [40]. Uno de los más reconocidos es el modelo de proceso de software de IEEE [18] que agrupa las actividades del proceso de software en tres categorías:

- *Grupo de Actividades de Gestión de Proyectos:* Entre estas actividades se encuentran la estimación de tiempos y recursos, planificación y seguimiento y control del proyecto.
- *Grupo de Actividades Orientadas al Desarrollo:* Agrupan las actividades técnicas relacionadas con el desarrollo de software tales como identificación de necesidades, definición de requisitos, diseño, implementación e instalación.
- *Grupo de Actividades de Soporte:* Abarca las actividades que se aplican para complementar tanto las actividades de gestión como las actividades técnicas, entre las que se incluyen evaluación, gestión de configuración, o documentación.

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos que permitan responder Quién debe hacer Qué, Cuándo y Cómo debe hacerlo [23].

En la Figura 2.1 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así las interrogantes se responden de la siguiente forma:

- *Quién:* Las Personas participantes en el proyecto de desarrollo desempeñando uno o más roles específicos.
- *Qué:* Un Artefacto es producido por un rol en una de sus Actividades. Los Artefactos se especifican utilizando Notaciones específicas. Las Herramientas apoyan la elaboración de Artefactos soportando ciertas Notaciones.
- *Cómo y Cuándo:* Las Actividades son una serie de pasos que lleva a cabo un rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

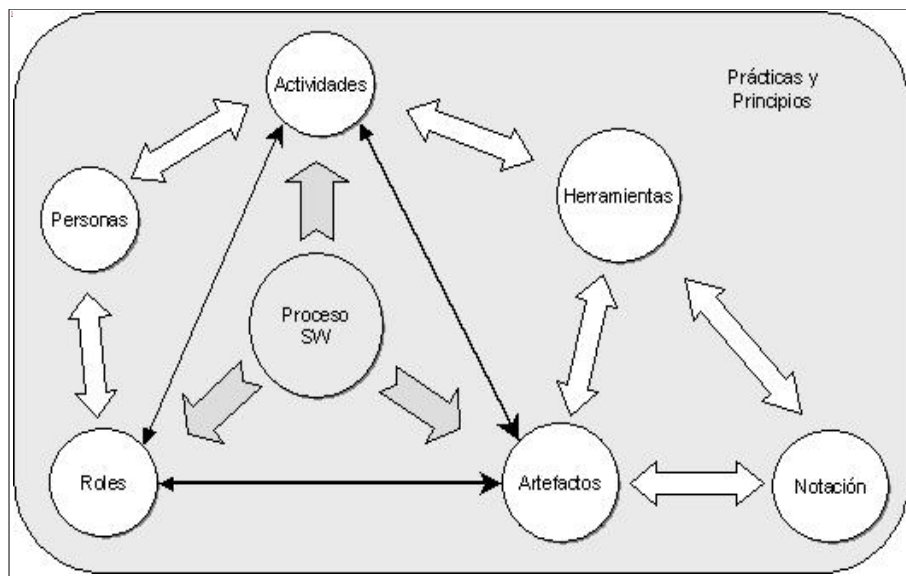


Figura 2.1. Relación entre elementos del proceso del software [24]

### II.1.2 Proceso de Negocio

Pressman [35] expresa que proceso de negocio es “*un conjunto de tareas lógicamente relacionadas que se llevan a cabo para obtener un determinado resultado de negocio*”. Dentro del proceso de negocio, se combinan las personas, los equipos, los recursos materiales y los procedimientos de negocio con el objeto de producir un resultado concreto. Cada proceso de negocio posee un cliente bien definido -una persona o grupo que recibe el resultado (por ejemplo: una idea, un informe, un diseño, un producto). Además, los procesos de negocio cruzan los límites organizativos. Requieren que distintos grupos de la organización participen en las tareas lógicamente relacionadas que definen el proceso. El negocio global se puede segmentar de la siguiente manera: negocio, sistemas de negocio, proceso de negocio, subprocesos de negocio. Cada uno de los sistemas de negocios (también llamados función de negocios) están compuestos por uno o más procesos de negocio, y cada proceso está definido por un conjunto de subprocesos.

Un proceso de negocio [16] se puede ver como un conjunto estructurado de tareas, que contribuyen colectivamente a lograr los objetivos de una organización. Los procesos de negocio de una organización son parte de su cultura. Se registran y difunden en manuales de procedimientos, diagramas de flujo y hasta en forma verbal. Son la base operativa de una empresa y el éxito de la misma depende fuertemente de la eficiencia con que sean gestionados. Una mala gestión de los procesos trae aparejados altos costos, baja productividad, e inadecuados tiempos de respuesta tanto frente a las oportunidades como a las amenazas. Por esto el modelado de procesos viene siendo objeto de estudio desde hace ya tiempo.

Como soporte del modelado de procesos de negocio dentro de una organización, en la década del 90 surge la tecnología de *workflow*. Esta tecnología permitió representar total o parcialmente los procesos de negocio en un formato entendible por una máquina.

En síntesis un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un objetivo de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas.

### **II.1.3 Reglas de Proceso de Negocio**

En una organización, tanto los procesos como los datos, están restringidos por las reglas del negocio. Estas reglas aseguran que la actividad de la empresa se lleva a cabo de acuerdo a restricciones impuestas desde el entorno (leyes o normas) o desde dentro de la propia organización, de modo que nunca sea posible llevar a cabo acciones no válidas.

Una *regla de negocio (RN)* es una sentencia que indica cómo se hace el negocio, es una guía con restricciones de los estados y procesos de una organización [21]. En realidad, la información puede ser manipulada por muchos programas distintos, así una empresa puede tener un departamento de contabilidad que controle todo lo relacionado con compras, cobros, etc., y otro departamento técnico, que esté interesado en relacionar diversos parámetros de producción con los costos. La visión que ambos departamentos tendrán de la información y sus necesidades será distinta, pero en cualquier caso siempre se deberán respetar las reglas de negocios.

Las RN tienen una estrecha relación con las reglas definidas en Bases de Datos [41], las cuales utilizan sus reglas para responder ante eventos que sucedan en el mundo real y que deben ser reflejados en la base de datos. La estructura de dichas reglas es evento – precondición – acción [17], donde un *evento* es un suceso instantáneo de interés para el

negocio y asociado a un punto en el tiempo, una *precondición* es aquella condición que debe ser verificada antes de realizar la acción; y la *acción* define aquello que debe hacerse después de verificar la precondición.

## **Tipos de Reglas de Negocio**

La clasificación presentada a continuación se basa en la realizada por P. Solivares en [39].

- *Reglas del Modelo de Datos:* Este primer grupo de reglas de negocios engloba todas aquellas reglas que se encargan de controlar que la información básica almacenada para cada atributo o propiedad de una entidad u objeto sea válida; ejemplo de este tipo; no hay precios de artículos negativos, el sexo de una persona solo puede ser masculino o femenino, una fecha siempre debe ser una fecha válida, etc.
- *Reglas de Relación:* Otro grupo importante de reglas incluye todas aquellas reglas que controlan las relaciones entre los datos. Estas reglas especifican, por ejemplo, que todo pedido debe ser realizado por un cliente, y que el mismo debe ser atendido. Además, una vez que un cliente haya hecho algún pedido, se deberá garantizar que no es posible eliminarlo, a menos que previamente se eliminen todos sus pedidos.
- *Reglas de Derivación:* Es frecuente que a partir de cierta información se pueda derivar otra, este conjunto de reglas especifican y controlan la obtención de información que se puede calcular a partir de la ya existente. Por ejemplo, el total de un pedido se puede calcular a partir de las distintas líneas que lo componen, mientras que el total de cada línea se puede calcular a partir del número de unidades vendidas y el precio por unidad.
- *Reglas de Restricción:* Otro grupo de reglas de negocios es el compuesto por las reglas de restricción, que restringen los datos que el sistema puede contener. Nótese que este grupo de reglas se solapa en cierto modo con las reglas del modelo de datos, dado que aquellas también impiden la introducción de datos erróneos, como se vio anteriormente. La diferencia estriba en que las reglas de restricción restringen el valor de los atributos o propiedades de una entidad más allá de las restricciones básicas que sobre las mismas existen: por ejemplo, para un saldo existe una regla básica (regla del modelo de datos) que indica que éste debe ser un número. Adicionalmente puede haber una regla que indique que el saldo nunca puede ser menor que cierta cantidad tope establecida para cierto tipo de clientes. Esta sería lo que aquí se denomina una regla de restricción, y la diferencia fundamental se manifiesta en el hecho de que este tipo de reglas requiere para su verificación del acceso a otros fragmentos de información, algo que no sucede con las reglas del modelo de datos.
- *Reglas de Flujo:* Este grupo de reglas de negocios incluye aquellas reglas que determinan y limitan cómo fluye la información a través de un sistema. Por ejemplo, un cliente puede hacer una petición de análisis a un laboratorio, que anota un encargado, hecho esto, se genera una orden para uno o más analistas, estos realizan las mediciones correspondientes y devuelven las partes con la información pertinente, a partir de la cual se genera un informe de análisis, que será un análisis válido sólo cuando sea firmado por los responsables de garantizar su corrección. Estas reglas indican qué camino recorre la información y obligan a que se sigan solo los caminos válidos.

- *Reglas de Decisión:* Hay otro grupo de reglas de negocios, que son las encargadas de representar las reglas para la toma de decisiones, las cuales determinan cuando, donde y quién debe realizar una tarea específica.

#### **II.1.4 Modelo**

Un modelo de un sistema es una descripción o una especificación de ese sistema y su entorno para desempeñar un determinado objetivo. Los modelos se presentan normalmente como una combinación de texto y dibujos. [9]

Un modelo es una descripción de todo o parte de un sistema bien escrito en un lenguaje bien definido. El hecho de que un modelo este escrito en un lenguaje bien definido es de gran importancia, ya que supone que tiene asociadas una sintaxis o una semántica bien definida. Esto permite la interpretación automática por parte de transformadores o compiladores de modelos. [32].

#### **II.1.5 Metamodelo**

Se puede considerar que un metamodelo es una definición precisa de los constructores y reglas necesarias para definir la semántica de los modelos [31]. También es un modelo y debe estar escrito en un lenguaje bien definido.

El metamodelo define la estructura, semántica y restricciones para una familia de modelos. Proporciona la representación específica de elementos de un dominio específico. Su uso facilita la unificación de una terminología común, reducción de las ambigüedades, la reproducción de una documentación completa, control de la consistencia de los modelos y la trazabilidad de los elementos del modelo. [7]

Por ejemplo el Metamodelo del Diagrama de Actividad implementado en AToM<sup>3</sup> [4][11], define los conceptos y las reglas que se necesitan para crear los modelos de comportamiento, por ejemplo diagramas de actividad de UML.

#### **II.1.6 Modelo de Procesos de Negocio**

Un modelo de Procesos de Negocio es una abstracción que muestra cómo funciona la organización. Para su realización se basa en los procesos estratégicos del negocio, ya que por ellos se puede identificar actividades y objetivos que interesan al cliente. El uso de un modelo permite un fácil seguimiento al Proceso de Negocio y vuelve más sencilla la documentación, logrando una mejor descripción y entendimiento. [3].

Los procesos estratégicos se refieren principalmente a procesos de planificación que están ligados a factores clave dentro de la organización. Generalmente están vinculados a responsabilidades de la dirección de la empresa, los cuales dan la orientación al negocio de hacia dónde debe enfocarse. [14]

Contar con los modelos de negocio elaborados dentro de la organización otorga las siguientes ventajas:

- Permite hacer un mejor seguimiento a las actividades relevantes del proceso.

- Abre un puente de comunicación entre los directos involucrados en el proceso (clientes, analistas, desarrolladores, gerentes, etc.), permitiendo que para todos ellos se muestre el proceso de una forma clara y concisa, orientando mejor el trabajo hacia un fin específico.
- Fácil localización de problemas dentro del proceso.
- Así como permite una mejor detección de fallas, puede también generar soluciones a estos problemas.

Según Morris y Brandon [26] un modelo de procesos de negocios es una descripción gráfica de la estructura y actividades de la operación que muestra las relaciones entre las etapas de trabajo y su secuencia. En conjunto, estas representaciones reflejan el flujo de trabajo.

Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

Todo proceso de negocio se compone de una serie de elementos que permiten desarrollar las labores para los cuales fue propuesto, trabajando todo como un conjunto hacia un objetivo final. Un proceso de negocio se compone de las siguientes partes [42] (ver Tabla 2.1):

*Tabla 2.1. Componentes de los procesos de negocio.*

<b>Nombre</b>	<b>Descripción</b>
Subproceso	Es parte de un proceso de más alto nivel que posee sus entradas, salidas y actividades.
Actividades	Son los pasos que deben ejecutarse para transformar las entradas del proceso en el resultado esperado. Son partes del proceso de negocio que están completamente atomizadas.
Decisiones	Teniendo en cuenta los objetivos y la unión de todo el sistema, se toma una decisión que beneficie y de valor agregado a lo que busca el cliente.
Entradas	Son aquellos insumos, datos o información del cliente utilizados a lo largo del proceso.
Salidas	Son los productos obtenidos como resultado del proceso.
Recursos o mecanismos	Son las herramientas que permiten que se lleve a cabo el proceso, ejecutando sus actividades.
Políticas, Controles y Manuales	Las políticas, controles y manuales, son las reglas que gobiernan el proceso y por las cuales deben regirse las actividades que se ejecutan.
Propietario	Es el encargado de realizar la actividad. Pueden ser individuos, grupos de personas o componentes de la organización.
Objetivo	Es una característica que indica la función por la cual existe al interior del proceso de negocio al que pertenece.

## II.2. MARCO METODOLÓGICO

A continuación se describen las técnicas y procedimientos que se utilizaron en este trabajo para dar solución al problema planteado.

### II.2.1 Notación de Modelado de Proceso de Negocio (BPMN)

El BPMI (Business Process Management Initiative) [46] ha desarrollado una notación gráfica estándar BPMN (Business Process Modeling Notation) para modelar flujos de proceso de negocio, cuya base formal son las redes de Petri. [8]

BPMN es un estándar de representación gráfica de procesos que proporciona una notación legible que pueda ser comprendida por los analistas de negocios; por los técnicos desarrolladores; y finalmente por las personas de negocio. BPMN es el puente de unión entre la descripción operacional de una organización y el sistema de información que le dará soporte. [10]

Esta especificación define la notación y semántica de un diagrama de proceso de negocio. Un diagrama de proceso de negocio está compuesto de elementos agrupados en dos categorías básicas: la primera, son los elementos centrales con los cuales es posible desarrollar modelos de procesos simples y, la segunda, incluye los elementos que permiten la creación de modelos de procesos de negocios complejos o de alto nivel. Las cuatro clases que componen la lista de elementos centrales son los Objetos de Flujo, Objetos de Conexión, Carriles y Artefactos. [8]

BPMN es un estándar internacional de modelado de procesos, es independiente de cualquier metodología de modelado de procesos. Crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos y permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización.

### II.2.2 AToM<sup>3</sup>

AToM<sup>3</sup> [4] es una herramienta para el modelado multi-paradigma (que incluye metamodelado, modelado multi-formalismo y en múltiples niveles de abstracción) que desarrolló Juan de Lara Jaramillo en colaboración con el laboratorio MSDL de la Universidad de McGi Il en Montreal. La idea principal de la herramienta es: “*todo es un modelo*”, en el sentido de que incluso la interfaz de usuario de AToM<sup>3</sup> es un modelo, que se interpreta al cargarse, y que por tanto puede modificarse.

AToM<sup>3</sup> se puede utilizar como una herramienta de metamodelado para definir formalismos de cualquier tipo de esquema conceptual; esta herramienta MetaCASE permite la creación, edición, transformación, simulación y optimización de metamodelos y modelos, y la generación de código en lenguaje Python. Además, utiliza técnicas de escritura gráfica y gramática de grafos para realizar las transformaciones entre formalismos.

En AToM<sup>3</sup>, los metamodelos son creados y editados en un ambiente que emplea el meta-formalismo DER (Diagramas Entidad Relación), extendido con restricciones. Este es el (meta) formalismo que se carga en AToM<sup>3</sup> por defecto. Utilizando DER, el usuario puede definir el metamodelo del formalismo que desee definir, simplemente dibujando el diagrama

entidad-relación correspondiente. Por ejemplo, se ha utilizado el metaformalismo DER para definir el formalismo “Autómatas”. Una vez que se ha definido el metamodelo de un formalismo, éste puede cargarse de nuevo en AToM<sup>3</sup>, de tal forma que el comportamiento de la herramienta cambiará de acuerdo a la sintaxis definida en el metamodelo.

Esta manipulación se puede expresar en AToM<sup>3</sup> mediante gramáticas de grafos, que a su vez también son modelos. Para definir una gramática de grafos, se ha de cargar el formalismo “*gramáticas de grafos*” [34], junto con el formalismo para el que se desea definir la transformación (“*Autómatas*”, “*Interacción de Procesos*”, etc.) Una vez definida la gramática es posible aplicarla sobre modelos definidos con el formalismo, para su transformación.

La Gramática de Grafos en AToM<sup>3</sup> permite expresar restricciones y describir las transformaciones entre grafos de manera gráfica. La Gramática de Grafos está compuesta por reglas que mapean desde el lado izquierdo (LHS) al lado derecho (RHS), condiciones y acciones. En las condiciones, se define cuándo una regla puede ser aplicada; en las acciones, se especifica lo que se va a realizar y pueden tener asociadas una prioridad. El LHS puede ser especificado en un formalismo diferente al RHS, y en este caso se trataría de una transformación entre diagramas, o pueden estar especificados en el mismo formalismo, y en ese caso se podría tratar del refinamiento de un mismo diagrama.

AToM<sup>3</sup> permite la descripción de lenguajes visuales específicos de dominio mediante metamodelos. Por ejemplo, se ha usado la herramienta para describir lenguajes visuales tales como diagramas de actividad, diagramas de clases, diagramas de proceso, Redes de Petri, entre otros [1].

En el ámbito de la verificación de sistemas AToM<sup>3</sup> proporciona una técnica de verificación automática denominada model checking, orientada a realizar la verificación mediante la exploración exhaustiva del espacio de estados del modelo en busca de situaciones que no cumplan con las propiedades especificadas. En este último caso la prueba es parcial, pero conservativa, en el sentido de que si el sistema cumple las propiedades, entonces el sistema real también lo hace.

III.1 INTRODUCCIÓN

En este capítulo se describen las actividades que forman parte del procedimiento de construcción y comprobación del metamodelo así como su implementación en una plataforma con enfoque multiparadigma denominada ATOM<sup>3</sup> [4] [11].

El metamodelo propuesto toma en cuenta todos los elementos conceptuales de la notación BPMN y genera cuatro vistas que pueden complementarse e integrarse en un único modelo. El trabajo propuesto se describe en la Figura 3.1

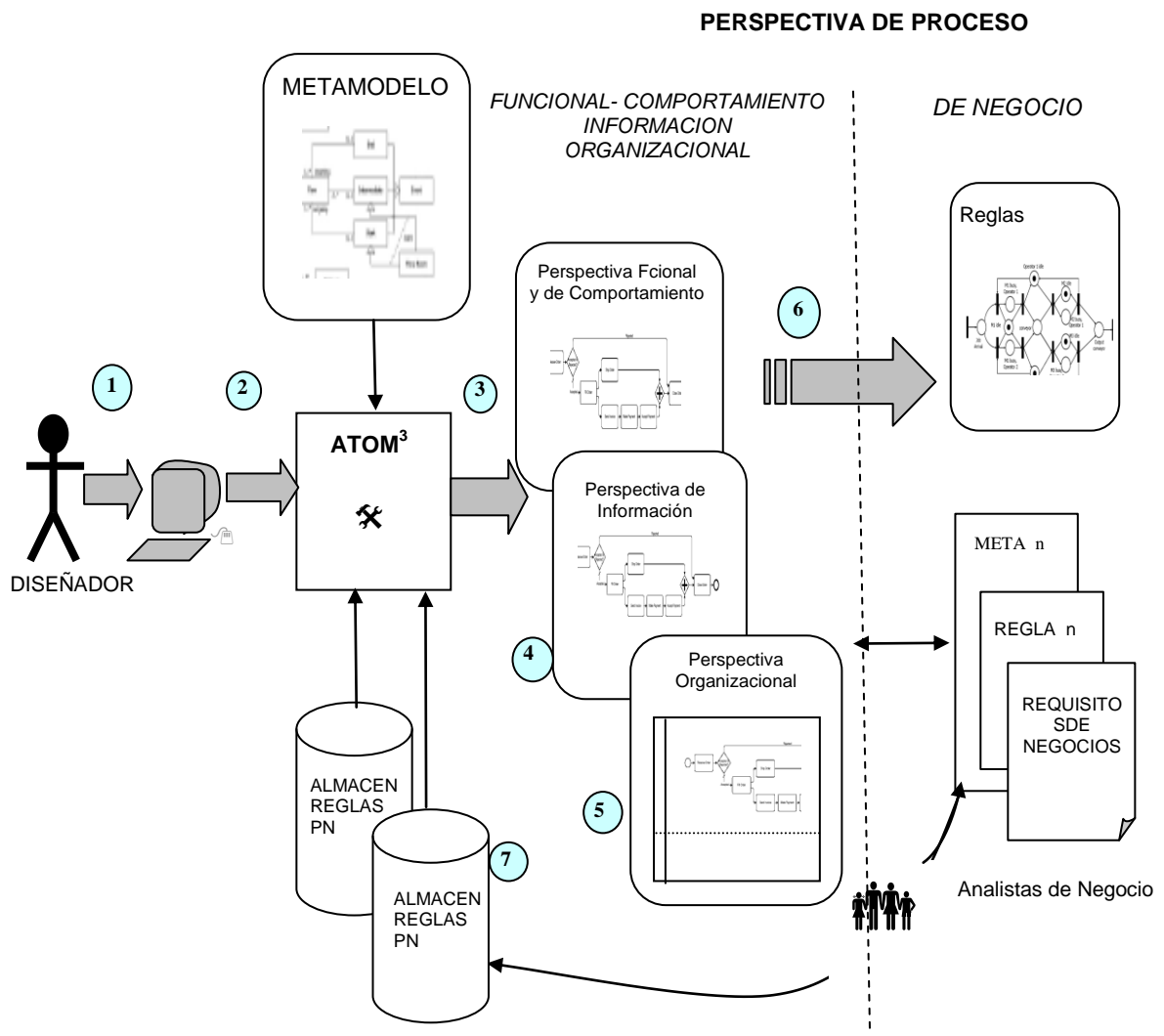


Figura 3.1. Representación de la propuesta de trabajo.

Una vez que el diseñador carga un modelo de Proceso Software (1 y 2), el mismo puede visualizarse desde una perspectiva funcional y de comportamiento (3) (el modelo presenta los elementos que son esenciales en los procesos de negocio: actividad, evento, gateway, secuencia de flujo); la perspectiva de información (4) (se visualizan los artefactos que se manejan e intercambian dentro del proceso); la perspectiva organizacional (5) (muestra los elementos que mejoran la visualización de los procesos, pero no tienen incidencia sobre los mismos, tales como pool, lane y agrupamiento); en (6) se visualiza el comportamiento del modelo a partir de la aplicación de las reglas de la gramática de grafos definida para el formalismo BPMN. Por último en (7) se prevé un repositorio de reglas de negocio que serán aplicadas para comprobar la validez conceptual del modelo.

## III.2 PROCEDIMIENTO DE CONSTRUCCIÓN DEL METAMODELO

Los pasos para la construcción del metamodelo involucra el análisis de la notación BPMN [8], sus aspectos sintácticos, semánticos, sus relaciones y representación gráfica. En los siguientes apartados se describen detalladamente cada uno de los pasos.

### III.2.1 Análisis de la Notación BPMN

BPMN es un lenguaje o notación gráfica que permite diseñar y gestionar procesos de negocio con Diagramas de Procesos de Negocio (DPN).

BPMN organiza la notación en cuatro categorías específicas. Estas categorías permiten representar fácilmente los elementos básicos de un DPN. Las cuatro categorías básicas de elementos son:

- *Objetos de Flujo.*
- *Objetos de Conexión.*
- *Swimlanes y*
- *Artefactos.*

Dentro de estas categorías de elementos se pueden incluir variaciones adicionales o información para soportar requerimientos complejos sin agregar demasiada complejidad al diagrama.

Los *Objetos de Flujo* son los elementos gráficos que definen el comportamiento de un proceso de negocio, ellos son; gateway, actividad y evento.

Los *Objetos de Conexión* conectan los elementos que forman el diagrama mostrando el flujo de control del proceso. Estos objetos son: “*flujo secuencial*”, “*flujo de mensaje*”, “*asociación*”.

Los *Swimlanes* organizan las actividades en categorías separadas visualmente para ilustrar diferentes capacidades funcionales o responsabilidades. Los elementos son: “*pool*” y “*lanes*”.

Los *Artefactos* permiten mostrar información adicional para una situación específica de modelado, indicando los datos, documentos y otros objetos que son usados. Son conectadas a los elementos del flujo usando una asociación. Los artefactos son: “*objeto de dato*”, “*agrupación*” y “*anotación textual*”.

### III.2.1.1 Características Semánticas de los Elementos de BPMN.

A continuación se explica la semántica de cada elemento de un DPN. Como se menciona en el apartado anterior están agrupados en:

#### a) *Objetos de Flujo:*

✍ *Actividad:* Son acciones ejecutadas durante el proceso de negocio. Representan el trabajo que una empresa u organización realiza dentro de un proceso de negocio. El tipo de actividad puede ser *Tarea Atómica* o *Proceso*.

También se pueden definir a las actividades como “acción sobre un objeto”, es decir una actividad se denomina siempre con un verbo (acción) y un sustantivo (objeto).

#### ➤ *Tipos de actividades*

- *Tareas:* Son actividades que ejecutan una simple acción. Representan el nivel más bajo del proceso, por lo que no pueden ser descompuestas en sub-tareas. La tarea es una actividad indivisible. Generalmente un usuario o una aplicación son usados para ejecutarlas. Las tareas al ser un tipo de actividad heredan los atributos de una actividad.
- *Subprocesos:* Es un conjunto de actividades incluidas dentro de un proceso. Es una actividad compuesta, que permite diagramar con más profundidad los procesos suministrando más información y claridad al lector. Los subprocesos son actividades divisibles. Los subprocesos al ser un tipo de actividad heredan los atributos de una la misma.

Una actividad puede tener distintos marcadores, los cuales especifican el comportamiento particular de las actividades durante su ejecución; ellos son:

- *Loop y múltiples instancias:* Son actividades que pueden ser configuradas para iterar en modo estándar (*while o until*) o para soportar la ejecución de múltiples instancias en paralelo (*foreach*). Una actividad en modo estándar se indica con una flecha circular y una actividad con múltiples instancias tiene un símbolo con un par de barras paralelas.
- *Ad hoc:* Contiene un conjunto de actividades que ocurren en cualquier orden, por lo que el flujo de control está sin estructurar.
- *Compensación:* Es una actividad fuera del flujo normal del proceso, capaz de revertir la lógica de una actividad, deshaciendo sus efectos, después de que se hayan completado. Se indica con un símbolo de rebobinar.
- *Subproceso:* Indica que es una actividad compuesta.

☞ *Evento*: Indica “algo” que sucede durante el curso de un proceso de negocio. Los eventos afectan el flujo del proceso y normalmente tienen una causa (trigger) o un impacto (resultado). Hay tres tipos de eventos: inicial, intermedio y final.

#### ➤ *Tipos de eventos*

El evento, tal y como se definió previamente es “algo” que ocurre en el transcurso de un proceso de negocio. Pueden ser: inicial, intermedio y final. Los eventos pueden influir en el curso del mismo. Tiene un único atributo, el cual es heredado por el evento inicial, intermedio y final.

Además cada uno de estos tipos de evento, estos a su vez se clasifican en subtipos dependiendo del impacto en el flujo del proceso. Estos subtipos son representados por diferentes marcadores que tiene cada tipo de evento.

- *Inicial*: Se utiliza para iniciar un proceso. Estos eventos comienzan el flujo del proceso, por lo cual no deben tener flujos de secuencia entrantes. Sólo se deberá colocar un evento de inicio por proceso de negocio, ya que más de uno dificulta el entendimiento del diagrama.

Hay diferentes “desencadenantes” que indican las circunstancias que inician el Proceso y para ello existen los marcadores de eventos de inicio que son: mensaje, tiempo, regla y múltiple

- *Intermedio*: Indica el lugar donde un evento sucede entre el inicio y el fin del proceso, el cual afecta el flujo del proceso. Un evento intermedio que se encuentre en el flujo normal de un proceso puede ser usado para dos propósitos: para responder a un disparador de evento, o para disparar un evento.

Los diferentes marcadores de eventos intermedios pueden indicar como se captura o activa el evento. Estos son: mensaje, tiempo, regla, múltiple, cancelar, error y compensación.

- *Final*: Indica dónde termina un proceso. Estos eventos terminan el flujo del proceso, por lo cual no deben tener flujos de secuencia salientes.

Existen diferentes “resultados” que indican las circunstancias específicas por las que termina el proceso. Los diferentes marcadores de eventos de fin son: mensaje, múltiple, cancelar, error, compensación y terminal

Dentro de BPMN existen muchas formas de iniciar o finalizar un proceso e igualmente existen muchas cosas que pueden llegar a suceder durante el transcurso del proceso, por lo tanto existen diferentes tipos de eventos de inicio, eventos de fin y eventos intermedios. Estos tipos los podemos ver en la Tabla 3.1.

Tabla 3.1: Tipos de eventos de BPMN

Evento Tipo	EVENTO INICIAL	EVENTO INTERMEDIO	EVENTO FINAL
SIN ESPECIFICAR	No tiene establecida una condición para dar inicio.	Indica algo que ocurre o puede ocurrir dentro del proceso, sólo se pueden utilizar dentro de la secuencia del flujo.	Indica que un camino del flujo llegó al fin.
MENSAJE	Un proceso inicia cuando un mensaje es recibido.	Indica que un mensaje puede ser enviado o recibido. Si es de recepción, indica que el proceso no continúa hasta que el mensaje sea recibido.	Permite enviar un mensaje al finalizar el flujo.
TIEMPO	Indica que un proceso inicia cada ciclo de tiempo o en una fecha específica. Se puede fijar una hora-fecha específica en la que se activara el inicio del proceso.	Indica una espera dentro del proceso. Este tipo de evento puede utilizarse dentro del flujo de secuencia indicando una espera entre las actividades.	
ERROR		Lanza o captura un error	Permite Genera un error al finalizar el flujo.
CANCELACION		Ejecuta la cancelación den una actividad	Cancela la transacción
COMPENSACION		Lanza el manejo de compensaciones	Ejecuta la compensación
CONDICIÓN O REGLA	Un proceso inicia cuando una condición de negocio se cumple.	Se utiliza para esperar que una condición de negocio se cumpla para iniciar el manejo de excepciones	
MULTIPLE	Dos o más triggers pueden iniciar el proceso; es suficiente con que uno de ellos ocurra	Dos o más triggers pueden reanudar un proceso en espera; es suficiente con que uno de ellos ocurra	Varios resultados son requeridos para acabar el proceso
TERMINAL			Termina todas las actividades del proceso. No ejecuta manejo de excepciones ni de compensación.

☞ *Gateway*: Se usan para controlar la divergencia y convergencia de la secuencia de flujo. Con este elemento se determinan selecciones, paralelismos, fusiones y uniones de las trayectorias. Pueden agregarse marcadores internos para indicar el tipo de comportamiento de control.

### ➤ *Tipos de Gateway*

Los gateways son los elementos que permiten realizar el control del flujo dentro de un diagrama BPMN. Existen cinco tipos de marcadores internos de un gateway que indican el comportamiento del control de flujo. Estos tipos de gateway son:

- *Gateway exclusivo basado en datos (XOR)*: Son los más comúnmente usados. El nodo de *decisión* crea trayectorias alternativas basadas en condiciones definidas. Al *XOR* se lo puede utilizar como elemento de bifurcación o unión. La notación permite incluso usar el *XOR* como una unión de entrada y bifurcación de salida con un solo símbolo.

La bifurcación (*XOR-Split*) ocurre cuando en un punto del flujo basado en los datos del proceso se escoge exactamente un flujo de secuencia de entre las alternativas existentes.

La unión (*XOR-Join*) ocurre cuando la compuerta espera a que un flujo incidente se complete para activar el flujo saliente, sólo una rama puede ser ejecutada.

- *Gateway exclusivo basado en eventos (XOR)*: La compuerta exclusiva basada en eventos representa un punto del proceso donde se escoge un camino de varios disponibles, pero la decisión no se basa en datos del proceso sino en eventos, es decir este tipo de decisión representa un punto de bifurcación en el proceso en donde las alternativas están basadas en eventos que ocurren en ese punto del proceso, más que en condiciones. El evento elegido en la decisión determina la trayectoria elegida.
- *Gateway paralelo (AND)*: Son lugares en el proceso en donde se definen múltiples trayectorias paralelas. Proporcionan un mecanismo para sincronizar (*And-Join*) o para crear flujos paralelos (*And-Split*).

La bifurcación (*And-Split*) se utiliza cuando varias actividades pueden realizarse concurrentemente o en paralelo., es decir, todos los caminos salientes serán activados simultáneamente.

La unión (*And-Join*) permite sincronizar varios caminos paralelos en uno solo, espera a que todos los flujos incidentes completen antes de activar el flujo saliente.

- *Gateway inclusivo (OR)*: Son decisiones en donde hay más de un resultado posible. Las trayectorias alternativas están basadas en expresiones condicionales contenidas dentro del flujo de secuencia resultante. La evaluación verdadera de una condición no excluye la evaluación de las otras condiciones Al ser independiente cada trayectoria, todas las combinaciones pueden ser seguidas o al menos una de ellas. Una o más ramas pueden ser ejecutadas.

La bifurcación (*OR-Split*) se utiliza cuando en un punto se activan uno o más caminos de varios caminos disponibles, basado en los datos del proceso.

La unión (*OR-Join*) se utiliza para sincronizar caminos activados previamente por una compuerta inclusiva usada como punto de unión.

- *Gateway complejo (Complex)*: Son decisiones en donde hay definiciones más avanzadas de comportamiento. Considera situaciones que no son fácilmente tomadas a través de otros tipos de decisiones. Pueden ser usadas para combinar un conjunto de decisiones simples vinculadas en una situación única más compacta.

La bifurcación (*Complex-Split*) es utilizada para controlar puntos de decisión complejos.

La unión (*Complex-Join*) permite continuar al siguiente punto del proceso cuando una condición de negocio se cumple.

### **b) Objetos de Conexión:**

- ✍ *Flujo de secuencia:* Se usa para mostrar el orden en que las actividades son ejecutadas en un proceso. El flujo de secuencia describe la secuencia temporal o lógica en el cual se combinan los elementos de flujo, es decir las actividades, eventos y Gateway.
- ✍ *Flujo de mensajes:* Indican el flujo de mensajes entre dos entidades separadas y preparadas para enviarlos y recibirlos. Pueden ser entidades de negocio o roles de negocio. Estos flujos no pueden conectar dos objetos dentro de la misma entidad.
- ✍ *Asociación:* Es un elemento de conexión que se usa para asociar información, datos, documentos, y otros tipos de artefactos con los objetos de flujo. Las asociaciones se usan para mostrar entradas y salidas de los objetos de flujo.

### **c) Artefactos:**

- ✍ *Objetos de datos:* Permiten mostrar la información que una actividad necesita, como las entradas y las salidas. Es decir, representan los documentos, la información y otros objetos que son usados o actualizados durante el proceso. Los objetos de datos no afectan directamente los flujos de secuencia o los flujos de mensajes del proceso.
- ✍ *Agrupamiento:* Se utiliza para agrupar un conjunto de elementos de un DPN, ya sea para efectos de documentación o análisis, pero no afecta la secuencia del flujo.
- ✍ *Anotación textual:* Es un mecanismo para que el modelador pueda proporcionar información textual adicional sobre el proceso al lector de un DPN. No afecta al flujo del proceso.

### **d) Swimlanes:**

- ✍ *Pool:* Actúa como un contenedor gráfico para separar un conjunto de actividades dentro de otros *pools*. Representa un “participante” de un proceso, este puede ser una entidad de negocio (una empresa, división de la empresa, o un cliente, etc.) o una función de negocio (un comprador o vendedor, etc.) que controla o es responsable de un proceso de negocio.
- ✍ *Lanes:* Representan subdivisiones dentro de un *pool*. Son usadas para organizar y categorizar las actividades y procesos del mismo.

Considerando lo expuesto anteriormente, en la Figura 3.2, se presenta un mapa conceptual que representa la semántica del lenguaje BPMN, donde los nodos representan los elementos asociados al lenguaje, y los enlaces representan las relaciones heredadas entre los elementos.

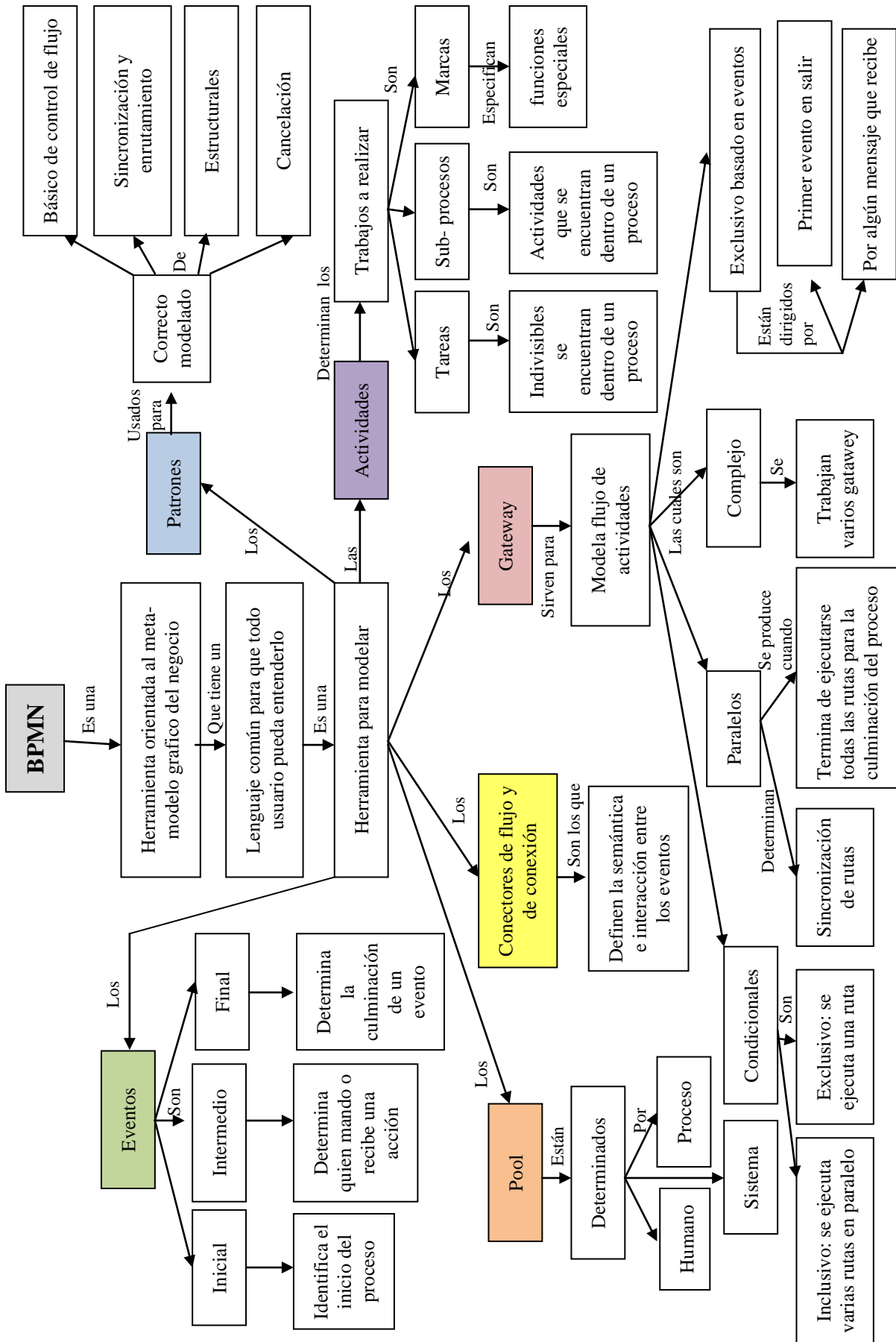


Figura 3.2. Mapa conceptual de la semántica del lenguaje BPMN.

Para evitar problemas semánticos en la construcción de los procesos de negocio se tiene en cuenta diferentes patrones. [44][45]

Estos patrones guían el proceso de modelado e implementación de soluciones y son tomados en cuenta para la definición de reglas de negocio del proceso de software. Los patrones representan “*workflows*” y son [43]:

- 1) Patrones básicos de flujo de control.
- 2) Patrones avanzados de división y unión.
- 3) Patrones estructurales.
- 4) Patrones de cancelación.

Por cada uno de ellos se presenta una descripción breve, características y el contexto donde pueden ser usados, y en los casos que corresponda el/los sinónimos que tiene el elemento descripto.

## A) PATRONES BÁSICOS DE FLUJO DE CONTROL

### 1. *Patrón de secuencia*

*Descripción:* Se ilustra como una serie de actividades conectadas por flujos de secuencia. La secuencia indica que una actividad será habilitada, sólo hasta que la actividad anterior haya terminado.

*Característica:* La secuencia es la base fundamental para los procesos. Se utiliza para construir una serie consecutiva de ejecución de tareas una tras otra. Dos tareas forman parte de una secuencia si un flujo de salida es el flujo de entrada de otra, sin verse afectado por ningún tipo de condición.

*Contexto:* El contexto de este patrón, está asociado a una condición; una instancia de este patrón no puede iniciarse de nuevo hasta que haya terminado la ejecución del hilo de control anterior.

### 2. *Patrón de división paralela*

*Descripción:* Una división paralela es un punto del proceso donde un camino es dividido en dos o más ramas paralelas las cuales son ejecutadas al mismo tiempo. Utiliza un punto de división paralelo en modo división.

Sinónimo: AND-Split.

*Característica:* El patrón de división paralela permite un solo hilo de ejecución, para ser dividido en dos o más ramas de actividad que pueden ejecutar tareas simultáneamente. Estas ramas pueden o no estar sincronizadas en algún momento del futuro.

*Contexto:* No hay condiciones específicas de contexto para este patrón.

### **3. Patrón sincronizador**

*Descripción:* La sincronización es un punto en el proceso donde dos o más ramas del proceso se unen en una sola. Se llama sincronización porque se espera a que todas las ramas entrantes se completen antes de continuar con la siguiente actividad. Utiliza el punto de decisión paralelo en modo unión.

*Sinónimos:* Unión paralela, AND-Join,

*Característica:* La sincronización proporciona un medio de comunicar los hilos de ejecución de dos o más ramas paralelas. En general, estas ramas son creadas usando la división paralela con anterioridad a construir el modelo de proceso. El hilo de control se pasa a la tarea inmediatamente después de que todos los hilos de la sincronización hayan acabado.

*Contexto:* Para toda señal que se reciba en el lugar, hasta que no haya recibido todas, no pasará al siguiente paso.

### **4. Patrón de elección exclusiva**

*Descripción:* La decisión exclusiva representa un punto en el proceso donde se debe escoger un solo camino de varios disponibles dependiendo de una decisión o de datos del proceso. Está marcada por un punto de decisión exclusivo basado en datos o en eventos.

*Sinónimo:* XOR-Split.

*Característica:* El patrón de elección exclusiva permite a los hilos de control ir a una tarea específica dependiendo de la tarea precedente, de los valores de los elementos del proceso, del resultado de una expresión de evaluación o de algún otro mecanismo de selección. La decisión de enrutamiento es dinámica permitiendo ser aplazadas hasta el último momento posible en tiempo de ejecución.

*Contexto:* Existe una condición para este patrón, el mecanismo que evalúa la elección exclusiva es capaz de acceder a cualquier tipo de datos necesarios, otros elementos o recursos para determinar por cuál rama de las salientes debe dirigirse el hilo de control.

### **5. Patrón de fusión simple**

*Descripción:* La fusión simple es un punto del proceso donde dos o más caminos alternativos convergen en uno solo. Es un supuesto de este patrón que los caminos alternativos no son ejecutados en paralelo. Usa un punto de decisión exclusivo basado en datos.

*Sinónimos:* Unión exclusiva, XOR-Join.

*Característica:* La simple fusión de patrón proporciona un medio de unión de distintas tramas sin necesitar de la sincronización. Permite la simplificación del modelo eliminando la necesidad de replicar explícitamente una secuencia de tareas que es común a dos o más tramas.

*Contexto:* Hay una condición asociada con la distribución: el lugar donde se produce la fusión es seguro y nunca puede contener más de un token.

## B) PATRONES AVANZADOS DE DIVISIÓN Y UNIÓN

Aquí se presentan una serie de pautas que caracterizan una ramificación más compleja y la fusión de conceptos que se plantean en los procesos de negocio. Aunque es relativamente frecuente, en la práctica, estos patrones a menudo no están soportados directamente o incluso muchas herramientas no son capaces de representarlos. Los patrones originales de flujo de control identifican cuatro de estos patrones: Multi-elección, Fusión sincronizada, Fusión múltiple y Discriminador.

Existen cinco patrones para representar paralelismo y sincronización avanzada, que son los siguientes.

### 6. *Multi-elección:*

*Descripción:* El patrón de elección múltiple es utilizado para modelar puntos en el flujo de proceso donde varios caminos son escogidos dependiendo de datos del proceso y/o puntos de decisión. Usa un punto de decisión incluido en modo división.

*Sinónimos:* Elección múltiple, selección múltiple.

*Característica:* Este patrón proporciona a un hilo de ejecución la capacidad de dividir en varios hilos concurrentes. La decisión en cuanto a si hay que pasar el hilo de ejecución a una rama u otra se hace en tiempo de ejecución y en la condición de la rama que va a separar. Puede estar basado en una variedad de factores incluyendo el resultado de una tarea precedente, los valores de los elementos de datos en un proceso, de los resultados de la evaluación de una expresión asociada con la rama saliente o algún otro mecanismo de selección.

*Contexto:* Hay una condición de contexto asociada a este modelo: el mecanismo que evalúa la multi-elección es capaz de tener acceso a cualquier elemento requerido o recursos necesarios, determinando cuál de las ramas salientes debe ser tomada por el hilo de control.

### 7. *Fusión de sincronización:*

*Descripción:* Es un punto en el proceso donde múltiples caminos que fueron activados antes en el proceso convergen en una sola rama, el proceso continua hasta que todas las actividades de las ramas entrantes hayan terminado. Se usa un punto de decisión incluido en modo unión.

*Sinónimo:* Unión sincronizada.

*Característica:* Este patrón proporciona el medio para activar algunas de las ramas salientes y a su vez esperar que todas las ramas activadas lleguen a un determinado punto para continuar con el proceso.

*Contexto:* No hay condición de contexto asociado pero se deben esperar a que todas las actividades finalicen para poder continuar.

## 8. *Multi-fusión:*

*Descripción:* El patrón de unión múltiple es usado para converger dos o más ramas en un solo camino. Cada activación de una de las ramas entrantes resulta en la activación de la siguiente actividad en el proceso

*Sinónimo:* Unión múltiple.

*Característica:* El patrón utiliza una compuerta paralela para habilitar los dos caminos. Cada vez que la actividad de uno de los caminos es terminada, la siguiente actividad, en este caso Informar sobre Referencias, es ejecutada.

*Contexto:* Hay una condición asociada a este patrón. Siempre estará precedido por la construcción del patrón multi-elección.

## 9. *Patrón discriminador*

*Descripción:* El discriminador es un punto del proceso que espera por una de las ramas entrantes para completar la condición, antes de continuar a la actividad siguiente. A diferencia del sincronizador, las ramas restantes que lleguen serán ignoradas y hace caso omiso de las mismas.

*Característica:* Con el objetivo de esperar por un número específico de ramas entrantes, una compuerta compleja (elemento de convergencia) es usada. Es necesario utilizar un código a la entrada de la compuerta compleja (Join) y una regla asociada a la transición de la misma que se dirige a la siguiente actividad. La implementación del discriminador también puede ser realizada a través de una compuerta exclusiva (elemento de convergencia) en la cual se evalúa la condición y en caso de no cumplirse se ignora a través del evento de fin.

## 10. *Patrón de unión N/ M*

*Descripción:* Este patrón puede ser visto como una generalización del discriminador. Se pretende sincronizar N caminos de M transiciones entrantes. Este patrón es un punto en el proceso en donde M ramas en paralelo convergen en una sola y la actividad siguiente es habilitada una vez que se han activado N ramas de entrada. Se ignora el resultado del resto de las M menos N ramas. Semejante al discriminador, la mezcla se desbloquea una vez que se han activado las M ramas de entrada.

*Característica:* Cuando N ramas se activan, se utiliza una compuerta compleja (elemento de convergencia) con el objeto de sincronizar el número de actividades requeridas. Por cada rama que llega a la compuerta compleja, el contador se incrementa. Cuando el contador alcanza el valor M, la actividad siguiente se activará. Después de esto, los caminos que lleguen serán ignorados.

## C) PATRONES ESTRUCTURALES

### 11. Patrón de ciclos arbitrarios

*Definición:* Consiste en puntos de decisión que hacen que se vuelva a una actividad anterior del proceso para que parte de él se repita.

*Característica:* El patrón utiliza una compuerta exclusiva que controla el número de veces que se ha aprobado la condición, cuando se cumpla la misma (número de aprobaciones requeridas) el proceso continúa por el flujo normal.

*Contexto:* Una o más actividades pueden realizarse repetidamente, sin restricción para el punto de entrada.

### 12. Patrón de terminación implícita

*Definición:* El patrón es utilizado para determinar cuándo la instancia de un proceso se considera completa. Un determinado proceso debe terminar cuando no hay tareas pendientes o cuando no hay posibilidad de crear nuevos trabajos en el futuro.

*Característica:* El patrón utiliza evento al final de los caminos que deben ser completados. El proceso se considera terminado cuando todos los eventos de fin son alcanzados.

*Contexto:* Si no hay otra actividad por ejecutar en cualquiera de las ramas, cuando se llega al evento de fin, el proceso es cerrado automáticamente.

## D) PATRONES DE CANCELACIÓN

Si el diseño necesita una estrategia para cancelar el proceso en algún punto durante la ejecución, una forma de resolver el problema puede ser añadir chequeos de cancelación en cada uno de los pasos.

### 13. Patrón de cancelación de actividad

*Definición:* Consiste en parar la ejecución de una particular actividad del proceso con un evento de cancelación. Es usado para abortar una ejecución larga, suspender una actividad o cambiar la ruta del proceso.

*Característica:* Este patrón puede ser modelado usando una compuerta basada en evento y un evento intermedio. La compuerta basada en evento habilita la actividad y el evento intermedio está disponible para tener la opción de cancelarla.

### 14. Patrón de cancelación del proceso

*Definición:* Consiste en parar la ejecución de un proceso entero con un evento de cancelación.

*Característica:* El patrón utiliza una compuerta paralela y un evento de fin terminal. Una vez que el proceso alcance el fin terminal, las actividades pendientes son canceladas y el caso termina.

*Contexto:* Cuando alguna transición llega a un evento de fin terminal, el proceso termina. Si se cuenta con un evento intermedio o actividad de cancelación que llegue posteriormente al evento de fin terminal, es posible ver este patrón.

### III.2.1.2 Aspectos Sintácticos de la Notación BPMN

En esta etapa se analiza la sintaxis abstracta del lenguaje BPMN, para lo cual se han identificado los principales objetos con sus atributos y las relaciones entre los mismos. Estos atributos son valores que corresponden a un objeto. Generalmente se conoce como la información detallada del objeto. [46]

a) **OBJETO DE FLUJO:** Son los principales elementos que definen el comportamiento de los procesos. Dentro de los objetos de flujo encontramos a las actividades, eventos y gateway, los cuales heredan los atributos de un objeto de flujo. Los atributos de un objeto de flujo son:

1. Identificador: Código único correspondiente al objeto de flujo en un DPN.
2. Tipo de objeto: Determina el tipo de objeto de flujo: *actividad*, *gateway* o *evento*.

a.1) *Actividad:* Sus atributos son:

1. Nombre: Corresponde a la descripción de la tarea o subproceso.
2. Tipo: Puede ser una *tarea* atómica o un *subproceso*.
3. Estado: Indica el estado del objeto actividad. El estado puede indicar: “*activada*”; “*en proceso*”; “*completada*”; “*cancelada*”.
4. Responsable: Se refiere al responsable de realizar la actividad. Puede referirse a una persona, departamento o área, sistema etc.

a.1.1) *Tarea:* Los atributos de una tarea son:

1. Cantidad de token: Se refiere a los recursos que llegan a la actividad para que esta se inicie.
2. Loop: Indica la cantidad de iteraciones sobre sí misma.
3. Tipo de loop: Indica el tipo de loop, es decir si es estándar, multi-instancia, o bien no tiene ningún bucle. El marcador de una tarea estándar es un circuito es una pequeña línea con una punta de flecha en la punta que se enrosca sobre sí misma. El marcador para una tarea de multi-instancia es un par de líneas verticales en paralelo.
4. Compensación: Indica si la tarea es de compensación, se establecen reglas para deshacer tareas si una falla. El marcador de una tarea de compensación es un par de triángulos enfrentados.

**a.1.2) Subproceso:** Los atributos de un subproceso son:

1. Clase se subproceso: Indica la clase de subproceso, si es de compensación o ad-hoc. Un ad-hoc se representa con un "tilde".
2. Índice: Indica número del diagrama de proceso correspondiente al subproceso expandido.
3. Tipo de loop: Indica el tipo de loop, es decir si es estándar o multi-instancia.

**a.2) Evento:** El atributo de un evento es:

Clase de evento: Indica si el evento es *inicial*, *intermedio* o *final*.

**a.2.1) Evento inicial:** Es un tipo de evento que crea una nueva instancia de un proceso. Tiene un único atributo, el cual es:

Tipo de evento: Indica el tipo de evento es: sin especificar o simple, mensaje, tiempo, regla, link, múltiple.

**a.2.2) Evento intermedio:** Es un acontecimiento que ocurre durante el curso del proceso indicando que algo ha ocurrido. Tiene un único atributo, el cual es:

Tipo de evento: Indica el tipo de evento es: sin especificar o simple, mensaje, tiempo, error, cancel, compensación, regla, link, múltiple.

**a.2.3) Evento final:** Indica que el proceso ha terminado y por tanto, cierran la instancia activa del mismo. Este tipo de eventos puede además disparar una acción al momento del cierre. Tiene un único atributo, el cual es:

Tipo de evento: Indica el tipo de evento es: general o vacío, mensaje, error, cancel, compensación, link, terminal, múltiple.

**a.3) Gateway:** Son mecanismos de bifurcación o unión de flujos. Sus atributos son:

**a.3.1) Tipo de gateway:** Indica el tipo, el que puede ser de datos o eventos.

**a.3.2) Tipo de control:** Pueden ser gateway de los siguientes tipos: *decisions/branching* (OR-Split; exclusive-XOR, inclusive-OR, complex); *merging* (OR-Join), *forking* (AND-Split), y *joining* (AND-Join).

**a.3.3) Cantidad de entradas:** Corresponde al número de secuencias de flujo que llegan al gateway. Este dato se actualiza durante la creación del modelo de proceso de negocio y está controlado por un método de la clase actividad.

**b) ARTEFACTOS:** Son elementos documentales que permiten agregar información adicional a los diagramas para hacer más entendible su lectura. Dentro de los artefactos encontramos a objeto de datos, agrupaciones y anotaciones textuales, los cuales heredan los atributos de un artefacto. Los atributos de artefacto son:

1. Tipo: Describe el tipo de artefacto (agrupamiento, objeto de datos, anotación textual).
2. Nombre: Se especifica la descripción del artefacto.

**b.1) Objeto de datos:** Sus atributos son:

1. Identificador: Código único del artefacto objeto datos.
2. Descripción: Se especifica la descripción del artefacto.

**b.2) Agrupación:** este artefacto permite agrupar visualmente varios elementos del diagrama. Sus atributos son:

1. Identificador: Código único del artefacto agrupamiento.
2. Nombre: Descripción del agrupamiento.

**b.3) Anotación textual:** Este artefacto permite colocar un texto en el diagrama. Sus atributos son:

1. Identificador: Código único de la anotación textual.
2. Descripción: Información adicional que se asocia a un objeto determinado.

**c) SWIMLANE:** Son mecanismos de organización de las actividades de un proceso. Dentro de esta categoría encontramos a los pool y a lanes.

**c.1) Pool:** Este swimlane se utiliza para delimitar el proceso. Sus elementos son:

1. Identificador: Código único correspondiente al pool.
2. Nombre: Denominación del pool.

**c.2) Lane:** Este swimlane se utiliza para delimitar la responsabilidad de un rol o grupo de usuarios sobre un conjunto de actividades. Sus atributos son:

1. Identificador: Código único de lane.
2. Pool: Código del pool en el que se incluye el elemento lane.

**d) OBJETO DE CONEXION:** Son elementos que permiten indicar el flujo del proceso y el orden en que se realizan las distintas actividades. Dentro de esta categoría encontramos *flujo de Secuencia, flujo de mensajes, asociaciones y grupos*.

**d.1) Relación de flujo de secuencia:** Indica el paso de la ejecución del proceso de una actividad (o evento) a otra. Sus atributos son:

1. Identificador de flujo: Código único correspondiente a la secuencia de flujo.
2. Nombre del flujo: Descripción textual del flujo.
3. Objeto de origen: Indica el objeto desde el cual parte el flujo de secuencia.

4. Objeto destino: Indica el objeto a hasta donde llega el flujo de secuencia.

**d.2) Relación de flujo de mensaje**: Indica que se envía un mensaje con información sobre las variables del proceso a otro proceso. Sus atributos son:

1. Identificador: Código correspondiente al flujo de mensaje.
2. Mensaje: Descripción textual del mensaje
3. Pool de entrada: Indica el pool desde donde sale el mensaje.
4. Pool de salida: Indica el pool hasta donde llega el mensaje.

**d.3) Flujo de asociación**: Este flujo se utiliza para conectar objetos de datos o artefactos con los elementos del proceso. Cuando conecta objetos de datos puede definir alguna regla de transformación en la información. Dentro de este tipo de flujo se encuentra la *Asociación Actividad - Objetos de datos* y la *asociación Objetos de datos –Actividad*.

**d.3.1) Asociación actividad - objetos de datos**

1. Identificador de la asociación de salida: Código correspondiente a la asociación de un objeto de flujo con un objeto de datos.
2. Artefacto: Indica el identificador del objeto de datos asociado.
3. Origen: Indica el objeto de origen de la asociación.

**d.3.2) Asociación objetos de datos - actividad**

1. Identificador de la asociación de entrada: Código correspondiente a la asociación de un objeto de flujo con un objeto de datos.
2. Artefacto: Indica el identificador del objeto de datos asociado.
3. Destino: Indica el objeto destino de la asociación.

**d.4) Asociación de información adicional**: Se utiliza para conectar las anotaciones textuales con los elementos del proceso.

1. Identificador: Código correspondiente a la asociación del objeto con una anotación textual.
2. Artefacto: Indica anotación textual asociado.
3. Origen: Indica el origen de la asociación.
4. Destino: Indica el destino de la asociación.

Por último, como resultado del análisis semántico se resume las consideraciones a tener en cuenta para crear un diagrama de proceso de negocio (DPN):

- En un DPN debe haber al menos un evento de inicio y otro de fin; los eventos de inicio no tienen entradas, pero sí una única salida, mientras que los eventos de fin tienen una sola entrada y ninguna salida.

- El evento de inicio solo puede aparecer sólo una vez por proceso, es decir dentro de un *pool*, aún cuando por múltiples razones se pueda dar inicio al proceso.
- Todos los flujos deben tener un comienzo, independientemente de si se refieren a un proceso o un subproceso.
- Dos o más transiciones pueden salir de una compuerta exclusiva como elemento de divergencia, y una debe ser un “*else*” o una línea “*default*”. El resto de transiciones salientes son transiciones de condición, es decir, tienen asociada una regla de negocio.
- Una compuerta exclusiva como elemento de convergencia es utilizada para la sincronización de caminos que salen de una compuerta basada en eventos.
- Toda actividad está en una ruta que va desde un evento inicio a un evento fin.
- El marcador *loop* no puede ser usado con un marcador de multi-instancia, cualquier otra combinación es posible.
- La compensación puede ser usada en combinación con el marcador de multi-instancia o Ad-hoc, a su vez el ad-hoc con un marcador multi-instancia.
- Todos los elementos deben pertenecer a un *pool*. Por lo tanto, todo proceso debe tener al menos un *pool*.
- Los objetos de flujo de *pool* diferentes se conectan con flujo de mensaje, mientras que los del mismo *pool* se conectan con flujo de secuencia.
- El número de *pool* requeridos y los nombres usados dependen de los estados del proceso.

