



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



DEPARTAMENTO
DE SISTEMAS
INFORMÁTICOS
Y COMPUTACIÓN

Trabajo Fin de Master

Máster Universitario en Ingeniería y Tecnología
de Sistemas Software (**MUITSS**)
2016 – 2017

Modelado de procedimientos de respuesta a emergencias mediante CMMN

AUTOR: Marco Polo Ruiz Herrera

TUTOR: Juan Sánchez Díaz

A mi esposa Silvia por brindarme su amor y apoyo en cada etapa en cada día, a mis hijos Kamila y Diego por llenarme de alegría para renovar mis energías cuando más lo he necesitado.

A mi madre Yolanda, mi hermano Luis y mi abuela (mamá) Olga, por el apoyo y amor que me han brindado para realizar mis estudios.

Gracias amada familia por darme la fortaleza y ser el motivo que tengo para dar lo mejor.

Agradecimientos

A mi director de tesis Dr. Juan Sánchez Díaz, por su confianza depositada en mi para la realización del presente trabajo, el cual he podido culminar gracias a su ayuda y al seguimiento constante durante el tiempo transcurrido en que he desarrollado este trabajo fin de máster, gracias por ser más que un director, gracias por ser un amigo.

A todos los docentes del Máster Universitario en Ingeniería y Tecnología de Sistemas Software (MUITSS), tanto por los conocimientos impartidos como por la pedagogía utilizada en cada sesión académica.

A mis compañeros de clase quienes gracias a los retos y al buen trabajo en equipo hemos terminado siendo amigos.

Al grupo de investigación de Ingeniería del Software y Sistemas de Información (ISSI) perteneciente al Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia (UPV), quienes me permitieron colaborar en este proyecto de investigación.

A mis amigos, Dr. Joel Moreira Acosta, Dr. Nein Farrera Vázquez y Dr. Daniel González Scarpulli quienes con apoyo y ejemplo me han inspirado a estar en constante aprendizaje para crear tecnologías que brinden a la sociedad una mejor calidad de vida.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) del Gobierno de México y al Consejo de Ciencia y Tecnología del Estado de Chiapas (COCYTECH) por brindarme la oportunidad de mejoramiento profesional y académico por medio del programa de becas "CONACYT- GOBIERNO DEL ESTADO DE CHIAPAS" en la convocatoria del 2016.

Resumen

El objetivo del trabajo fin de master es llevar a cabo un estudio sobre la aplicación de la gestión flexible de casos sobre el dominio de la gestión de emergencias, con el fin de modelar los procedimientos de respuesta expresados en lenguaje natural de un plan de autoprotección y con ello obtener un modelo en lenguaje Modelo y Notación de Gestión de Casos (CMMN) que pueda ser consumido por la plataforma Camunda BPM para ser ejecutable con base al contexto y las situaciones cambiantes que se presentan durante una contingencia.

En primer lugar, se analizan los distintos lenguajes de modelado de negocios intensivos que pueden ser utilizados en la gestión flexible de casos, seguido de ello, se estudian los conceptos y elementos que conforman el lenguaje CMMN.

En segundo lugar, se realiza un caso de estudio en el cual se analiza y diseña un modelo de respuesta, partiendo de un procedimiento de respuesta de tipo incendio del plan de autoprotección del Hospital Universitario y Politécnico La Fe, para aplicar el lenguaje CMMN. Posteriormente, con ayuda de la plataforma Camunda BPM se ejecuta el modelo en distintos escenarios con la finalidad de obtener distintos flujos de ejecución adaptables a un contexto cambiante.

Por último, se presenta una propuesta de software que sirva de framework al trabajador del caso para facilitar el diseño de modelos y la ejecución de los mismos brindando interoperabilidad, a través de la implementación del servicio de API REST, con otros sistemas que podrían interactuar con el modelo durante la ejecución.

Palabras clave: Gestión de casos, gestión de emergencias, modelado de casos, CMMN, Camunda, procedimiento de respuesta, plan de autoprotección.

Abstract

The goal of my work to conduct a study on the application of adaptive case management on the domain of emergency management, in order to model the response procedures expressed in natural language of a self-protection plan and with this to obtain a model in the Language Management Model and Notation (CMMN) that can be used by Camunda BPM platform to be executable based on the environment and the changing situations that arise during a contingency.

Firstly, we analyze the different intensive business modeling languages, which can be used in adaptive case management, followed by the concepts and elements that set the CMMN language.

Secondly, a study case is carried out in which a response model is analyzed and designed, starting from a fire response procedure of the self-protection plan of the Hospital Universitario y Politécnico La Fe, to apply the CMMN language. Subsequently, with the help of the Camunda BPM platform, the model is executed in different scenarios in order to obtain different execution flows adaptable to a changing context.

Finally, the aim of this software is to work as a framework that facilitates the design and execution of models to case workers. We achieve that goal by providing interoperability, linking the implementation of the REST API service with other systems that could interact with the model during execution.

Key words: Case management, emergency management, case modelign, CMMN, Camunda, response procedure, self-protection plan.

Tabla de Contenidos

Agradecimientos.....	II
Resumen.....	III
Abstract.....	IV
Índice de Tablas.....	VII
Índice de Figuras.....	VIII
Abreviaturas.....	IX
1. Introducción.....	1
1.1 Motivación.....	2
1.2 Identificación del marco del Dominio de la investigación.....	2
1.3 CMMN.....	2
1.4 Estructura del informe.....	3
1.5 Sector de la población que puede beneficiarse.....	3
2. Marco de la investigación.....	4
2.1 Planteamiento del Problema.....	4
2.2 Delimitación del problema.....	5
2.3 Preguntas de investigación.....	5
2.4 Hipótesis del trabajo.....	5
2.5 Objetivos.....	6
2.5.1 General.....	6
2.5.2 Específicos.....	6
2.6 Metodología de la Investigación.....	6
3. Marco Teórico.....	7
3.1 Estado del arte: Gestión Flexible de Casos.....	7
3.2 Evaluación de lenguajes de modelado.....	8
3.2.1 BPM vs ACM.....	8
3.2.2 Lenguaje Modelo y Notación de Procesos de Negocio.....	12
3.2.3 Lenguaje Modelo y Notación de Decisión.....	12
3.2.4 Lenguaje Modelo y Notación para la Gestión de Casos.....	12
3.2.5 Diferencias entre lenguajes para el modelado.....	12
3.2.6 Ejemplos de aplicaciones industriales de ACM.....	13
3.2.7 Elección de lenguaje para modelar la Gestión de Emergencias.....	14
3.3 Lenguaje CMMN.....	15
3.3.1 Tiempo de diseño vs tiempo de ejecución.....	16
3.3.2 Elementos de CMMN.....	17
3.3.3 Ejecución Semántica de CMMN.....	26
4. Marco tecnológico.....	32
4.1 Acerca de la Plataforma de Camunda.....	32
4.2 Arquitectura.....	32
4.3 Etapa de Diseño: Camunda Modeler.....	33
4.4 Etapa de Implementación: Núcleo del Motor.....	34
4.5 Etapa de Ejecución: Lista de Tareas.....	35
4.6 Etapa de Análisis: Cockpit y Admin.....	37

5.	Caso de estudio del procedimiento de respuesta.....	39
5.1	Introducción	39
5.2	Planes de Autoprotección	39
5.3	Caso de estudio en el procedimiento de respuesta de un Incendio	40
5.3.1	Conocer el Procedimiento de Respuesta.	40
5.3.2	Diagrama del Procedimiento de Respuesta.	44
5.3.3	Identificación de los elementos críticos de negocio	45
5.3.4	Diseño del modelo en Lenguaje CMMN	45
5.3.5	Ejecución del modelo CMMN del Procedimiento de Respuesta	47
5.3.6	Observaciones	54
6.	Propuesta del Marco de trabajo	56
6.1	Modelador CMMN	56
6.1.1	Arquitectura de Modelador CMMN	56
6.1.2	Análisis y diseño conceptual del Modelador CMMN	57
6.1.3	Diseño de la presentación del Modelador CMMN.....	64
6.2	Ejecutor de Modelos.....	66
6.2.1	Arquitectura del Ejecutor de Modelos.....	66
6.2.2	Análisis y diseño conceptual del Ejecutor de Modelos.....	67
6.2.3	Diseño de la presentación del Ejecutor de Modelos	69
7.	Conclusiones y trabajos futuros	71
7.1	Conclusiones.....	71
7.2	Trabajos futuros	72
	Apéndice A - Definiciones	76
	Apéndice B – Forma CMMN.....	78
	Apéndice C – Guía para iniciar Camunda con CMMN 1.1	98

Índice de Tablas

Tabla 3-1 Características entre BPM y ACM.....	11
Tabla 3-2 Ejemplo de aplicaciones industriales de ACM.....	13
Tabla 3-3 Áreas de segmento para casos ACM.....	14
Tabla 3-4 Reglas de propiedad de comportamiento.....	25
Tabla 5-1 Propiedades de los elementos del modelo Incendio.....	47
Tabla 5-2 Valores de las variables del flujo de trabajo Falsa Alarma.....	50
Tabla 5-3 Valores de las variables del flujo de trabajo Conato de Incendio.....	51
Tabla 5-4 Valores de las variables del flujo de trabajo Emergencia Parcial.....	52
Tabla 5-5 Valores de las variables del flujo de trabajo Emergencia General.....	53
Tabla 6-1 Usuario Administrador del Modelador CMMN.....	58
Tabla 6-2 Usuario Diseñador del Modelador CMMN.....	58
Tabla 6-3 Caso de uso de crear usuario.....	59
Tabla 6-4 Caso de uso de modificar usuario.....	60
Tabla 6-5 Caso de uso de eliminar usuario.....	60
Tabla 6-6 Caso de uso de listar usuarios.....	60
Tabla 6-7 Caso de uso de buscar usuario.....	61
Tabla 6-8 Caso de uso de crear modelo.....	62
Tabla 6-9 Caso de uso de editar modelo.....	62
Tabla 6-10 Caso de uso de editar modelo.....	62
Tabla 6-11 Caso de uso de listar modelos.....	63
Tabla 6-12 Caso de uso de buscar modelo.....	63
Tabla 6-13 Caso de uso de exportar modelo.....	63
Tabla 6-14 Caso de uso de importar modelo.....	64
Tabla 6-15 Caso de uso de leer documento.....	64
Tabla 6-16 Usuario Administrador del Ejecutor de modelos.....	68
Tabla 6-17 Usuario Experto del Ejecutor de modelos.....	68
Tabla B-1 Atributos del caso.....	79
Tabla B-2 Atributos del Rol.....	80
Tabla B-3 Atributos de PlanItemControl y asociaciones de modelos.....	81
Tabla B-4 Atributos de CaseFile y asociaciones de modelos.....	83
Tabla B-5 Atributos de PlanItemDefinition.....	84
Tabla B-6 Atributos de Tarea.....	84
Tabla B-7 Atributos de EventListener.....	86
Tabla B-8 Atributos de Sentry.....	88
Tabla B-9 Atributos de PlanFragment.....	89
Tabla B-10 Atributos de Stage.....	91
Tabla B-11 Atributos de PlanningTable.....	93
Tabla B-12 Definición de elementos CMMN.....	93

Índice de Figuras

Figura 3-1 Organización de un sistema BPM y un ACM.....	11
Figura 3-2 Diferencias en términos de conceptos básicos, enfoque y semántica.....	13
Figura 3-3 Tres categorías en la Gestión Flexible de Casos.....	14
Figura 3-4 Tiempo de diseño vs tiempo de planeación y ejecución.....	17
Figura 3-5 Tabla resumen de aplicabilidad del decorador.....	26
Figura 3-6 Conceptos del ciclo de vida de CMMN.....	27
Figura 3-7 Ejemplo para explicar el concepto del ciclo de vida de CMMN.....	28
Figura 4-1 Arquitectura de Camunda BPM.....	33
Figura 4-2 Interfaz de Camunda Modeler.....	34
Figura 4-3 Notación CMMN 1.1 soportada.....	35
Figura 4-4 Interfaz de Camunda TaskList.....	36
Figura 4-5 Interfaz de Camunda Cockpit.....	37
Figura 4-6 Interfaz de Camunda Admin.....	38
Figura 5-1 Diagrama de Actuación del Procedimiento de Respuesta.....	44
Figura 5-2 Modelo CMMN del Procedimiento de Respuesta de Incendio.....	46
Figura 5-3 Tarea Procesar inicio Incidente.....	50
Figura 5-4 Flujo de ejecución de Falsa Alarma.....	50
Figura 5-5 Flujo de ejecución de Conato de Emergencia.....	51
Figura 5-6 Flujo de ejecución de Emergencia Parcial.....	52
Figura 5-7 Flujo de ejecución de Emergencia General.....	53
Figura 6-1 Arquitectura de Modelador CMMN.....	56
Figura 6-2 Diagrama del contexto del Modelador CMMN.....	58
Figura 6-3 Diagrama estructurado de casos de uso.....	58
Figura 6-4 Diagrama de casos de uso de Gestión de usuarios.....	59
Figura 6-5 Diagrama de casos de uso de Gestión de modelos.....	61
Figura 6-6 Prototipo de Interfaz de ingreso.....	65
Figura 6-7 Prototipo de Interfaz de área de trabajo.....	65
Figura 6-8 Arquitectura del Ejecutor de Modelos.....	66
Figura 6-9 Diagrama del contexto del Ejecutor de modelos.....	68
Figura 6-10 Diagrama estructurado de casos de uso.....	69
Figura 6-11 Prototipo de Interfaz de ingreso.....	70
Figura 6-12 Prototipo de Interfaz de área de trabajo.....	70
Figura B-1 Definición del diagrama de clases.....	78
Figura B-2 Diagrama de la clase del caso.....	79
Figura B-3 PlanItemControl diagrama de clases.....	81
Figura B-4 CaseFile diagrama de clases.....	82
Figura B-5 PlanItemDefinition diagrama de clases.....	83
Figura B-6 Diagrama de clases de Tarea.....	84
Figura B-7 Diagrama de clases de EventListener.....	86
Figura B-8 Diagrama de clases de Sentry.....	87
Figura B-9 Diagrama de clases de PlanFragment.....	89
Figura B-10 Diagrama de clases de Stage.....	90
Figura B-11 Diagrama de clases de PlanningTable.....	92
Figura C-1 Asistente para proyectos nuevos Maven.....	99
Figura C-2 Nuevo archivo CMMN del Modelador de Camunda.....	101
Figura C-3 Definir el Id del Caso.....	102
Figura C-4 Añadir una tarea al caso.....	103
Figura C-5 Activación manual.....	104
Figura C-6 Activación automática.....	104

Figura C-7 Asignación de usuario.....	105
Figura C-8 Camunda Tasklist.....	107
Figura C-9 Detalle de una tarea.....	108
Figura C-10 Añadir un hito al caso.....	109
Figura C-11 Añadir un centinela al caso.....	110
Figura C-12 Añadir una condición a un centinela.....	110
Figura C-13 Añadir un Oyente del Hito.....	111
Figura C-14 Agregar variable a la Tarea Check Application.....	112
Figura C-15 Agregar variable a la Tarea Provide Customer Rating.....	112
Figura C-16 Agregar una condición a un sentry de salida.....	113
Figura C-17 Agregar variable con valor False a la Tarea Check Application.....	114

Abreviaturas

ACM	Adaptive Case Management
BAM	Business Activity Monitoring
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BRM	Business Rules Management
BRMS	Business Rules Management System
CMMN	Case Management Model and Notation
DMN	Decision Model and Notation
ECM	Enterprise Content Management
IDE	Integrated Development Environment
ISV	Independent Software Vendor
OMG	Object Management Group
UML	Unified Modeling Language
MVC	Model – View – Controller
XML	Extensible Markup Language

1. Introducción

Este trabajo fin de máster se ha desarrollado en colaboración con el grupo de investigación de Ingeniería del Software y Sistemas de Información (ISSI) perteneciente al Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia (UPV).

Tanto los estudios del master como este trabajo de investigación han sido financiados con la beca “CONACYT- GOBIERNO DEL ESTADO DE CHIAPAS” cuyos recursos son aportados por el Consejo Nacional de Ciencia y Tecnología (CONACYT) del Gobierno de México y el Consejo de Ciencia y Tecnología del Estado de Chiapas (COCYTECH).

Las soluciones basadas en la Gestión de Procesos de Negocios (Business Process Management, BPM) han aportado importantes avances a las organizaciones de trabajo y a la automatización. Sin embargo, a pesar de la sólida base de BPM en la definición formal del flujo de trabajo, en algunas ocasiones las soluciones de BPM no son adecuadas para escenarios de trabajo en los que un flujo de trabajo preciso no es la opción requerida. En esos escenarios, el trabajo depende en gran medida de las decisiones basadas en el conocimiento sobre las actividades y los resultados, dando lugar a múltiples vías de trabajo y reglas de negocio que pueden llegar a ser complejas o incluso inviables para modelar y automatizar completamente. En estos casos, se necesita un enfoque dinámico soportado tecnológicamente. El enfoque no es aislar y automatizar las decisiones y las reglas, sino más bien ofrecer un soporte de información oportuno al trabajador del caso para alcanzarlas. La Gestión Flexible de Casos (Adapted Case Management, ACM) se perfila como un patrón de diseño exitoso para esto.

Una metodología de desarrollo de software se refiere al marco y las disciplinas que se utilizan para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Este enfoque proporciona criterios de separación de interés bien entendidos, con el objetivo de facilitar a los analistas de negocio y arquitectos de software la comprensión e incorporación de las prácticas de diseño de ACM en sus actuales habilidades profesionales.

El Grupo de Gestión de Objetos (Object Management Group, OMG) introduce recientemente el lenguaje Modelo y Notación de Gestión de Casos (Case Management Model and Notation, CMMN) 1.1, el cual tiene un impacto similar a ACM como lo es el Modelado y Notación de Procesos de Negocios (Business Process Modeling Notation, BPMN) para BPM.

Este trabajo de investigación toma las Metodologías actuales de ACM un paso más allá, busca dejar las bases de diseño de una herramienta que permita al usuario modelar con base a la notación del lenguaje CMMN, así mismo, utilizando como herramientas el modelador y el motor de procesos de la plataforma Camunda, se busca demostrar que los problemas clasificados como modelo de negocios intensivos e inciertos como lo es el dominio de la gestión de emergencias, puedan ser abordados por la gestión flexible de casos, de tal modo que un procedimiento de respuesta de un plan de autoprotección presentado en lenguaje natural pueda ser modelado y posteriormente ejecutado, y que dependiendo del entorno y de las decisiones que elija el trabajador del caso, el comportamiento del modelo sea adaptable.

Para ello se partirá desde la etapa de análisis, después el diseño del modelo y posteriormente la ejecución como parte de una metodología ACM basada en la práctica.

1.1 Motivación

Los planes de autoprotección son un conjunto de acciones y medidas encaminadas a prevenir y controlar los riesgos sobre las personas y los bienes, a dar una respuesta adecuada a las posibles situaciones de emergencias a las que está expuesta una organización.

Los procedimientos de respuesta, dentro de los planes de autoprotección, son acciones que se realizan de manera inmediata, ante la ocurrencia de un desastre, con el objetivo de prevenir la pérdida de vidas, salvaguardar los bienes y normalizar la situación tan pronto como sea posible. El procedimiento de respuesta contiene la identificación de los actores participantes y la lista de acciones o tareas que habrán de ejecutarse; normalmente las tareas y la secuenciación de las mismas están relatadas en lenguaje natural; sin embargo, por su naturaleza y las variantes que pueden ocurrir es preciso que el plan de autoprotección sea ejecutable y flexible [1].

Los problemas de naturaleza clasificados como modelos de negocios intensivos en conocimiento deben ser atendidos con una notación basada en la gestión flexible de casos. Uno de los campos de aplicación con mayor interés en la actualidad es la gestión de emergencias, debido al incremento en los desastres naturales que se han venido registrando en la última década.

Con la gestión flexible de casos se utilizará el lenguaje de modelado de casos CMMN [2], el cual habrá de permitir representar procesos de respuesta flexibles, basados en los procedimientos de respuesta, para dar atención oportuna a una situación de emergencia y con ello poder ejecutar en tiempo real los procedimientos de respuesta, los cuales podrán adaptarse a eventualidades cambiantes, es por ello por lo que la gestión flexible de casos evoluciona al concepto de gestión flexible de casos.

1.2 Identificación del marco del Dominio de la investigación

El presente proyecto es situado en las dimensiones de información y presentación de la fase de planificación del dominio de Gestión de Emergencias [3], y sustenta la hipótesis de que la gestión flexible de casos con el lenguaje CMMN puede ser aplicada con éxito al modelado de procedimientos de respuesta para que puedan ser ejecutables y adaptados al contexto en tiempo de ejecución.

1.3 CMMN

En mayo del 2014 la OMG formalizó el lenguaje de modelado de casos CMMN, el cual permite representar procedimientos de respuesta flexibles ante una situación de emergencia.

CMMN fue diseñado como un enfoque declarativo para el diseño, ejecución de la planificación y ejecución de casos de negocio.

El lenguaje CMMN define un modelo y una notación común para expresar gráficamente un caso y sus elementos de negocios con respecto a la gestión flexible de casos.

1.4 Estructura del informe

Capítulo 1: “Introducción” proporciona la información sobre la tesis, para ello explica la motivación para la realización de esta investigación, el marco de dominio, la estructura del informe y el sector de la población que puede beneficiarse.

Capítulo 2: “Marco de la investigación” contiene la información sobre el planteamiento y delimitación del problema que conducirán a la formulación de las preguntas de la investigación, la hipótesis de trabajo, y el objetivo general y los objetivos específicos que se pretenden alcanzar como resultado de la investigación.

Capítulo 3: “Marco teórico” como resultado de un extenso estudio bibliográfico, brinda el contenido del contexto teórico sobre el modelado de gestión flexible de casos.

Capítulo 4: “Marco tecnológico” se presenta la plataforma Camunda BPM [26] la cual se utiliza como base para diseñar y ejecutar el modelo del caso de estudio de la emergencia de tipo incendio de un hospital.

Capítulo 5: “Experimento en procedimientos de respuesta” explica el enfoque de desarrollo, los resultados del diseño y las observaciones como producto de aplicar la gestión flexible de casos con el lenguaje CMMN al modelado de procedimientos de respuesta.

Capítulo 6: “Propuesta del Marco de trabajo” se describe el diseño de las herramientas que serán el marco de trabajo en el cual el trabajador del caso pueda diseñar modelos y brindar una interfaz para la ejecución de los modelos en lenguaje CMMN y que estos a su vez interactúen con otros entornos.

Capítulo 7: “Conclusiones y recomendaciones” contiene las conclusiones, recomendaciones y trabajos futuros como resultado de aplicar la gestión flexible de casos con el lenguaje CMMN a la gestión de emergencias.

1.5 Sector de la población que puede beneficiarse

La información de esta investigación puede ser útil para la siguiente audiencia:

- Los responsables de la seguridad de las organizaciones que estén interesados en ejecutar en tiempo real sus procedimientos de respuesta y evaluar para una mejora continua sus procedimientos.
- Las autoridades encargadas de brindar a la población acciones en materia de protección civil.
- Profesores y alumnos que con fines académicos trabajan en el campo del modelado de procesos de negocios declarativos y la gestión flexible de casos.
- Profesionales que participan en el diseño, desarrollo e implementación de software con aplicación en la gestión flexible de casos.
- Organizaciones que están considerando el uso de software con aplicación de gestión flexible de casos para la gestión de sus procesos y mejora de los mismos.

2. Marco de la investigación

2.1 Planteamiento del Problema

Es importante señalar que, si bien el presente trabajo para su desarrollo fue basado en la normatividad de protección civil en España y que el caso práctico presentado es con referencia a una incidencia de Incendio en el Hospital Universitario y Politécnico La Fe ubicado en la ciudad de Valencia, se tiene como objetivo que la metodología de trabajo, el diseño del marco de trabajo propuesto y los resultados obtenidos en el experimento del caso de estudio sirvan de guía para que puedan ser aplicados de manera general al dominio de la gestión de emergencias.

De acuerdo con la norma en protección civil las organizaciones en España deben contar con un plan de autoprotección [4] el cual, en la mayoría de los casos con el paso del tiempo, al estar en una presentación de documento impreso o en archivo en formato de texto, se encuentra desactualizado cuando en la organización se han realizado modificaciones en la estructura organizacional, en los procesos de negocio o la infraestructura del edificio.

Así mismo, generalmente los que integran el equipo de “responsables de seguridad” sus actividades principales son ajenas a la seguridad y están orientados en la colaboración de los procesos de negocio de la organización, si bien reciben capacitación para actuar ante una situación de emergencia con base al plan de autoprotección, la presentación robusta del plan de autoprotección dificulta encontrar el procedimiento de respuesta a realizar en el momento de una determina situación de emergencia.

Los lenguajes basados en procesos han sido destinados durante la última década para modelar procedimientos de respuesta a acontecimientos de emergencia. La validez práctica y operativa de estos modelos depende de la capacidad de los mismos para operar contingencias y contextos excepcionales durante la respuesta a emergencias. Estos modelos suministran una flexibilidad limitada, debido a que suelen tener únicamente en cuenta un pequeño número de variaciones y excepciones, donde los usuarios en el proceso tienen poca autonomía para decidir qué debería realizarse en cada acontecimiento.

En este trabajo se pretende que los procedimientos de autoprotección puedan ser modelados y ser ejecutables, para ello se utilizará la Gestión Flexible de Casos (ACM) la cual es una aproximación de reciente creación que se utiliza para gestionar procesos intensivos de conocimiento, el modelado de casos será con CMMN para representar los procedimientos de respuesta ante una situación de emergencia. Las razones por las que se eligió CMMN para el modelado son la planificación en tiempo de ejecución para la selección de tareas para un caso, poder ordenar en tiempo de ejecución la secuencia en que las tareas serán realizadas, colaboración ad-hoc con otros trabajadores del caso sobre tareas [5].

Así mismo se propone el diseño de un marco de trabajo el cual será una herramienta basada en la plataforma Camunda BPM [6] con interfaz web que permita editarlos modelos de los procedimientos de respuesta con la notación CMMN, almacenar y recuperar los modelos, exportarlos en formato Extensible Markup Language (XML) y posteriormente ejecutarlos a través del motor de Camunda [7].

2.2 Delimitación del problema

Los sistemas basados en la gestión flexible de casos permiten a los participantes en un proceso gestionar situaciones donde el nivel de flexibilidad es más avanzado que el proporcionado por los procesos clásicos, con lo anterior se pretende un salto desde los modelos de procesos estructurados.

Con el supuesto de que esta es la forma para gestionar procesos intensivos de conocimiento, el presente proyecto propondrá el diseño de una aplicación web que permita al trabajador del caso editar modelos de respuesta flexibles basados en los planes de autoprotección descritos en lenguaje natural con presentación en texto y dar atención a la validación de los procesos de autoprotección mediante la simulación de distintos contextos, con base a lo anterior se podrá contar con un objeto digital que podrá ser ejecutable y que pueda ser consumido por el motor del sistema Camunda.

2.3 Preguntas de investigación

Existen una serie de aspectos que surgen como interrogantes al momento de evaluar esta propuesta de solución.

RQ1: ¿Es el lenguaje CMMN la mejor opción para modelar la gestión flexible de casos aplicado a la gestión de emergencias con respecto a otras notaciones de modelado de negocios?

RQ2: ¿Cuáles son los conceptos teóricos del Lenguaje CMMN?