### III.2.1.3 Relaciones entre los Elementos Notacionales

Un DPN representa una secuencia de flujos que indica el orden en que se realiza el proceso y las relaciones entre los objetos de flujo mediante la secuencia de flujos.

Las reglas para establecer las relaciones permitidas entre los objetos de flujos se presentan en la Tabla 3.2, y las reglas para la conexión de flujos de mensajes se presentan en la Tabla 3.3.

Por ejemplo, en la tabla 3.2 se muestra una secuencia de flujo puede salir desde una *tarea* hasta otra *tarea*, esta correspondencia permitida se representa con el símbolo “✓”; además desde un *evento final* no puede salir ninguna secuencia de flujo, las correspondencias no permitidas se representan con “X”.

Otro ejemplo, en la Tabla 3.3 se muestra que un flujo de mensajes puede salir desde un *pool* hasta otro *pool*. Los eventos no envían ni reciben mensajes.

En cuanto a la relación ‘asociación’ es posible asociar el artefacto de  *anotación textual* con un objeto de flujo  *actividad*. Los objetos de datos se pueden asociar a las actividades o a una secuencia de flujo.

Tabla 3.2. Reglas de conexión de secuencias de flujos

Correspondencia entre objetos de flujo mediante secuencia de flujo							
Hasta / Desde	Tarea	Subproceso	Evento Inicial	Evento Intermedio	Gateway	Evento final	Inicial
Tarea	✓	✓	✓	✓	✓	✓	X
Subproceso	✓	✓	✓	✓	✓	✓	X
Evento Inicial	✓	✓	X	✓	✓	✓	X
Evento Intermedio	✓	✓	X	✓	✓	✓	X
Evento final	X	X	X	X	X	X	X
Gateway	✓	✓	X	✓	✓	✓	X

Tabla 3.3. Reglas de conexión de secuencias de mensajes

Correspondencia de elementos mediante flujos de mensajes							
Hasta / Desde	Tarea	Subproceso	Evento Inicial	Evento Intermedio	Pool	Evento final	Inicial
Tarea	✓	✓	X	X	✓	X	X
Subproceso	✓	✓	✓	✓	✓	X	X
Evento Inicial	X	X	X	X	X	X	X
Evento Intermedio	X	X	X	X	X	X	X
Pool	✓	✓	✓	✓	✓	X	X
Evento final	✓	✓	✓	✓	✓	X	X

A continuación en la Figura 3.3 se muestra gráficamente y en forma sintetizada las relaciones entre los elementos de un DPN, donde se pueden ver las cuatro categorías de BPMN presentadas en el apartado III.2.1 de este capítulo.

Se observa que la categoría *objeto de conexión*, posee un conjunto de conectores representados por las *asociaciones*, *flujos de mensaje* y los *flujos de secuencia*, los cuales representan la conexión entre los elementos de un DPN. El *flujo de secuencia* representa el orden en que las actividades se ejecutan. El *flujo de mensaje* relaciona los pools y los objetos de datos contenidos en diferentes pools. La *asociación* conecta los artefactos con los objetos de flujo asociando información a los mismos.

En la categoría *objeto de flujo* se encuentran las actividades, eventos y gateway. Los eventos activan o convocan actividades así como los *Gateway*, a su vez los *Gateway* controlan la convergencia o divergencia de los flujos de secuencia.

Los elementos de la categoría *artefactos* permiten agregar información adicional a los objetos de flujo. Dentro de los artefactos encontramos a objeto de datos, agrupaciones y anotaciones textuales.

En la categoría *swinlanes* se encuentran *pools* y *lanes*, los cuales organizan y categorizan el DPN. Los pools pueden contener *objetos de flujos*, *artefactos*, *lanes*, *flujos de secuencia* y *asociaciones*, pero solo pueden relacionarse por medio de los *flujos de mensajes*.

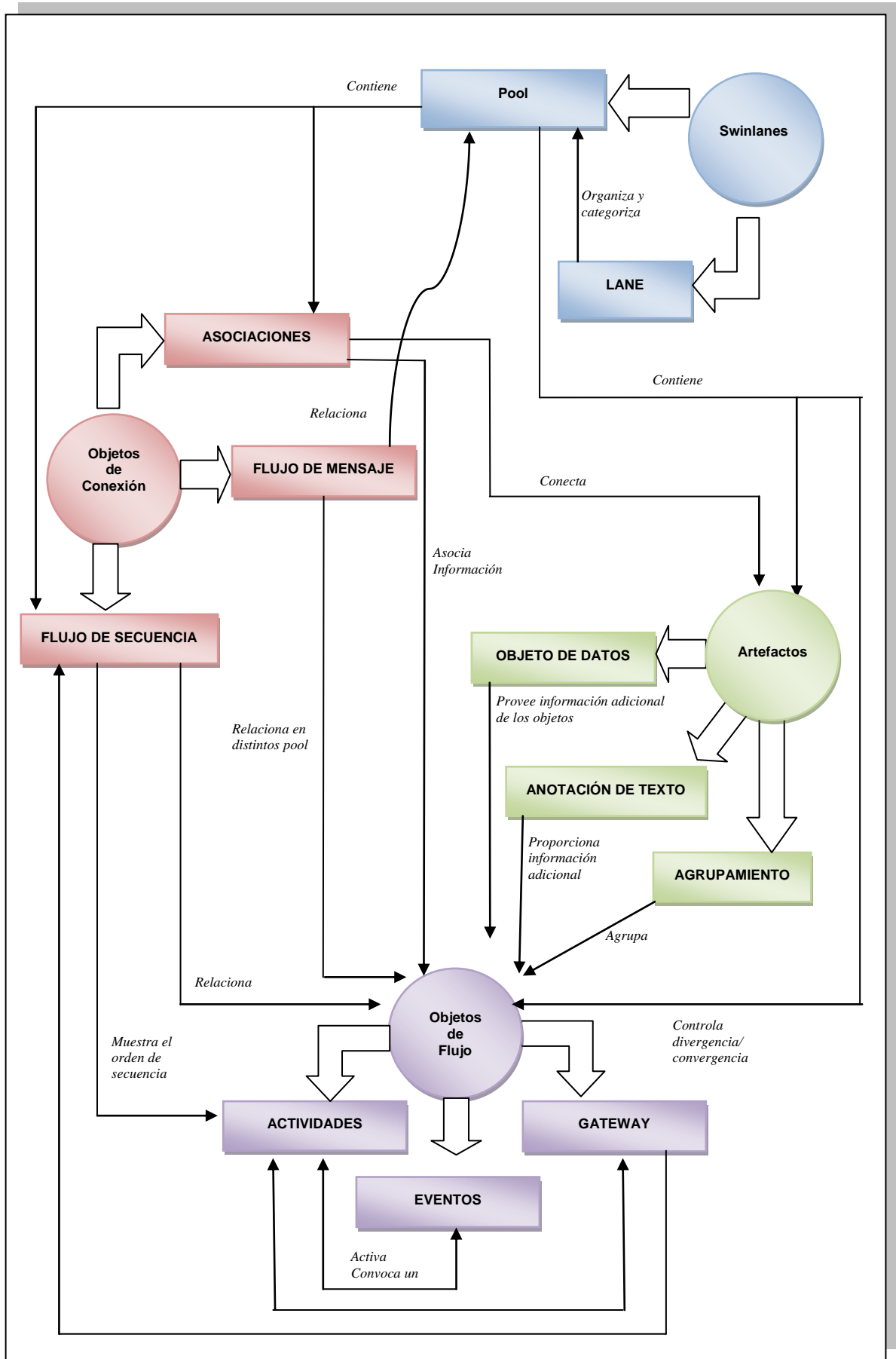





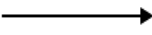

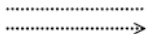



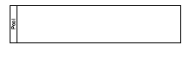
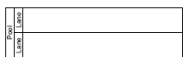
Figura.3.3. Relaciones entre los elementos básicos de BPMN

### III.2.1.4 Asociación entre Elementos Notacionales y Símbolos Gráficos que lo Representan

En esta etapa se describen los objetos gráficos que componen la notación, así como también la representación gráfica de los patrones de BPMN como parte de un diagrama de procesos de negocio, los cuales fueron presentados en el apartado III.2.1.1.







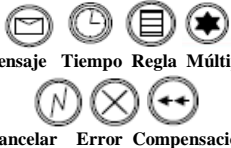
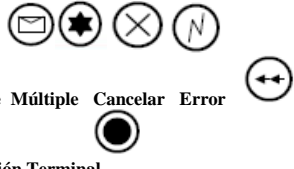





En la Tabla 3.4 se muestran las cuatro categorías de elementos de BPMN las cuales están separadas en columnas y en cada una se incluyen los elementos gráficos que las representan.

Tabla. 3.4. Elementos básicos de BPMN

1. Objetos de flujo	2. Objetos de conexión	3. Artefactos	4. Swimlanes
 <p>Inicial Intermedio Final</p> <p><b>1.1 Eventos</b></p>  <p>Tarea Proceso</p> <p><b>1.2 Actividades</b></p>  <p><b>1.3 Gateway</b></p>	 <p>2.1 Flujo de secuencia</p>  <p>2.2 Flujo de mensaje</p>  <p>2.3 Asociación</p>	 <p>3.1 Objeto de dato</p>  <p>3.2 Grupo</p>  <p>3.3 Anotación textual</p>	 <p>4.1 Pool</p>  <p>4.2 Lane</p>

Los elementos de un DPN pueden contener “*marcadores internos*”, lo que da lugar a variaciones en la representación gráfica mostrados en la tabla 3.5.

Tabla. 3.5. Variaciones de los Objetos de Flujo de BPMN.

2. Actividades	1. Eventos	3. Gateway
 <p>2.1 Ad Hoc</p>  <p>2.2 Compensación</p>  <p>2.3 Multi-instancia</p>  <p>2.4 Loop</p>  <p>2.5 Subproceso</p>	 <p>Mensaje Tiempo Regla Múltiple</p> <p>1.1 Eventos iniciales</p>  <p>Mensaje Tiempo Regla Múltiple</p> <p>Cancelar Error Compensación</p> <p>1.2 Eventos intermedios</p>  <p>Mensaje Múltiple Cancelar Error</p> <p>Compensación Terminal</p> <p>1.3 Eventos finales</p>	 <p>3.1 Exclusivos Basado en datos(XOR)</p>  <p>3.2 Exclusivos basado en eventos (OR)</p>  <p>3.3 Paralelo(AND)</p>  <p>3.4 Inclusivos(OR)</p>  <p>3.5 Complejo(COMPLEX)</p>

La representación gráfica de los patrones de control de flujo de BPMN, en los cuales se tratan los esquemas de divergencia, sincronización y cancelación en la secuencia de actividades en un proceso, se presentan en el **Anexo A**.

### III.2.2 Definición Conceptual del Metamodelo de BPMN.

A partir del análisis realizado del lenguaje de notación BPMN, sintáctico, semántico y las relaciones que se pueden establecer entre los elementos y los objetos gráficos que lo representan, se construye el metamodelo de BPMN. [8]

Un metamodelo define la sintaxis abstracta de un lenguaje estableciendo los conceptos y sus relaciones, e incluye las reglas que determinan si el modelo está correctamente construido.

El metamodelo resultante está representado por un diagrama de clases de UML (Figura 3.4). El diagrama de clases incluye los elementos principales de BPMN, se representan mediante una clase donde se identifican sus nombres y atributos conceptuales.

El diagrama está representado por la clase DPN, y a esta clase pertenecen todos los elementos de un diagrama de proceso de negocio: **entidades** (clases *ObjFlujo*, *Artefacto*, *Pool*, *Actividad*, *Evento*, etc.) y **relaciones** (asociaciones *FlujoMens*, *FlujoSec*, *AsociacionAT* y *AsociacionOD*).

El principal elemento de un proceso es la actividad, definida en el metamodelo como clase **Actividad**. En el mismo nivel se encuentran las clases **Gateway** y **Evento**, que heredan los atributos de la superclase objeto de flujo llamado **ObjFlujo**. Los objetos de flujo representados por las clases *Actividad*, *Gateway* y *Evento* se relacionan mediante un flujo de secuencia con la que determina el orden de ejecución del proceso. Para los flujos de secuencia se ha especificado la clase de nombre **FlujoSec**, en la que se definen atributos de la relación: identificador, el objeto origen desde el que parte el flujo y el objeto destino hasta donde llega el flujo, además el nombre del flujo que se corresponde con una descripción textual. De igual forma se define la clase **FlujoMens** para establecer el envío de mensajes entre diferentes pools, con los atributos referidos al mensaje correspondiente.

Para definir las asociaciones entre artefactos y objetos de flujo, se crean dos clases: **AsociacionAT** (define la asociación entre las clases *AnotacionT* y *ObjFlujo*), y **AsociacionOD** (relaciona las clases *ObjetoDato* y *Actividad*).

Las relaciones de herencia que se especifican se definen como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre. Por ejemplo: Una tarea y un subproceso heredan los atributos de la clase **Actividad**; estas dos clases comparten datos básicos, pero además cada uno tendrá información adicional que depende del tipo de clase.

El metamodelo se completa con la relación “*Proceso de negocio (PN)*, se representa con un *DPN*”. A su vez se considera que un PN tiene metas de negocio. Para cada clase *Meta* existe una o más *Reglas* que se deben cumplir para alcanzar una meta de negocio.

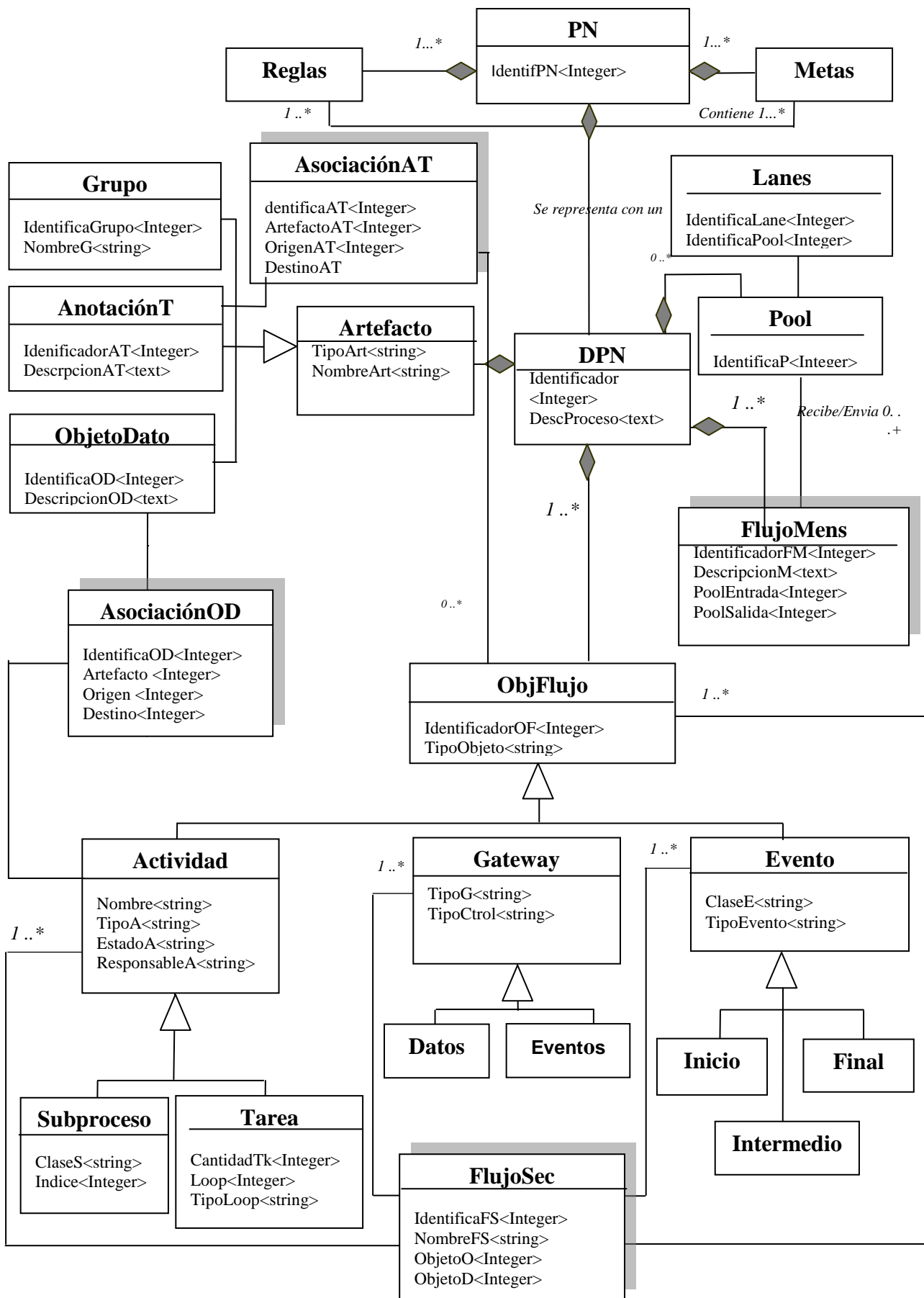


Figura 3.4. Diagrama de clases del Metamodelo BPMN

### III.2.3 Implementación del Metamodelo BPMN

En esta etapa se implementa el metamodelo definido en el apartado III.2.2, en la herramienta AToM (A Tool for Multi-Formalism Modelling and Meta-Modelling) [4] para lo cual el diagrama de clase es “cargado” en la herramienta. En la figura 3.5 se presenta el diagrama de clase en la interfaz de AToM<sup>3</sup>.

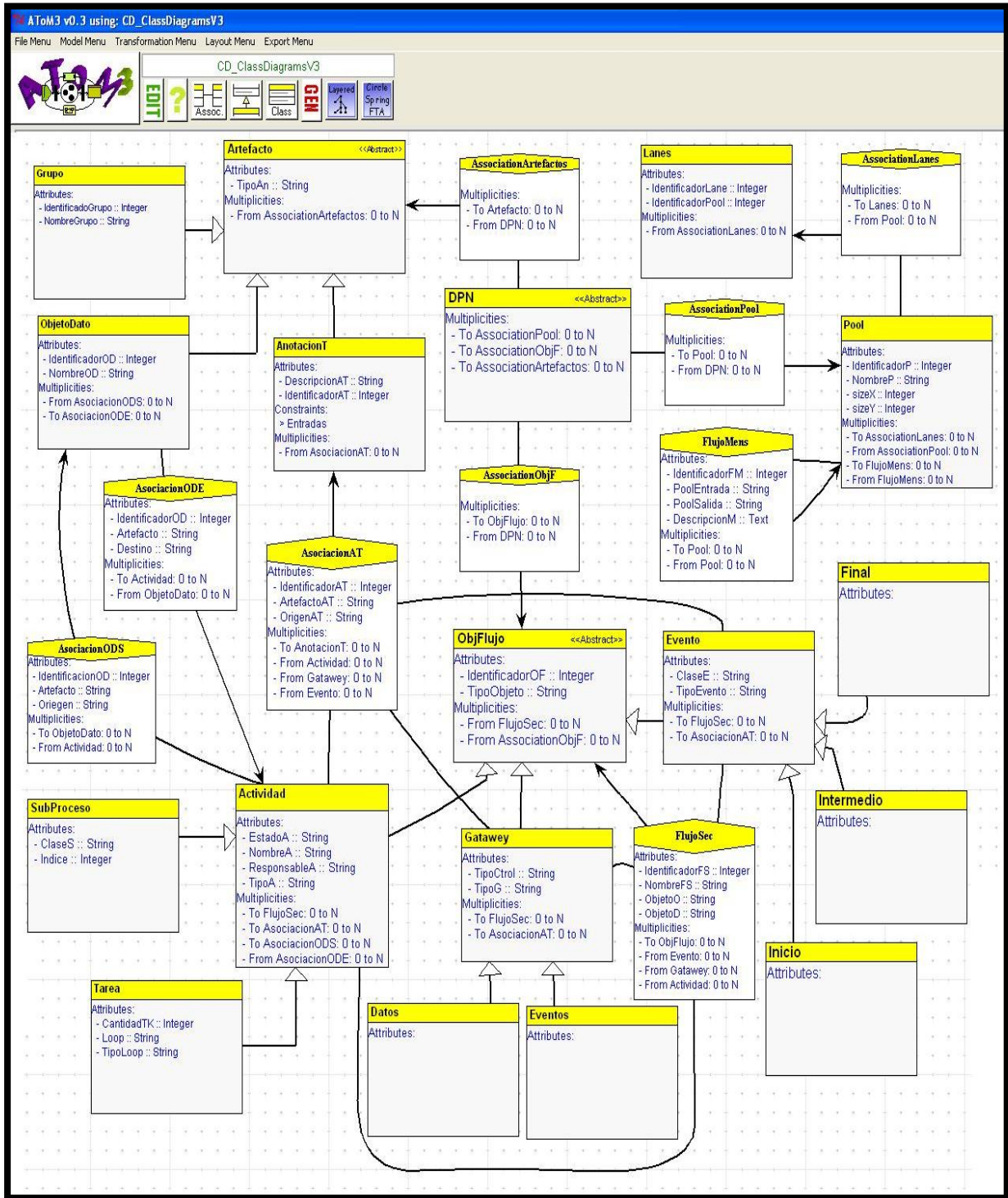


Figura 3.5. Diagrama de clases del Metamodelo BPMN implementado en AToM<sup>3</sup>

Para completar la implementación se define la *sintaxis concreta* que representará visualmente la *sintaxis abstracta* en ATOM<sup>3</sup>. Para completar esta etapa se asignan los iconos a las entidades y se dibuja la apariencia grafica de las asociaciones. En este caso a cada elemento de la sintaxis concreta se asigna el símbolo gráfico definidos por BPMN [8], tal como se presenta en el apartado III.2.1.4. Se personaliza la interfaz de ATOM<sup>3</sup>, se definen los botones, se asignan los iconos a cada elemento de la notación y se especifican las acciones y restricciones dadas por la notación.

Finalmente se obtiene la herramienta ATOM<sup>3</sup> actualizada que permite diagramar procesos de negocio basados en el estándar BPMN (en el **Anexo B** se ilustran pantallas con cada símbolo gráfico y sus atributos correspondientes).

### III.2.4 Verificación del Metamodelo en ATOM<sup>3</sup>

La verificación del metamodelo BPMN resultante se obtiene de las relaciones permitidas entre los elementos de la notación presentada en las Tablas 3.2 y 3.3 del capítulo III.

A continuación se simula la relación entre dos tareas dicha asociación resultan validos, como se puede ver en la Figura 3.6. Luego se intenta asociar una “tarea” con un evento “inicial”, inmediatamente la herramienta emite un mensaje “Flujo no Valido” debido a que no se puede asociar dichos elementos, esto se muestra en la Figura 3.7. El resto de estos elementos se muestran en el **Anexo C**.

#### V1: Tarea – Tarea

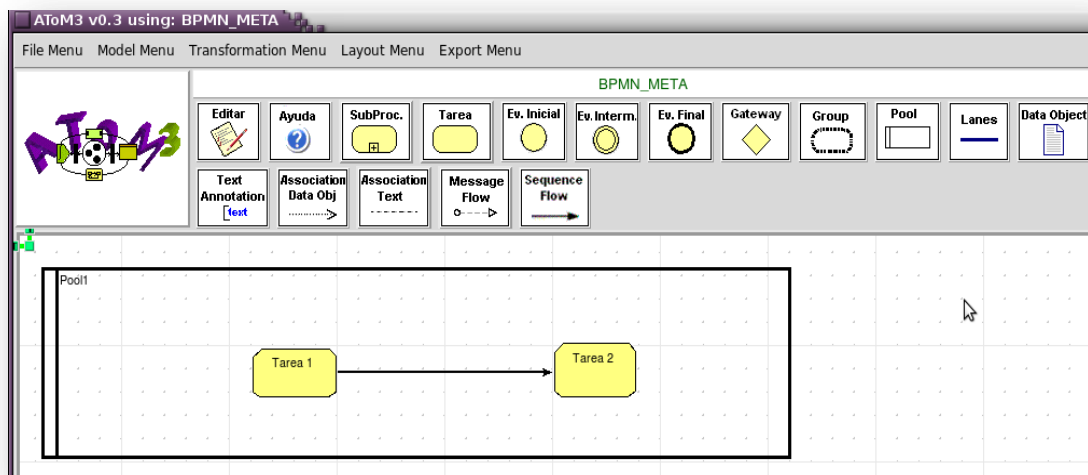


Figura 3.6. Conexión entre Dos Tareas

### V3: Tarea – Evento Inicial

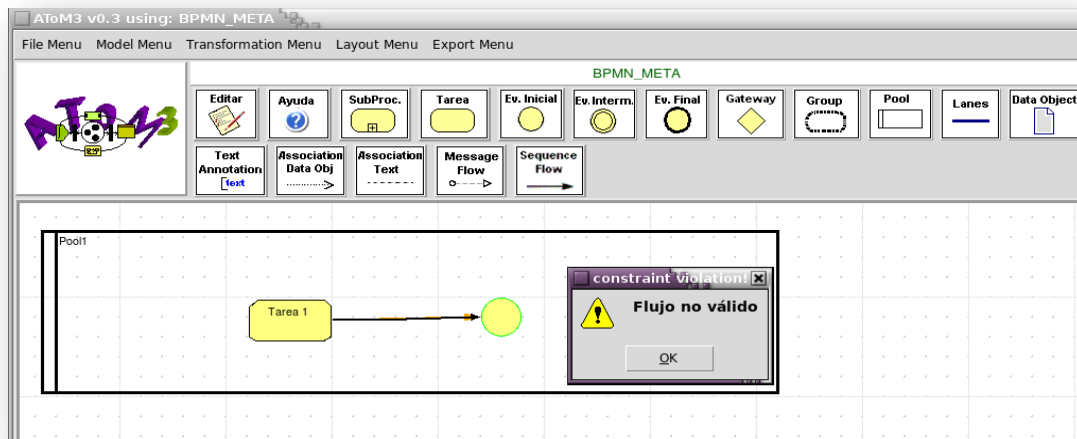


Figura 3.7. Conexión entre una Tarea y un Evento Inicial

### III.3 MODELADO DEL PROCESO DE SOFTWARE CON BPMN

El modelo de Proceso de Software considerado en este trabajo corresponde a una adaptación del proceso de IEEE [18]. El modelo adaptado agrupa las actividades del proceso de software en tres categorías:

- *Grupo de Actividades de Gestión de Proyectos:* Entre estas actividades se encuentran la estimación de tiempos y recursos, planificación y seguimiento y control del proyecto.
- *Grupo de Actividades Orientadas al Desarrollo:* Agrupan las actividades técnicas relacionadas con el desarrollo de software tales como identificación de necesidades, definición de requisitos, diseño, implementación e instalación.
- *Grupo de Actividades de Soporte:* Abarca las actividades que se aplican para complementar tanto las actividades de gestión como las actividades técnicas, entre las que se incluyen evaluación, gestión de configuración, o documentación.

En la Tabla 3.6 se describen las actividades del *Proceso de Desarrollo de Software*, y las actividades correspondientes al proceso de Gestión, Desarrollo y Soporte del Software.

En la Figura 3.8 se presenta el modelo del *proceso de desarrollo de software* orientado al negocio, en un nivel de detalle global. Las actividades del proceso de gestión del proyecto, del desarrollo y de Soporte se rotulan como  $G_i$ ,  $D_i$  y  $S_i$  respectivamente. Para poder ser verificado se lo implementa en ATOM<sup>3</sup>, con la notación BPMN.

En la Figura 3.9 se muestra el proceso de desarrollo del software modelado con BPMN en la interfaz de ATOM<sup>3</sup>.

<b>Actividades de Gestión (Gi)</b>	<b>Actividades de Soporte (Si)</b>	<b>Actividades de Desarrollo (Di)</b>
G <sub>1</sub> : Elaborar mapa de actividades según MCVS	S <sub>1</sub> : Elaborar plan de verificación y validación	D <sub>1</sub> : Identificación de necesidades del proyecto
G <sub>2</sub> : Definir entorno del proyecto	S <sub>2</sub> : Identificar configuración del sistema	D <sub>2</sub> : Especificar requisitos funcionales
G <sub>3</sub> : Estimar proyecto	S <sub>3</sub> : Planificar gestión de configuración del software	D <sub>3</sub> : Realizar diseño preliminar
G <sub>4</sub> : Elaborar plan de gestión del proyecto de software (PGPS)	S <sub>4</sub> : Verificar planes	D <sub>4</sub> : Diseñar base de datos
G <sub>5</sub> : Elaborar plan de garantía de calidad	S <sub>5</sub> : Corregir inconsistencias	D <sub>5</sub> : Diseñar interfaces
G <sub>6</sub> : Planificar riesgos y contingencias	S <sub>6</sub> : Guardar cambios	D <sub>6</sub> : Realizar diseño detallado
G <sub>7</sub> : Gestionar proyecto	S <sub>7</sub> : Verificar documento PGPS	D <sub>7</sub> : Crear código
G <sub>8</sub> : Guardar documentos	S <sub>8</sub> : Corregir inconsistencias del PGPS	D <sub>8</sub> : Planificar integración
G <sub>9</sub> : Gestionar calidad del proyecto.	S <sub>9</sub> : Revisar diseño	D <sub>9</sub> : Crear datos de prueba
	S <sub>10</sub> : Corregir diseño	D <sub>10</sub> : Integrar módulos
	S <sub>11</sub> : Verificar planes	D <sub>11</sub> : Instalar sistema
	S <sub>12</sub> : Corregir inconsistencias	D <sub>12</sub> : Guardar informes finales.
	S <sub>13</sub> : Guardar cambios	
	S <sub>14</sub> : Recibir Código	
	S <sub>15</sub> : Verificar Código	
	S <sub>16</sub> : Corregir errores código	
	S <sub>17</sub> : Guardar cambios.	

*Tabla 3.6. Descripción de las actividades del modelo de Proceso de Desarrollo de Software*



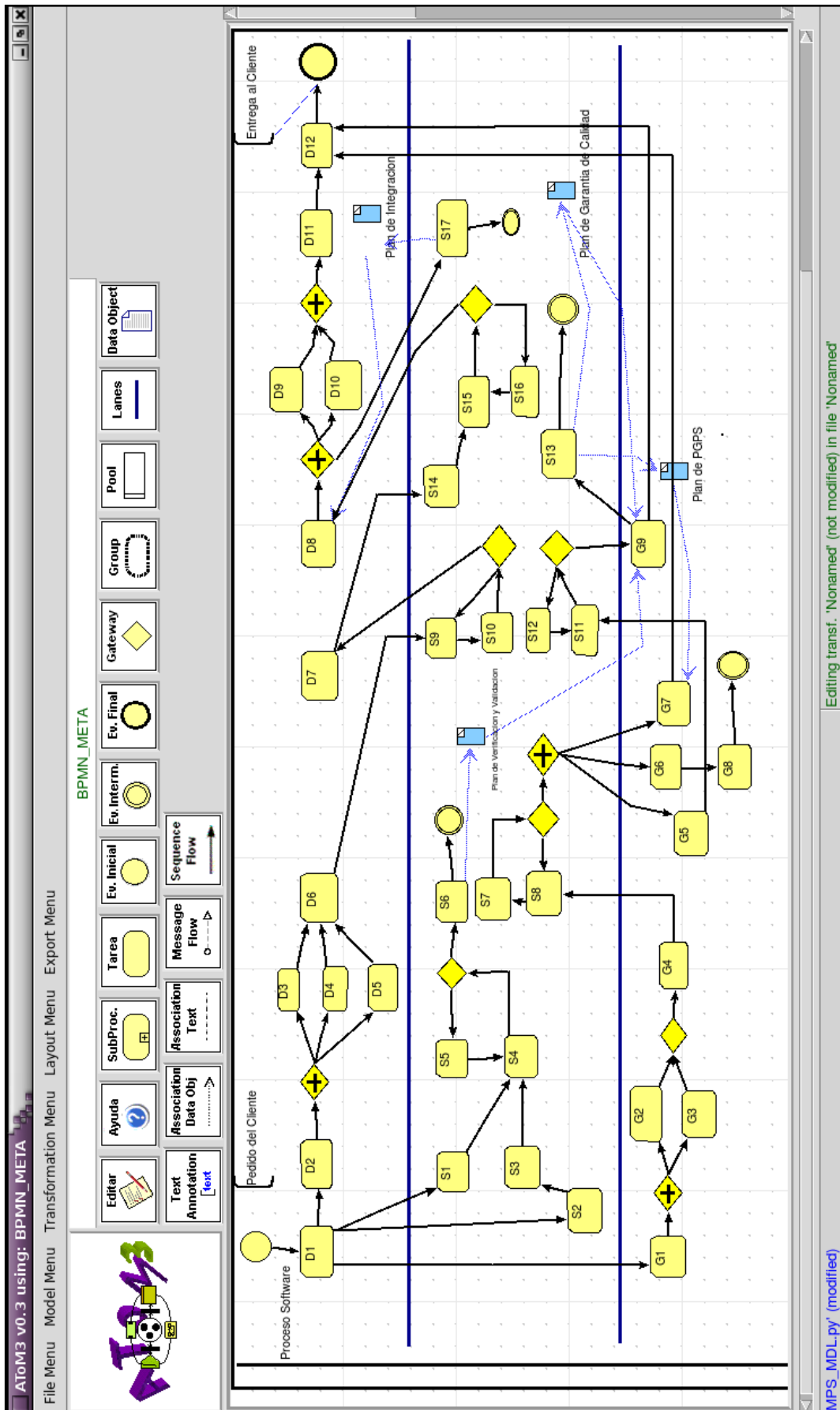


Figura 3.9. Proceso de Software modelado con BPMN en ATOM<sup>3</sup>

### III.3.1 Comprobación del Modelo de Software Orientado al Proceso de Negocio

AToM<sup>3</sup> posibilita la simulación de los modelos representados con BPMN usando gramáticas de grafos, lo cual permite la comprobación de los modelos orientados al negocio resultante.

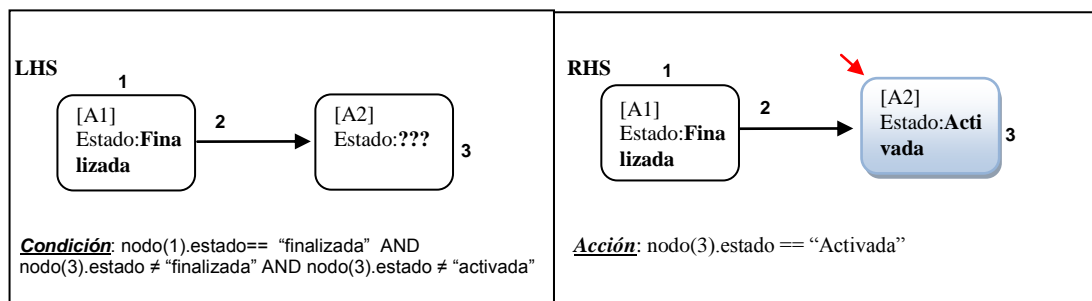
Las gramáticas de grafos se componen de una serie de reglas. Cada regla define en su parte izquierda (LHS, Left-handside) un subgrafo patrón y en su parte derecha (RHS, Right-handside) un subgrafo sustitución. Cuando se aplica una regla a un grafo de entrada, las ocurrencias de los patrones de las LHS de las reglas son sustituidas por las RHS. De este modo, el grafo se va transformando. Las reglas pueden contener también una condición, que debe ser satisfecha para que se pueda aplicar la regla, así como acciones que se ejecutan si la regla es aplicada. Los sistemas de reescritura de grafos aplican iterativamente las reglas de la gramática al grafo, hasta que ninguna regla sea aplicable.

#### III.3.1.1 Definición de Reglas Gramaticales.

La aplicación de gramáticas de grafo y su semántica operacional permite comprobar la correctitud semántica y sintáctica del modelo del Proceso de Software orientado al Negocio obtenido. Para ello se requiere definir las reglas semánticas operacionales de acuerdo a los patrones de comportamiento en el que se basa BPMN. Se usa una representación gráfica similar a la utilizada en [41].

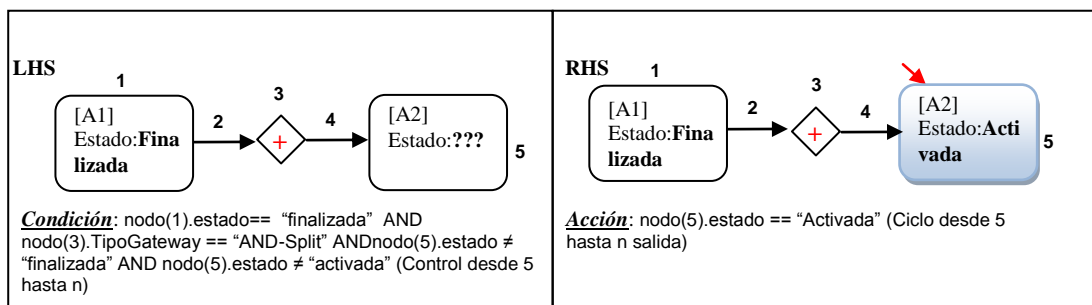
A continuación se presentan algunas de las reglas que componen la gramática para la simulación del comportamiento de los patrones definidos (ver descripciones de cada patrón en el apartado III.2.1.1).

**Regla 1. Generación de secuencia de actividades:** define el avance secuencial en la ejecución de las actividades. Esta regla se aplica si se encuentra el patrón de la parte izquierda en el modelo. Es decir si hay dos actividades conectadas por un flujo de secuencia. En este caso se reemplaza en el modelo por la parte derecha de la regla que indica el avance de la ejecución a la actividad siguiente. Se actualiza el atributo estado con el valor “activada”. La condición de aplicabilidad especifica que el atributo estado de la actividad del nodo 1 debe estar “finalizado”. Además el estado de la actividad del nodo 3 no debe ser “finalizado” ni “activada”. Se realiza el control iterativamente para cada una de las entradas de la actividad (nodo 3).

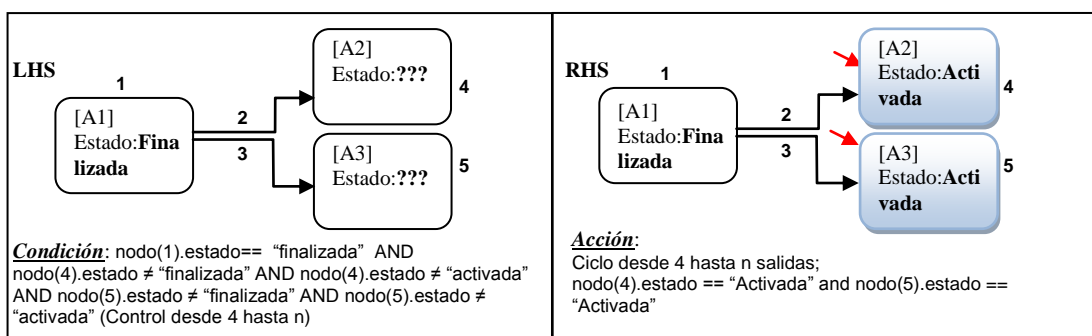


**Regla 2. Generación de división de actividades paralelas:** la regla define la realización de actividades en forma concurrente; donde un camino único en el proceso es particionado en dos o más actividades que pueden comenzar al mismo tiempo y realizarse concurrentemente.

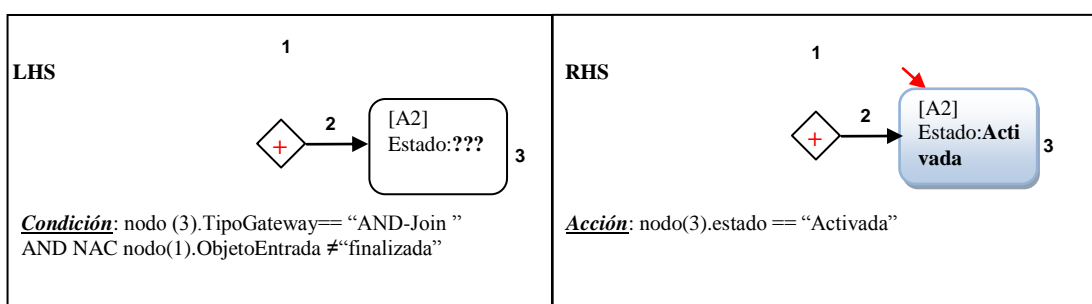
En la LHS se define la estructura de control donde se verifica la terminación de las actividades, y el tipo de gateway. Si este patrón coincide en el modelo, se indica el inicio a las actividades, que se realizan paralelamente. La regla se aplica tantas veces como sea posible y como resultado se pondrá “activada” en todas las salidas.



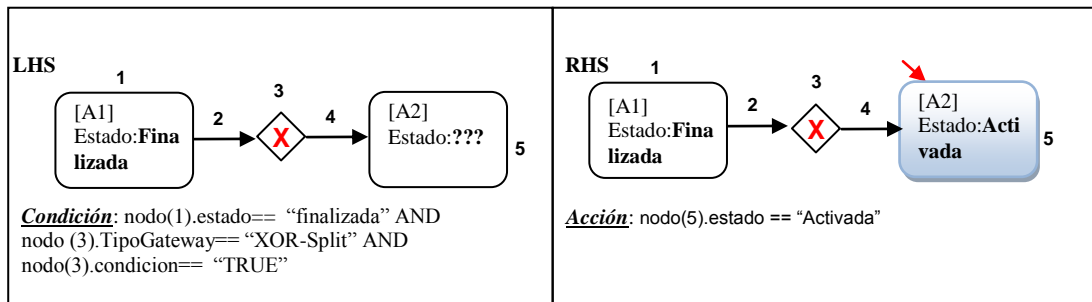
**Regla 3. Generación de partida paralela:** la regla define otro modo de realización de actividades en forma concurrente; donde un camino único en el proceso es particionado en dos o más caminos, de modo que dos o más actividades puedan comenzar al mismo tiempo y realizarse concurrentemente. Este paralelismo está modelado implícitamente. Los controles de aplicabilidad son similares a la regla 2.



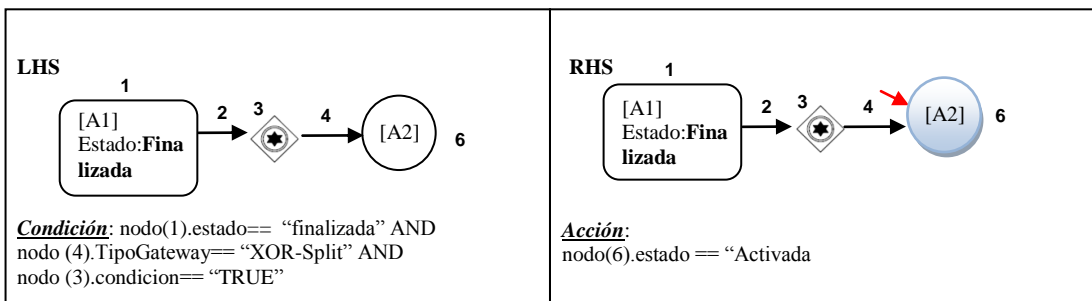
**Regla 4. Sincronización de actividades:** BPMN implementa un gateway que controla la culminación de los flujos de entrada. Cuando todas las actividades se completan, el flujo continúa en un único proceso. En la regla se añade una condición negativa de aplicación (NAC) que compruebe que no exista ninguna entrada no finalizada. Es decir, la regla no se aplica si alguna entrada al gateway no está finalizada.



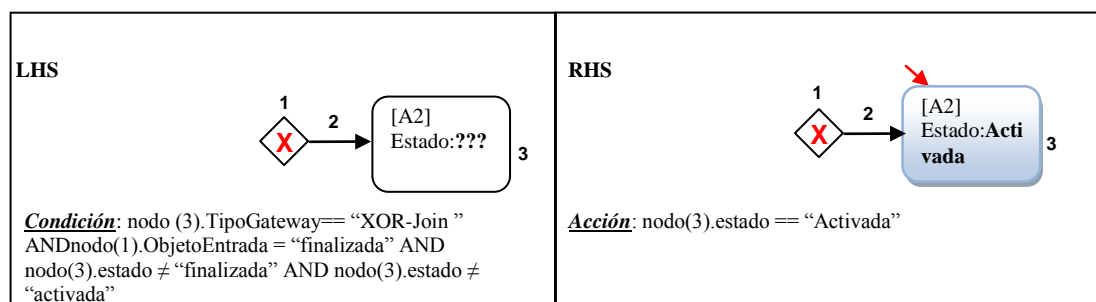
**Regla 5. Selección exclusiva de datos:** Ocurre cuando en un punto del flujo de trabajo se escoge sólo una de las varias ramas del proceso, generalmente esta decisión se toma basándose en datos de control del flujo de proceso. Las transiciones de salida tendrán asociada una regla de negocio booleana que será evaluada para determinar cuál secuencia del flujo deberá ser seleccionado para continuar al siguiente paso. Se debe asociar además una condición “default” o “else” en caso de no ser válido el otro u otros caminos. La regla se aplica cuando por lo menos una actividad se haya terminando y se cumple la condición. Como resultado se pondrá “activada” en una sola de sus salidas.



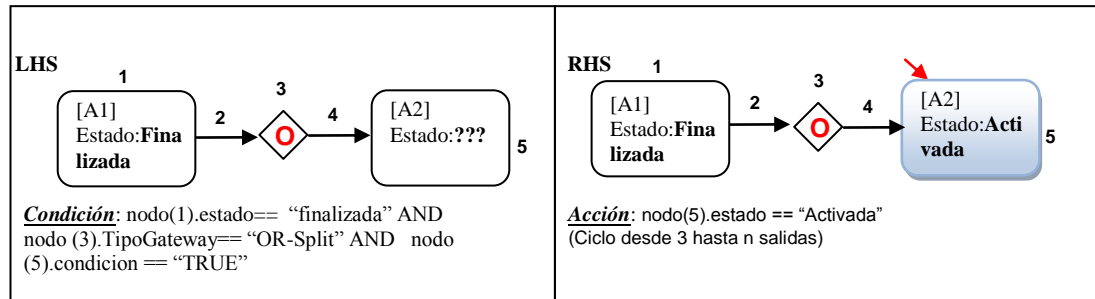
**Regla 6. Selección exclusiva de eventos:** Ocurre cuando en un punto del flujo de trabajo se escoge sólo una de las varias ramas del proceso, generalmente esta decisión se toma basándose en datos de control del flujo de proceso. Las transiciones de salida tendrán asociada una regla de negocio booleana que será evaluada para determinar cuál secuencia del flujo deberá ser seleccionado para continuar al siguiente paso. Se debe asociar además una condición “default” o “else” en caso de no ser válido el otro u otros caminos. La regla se aplica cuando por lo menos una actividad se haya terminando y se cumple la condición. Como resultado se pondrá “activada” en una sola de sus salidas.



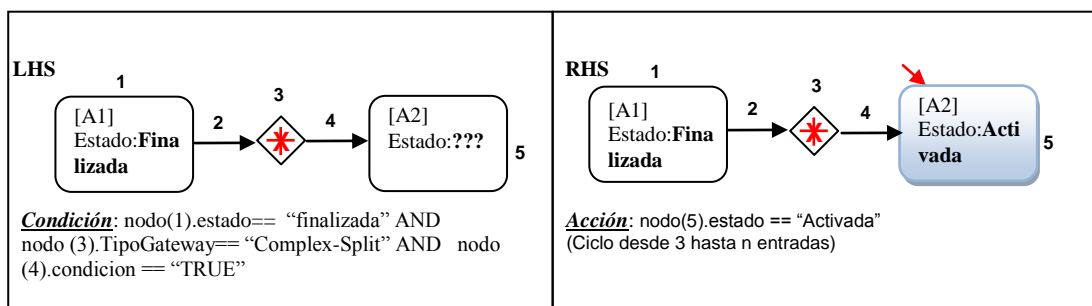
**Regla 7. Fusión simple:** La unión simple es un punto del proceso donde dos o más caminos alternativos convergen en uno solo sin necesitar de la sincronización para continuar al siguiente paso. En la LHS se define la estructura de control donde se verifica la terminación de las actividades, y el tipo de gateway. Si este patrón coincide en el modelo, se indica el inicio a una actividad. La regla se aplica tantas veces como sea posible y como resultado se pondrá “activada” a la salida y nunca puede contener más de un token.



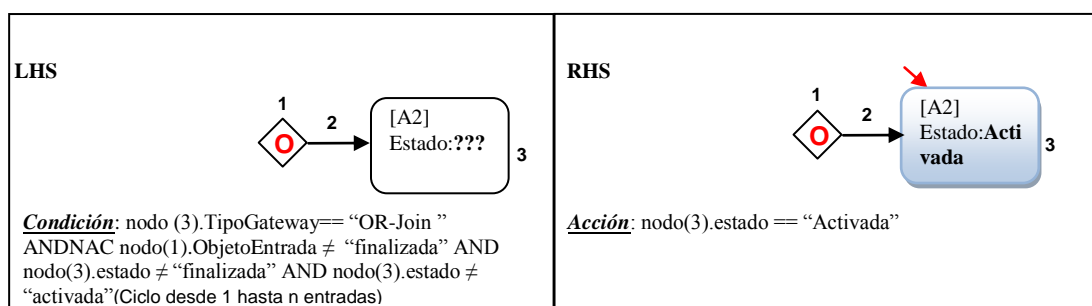
**Regla 8. Selección múltiple:** Se encuentra en un punto del flujo de trabajo en donde, basándose en una decisión o en los datos de control del mismo, se activa uno o más caminos del proceso. En la LHS se define la estructura de control donde se verifica la terminación de las actividades, si es un gateway del tipo *OR-Split* y si se cumplen la condición definida en el gateway. Si este patrón coincide en el modelo, se indica el inicio a las actividades. Como resultado se pondrá “activada” en una o más salidas.



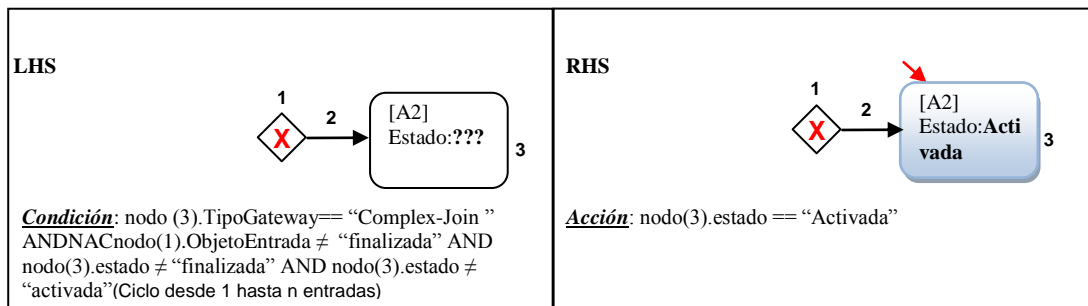
**Regla 9. Selección compleja:** Esta regla define otro modo de selección múltiple; donde uno o varios caminos pueden ser elegidos. Los controles de aplicabilidad son similares a la regla 7, pero con él un gateway del tipo *Complex-Split*.



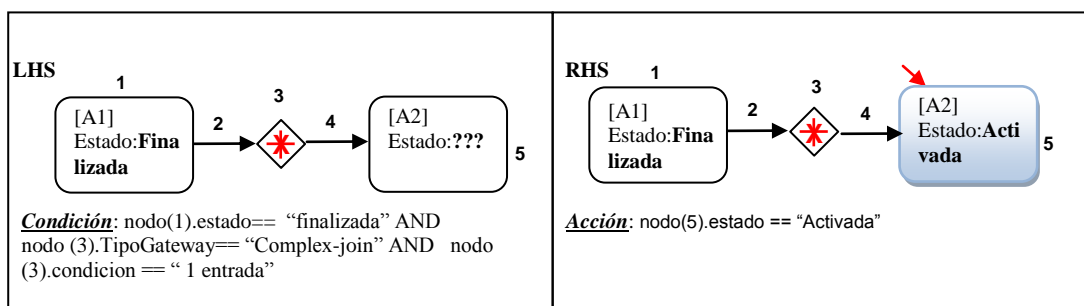
**Regla 10. Fusión de sincronización:** Es un punto del proceso en donde múltiples caminos convergen en uno solo. Cuando todas las actividades se completan, el flujo continúa en una única actividad. En la regla se añade una condición negativa de aplicación (NAC) que compruebe que no exista ninguna entrada no finalizada. Es decir, la regla no se aplica si alguna entrada al gateway *OR-Join* no está finalizada.



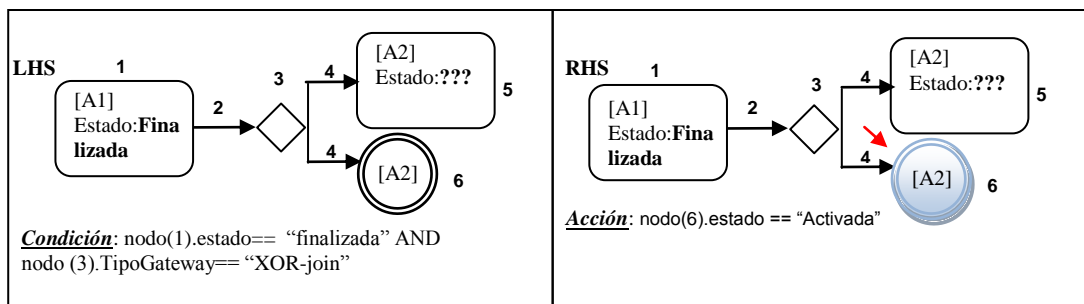
**Regla 11. Fusión de sincronización compleja:** Esta regla define otro modo de fusión sincronizada; donde se espera que todas las actividades entrantes terminen para continuar solo una actividad saliente de un gateway *Complex-join*. Los controles de aplicabilidad son similares a la regla 9.



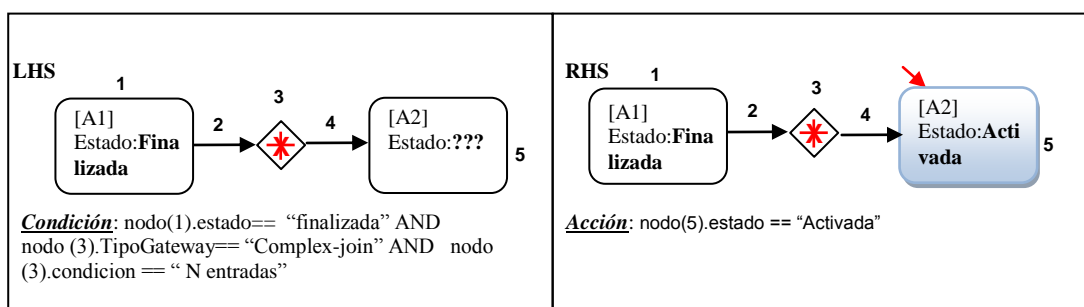
**Regla 12. Discriminador:** Es un punto del proceso que espera por una de las ramas entrantes para completar la condición, antes de continuar a la actividad siguiente. A diferencia del sincronizador, las ramas restantes que lleguen serán ignoradas y hace caso omiso de las mismas. En el gateway *Complex-join* hay una condición que indica el número de actividades requeridas para activar la actividad saliente.



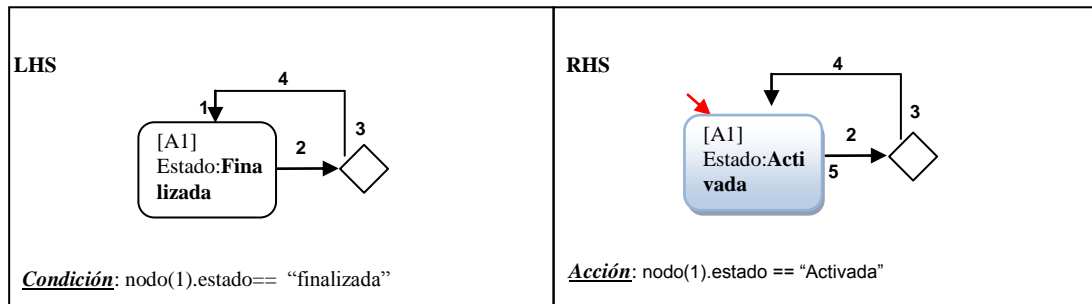
**Regla 13. Discriminador con exclusión:** Es otro caso de discriminador, solo que puede ser realizada a través de una compuerta exclusiva de convergencia en la cual se evalúa la condición y en caso de no cumplirse se ignora a través del evento de fin.



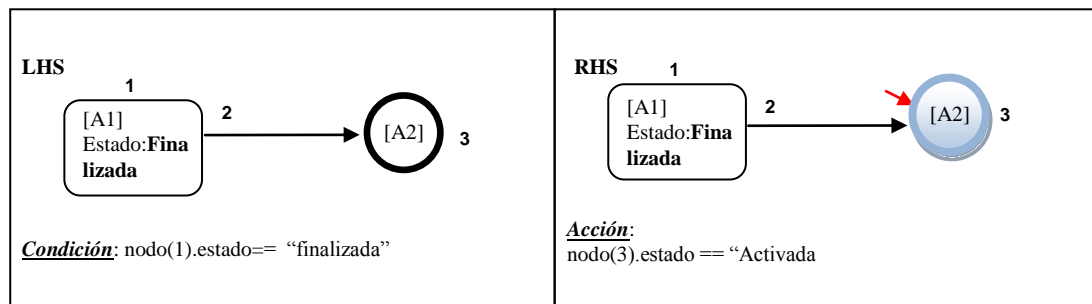
**Regla 14. Generación de unión N/ M:** El patrón modela la convergencia de M ramas que han sido habilitadas en un punto anterior del proceso en un solo camino. El proceso continua cuando N de las M ramas han sido completadas



**Regla 15. Generación de ciclo arbitrario,** como mecanismo que controla la repetición de subprocesos dentro de un loop. El bloque puede ser estructurado o no estructurado. Es decir, el punto donde recicla el proceso puede permitir más de una entrada o salida. BPMN soporta ciclos arbitrarios en el diagrama de proceso de negocio, conectando un punto de control a las actividades donde el flujo de control recicla.

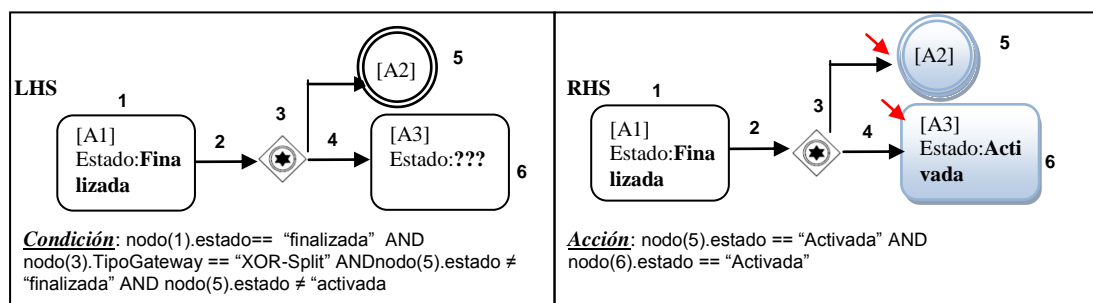


**Regla 16. Terminación implícita:** Es utilizado para determinar cuándo la instancia de un proceso se considera completa. Un determinado proceso (o sub proceso) debe terminar cuando no hay tareas pendientes o cuando no hay posibilidad de crear nuevos trabajos en el futuro. Si no hay otra actividad por ejecutar en cualquiera de las ramas, cuando se llega al Evento de fin, el proceso es cerrado automáticamente.



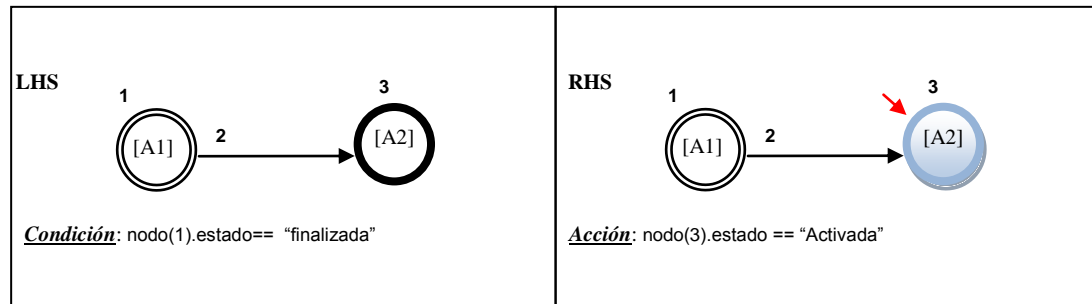
**Regla 17. Patrón de cancelación de actividad:** la regla define parar la ejecución de una particular actividad del proceso con un evento de cancelación. Cuando LHS se encuentra en el modelo

En la LHS se define la estructura de control donde se verifica la terminación de las actividades, si la compuerta es basada en eventos. Si este patrón coincide en el modelo, se habilita la actividad y el evento intermedio está disponible para tener la opción de cancelarla.