RQ3: ¿Cómo puede aplicarse el lenguaje CMMN para analizar y diseñar modelos de respuesta partiendo de procedimientos de respuesta de los planes de autoprotección?

RQ4: ¿Cómo se puede ejecutar modelos basados en lenguaje CMMN que sean adaptables al entorno y a las decisiones elegidas por el trabajador del caso?

RQ5: ¿Cuál sería el marco de trabajo a utilizar cuya interfaz facilite al trabajador del caso el diseño y ejecución de procedimientos de respuesta en modelos basados en lenguaje CMMN?

RQ6: ¿Qué conclusiones y recomendaciones pueden realizarse sobre la aplicación del Lenguaje CMMN en el dominio de casos de emergencia?

2.4 Hipótesis del trabajo

La gestión flexible de casos puede ser aplicada con éxito en los procedimientos de respuesta teniendo en cuenta el contexto de ejecución y las situaciones cambiantes que se producen durante la mitigación de modo que sea posible la incorporación de nuevas tareas de acuerdo con el criterio de los participantes.

La hipótesis que se sustenta es que se puede atender con la gestión flexible de casos la gestión de emergencias, utilizando el modelado de casos para representar respuestas flexibles con base a los procedimientos de respuesta.

Además, con tales tecnologías orientadas a la gestión flexible de casos y modelos de respuesta flexibles de los procedimientos de respuesta de los planes de autoprotección debe ser posible desarrollar una herramienta informática que integre y sirva de marco de trabajo para que los usuarios encargados de la seguridad de las organizaciones puedan acceder en tiempo de ejecución a toda la información que les permita dar atención oportuna ante una emergencia.

2.5 Objetivos

2.5.1 General

Estudiar la adecuación del lenguaje CMMN para modelar y ejecutar procedimientos de respuesta en el dominio de las emergencias.

2.5.2 Específicos

SO1: Fundamentar la elección del lenguaje CMMN para modelar la gestión flexible de casos aplicada a la gestión de emergencias con respecto a otras notaciones con base al estudio y análisis del estado del arte.

SO2: Estudiar y analizar a profundidad los conceptos teóricos del Lenguaje estándar de CMMN.

SO3: Aplicar el lenguaje CMMN para analizar y diseñar modelos de respuesta flexibles partiendo de procedimientos de respuesta de los planes de autoprotección.

SO4: Ejecutar un modelo CMMN y que sea adaptable al entorno y a las decisiones elegidas por el trabajador del caso.

SO5: Proponer el diseño de aplicaciones web basadas en la plataforma Camunda BPM que sirva de Framework para el trabajador del caso con la finalidad de facilitar el diseño y ejecución de un modelo CMMN, así como la interacción con otros entornos.

SO6: Presentar conclusiones y recomendaciones que aporten futuras líneas de investigación sobre la aplicación del lenguaje CMMN en el dominio de casos de emergencia.

2.6 Metodología de la Investigación

La naturaleza fundamental de la investigación que se llevará a cabo es de carácter desarrollo tecnológico, la metodología de la investigación establecido por Sampieri [8], será cualitativa.

Así mismo, considerando el conjunto de elementos básicos de evaluación de la investigación para la Ingeniería del Software establecidos por Shaw [9], encontramos que la combinación de los elementos de la investigación que permitirá cumplir con los objetivos propuestos será el siguiente:

El tipo de pregunta corresponde al **método de desarrollo** porque se desea definir la forma de construir un tipo de producto particular o desarrollar alguna actividad específica. Así mismo la clasificación del tipo de resultado obtenido por el proyecto será **solución específica o prototipo** porque se pretende diseñar y/o desarrolla una aplicación problema que permite observar algún principio de ingeniería. Por último, el mecanismo de validación será el **ejemplo** porque se muestra una sola aplicación que sirve de verificación.

3. Marco Teórico

3.1 Estado del arte: Gestión Flexible de Casos

De una manera general se entienden los sistemas basados en modelos de negocios intensivos en conocimiento como aquellos que permiten que los usuarios dispongan de la mayor cantidad de información posible de manera ordenada, con la finalidad de facilitar la toma de decisiones y se elijan las tareas a realizar en cada situación para alcanzar una meta de negocio.

Los diferentes trabajos que se han desarrollado y publicado con el enfoque de modelos de negocios intensivos en conocimiento se listan a continuación.

Fue a partir de la década de los 80's cuando, a raíz del modelo japonés (Sistema de producción Toyota) y de la aparición de normas internacionales de calidad principalmente, cuando se impulsó la implantación de un sistema estructural basado en la gestión por procesos.

Para el 2005, Object Management Group presenta el Modelo y Notación de Procesos de Negocio (BPMN) una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow).

En 2008, Oracle combina la incorporación de la gestión de procesos de negocios (BPM) y la Arquitectura Orientada a Servicios.

En marzo del mismo año, con sede en Berlín, Alemania, fundan la organización Camunda, la cual se ha centrado en el tema de la gestión de procesos de negocio (BPM) desde el primer día y, entre otras cosas, participó en la definición del estándar BPMN 2.0.

En mayo del 2014 la Object Management Group formalizó el lenguaje de modelado de casos CMMN (Case Management Modeling Notation), el cual es la extensión de BPMN para atender requerimientos en la gestión flexible de casos.

A partir de noviembre del 2015, el software Camunda en su versión 7.4.0 combina el uso de BPMN 2.0, CMMN 1.0 y DMN 1.1

En noviembre del 2005, el equipo de Google drive desarrolla la librería mxGraph [10] la cual es la base del sitio Draw.io el cual es un editor libre con entorno web en donde se pueden dibujar diagramas de flujo, de procesos, de organización, de casos, de red, de circuitos electrónicos; sin embargo, no utiliza la notación CMMN, a partir del 2016 la librería mxGraph está disponible como código libre para su uso y colaboración, por ello esta librería fue la base para desarrollar el modelador de Camunda en donde se diseñan los modelos en lenguaje CMMN.

Aunado a lo anterior, investigadores y miembros del Grupo de Investigación de Ingeniería del Software y Sistemas de Información (ISSI) del Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia; han realizado trabajos de investigación en los cuales utilizan las herramientas orientadas a la gestión flexible de casos para atender incidentes en la gestión de emergencias.

3.2 Evaluación de lenguajes de modelado

Con la finalidad de definir como se modelarán los requerimientos de los sistemas se introdujeron las notaciones gráficas, al inicio los Diagramas de Flujo, posteriormente se evolucionó al Leguaje Unificado de Modelado (Unified Modeling Language, UML), y a continuación a la Notación para el Modelado de Procesos de Negocios (BPMN); que permiten modelar una serie de normas y estándares gráficos respecto a cómo se deben representar los esquemas relativos al software, con el objeto de ofrecer soluciones para atender los requerimientos de los procesos de negocio, donde la ejecución de una actividad viene determinada por la ejecución de actividades previas.

Por otro lado, existen modelos de negocios intensivos en conocimiento donde la ejecución de una actividad no viene siempre determinada por la ejecución de alguna actividad previa, estos modelos de negocios están centrados en datos, adaptables y flexibles; conocido en el contexto organizacional como “Gestión de Casos”, presentado por Van der Alts y Weske [11][12], también es encontrado en la literatura como Gestión de Flexible de Casos (ACM).

ACM permite que los usuarios, basándose en el contexto de ejecución y en su experiencia o conocimiento en la resolución de casos previos, elijan que tareas pueden realizarse en cada situación para alcanzar un meta de negocio. Uno de los campos de aplicación con mayor interés en la actualidad es la gestión de emergencias, debido al incremento en los desastres naturales que se han venido registrando.

El concepto de Gestión Flexible de Casos no es nuevo. Ha sido aplicado durante años a algunos tipos de casos específicos, sobre todo clínicos y legales. Poco a poco se ha ido extendiendo al resto de sectores y asuntos.

Al hablar de Gestión Flexible de Casos, nos estamos refiriendo de forma general a la tramitación de asuntos puntuales de cualquier índole a resolver en una organización y en un determinado período de tiempo. En este contexto, los Casos provienen de solicitudes (internas o externas) en su sentido más amplio: incidencias, expedientes, petición de bienes o servicios, atención al cliente, compras, acciones de marketing, proyectos, etc. y solo se terminan cuando el caso ha sido, de alguna manera, completamente resuelto.

La gestión de estas solicitudes es de vital importancia para cualquier entidad, ya que suponen una parte importante de la actividad cotidiana de la mayoría de las empresas privadas y entidades públicas.

Además, puede representar un gran reto, ya sea por la complejidad de su gestión, por la carencia de las herramientas adecuadas para su resolución o por el coste de la implantación de éstas.

Las características y diferencias entre el modelado de procesos de negocio y el modelado de gestión flexible de casos se especifican en el siguiente tema.

3.2.1 BPM vs ACM.

La gestión flexible de casos no puede considerarse como una especificación independiente, sino como un complemento de los conceptos existentes como la Gestión de Proceso de Negocios (Business Process Management, BPM) y la Administración de Contenido Empresarial (Enterprise Content Management, ECM). En los últimos años BPM ha crecido realmente como una herramienta mundial para el diseño, promulgación, gestión y análisis de procesos de negocio predeterminados [11].

Se ha pronosticado un crecimiento exponencial del mercado de BPM de \$6,96 mil millones de dólares desde el 2016 a \$13,52 mil millones de dólares en el 2021 [13]. Gartner en el 2013 afirmó que para 2016, el 60% de las iniciativas de BPM fallarían a menos que puedan demostrar un impacto positivo directo en la estrategia organizacional. Dado que, en el entorno empresarial actual, el conocimiento y la incertidumbre se vuelven cada vez más importantes, por lo tanto, el enfoque en la gestión flexible de casos crecerá [14].

Uno de los principales factores que influyen en la adopción de un estándar por parte de los usuarios finales a los que el CEO de Camunda Jakob Freund, principal promotor de CMMN como lenguaje enfocado a resolver situaciones en el dominio de la gestión flexible de casos, se refiere son las ventas de los proveedores de productos BPMN que no terminan por aceptar las limitantes de esa notación y que señalan que BPMN puede utilizarse para modelar la gestión flexible de casos, sin embargo, el pronostica que al final CMMN tendrá éxito en su utilización tal y como pasó con la introducción de BPMN 2.0 hace 9 años cuando el mercado, impulsado por IBM, presionó para que se facilitará la comercialización y adopción del nuevo estándar por parte de los usuarios finales con base a las ventajas que proveía con respecto a su antecesor versión [15].

Como ya se mencionó anteriormente, las empresas operan en entornos volátiles y, por lo tanto, no sólo pueden funcionar a través de flujos de trabajo secuenciales. En "Gestión Flexible de casos: revisión de enfoques de modelado", se destacan algunos elementos para ilustrar por qué ciertos procesos (procesos de gestión de casos) no pueden ser modelados de la misma manera que se ha hecho con flujos de trabajo secuenciales BPMN. Estos elementos representan básicamente las diferencias básicas entre el modelado de procesos de gestión flexible de casos y el modelado de procesos de negocios [16]:

- Un expediente de caso es el artefacto central, crea el contexto a través de todas las actividades en el caso. BPMN carece de un artefacto similar.
- La gestión flexible de casos se basa en eventos y datos. Ser impulsado por eventos significa que los eventos pueden desencadenar procesos, actividades y tareas en cualquier secuencia. En la gestión flexible de casos, los datos, que se recogen y almacenan en el archivo del caso, ayudan a los trabajadores de casos en sus habilidades de toma de decisiones y las nuevas etapas, tareas y actividades se activarán basándose en la evaluación de los trabajadores de casos de esos datos. BPMN 2.0 permite elementos basados en eventos (pasarelas, subprocesos) con la diferencia de que un proceso todavía se ejecuta en una secuencia predefinida y en general. BPMN 2.0 también permite gateways exclusivos basados en datos para insertar la posibilidad de comprobar los datos basados en una condición expresada [17]. Cuando los trabajadores de casos toman decisiones basadas en su conocimiento, habilidad e intuición, los gateways exclusivos basados en datos automatizarán esta toma de decisiones basada en condiciones predefinidas.
- La gestión flexible de casos también es muy colaborativa (porque el caso es colaborativo). De alguna manera, el BPMN también es colaborativo porque diferentes elementos pueden ser parte del mismo proceso, pero esa colaboración puede ser vista como mecánica, ya que todo forma parte de una secuencia predefinida mientras que la colaboración en la gestión flexible de casos es ad hoc. Dependiente de la situación y, por tanto, más impredecible y flexible.

- La gestión flexible de casos se basa en la importancia de los trabajadores humanos, también llamados trabajadores del caso. No se supone que sólo ejecutan ciertas tareas en una secuencia como están en BPMN. Ellos se benefician de la libertad y la experiencia para ser creativos en la forma en que manejan ciertas situaciones no se limitan a lo que se debe hacer, sino que deben guiarse por lo que se puede hacer [16]. Como consecuencia, estos trabajadores del caso pueden tener una gran influencia en un proceso.

Joern Franke en el 2011 agrega diferencias adicionales entre BPM y gestión flexible de casos [18]:

- En BPM, los activos de la empresa sólo desempeñaron un papel menor en lo que respecta a la ejecución y modelado de procesos. En caso de gestión, los activos de negocio están recibiendo mucha más atención. Cohn y Hull hablan de los datos manipulados por procesos centrados en los flujos de actividad (como BPMN). En estos procesos, los datos, como los activos de la empresa, se ven como un segundo ciudadano durante todo el proceso y no se toman en cuenta [19]. Ejemplo de la importancia de los activos empresariales en los casos: Un caso de seguro regula todos los contratos de seguros, incidentes, procesos, reglas, regulaciones, datos, clientes y personas relacionadas con él. Por lo tanto, un caso de seguro contiene importantes activos comerciales de cualquier compañía de seguros.
- La gestión flexible de casos se caracteriza por una fuerte integración con la Gestión de reglas de negocio (Business Rule Management, BRM). Un sistema BRM (Business Rule Management System, BRMS) es definido por IBM como "Un sistema que permite definir, implementar, supervisar y mantener las políticas organizacionales y las decisiones operativas asociadas con esas políticas" [20]. Según Schlosser uno de los principales impulsores de BRM es la necesidad de agilidad organizacional para permitir una toma de decisiones rápida y ad hoc, de ahí la importancia en la gestión flexible de casos.
- Los procesos de negocio modelados en BPM y ACM son fundamentalmente diferentes. BPM intenta automatizar una secuencia estable y predefinida de actividades y tareas, la ACM se centra en procesos ad hoc impredecibles. Aunque son muy diferentes, ambos tratan de lograr lo mismo: aumentar la eficiencia del flujo del proceso. Debido a que intentan alcanzar el mismo objetivo, y ambos sistemas se centran en un aspecto diferente del espectro del proceso, las empresas están experimentando para combinarlos en un sistema capaz de administrar procesos estructurados y no estructurados.

Desde la perspectiva técnica, los pasos de un motor BPM deben realizarse en un orden persistente. Una vez que se ha iniciado una instancia de proceso, no se pueden agregar nuevos pasos. Sin embargo, un trabajador del conocimiento puede agregar o cambiar un paso del motor ACM en tiempo de ejecución.

En la Figura 3-1 se muestra la organización de un sistema BPM y un sistema ACM.

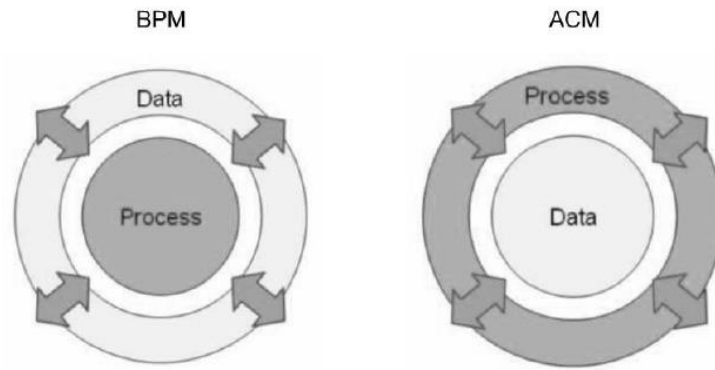


Figura 3-1 Organización de un BPM y un ACM [21].

La principal fortaleza de la Gestión de Procesos de Negocio (BPM) es interpretar de manera rígida el proceso de estructuras para lograr un objetivo. En comparación con ello, la fortaleza de la Gestión Flexible de Casos (ACM) es la combinación de la experiencia del trabajador del conocimiento con información adicional basada en Tecnologías de la Información, para permitir que el trabajador del conocimiento tome mejores decisiones.

En la siguiente tabla se resumen las principales diferencias entre los modelados de Proceso de Negocios y Gestión Flexible de Casos:

Modelado de Proceso de Negocios	Modelado de Gestión Flexible de Casos
Puede definir una secuencia ordenada de actividades que se pueden completar para resolver un reto empresarial.	Puede definir un conjunto desordenado de actividades que se pueden completar para resolver un reto comercial.
La secuencia de actividades es estable y rara vez cambia; es decir, el proceso es predecible y repetible.	Las actividades ocurren en un orden impredecible.
El proceso determina los eventos. La primera actividad determina el primer conjunto de eventos, que luego conduce a la siguiente actividad y al siguiente conjunto de eventos. Las actividades están conectadas entre sí, lo que determina la secuencia.	Los sucesos determinan el proceso. A medida que los eventos ocurren, un trabajador selecciona la actividad apropiada. El proceso resultante puede variar dependiendo del evento actual y de la selección posterior por parte del trabajador. Las actividades no están conectadas entre sí.
Las actividades a menudo son programáticas. Una secuencia repetible, como seleccionar un conjunto de posibles propietarios de tarjetas de crédito de una base de datos, puede ser automatizada.	La gente determina principalmente las actividades. El manejo de un cliente con un error de facturación es realizado por una persona que usa el juicio para determinar la mejor resolución de este caso particular.
Los documentos externos no forman parte del proceso.	Los documentos externos desempeñan un papel clave. Por ejemplo, los recibos proporcionan un registro de cómo el problema que debe ser resuelto comenzó.

Tabla 3-1 Características entre BPM y ACM [22].

La Gestión Flexible de Casos desde el 2014 es uno de los proyectos desarrollados por la OMG el cual es un organismo internacional sin fines de lucro que tiene como misión desarrollar estándares tecnológicos que proporcionen valor real a miles de industrias verticales.

OMG se dedica a reunir a sus miembros internacionales de usuarios finales, vendedores, agencias gubernamentales, universidades e instituciones de investigación para desarrollar y revisar estas normas a medida que las tecnologías cambian a lo largo de los años [23]; es este mismo organismo quien ha diseñado y desarrollado los principales lenguajes estándares de modelado.

A continuación, se describen los lenguajes estándares de modelado con mayor aceptación en el mercado.

3.2.2 Lenguaje Modelo y Notación de Procesos de Negocio

Business Process Modeling Notation (BPMN) es una notación precisa, completa y gráfica para documentar procesos de negocio bien definidos. Resuelve muchas ambigüedades encontradas en las especificaciones del proceso textual asignando actividades a actores específicos.

El análisis de los modelos resultantes puede utilizarse para impulsar iniciativas de mejora de procesos, independientemente de si los procesos son automatizados o manuales. Debido a que el modelo gráfico es fácilmente comprensible por personas no técnicas, sirve como un puente que permite la colaboración entre las partes interesadas del negocio y el personal de Tecnologías de la Información [24].

3.2.3 Lenguaje Modelo y Notación de Decisión

Decision Model and Notation (DMN) es una notación para la especificación precisa de las decisiones y reglas empresariales. DMN es fácilmente legible por los diferentes tipos de personas involucradas en la gestión de decisiones. Éstos incluyen gente de negocios que especifica las reglas, pero también monitorea su aplicación; Analistas de negocios que transforman la aportación de los usuarios en modelos de decisión detallados; y desarrolladores de software que los implementan en sistemas empresariales. DMN está diseñado para trabajar junto con BPMN o CMMN proporcionar un mecanismo para modelar la toma de decisiones dentro de los modelos de procesos y modelos de casos [24].

3.2.4 Lenguaje Modelo y Notación para la Gestión de Casos

Case Management Modeling Notation (CMMN) es una notación gráfica utilizada para capturar métodos de trabajo que se basan en el manejo de "casos" que requieren diversas actividades que pueden realizarse en un orden impredecible en respuesta a situaciones en evolución. Utilizando un enfoque centrado en el evento y el concepto de un expediente, CMMN expande los límites de lo que puede modelarse con BPMN, incluyendo esfuerzos de trabajo menos estructurados y aquellos impulsados por trabajadores del caso [24].

Los escenarios empresariales pueden ser muy dinámicos en cuanto a que el flujo de trabajo involucrado y los resultados dependen mucho de decisiones ad-hoc y eventos empresariales. El Modelo y Notación para la Gestión de Casos de OMG es un lenguaje de modelado declarativo que proporciona una forma ágil y dinámica de abordar las situaciones que pueden ocurrir dentro de un contexto, conocido como un caso.

3.2.5 Diferencias entre lenguajes para el modelado

En la siguiente Figura 3-2 se enlistan las principales características entre los diferentes lenguajes de modelado diseñados por la OMG.

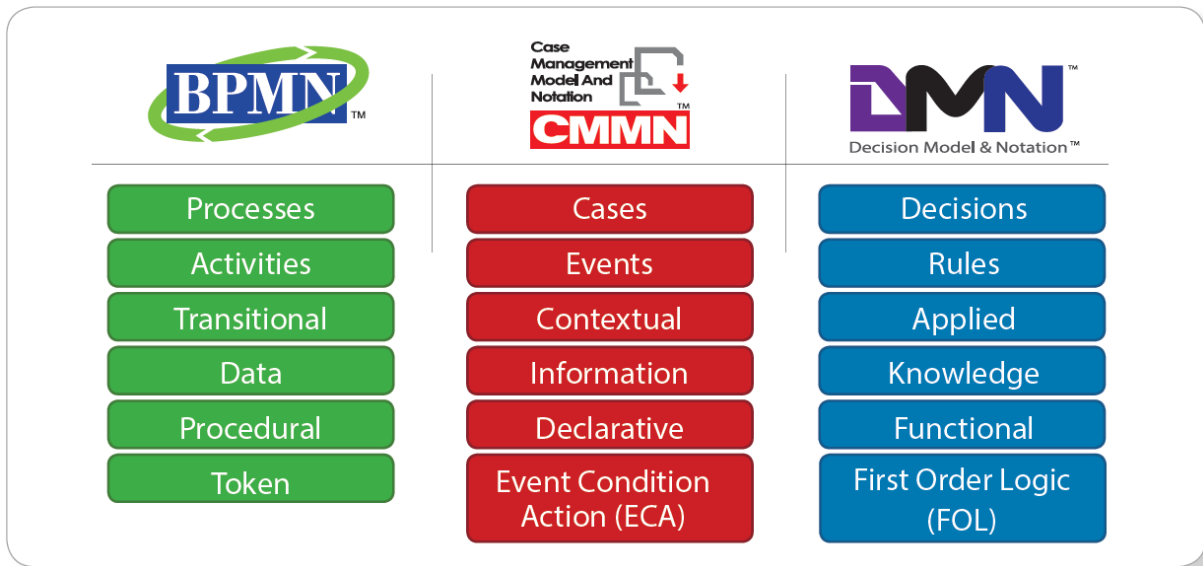


Figura 3-2 Diferencias en términos de conceptos básicos, enfoque y semántica [24].

En esta tabla se resumen las características descritas en las secciones 3.2.2, 3.2.3 y 3.2.4 del presente informe; se concluye que mientras BPMN, CMMN y DMN pueden ser utilizados de forma independiente, también fueron cuidadosamente diseñados para ser complementarios. De hecho, muchas organizaciones requieren una combinación de modelos de proceso para sus flujos de trabajo prescriptivos, modelos de casos para sus actividades reactivas y modelos de decisión para sus reglas de negocio más complejas y multicriterio. Estas organizaciones se beneficiarán al utilizar las tres normas en combinación, seleccionando cuál es la más apropiada para cada tipo de modelado de actividades.

3.2.6 Ejemplos de aplicaciones industriales de ACM

En la siguiente Tabla 3-2 se muestran algunos ejemplos de la aplicación industrial de la Gestión Flexible de Casos:

Sector Público	Servicios Financieros	Aseguradoras	Cuidado de la Salud	Energía y Utilidad	Servicio de Negocios
Procesamiento de Impuestos	Origen de Prestamos	Reglamento de Reclamaciones	Proceso de reclamación	Gestión de Emergencias	Gestión del Servicio al Cliente
Desempleo	Servicio a Inversores	Suscripciones	Historial de Paciente	Transmisión de Procesos	Gestión de Reclamaciones
Servicios Sociales	Apertura de cuenta nueva	Gestión de Políticas	Gestión de contactos legales	Recursos Exploratorios	Estudio de Aplicaciones
Investigación Policiaca	Resolución de Disputa		Inscripción de miembros		Gestión de Campañas
Trabajos Jurídicos	Gestión de Patrimonios				Gestión de Proyectos

Tabla 3-2 Ejemplo de aplicaciones industriales de ACM [25].

Basado en los ejemplos anteriores de la industria de ACM pueden definirse diferentes categorías de ACM. En 2009, Forrester [26][25] definió tres tipos de categorías de manejo de casos:

- Investigación
- Gestión de incidentes
- Solicitudes de servicio

La segmentación de Forrester es similar a la definición de Gartner [27]. Además, Forrester define tres factores clave entre los segmentos: mitigación de riesgos, control de costos y experiencia del cliente.

Por ejemplo, el caso de gestión de reclamaciones de la industria servicios de negocios, se clasifica como solicitud de servicio e implica mitigación del riesgo y experiencia del cliente. La compañía de seguros tiene el objetivo de administrar sus riesgos mediante la moderación de los pagos de la reclamación y el objetivo de lograr la experiencia del cliente mediante una buena y rápida regulación de reclamaciones. La Figura 3-3 muestra las categorías de manejo de casos de Forrester [26].

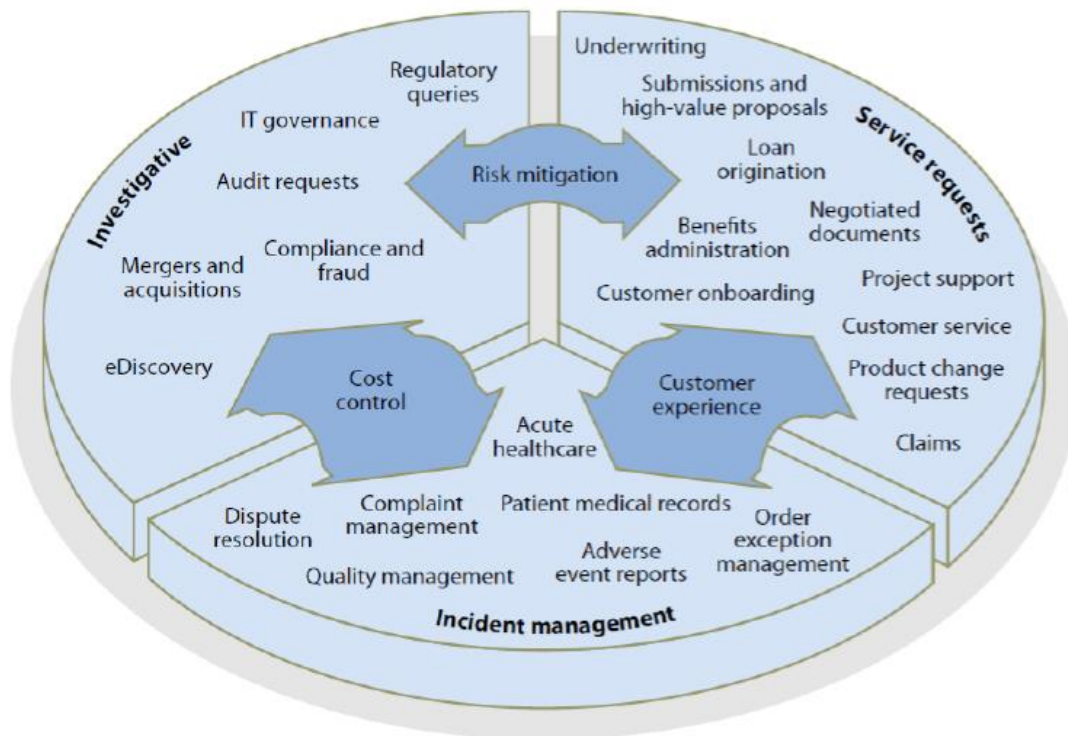


Figura 3-3 Tres categorías en la Gestión Flexible de Casos [26].