**Regla 18. Patrón de cancelación del proceso:** la regla define parar la ejecución de un proceso entero con un evento de cancelación.

En la LHS se define la estructura de control donde se verifica la ocurrencia de un evento entremedio de cancelación, que lleva posteriormente un evento de fin terminal.



Una vez realizada la descripción de las reglas de negocio, llega el momento de realizar el análisis de implementación de estas reglas.

En el **Anexo D** se muestran las reglas gramaticales implementadas en la interfaz de la herramienta.

### III.3.1.2 Aplicación de las Reglas Gramaticales.

En esta sección, se aplican las reglas gramaticales definidas en el apartado III.3.1.1 para la simulación del “Proceso de Desarrollo de Software” [18]. Los modelos definidos mediante el formalismo de la Figura 3.9, muestran una gramática de grafos que expresa la semántica operacional de este tipo de modelos a ser simulados. En AToM<sup>3</sup>, se pueden definir gramáticas de grafos de manera visual.

A continuación se describe la representación de los patrones de comportamiento en el “Proceso de Desarrollo de Software”. Las descripciones se basan en la expresividad de los lenguajes y notaciones de procesos desde la perspectiva de patrones de control de flujo, descritos en el apartado III.2.1.1, de este capítulo.

A continuación se presenta el análisis de los subprocesos del *Proceso de Desarrollo de Software*:

- ***Gestión de Proyectos:*** La Gestión del Proyecto consta de un conjunto de actividades para asegurar que la ejecución del proyecto sea exitosa. La responsabilidad principal de este proceso recae en el Líder del Equipo de Desarrollo. A través de este proceso, el Líder planifica el proyecto, determina la organización más apropiada para el grupo ó equipo de actores que desarrollará la aplicación, dirige a este grupo, administra los recursos que le han asignado al proyecto y controla el desarrollo de dicho proyecto con respecto al plan establecido.

La Gestión del Proyecto es un grupo de procesos que se ejecuta a lo largo de toda la duración del proyecto. Se inicia desde el instante en se aprueba el desarrollo de una nueva aplicación y culmina con la entrega de la aplicación.

En un proceso de gestión de proyectos existen ciertas Reglas de Negocio (RN) que se deben implementar, las cuales se detallan a continuación:

**RNI:** “Un gestor de proyecto es responsable de *elaborar el mapa de actividades según MCVS ( $G_1$ )*. Una vez que se ha completado la *identificación de necesidades del proyecto ( $D_1$ )*, realizado por el desarrollador; se genera una orden para que uno o más analistas *definan el entorno del proyecto ( $G_2$ )* y *estimen el proyecto ( $G_3$ )* y luego devuelvan los documentos con información sobre  $G_2$  y  $G_3$ . A partir de estos informes se *elabora el plan de gestión del proyecto de software ( $G_4$ )*; que será considerado válido sólo una vez que sea *verificada inconsistencias ( $S_8$ )*. Posteriormente este plan se lo utiliza para *elaborar el plan de garantía de calidad ( $G_5$ )*, *planificar riesgos y contingencias ( $G_6$ )* y *Gestionar el proyecto ( $G_7$ )*; donde a  $G_7$  se lo utiliza hasta el final de  $D_{12}$ . Paralelamente tanto el proceso  $G_6$  que al finalizar activa el proceso *Guardar documentos ( $G_8$ )* de Planificación de Riesgos y Contingencia; así como también una vez completado el plan de Garantía de Calidad en  $G_5$ , será considerado valido una vez que se *verifique los planes ( $S_{11}$ )* para luego *gestionar la calidad del proyecto ( $G_9$ )* que concluirá *con guardar cambios ( $S_{13}$ )*, archivando el Documento de calidad del proyecto software”.

Para describir la estructura y ejecución del subproceso se han identificado diferentes patrones de control de flujo a lo largo de su ejecución, los cuales se muestra en la Figura 3.10 sombreado con color rosa:

- ✓ Patrones básicos tal como parallel split o partición paralela, usada como mecanismo que representa la realización de actividades en forma concurrente. Es decir, un camino único en el proceso es particionado en dos o más caminos, de modo que dos o más actividades puedan comenzar al mismo tiempo. (Figura 3.10).

En el modelo de proceso de software, en las actividades de Gestión usamos la estructura gateway con marca “+” que controla el flujo de las salida subsecuentes. En la implementación la estructura de control verifica la terminación de la actividad  $G_1$ , es decir la disponibilidad del documento “Modelo de Ciclo de Vida del Proyecto” para dar inicio a las actividades  $G_2$  y  $G_3$  paralelamente.

En la **Regla 2: Generación de división de actividades paralelas**, se visualiza este patrón.

- ✓ Otro patrón de control básico implementado es el de sincronización, mediante el cual dos o más actividades deben completarse antes de que el proceso pueda continuar. Combina los flujos de actividades generados por la partición paralela o parallel Split. (Figura 3.11 rotulo b)

En el modelo del proceso software este caso se ha implementado mediante un gateway que controla la culminación de los flujos de entrada. Cuando todas las actividades se completan el flujo continúa en un único proceso. En el mismo se identifica las siguientes reglas de flujo:

“En el proceso de desarrollo las actividades  $G_2$  y  $G_3$  deben completarse antes de iniciarse la actividad  $G_4$  para el posterior control de documentos. Los documentos  $G_2$  y  $G_3$  contienen información relevante para la revisión de los planes generados.

En la **Regla 4. Sincronización de actividades**, se visualiza este patrón.

- ✓ Otro patrón identificado es el de *elección exclusiva*: representa un punto en el proceso donde la decisión se toma por elección de una de las posibles rutas. Está marcada por un punto de decisión exclusivo basado en datos o en eventos (*Figura 3.11 rotulo d*). En el modelo del proceso de software encontramos este patrón en las siguiente regla de flujo:

“En el proceso de desarrollo del software la actividad S<sub>11</sub> la cual consiste en verificar el documento de Plan de Garantía de Calidad, la cual al finalizar se activa un nodo de decisión para que el gestor del proyecto decida el camino a seguir; si el Proyecto es válido se activa G<sub>9</sub> caso contrario se activa S<sub>12</sub>”

En la **Regla 5. Selección exclusiva de datos**, se visualiza este patrón.

- ✓ Otro patrón identificado es el de *secuencia de actividades*: en el modelado del proceso Software identificamos este patrón en diferentes secuencias de procesos de actividades (*Figura 3.11, rotulo c*).

“En el proceso G<sub>6</sub>, donde se gestiona la Planificación de Riesgo y Contingencia se debe completar esta actividad para recién dar inicio a la actividad G<sub>8</sub>, en la cual se archiva dicho planificación.”

En la **Regla 1. Generación de secuencia de actividades**, se visualiza este patrón.

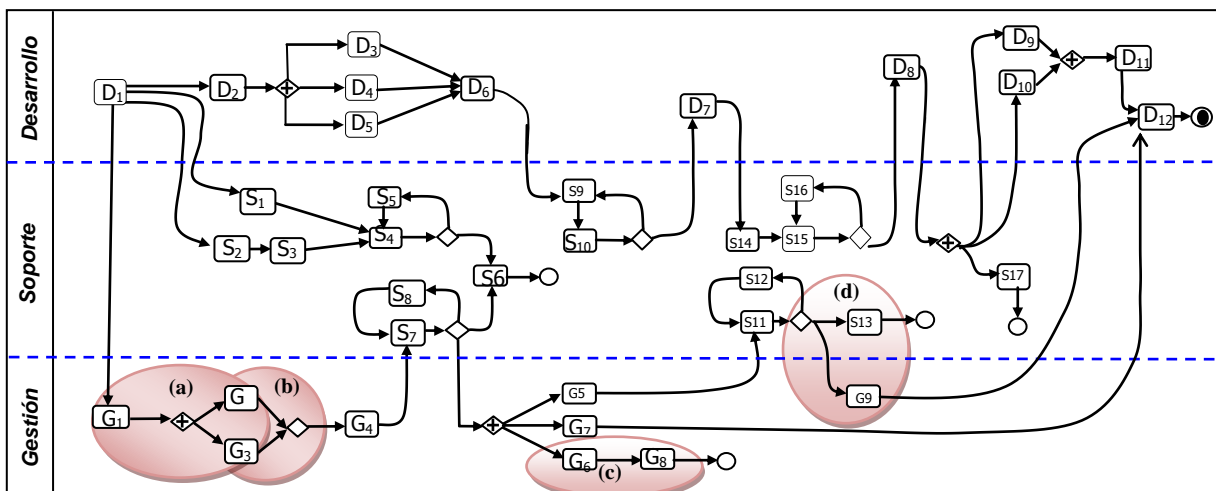


Figura 3.10. Identificación de reglas de negocio en la Gestión de Proyectos del Proceso de Software.

- **Gestión de soporte:** Adicionalmente a los procesos de gestión, existe un grupo de procesos que contribuye a hacer más efectivos los procesos de gestión. Este grupo de procesos los denominamos procesos de soporte. Su objetivo principal es complementar los procesos de gestión a través de la gestión de los productos, personas y procesos asociados al desarrollo de aplicaciones.

En un proceso de gestión de soporte existen ciertas reglas de negocio que se deben implementar, las cuales se detallan a continuación:

**RN2:** “Una vez que se ha *identificado las necesidades del proyecto* (D<sub>1</sub>), se realiza la *identificación de la configuración del sistema* (S<sub>2</sub>) y consecuentemente se *planifica la gestión de configuración del software* (S<sub>3</sub>), paralelamente se *elabora el plan de verificación y validación* (S<sub>1</sub>). Cuando se ha completado la actividad S<sub>1</sub>. Cuando culminan las actividades S<sub>2</sub> y S<sub>3</sub> comienza la *verificación de los planes* (S<sub>4</sub>), en el caso que haya que realizar algún ajuste, se *corrige el plan* (S<sub>5</sub>) y se vuelve a verificar hasta que sea aprobado y se procede a *guardar la documentación* (S<sub>6</sub>).

Para describir la estructura y ejecución del subproceso se han identificado diferentes patrones de control de flujo a lo largo de su ejecución, los cuales se muestra en la Figura 3.11 sombreado con color verde:

- ✓ **Regla 1. Generación de secuencia de actividades:** Esta regla se identifica en el modelo de gestión de proyecto, ya que se puede observar en la *Figura 3.11 rotulo f*. En donde la actividad S<sub>2</sub>, que representa la *identificación de la configuración del sistema* se debe completar para recién dar inicio a la actividad S<sub>3</sub> que es la *planificación de la gestión de configuración del software*.
- ✓ **Regla 15. Generación de ciclos arbitrarios:** Se ha identificado este *patrón estructural* cuando la actividad G<sub>4</sub>, activa S<sub>7</sub> para que este verifique el *Plan de Gestión del Proyecto del Software* obtenido con la finalización de dicho proceso, una vez realizada la verificación se existieran inconsistencias se corrige las mismas a través del proceso S<sub>8</sub> y esto puede generar un ciclo las veces que sean necesario hasta que el documento sea considerado como valido. “(Figura 3.11 rotulo C).Esta regla también se puede observar que esta regla se encuentra en la Figura 3.11 rotulo G, rotulo H, rotulo I, y rotulo J.

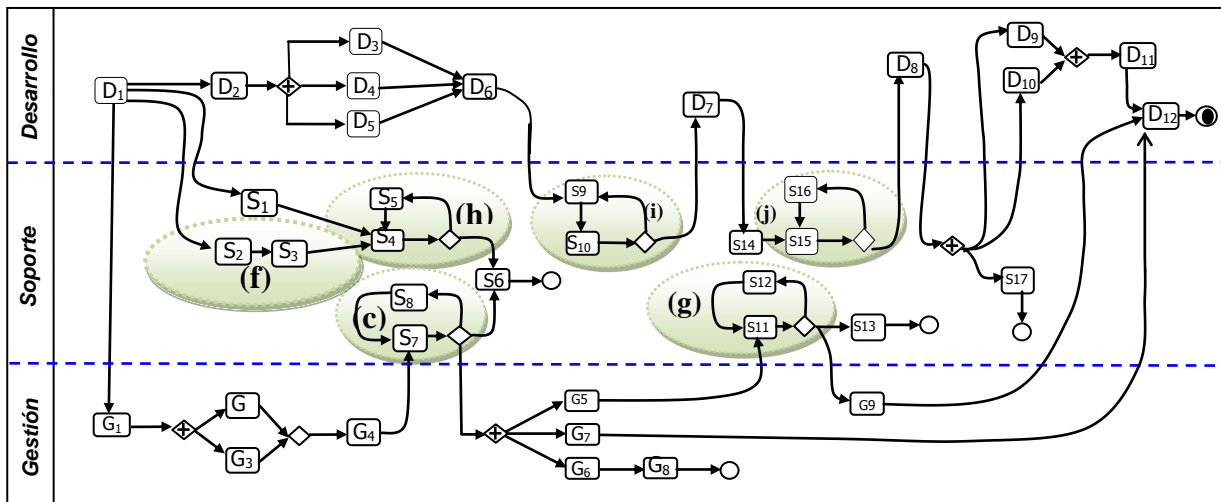


Figura 3.11. Identificación de reglas de negocio en la Gestión de Soporte del Proceso de Software.

- **Gestión de desarrollo:** Este proceso contiene las tareas de creación de software propiamente dichas.

En un proceso de gestión de desarrollo existen ciertas reglas de negocio que se deben cumplir, las cuales se detallan a continuación:

**RN3:** “La gestión de desarrollo comienza con la *identificación de necesidades del proyecto* ( $D_1$ ), una vez que este proceso termina se da inicio a la *especificación de requisitos funcionales* ( $D_2$ ). La finalización de  $D_2$  dará inicio a un *Gateway paralelo*, el cual activara de manera simultánea los procesos de: *realizar diseño preliminar* ( $D_3$ ), *diseñar base de datos* ( $D_4$ ) y *diseñar interfaces* ( $D_5$ ). Culminado estos procesos se *realiza el diseño detallado* ( $D_6$ ). Cabe destacar que estas tareas se ejecutan de manera paralela con las actividades de Soporte y de Gestión del proceso de Software.

Una vez que en las actividades de Soporte se hayan *revisado el diseño* ( $S_9$ ) y haya pasado las pruebas pertinentes se comienza a *crear el código* ( $D_7$ ), el cual finalizado se *verifica* ( $S_{15}$ ), si la verificación resulta exitosa se comenzara a *planificar la integración* ( $D_8$ ) que genera un Documento de Plan de Integración. Finalizado  $D_8$ , se inicia de manera paralela la *creación de los datos de prueba* ( $D_9$ ) y la *integración de los módulos* ( $D_{10}$ ). Solo cuando estos dos procesos terminen se dará comienzo a la *instalación del sistema* ( $D_{11}$ )

Para entregar los documentos del proceso de desarrollo de software al cliente se *guardan los informes finales* ( $D_{12}$ ), solo se realiza cuando hayan finalizado la *gestión de calidad del proyecto* ( $G_9$ ) y la *gestión del proyecto* ( $G_7$ ). Dadas las condiciones mencionadas se entrega al cliente los documentos y finaliza el proceso.”

Para describir la estructura y ejecución del subproceso se han identificado diferentes patrones de control de flujo a lo largo de su ejecución, los cuales se muestra en la Figura 3.10 sombreado con color violeta:

- ✓ **Regla 1. El patrón de secuencia de actividades:** en el modelado del proceso Software identificamos este patrón en diferentes secuencias de procesos de actividades:

“En el proceso  $D_1$  se identifican las necesidades del proyecto, una vez que este proceso finalice se da comienzo con los procesos  $D_2$ , especificar requisitos funcionales,  $S_1$ , elaborar plan de verificación y validación,  $S_2$ , identificar configuración del sistema y  $G_1$ , elaborar el mapa de actividades según el MCVS.” (Ver Figura 3.12 rotulo k). Otro patrón donde se expresa la secuencia de actividades es en el proceso  $D_{11}$ , instalar el sistema, que una vez finalizado comienza el proceso  $D_{12}$ , guardar informes finales (Figura 3.12 rotulo l).

- ✓ **Regla 2. Patrones básicos parallel split o partición paralela:** En el modelo de proceso de software, en las actividades de Gestión de Desarrollo identificamos que en la culminación del proceso  $D_2$ , donde se realiza la especificación de los requisitos funcionales, se activan de manera paralela las actividades  $D_3$ , Realizar el diseño

preliminar, D<sub>4</sub>, diseñar base de datos y D<sub>5</sub>, diseñar interfaces. Otro patrón identificado es en el proceso D<sub>8</sub>, planificar integración, que finalizado da inicio a dos actividades paralelas como son D<sub>9</sub>, crear datos de prueba, D<sub>10</sub>, integrar módulos del sistema y S<sub>17</sub>, Guardar cambios (Figura 3.12 rotulo m y rotulo n).

- ✓ **Regla 3. Sincronización de actividades:** Otro patrón de control básico implementado es el de sincronización, mediante el cual dos o más actividades deben completarse antes de que el proceso pueda continuar. En el modelo del proceso software en la gestión de desarrollo, cuando todas las actividades se completan el flujo continúa en un único proceso. En el mismo se identifica las siguientes reglas de flujo:

“En el proceso de desarrollo las actividades D<sub>3</sub>, D<sub>4</sub> y D<sub>5</sub> deben completarse antes de iniciarse la actividad D<sub>6</sub>, que es realizar el diseño detallado. (Ver Figura 3.12 rotulo o)

- ✓ **Regla 4. Selección exclusiva:** En el modelo del proceso de software encontramos este patrón en las siguiente regla de flujo:

“En el proceso de Gestión de desarrollo del software la actividad D<sub>8</sub>, planificar integración, se activa si en el proceso de gestión de soporte se selecciona entre S<sub>16</sub>, corregir errores de código o D<sub>8</sub>, si se selecciona este último se sigue gestionando la etapa de desarrollo” (Figura 3.12 rotulo p)

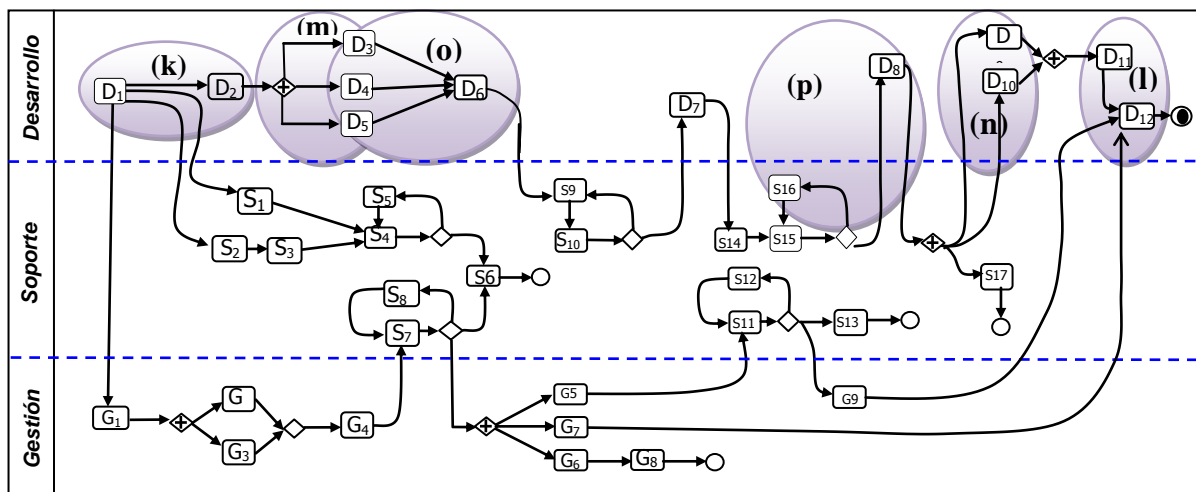


Figura 3.12. Identificación de reglas de negocio en la Gestión de Desarrollo del Proceso de Software.

Con respecto al modelado de proceso software con enfoque de negocio se puede constatar la factibilidad de contar con un modelo de proceso integrado de software y negocio, el cual ha sido modelado con técnicas orientadas por los negocios, lo que ha permitido plasmar de una forma adecuada la semántica presente entre los patrones que conforman el modelo. También se ha mostrado que el modelo obtenido es válido ya que refleja las reglas que indican como se hace el negocio y las restricciones que existen en el mismo, de modo que nunca sea posible llevar a cabo acciones no válidas. En el **Anexo E** se muestra la simulación del proceso de desarrollo del software mediante la definición de las reglas de negocio implementadas en el entorno de AToM<sup>3</sup>.

Los resultados alcanzados a partir de la realización de este trabajo constituyen los diferentes aportes que se describen a continuación:

- De tipo *conceptual*: se obtiene la definición de dos modelos conceptuales, por un lado el **metamodelo de BPMN**, y por otro lado, el modelo del **Proceso Software orientado al Negocio** optimizado y verificado.

En cuanto a los modelos definidos: el metamodelo BPMN se ha representado mediante un diagrama de proceso de negocio en el cual se define la sintaxis y semántica del mismo. A través de sus elementos principales, se presentan gráficamente una secuencia de actividades que ocurre en un proceso de negocio, y posibilita la generación de transformaciones de manera más fácil. En cuanto el modelo de Proceso Software orientado al negocio obtenido permite representar el flujo de trabajo, la estructura fundamental y los elementos requeridos haciendo uso de la herramienta AToM<sup>3</sup>.

- De tipo *metodológico*: se ha logrado la definición de dos procedimientos metodológicos, el primero está orientado a la definición de metamodelos, el segundo está orientado a la comprobación de la correctitud de los modelos.

Con respecto al primero, el proceso de metamodelado, ha ayudado a conocer los detalles del formalismo BPMN y por tanto, a comprender sus posibilidades de especificación, y también las limitaciones de dicho formalismo.

Incluido en el contexto de metamodelado, se demuestra la factibilidad de utilizar las gramáticas de grafos, para representar y simular el comportamiento de los patrones de procesos de negocio instanciados. También se puede decir que la especificación gráfica y formal de las manipulaciones sobre los modelos permite obtener una mejor comprensión de la dinámica del proceso modelado.

- Y por ultimo de tipo *práctico*: obtenido con la actualización de la herramienta AToM<sup>3</sup> con la implementación del metamodelo BPMN para manipular los modelos que serán representados con dicha notación.

La actualización de AToM<sup>3</sup> con lenguaje BPMN permite contar con una herramienta que proporciona la capacidad de *diseñar, modelar y simular* los procesos de negocio, flujos de información y almacenamientos de datos con una interfaz de fácil manejo por medio de un ambiente gráfico sin necesidad de programar.

En este trabajo ha surgido un interrogante en cuanto a, *si la aplicación y el uso de AToM<sup>3</sup> para la generación de entornos de modelado con BPMN aumentaría la productividad y produciría una disminución del tiempo insumido en el desarrollo de procesos*; quedando la pregunta abierta y dando lugar a un trabajo futuro que consistiría en la realización de diversas pruebas experimentales en diferentes contextos de aplicación para llegar a conclusiones más acertadas.

- [1] Acuña S. T., de Antonio A., Ferré X., López M. y Maté L., *The Software Process: Modelling, Evaluation and Improvement. Handbook of Software Engineering and Knowledge Engineering*. Vol. I (World Scientific, 2001) 193-237.
- [2] Andersson B., Bider I., Johannesson P., Perjons E., *Towards a Formal Definition of Goal-Oriented Business Process Patterns*, *Business Process Management Journal* 11(6): pp. 650-662 (2005).
- [3] Araújo, J., Hernández, J., Navarro, E. y Pinto, M., *Desarrollo de Software Orientado a Aspectos: actas del IV Taller de Desarrollo de Software Orientado a Aspectos DSOA'06*, 3 de Octubre de 2006, Sitges (Barcelona). págs. 41-49. Servicio de Publicaciones, Universidad de Extremadura: (Cáceres), España, 2006. [Consultada Enero 2010]
- [4] AToM3 Home page: <http://atom3.cs.mcgill.ca> [Consultada Diciembre 2009]
- [5] Bauer B., Lautenbacher F., Roser S., *AgilPro Metamodel Description*. [http://wiki.eclipse.org/images/2/2f/AgilPro\\_MetamodelDescription.pdf](http://wiki.eclipse.org/images/2/2f/AgilPro_MetamodelDescription.pdf). (Acceso marzo de 2008).
- [6] Berrocal J., García J. M., Murillo J. M., *Quercus Software Engineering Group*. Dept. de Ingeniería de Sistemas Informáticos y Telemáticos. Universidad de Extremadura 10071 Cáceres
- [7] Bjorn A., Kindler E., Rubin V., *AMFIBIA: A Meta-Model for the Integration of Business Process Modelling Aspects. Dagstuhl Seminar Proceedings 06291. The Role of Business Processes in Service Oriented Architectures*. <http://drops.dagstuhl.de/opus/volltexte/2006/823>. [Consultada Diciembre 2009]
- [8] BPMN, *Business Process Modeling Notation, Specification Version 1.0*. Disponible en <http://www.bpmn.org/> [Consultada Diciembre 2009].
- [9] Corredera de Colsa, Luis E., *Arquitectura dirigida por modelos para J2ME*. Disponible en: [http://personal.telefonica.terra.es/web/lencorredera/mda\\_j2me.pdf](http://personal.telefonica.terra.es/web/lencorredera/mda_j2me.pdf). [Consultada febrero 2010].
- [10] De la Vara González, J., Sánchez Díaz, J., *Derivación de modelos de tareas a partir de modelos BPMN*. Disponible en: <http://alarcos.inf-cr.uclm.es/pnis/articulos/pnis-07-SanchezJ-DMTBPMN.pdf> [Consultada Diciembre 2009].
- [11] De Lara, J., y Taentzer, G., *Automated Model Transformation and its Validation with AToM3 and AGG. Diagrams'2004*. Lecture Notes in Artificial Intelligence 2980, Springer-Verlag. 82-198. (2004).
- [12] De la Vara J, L., M., Fortuna H., Sánchez J., Werner C. M. L., Borges M. R. S., *Centro de Investigación en Métodos de Producción de Software*, Universidad Politécnica de Valencia, España.

- [13] Delgado. A, *Desarrollo de Software con enfoque en el Negocio*, Disponible en: <http://alarcos.inf-cr.uclm.es/pnis/articulos/pnis-07-delgado-dsen.pdf>. [Consultada diciembre 2009].
- [14] Excellentia Consultores. *Procesos de negocio y ventaja competitiva*. Universidad ORT (Uruguay). 2007. Disponible en [http://www.excellentia.com.uy/articulos/excellentia\\_articulo\\_1185641521.pdf](http://www.excellentia.com.uy/articulos/excellentia_articulo_1185641521.pdf). [Consultada diciembre 2009].
- [15] Finkelstein A., Kramer J. y Nuseibeh B., *Software Process Modelling and Technology*. Research Studies Press, (1994).
- [16] Geoffrey Sparks, Sparx Systems. *El Modelo de Proceso de Negocio*. Disponible en: [http://www.sparxsystems.com.ar/resources/tutorial/business\\_process\\_model.html](http://www.sparxsystems.com.ar/resources/tutorial/business_process_model.html). [Consultada febrero 2010].
- [17] Hsu L. y R. Obe. *Business rules and objects orientation* <http://www.paragoncorporation.com/papers/SACM-96-282-0.htm>. 1996. [Consultada diciembre 2009].
- [18] IEEE, *Standard for Developing Software Life Cycle Processes*, IEEE Standard 1074-1997. (1997).
- [19] Jacobson I., Booch, Grady. Rumbaugh J.. *El proceso unificado de desarrollo de software*. Addison Wesley. España. .2000. Pág XVI.
- [20] Khodabandeh, A., Palazzi P., *Software Development: People, Process, Technology. Proceedings of the 1994 CERN*. School of Computing, Sopron, Hungary. 1994.
- [21] Kolmayer G., H. Herbst y M. Schlesinger. *Enforcing business rules by the applications trigger concepts. Methods and associated tools for the information systems life cycle*. Actas de IFIP WG8.1. A. A. Verrijn-Stuart y T. W. Olle (Eds.). Elsevier. 1994. pp. 29-46
- [22] Lehman M. M., *Software Engineering, the Software Process and their Support*. *Software Engineering Journal* 6, 5 243-258. (1991)
- [23] Letelier, P. *Proyecto Docente e Investigador*, DSIC, 2003.
- [24] McChesney I. R., *Toward a Classification Scheme for Software Process Modelling Approaches*. *Inform. and Soft. Technology* 37, 7 363-374. (1995).
- [25] Mendling J., Neumann G., *A Comparison of XML. Interchange Formats for Business Process Modelling. Proceedings of the EMISA 2004 Workshop "Information Systems in E-business and E-Government"*. Vol.56. of Lecture Notes in Informatics (LNI). pp. 129-140 (2004)
- [26] Morris D., Brandon, J. *Reingeniería: como Aplicarla con Éxito en los Negocios*. Editorial Mc Graw Hill. España. (2000).
- [27] Object Management Group (OMG) Disponible en: <http://www.omg.org>. [Consultada Diciembre 2009].
- [28] Paz Pérez González, Framiñán Torres J. M., Parra C. L., Calderón, González Rodríguez L. P., León Blanco J. M., *Interoperabilidad de sistemas guiado por modelos de procesos*

*de negocios: una aplicación en el sector sanitario. X Congreso de Ingeniería de Organización Valencia, 7 y 8 de septiembre de 2006*

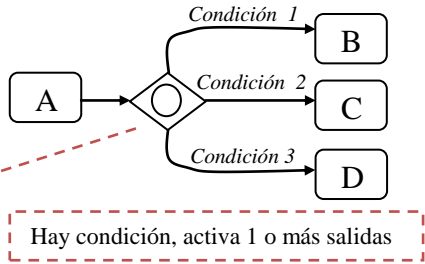
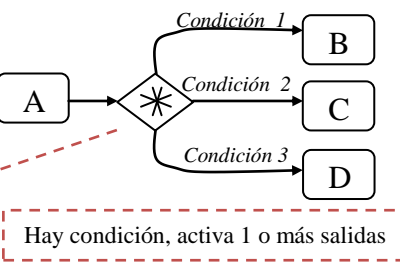
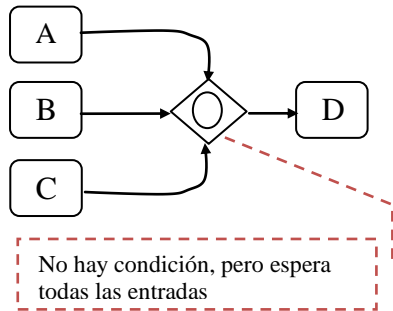
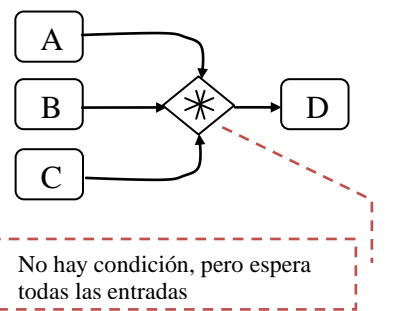
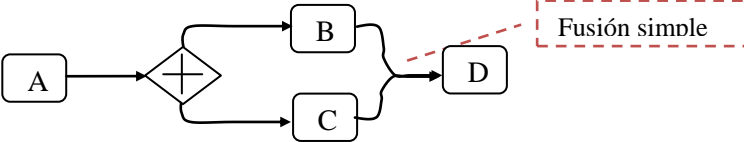
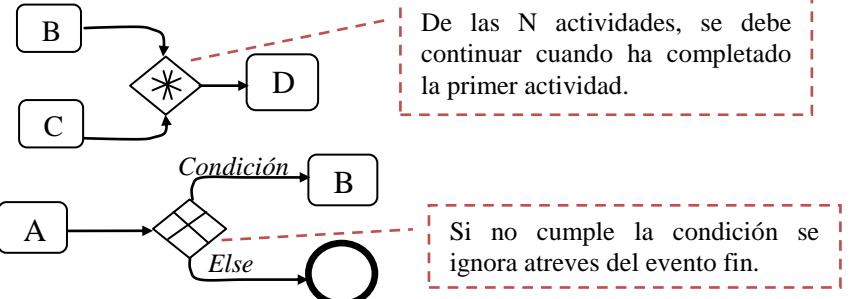
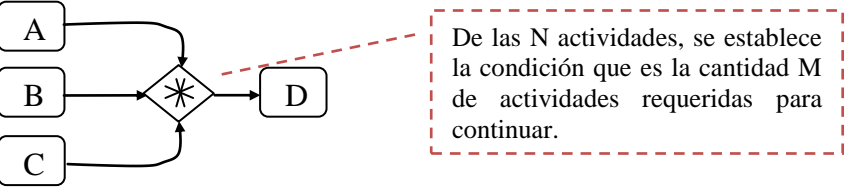
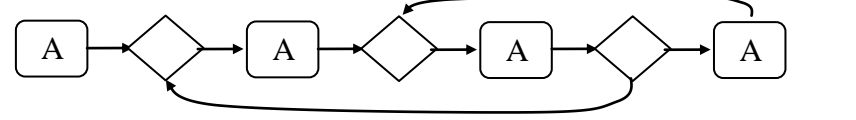
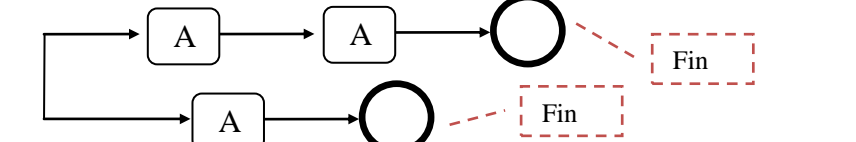
- [29] Perry, D., A. Porte, et al. (2000). *Empirical Studies of Software Engineering: A Roadmap. Future of Software Engineering*: 345-355.
- [30] Pfleeger, Shari. Lawrence. *Ingeniería de software teoría y práctica. Prentice Hall*. 1ra edición Buenos Aires 2002.
- [31] Reina A. M. y Torres J., Toro M. *Hacia lenguajes de metamodelado orientados a aspectos. Desarrollo de Software Orientado a Aspectos, DSOA 2006*. Disponible en: <http://www.info-ab.uclm.es/personal/ElenaNavarro/DSOA06/papersCR/DSOA06-reinaqu.pdf>
- [32] Rodríguez, V. y Garcia Molina, J., *Estudio comparativo entre modelos. Proyecto informático, facultad de informática de la universidad de Murcia: ingeniería de modelos con MDA*. Disponible en: <http://dis.um.es/~jmolina/pfcs/proyecto-mda.pdf>. [Consultada Enero 2010].
- [33] Rolón E., Ruiz F., Garcia F. y Piatini M., *Aplicación de Métricas Software en la Evaluación de Modelos de Procesos de Negocio. Revista Sociedad Chilena de Ciencia de la Computación* (6)1. (2005).
- [34] Rozemberg G., *Handbook of Graph grammars and Computing by Graph Transformations*, Vol.1:Foundations. World Scientific. (1997)
- [35] Pressman R. *Ingeniería del Software - Un Enfoque Práctico*. Quinta Edición. Editorial Mc Graw Hill. España. (2002).
- [36] Singh R., *Roceedings of the Winter Simulation Conference*. (1996).
- [37] Smith, H. y Fingar, P., *Business Process Management, the third wave. Meghan-Kieffer Press*, (2003).
- [38] Söderström E. et al., *Towards a Frameworks for Comparing process Modelling Languages*. Proceeding of the 14th International Conference on Advanced Information Systems Engineering (CAISE 2002). Springer Verlag. pp. 600-611 (2002)
- [39] Solivares P. A., *Desarrollo Cliente/Servidor: Ubicación de las reglas de negocio. Revista Profesional para Programadores (RPP)*. (1997)
- [40] Sommerville, I., *Ingeniería de Software*, Pearson Educación. (2002).
- [41] Sosa M., Acuña S., De Lara J., Leguizamon F., *Simulación de modelos de procesos de negocio mediante metamodelado y gramática de grafos*. Disponible en <http://lahuen.dcc.uchile.cl/~jcc2008/actas/ECC/booklet.pdf> [Consultada diciembre 2009].
- [42] Von V., Karow M., Radebeul A. *Business Process Documentation in Creative Work Systems. A Design Science Study in Television Production*
- [43] Web The Workflow Management Coalition: [www.wfmc.org](http://www.wfmc.org).

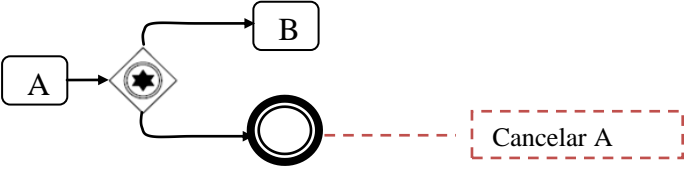
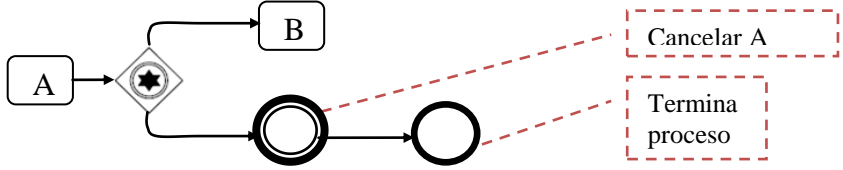
- [44] Wohed P., van der Aalst W.M.P., Dumas M., ter Hofstede A.H.M., Russell N.: *Pattern-based Analysis of BPMN. An extensive evaluation of the Control flow, the Data and Resource Perspectives. BPM Center Report BPM*. Disponible en <http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports.htm> (Acceso junio de 2008)
- [45] White S.A., *Process Modeling Notations and Workflow Patterns. IBM Corp., United States*. (2004). <http://www.itp-commerce.com/pdf/Intro%20to%20BPMN.pdf> (Acceso junio de 2008)
- [46] White S. A. (Ed.), Introduction to BPMN. BPML Specification Version 1.0. Business Process Management Initiative 1.0, BPMI, Disponible en: <http://www.BPMI.org..> [Consultada Enero 2010].
- [47] Yourdon, E., *Decline an Fall of the American Programmer. Englewood Cliffs, NJ. USA, Prentice Hall Inc, 1992*.
- [48] Zaniolo C., S. Ceri, C. Faloutsos, R. Snodgrass, V. Subrahmanian y R. Zicari. *Advanced database systems*. Morgan Kaufmann Pub. 1997.
- [49] Zorzan. F. y Riesco D., *Automatizacion de procesos de desarrollo de software definidos con SPEM*, Disponible en: <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/070.pdf>. [Consultada Septiembre 2008].

En este Anexo se hará una presentación del diseño de los patrones de workflows de BPMN (Ver Tabla A.1).

Tabla A.1. Patrones de BPMN

Tipo		Representación			
Patrones Básicos de Control de Flujo	Patrón de secuencia				
	Patrón de división paralela (AND/split)				
	Patrón sincronizador o unión paralela (AND/join)				
	Patrón de elección exclusiva (XOR/split)	Basado en datos			
		Basado en eventos			
		Patrón de Fusión simple o unión exclusiva (XOR/join)			
			Patrón de Multi-	Con condición inclusiva	Con evaluación compleja

	<p><i>Elección o elección múltiple (OR/split)</i></p>		
	<p><i>Patrón de fusión de sincronización unión sincronizada (AND/joint)</i></p>	<p><i>Con condición inclusiva</i></p> 	<p><i>Con evaluación compleja</i></p> 
	<p><i>Patrón de Multi-Fusión o unión múltiple</i></p>		
	<p><i>Patrón discriminador</i></p>	 <p>De las N actividades, se debe continuar cuando ha completado la primer actividad.</p>	
	<p><i>Patrón unión N/M</i></p>	 <p>De las N actividades, se establece la condición que es la cantidad M de actividades requeridas para continuar.</p>	
<p><b>Patrones estructurales</b></p>	<p><i>Patrón de ciclos arbitrarios</i></p>		
	<p><i>Patrón de terminación implícita</i></p>		

<b>Patrones de cancelación</b>	<i>Patrón de cancelación de actividad</i>	 <p>The diagram shows a flow starting with a rounded rectangle labeled 'A'. An arrow points from 'A' to a diamond-shaped gateway containing a star. From the gateway, two arrows branch out: one to a rounded rectangle labeled 'B' and another to a thick-bordered circle. A dashed red line connects this thick-bordered circle to a dashed red box labeled 'Cancelar A'.</p>
	<i>Patrón de cancelación del proceso</i>	 <p>The diagram shows a flow starting with a rounded rectangle labeled 'A'. An arrow points from 'A' to a diamond-shaped gateway containing a star. From the gateway, two arrows branch out: one to a rounded rectangle labeled 'B' and another to a thick-bordered circle. An arrow points from this thick-bordered circle to another circle. Two dashed red lines originate from the second circle: one points to a dashed red box labeled 'Cancelar A' and the other points to a dashed red box labeled 'Termina proceso'.</p>

HERRAMIENTA ATOM<sup>3</sup> ACTUALIZADA

En este Anexo, se muestra la herramienta ATOM<sup>3</sup> con el enfoque de negocio BPMN, la cual se obtuvo con la implementación del metamodelo desarrollado en este trabajo. Mostraremos detalladamente, cómo está reflejada la sintaxis abstracta y concreta.

El diseño de cada símbolo lo mostramos a través de este anexo, en donde en primera medida le indicamos como dibujar en el entorno cada simbología como así también definir los tipos y atributos requeridos por cada uno.

En este anexo se presentan las pantallas e interfaces con cada uno de los símbolos de BPMN tal cual son implementados y representados. Se presenta de forma grafica cada elemento y las pantallas correspondientes en ATOM<sup>3</sup> a sus atributos.

Tal como se puede apreciar, en la Figura B.1, la pantalla principal de la herramienta ATOM<sup>3</sup> con enfoque BPMN, la cual se compone de diferentes menús, áreas de trabajo y vínculos que permiten acceder a cada una de las funcionalidades de la herramienta. La pantalla principal, está dividida en las siguientes partes:

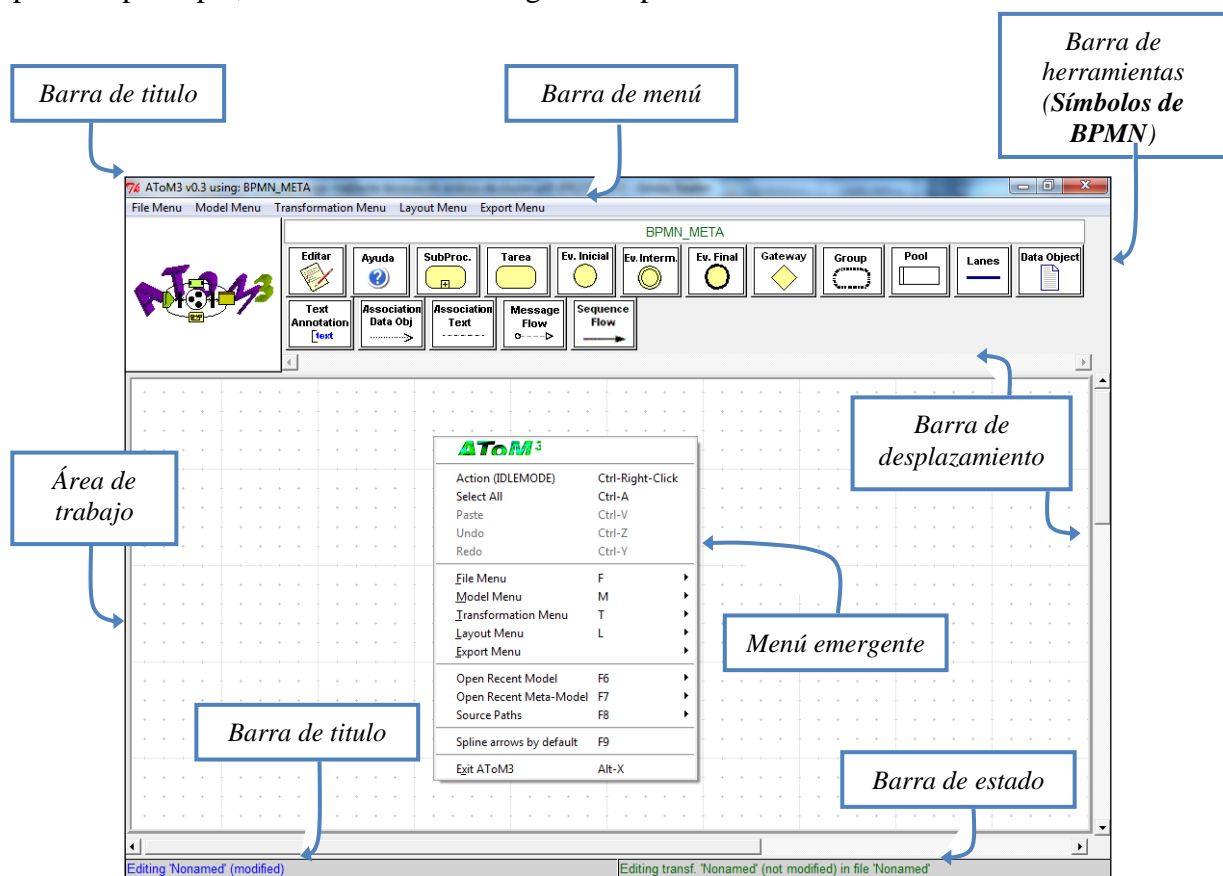


Figura B.1. Herramienta ATOM<sup>3</sup> actualizada con BPMN.

### **Barra de Título:**

Existen dos barras de título. La barra de título de arriba indica el nombre del metamodelo, y el icono que identifica al programa. En la barra de título de abajo indica el nombre del archivo, cuando guardemos por primera vez el modelo nos pedirá un nombre para guardarlo, este nombre es el que aparecerá en esta barra de título cada vez que abramos el modelo.

### **Barra de menú:**

Menú ubicado en la parte superior de la aplicación, posee menús desplegables, es decir listados de comandos para la manipulación de la información. En estos menús se puede realizar todas las operaciones permitidas en AToM<sup>3</sup>. En este menú se ejecutan las siguientes funcionalidades:

- *Menú archivo:* En este sub-menú se encuentran las opciones abrir o guardar un modelo, abrir o cerrar el meta-modelo, abrir la consola y las opciones de configuración de AToM3.
- *Menú modelo:* En este sub-menú se encuentran las opciones insertar o expandir un modelo, editar los atributos de un modelo y generar código en lenguaje *Python*.
- *Menú transformaciones:* En este sub-menú se encuentran las opciones de abrir, guardar y generar transformaciones, en la cual se cargan las reglas de negocio para realizar la simulación del modelo.

### **Barra de herramientas:**

Contiene iconos para dibujar los símbolos de BPMN, como ser actividad, evento, gateway, grupo, *pools*, *lane*, flujo de mensaje, flujo de secuencia, etc.

A continuación se muestra la sintaxis concreta propiedades o atributos de los elementos de un DPN mencionados en el apartado III.2.1.

- 1) **Tarea:** En la Figura B.2 se puede observar cómo se representa gráficamente una tarea en AToM<sup>3</sup> y sus atributos.

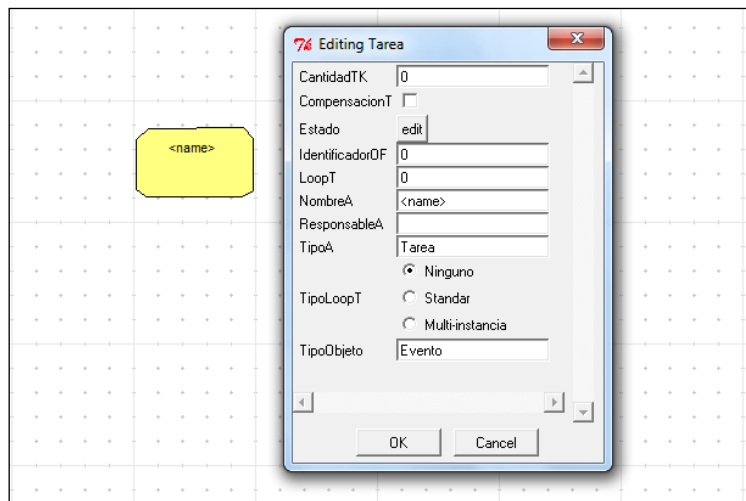


Figura B.2. Representación gráfica de una tarea y sus atributos.

Dependiendo del *tipo de loop* o la opción de *compensación* se agregan marcadores internos a una tarea, los cuales también se pueden combinar como se puede observar en la Figura B.3.

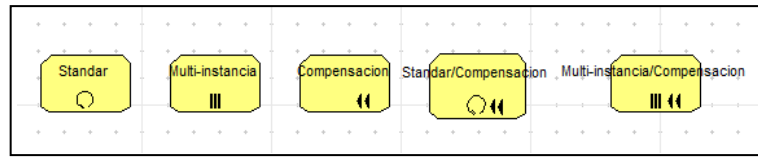


Figura B.3. Marcadores de una tarea.

- 2) **Subproceso:** En la Figura B.4 se puede observar cómo se representa gráficamente un subproceso en AToM<sup>3</sup> y sus atributos.

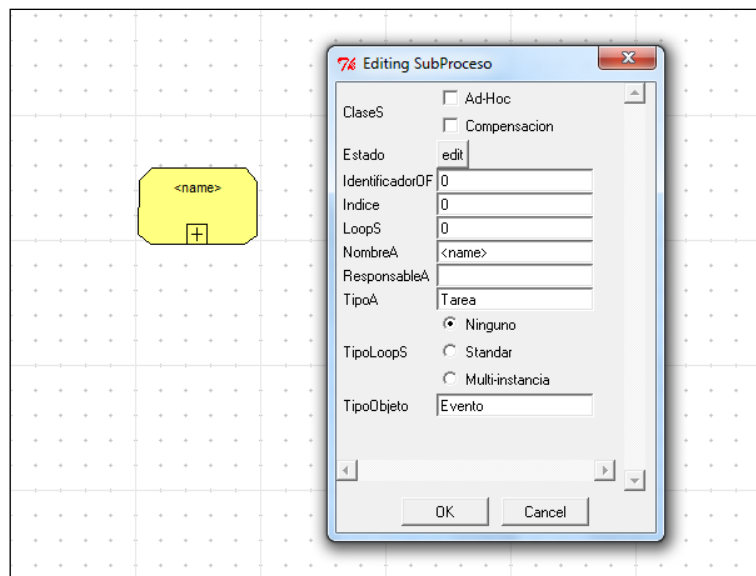


Figura B.4. Representación grafica de un subproceso y sus atributos.

Dependiendo el *tipo de loop* o la opción de *compensación* y *had-hoc*se agregan marcadores internos a un subproceso, los cuales también se pueden combinar como se puede observar en la Figura B.5

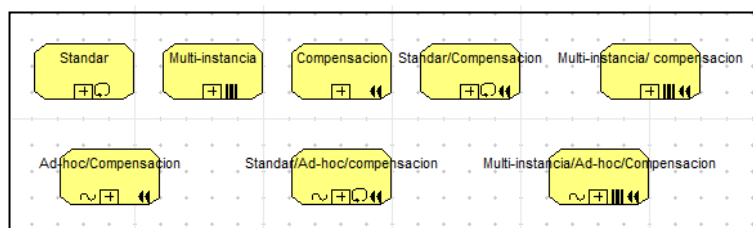


Figura B.5. Marcadores de un subproceso.

- 3) **Evento inicial:** En la Figura B.6 se puede observar cómo se representa gráficamente un evento inicial en AToM<sup>3</sup>, y sus atributos.

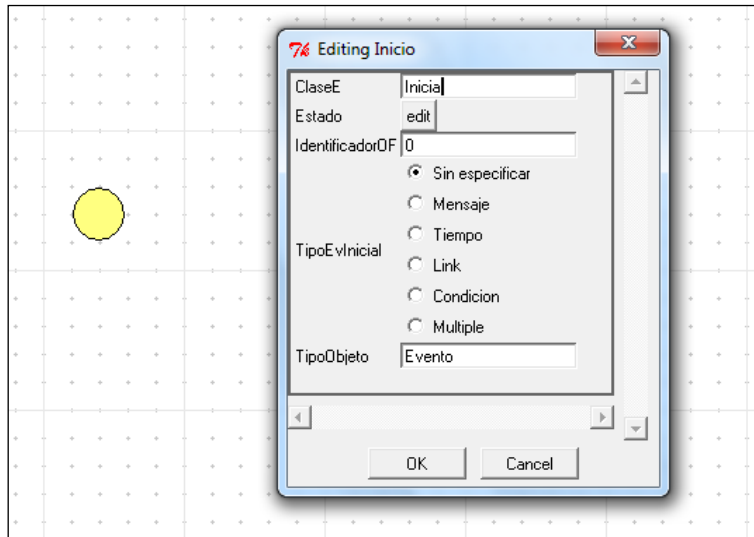


Figura B.6. Representación gráfica de un evento inicial y sus atributos

Dependiendo del *tipo de evento inicial* agregan los iconos internos a un evento inicial como se puede observar en la Figura B.7

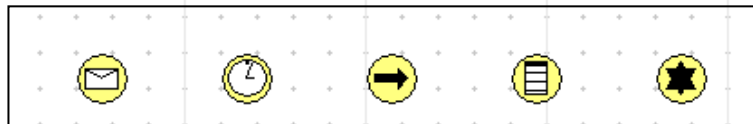
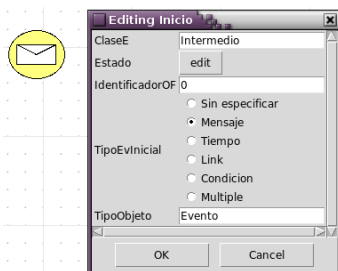
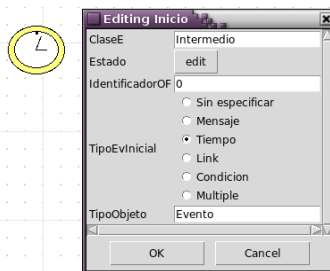


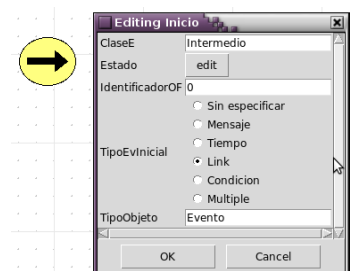
Figura B.7. Tipos de evento inicial.



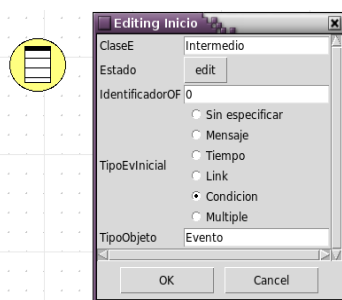
Mensaje



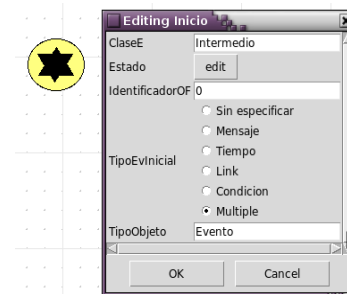
Tiempo



Link



Condición



Múltiple

4) **Evento intermedio:** En la Figura B.8 se puede observar cómo se representa gráficamente un evento intermedio en ATOM<sup>3</sup>, y sus atributos.

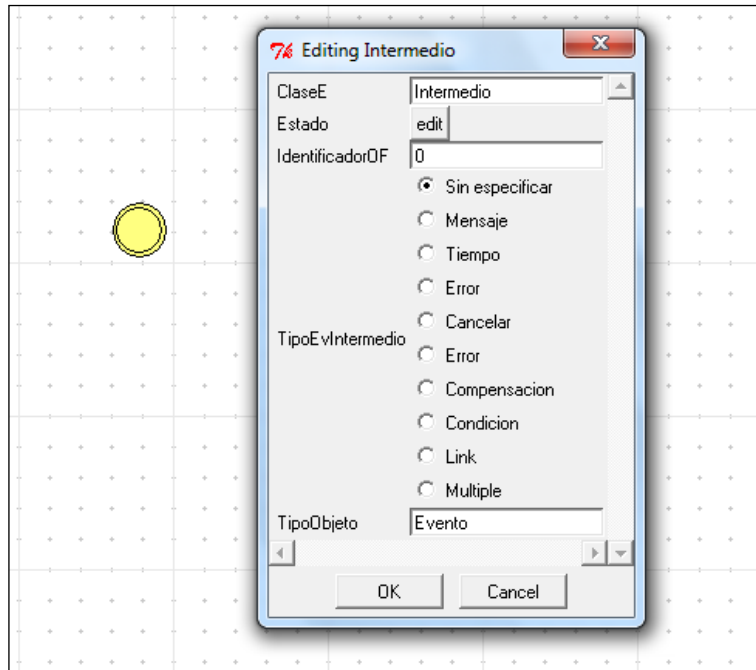


Figura B.8. Representación grafica de un evento intermedio y sus atributos.

Dependiendo del *tipo de evento intermedio* agregan los iconos internos a un evento inicial como se puede observar en la Figura B.9

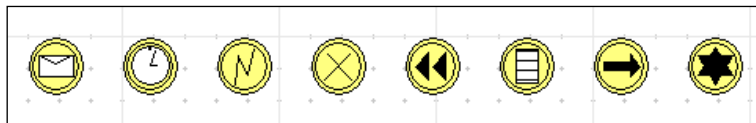
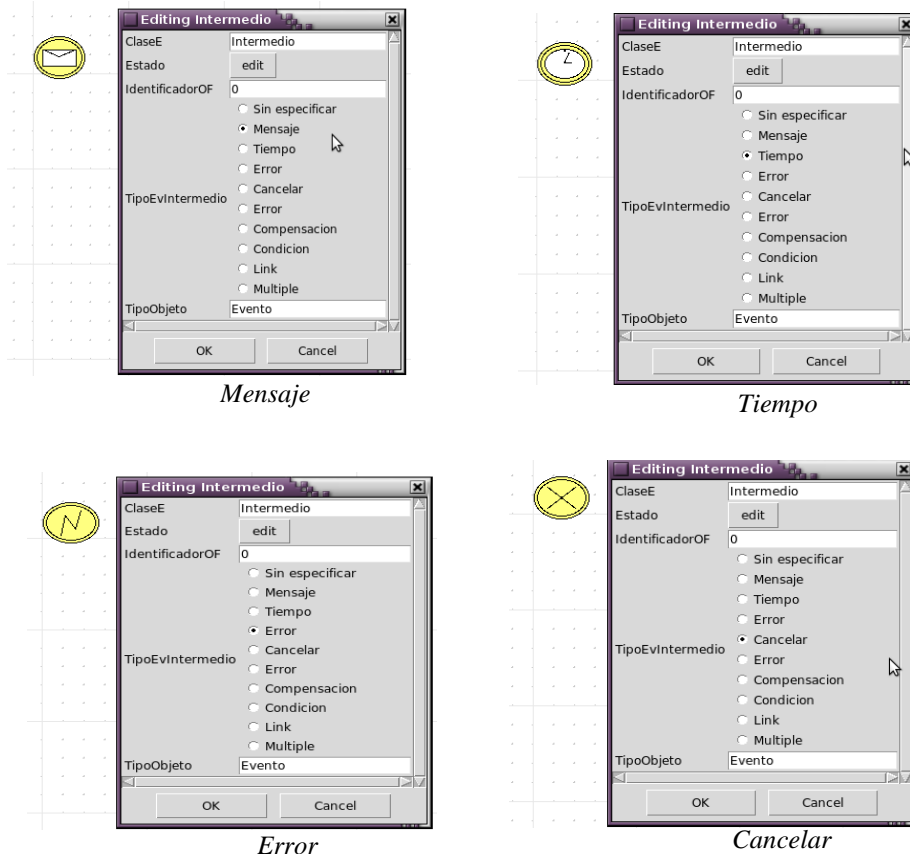


Figura B.9. Tipos de evento intermedio.