En 2014, Gartner segmentó ACM en cuatro áreas clave: Solicitud de servicio, Investigación, Incidente y Proceso a la decisión. Para cada área, se definen características y ejemplos [27], así se muestran en la siguiente Tabla 3-3.

Área	Características	Ejemplos
Solicitud de Servicios	Flujo de trabajo pesado	- La gestión de reclamaciones de seguros. - Asistencia sanitaria.
Investigación	Gran cantidad de datos.	- Investigación criminal.
Incidente	Alta colaboración.	- Gestión de emergencias
Proceso a la Decisión	Gran cantidad de reglas o políticas.	- Cumplimiento de auditorías.

Tabla 3-3 Áreas de segmento para casos ACM.

3.2.7 Elección de lenguaje para modelar la Gestión de Emergencias

Esta sección proporciona la respuesta a la primera pregunta de investigación.

RQ1: ¿Es el lenguaje CMMN la mejor opción para modelar la gestión flexible de casos aplicado a la gestión de emergencias con respecto a otras notaciones de modelado de negocios?

Como se ha venido desarrollando la investigación en el presente informe, **el lenguaje CMMN es el más apropiado para modelar la gestión flexible de casos con un enfoque de aplicación hacia la Gestión de Emergencias**. Lo anterior obedece a que CMMN está diseñado para la gestión flexible de casos en los que la flexibilidad es una de sus características principales, es decir, no se tiene un flujo de control predecible y es adaptativo en tiempo de ejecución; lo anterior dependiendo del entorno y de las decisiones que pueda elegir el trabajador del caso al momento de ejecutar el modelo.

Así mismo tenemos otras características que respaldan esta elección [28]:

- Pueden intervenir múltiples procesos que deberán estar coordinados entre sí y que, en su resolución en un momento dado, no deba seguirse una secuencia de flujo predefinido.
- Pueden necesitar una gran agilidad y flexibilidad para acometer todo tipo de cambios de forma inmediata, aunque, naturalmente, sin perder el Control.
- Pueden intervenir múltiples personas, tanto internas (empleados de diversos departamentos) como externas (el propio solicitante, colaboradores a la entidad, etc.) que deberán poder mantener entre ellos una comunicación fluida al tiempo que controlada.
- Puede requerirse la manipulación de gran cantidad de información estructurada (datos) y no estructurada (documentos) de diversa índole que debe poder ser creada y/o capturada, almacenada, consultada y eliminada de la forma más automática, fluida y controlada.

Después de haber estudiado y analizado el estado del arte de la Gestión flexible de casos y de haber dado respuesta a la pregunta de investigación RQ1, así mismo, se alcanza el objetivo específico SO1 del trabajo de investigación:

SO1: Fundamentar la elección del lenguaje CMMN para modelar la gestión flexible de casos aplicada a la gestión de emergencias con respecto a otras notaciones con base al estudio y análisis del estado del arte.

A continuación, se explica a detalle las características y notación del lenguaje CMMN.

3.3 Lenguaje CMMN

Esta sección proporciona la respuesta a la segunda pregunta de investigación.

RQ2: ¿Cuáles son los conceptos teóricos del Lenguaje CMMN?

Case Management Model and Notation (CMMN) versión 1.0 fue creado por Object Management Group (OMG) [2] y publicado en 2014.

Es una notación complementaria al Modelo de Negocio y Notación (BPMN) que se concentra en el flujo de control para explicar “como” realizar los procesos de negocio de manera imperativa, el flujo de control se encuentra completamente especificado, es predecible y repetible; mientras que CMMN es declarativa en la que se describe lo que se permite y se rechaza en el proceso. BPMN, CMMN y el Modelo de Decisión y la Notación (DMN) son las tres notaciones de modelado de negocios de OMG.

CMMN, es un modelo (es decir un diagrama) puede tener uno o múltiples casos siendo este el concepto principal en CMMN, y es similar a un proceso. Un caso se constituye por un **fichero del caso** (contenedor de datos de casos), y se describe mediante un **plan de caso** (un modelo o diagrama), por los **roles** asignados a los trabajadores del caso y por **entradas y salidas** que interactúan con él entorno.

Debido a la naturaleza de la gestión flexible de casos no todo el trabajo que sucede en un caso es modelado. En particular, la interacción de los trabajadores de casos con los casos y los datos de los casos no puede ser ni parcial ni completamente modelada. En la mayoría de las situaciones, no se modela la forma en que los datos entran en el caso. Los datos pueden ser añadidos, eliminados o modificados por los trabajadores del caso en cualquier momento durante el procesamiento de un caso sin necesidad de modelarlo.

3.3.1 Tiempo de diseño vs tiempo de ejecución.

En el contexto de CMMN, un caso tiene dos fases distintas: Modelado en tiempo de diseño y planificación y ejecución en tiempo de ejecución.

3.3.1.1 Modelado en tiempo de diseño (*Design-time Modelling*)

Durante esta fase, los analistas de negocios están involucrados en el modelado, lo que incluye definir las tareas que pueden ser:

- Requeridas, siempre parte de segmentos predefinidos en el modelo del caso.
- Opcionales, las cuales no se necesitan completar para que finalicen.
- Discrecionales, pueden ser seleccionados (instanciados) por los trabajadores del caso en tiempo de ejecución bajo los parámetros especificados en las tablas de planificación.

En esta etapa el modelo sirve como un plan para el manejo de casos en tiempo de ejecución, pero con flexibilidad de decisión para el trabajador del caso.

3.3.1.2 Planificación y ejecución en tiempo de ejecución (*Run-time Planning and Execution*)

En esta fase, los trabajadores de casos crean y ejecutan el plan, en particular realizando las tareas previstas, sin embargo, el plan puede evolucionar continuamente, debido al mismo o a otros elementos (tareas y/o etapas) que son instanciadas por el trabajo del caso en tiempo de ejecución. Esta característica fundamental de la gestión flexible de casos se conoce como “flexibilidad de tiempo de ejecución” la cual permite a los participantes del proceso responder a desafíos o nuevos requisitos que no se tuvieron en cuenta durante la planificación de los procesos.

Esta planificación en tiempo de ejecución se basa en la información que se ha puesto a disposición del caso. Cambiar el comportamiento del caso significa cambiar un proceso de caso. Esto incluye cambiar el procesamiento de casos de funcionamiento.

En la siguiente Figura se muestra una comparativa entre el tiempo de diseño y tiempo de planeación y ejecución.

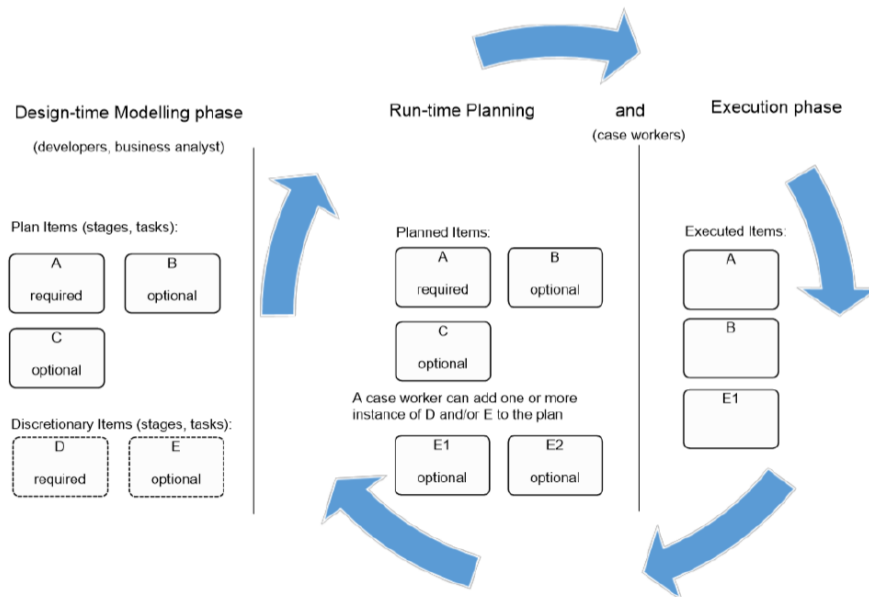


Figura 3-4 Tiempo de diseño vs Tiempo de planeación y ejecución [29].

3.3.2 Elementos de CMMN

Los elementos de CMMN son explicados a continuación:



3.3.2.1 Fichero del caso (Case File Model)

El fichero del caso no es parte de la norma; sin embargo, toda la información, o referencias a la información, que se requiere como contexto para manejar un caso, se define por un fichero del caso es algún tipo de modelo de información lógica.

La información en el fichero del caso sirve como contexto para levantar eventos y evaluar expresiones, así como punto de referencia para entradas y salidas de tareas. Un fichero del caso también sirve como una carpeta de datos que es accesible (a través de entradas y salidas) a otros sistemas y personas fuera del caso.

Un fichero del caso es compuesto por ítem(s) del fichero del caso, el cual está representado en la notación como una figura en forma de “Documento” y puede representar una pieza de información de cualquier naturaleza, que va desde estructurada hasta no estructurada y de simple a compleja, cuya información puede definirse sobre la base de cualquier “lenguaje” de modelado de información. Un ítem del fichero del caso por ejemplo puede ser una carpeta o documento, una jerarquía de la carpeta entera que haga referencia o contenga otros ítems de ficheros del caso o un documento XML.



3.3.2.2 Modelo del plan del caso (Case Plan Model)

El modelo del plan del caso comprende todos los elementos que representan el plan inicial del caso, tales como etapas, tareas, hitos y todos los elementos que apoyan la evolución futura del plan mediante la planificación en tiempo de ejecución por parte de los trabajadores de casos. Un modelo del plan del caso es representado mediante una figura en forma de carpeta o folder.

El comportamiento en tiempo de ejecución es descrito por el ciclo de vida de algunos objetos instanciados en el modelo del plan del caso; en este escenario los 4 elementos en la definición del plan son:

- Tareas / Tareas discrecionales
- Fragmentos del plan / etapas
- Oyente de eventos
- Hitos

Los elementos que interactúan durante la ejecución son relacionados mediante conectores a través de dependencias.

Así mismo existen dos tipos de elementos para el control del flujo:

- Sentries: Definen condiciones de entrada y salida a los elementos del plan.
- Tabla de planificación: Permiten que los elementos del plan puedan ser aplicados dinámicamente, habilitan los elementos discrecionales.

Así mismo existe una clase denominada control del elemento del plan en donde se especifica los valores por defecto que tendrán los elementos del plan del caso, antes mencionados, indicando si estas instancias deben ser completadas y/o pueden iniciarse mediante valores predefinidos.

3.3.2.3 Tarea (Task)



Una tarea es un elemento atómico de trabajo importante para el negocio que debe ser realizada por un agente externo (humano o máquina). Existen 3 tipos de tareas:

Tarea humana: Este tipo de tarea es realizada por un trabajador del caso y pueden ser de dos tipos:



Bloqueo: La tarea está esperando hasta que se complete el trabajo asociado con la tarea.



No bloqueo: La tarea no está esperando a que el trabajo se complete, se completa inmediatamente al ser instanciada.



Tarea de proceso: Se puede utilizar en el caso de invocar un proceso de negocio. Es posible intercambiar valores de variables entre las tareas proceso.



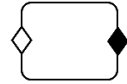
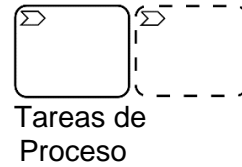
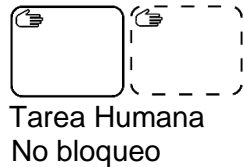
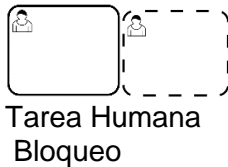
Tarea de caso: Se puede instanciar (llamar) a otro caso. Es posible intercambiar valores entre variables de tareas de casos.

Dado que se cumple un criterio de entrada, hay dos maneras de activar una tarea:

- Activación manual de un usuario: Un usuario puede decidir activar una tarea o desactivarla
- Activación automática.

Cualquier tipo de tarea (humana, de proceso, de caso) puede ser discrecional y está disponible para que el trabajador del caso la instancie de acuerdo con su criterio y con la frecuencia que decida.

Las tareas se representan mediante una figura en forma de cuadro con las esquinas redondas; sin embargo, las tareas requeridas u opcionales son dibujadas con líneas continuas, mientras que las tareas discrecionales se representan con líneas discontinuas.



3.3.2.4 Centinela (Sentries) - Criterios de entrada y salida

Es un elemento del modelo del plan del caso y es relacionado al control del flujo. Los sentries definen los criterios según los cuales los elementos del modelo plan del caso (tareas, etapas, hitos) se habilita (entrada) o se terminan (salida).

Un sentry “vigila” las situaciones (eventos) importantes que ocurren y que influyen en la evolución del caso. Un sentry es una combinación evento-condición.

De acuerdo con la parte que vigila existen 2 tipos de sentries:

- ◊ Entrada: Un sentry utilizado como criterio de entrada es representado por una figura en forma de rombo sin relleno.
- ◆ Salida: Un sentry empleado como criterio de salida es representado por una figura en forma de rombo con relleno sólido.

Con relación al evento-condición un sentry puede consistir en 2 partes:

- On -Part: Especifica el evento que sirve como disparador.
- If-Part: Especifica la condición que se evalúa sobre el fichero del caso.

Los sentries pueden tomar unas de las siguientes formas de acuerdo con su composición:

- 1.- Un OnPart (parte del evento) y una IfPart (parte de la condición) en el formulario
on <evento>
if <condición>
- 2.- Una parte del evento en el formulario
on <evento>
- 3.- Una para de la condición en el formulario
if <condición>

Un centinela debe tener al menos una IfPart al menos un OnPart. Un IfPart define una condición adicional que se comprueba cuando se cumplen todas las OnParts del sentry. Cuando se recibe el evento, la condición se puede aplicar para evaluar si el evento tiene acción o no.

Si todos los OnParts han ocurrido y el sentry tiene IfPart se evalúa como “verdadero” y el estado del sentry es “Satisfecho”. Un sentry satisfecho activa el elemento del plan al que se refiere a él (entrada o salida) en el fragmento del plan.

Cuando el sentry es referenciado por uno de los criterios de entrada del elemento del plan se habilitará una tarea, etapa o hito. Mientras que cuando un sentry es referenciado por uno de los criterios de salida del elemento del plan una tarea o etapa se terminará (saldrá).

Un sentry y una tarea es una regla de combinación ECA (evento-condición-acción).

3.3.2.5 Oyente de Eventos (Event Listener)



En CMMN se intenta predefinir los eventos y sus respectivas causas. Se representa mediante un círculo con doble línea. Estos eventos CMMN predefinidos se denominan eventos estándar y se pueden categorizar en 3 tipos:

- Cualquier cosa que pueda pasar a la información en el archivo del caso. Un elemento creado, modificado o eliminado.
- Cualquier cosa que pueda suceder en las etapas, tareas e hitos que denotan transiciones en la evolución del archivo del caso. Como una tarea que se ha iniciado, cancelado o terminado.
- El paso del tiempo no puede ser capturado a través de estos eventos por esta razón, la clase EventListener (Oyente de eventos) se ha introducido en el modelo CMMN esta es una clase abstracta que se especializa en dos clases concretas:



TimerEventListener (Oyente de eventos de tiempo). Se utiliza para capturar instancias de los elementos definidos en el plan con tiempos predefinidos. La forma en que se representa es con una figura en forma de círculo con doble línea y un reloj al centro.



UserEventListener (Oyente de eventos de usuario). Con el fin de capturar eventos que son planteados por un usuario y que se utilizan para influir en los procedimientos del caso directamente en lugar de indirectamente. La forma en que se representa es con un círculo con doble línea y un usuario al centro.

Estos EventListeners tienen su propio ciclo de vida predefinido CMMN, de modo que cualquier tiempo transcurrido, así como cualquier evento de usuario, pueden ser capturados como "eventos estándar" y por lo tanto manejados de manera uniforme.

3.3.2.6 Hito (Milestone)



Es un elemento definido en la planeación del caso que representa un objetivo alcanzable, definido para permitir la evaluación del progreso del caso. Se representa mediante una forma en rectángulo con extremos medio redondeados.

Ningún trabajo está directamente asociado con un hito, pero la finalización de un conjunto de tareas o la disponibilidad de entregas clave (información en el archivo de casos) suele conducir a lograr un hito. No está asociado con ningún trabajo, sino que señala que se han alcanzado ciertas condiciones dentro del caso. Por lo tanto, puede tener cero o más criterios de entrada, que definen, cuando se alcanza un hito.



Un hito y puede tener cero o más criterios de entrada, que definen cuando se ha alcanzado un hito.

Como un hito es un elemento definido en la planeación, la finalización de un hito puede usarse como criterio de entrada para otras tareas y etapas. De esta manera, un hito puede ser utilizado para traer las etapas lógicas dentro de un caso en orden.

Un hito puede ser:

- Se adjunta a una etapa para representar el objetivo alcanzado cuando se ha completado la etapa.
- Independiente para representar una condición de negocio importante que no es el resultado directo de ninguna etapa.

3.3.2.7 Etapa (Stage)



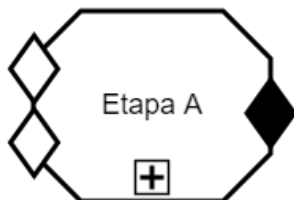
Son fragmentos de un plan que pueden ser monitorizados. Se consideran episodios de un caso es decir sub-casos. Una etapa es definida como un grupo de actividades relevantes para el negocio que pueden realizarse para, con o por una instancia de caso, con el fin de lograr uno (o más) de los hitos que pertenecen a esa etapa.

Una etapa puede contener cualquier elemento necesario para construir y seguir desarrollando planes de casos.

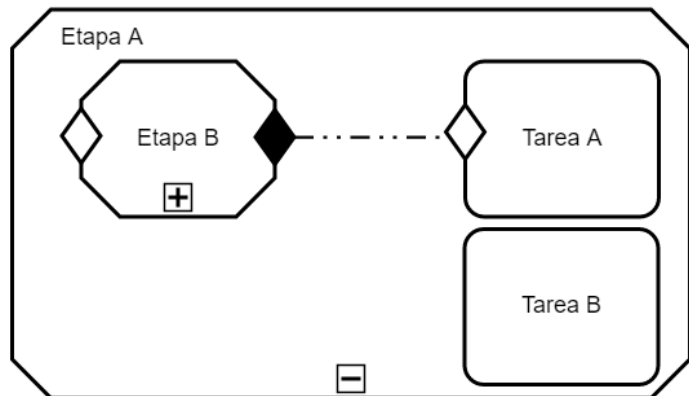
Una etapa puede contener otros elementos del plan como tareas y otras etapas. Es un concepto recursivo - las etapas pueden anidarse dentro de otras etapas. La etapa "más externa" del caso es su Modelo del plan del caso.

Una etapa tiene su representación en tiempo de ejecución, por lo tanto, su progreso y finalización pueden rastrearse en función de su ciclo de vida.

Una Etapa está representada por una forma de rectángulo con esquinas anguladas y un marcador en forma de signo "+" o "-" en una pequeña caja en su centro inferior ("+" o "-" designan etapas expandidas o colapsadas).



Etapa colapsada con dos criterios de entrada y uno de salida



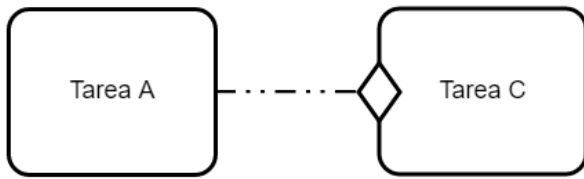
Etapa expandida que contiene otra etapa colapsada y 2 tareas.

3.3.2.8 Conectores (Connector)



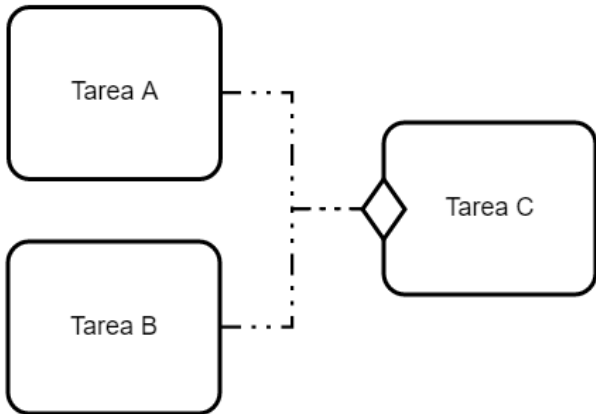
Los conectores pueden ser utilizados para representar dependencias entre los elementos del Plan, se representan mediante una línea discontinua.

Cuando los conectores van desde la parte On de un Sentry se pueden visualizar dependencias entre los elementos de un plan, a continuación, unos ejemplos:



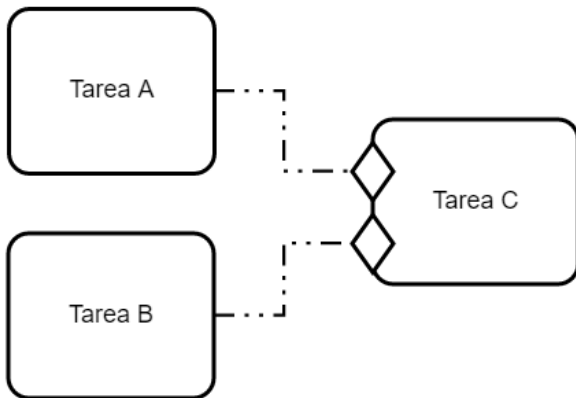
Secuencia de Flujo:

Sí la Tarea A es completada, se activa la Tarea C.



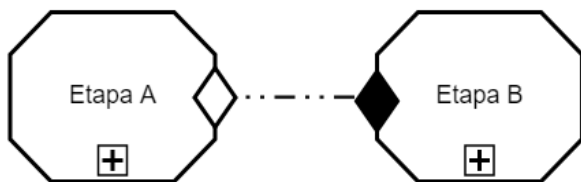
Conexión AND:

Sí la Tarea A y la Tarea B son completadas, se activa la Tarea C.

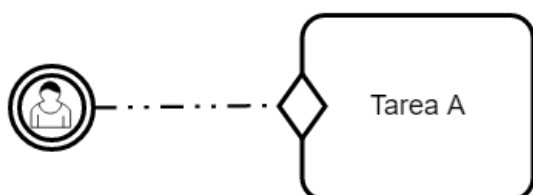


Conexión OR:

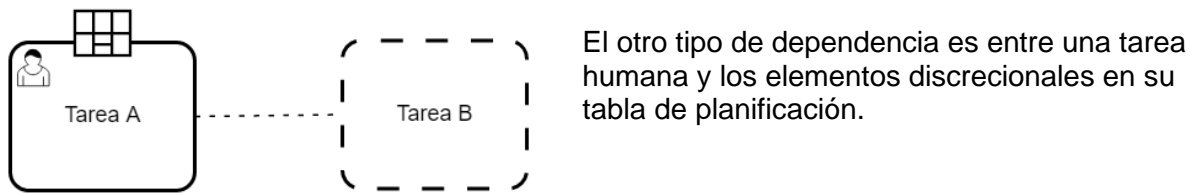
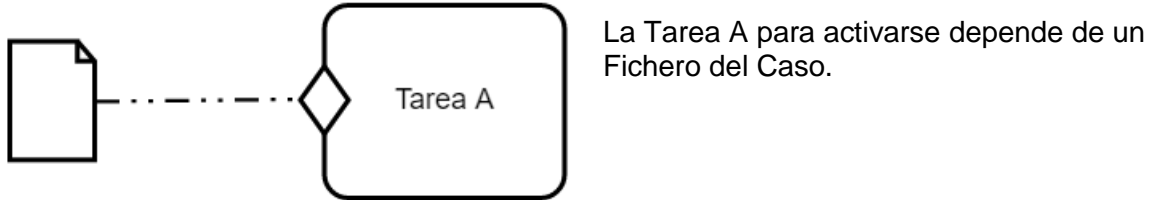
Sí la Tarea A o la Tarea B es completada, se activa la Tarea C.



La Etapa B depende del criterio de salida de la Etapa A para activarse.



La Tarea A para activarse depende de que se active el Evento de usuario.

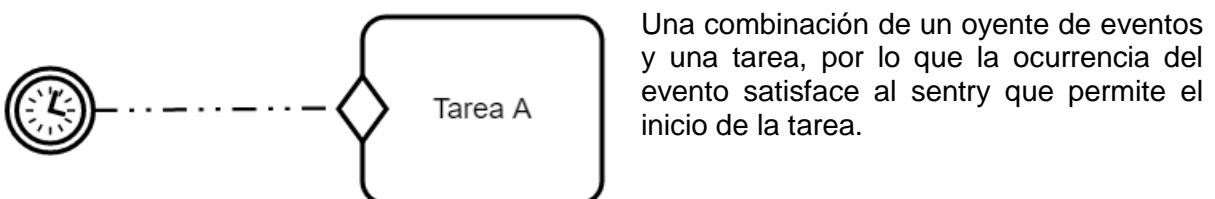
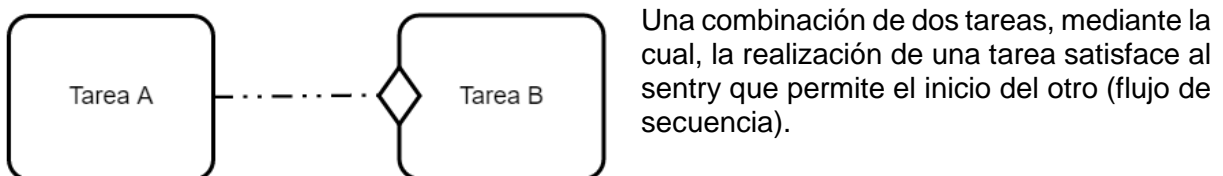


3.3.2.9 Fragmento del Plan (Plan Fragment)

Es una agrupación lógica de los elementos de un plan, es visto como un contenedor de elementos del plan, posiblemente dependientes unos de otros y opcionalmente sentries que definen los criterios según los cuales los elementos del plan están habilitados y terminados.

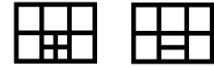
Un fragmento del plan se representa mediante una forma de rectángulo con líneas discontinuas y esquinas suavemente redondeadas y un marcador en forma de signo "+" en una pequeña caja en su centro inferior. Cuando el fragmento del plan se expande tiene un marcador en forma de un signo "-" en una pequeña caja en su centro inferior, cuando se expande un fragmento del plan, los elementos contenidos en él se hacen visibles.