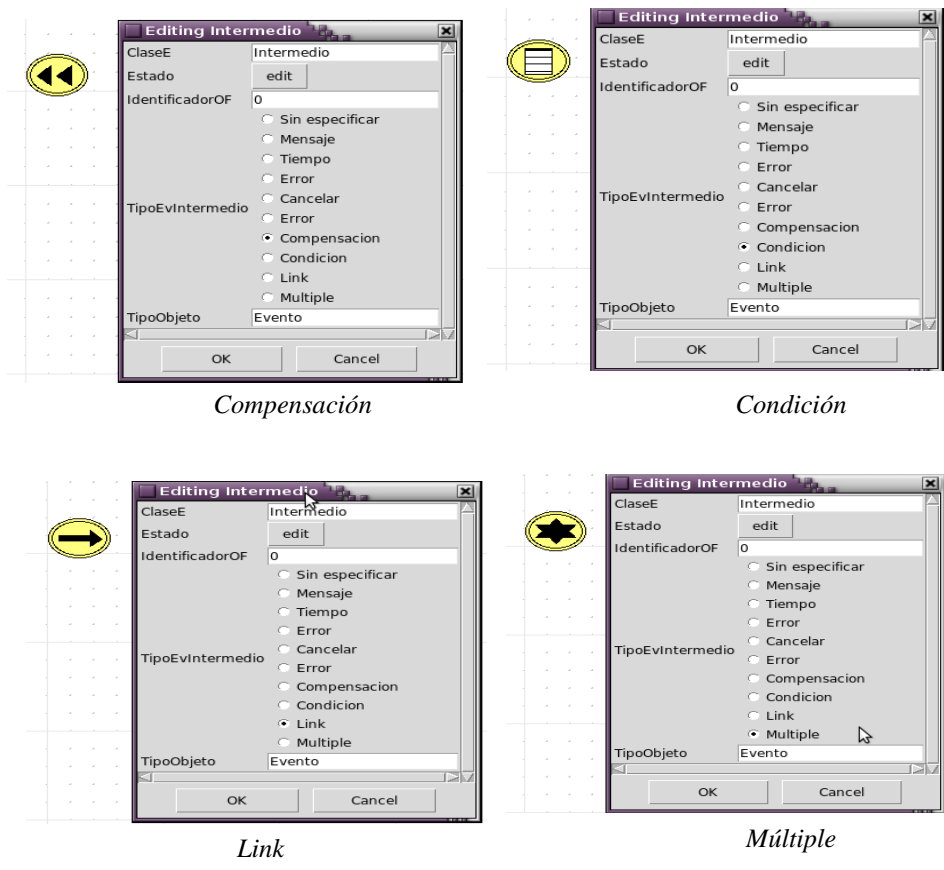


Mensaje

Tiempo

Error

Cancelar



5) **Evento final:** En la Figura B.10 se puede observar cómo se representa gráficamente un evento final en AToM<sup>3</sup>, y sus atributos.

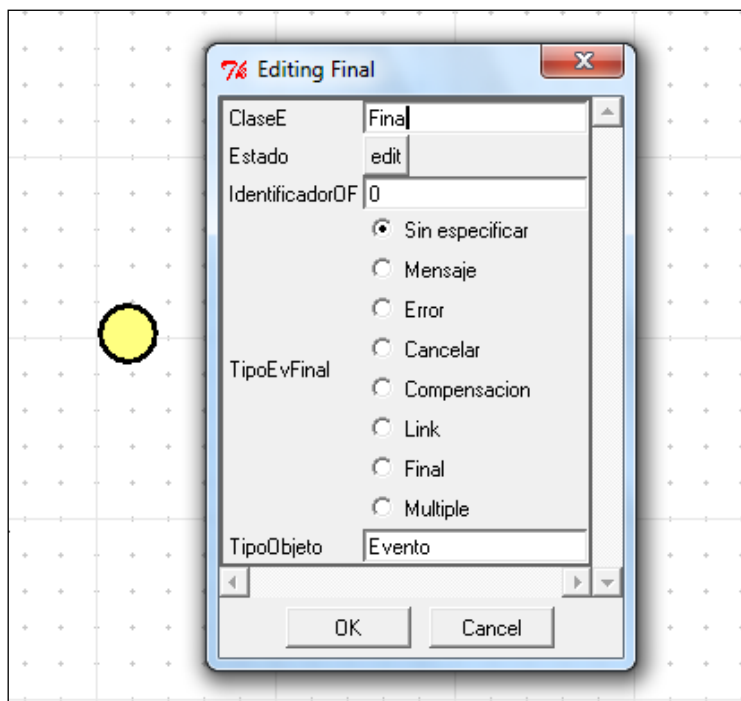


Figura B.10. Representación gráfica de un evento final y sus atributos.

Dependiendo del *tipo de evento final* agregan los iconos internos a un evento inicial como se puede observar en la Figura B.11

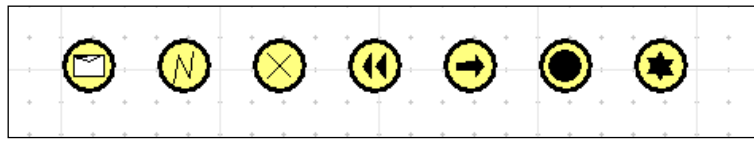
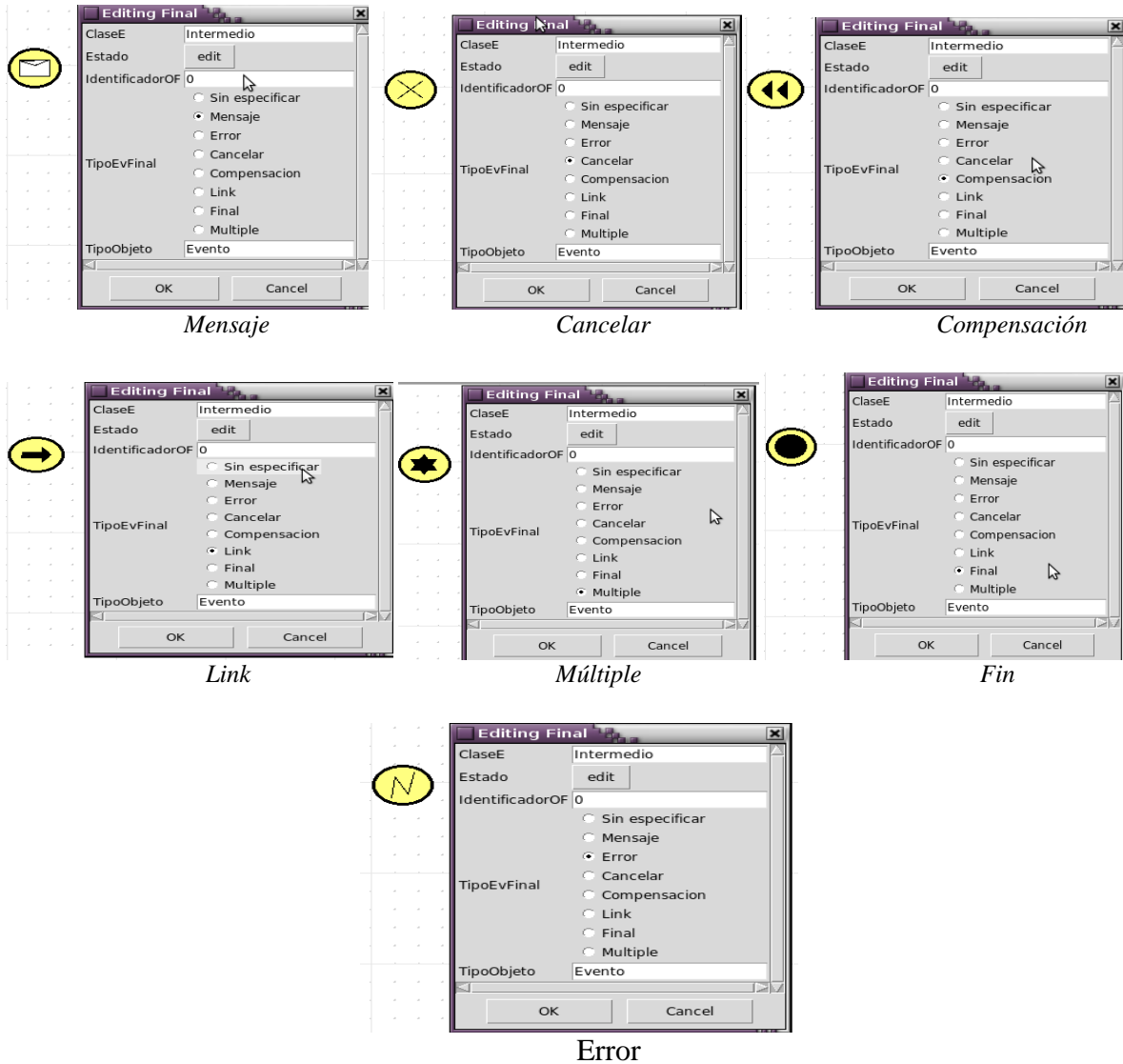


Figura B.11. Tipos de evento final.



- 6) **Gateway:** En la Figura B.12 se puede observar cómo se representa gráficamente un gateway en AToM<sup>3</sup>, y sus atributos.

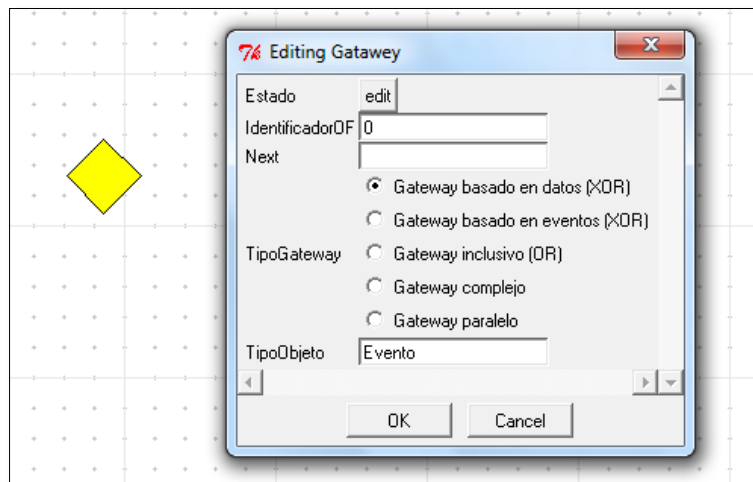


Figura B.12. Representación grafica de un gateway y sus atributos.

Dependiendo del *tipo de gateway* agregan los iconos internos a un gateway como se puede observar en la Figura B.13

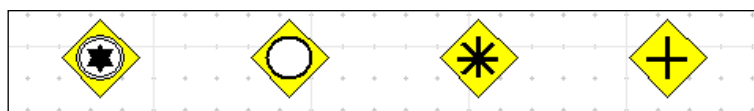
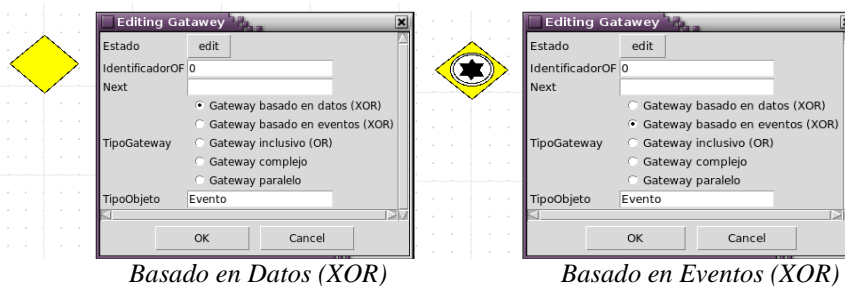


Figura B.13. Tipos de gateway.



- 7) **Objeto de dato:** En la Figura B.14 se puede observar cómo se representa gráficamente un objeto de dato en AToM<sup>3</sup>, y en la Figura B.14 se muestra sus propiedades.

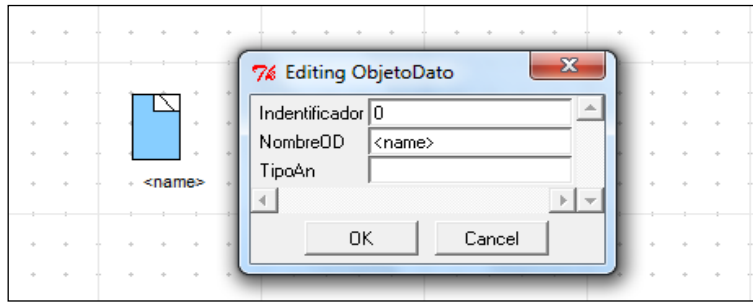


Figura B.14. Representación grafica de un objeto de dato y sus atributos

- 8) **Anotación textual:** En la Figura B.15 se puede observar cómo se representa gráficamente una anotación textual en AToM<sup>3</sup>, y sus atributos.

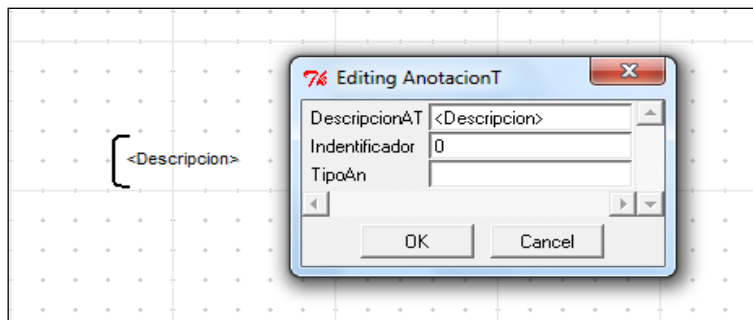


Figura B.15. Representación grafica de una anotación textual y sus atributos.

- 9) **Pool:** En la Figura B.16 se puede observar cómo se representa gráficamente un *pool* en AToM<sup>3</sup>, y sus atributos.

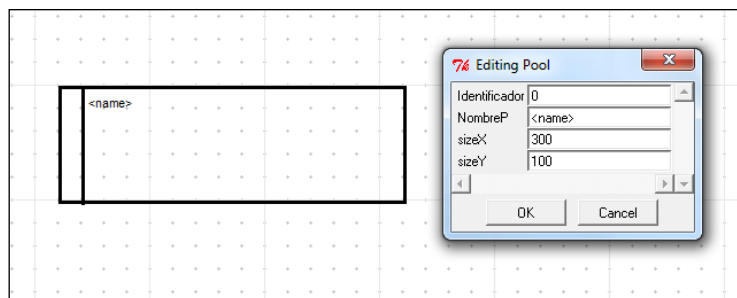


Figura B.16. Representación grafica de un pool y sus atributos.

- 10) **Lane:** En la Figura B.17 se puede observar cómo se representa gráficamente un *Lane* en AToM<sup>3</sup>, y sus atributos.

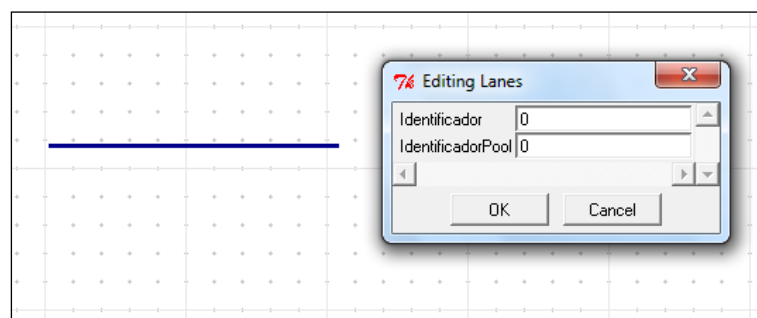


Figura B.17. Representación grafica de un lane y sus atributos.

**11) Flujo de asociación:** En la Figura B.18 se puede observar cómo se representa gráficamente un flujo de asociación en AToM<sup>3</sup>, y sus atributos.

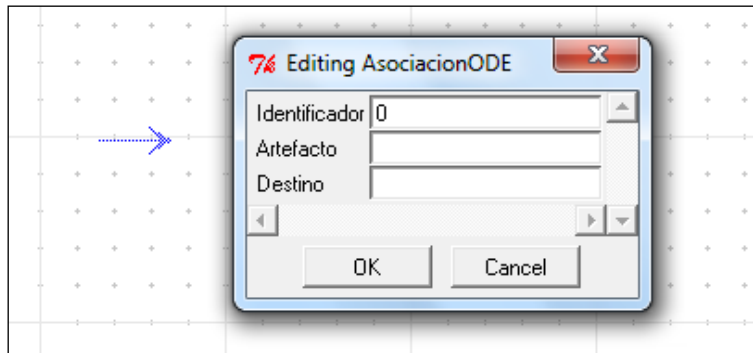


Figura B.18. Representación grafica de un flujo de asociación y sus atributos.

**12) Flujo de anotación:** En la Figura B.19 se puede observar cómo se representa gráficamente un flujo de anotación en AToM<sup>3</sup>, y sus atributos.

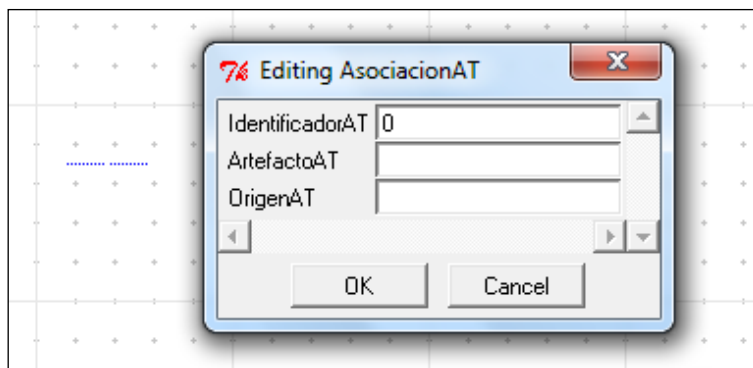


Figura B.19. Representación grafica de un flujo de anotación y sus atributos.

**13) Flujo de mensaje:** En la Figura B.20 se puede observar cómo se representa gráficamente un flujo de mensaje en AToM<sup>3</sup>, y sus atributos.

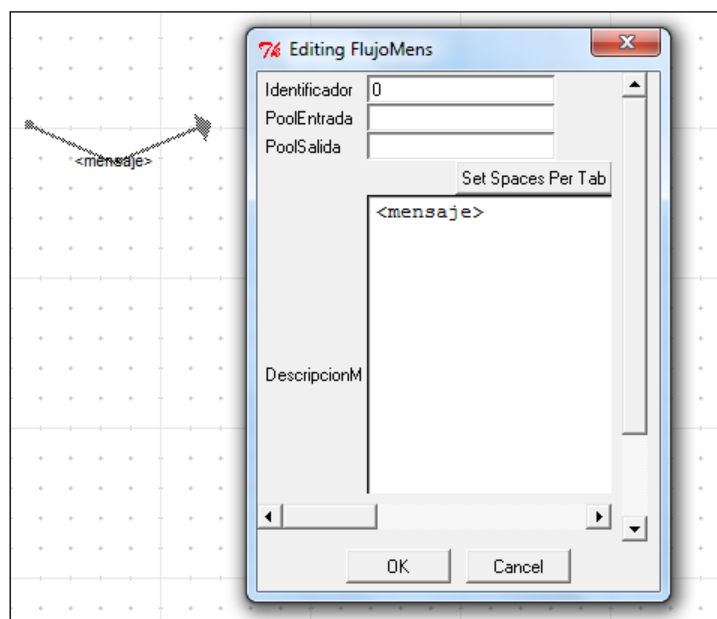
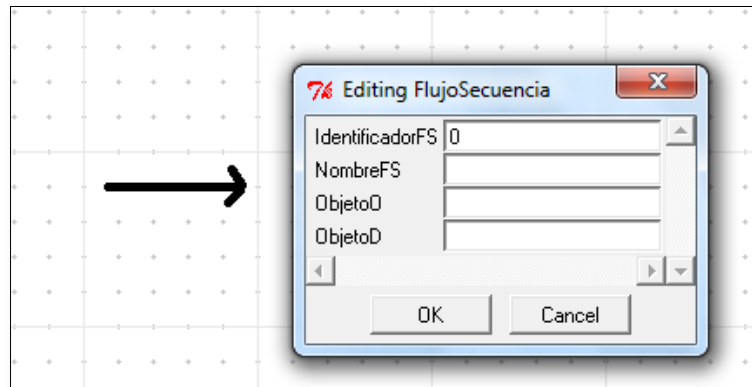


Figura B.20. Representación grafica de un flujo de mensaje y sus atributos.

**14) Flujo de secuencia:** En la Figura B.21 se puede observar cómo se representa gráficamente un flujo de secuencia en AToM<sup>3</sup>, y sus atributos.



*Figura B.21. Representación grafica de un flujo de secuencia y sus atributos.*

## VERIFICACION DEL METAMODELO EN ATOM<sup>3</sup>

Para la verificación del metamodelo BPMN resultante se tienen en cuenta las relaciones permitidas en las Tablas 3.2 y 3.3 del capítulo III.

Como producto de la verificación se presentan las interfaces resultantes de las pruebas siguiendo el orden de las filas de las Tablas 3.2 y 3.3 en los casos donde existan errores sintácticos y semánticos de correspondencia entre los elementos la relación muestra un cartel con el mensaje “no valido”, caso contrario no se observa ningún mensaje, dándose por aceptada la relación.

La correspondencia entre objetos de flujo mediante secuencia de flujo

### V2: Tarea – Subproceso

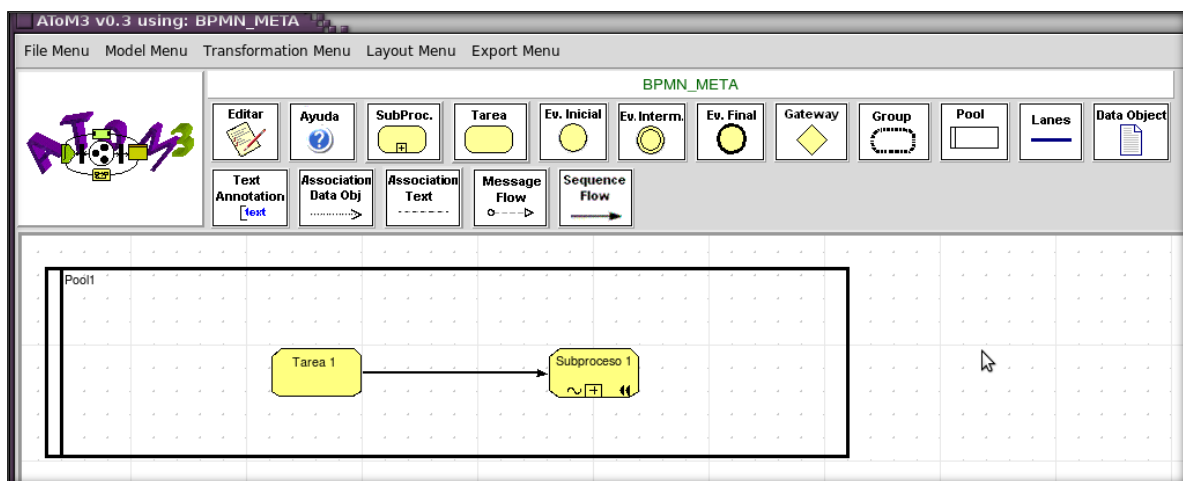


Figura C.1: Conexión de Tarea a Subproceso.

### V4: Tarea – Evento Intermedio

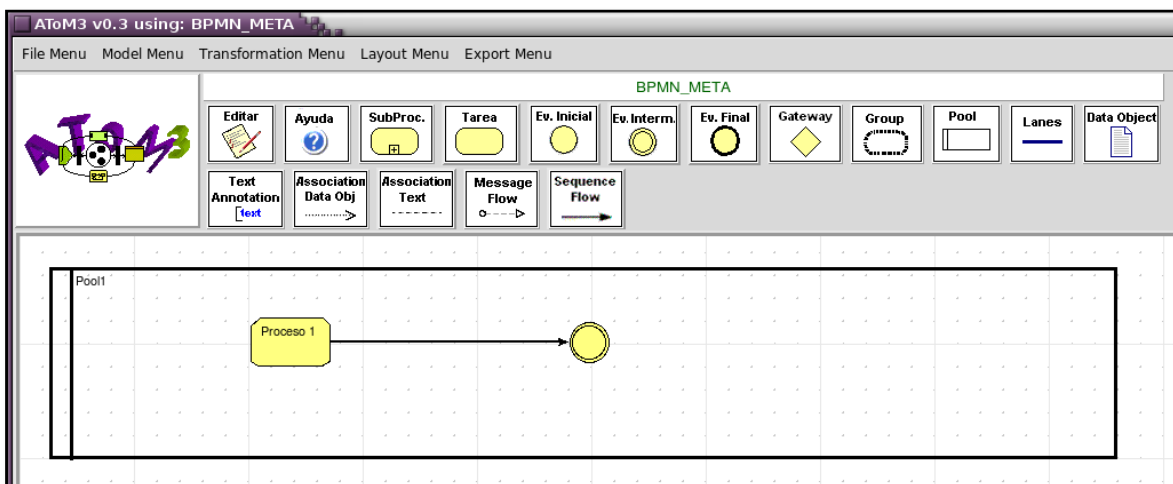


Figura C.2: Conexión de Tarea con Evento Final.

## V5: Tarea – Gateway

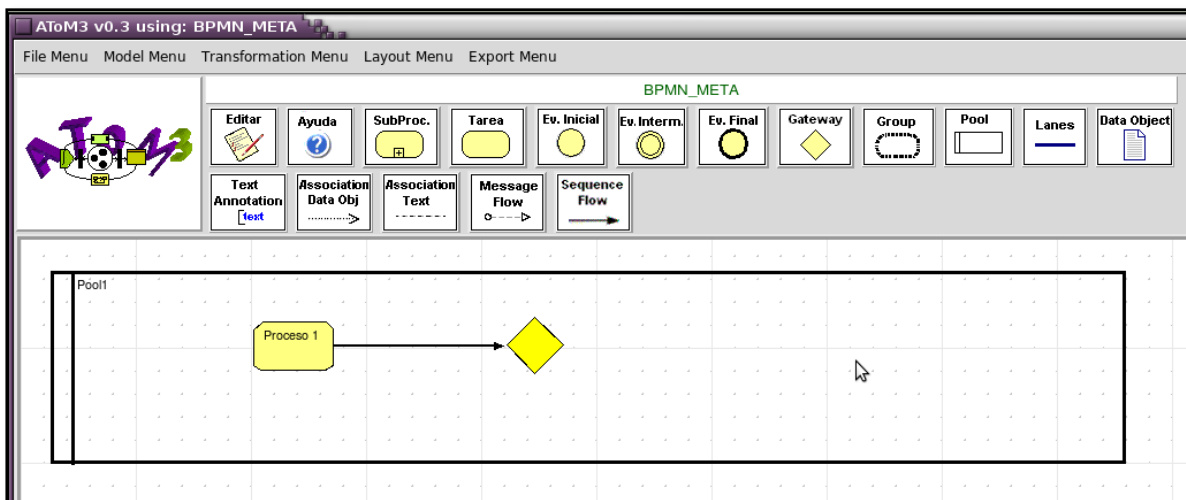


Figura C3: Conexión de Tarea con Gateway

## V6: Tarea – Evento Final

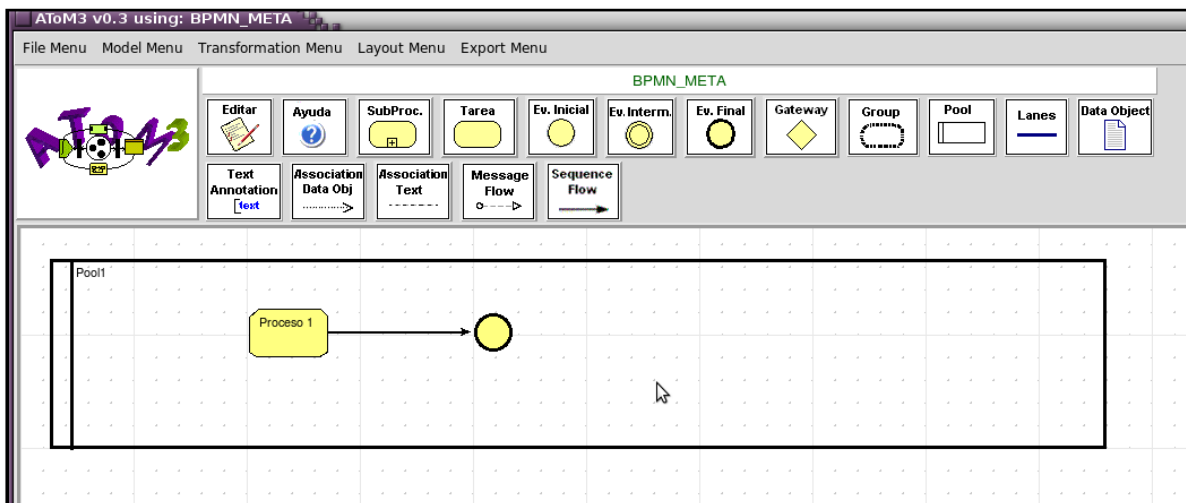


Figura C.4: Conexión de Tarea con Gateway

## V7: Subproceso – Tarea

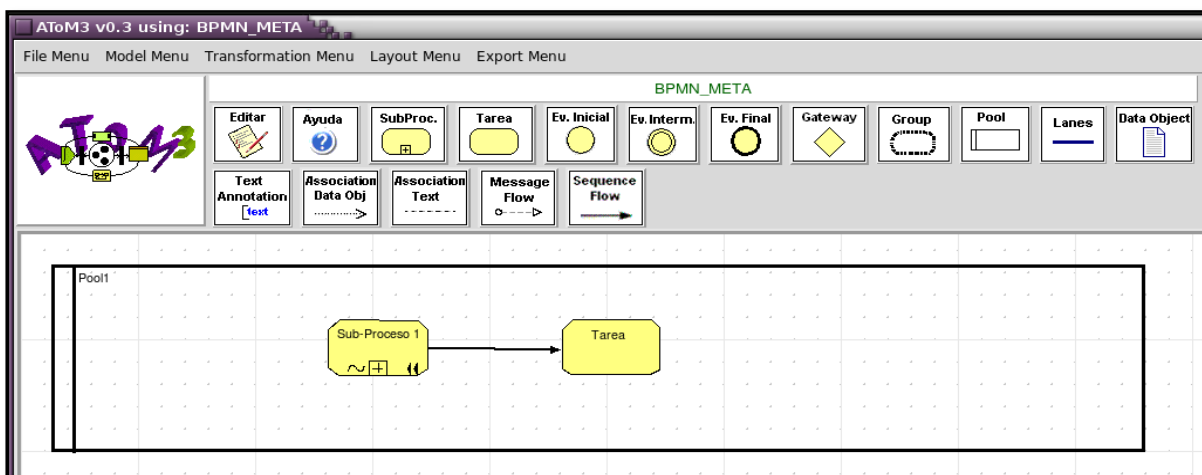


Figura C.5: Conexión de Subproceso con Gateway

## V8: Subproceso – Subproceso

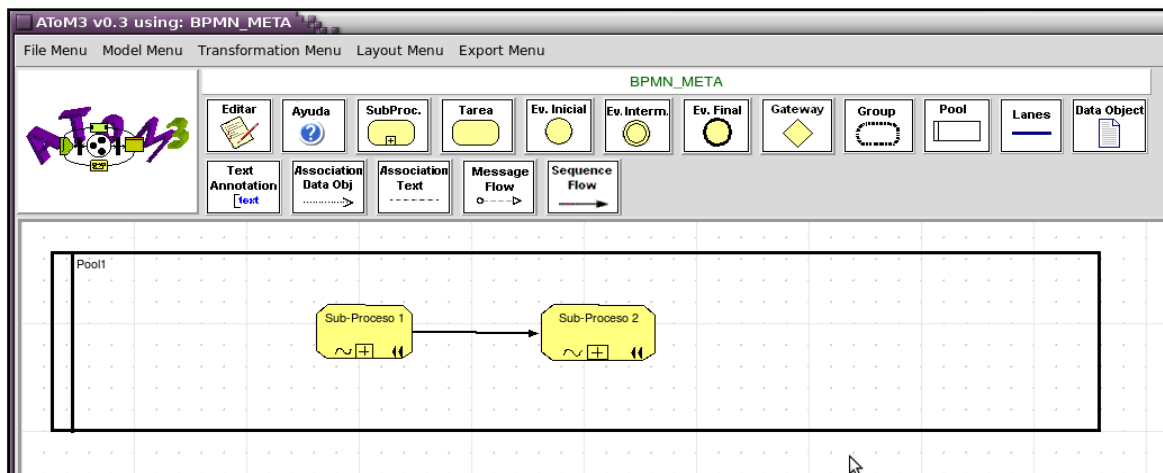


Figura C.6: Conexión de Subproceso a Subproceso

## V9: Subproceso – Evento Inicial

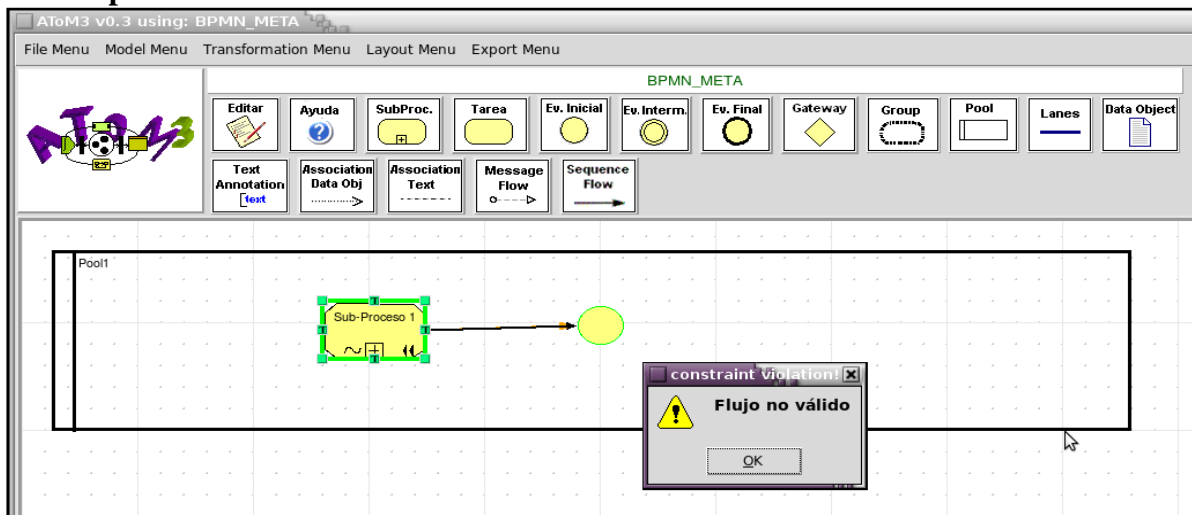


Figura C.7: Conexión de Subproceso a Subproceso

## V10: Subproceso – Evento Intermedio

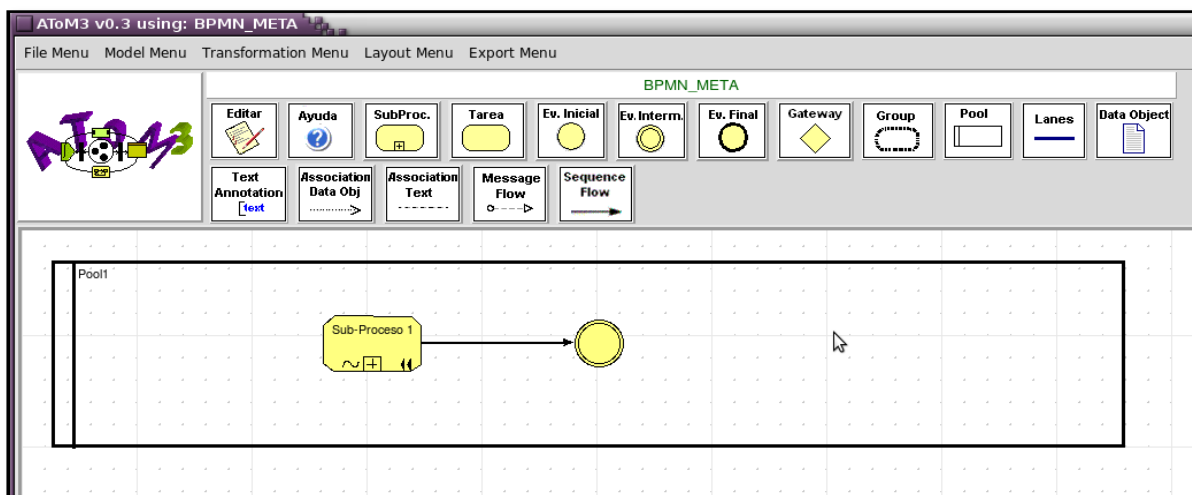


Figura C.8: Conexión de Subproceso a Evento Intermedio

## V11: Subproceso – Gateway

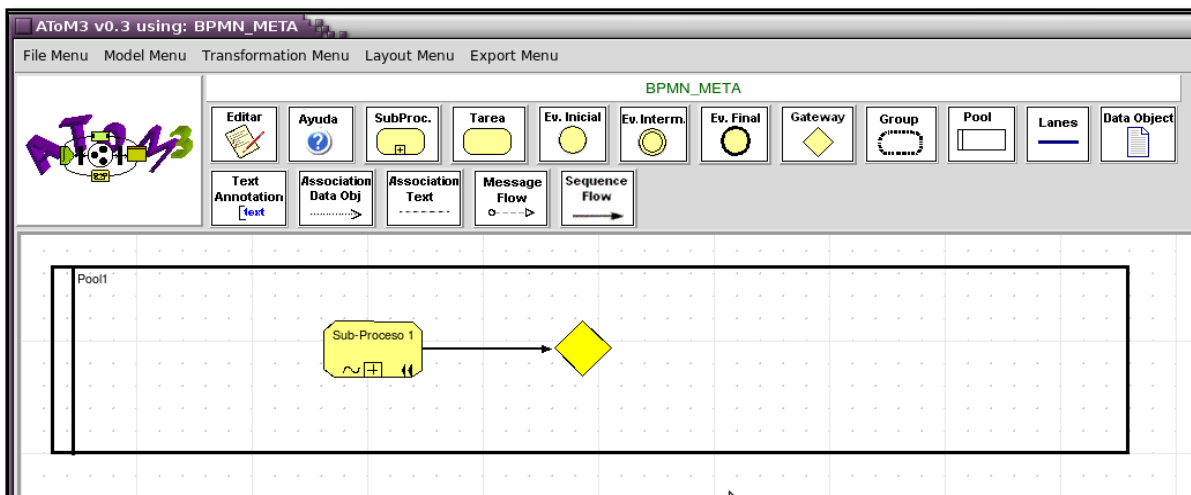


Figura C9: Conexión entre Subproceso a Gateway

## V12: Subproceso – Evento Final

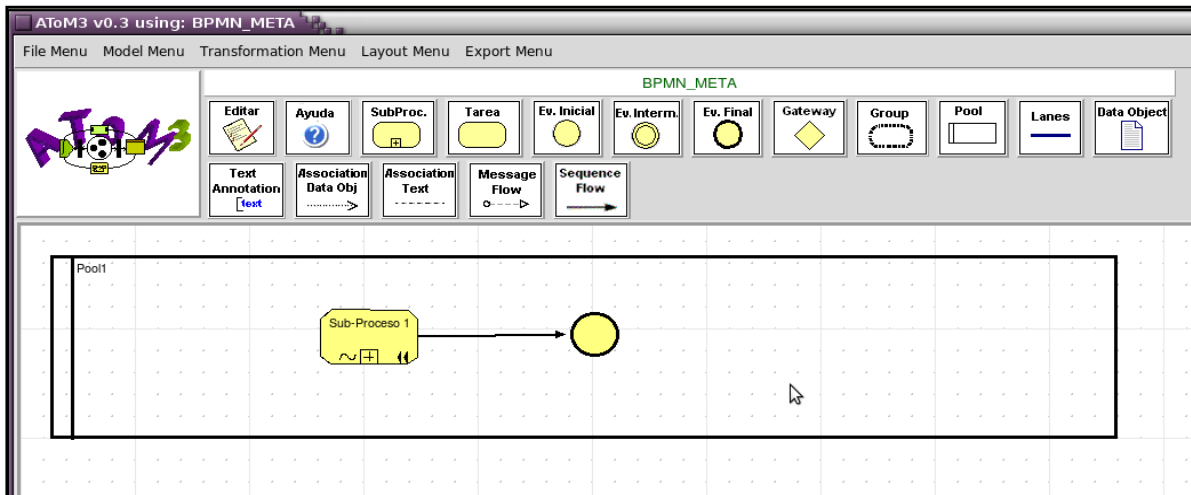


Figura C.10: Conexión entre Subproceso a Evento Final

## V13: Evento Inicial – Tarea

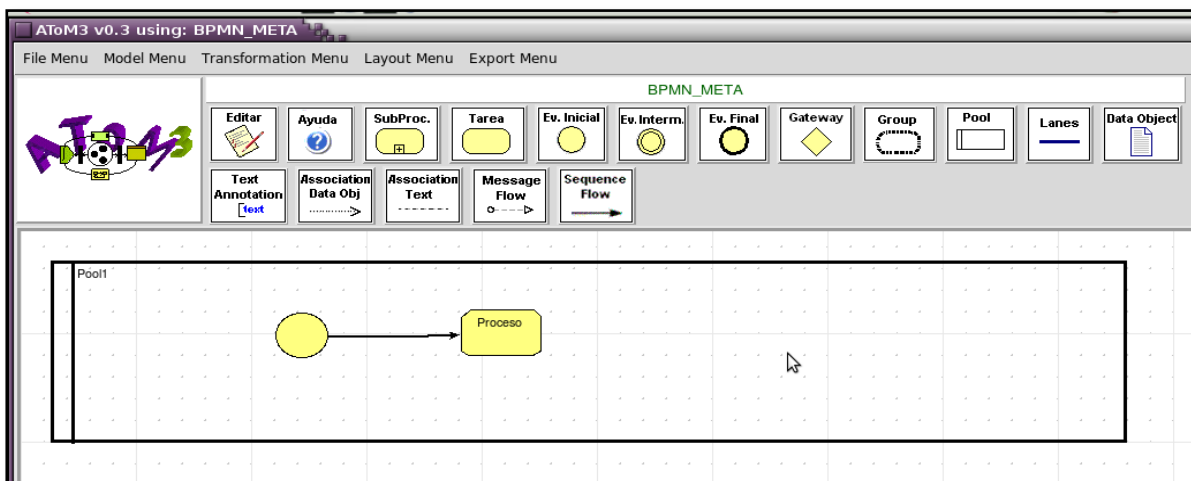


Figura C.11: Conexión entre Evento Inicial a Tarea.