El fragmento de un plan puede representar patrones PlanItem - Sentry de cualquier complejidad. Ejemplos de fragmento de plan son:



A diferencia con el elemento de notación de Etapa, el fragmento del plan no es monitorizado, es decir es únicamente una agrupación lógica.

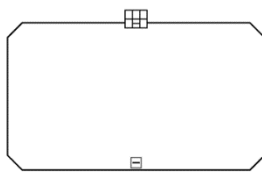
3.3.2.10 Tabla Planificación (Planning Table)



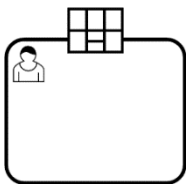
Una de las ventajas de la notación CMMN es que los usuarios (Trabajadores de caso) pueden “planear” en tiempo de ejecución, cuando seleccionan elementos discretos; en términos de determinar un subconjunto de PlanItemDefinitions que puede ser considerado para la planificación en un determinado contexto.

Una tabla de planificación puede tener varios elementos de tabla (es decir, tablas de planificación y elementos discretos). Una tabla de planificación define el ámbito (scope) de la planificación.

Las tablas de planificación se pueden asignar a:



Etapa: la tabla de planificación puede usarse para planificar instancias de tareas y etapas en esa instancia de etapa.



Tareas humanas: la tabla de planificación se puede utilizar para planificar instancias de tareas y etapas en el escenario que contiene la tarea humana.

Con las tablas de planificación, es posible hacer que los elementos discretos sean aplicables dinámicamente para la planificación utilizando reglas de aplicabilidad. Estas reglas especifican si un elemento de tabla es "aplicable" ("elegible", "disponible") para la planificación, en base a las condiciones que se evalúan sobre la información en el archivo de caso. Una tabla de planificación es representada mediante una forma de “tabla”.

Planificación en tiempo de ejecución: reglas de aplicabilidad

Con las tablas de planificación es posible hacer que los elementos discretos sean aplicables dinámicamente para la planificación. Las reglas de aplicabilidad se usan para especificar si un elemento de la tabla es “elegible” para la planificación, con base a las condiciones que se evalúan sobre la información contenida en el expediente del caso.

Si la condición de Reglas de aplicabilidad se evalúa como "verdadero", entonces el elemento de la tabla es aplicable para la planificación, durante la planificación, sólo los elementos discretos, para los que la Regla de aplicabilidad se evalúan como "verdaderos", deben ser mostrados al Trabajador de casos.

3.3.2.11 Reglas de propiedad de comportamiento y Decoradores

Las reglas de propiedad de comportamiento se utilizan para derivar valores booleanos que pueden influir en la ejecución de una instancia de caso. Ver Tabla ## Reglas de propiedad de comportamiento.






Regla	Descripción
Regla de Activación Manual 	Determina si la instancia de tarea o etapa debe pasar al estado 'activado' o 'activo'. Esta regla se evalúa y utiliza cuando se satisfacen los criterios de entrada de la instancia de tarea o etapa. Si esta regla se evalúa como "verdadero", la instancia de etapa o tarea pasa de "disponible" a "habilitado" (y debe esperar a que se active manualmente), de lo contrario pasará de "disponible" a "activo".
Regla Requerida 	Determina si el hito, etapa o tarea que tiene esta condición debe estar en el estado 'completado', 'terminado', 'fallado' o 'deshabilitado' para que su instancia de etapa primaria pase al estado 'completado'.
Regla de Repetición 	Debe evaluarse cuando se instancia la instancia de jalón, etapa o tarea y transiciones al estado 'disponible', y su valor booleano se debe mantener durante el resto de la vida del hito, etapa o tarea.
Regla de Aplicabilidad 	Una regla de aplicabilidad no es una regla de negocio, y representa una regla para la ejecución de procesos / casos en la capa de aplicación. Se utilizan para especificar si un elemento de tabla es "elegible" para la planificación, en función de las condiciones que se evalúan sobre la información en el archivo de caso. Si la condición de la regla de aplicabilidad se evalúa como "verdadero", entonces el elemento de la tabla (tabla de planificación y / o elemento discrecional) es aplicable para la planificación. Esta regla es una regla de 'restricción', ya que por defecto un elemento de tabla es aplicable para la ejecución (la regla se evalúa como verdadero).
Autocompletar 	Una instancia de etapa DEBE completar si un usuario no tiene ninguna opción para hacer más planificación o trabajar con la instancia de etapa. Este atributo de etapa controla la finalización de la etapa. Si el atributo se evalúa como 'verdadero', una etapa se completa cuando hay: - no hay hijos activos (tareas o sub etapas) Y, - todo requerido (Regla Requerida se evalúa como "true") los hijos están en estado 'disabled', 'completed', 'terminated' o 'failed' Si el atributo se evalúa como 'falso', una etapa se completa cuando hay: - no hay hijos activos (tareas o sub etapas) Y, - no hay artículos discretionales O (la terminación manual y todos los hijos requeridos están en estado 'inhabilitado', 'terminado', 'terminado' o 'fallado'). En otras palabras, la etapa requiere que un usuario la complete manualmente, lo cual es a menudo apropiado para etapas que tienen artículos discretionales.

Tabla 3-4 Reglas de propiedad de comportamiento

Así mismo, los Modelos de plan de casos, las etapas, tareas e hitos pueden tener decoradores relacionados a las Reglas de propiedad de comportamiento, en la siguiente tabla se resume dicha relación:












Aplicabilidad del Decorador	Tabla de Planificación 	Criterio de Entrada 	Criterio de Salida 	Autocompletar 	Activación Manual 	Requerido 	Repetición 
Modelo del plan del caso 	✓		✓	✓			
Etapa 	✓	✓	✓	✓	✓	✓	✓
Tarea 	Tarea Humana Solamente	✓	✓		✓	✓	✓
Hito 		✓				✓	✓

Figura 3-5 Tabla resumen de aplicabilidad del decorador.

3.3.3 Ejecución Semántica de CMMN

La semántica de ejecución de CMMN es descrita por el ciclo de vida de instancias dentro del contexto del modelo de gestión flexible de casos. Las instancias del caso CMMN y los elementos del plan pasan por un ciclo de vida de estados durante su ejecución.

Dependiendo de su estado, diferentes acciones pueden llevarse a cabo para interactuar con ellos. Además, las transiciones de estado pueden activar automáticamente cambios en otros elementos del plan. El ciclo de vida concreto de un elemento del plan depende de su definición del elemento del plan.

La Figura 3-4 Conceptos del ciclo de vida de CMMN, proporciona una visión general de los conceptos de ciclo de vida de los elementos:

- Instancia del caso
- Tarea
- Etapa (concepto similar del ciclo de vida como tarea)
- Oyente de evento
- Hito (concepto de ciclo de vida similar como él oyente de eventos).

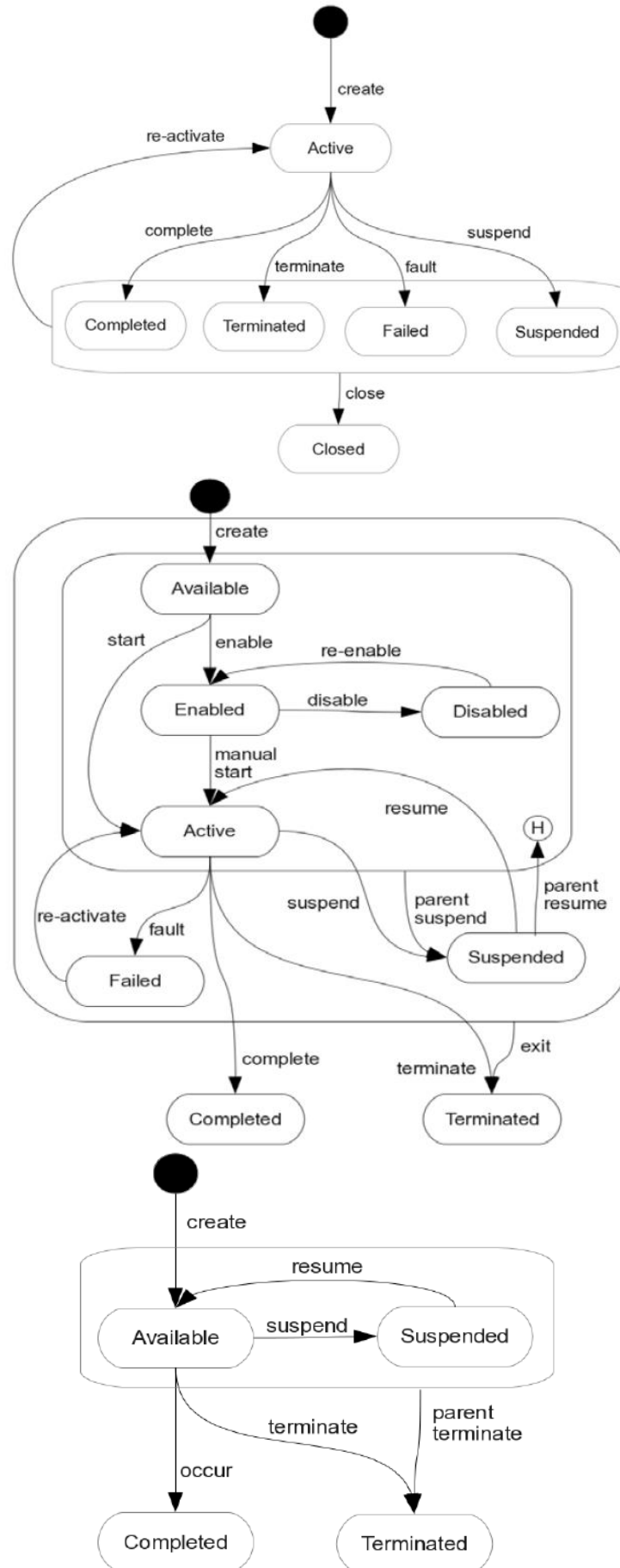


Figura 3-6 Conceptos del ciclo de vida de CMMN [2].

Para comprender el concepto de ciclo de vida de un elemento CMMN durante la etapa de ejecución, el siguiente caso servirá de ejemplo

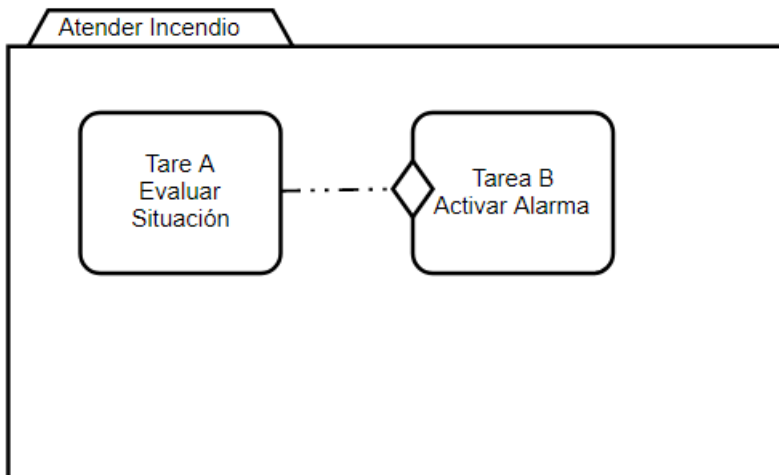


Figura 3-7 Ejemplo para explicar el concepto del ciclo de vida de CMMN

El modelo de plan de caso "Atender Incendio" en la Figura 3-5 Ejemplo para explicar el concepto de ciclo de vida CMMN contiene dos tareas humanas:

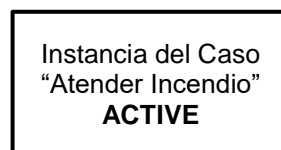
- Tarea A "Evaluar Situación".
- Tarea B "Activar Alarma".

Ambas tareas están conectadas por un conector y un criterio de entrada (sentry). Cuando una tarea no tiene criterio de entrada, progresa automáticamente desde el estado DISPONIBLE a ENABLED.

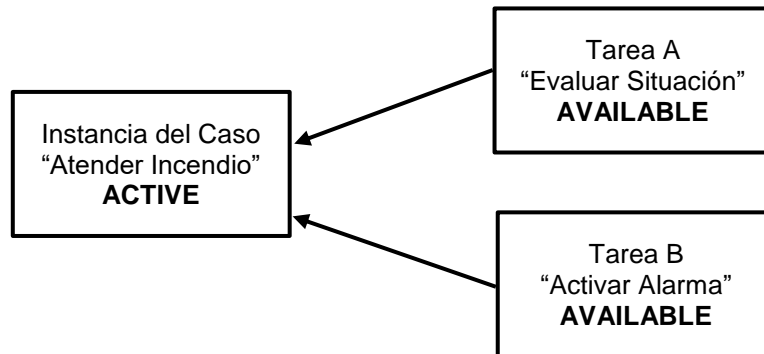
El sentry en la Tarea B expresa que la tarea B "Activar Alarma" puede ser realizada cuando la tarea A "Evaluar Situación" ha terminado.

Las siguientes actividades tienen lugar para cada objeto. Para mostrar las transiciones para el ciclo de vida de cada objeto, se da una presentación paso a paso para cada uno:

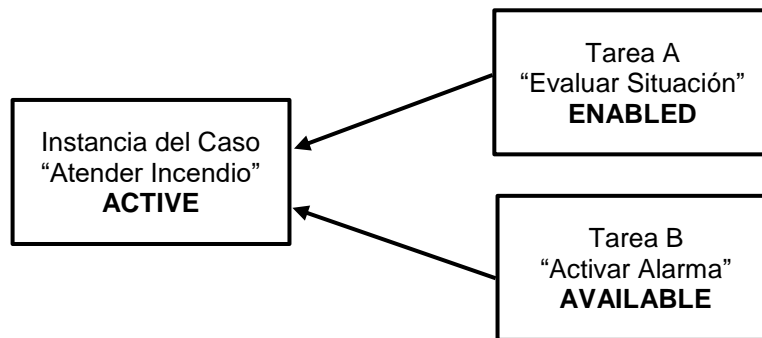
1. Un Jefe de Emergencia recibe información de que la actividad de un Incendio e instancia el caso, se crea una nueva instancia del caso "Atender Incendio" en estado ACTIVE.