## 14: Evento Inicial – Subproceso

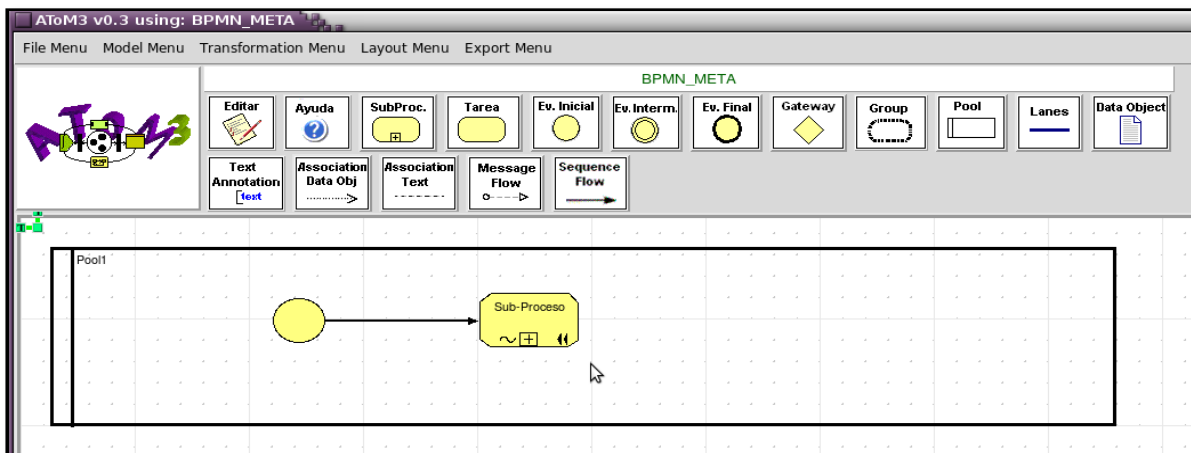


Figura C.12: Conexión entre Evento Inicial a Tarea

## V15: Evento Inicial – Evento Inicial

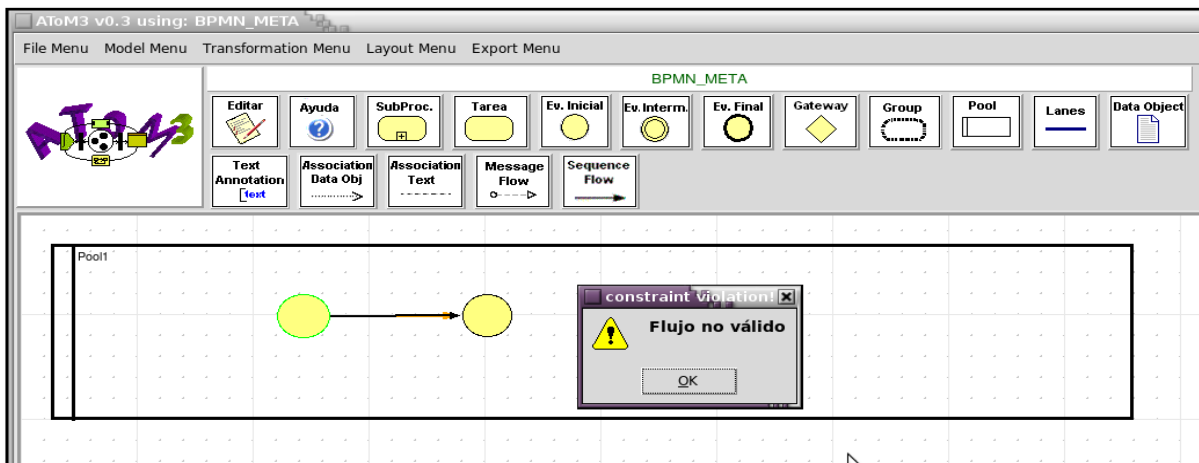


Figura C.13: Conexión entre Evento Inicial a Evento Inicial

## V16: Evento Inicial – Evento Intermedio

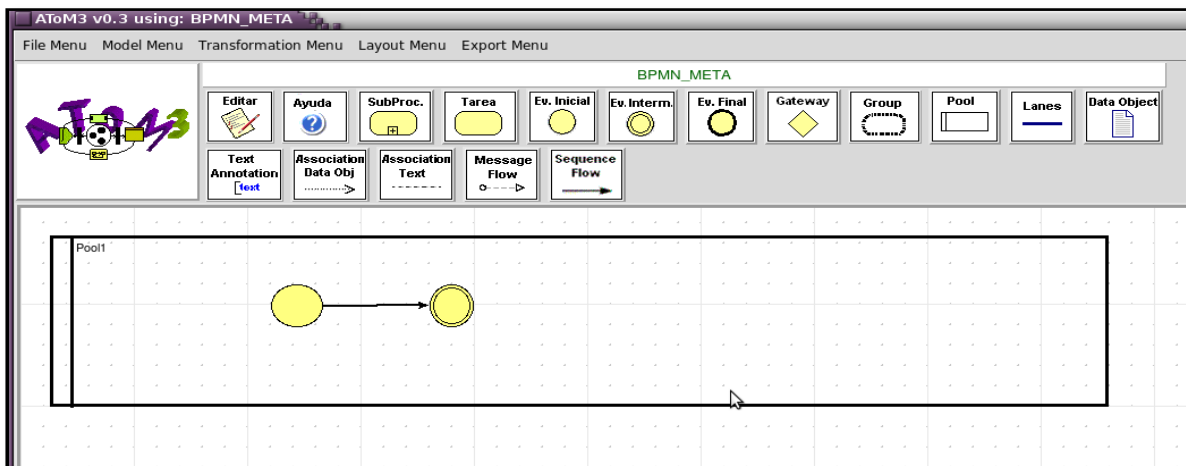


Figura C.14: Conexión entre Evento Inicial a Evento Intermedio

## V17: Evento Inicial – Gateway

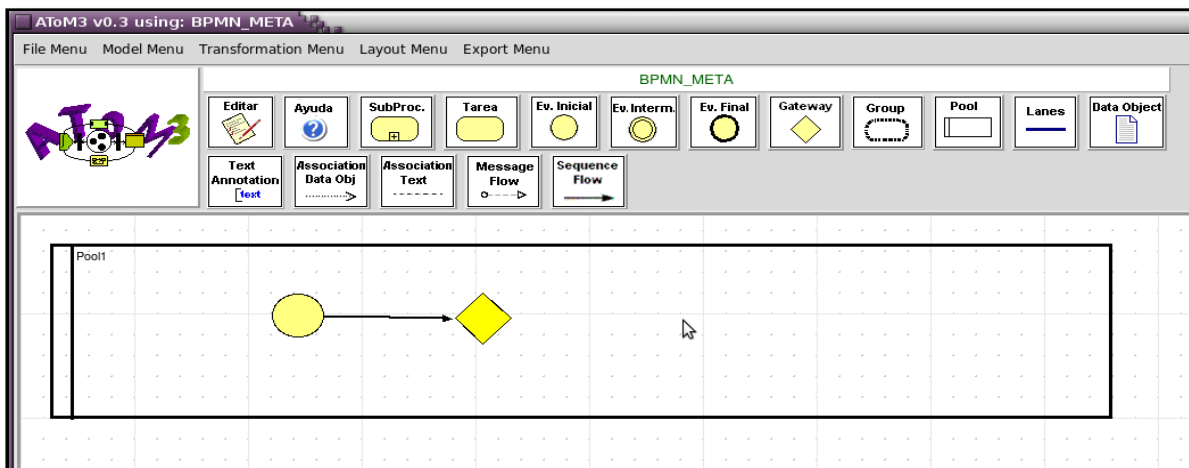


Figura C.15: Conexión entre Evento Inicial a Gateway Intermedio

## V18: Evento Inicial – Evento final

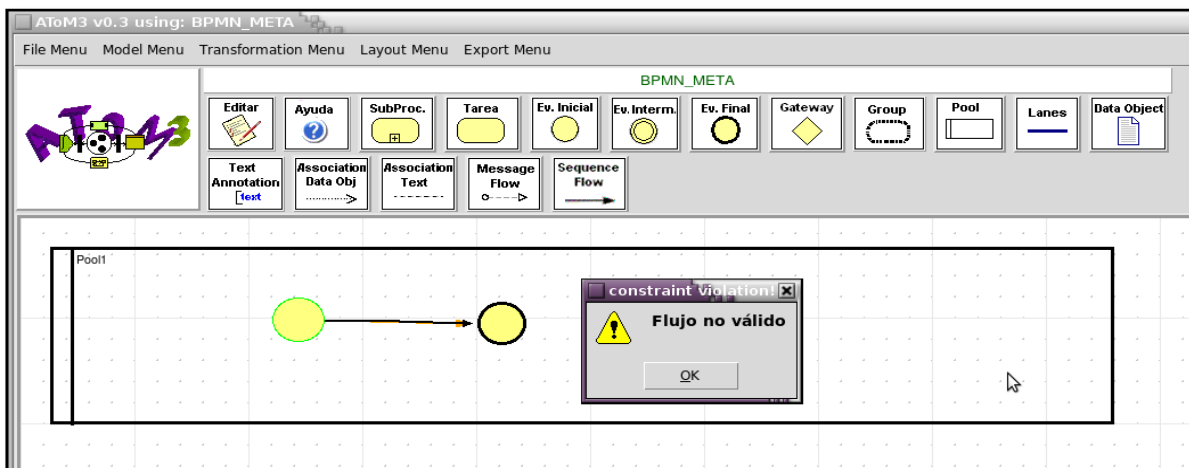


Figura C.16: Conexión entre Evento Inicial a Gateway Intermedio

## V19: Evento Intermedio – Tarea

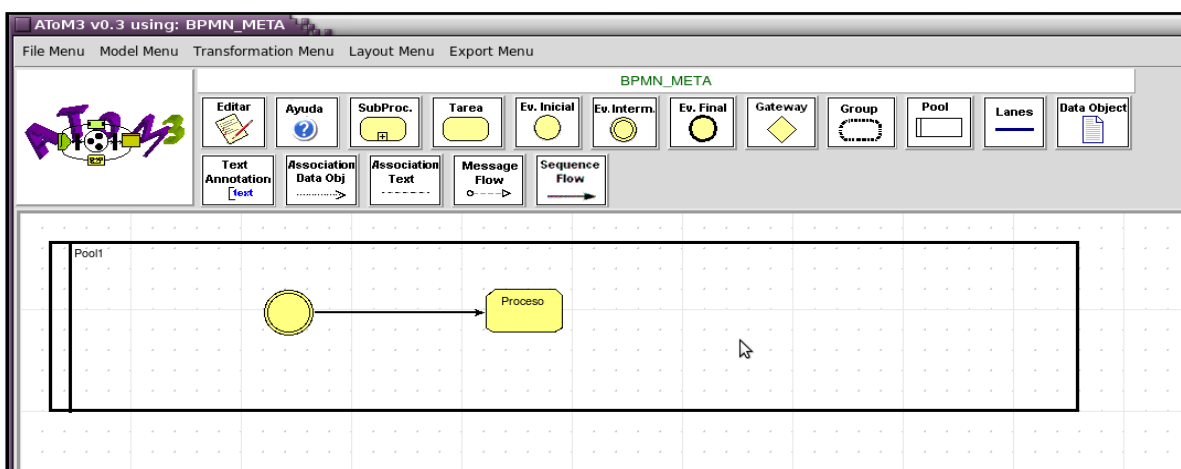


Figura C.17: Conexión entre Evento Intermedio a Tarea Gateway Intermedio

## V20: Evento Intermedio – subprocesso

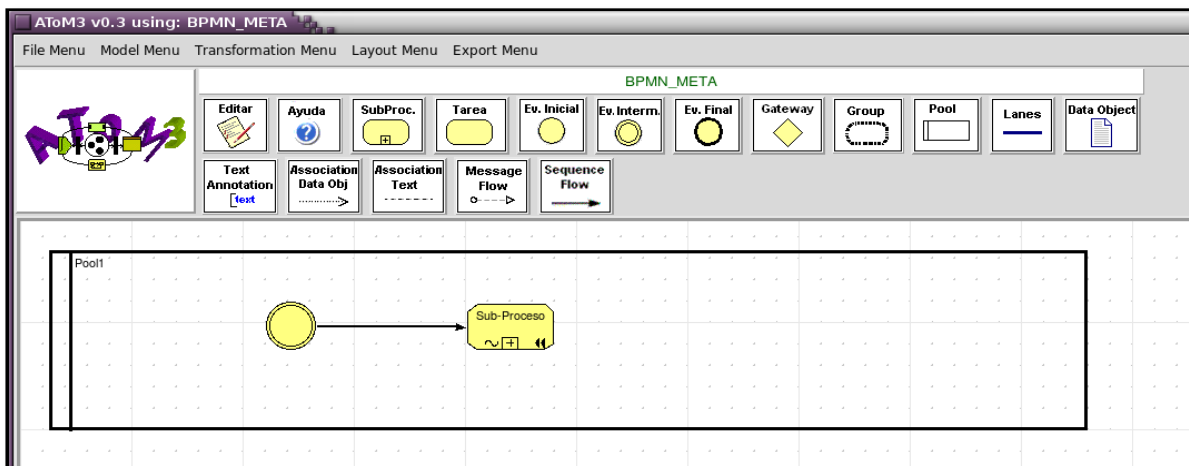


Figura C.18: Conexión entre Evento Intermedio a Subproceso

## V21: Evento Intermedio – Evento Inicial

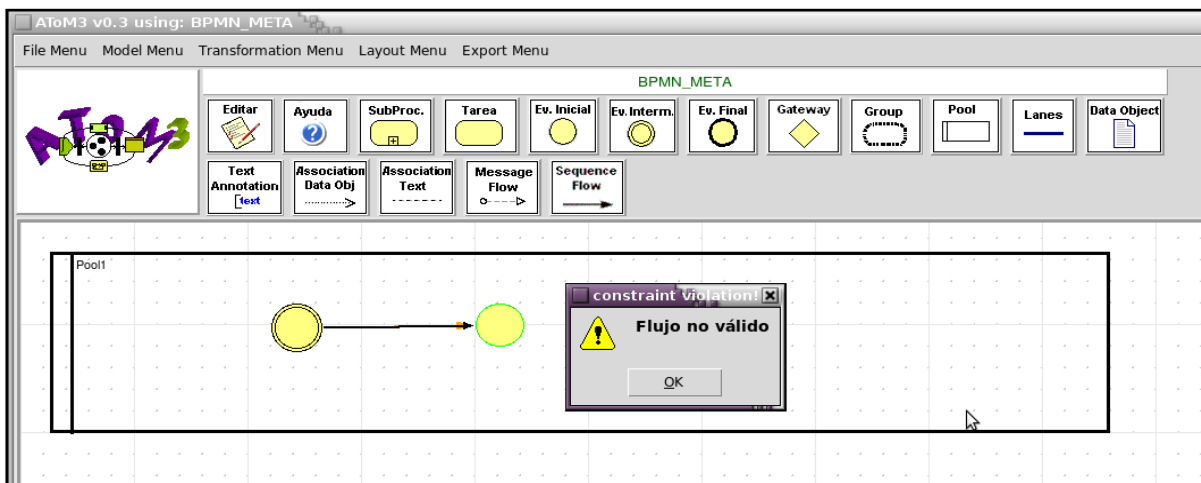


Figura C.19: Conexión entre Evento Intermedio a Evento Inicial

## V22: Evento Intermedio – Evento Intermedio

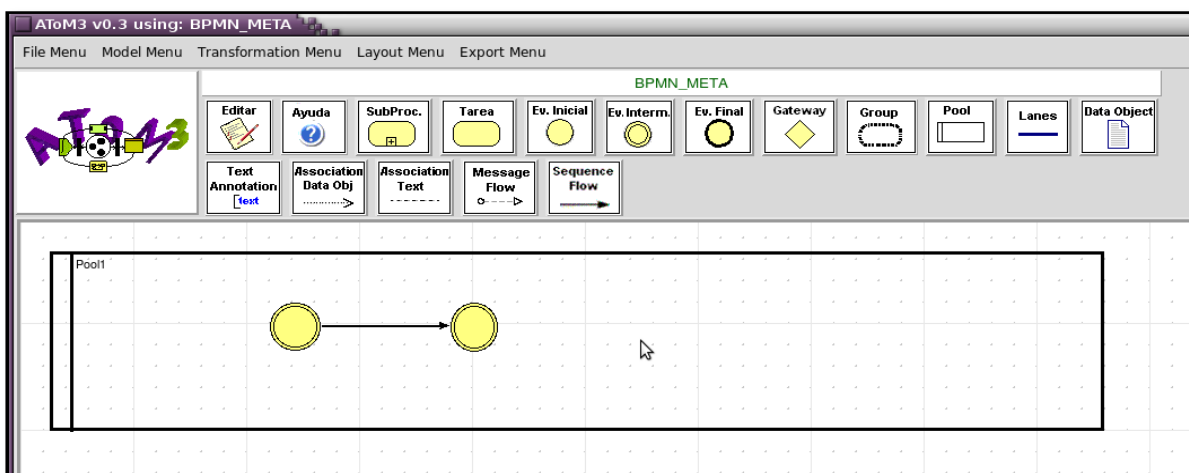


Figura C.20: Conexión entre Evento Intermedio a Evento Intermedio

## V23: Evento Intermedio – Gateway

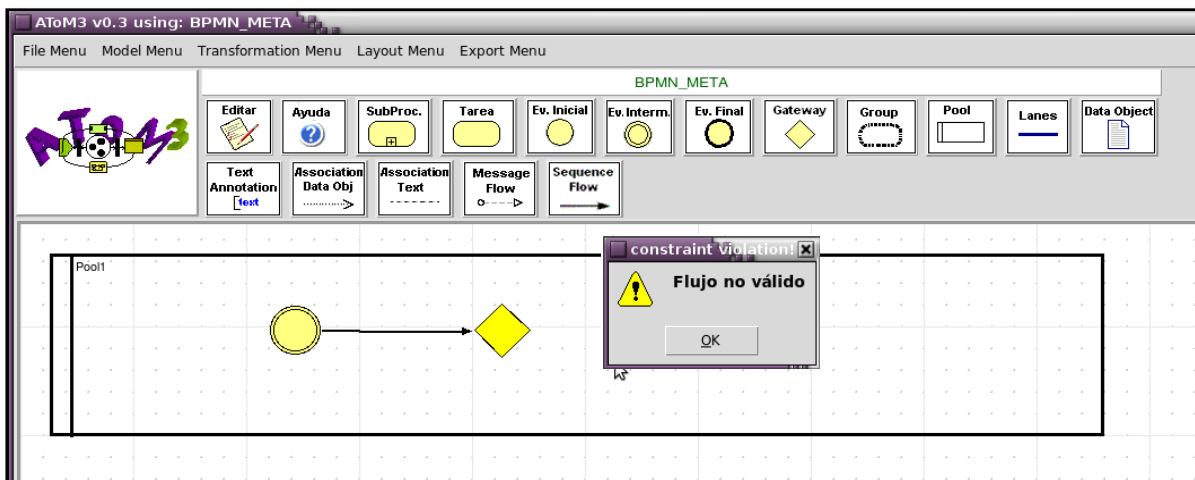


Figura C.21: Conexión entre Evento Intermedio a Gateway

## V24: Evento Intermedio – Evento Final

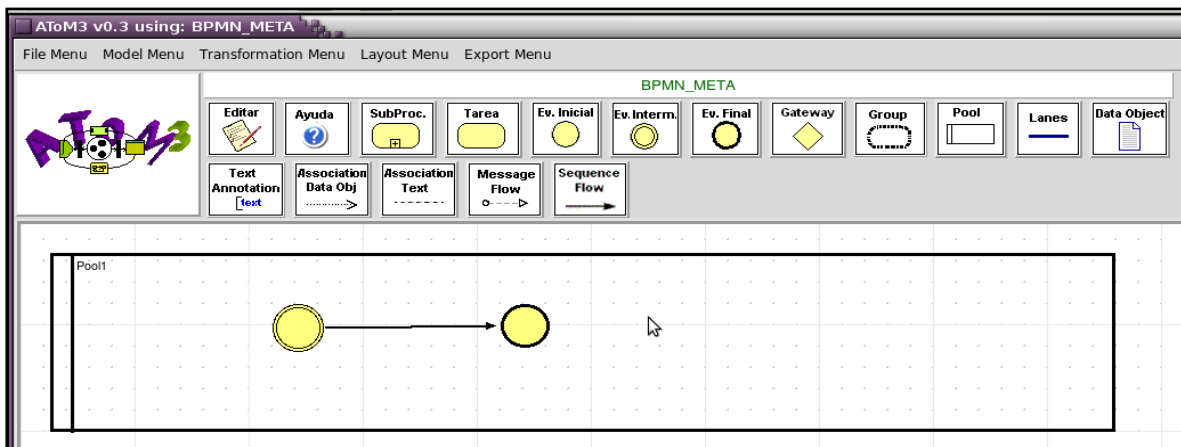


Figura C.22: Conexión entre Evento Intermedio a Evento Final

## V25: Evento Final – Tarea

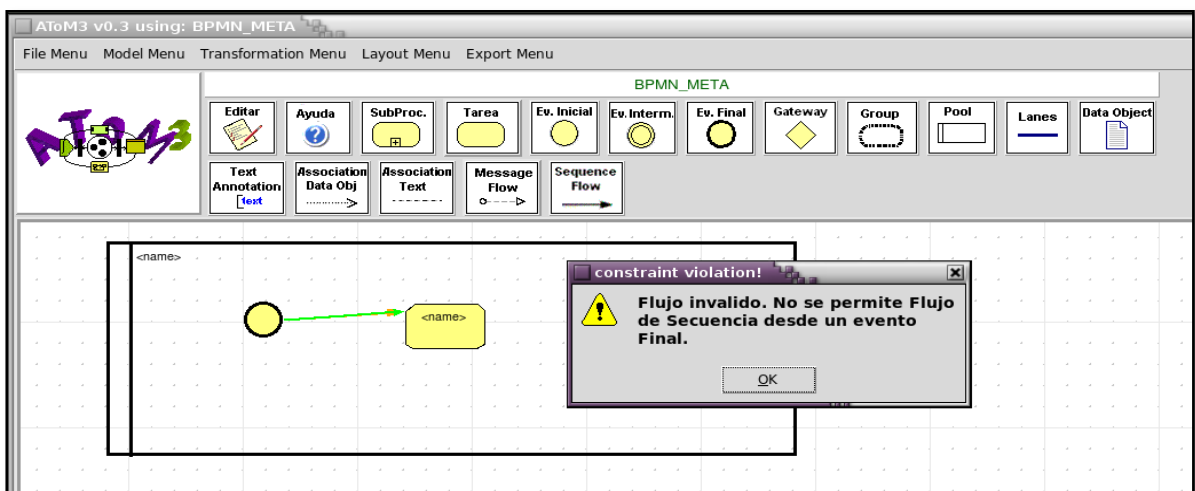


Figura C.23: Conexión entre Evento Final a Tarea

## V26: Evento Final – Subproceso

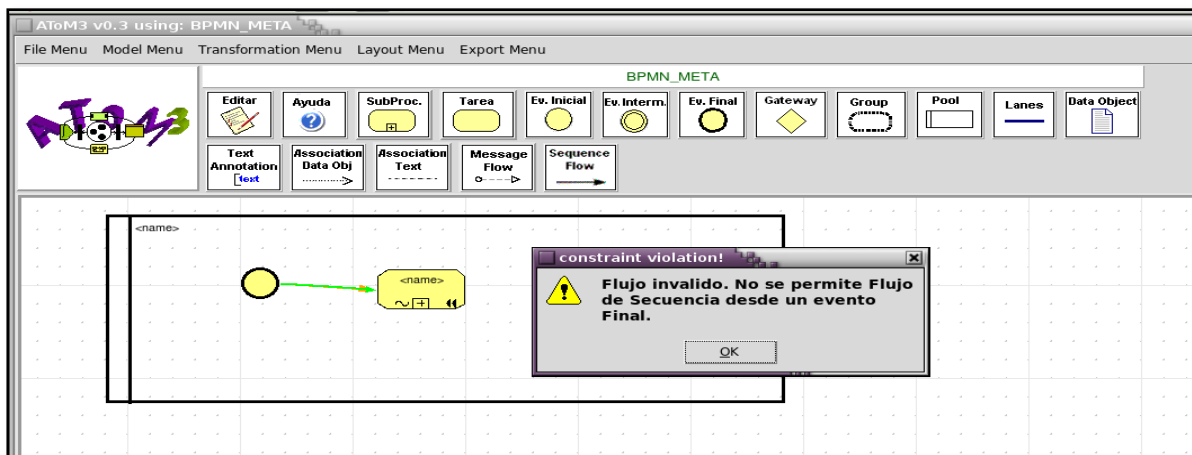


Figura C.24: Conexión entre Evento Final a Subproceso

## V27: Evento Final – Evento Inicial

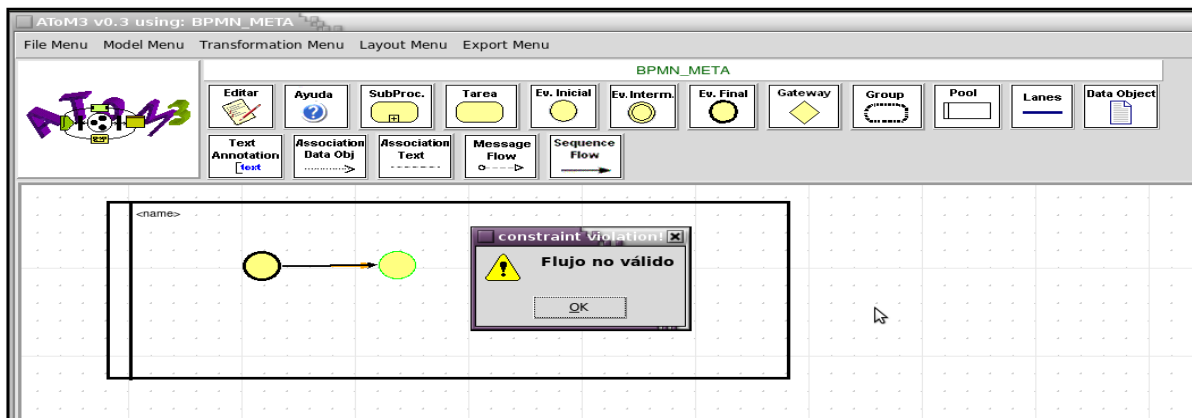


Figura C-25: Conexión entre Evento Final a Evento Inicial

## V28: Evento Final – Evento Intermedio

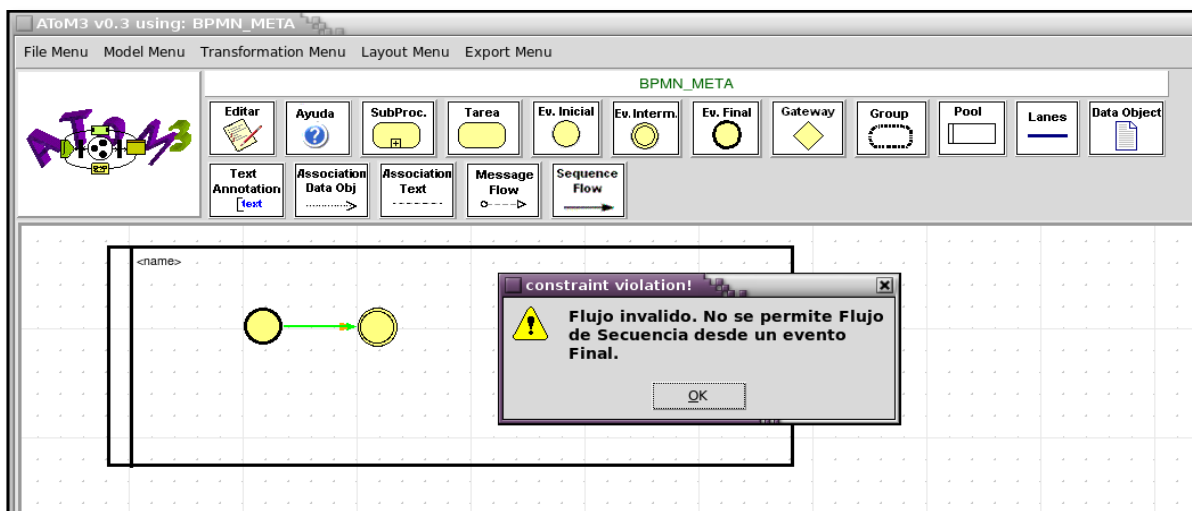


Figura C.26: Conexión entre Evento Final a Evento Intermedio

## V29: Evento Final – Gateway

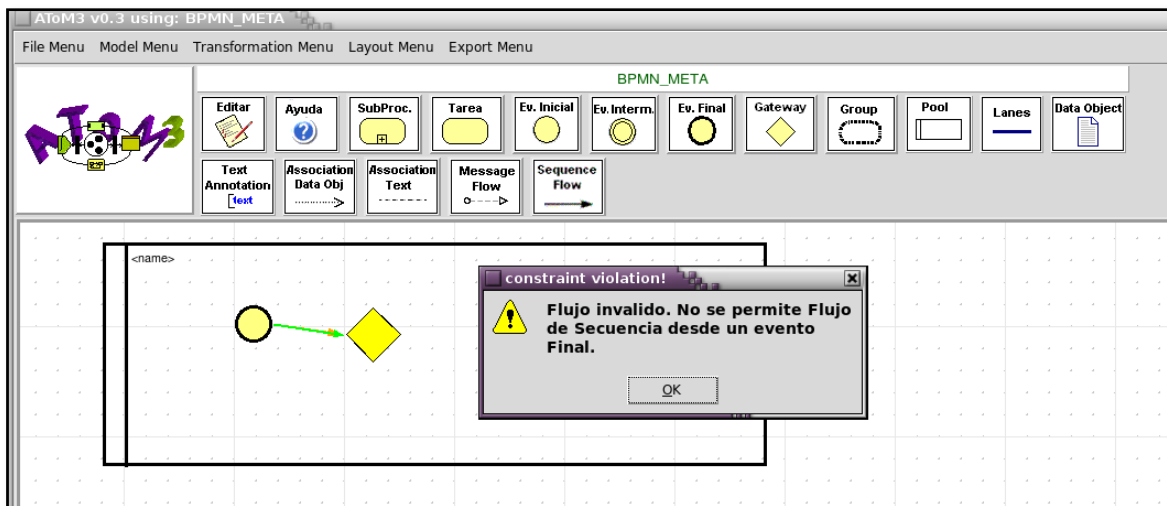


Figura C.27: Conexión entre Evento Final a Gateway.

## V30: Evento Final – Evento Final

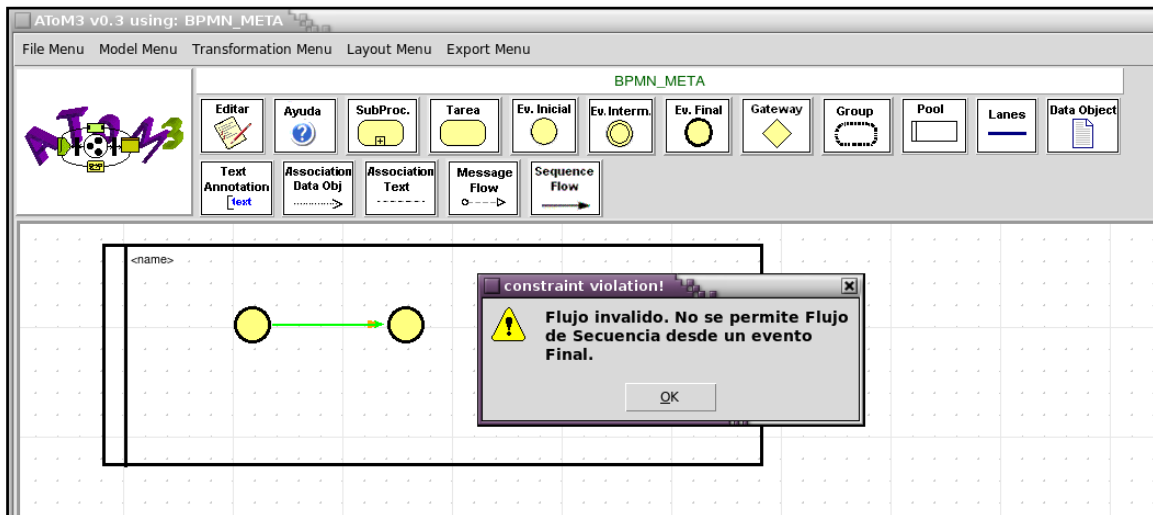


Figura C.28: Conexión entre Evento Final a Evento Final.

La correspondencia verificada de elementos relacionados mediante flujos de mensajes es:

### V31: Flujo de mensaje: Tarea - Tarea

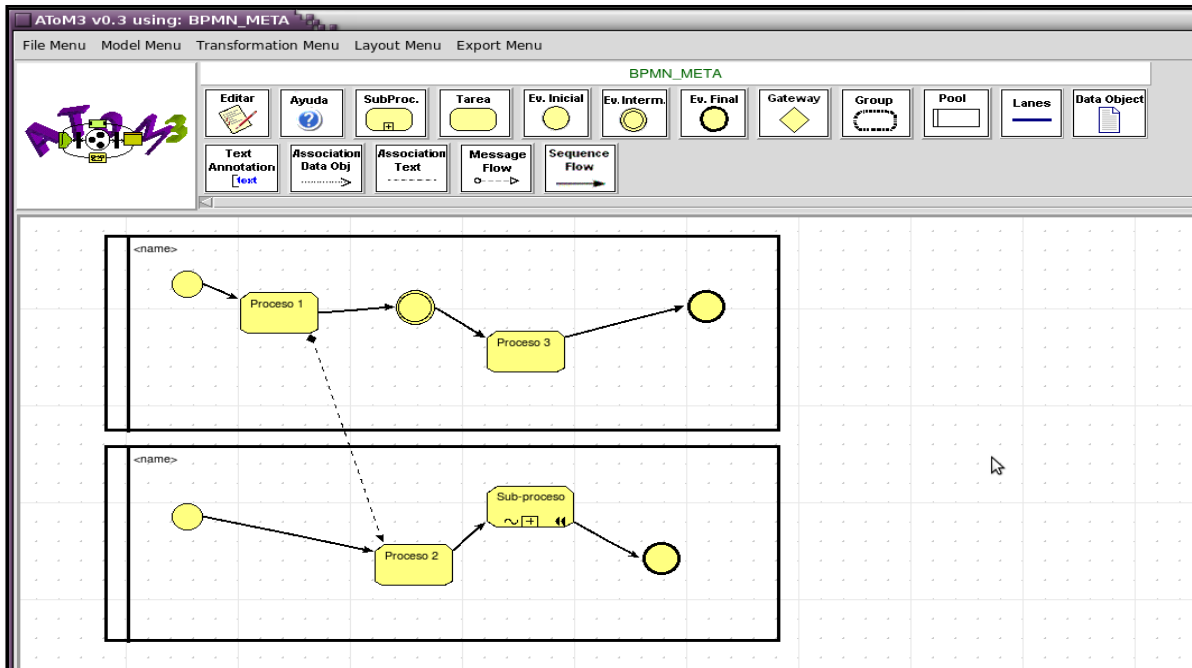


Figura C.29: Conexión entre flujo de mensaje de Tarea a Tarea.

### V32: Flujo de mensaje: Tarea – Subproceso

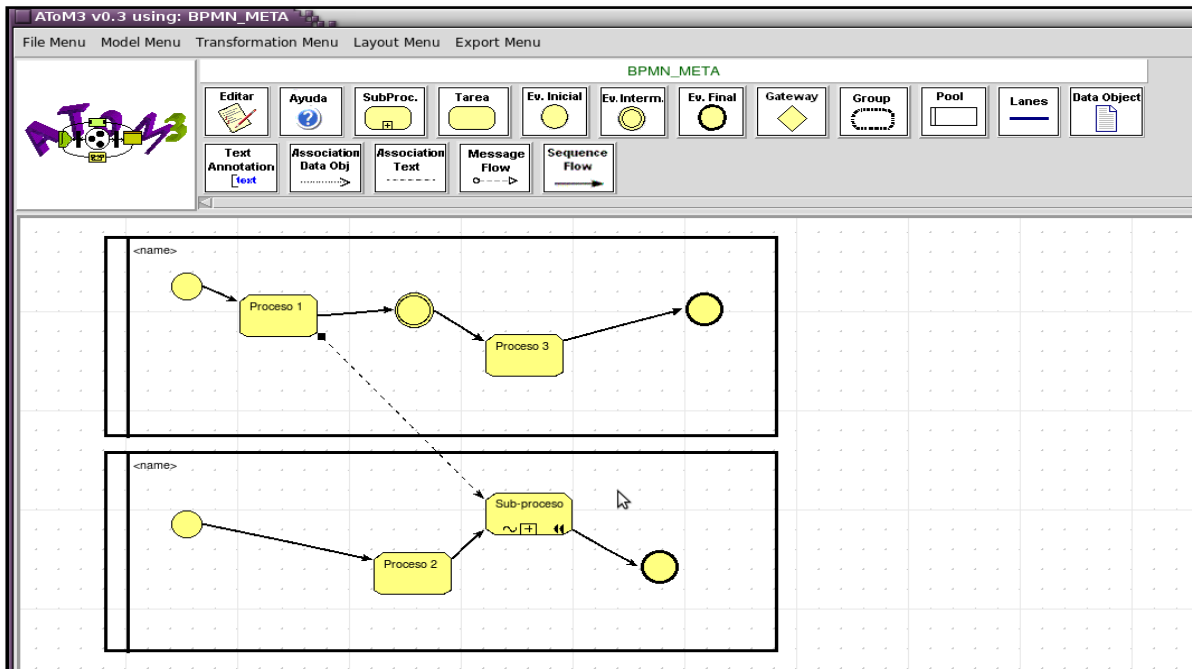


Figura C.30: Conexión entre flujo de mensaje de Tarea a Subproceso.

### V33: Flujo de mensaje: Tarea – Evento Inicial

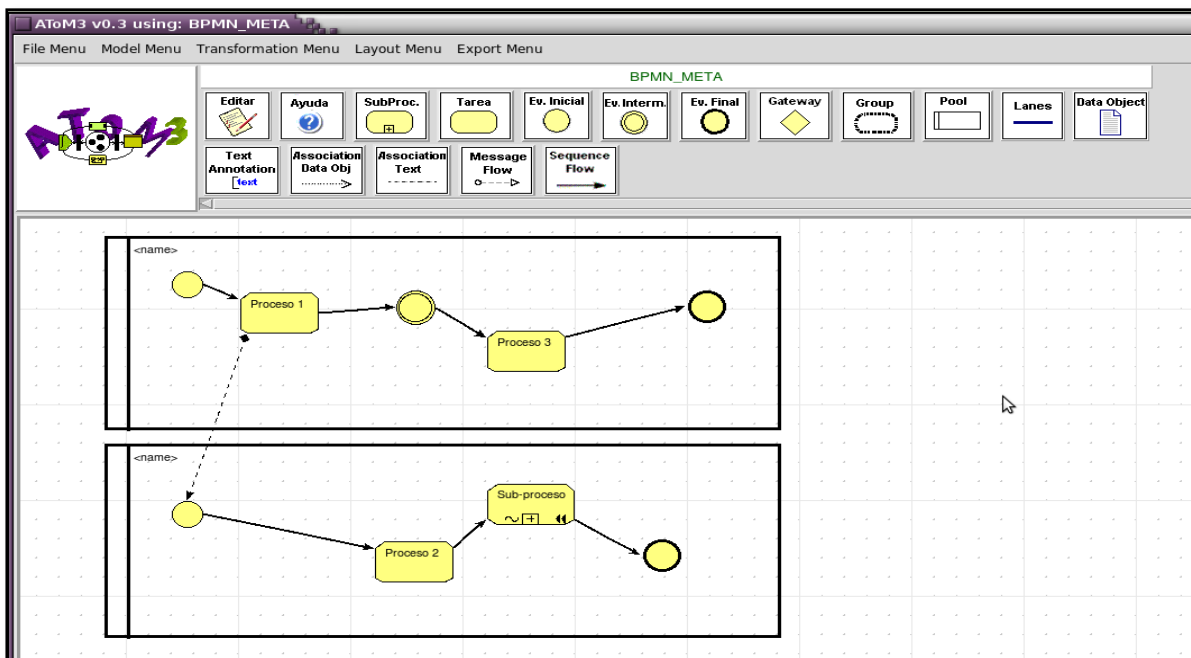


Figura C.31: Conexión entre flujo de mensaje de Tarea a Subproceso.

### V34: Flujo de mensaje: Tarea – Evento Intermedio

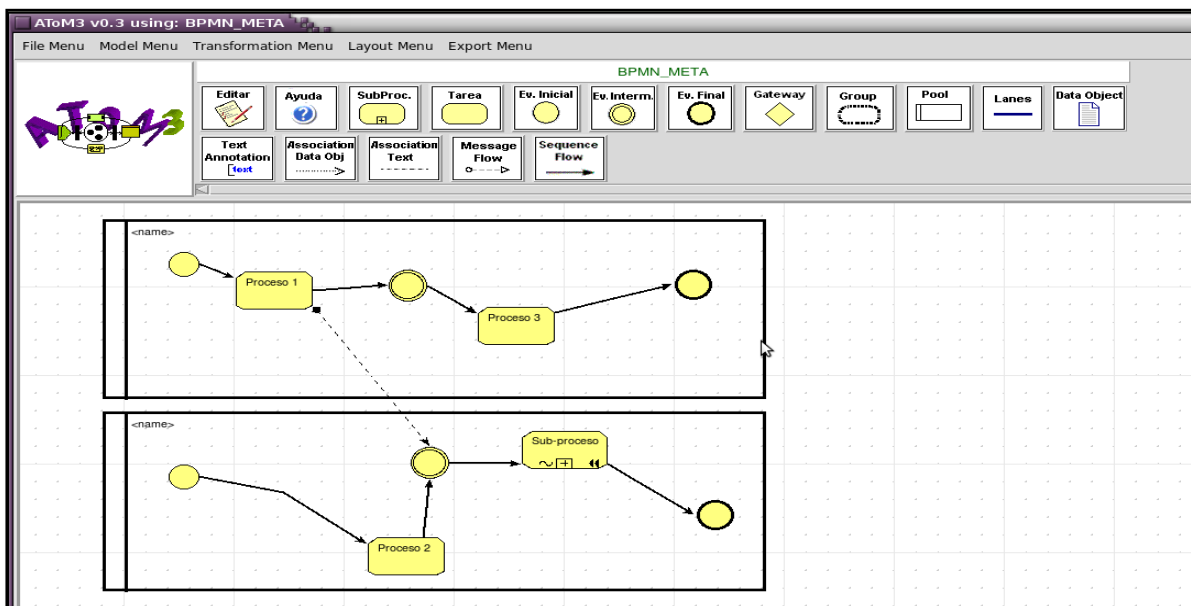


Figura C.32: Conexión entre flujo de mensaje de Tarea a Evento Intermedio

### V35: Flujo de mensaje: Tarea – Evento Final.

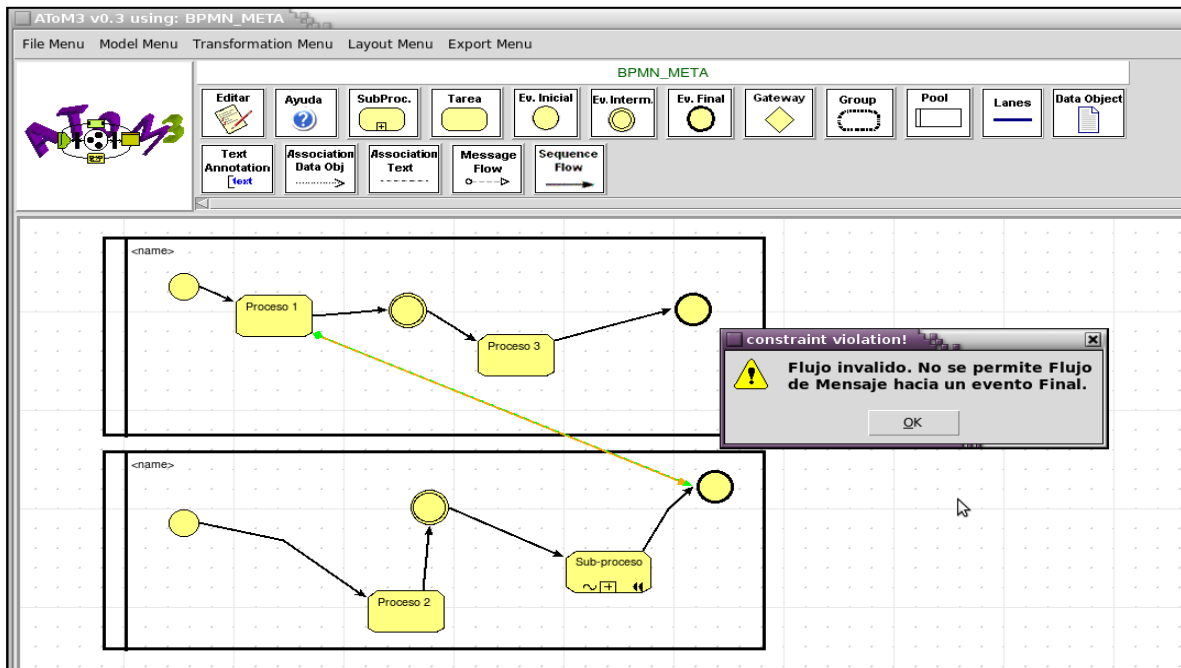


Figura C.33: Conexión entre flujo de mensaje de Tarea a Evento Final

### V36: Flujo de mensaje: Subproceso – Tarea

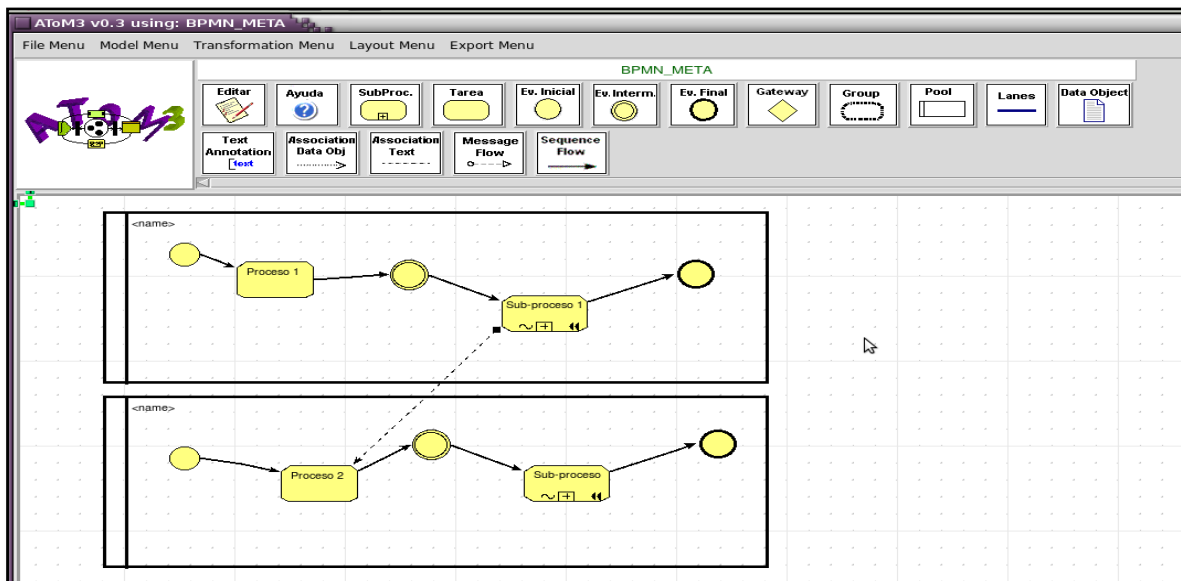


Figura C.34: Conexión entre flujo de mensaje de Subproceso a Tarea.

### V37: Flujo de mensaje: Subproceso – Subproceso

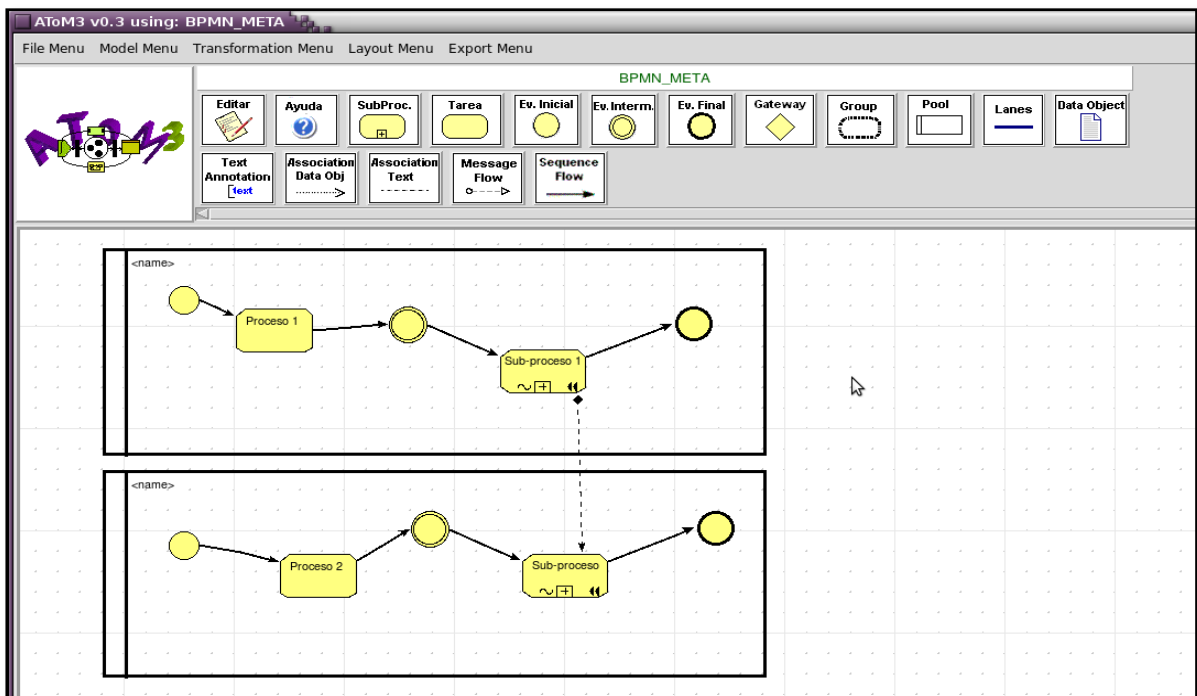


Figura C.35: Conexión entre flujo de mensaje de Subproceso a Tarea.

### V38: Flujo de mensaje: Subproceso – Evento Inicial

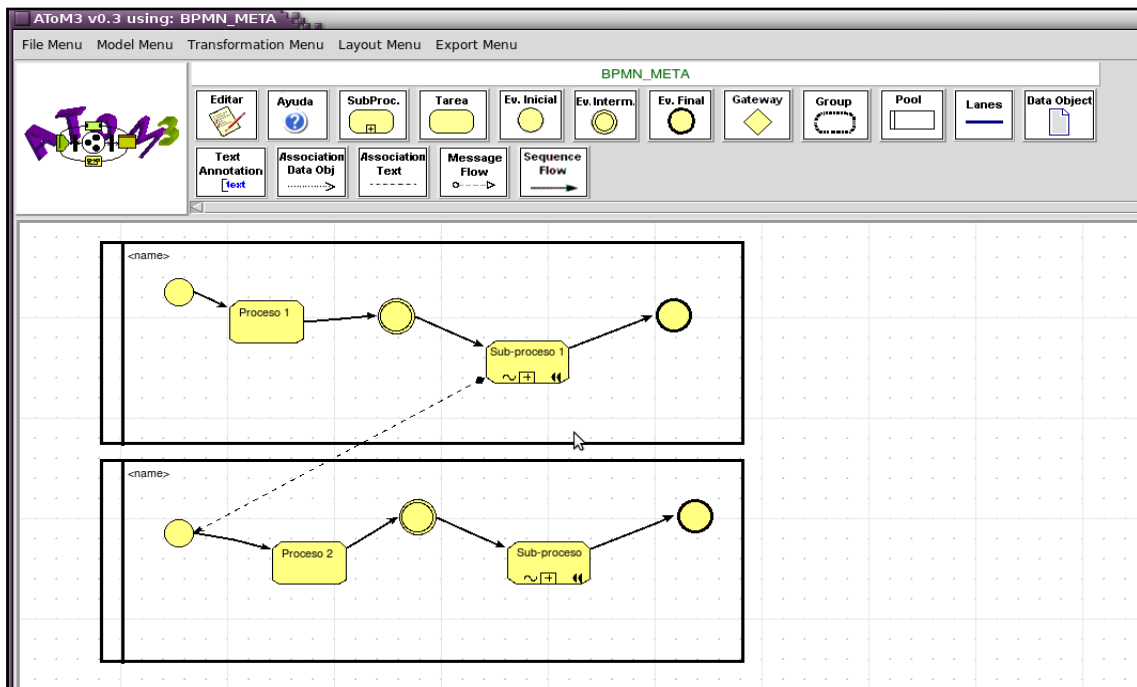


Figura C.36: Conexión entre flujo de mensaje de Subproceso a Evento Inicial.

### V39: Flujo de mensaje: Subproceso – Evento Intermedio

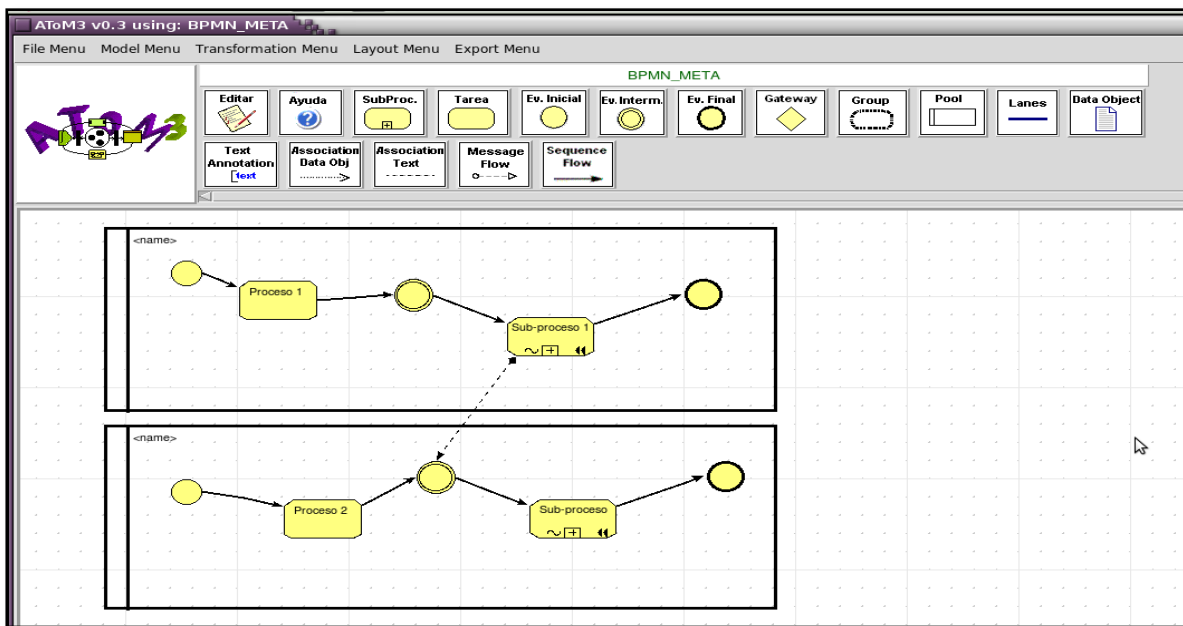


Figura C.37: Conexión entre flujo de mensaje de Subproceso a Evento Intermedio.

### V40: Flujo de mensaje: Subproceso – Evento Final

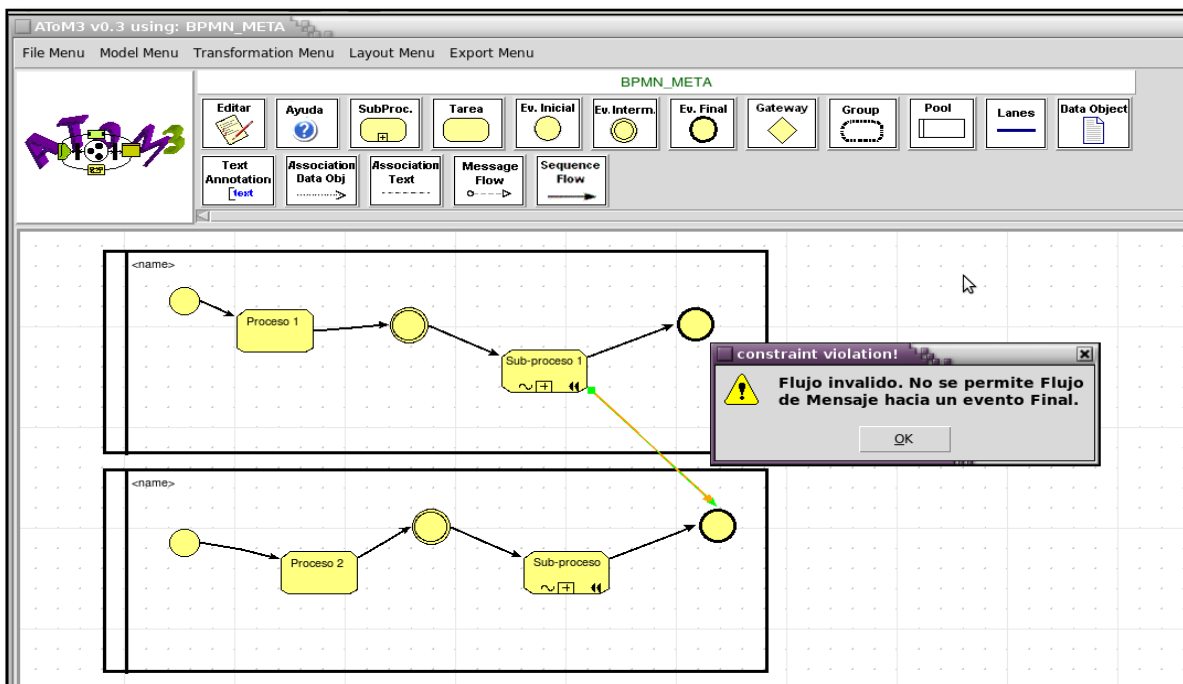


Figura C.38: Conexión entre flujo de mensaje de Subproceso a Evento Intermedio.

## V41: Flujo de mensaje: Evento Inicial – Tarea

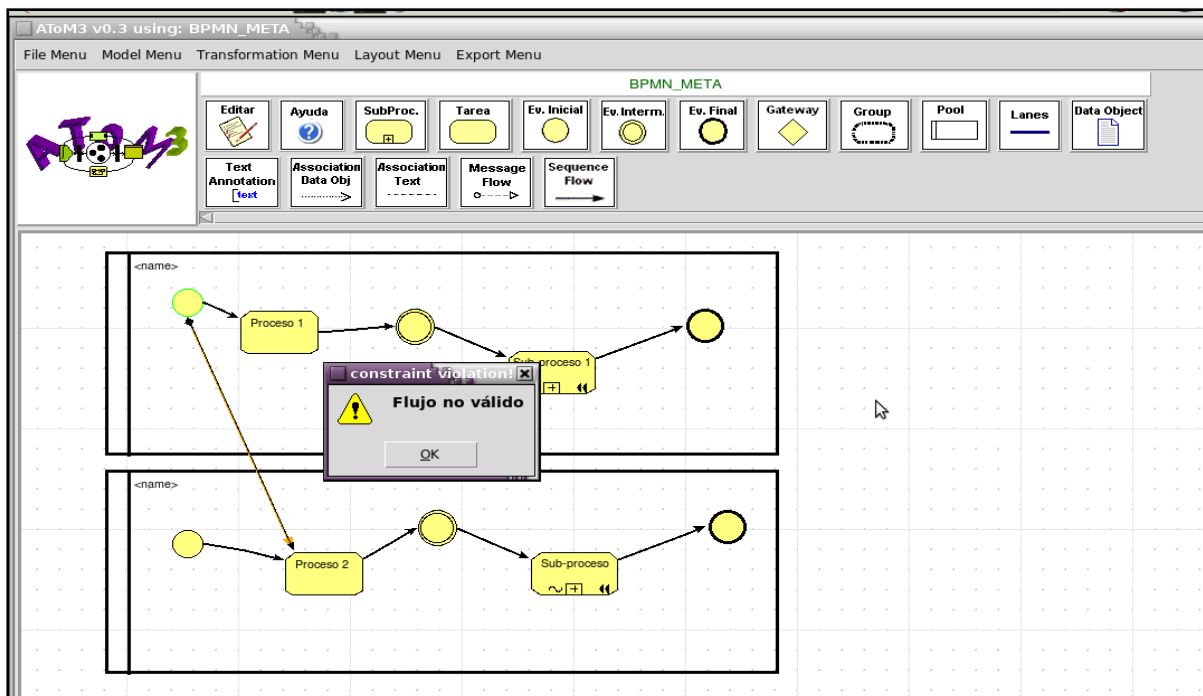


Figura C.39: Conexión entre flujo de mensaje Evento Inicial a Tarea.

## V42: Flujo de mensaje: Evento Inicial – Subproceso

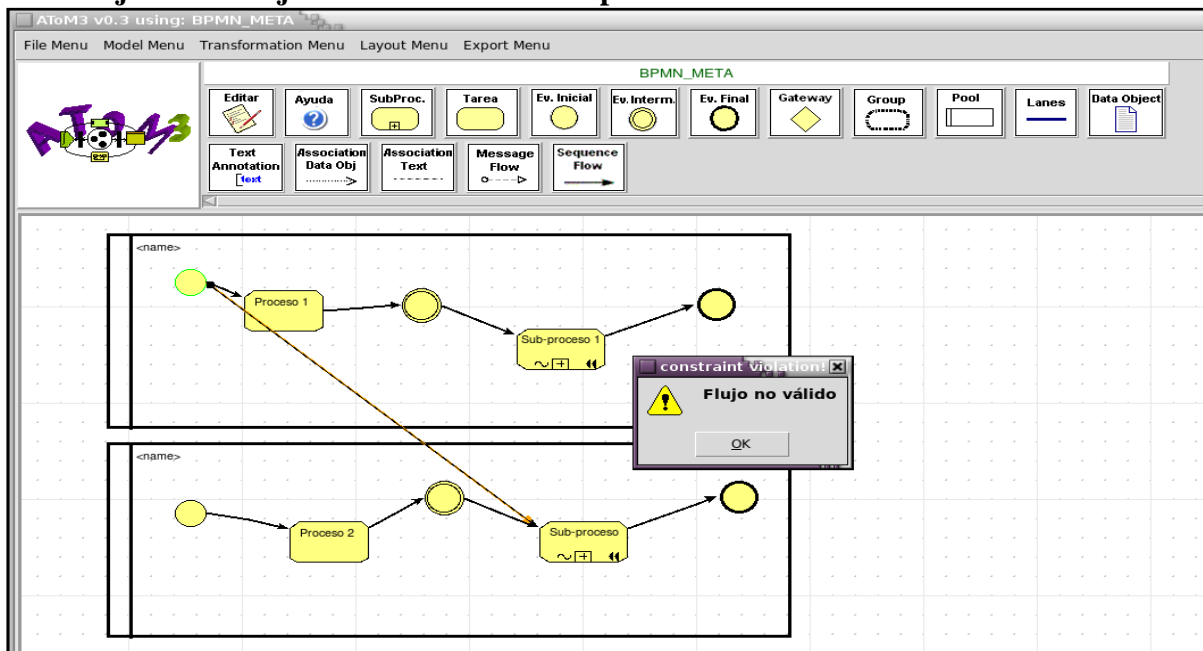


Figura C.40: Conexión entre flujo de mensaje Evento Inicial a Tarea.

### V43: Flujo de mensaje: Evento Inicial – Evento Inicial

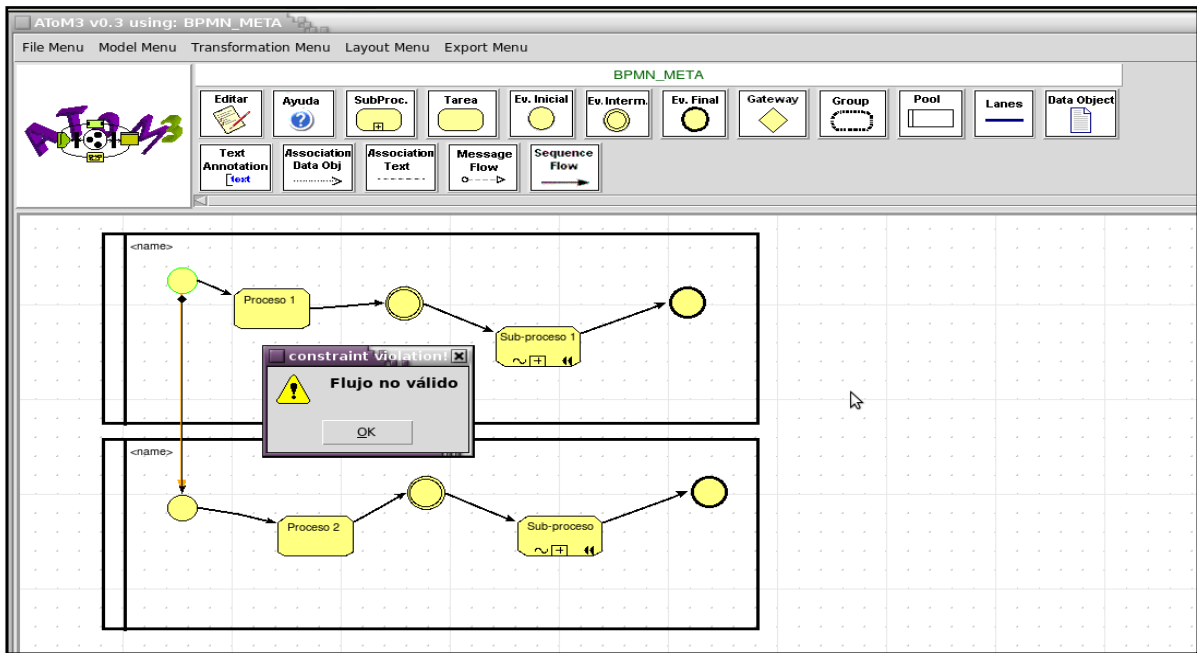


Figura C.41: Conexión entre flujo de mensaje Evento Inicial a Evento Inicial.

### V44: Flujo de mensaje: Evento Inicial – Evento Intermedio

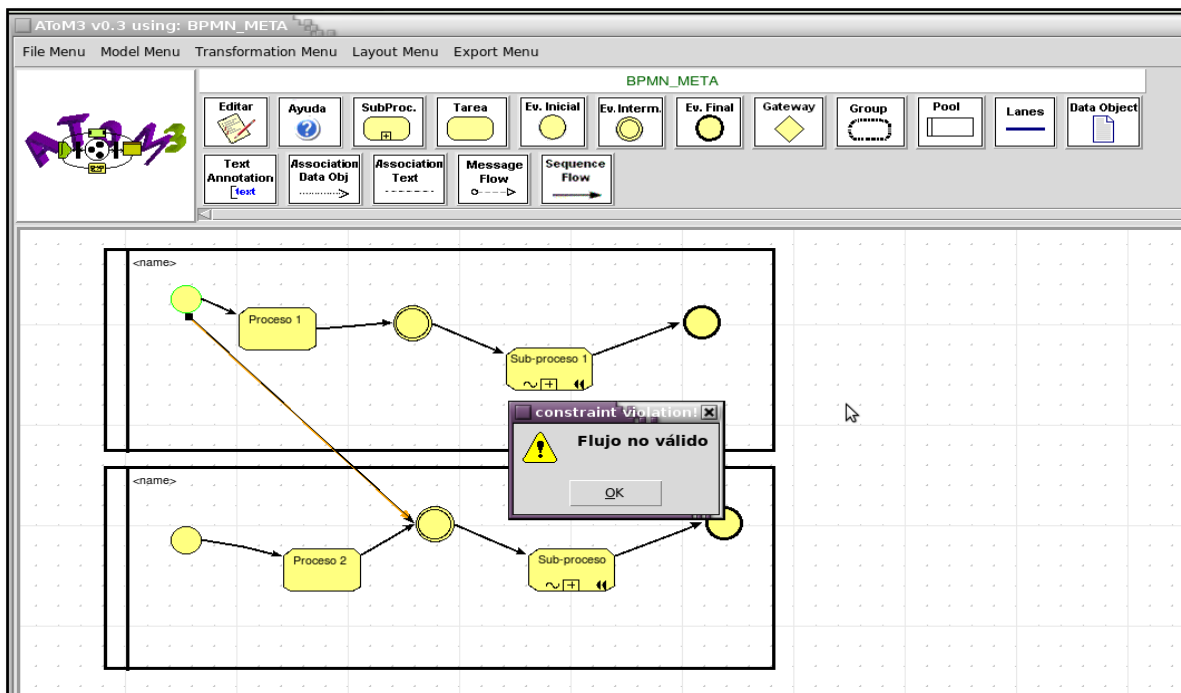


Figura C.42: Conexión entre flujo de mensaje Evento Inicial a Evento Intermedio.

## V45: Flujo de mensaje: Evento Inicial – Evento Final

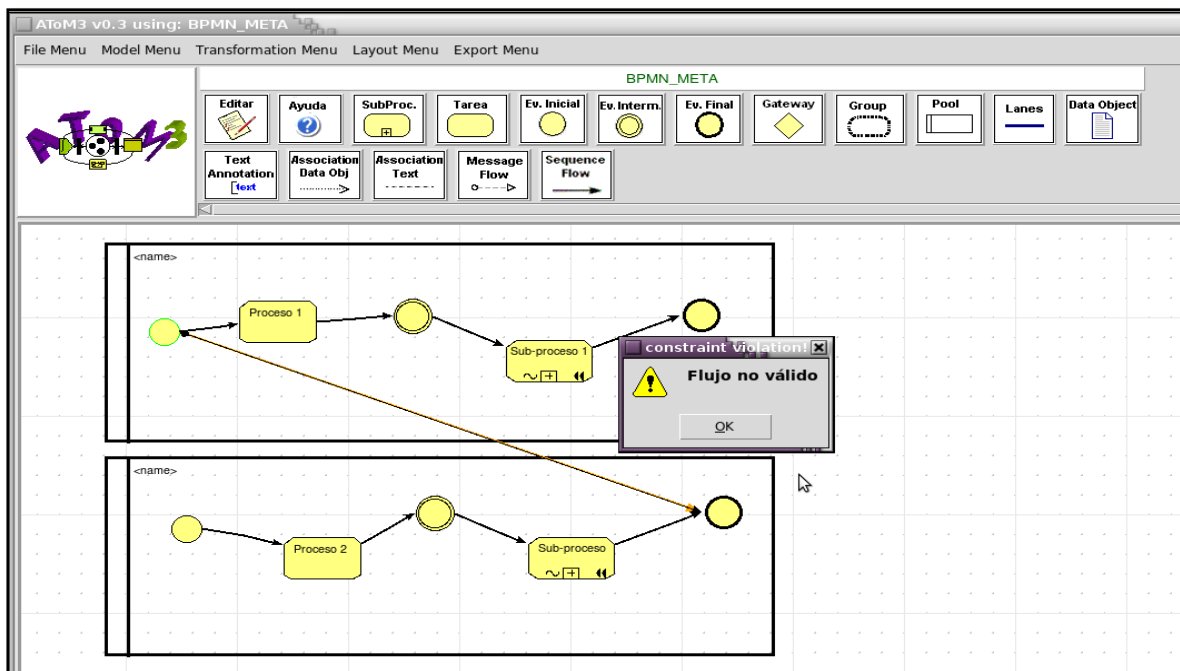


Figura C.43: Conexión entre flujo de mensaje Evento Inicial a Evento Final.

## V46: Flujo de mensaje: Evento Intermedio – Tarea

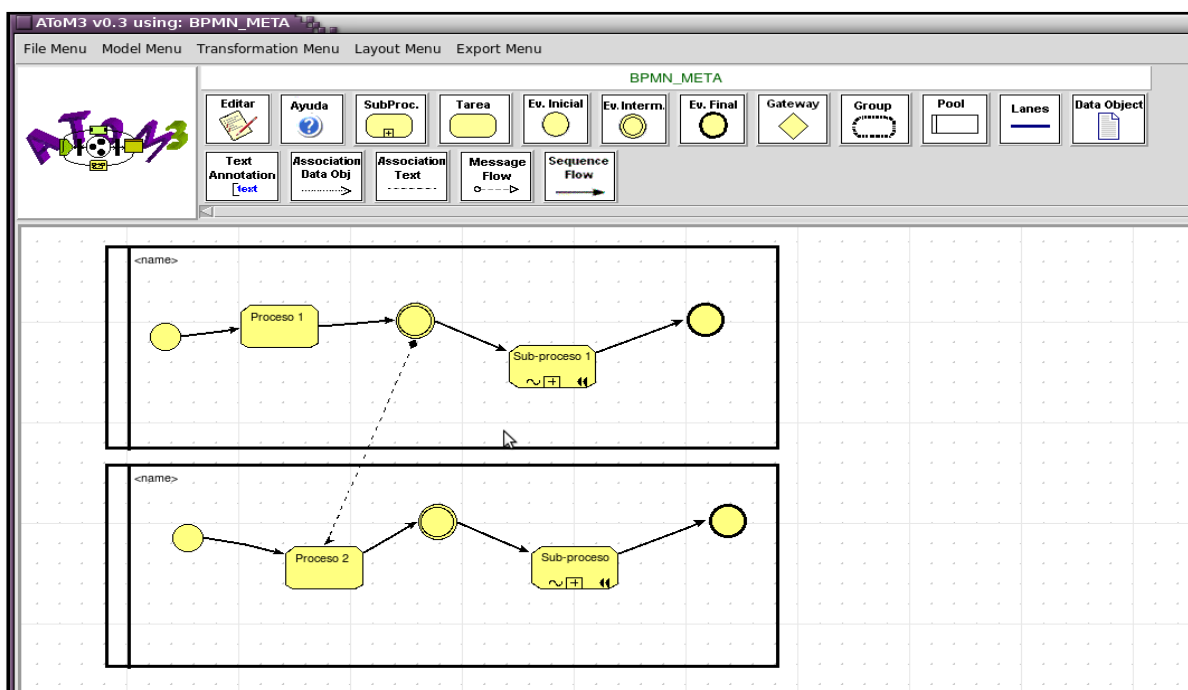


Figura C.44: Conexión entre flujo de mensaje Evento Intermedio y Tarea.

## V47: Flujo de mensaje: Evento Intermedio – Subproceso

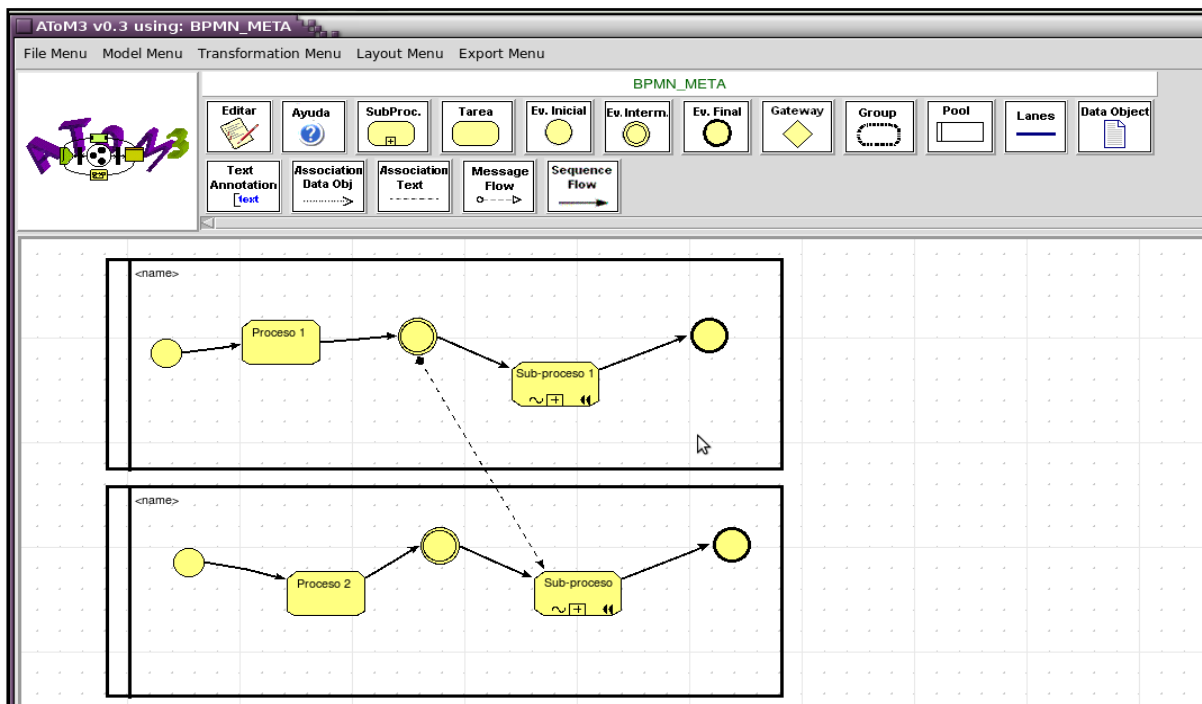


Figura C.45: Conexión entre flujo de mensaje Evento Intermedio y Subproceso.

## V48: Flujo de mensaje: Evento Intermedio – Evento Inicial

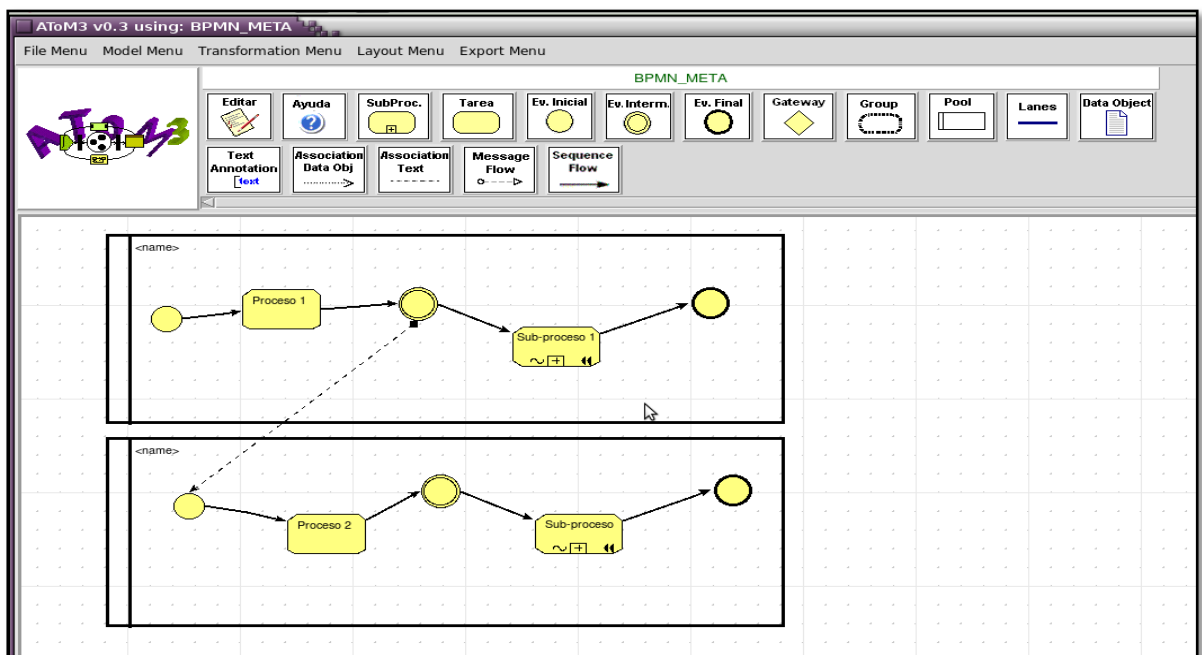


Figura C.46: Conexión entre flujo de mensaje Evento Intermedio y Evento Inicial.

## V49: Flujo de mensaje: Evento Intermedio – Evento Final

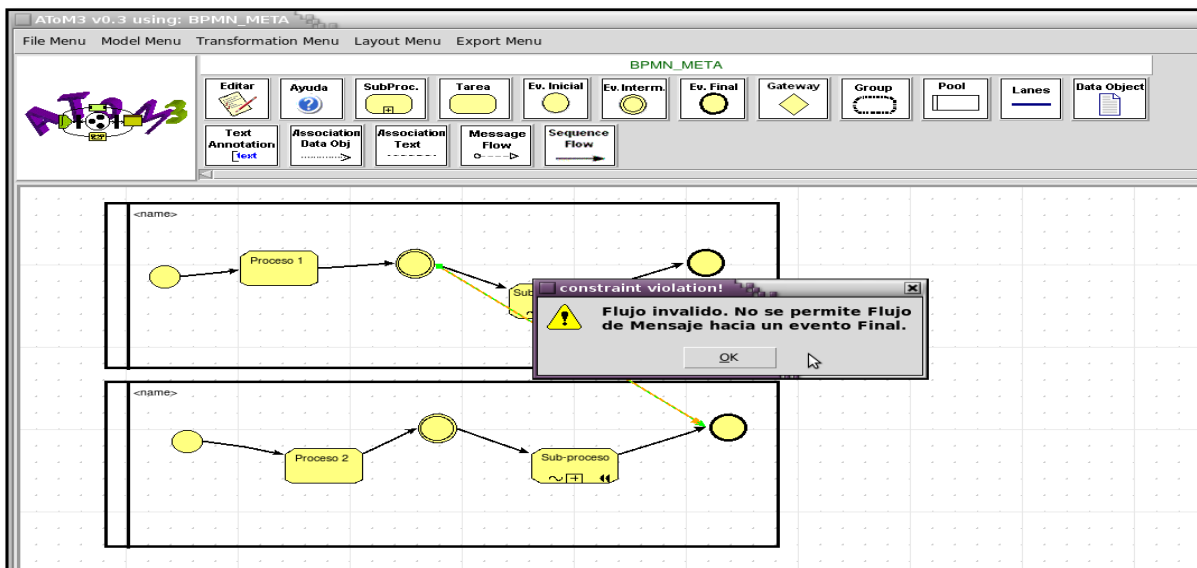


Figura C.47: Conexión entre flujo de mensaje Evento Intermedio y Evento Final.

## V50: Flujo de mensaje: Evento Final – Tarea

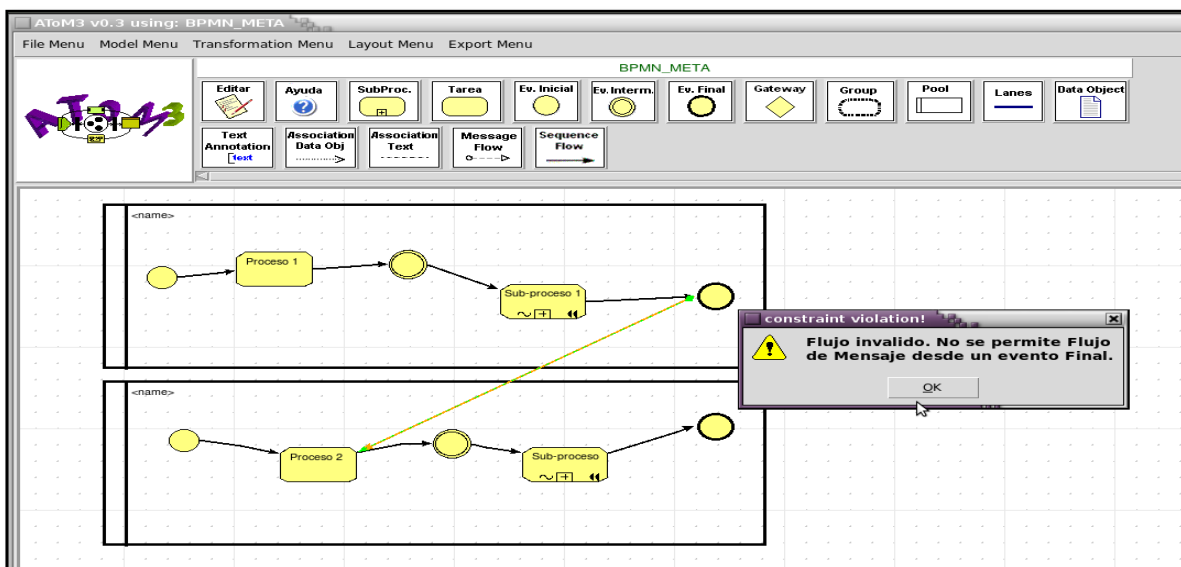


Figura E.48: Conexión entre flujo de mensaje Evento Final y Tarea.

## V51: Flujo de mensaje: Evento Final – Subproceso

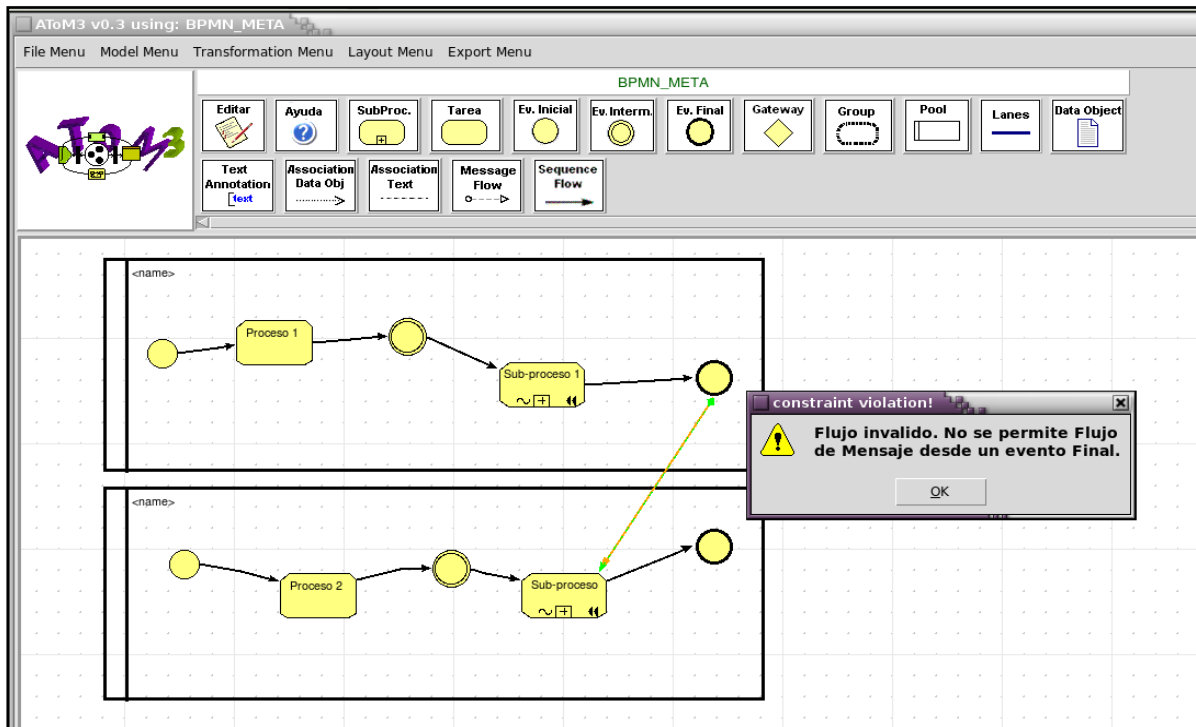


Figura C.49: Conexión entre flujo de mensaje Evento Final y Subproceso.

## V52: Flujo de mensaje: Evento Final – Evento Inicial

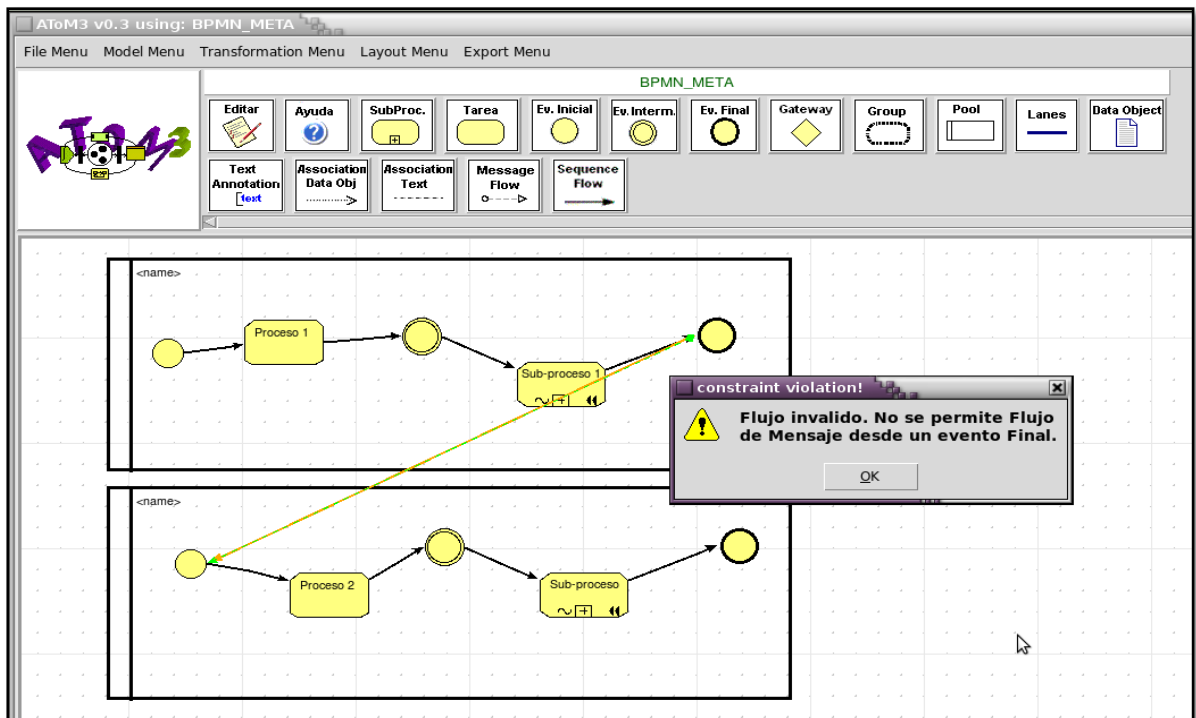


Figura C.50: Conexión entre flujo de mensaje Evento Final y Evento Inicial

### V53: Flujo de mensaje: Evento Final – Evento Intermedio

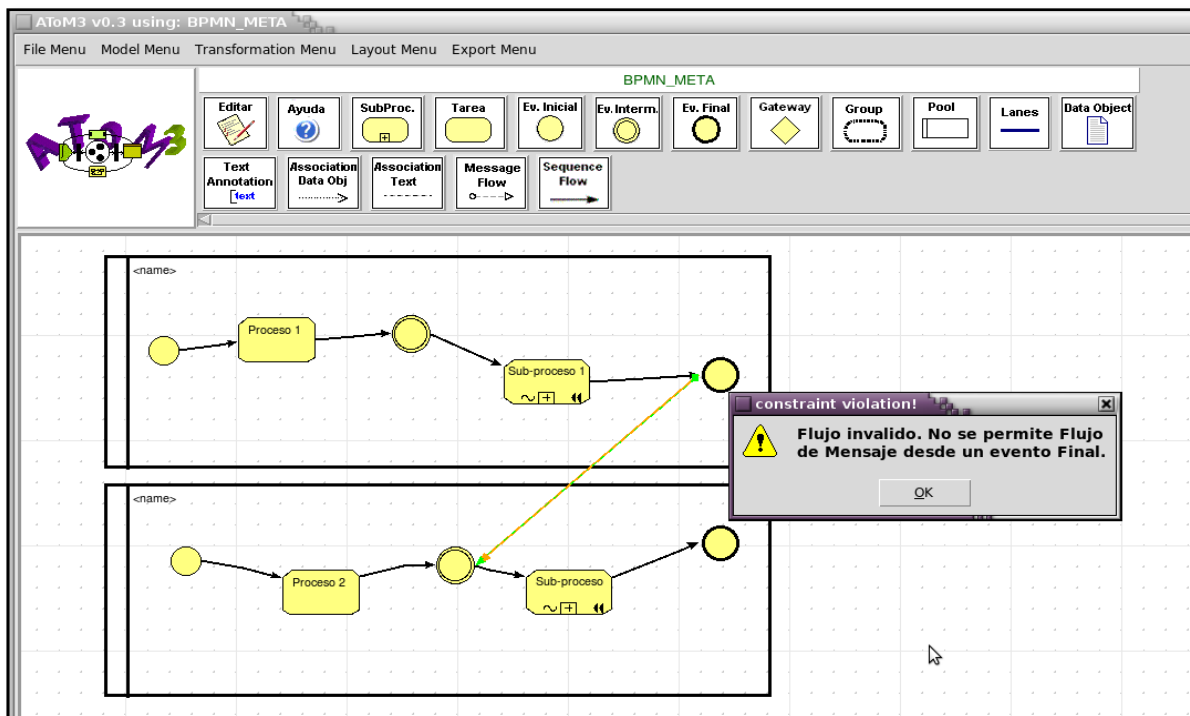


Figura C.51: Conexión entre flujo de mensaje Evento Final y Evento Intermedio.

### V54: Flujo de mensaje: Evento Final – Evento Final

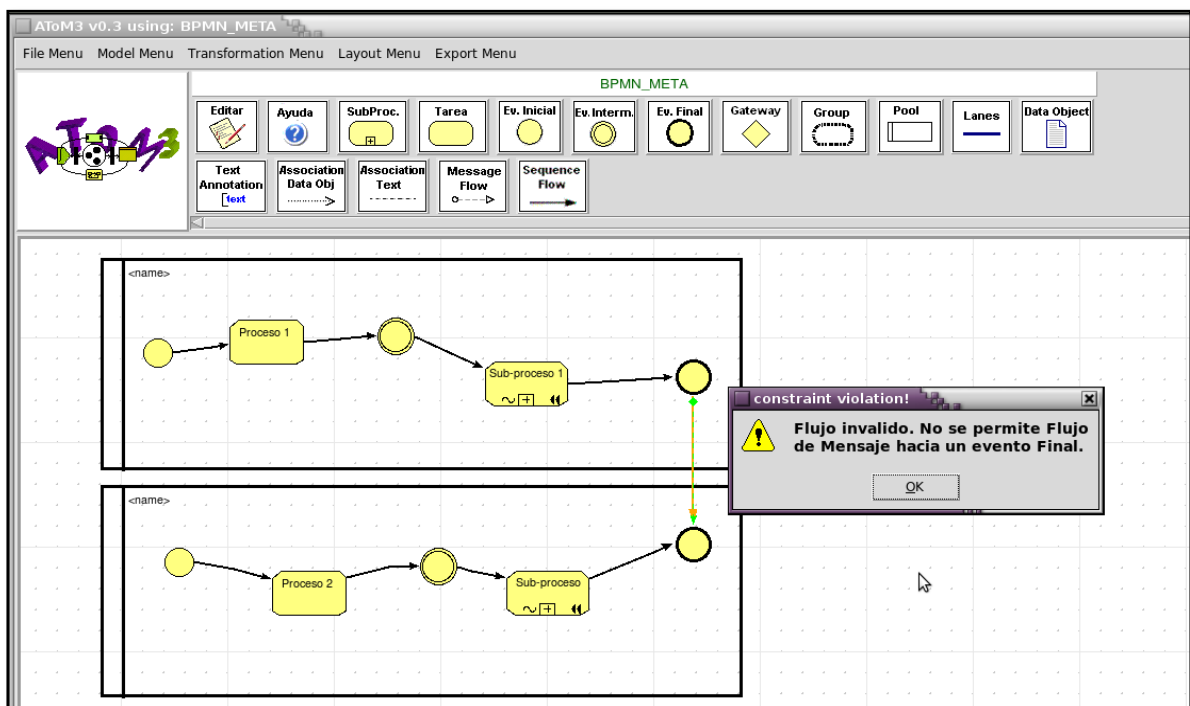


Figura C.52: Conexión entre flujo de mensaje Evento Final y Evento Intermedio.

REGLAS GRAMATICALES EN ATOM<sup>3</sup>

Para la simulación de los modelos definidos mediante el formalismo BPMN, en la sección III.3.1 se definió las reglas de gramática de grafos que expresan la semántica operacional de este tipo de modelos. A continuación se definen estas reglas gramaticales de manera visual en ATOM<sup>3</sup> y programadas en lenguaje *python*.

En la Figura D.1 se muestra la *regla de secuencia de flujo*, la cual se ejecuta cuando encuentra la parte izquierda en el modelo a simular y se cumple una condición, la cual es que el estado de la actividad es “Finalizada”. Si se cumple esta condición, entonces realiza una acción que es cambiar el estado de la actividad a “Activada”. Tanto la “condición” y la “acción” están programadas en lenguaje *python* (Ver Figura D.2.)

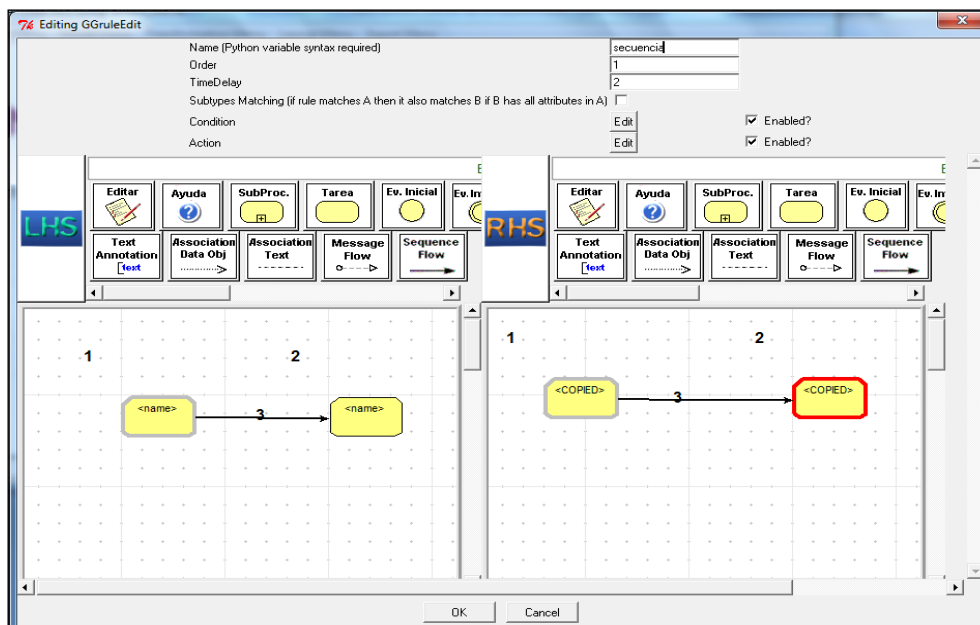


Figura D.1. Reglas de flujo de secuencia.

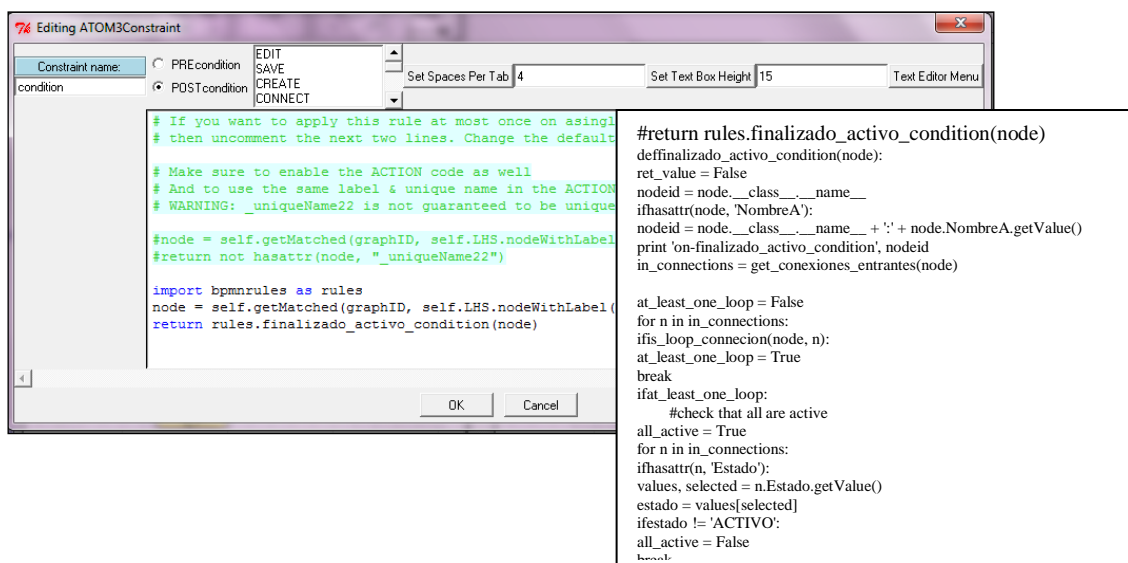


Figura D.2. Condición de la regla secuencia de flujo codificada en Python.

En la Figura D.3 se muestra la regla de división paralela, la cual se ejecuta cuando encuentra en la parte izquierda en el modelo y *Gatewayand-split* verifica la terminación de las actividades entrantes al mismo, dando inicio a que las actividades salientes se realicen paralelamente, colocando su estado en “activada”.

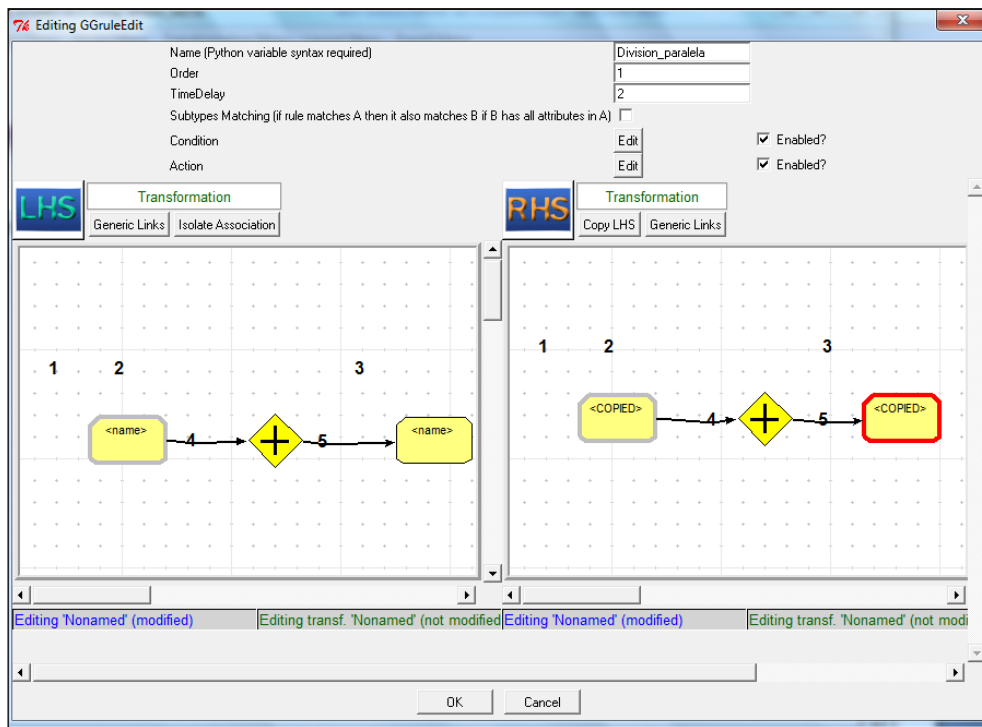


Figura D.3. Regla de división paralela

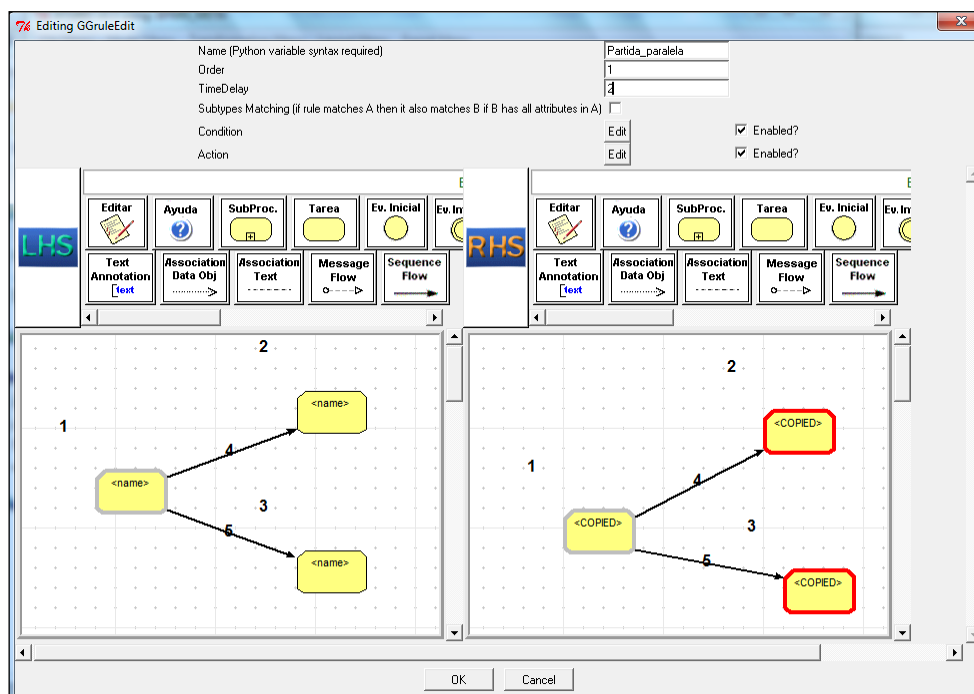


Figura D.4. Regla de partida paralela

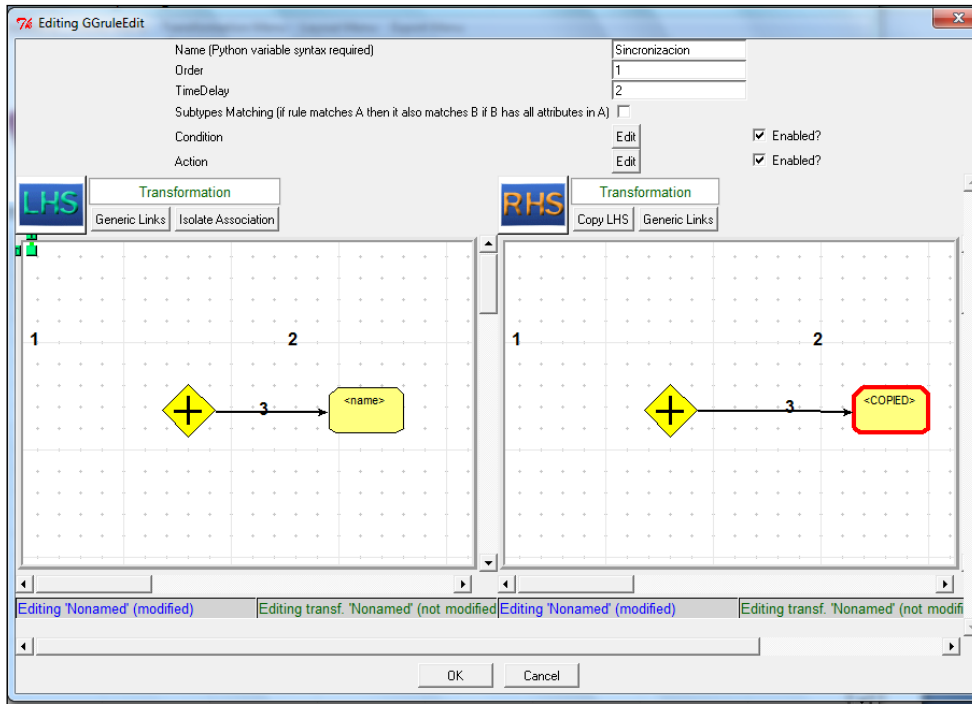


Figura D.5. Regla sincronización

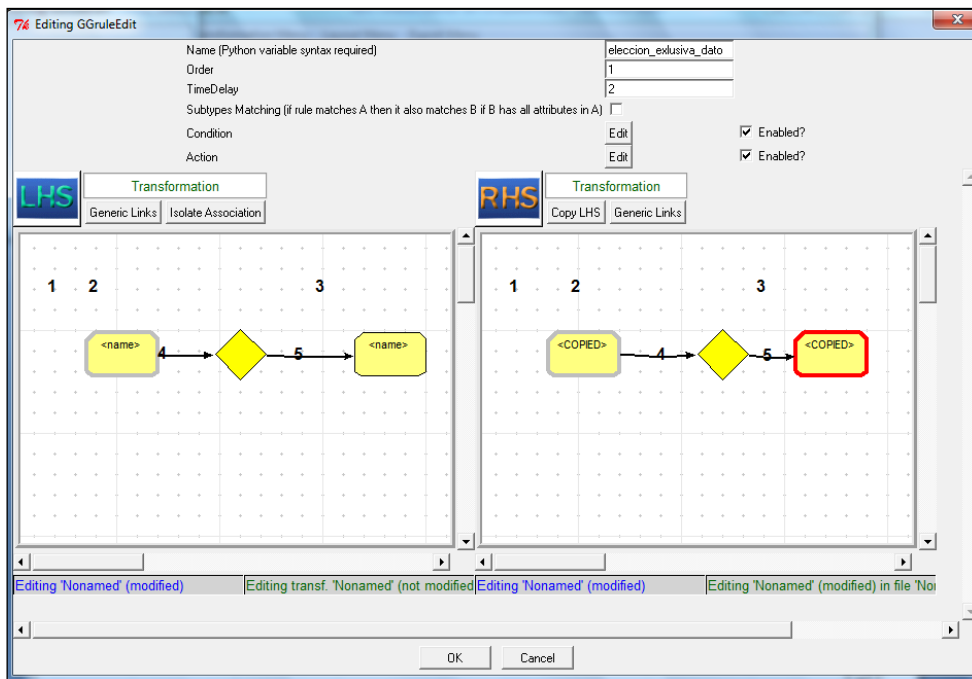


Figura D.6. Regla elección exclusiva basada en datos

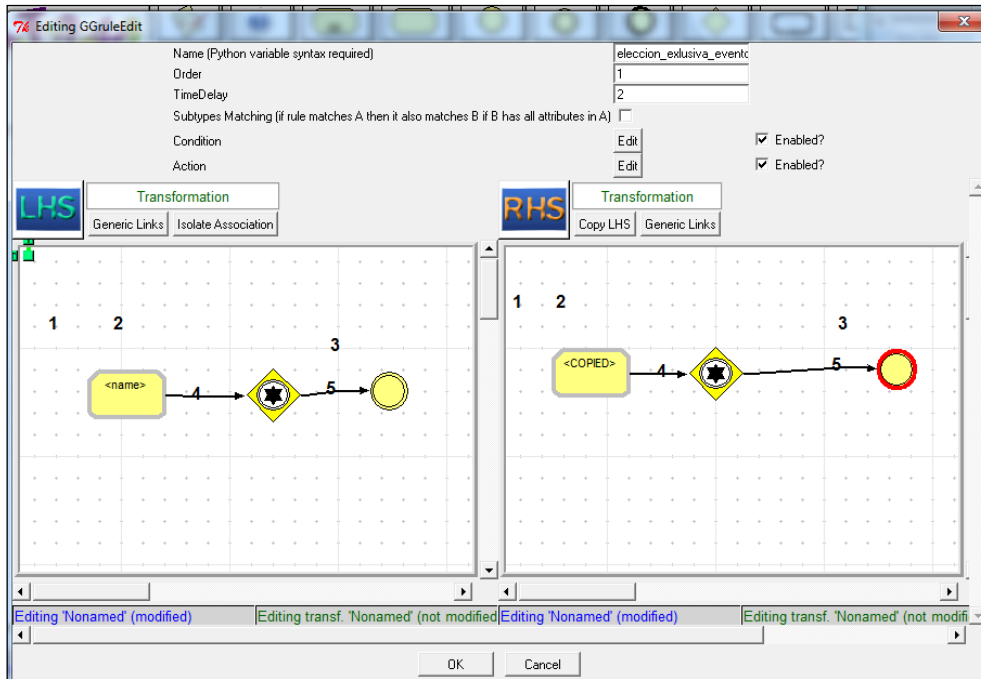


Figura D.7. Regla elección exclusiva basada en eventos

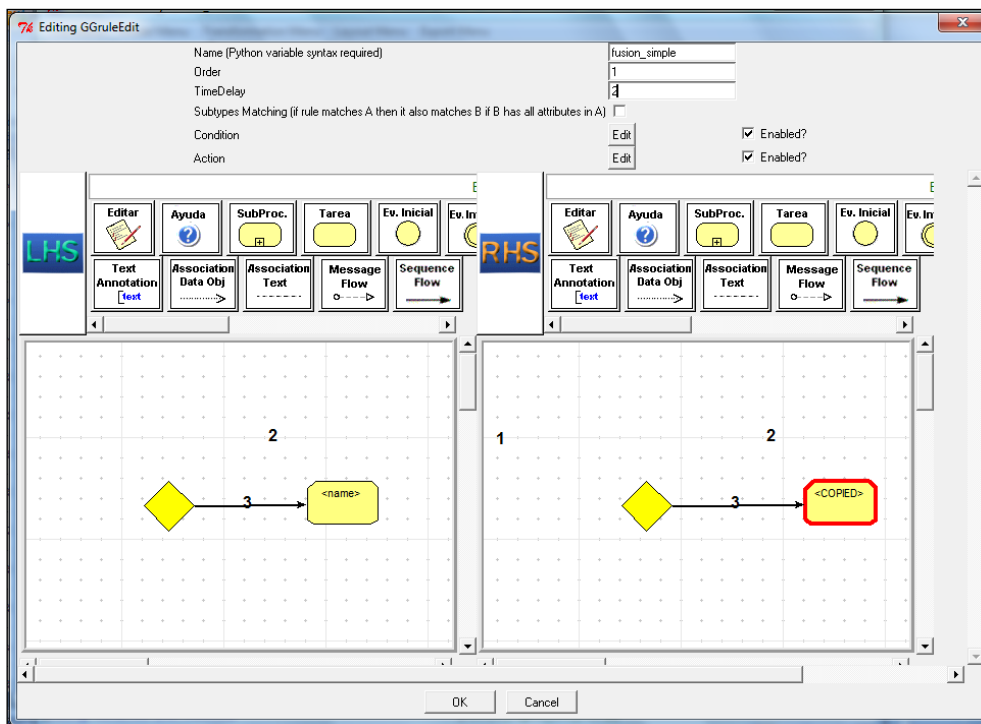


Figura D.8. Regla fusión simple

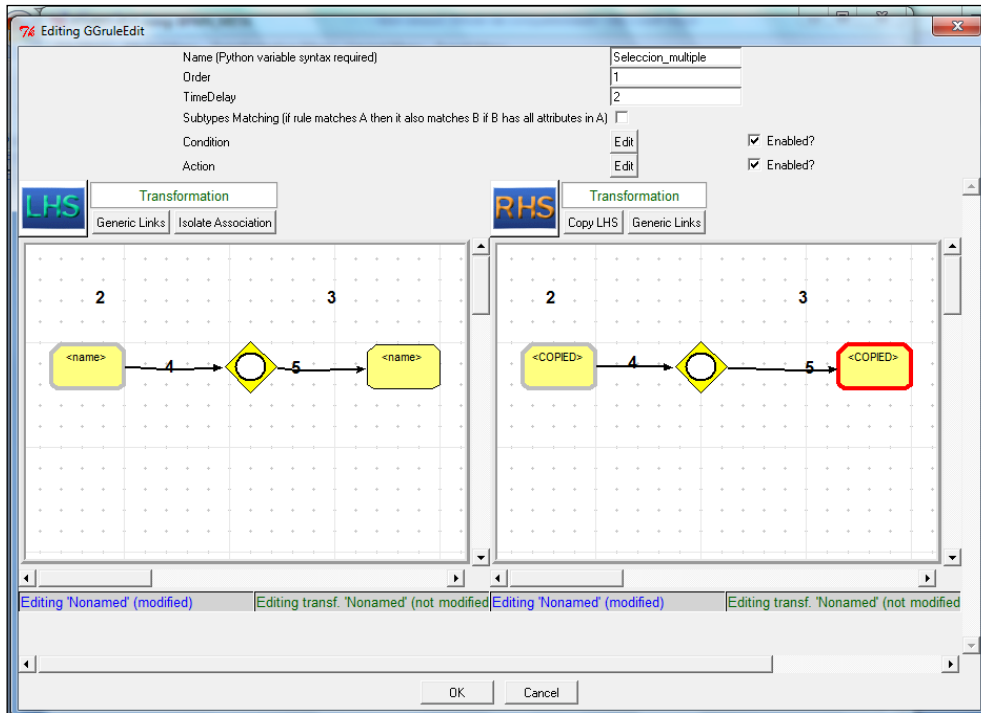


Figura D.9. Regla de selección múltiple

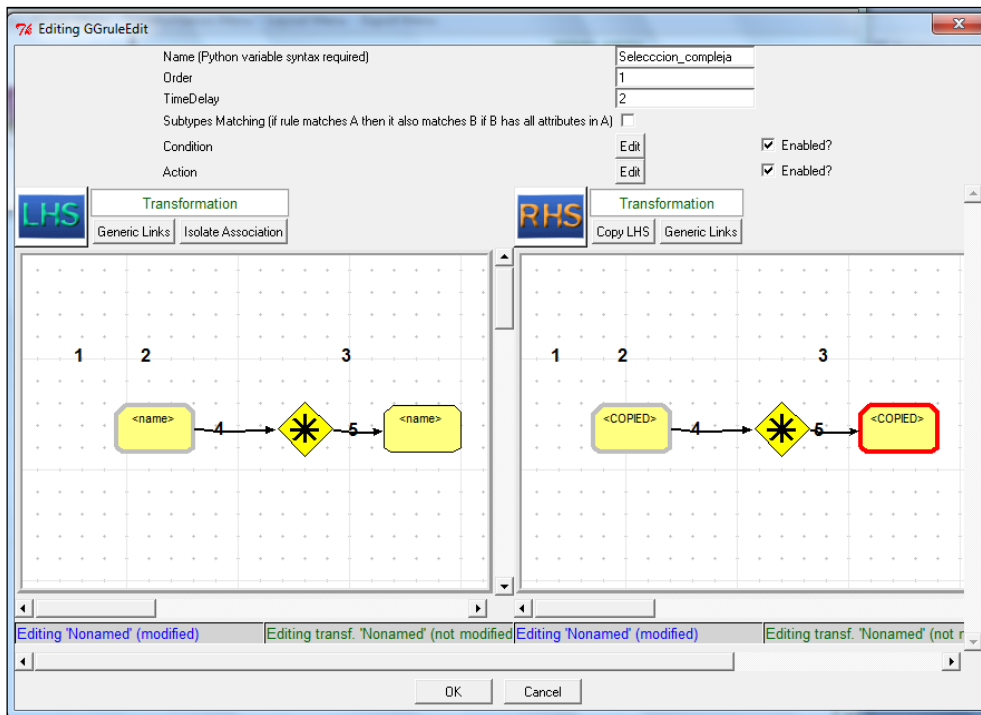


Figura D.10. Regla de selección compleja

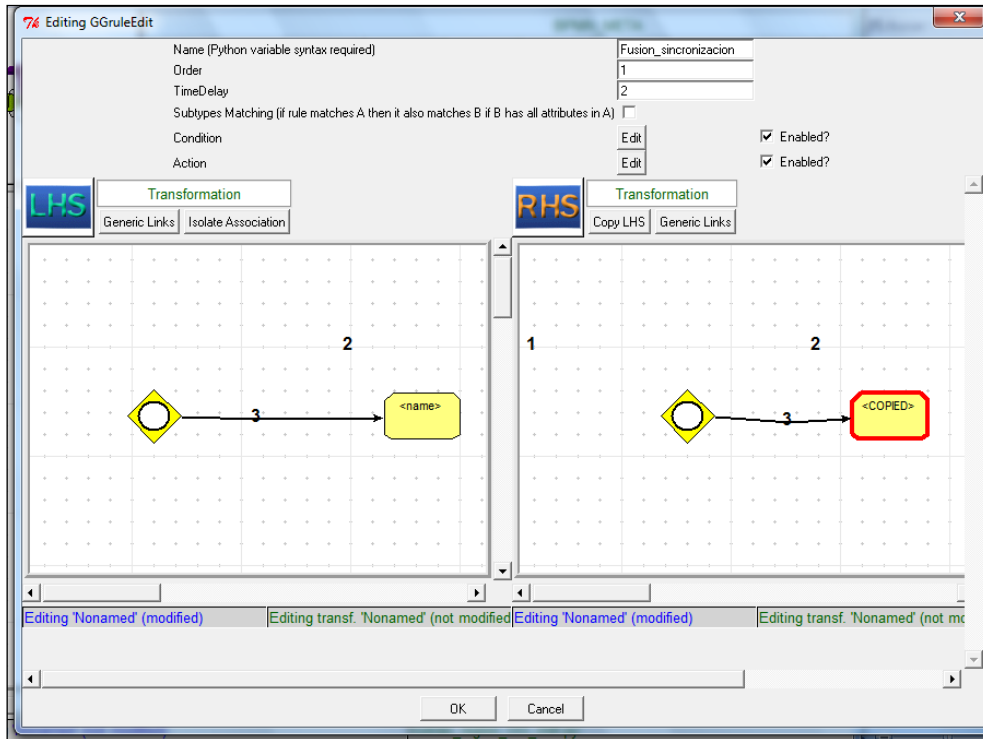


Figura D.11. Regla de fusión de sincronización

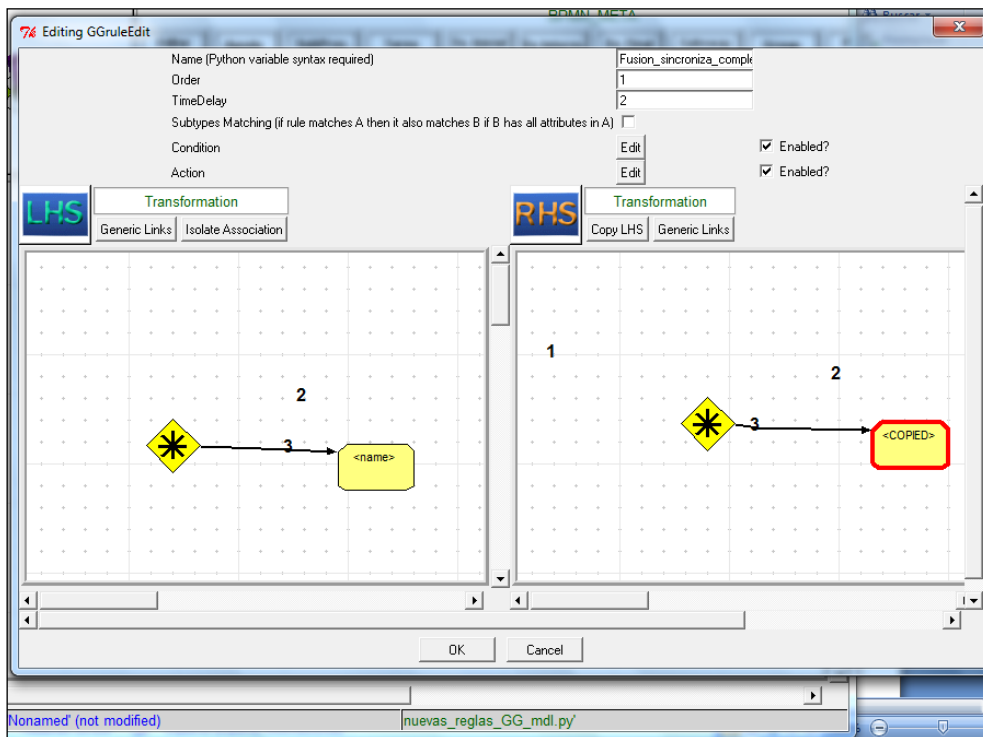


Figura D.12. Regla de fusión de sincronización compleja

## SIMULACIÓN DEL MODELO DE PROCESO DE SOFTWARE ORIENTADO AL NEGOCIO

Para Simular el proceso de software con Atom<sup>3</sup> debemos realizar los siguientes pasos:

### 1) Cargar el Modelo a Simular

Al realizar esta acción se activa la siguiente ventana, en la cual se selecciona la opción de *UserModel*, que activa la ventana de Selección del Modelo, para este caso en particular se abre el archivo **MPS\_MDL.py** (Ver Fig. E.1)

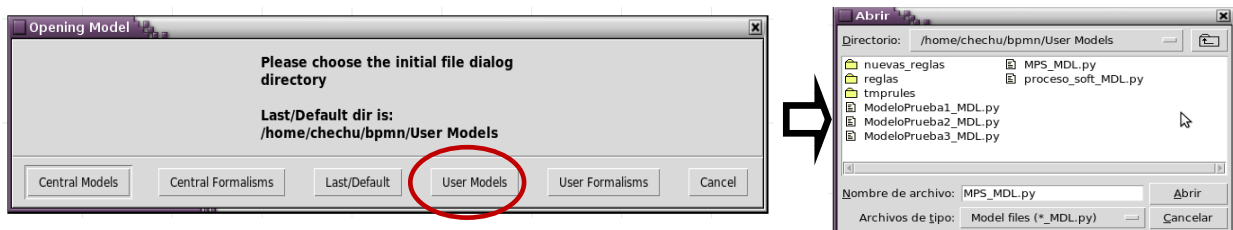


Figura E.1. Cargar el modelo a simular

Al Abrir el modelo, Atom<sup>3</sup> se muestra:

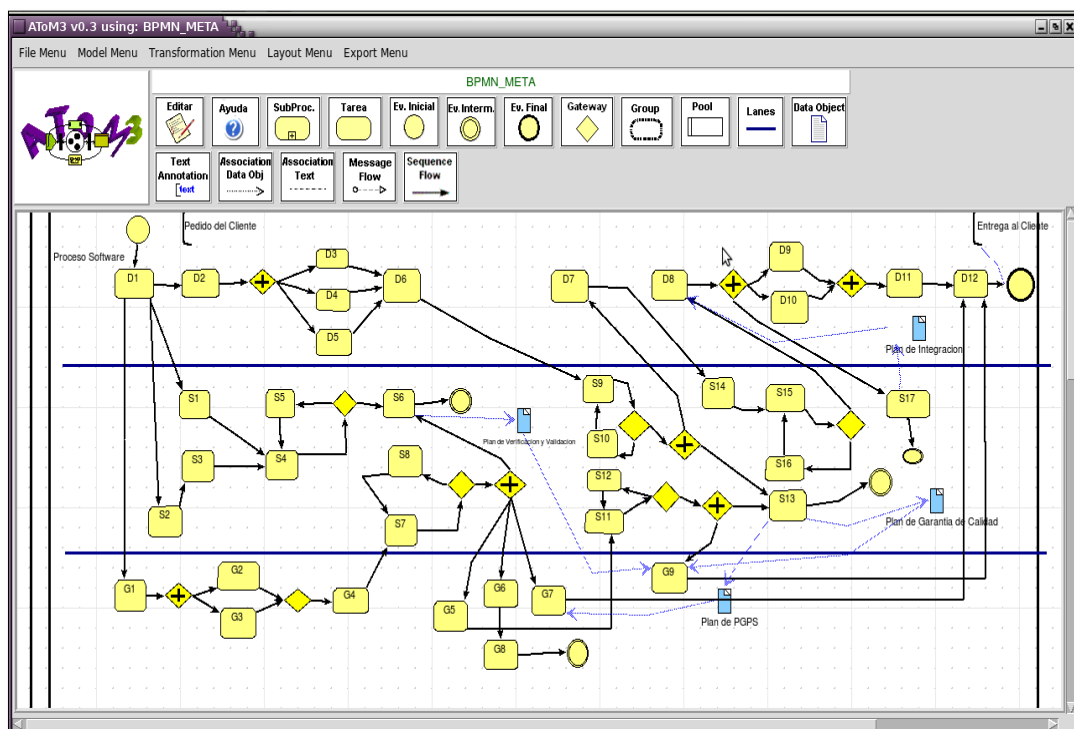
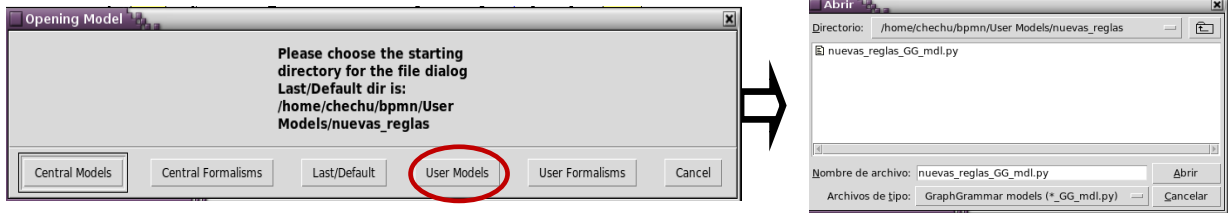


Figura E.2. Proceso de software.

2) Cargar las reglas de BPMN: con esta acción se carga al modelo las reglas de comportamiento de la notación. En la opción del menú se realiza;

## Transformation Menú ->Load transformation

Esta acción abre la siguiente ventana:



UserModel nos permite abrir las reglas de comportamiento de la notación

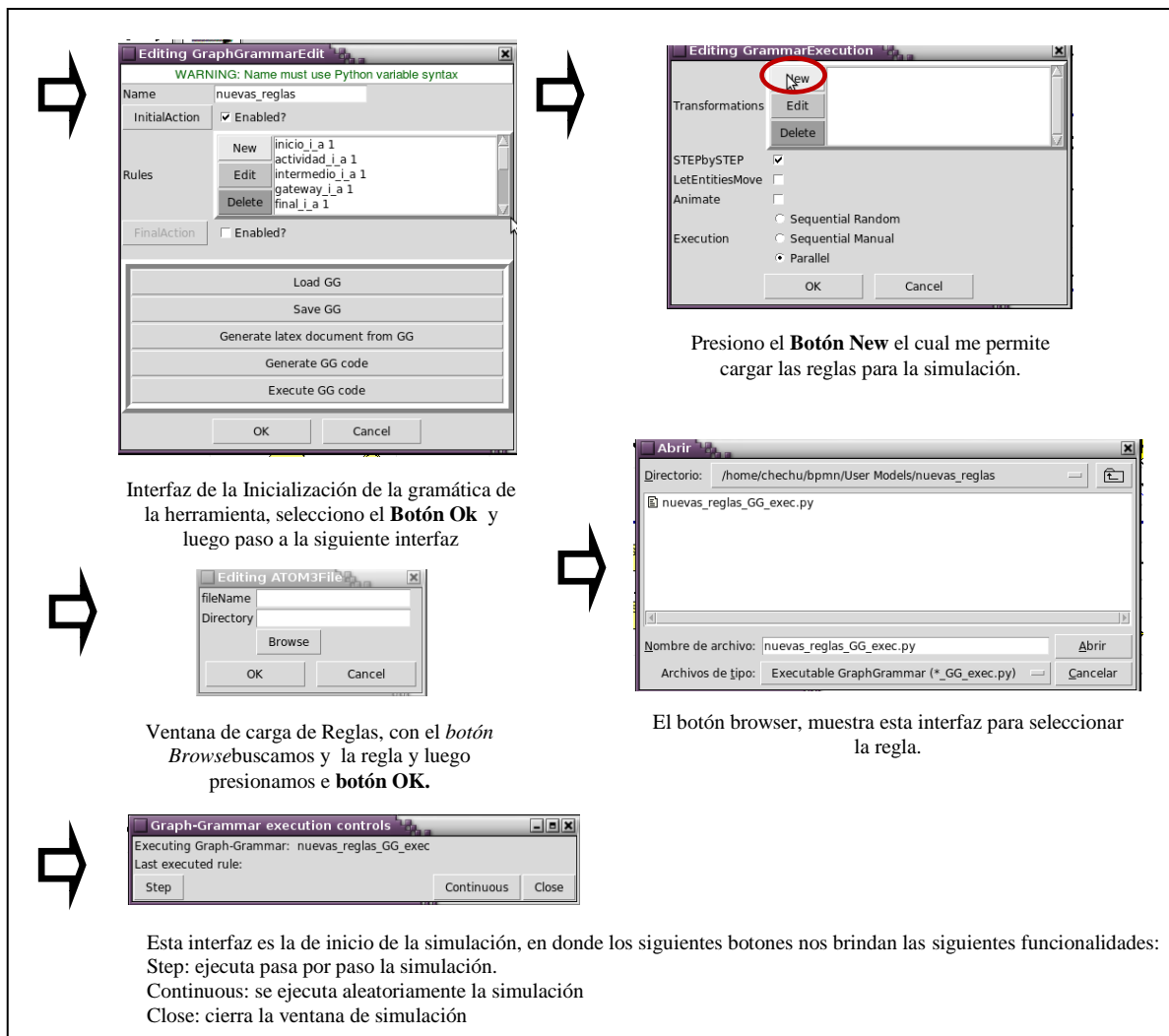
Abriremos la regla de comportamiento de BPMN, en nuestro caso seleccionamos *nuevas\_reglas\_GG\_md1.py*

Figura E.3 Cargar las reglas de BPMN

- 3) **Configurar la simulación:** para ejecutar se corroboran las reglas a ejecutar de la notación, al presionar en el menú la opción:

## TransformationMenu ->ExecuteTransformation

Al seleccionar esta opción del menú, se abre la siguiente interfaz:



Presiono el **Botón New** el cual me permite cargar las reglas para la simulación.

Interfaz de la Inicialización de la gramática de la herramienta, selecciono el **Botón Ok** y luego paso a la siguiente interfaz

Ventana de carga de Reglas, con el *botón Browse* buscamos y la regla y luego presionamos **botón OK**.

El botón *browser*, muestra esta interfaz para seleccionar la regla.

Esta interfaz es la de inicio de la simulación, en donde los siguientes botones nos brindan las siguientes funcionalidades:  
Step: ejecuta paso por paso la simulación.  
Continuous: se ejecuta aleatoriamente la simulación  
Close: cierra la ventana de simulación

Figura E.4. Configurar la simulación

#### 4) Ejecutar la Simulación:

Para Iniciar la simulación del modelo se presiona el botón de **Step** de la ventana *Graph-Grammarexecutioncontrols*.

✓ **Sim-Step1:** Se inicia el Evento de la Construcción de Software

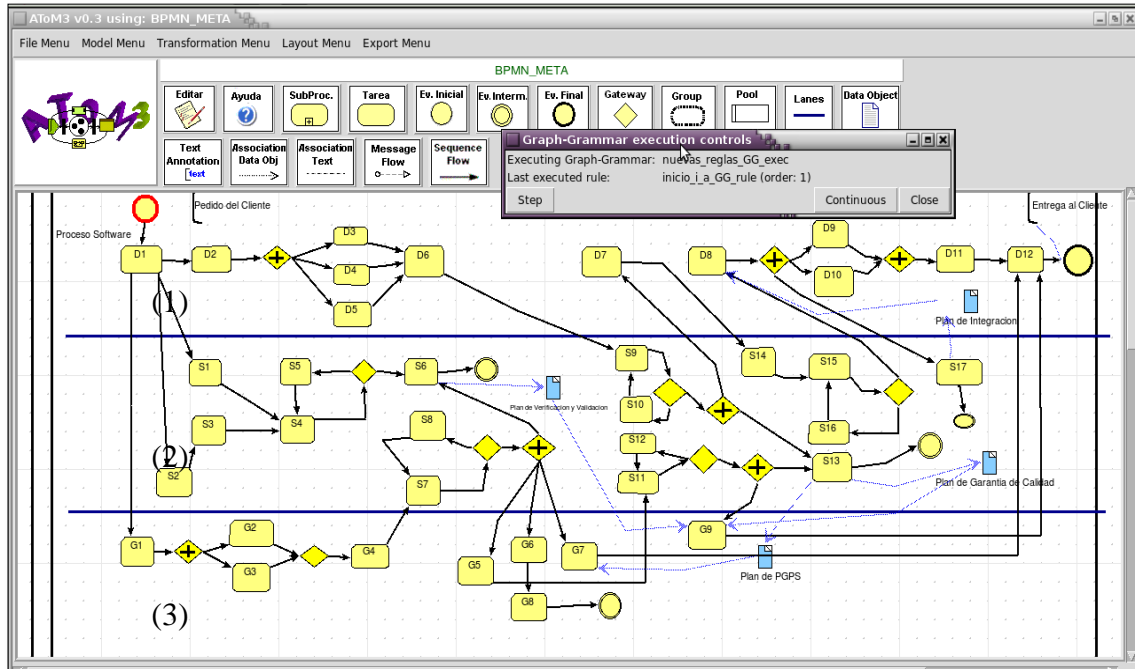


Figura E.5. Proceso de software.

En esta interfaz aparece el modelo cargado del Proceso de Software el cual está dividido por dos *lane* en:

- (1) Actividades de Gestión.
- (2) Actividades de Soporte.
- (3) Actividades de Desarrollo.

Cada uno de los eventos van a ir ejecutándose a medida que se avance, de manera dinámica y paralela. Cada evento de *Inicio* de un evento, actividad o gateway es marcado con un círculo color rojo y cada *Fin* de cualquier elemento con un círculo.

✓ **Sim-Fin-Step1:** En el modelo luego de iniciado el evento este finaliza y se representa de esta manera:

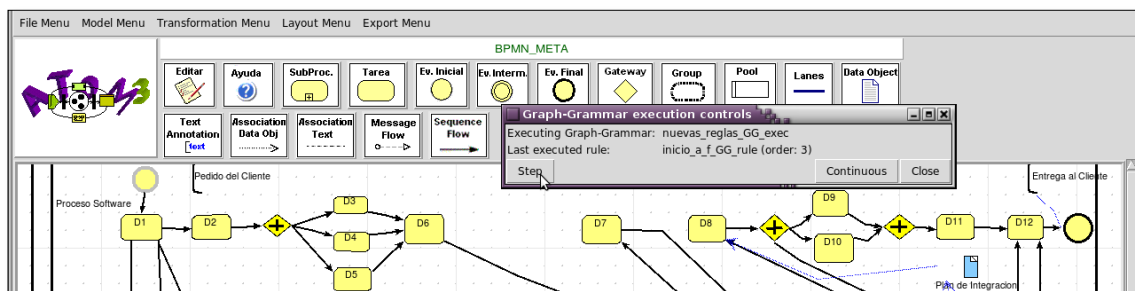


Figura E.6. Fin de ejecución del Evento Inicial.

- ✓ **Sim-Step2:** se activa la actividad D1 la cual pertenece a la *Actividad de Gestión* y corresponde a **D1=Identificación de necesidades del proyecto**.

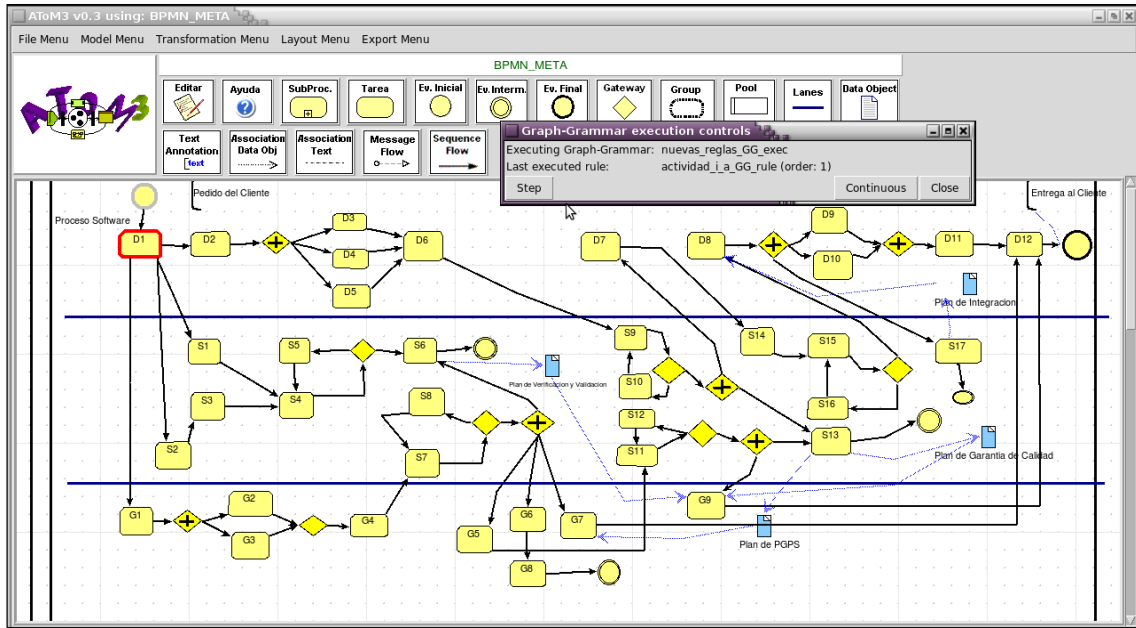


Figura E.7: Inicio del Proceso D1.

- ✓ **Sim-Fin-Step2:** finaliza la actividad D1

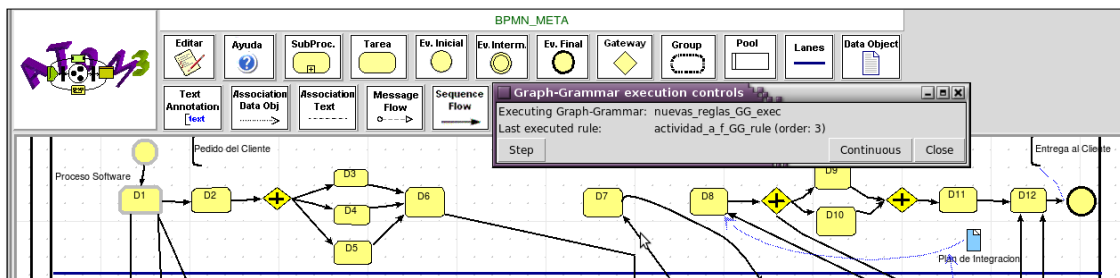


Figura E.8. Finalización del Proceso D1.

- ✓ **Sim-Step3:** Inicio de actividades paralelas

En esta acción se activan en forma paralela los procesos:  
 Actividades de Gestión - D2= Especificar requisitos Funcionales.  
 Actividades de Soporte – S1= Elaborar plan de verificación y validación.  
 Actividades de Soporte – S2= Identificar configuración del sistema.  
 Actividades de Soporte – G1= Elaborar mapa de actividades según MCVS.

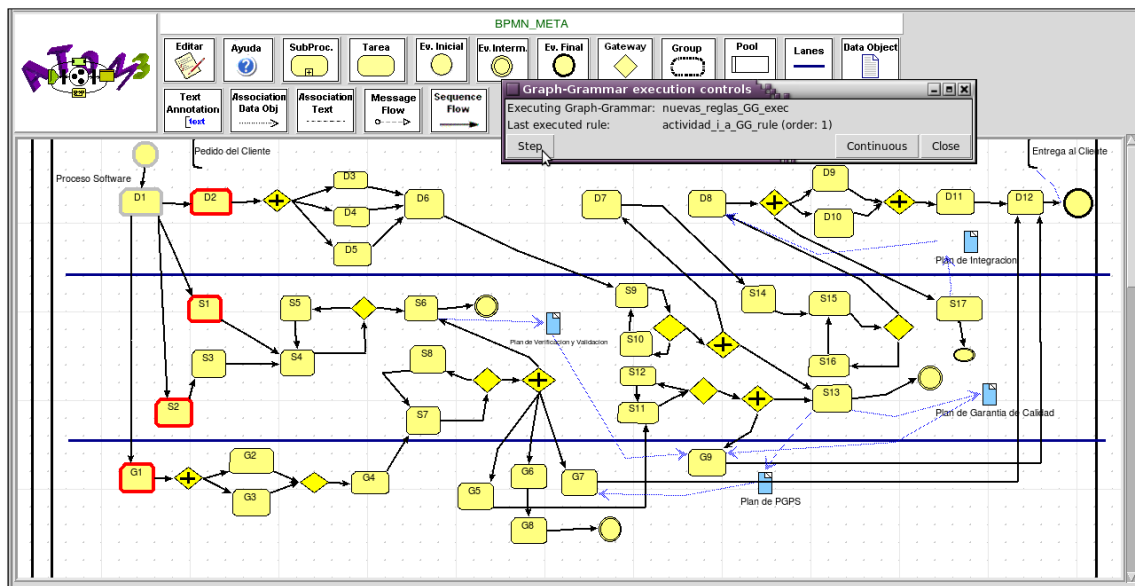


Figura E.9. Inicio de Actividades paralelas.

✓ **Sim-Fin-Step3:** Finalización de actividades paralelas

Actividades de Gestión - D2= Especificar requisitos Funcionales.

Actividades de Soporte – S1= Elaborar plan de verificación y validación.

Actividades de Soporte – S2= Identificar configuración del sistema.

Actividades de Soporte – G1= Elaborar mapa de actividades según MCVS

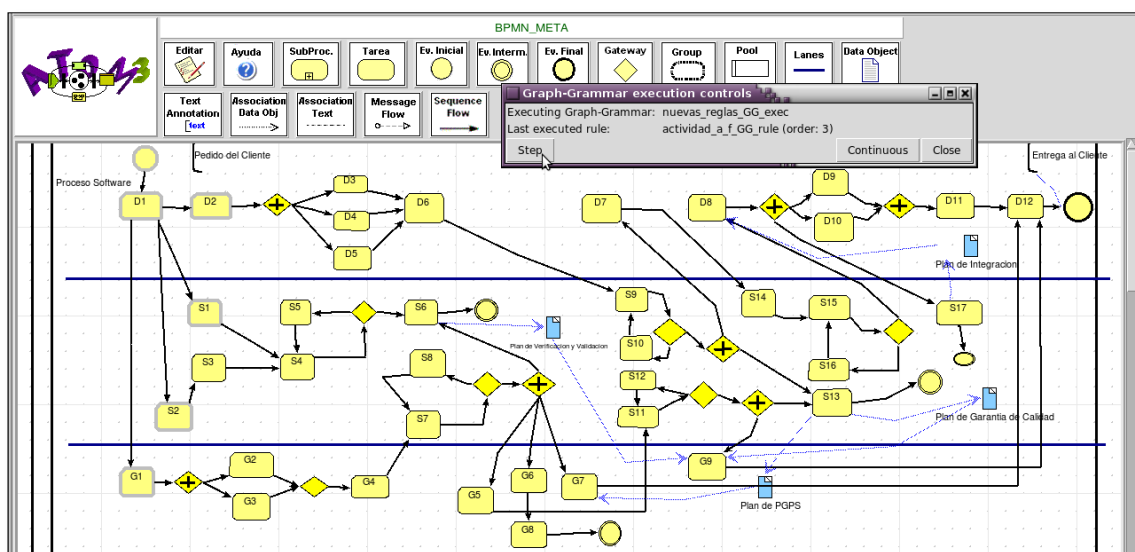


Figura E.10. Finalización de Actividades paralelas.

✓ **Sim-Step4:** Activación de Actividad de Soporte

S3 = Planificar gestión de configuración del Software.

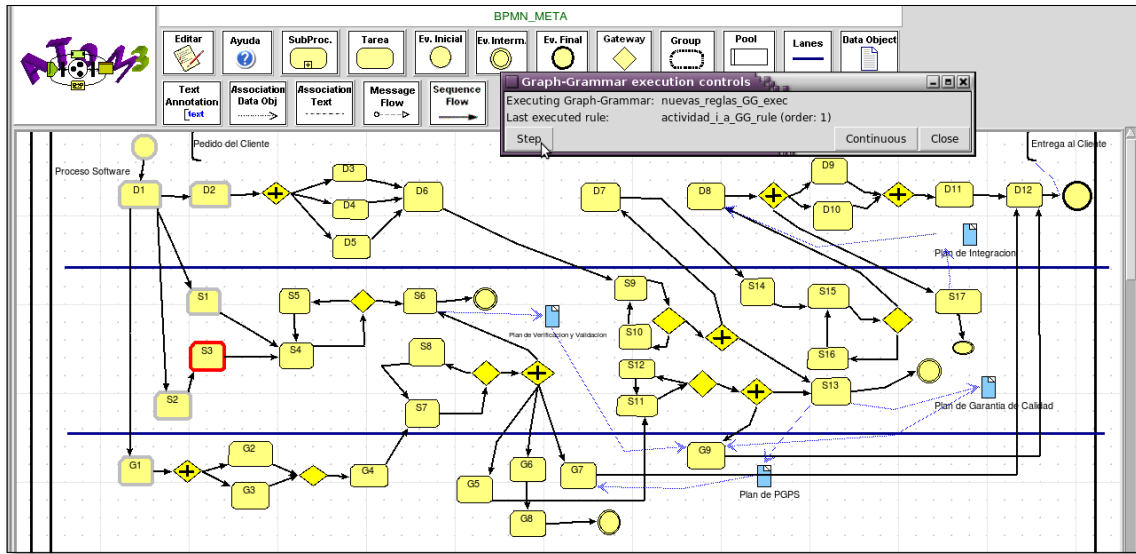


Figura E.11. Inicio de Actividad S3.

✓ **Sim-Step5:** Activación de Gateway Paralelos.

Los Gateway paralelos definen múltiples trayectorias para la activación de varios procesos, en esta etapa se inicializa el Gateway para las Actividades de Desarrollo y en las Actividades de Gestión.

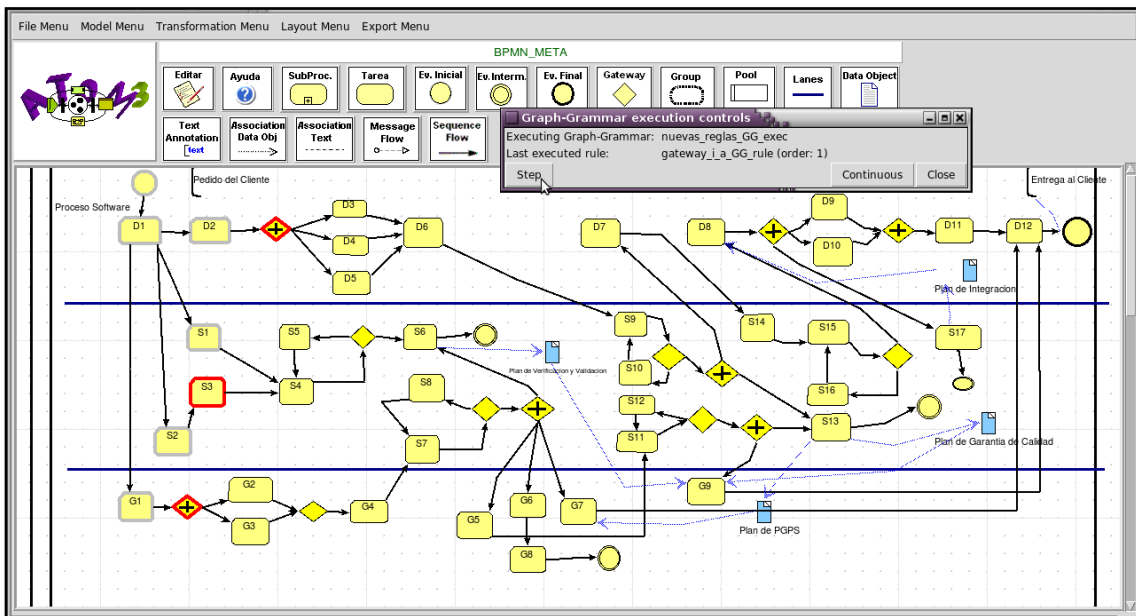


Figura E.12. Inicio de Gateway en Actividades de Desarrollo y Gestión.

✓ **Sim-Fin-Step5:** finalización de los Gateway activados en las actividades de Desarrollo y en las de Gestión

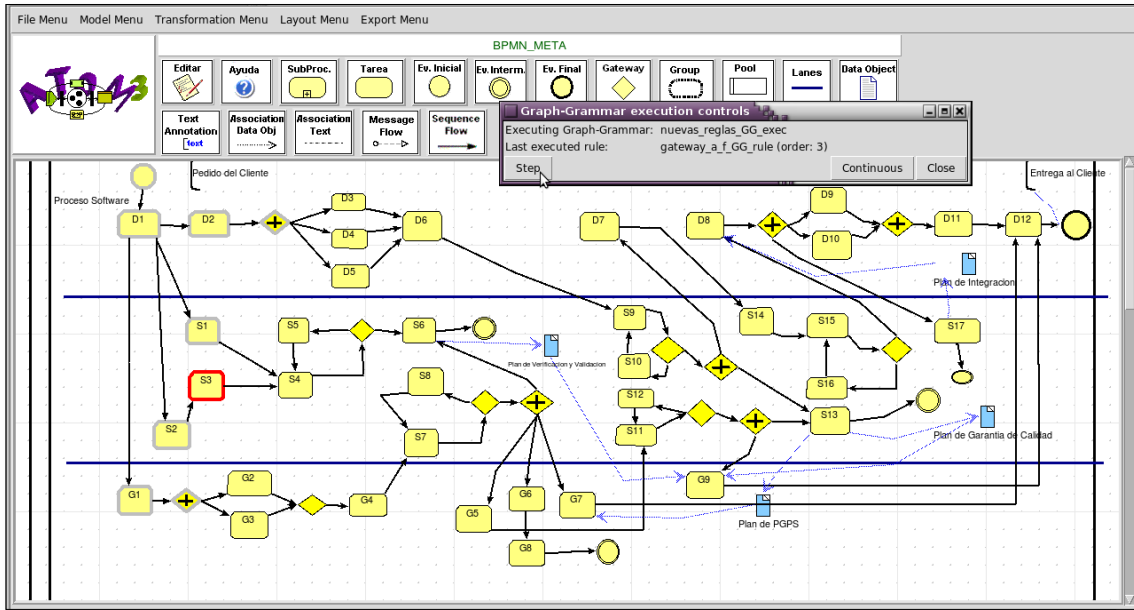


Figura E.13. Fin de ejecución de Gateway en Actividades de Desarrollo y Gestión.

✓ **Sim-Step6:** se inician las actividades paralelas de :

*Actividades de Desarrollo:*

- D3 = Realizar Diseño preliminar.
- D4 = Diseñar base de datos.
- D5 = Diseñar interfaces.

*Actividades de Gestión:*

- G2 = Definir entorno del proyecto.
- G3 = Estimar proyecto.

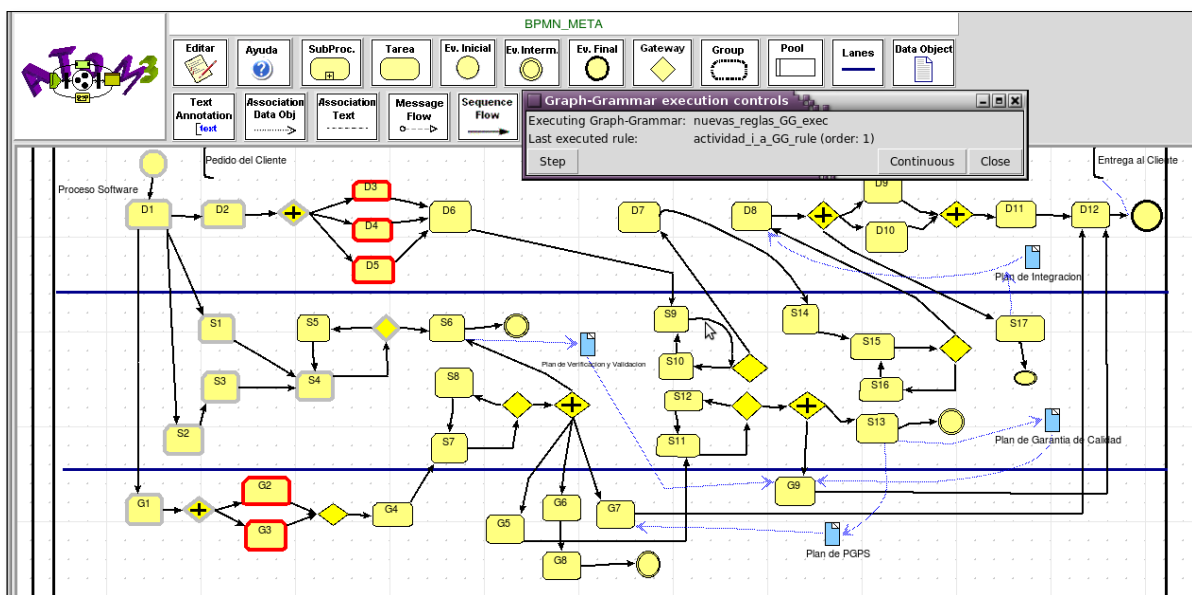


Figura E.14. Inicio de Actividades de Desarrollo y Gestión.

✓ **Sim-Fin-Step6:** Finalización de las Actividades paralelas.

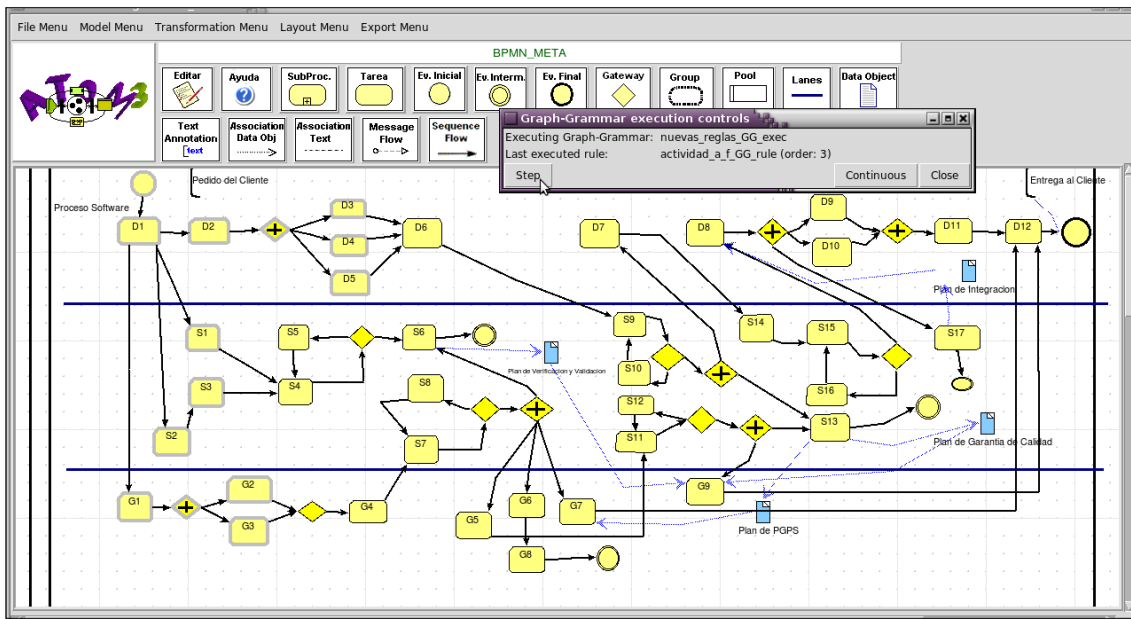


Figura E.15. Fin de ejecución de Actividades paralelas.

✓ **Sim-Step7:** Se activan los siguientes procesos:

**Actividades de Desarrollo** – D6 = Realizar Diseño Detallado.

**Actividades de Soporte** – S4 = Verificar Planes.

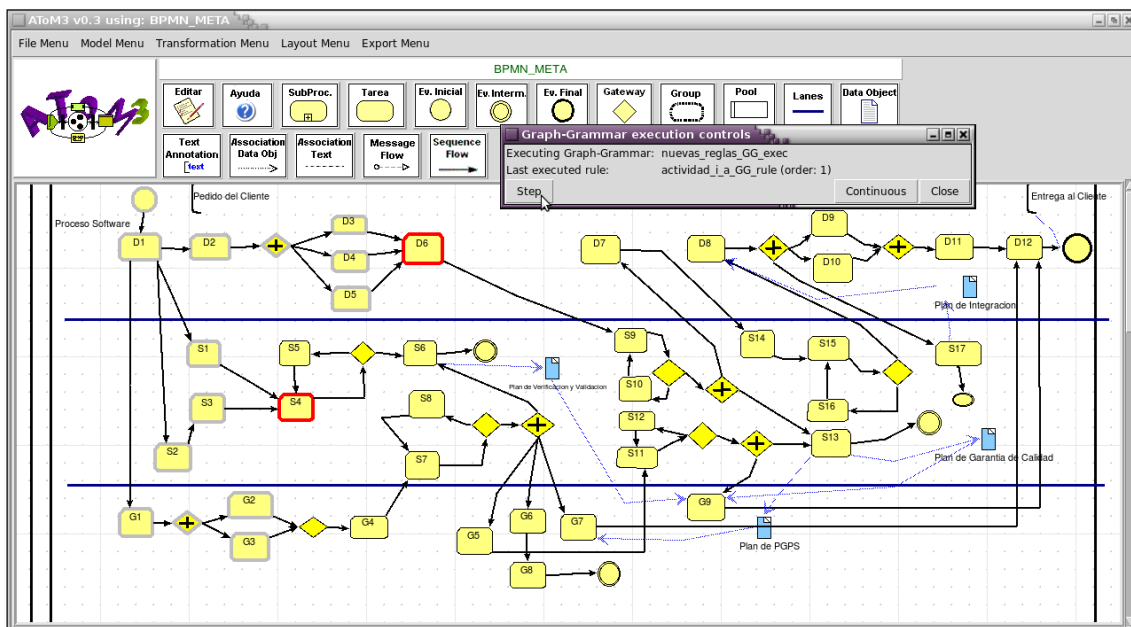


Figura E.16. Inicio de Actividades paralelas en los procesos de Desarrollo y de Soporte.

✓ **Sim-Step8:** Activación de un Gateway de decisión en las Actividades de Gestión.

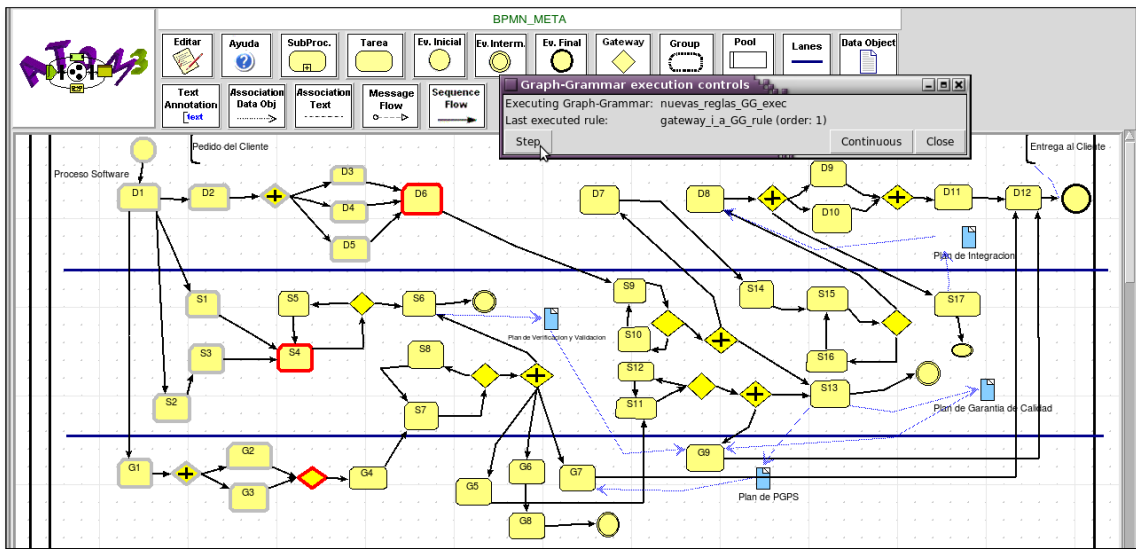


Figura E.17. Inicio Gateway en las Actividades de Gestión.

✓ **Sim-Step9:** Finalizan los proceso

S4=Verificar Planes.

D6=Realizar Diseño Detallado.

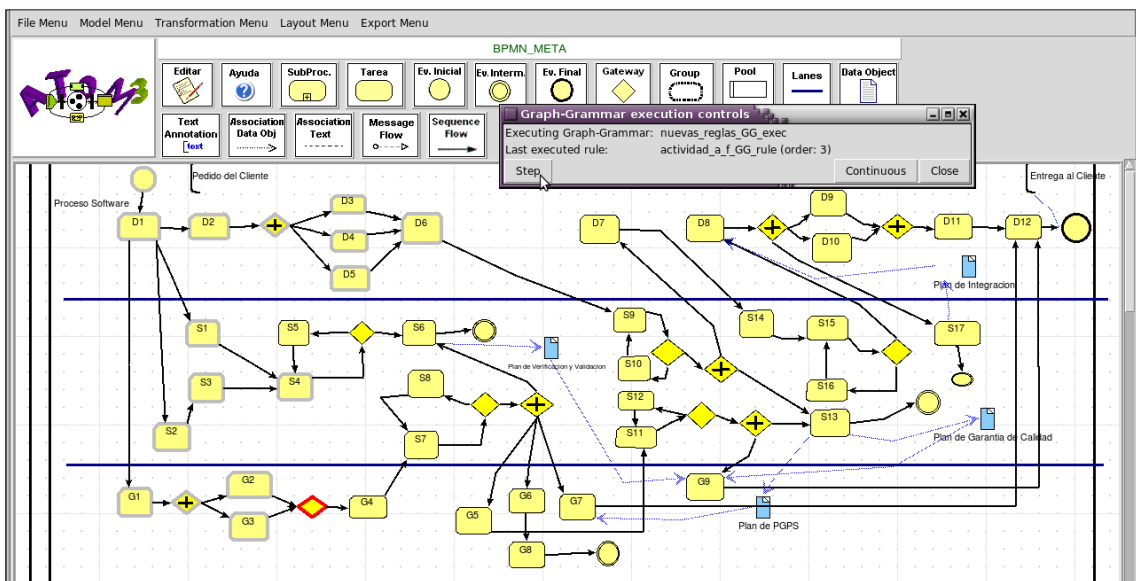


Figura E.18. Finalización de procesos Activos S4 y D6.

- ✓ **Sim-Step10:** Activación del Gateway proveniente de la finalización del proceso S4 = Verificar planes

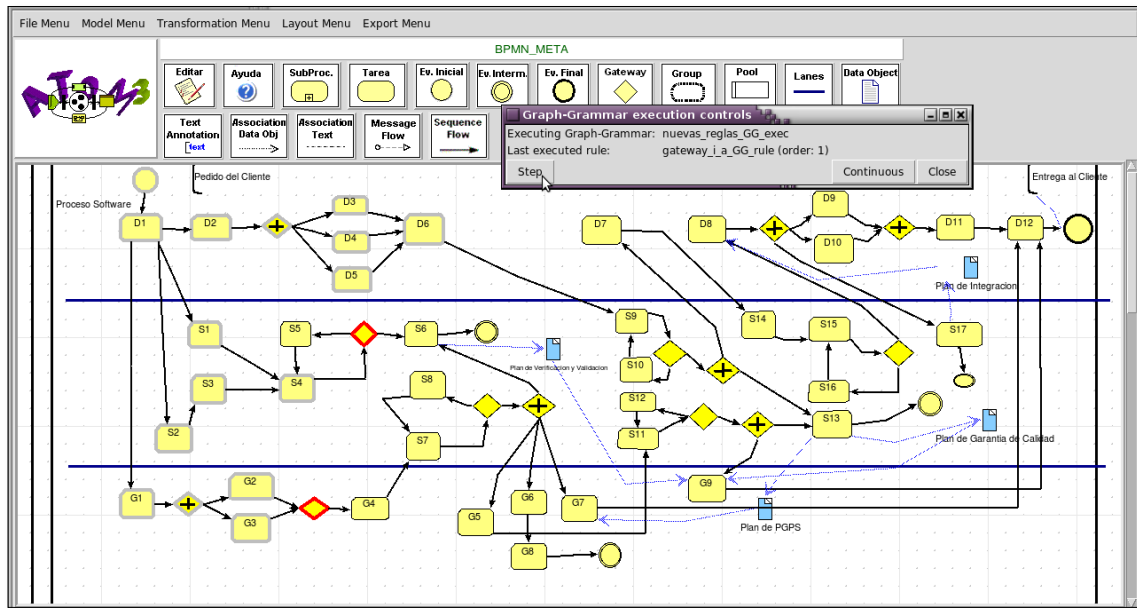


Figura E.19. Activación de Gateway por la finalización del proceso S4.

- ✓ **Sim-Step11:** Se activa el proceso S9=Revisar Diseño, de las Actividades de Soporte.

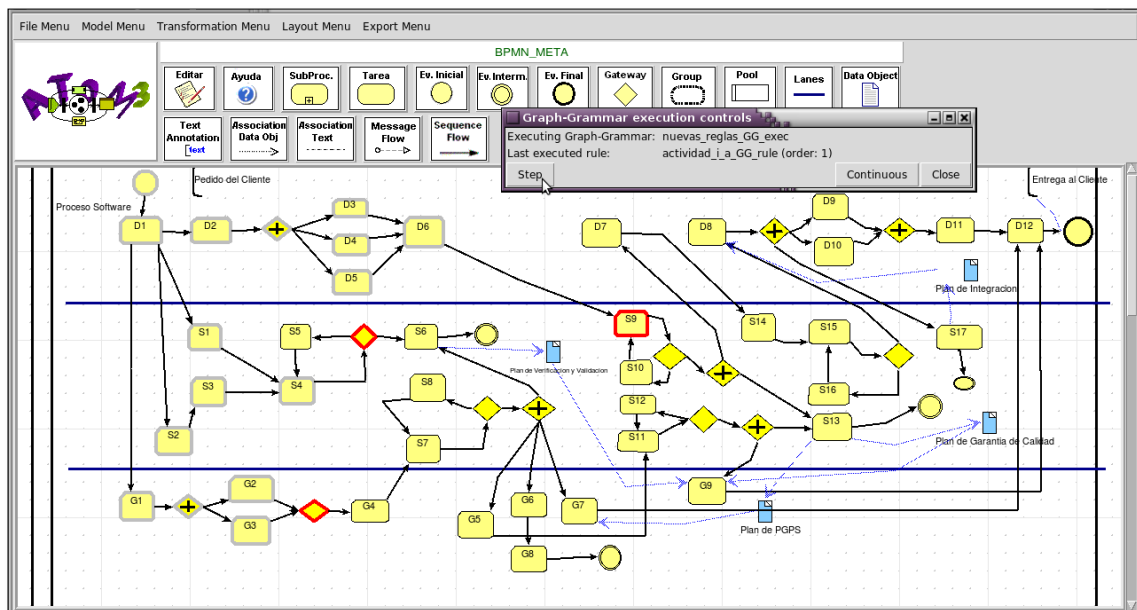


Figura E.20. Activación del proceso S9.

- ✓ **Sim-Fin-Step11:** Finaliza el proceso S9 = Verificar Diseño.

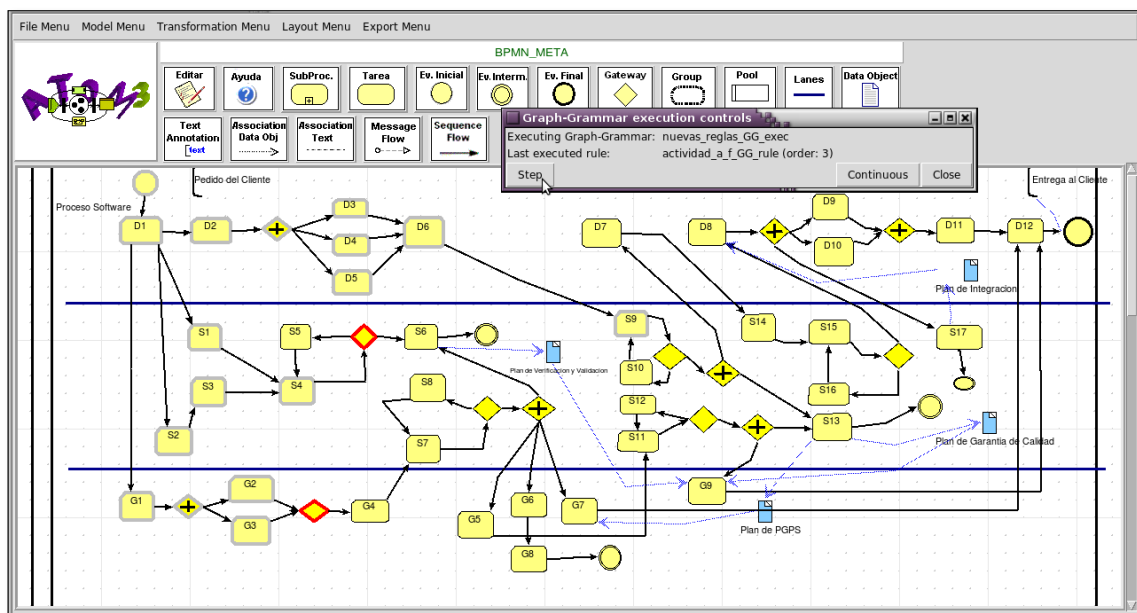


Figura E.21. Activación del proceso S9.

- ✓ **Sim-Step12:** Finalizado el proceso S9, se activa el Gateway correspondiente.

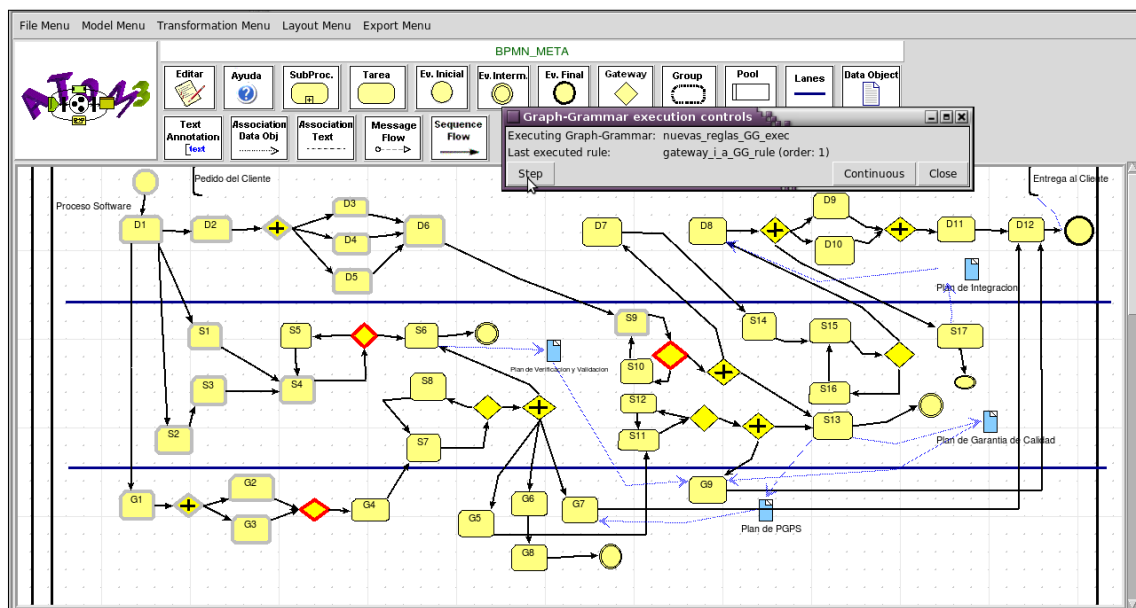


Figura E.22. Finaliza el proceso S9 y se inicia el gateway.

- ✓ **Sim-Step13:** Finalizado el Gateway activado por S4, se activar para el usuario la condición de los caminos posibles, en este caso se deberá elegir entre los procesos:

S5 = Corregir Inconsistencias.

S6 = Guardar cambios.

Elegimos en este caso el proceso S6.

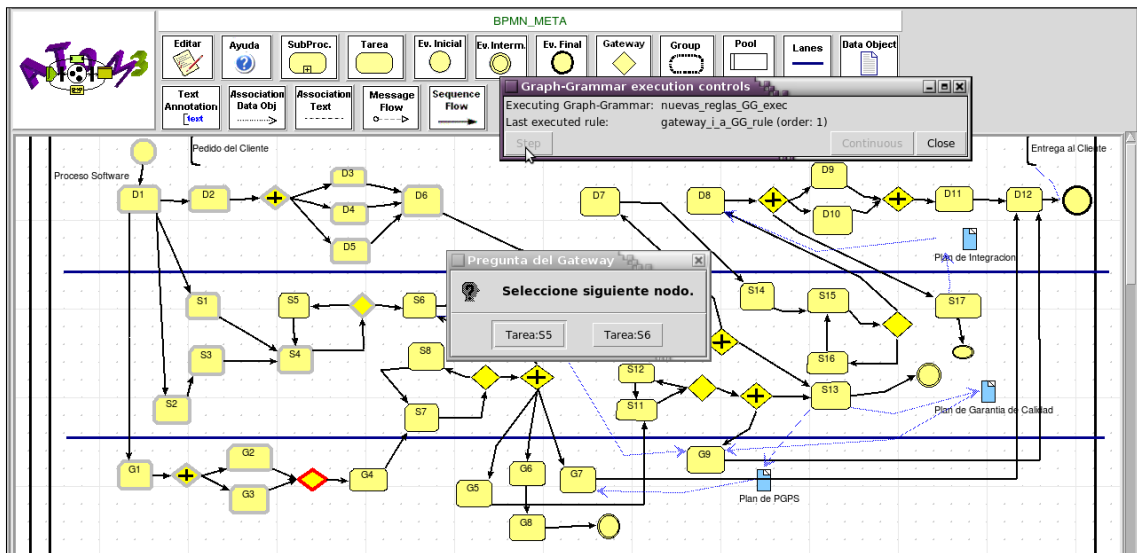


Figura E.23. Elección de procesos por el usuario.

- ✓ **Sim-Step14:** Terminado el proceso S9, se deberá elegir entre:

S10= Corregir Diseño.

**Gateway (Gateway basado en datos).**

Elegimos el Gateway para continuar con la simulación.

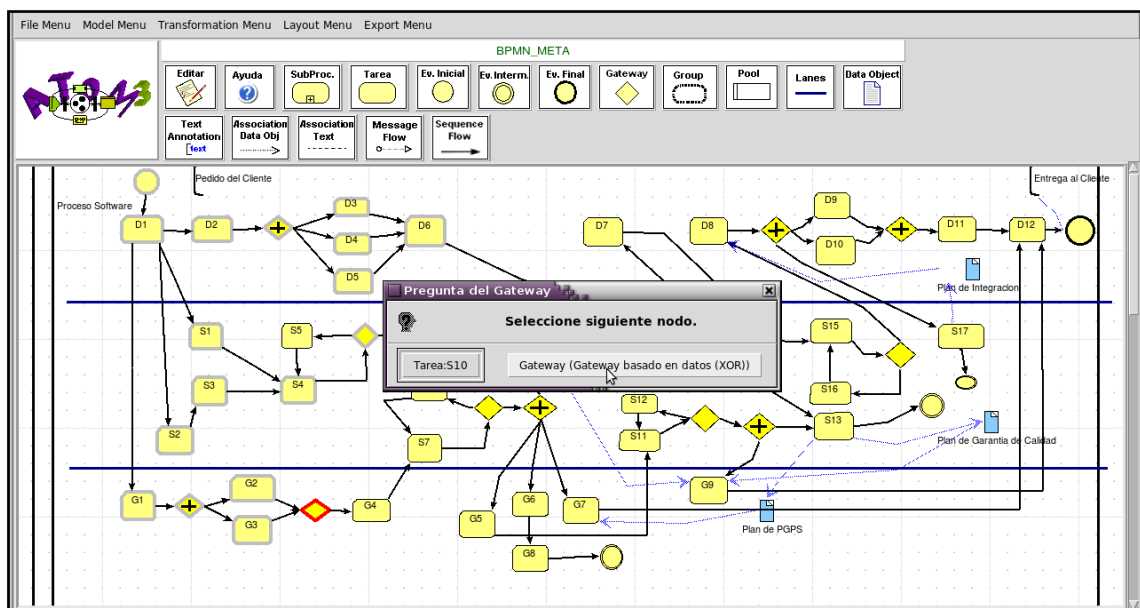


Figura E.24. Elección de procesos por el usuario (elección de Gateway)

✓ **Sim-Step15:** Finaliza el Gateway de Actividades de Gestión y se Activa el proceso:

G4 = Elaborar plan de gestión del proyecto de software.

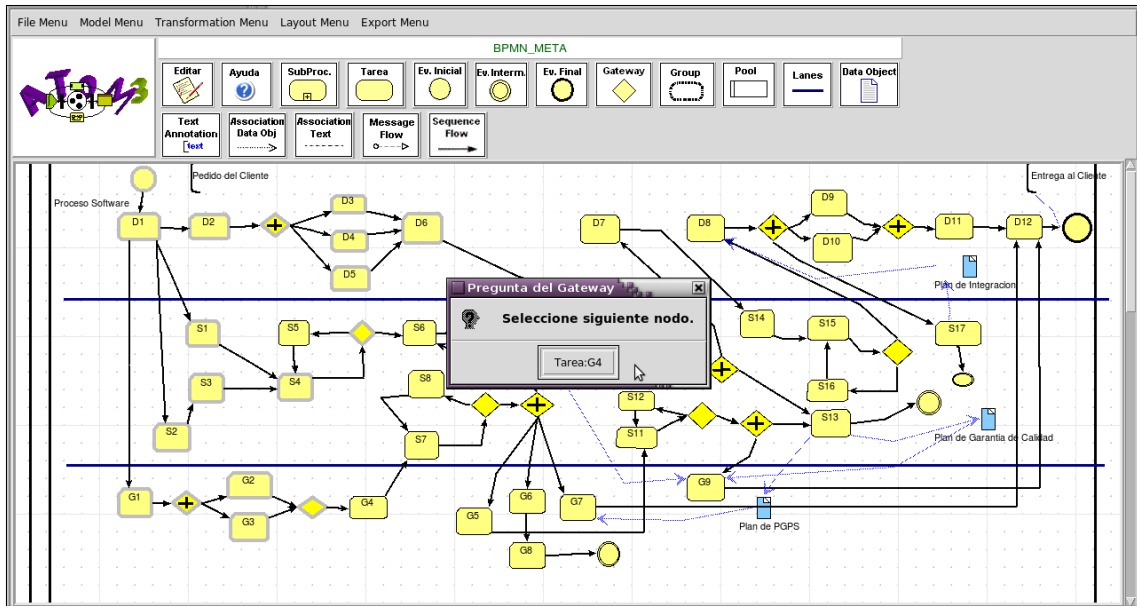


Figura E.25. Selección del proceso G4.

✓ **Sim-Step16:** Se inicia el proceso G4 = Elaborar plan de gestión del proyecto de software.

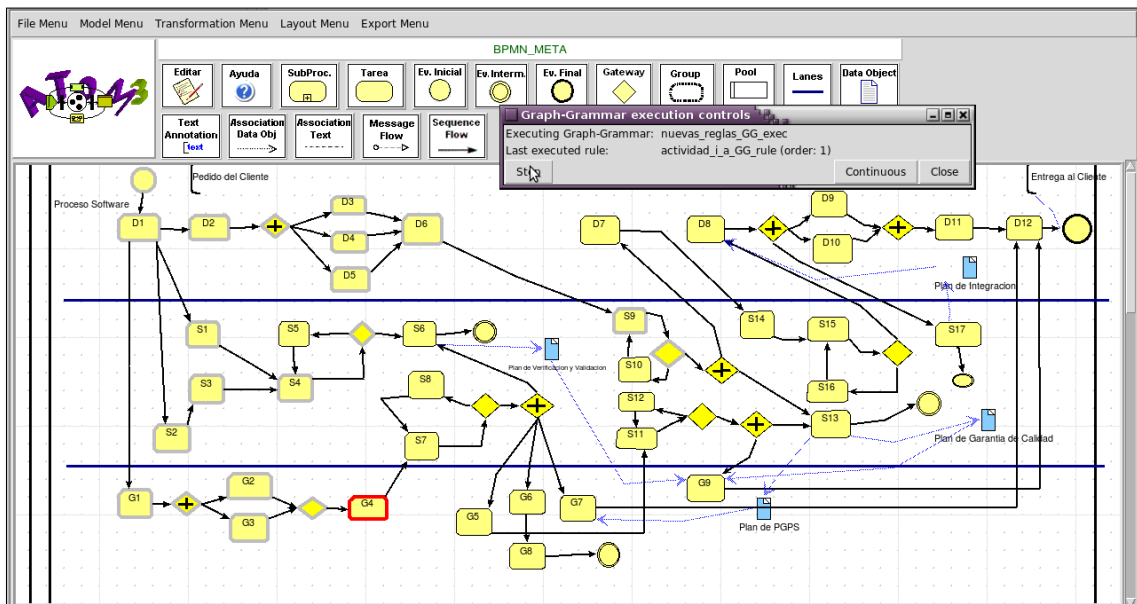


Figura E.26. Inicio de procesos G4.

- ✓ **Sim-Step17:** Se activa el Gateway derivado de la decisión de los procesos S9 y Gateway de eventos de datos.

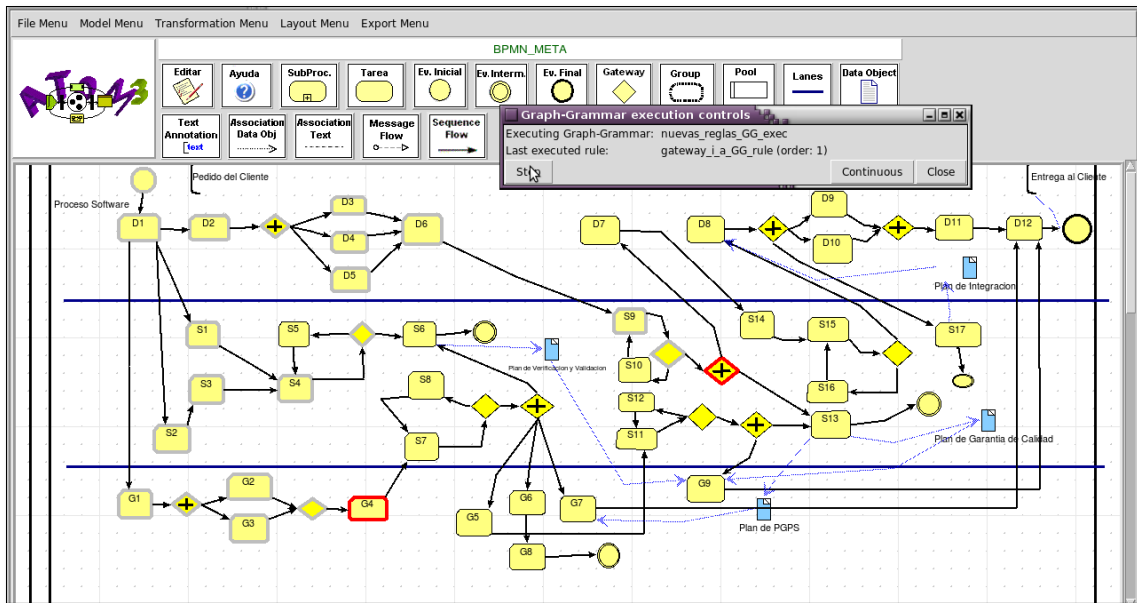


Figura E.27. Inicio de Actividad de Gateway.

- ✓ **Sim-Step18:** Finaliza el proceso G4 de las actividades de gestión.

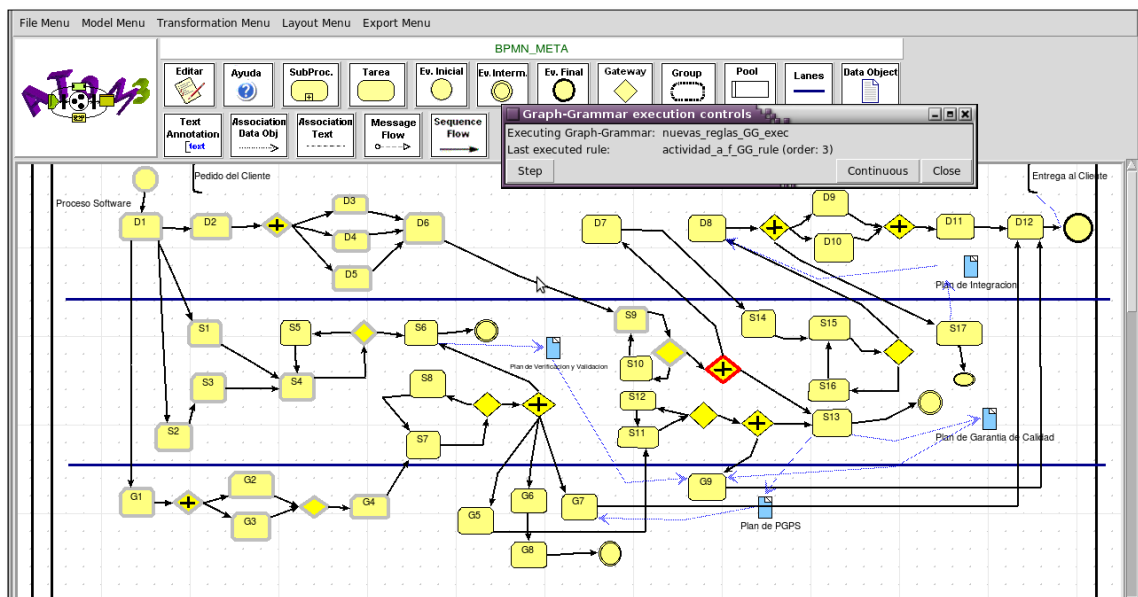


Figura E.28. Finalización de la Actividad G4.

- ✓ **Sim-Step19:** Por la finalización del proceso G4, se activa el proceso de las Actividades de Soporte, S7 = Verificar documentos PGPS.

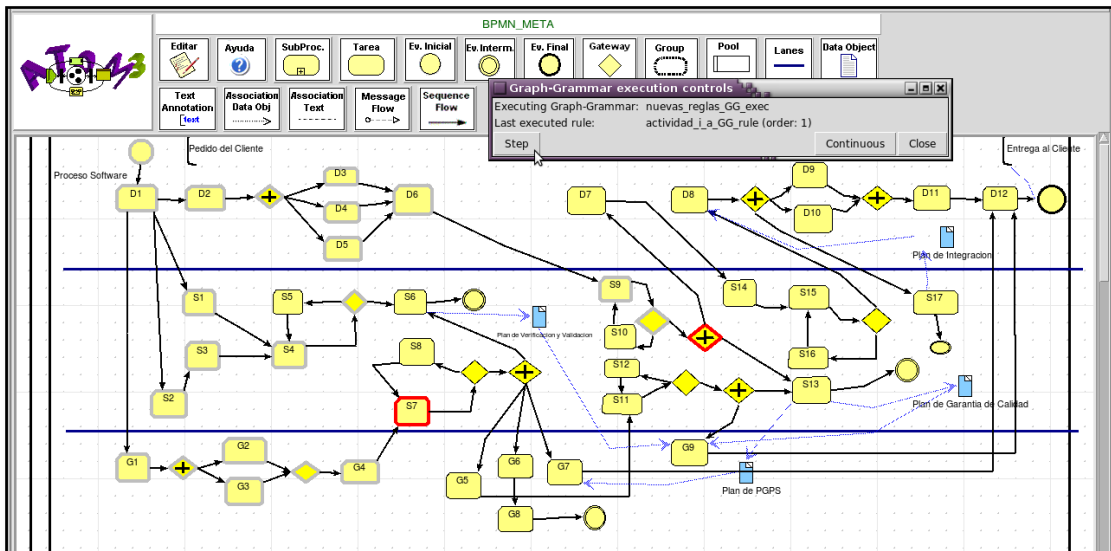


Figura E.29. Iniciación de la Actividad S7.

- ✓ **Sim-Fin-Step19:** Finaliza la actividad S7

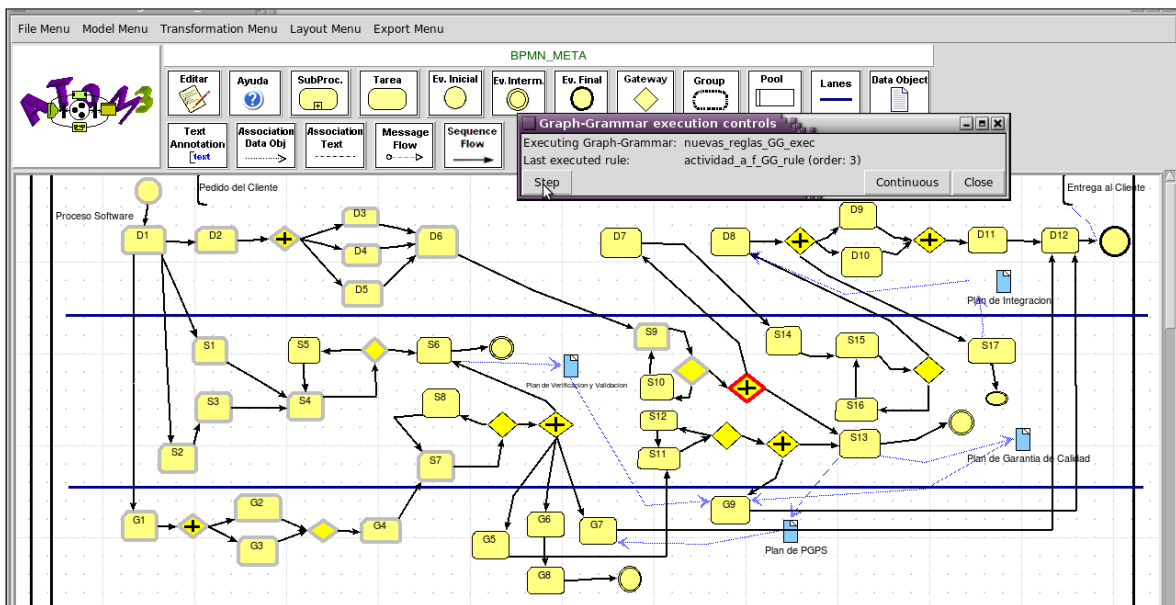


Figura E.30. Finalización de la Actividad S7.

✓ **Sim-Step20:** Se activa el Gateway de decisión en donde se deberá seleccionar una de las opciones:

- S8 = Corregir Inconsistencias del PGPS.
- Gateway de datos.

En nuestra simulación elegiremos el Gateway de Datos.

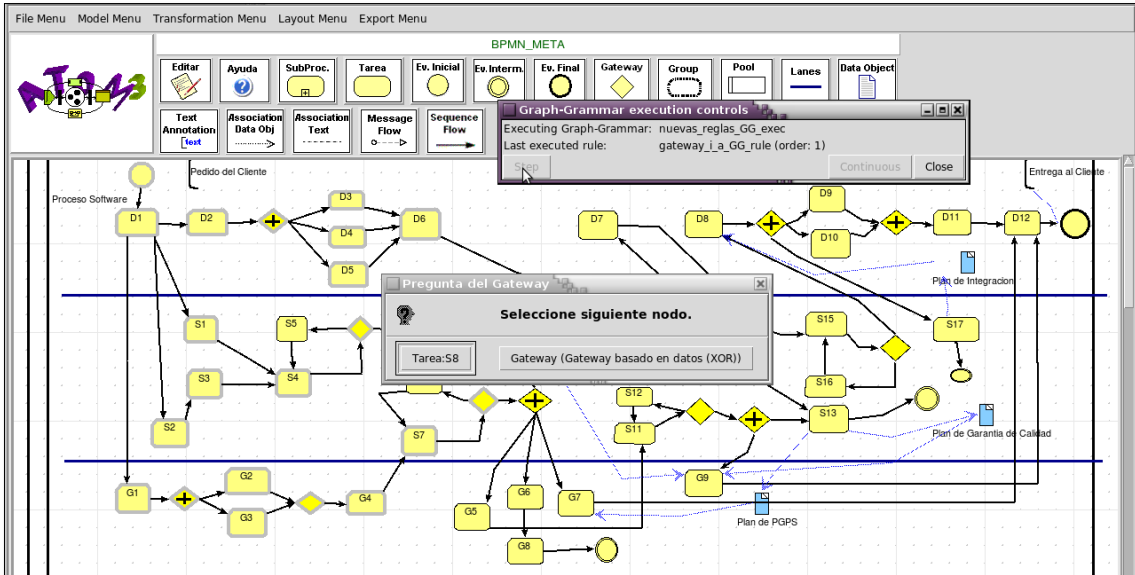


Figura E.31. Inicio de la ejecución de Gateway.

✓ **Sim-Step21:** Seleccionado el Gateway este se activa.

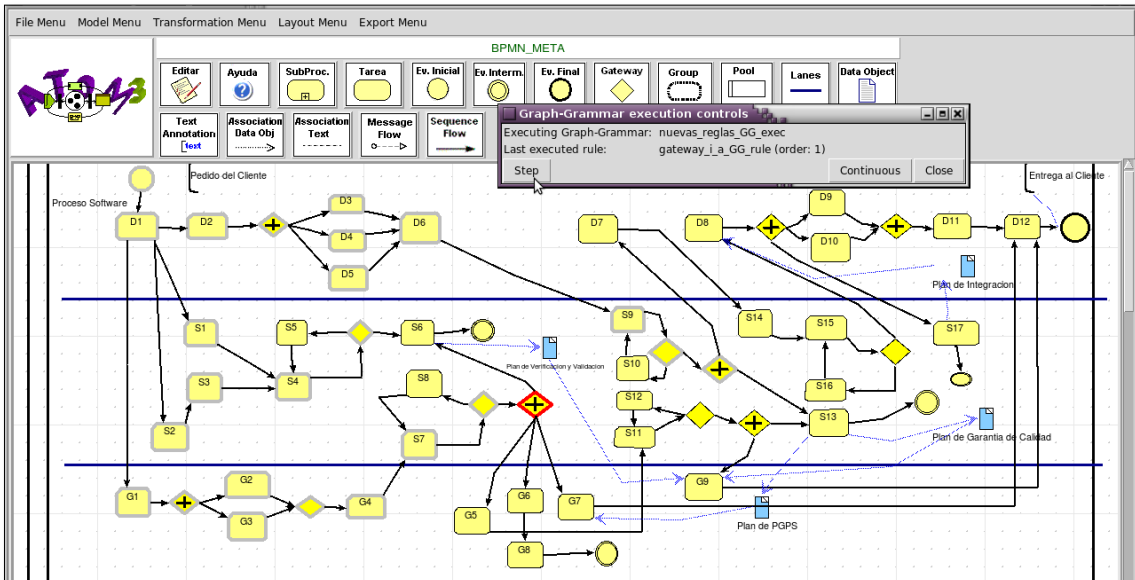


Figura E.32. Inicio de la ejecución de Gateway.

✓ **Sim-Step22:** Se activa el proceso de la Actividad de Desarrollo:

D7 = Crear Código.

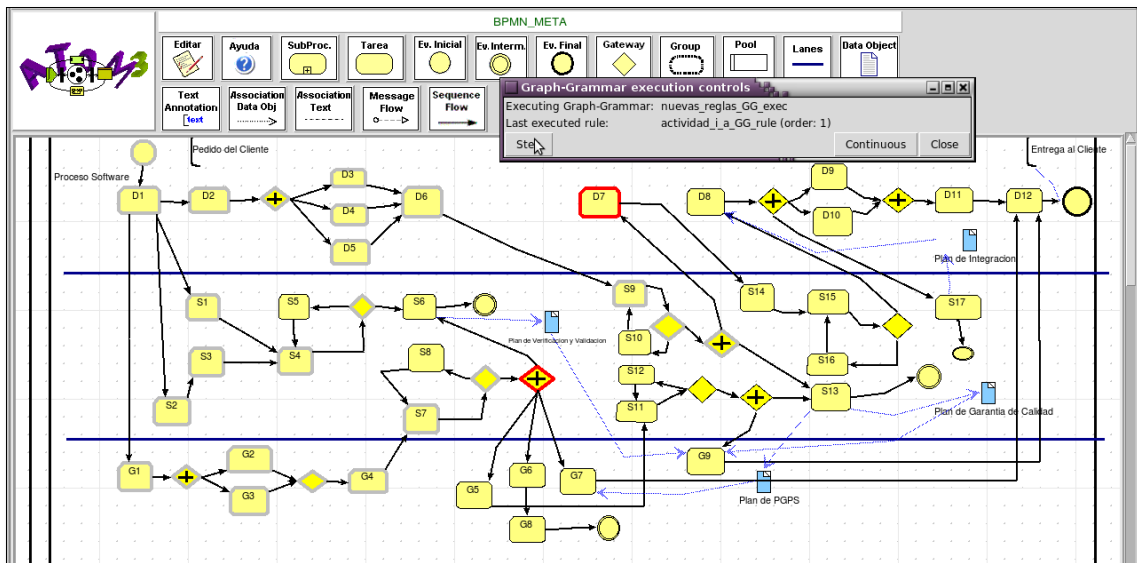


Figura E.33. Inicio de la Actividad D7.

✓ **Sim-Step23:** Finaliza el proceso D7 = Crear Código.

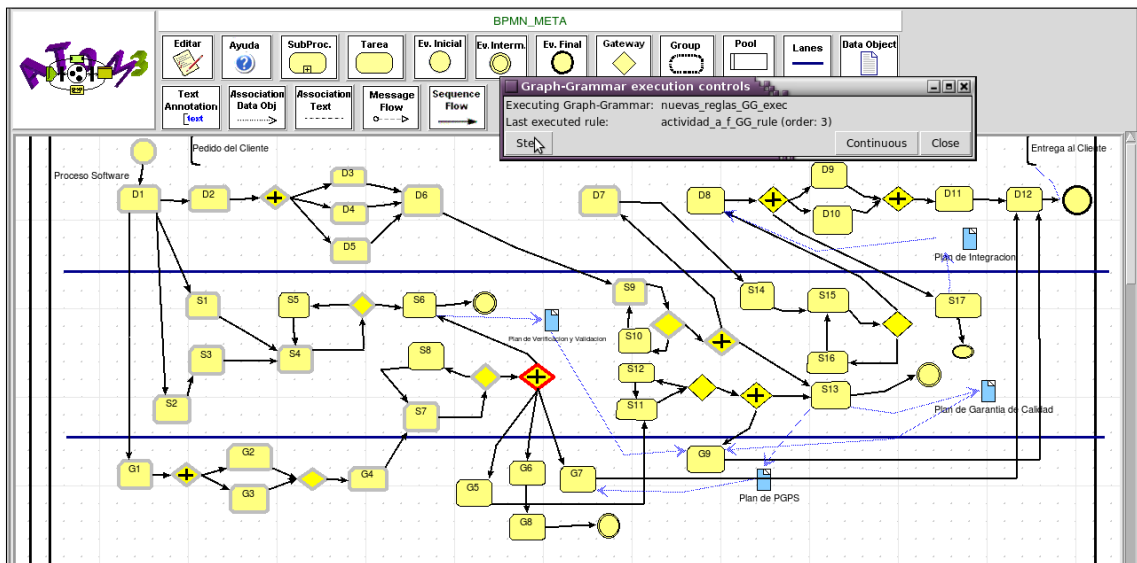


Figura E.34. Inicio de la ejecución del Gateway.

✓ **Sim-Step24:** Se activa el proceso de Gestión de Soporte

S14 = Recibir código.

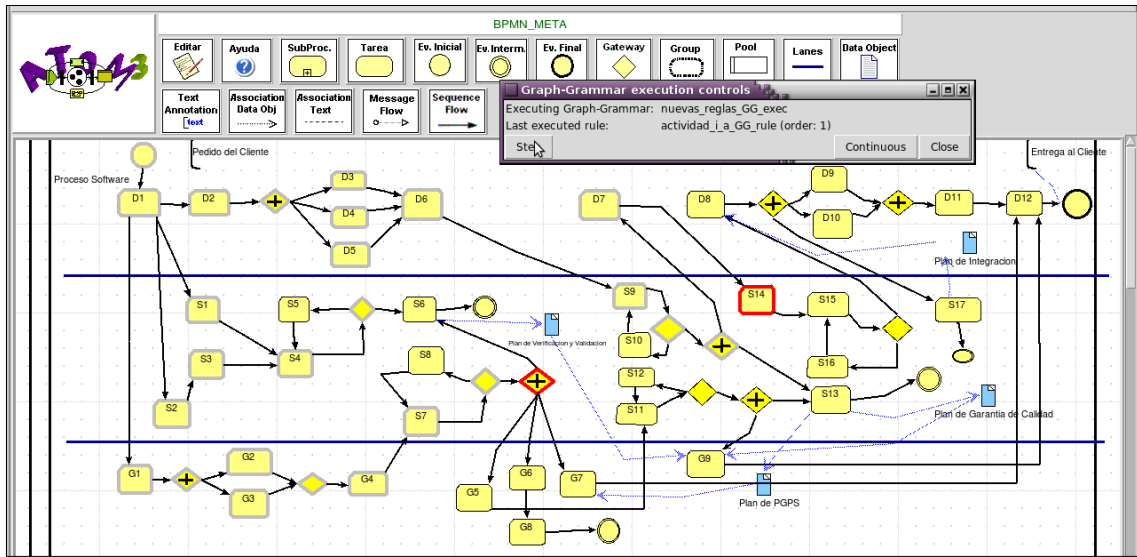


Figura E.35. Inicio de la Actividad S14.

✓ **Sim-Step25:** Finalizado el proceso S14 = Recibir código, se activa el proceso S15=Verificar código.

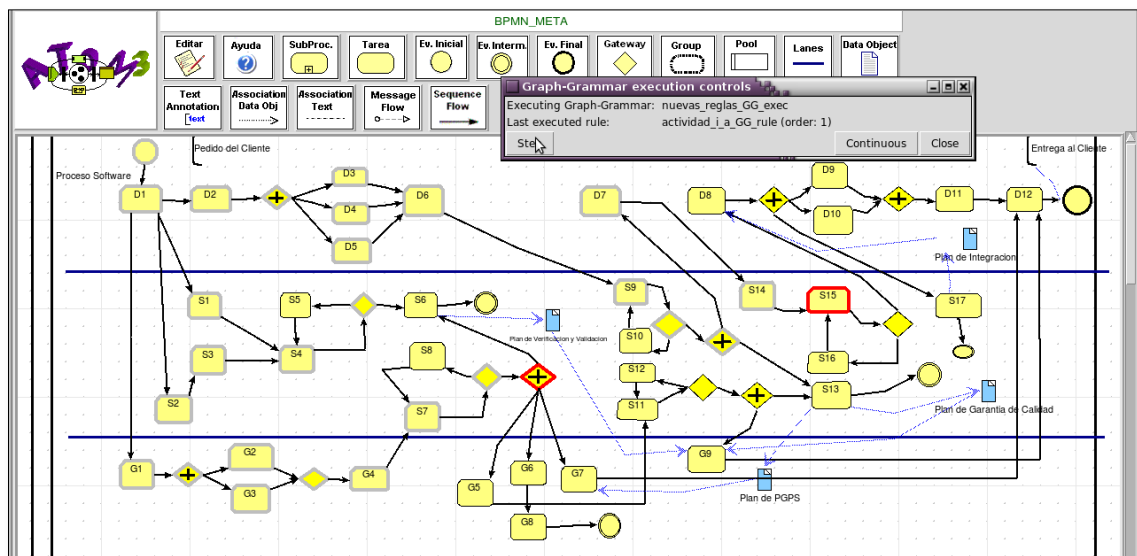


Figura E.36. Inicio de la Actividad S15.

- ✓ **Sim-Step26:** Finaliza el proceso de Gateway de Datos.

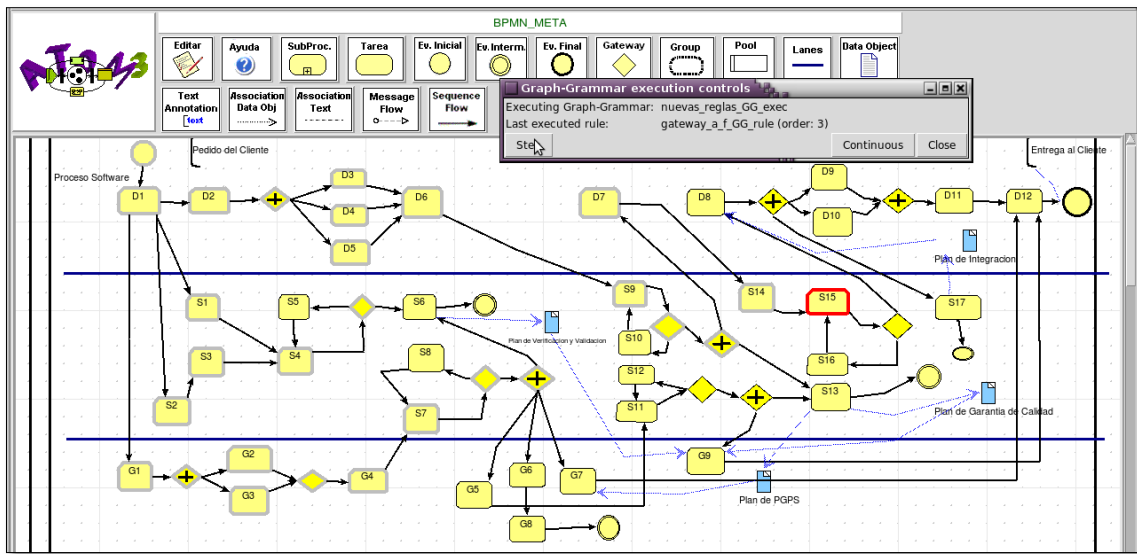


Figura E.37. Finalización de la ejecución del Gateway.

- ✓ **Sim-Step27:** Finaliza el Gateway de decisión y se activa el Gateway de activación Paralela el cual activara los proceso de Actividades de Gestión

- G5 = Elaborar plan de garantía de calidad.
- G6 = Planificar plan de riesgos y contingencias.
- G7 = Gestionar proyectos.
- S6 = Guardar cambios.

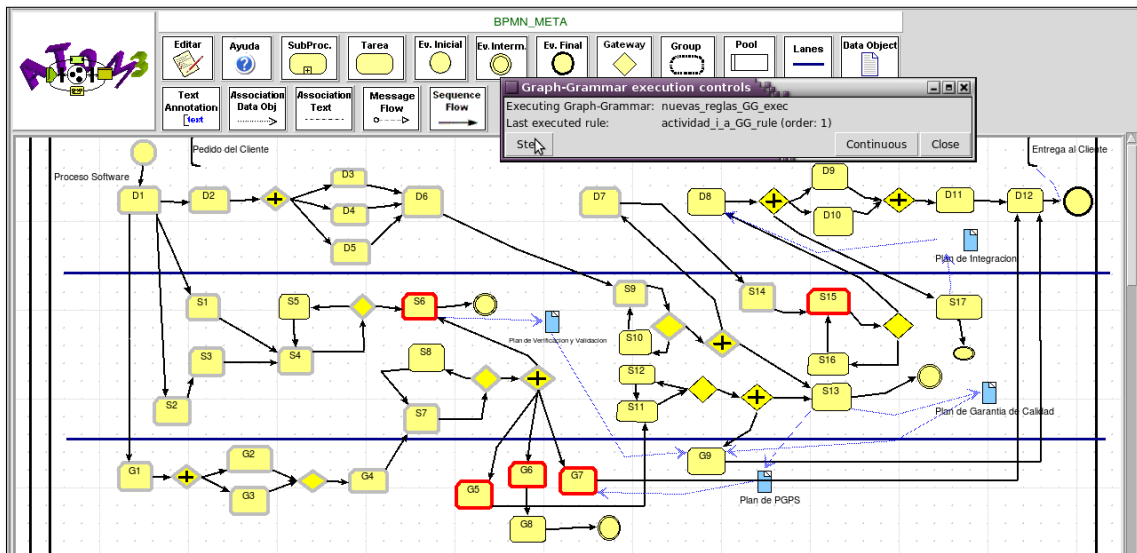


Figura E.38. Inicio de la ejecución del Gateway.

✓ **Sim-Fin-Step27:** Finalizan los procesos:

- G5 = Elaborar plan de garantía de calidad.
- G6 = Planificar plan de riesgos y contingencias.
- G7 = Gestionar proyectos.
- S6 = Guardar cambios.

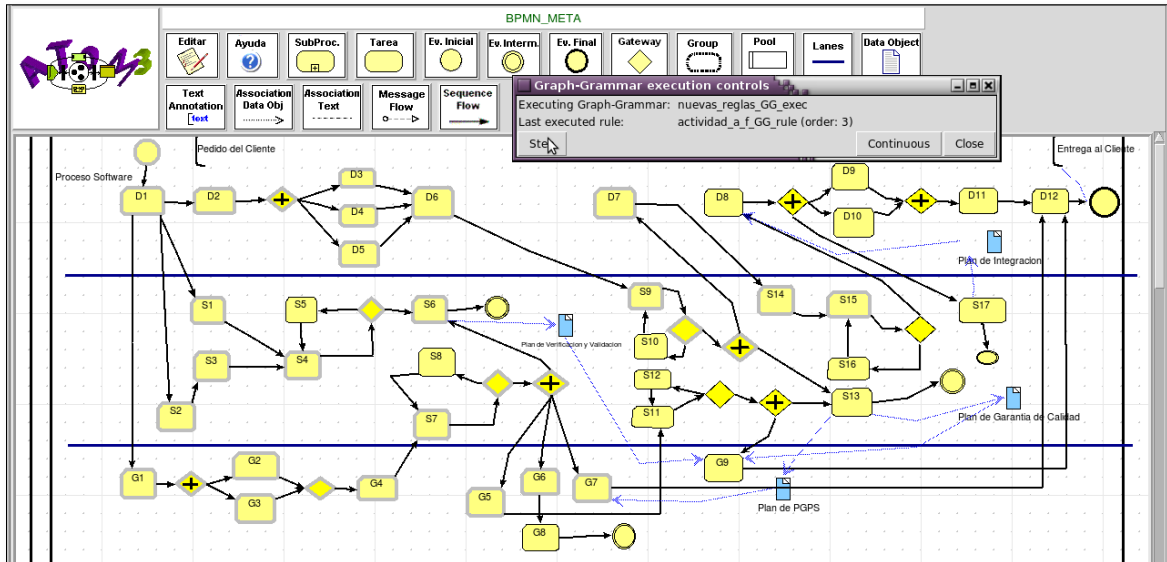


Figura E.39. Finalización de los procesos G5, G6, G7, S6.

✓ **Sim-Step28:** Se activa los proceso G8=Guardar Documentos, por la culminación de G6=Planificar riesgos y contingencias y el proceso S11=Verificar planes se activa por la finalización del proceso G5= Elaborar plan de garantía de calidad.

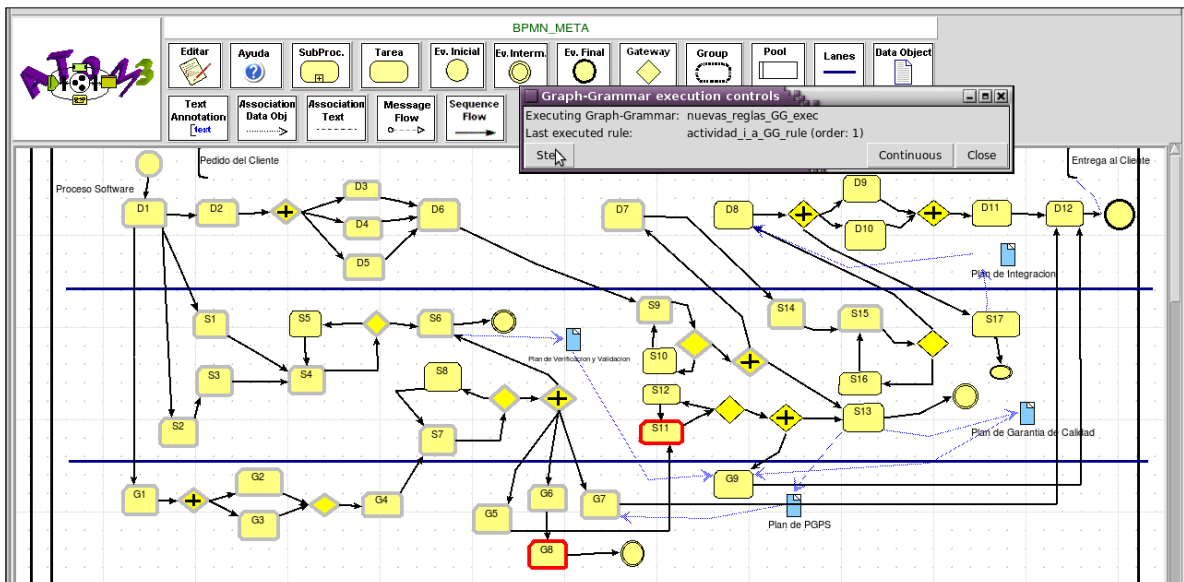


Figura E.40. Finalización de los procesos G5, G6, G7, S6.

- ✓ **Sim-Step29:** Se activa el evento Intermedio por la finalización del proceso S6 = Guardar cambios.

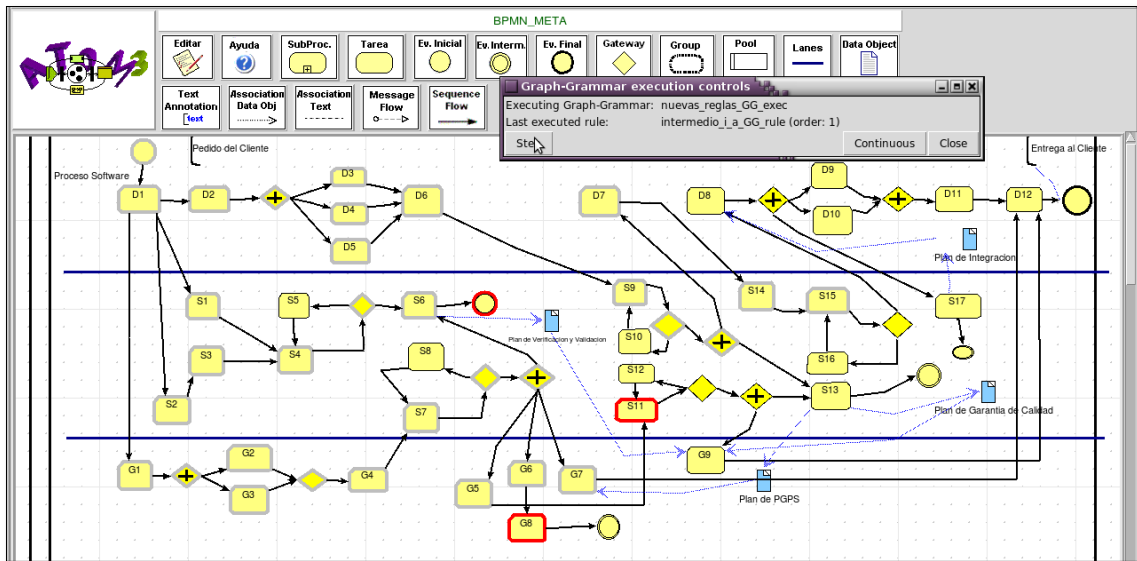


Figura E.41. Inicio de Evento Intermedio y Finalización de la Actividad S6.

**Sim-Step30:** Finaliza el proceso S15 y se activa el Gateway de decisión.

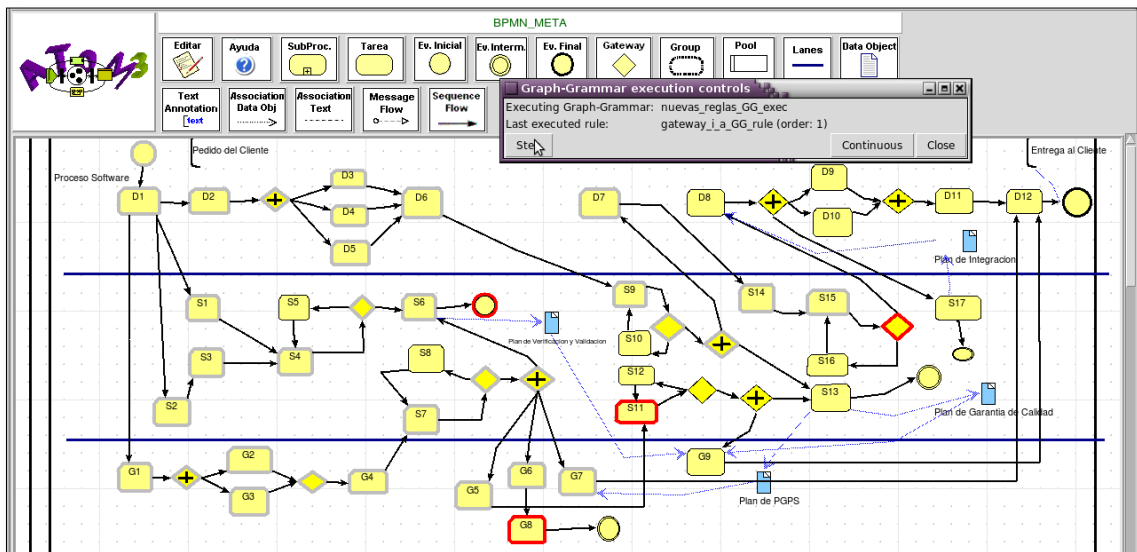


Figura E.42. Inicio de Gateway y Finalización de la Actividad S15.

- ✓ **Sim-Step31:** Finaliza el evento intermedio.

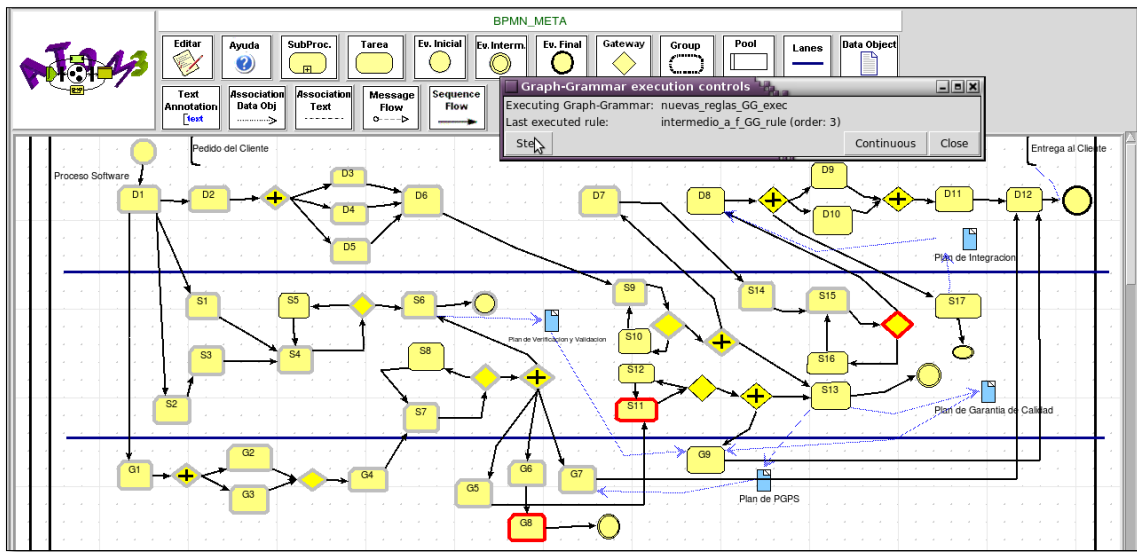


Figura E.43. Finalización del Evento Intermedio.

- ✓ **Sim-Step32:** Se activa el Gateway en donde debemos elegir entre los procesos:

S16 = Corregir errores de código.

D8 = Planificar Integración.

En nuestra simulación elegimos el proceso correspondiente a la actividad de Desarrollo D8=Planificar Integración.

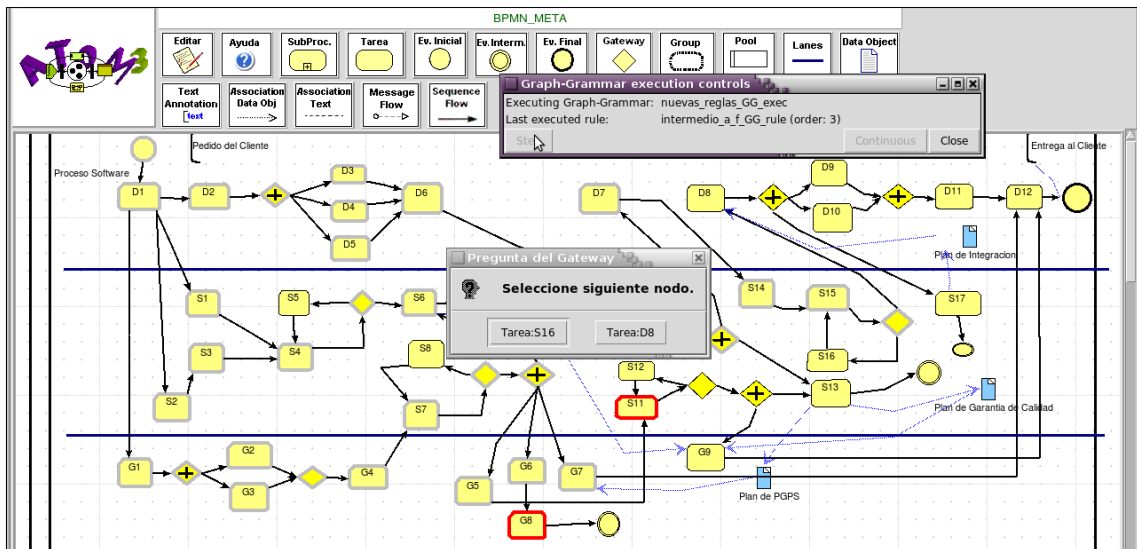


Figura E.44. Inicio de Gateway.

- ✓ **Sim-Step33:** Se Activa el proceso D8 seleccionado en paso anterior.

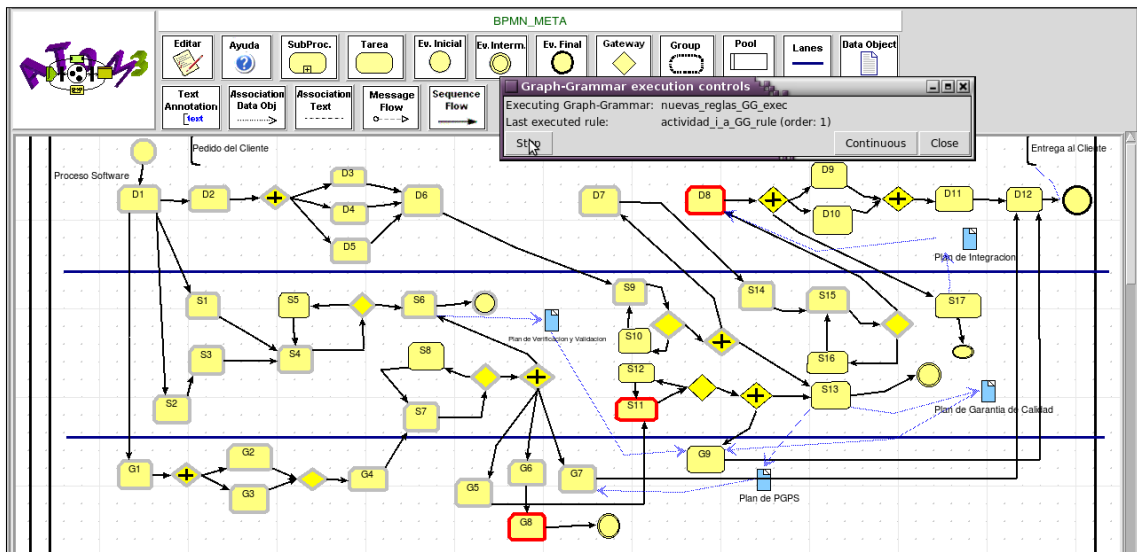


Figura E.45. Inicio de Actividad D8.

- ✓ **Sim-Step34:** Finalizan los procesos activos G8, D8, S11.

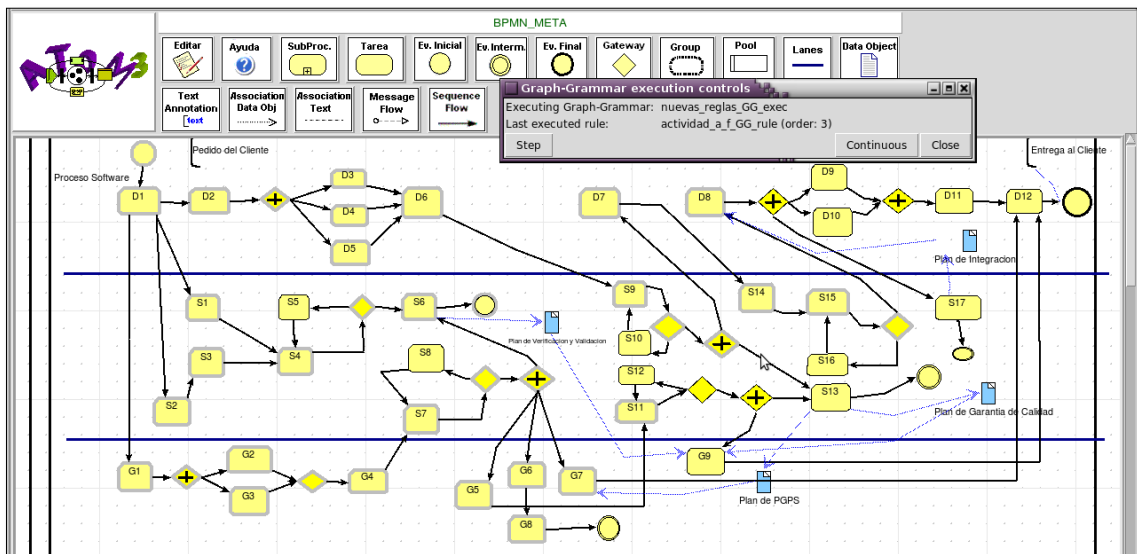


Figura E.46. Finalización de Actividades G8, D8 y S11.

- ✓ **Sim-Step35:** Al finalizar el proceso G8 = Guardar documentos, se activa el evento intermedio.

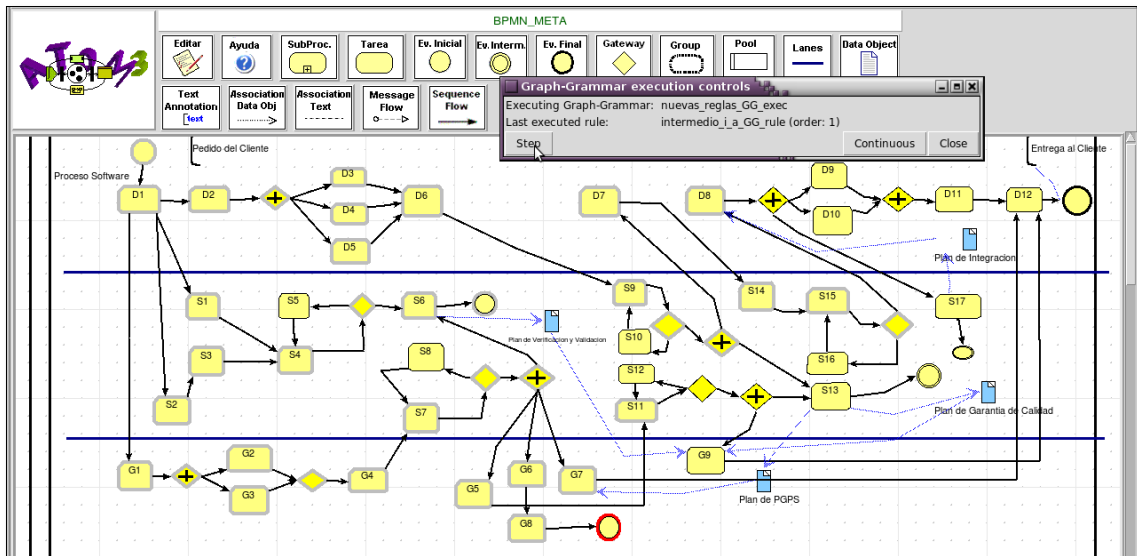


Figura E.47. Inicio de Evento Intermedio.

- ✓ **Sim-Step36:** Se activa los procesos para las Actividades de Soporte, el Gateway de decisión y para las Actividades de desarrollo se activa el Gateway de activación paralela.

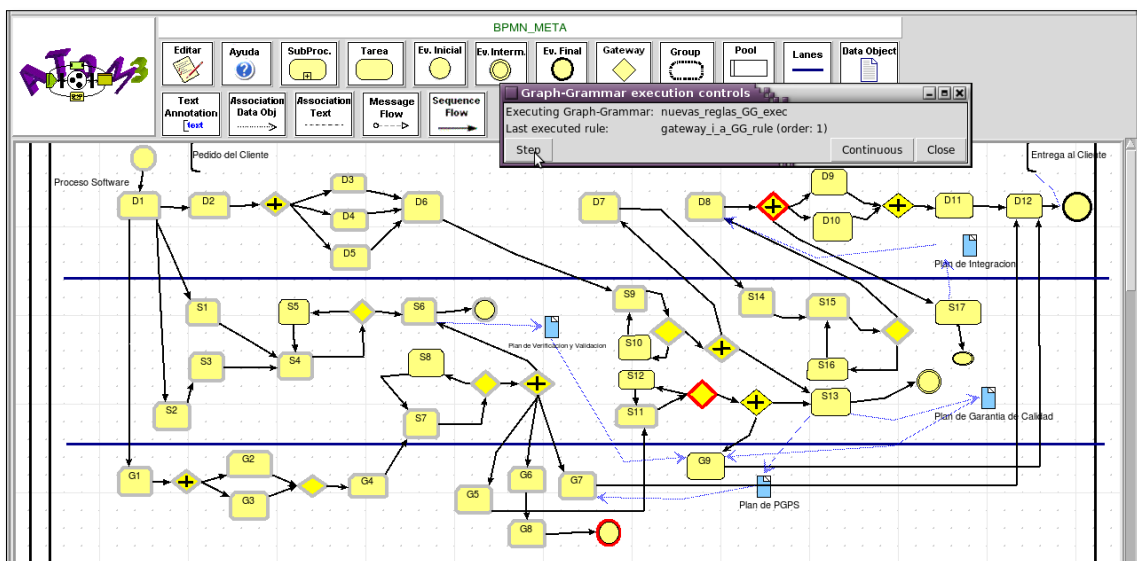


Figura E.48. Inicio de Gateway paralelos.

- ✓ **Sim-Step37:** Finaliza el evento intermedio activado por G8.

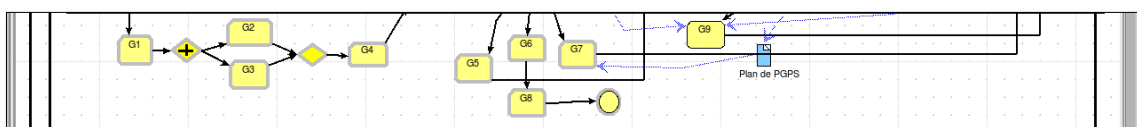


Figura E.49. Finalización de Evento Intermedio.

- ✓ **Sim-Step38:** Se decide entre los proceso S13 = Guardar Cambios y el Gateway de decisión.

En nuestra simulación elegiremos el Gateway.

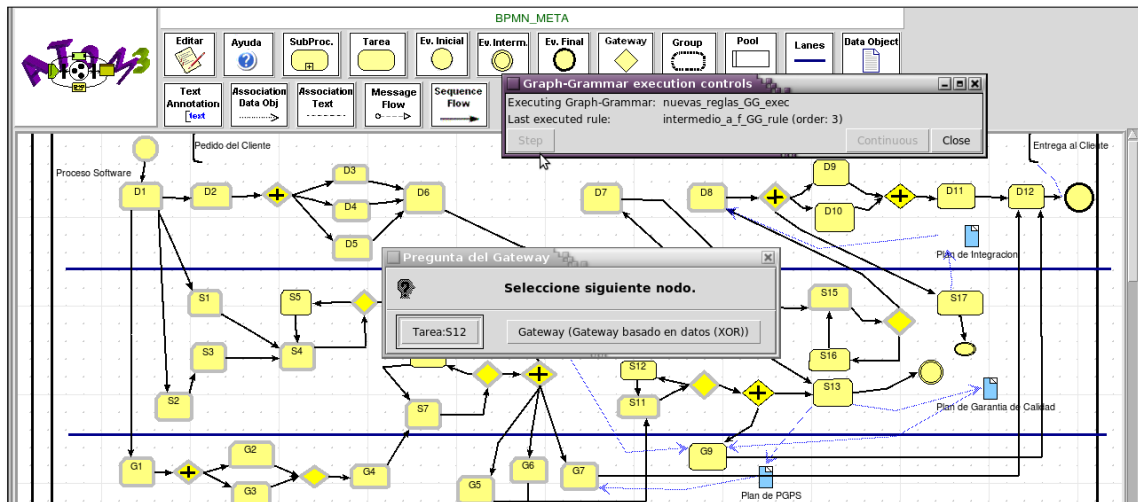


Figura E.50. Inicio de Actividad de Gateway.

- ✓ **Sim-Step39:** Se activan los procesos:

D9 = Crear datos de prueba.

D10= Integrar Módulos.

S17=Guardar cambios

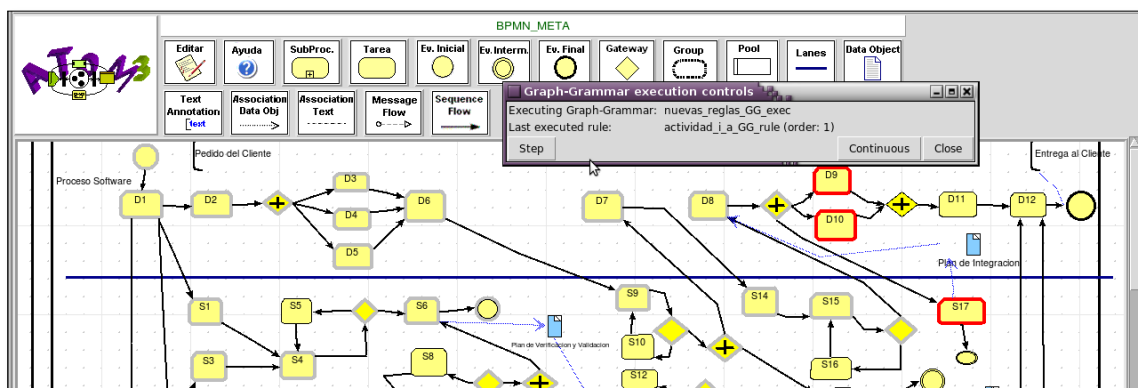


Figura E.51. Inicio de Procesos D9, D10 y S7.

- ✓ **Sim-Step40:** Se activa el Gateway paralelo.

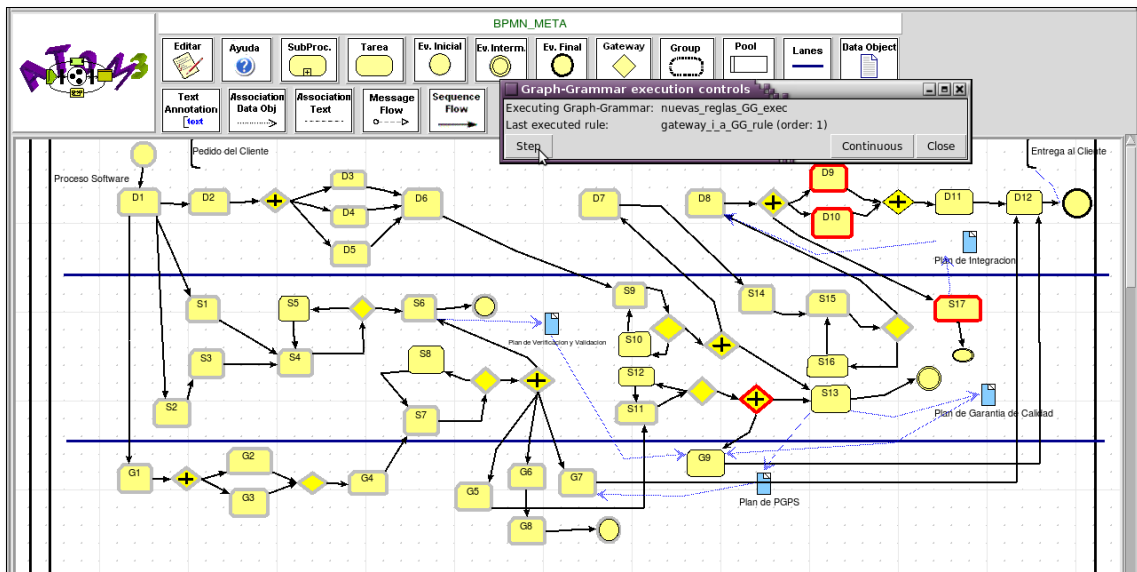


Figura E.52. Inicialización de Gateway Paralelos.

- ✓ **Sim-Step41:** Finalizada la activación del Gateway se da inicio a los procesos:

G9= Gestionar calidad del proyecto.

S13= Guardar Cambios.

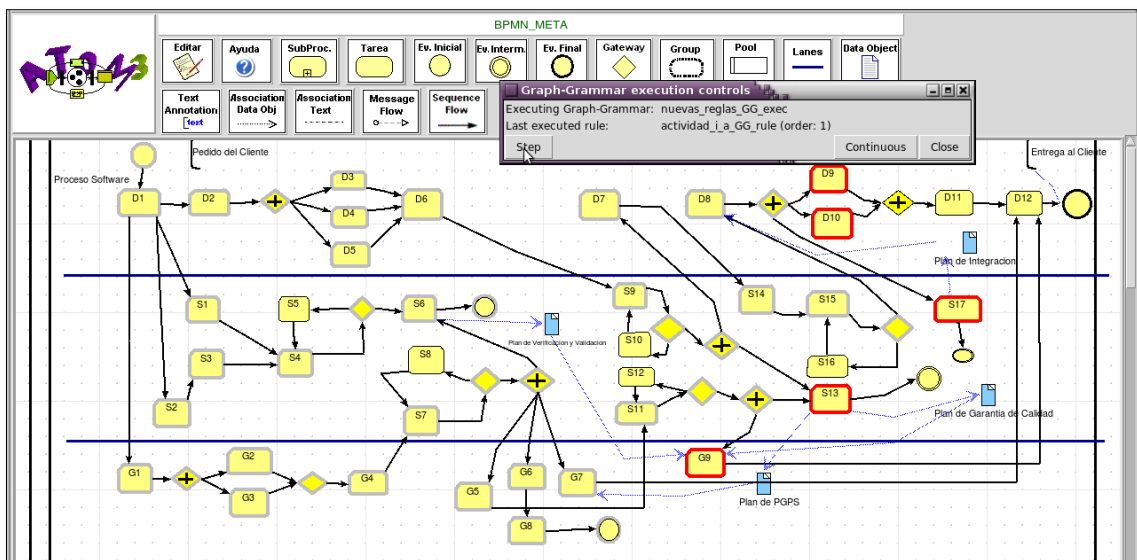


Figura E.53. Finalización de Gateway Paralelos.

- ✓ **Sim-Step42:** Finalizan los procesos activos paralelos, D9,D10, S17, S13, G9.

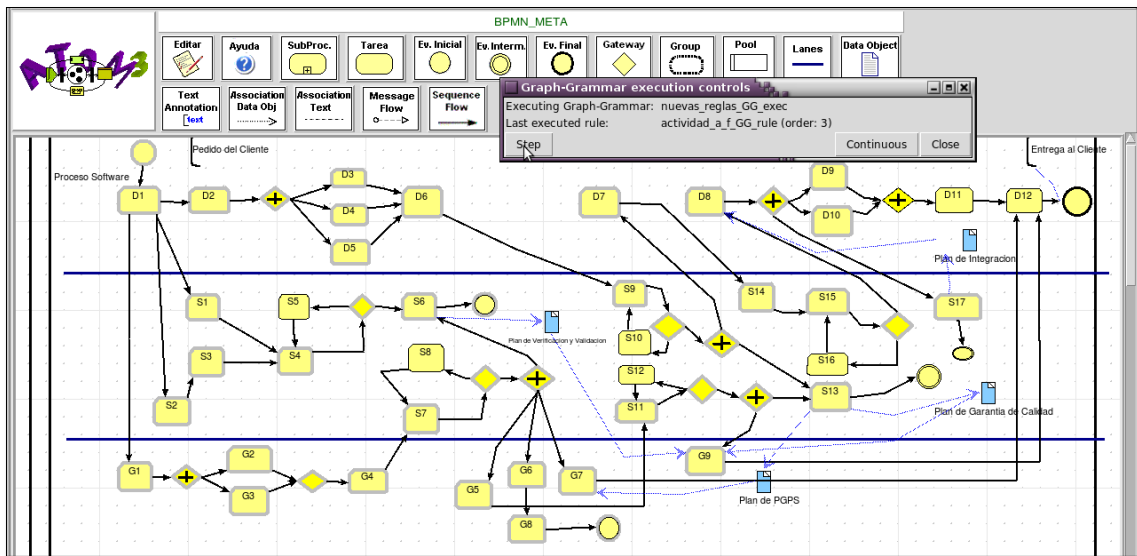


Figura E.54. Finalización de Procesos D9, D10, S17, S13 y G9.

- ✓ **Sim-Step43:** Finalizado S13 y S17, se activan los eventos intermedios.

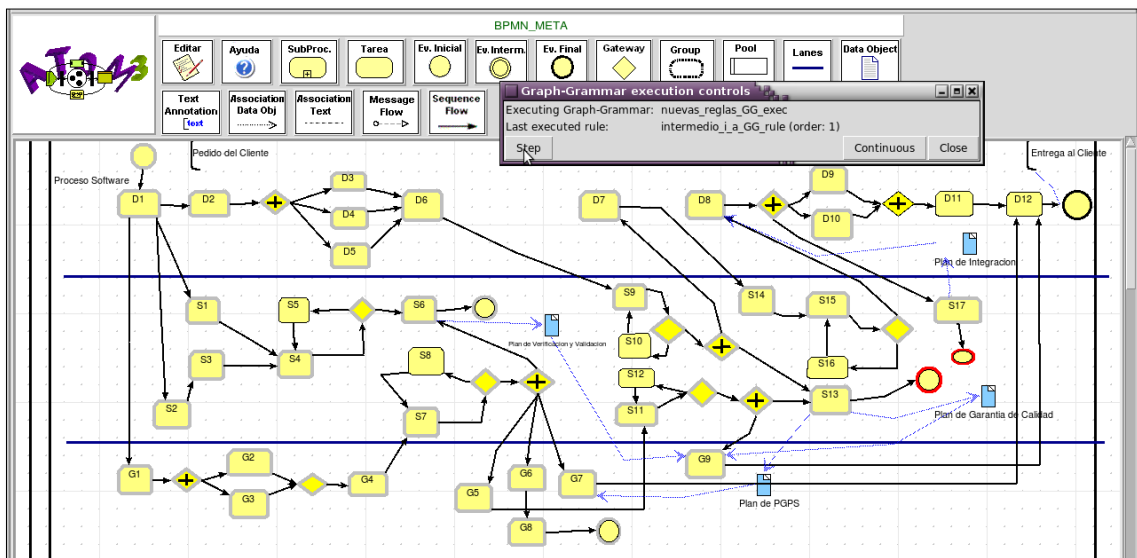


Figura E.55. Inicio de Evento Intermedio.

- ✓ **Sim-Step44:** Finalizado pos procesos D9 D10 se activa el Gateway.

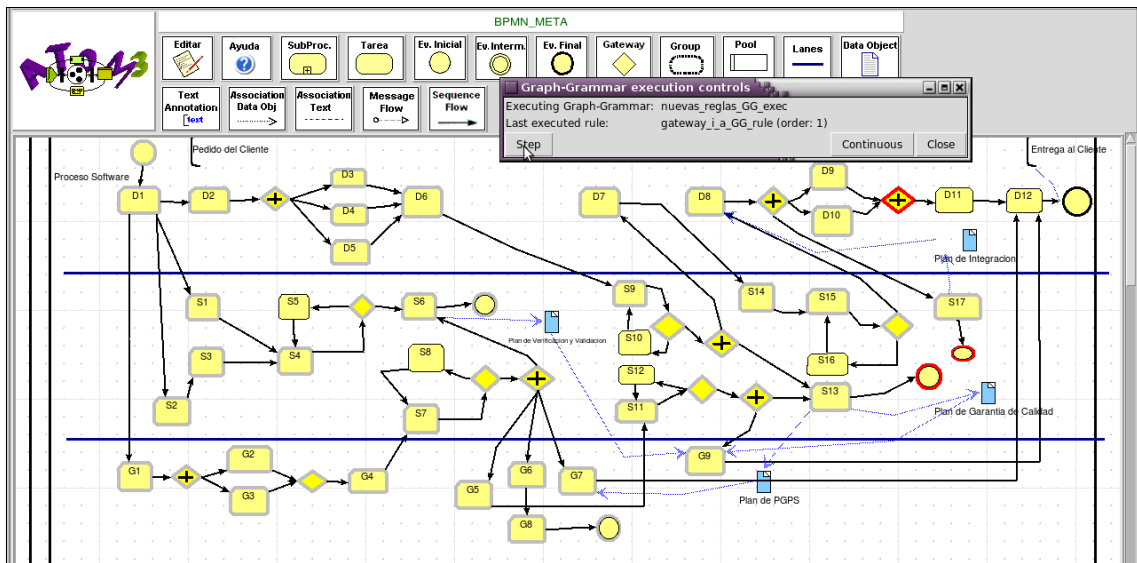


Figura E.56. Finalización de Procesos y Gateway.

- ✓ **Sim-Step45:** Finalizan los eventos intermedios de S17 y S13.

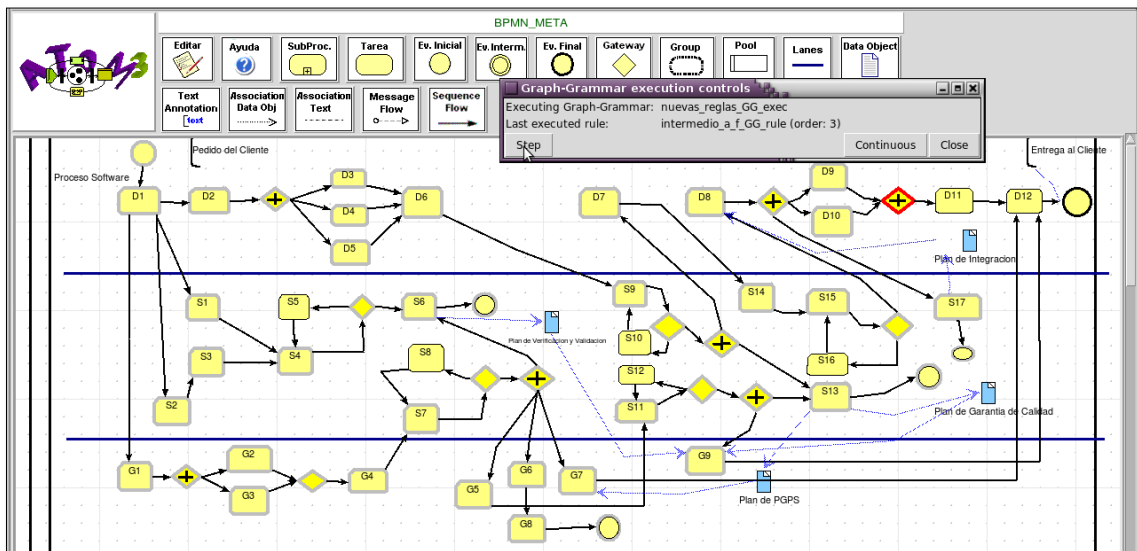


Figura E.57. Finalización de Eventos Intermedios.

- ✓ **Sim-Step46:** Finaliza el Gateway activado en el paso anterior.

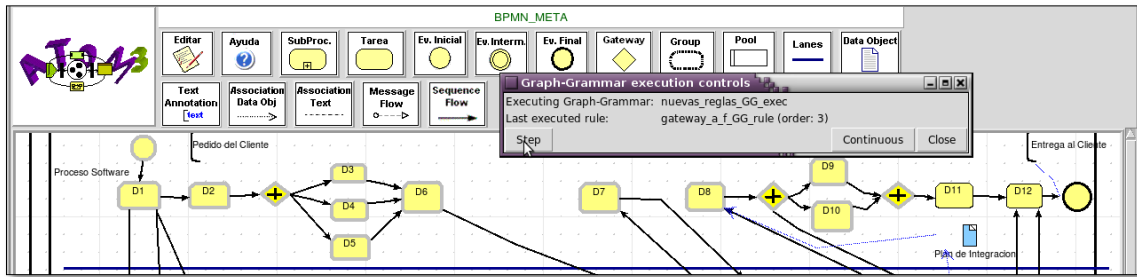
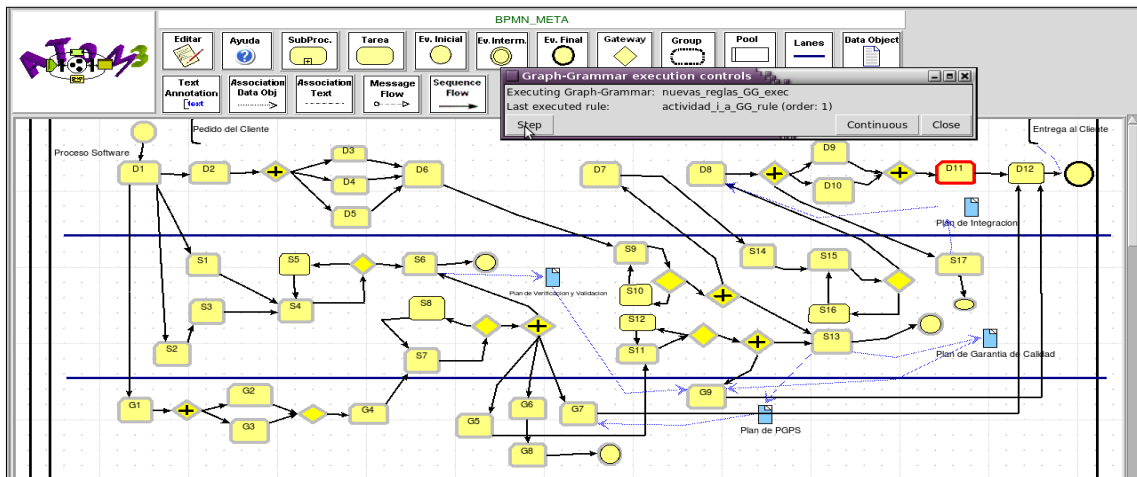


Figura E.58. Finalización de Gateway.

- ✓ **Sim-Step47:** Se activa el proceso D11 = Instalar el Sistema.



FiguraE.59. Inicio del Proceso D11.

- ✓ **Sim-Step48:** Finalizado el proceso D11 se activa el proceso D12 = Guardar Informes Finales.

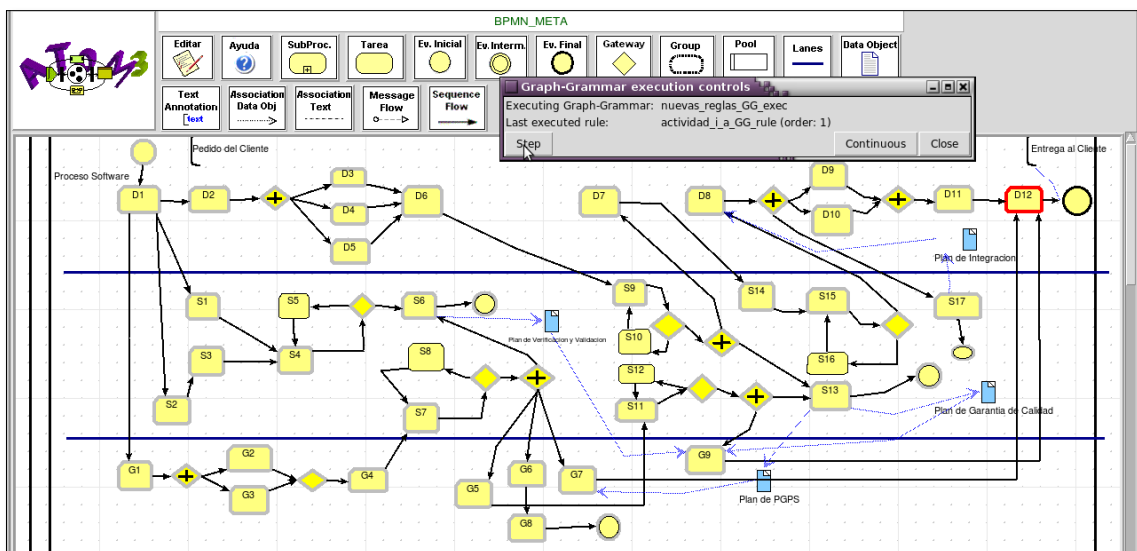


Figura E.60. Finalización del Proceso D11 e Inicio del Proceso D12.

- ✓ **Sim-Step49:** Finalizado el proceso D12 se activa el Evento Final

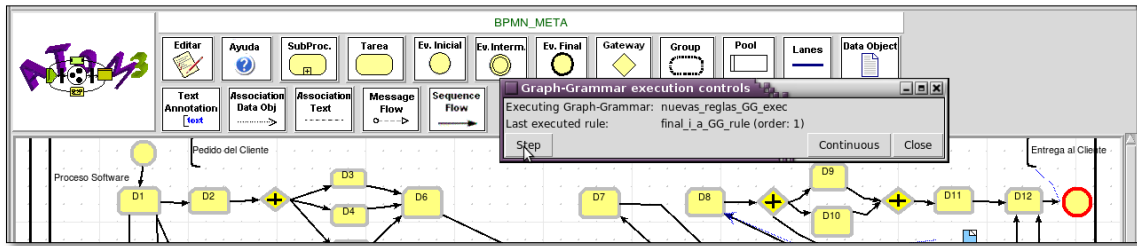


Figura E.61. Activación del Evento Final

✓ **Sim-Step50:** Finaliza el evento final.

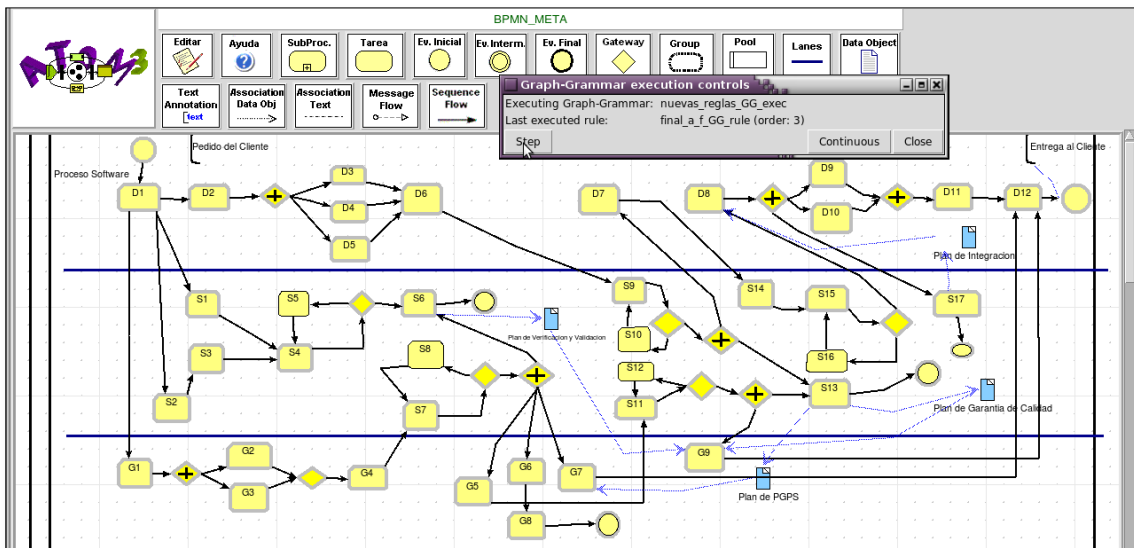


Figura E.62. Finalización del Evento Final

✓ **Sim-Fin-Step50:** Finaliza la ejecución de las reglas.

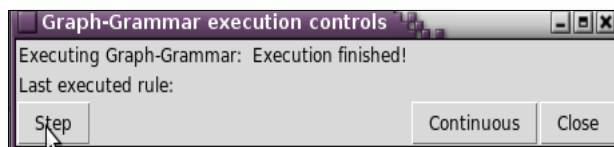


Figura E.63. Finalización de la Ejecución de las Reglas