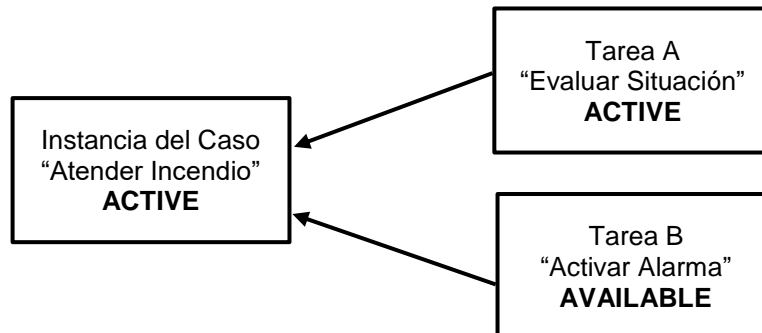
2. Dos instancias para cada tarea humana son creadas automáticamente en estado AVAILABLE (Disponible).



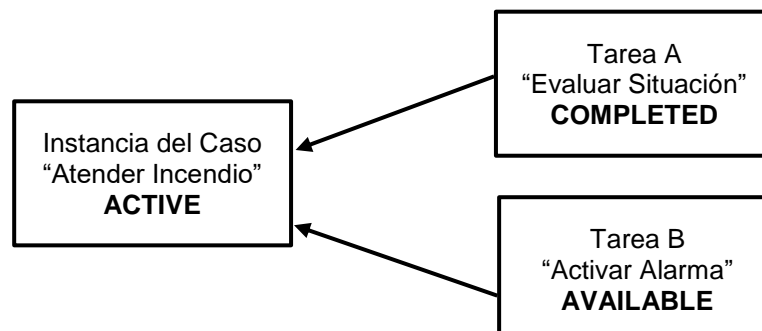
3. La tarea A no tiene un criterio de entrada sentry para iniciar, por lo que inmediatamente alcanza el estado ENABLED.



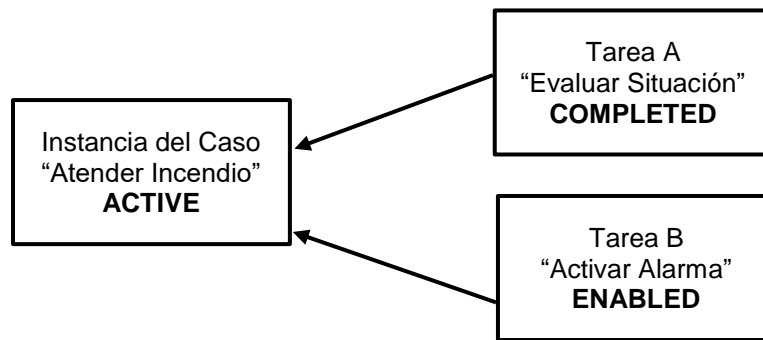
4. El Jefe de Emergencia al recibir la notificación inicia la tarea A que llega a estado ACTIVE. La tarea A se agrega en la lista de tareas de los usuarios que tienen acceso al modelo.



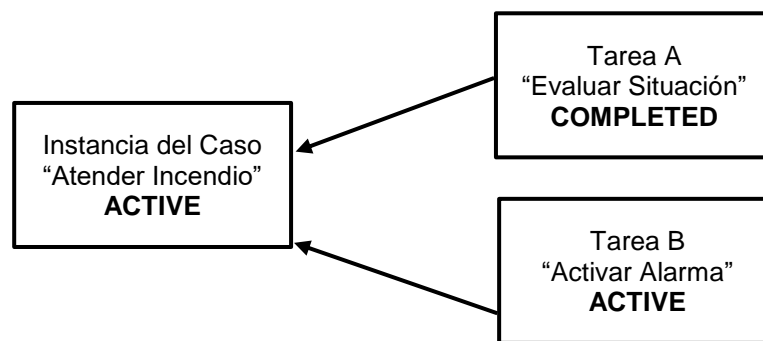
5. El usuario asignado, en este caso el Jefe de Emergencia, completa la tarea A. La tarea A llega a estado COMPLETED.



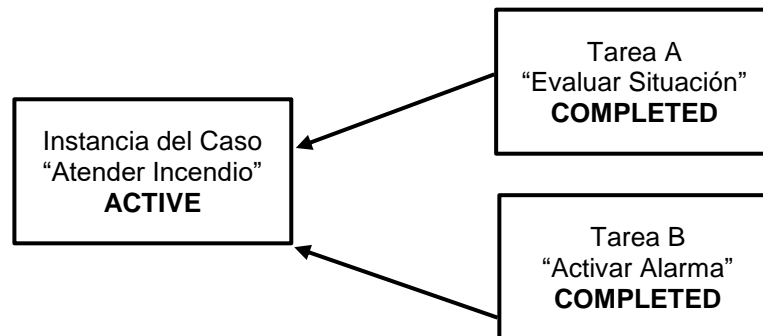
6. La transición al estado COMPLETED de la tarea A activa el sentry de la tarea B. El estado de la tarea B es ENABLED.



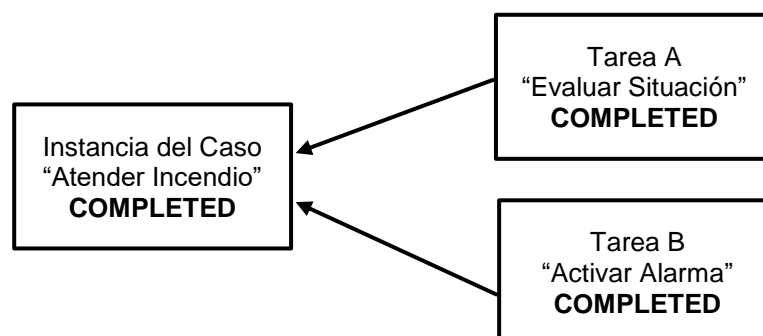
7. Un Jefe de Intervención inicia la tarea B que llega a estado ACTIVE. La tarea B se asigna a la lista de tareas de los usuarios que tienen acceso.



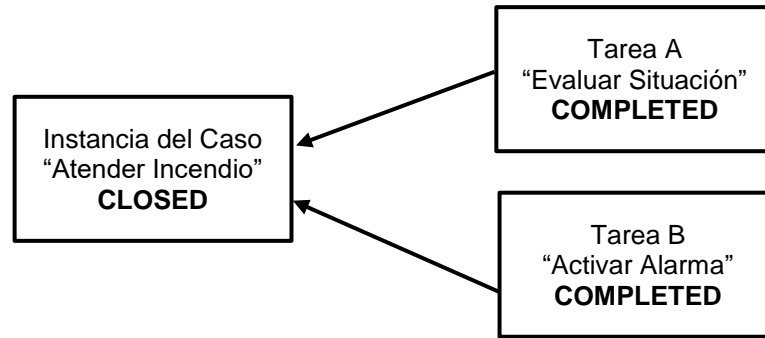
8. Un Jefe de Intervención completa la tarea B, la cual llega a estado COMPLETED.



9. Con ambas tareas en estado COMPLETED, en el ejemplo la instancia del caso automáticamente alcanza el estado COMPLETED.



10. El Jefe de Emergencia puede ser la instancia del caso, la cual llega a estado CLOSED.



Después de haber estudiado y comprendido los conceptos del lenguaje CMMN, se ha dado respuesta a la pregunta de investigación RQ2, y se alcanza el objetivo específico SO2 del trabajo de investigación:

SO2: Estudiar y analizar a profundidad los conceptos teóricos del Lenguaje estándar de CMMN.

En el siguiente capítulo se detalla el marco de tecnológico a utilizar en este trabajo de investigación.

4. Marco tecnológico

4.1 Acerca de la Plataforma de Camunda

Para el experimento del caso práctico que se aborda en el capítulo 5 de este trabajo de investigación se utiliza la plataforma Camunda (proviene de los verbos latinos "capere"- "comprender" y "munda" - "limpia"), la cual es una empresa con sede en Berlín, Alemania; fue fundada en marzo de 2008 y actualmente emplea a más de 70 personas. Originalmente inició como un negocio de consultoría de BPM, hoy su principal giro es ser proveedor de software BPM. La empresa es administrada por los fundadores y propietarios y no ha recaudado fondos externos. Ha sido rentable desde la fecha de fundación y ha crecido de forma exponencial.

Camunda BPM es una plataforma ligera y de código abierto [30] para BPM. Es una buena combinación para el desarrollo general de software y proporciona una potente alineación de negocios basada en BPMN para flujos de trabajo estructurados, CMMN para casos menos estructurados y DMN para reglas de negocio.

Los casos típicos de uso de Camunda BPM son:

- Procesamiento Directo (STP)
- Gestión del flujo de trabajo humano
- (Dinámica) Gestión Flexible de Casos
- Automatización de reglas de negocio

Algunos módulos de la plataforma Camunda pueden ser consumidos en versión Demo, Enterprise o Gratis en línea.

4.2 Arquitectura

La arquitectura de la plataforma Camunda BPM está conformada por su motor de núcleo y un kit de herramientas que abarcan el ciclo de vida para un proceso de negocios.

El núcleo de Camunda BPM es un motor de ejecución de modelos que soporta los estándares OMG BPMN 2.0 para automatización de procesos, CMMN 1.1 para gestión de casos y DMN 1.1 para la gestión de decisiones.

Camunda BPM incluye un conjunto de aplicaciones para ayudar a modelar, ejecutar y administrar aplicaciones de proceso que se ejecutan en el motor principal de Camunda.

Estas aplicaciones interactúan con la API REST pública del motor central. También los usuarios pueden crear sus propias aplicaciones que también pueden utilizar la API Java pública del motor central.

A continuación, observamos en la figura 4-1 la arquitectura de Camunda.

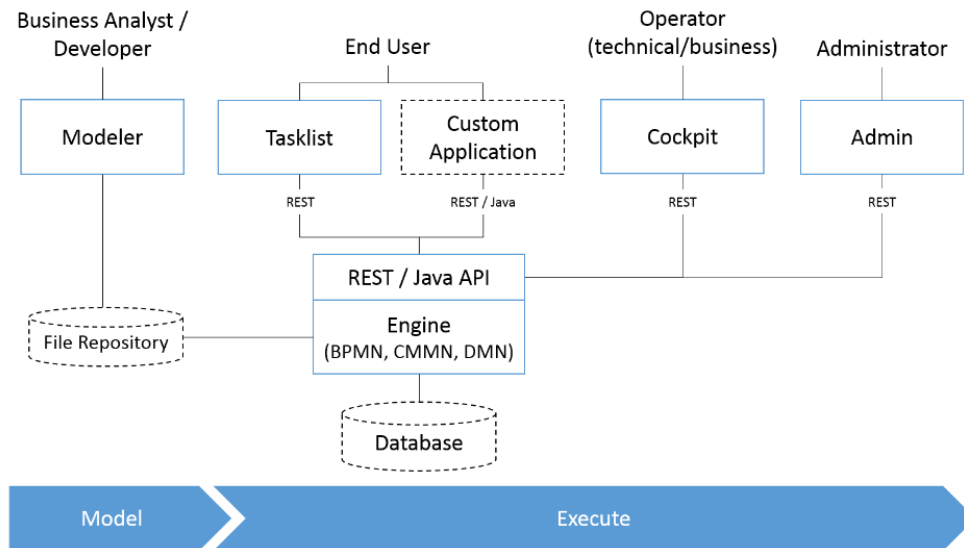


Figura 4-1 Arquitectura de Camunda BPM [6].

Observamos en la figura 4-1, las herramientas que conforman la plataforma Camunda BPM y la colaboración de cada uno de los usuarios de acuerdo con el perfil del rol profesional con el que cuentan.

En los siguientes temas se exponen las aplicaciones con las que cuenta la plataforma de Camunda BPM con relación al ciclo de vida de un proceso de negocios para las etapas de Diseño, Implementación, Ejecución y Análisis.

4.3 Etapa de Diseño: Camunda Modeler

Camunda Modeler es una aplicación que cuenta con presentación de interfaz de usuario en escritorio o en web, ambas son gratuitas para editar diagramas de procesos BPMN, modelos CMMN y tablas de decisiones en DMN. Es muy fácil de usar, lo que significa que los analistas de negocios pueden utilizarlo, así como los desarrolladores, trabajando en los mismos diagramas.

Además del modelado visual, Camunda Modeler también permite editar todas las propiedades que son necesarias para la ejecución técnica. Dado que Camunda Modeler trabaja directamente en los archivos BPMN, CMMN y DMN en formato XML, los desarrolladores pueden combinarlo fácilmente con los IDE's (Integrated Development Environment) de trabajo como Eclipse, Netbeans o IntelliJ.

Camunda Modeler está basado en la librería de JavaScript mxGraph [10], la cual es de código abierto y con ella se realizó el sitio Draw.io el cual es un editor libre con entorno web en donde se pueden editar diagramas de flujo, de procesos, de organización, de casos, de red, de circuitos electrónicos.

En esta etapa el perfil del usuario generalmente es el Analista de Negocios o el Desarrollador. En la siguiente figura 4-2 se presenta el entorno de Camunda Modeler en versión escritorio, la cual se utiliza en este trabajo para diseñar el modelo del caso de estudio. Así mismo en el apéndice C se encuentra la guía para su instalación correspondiente.

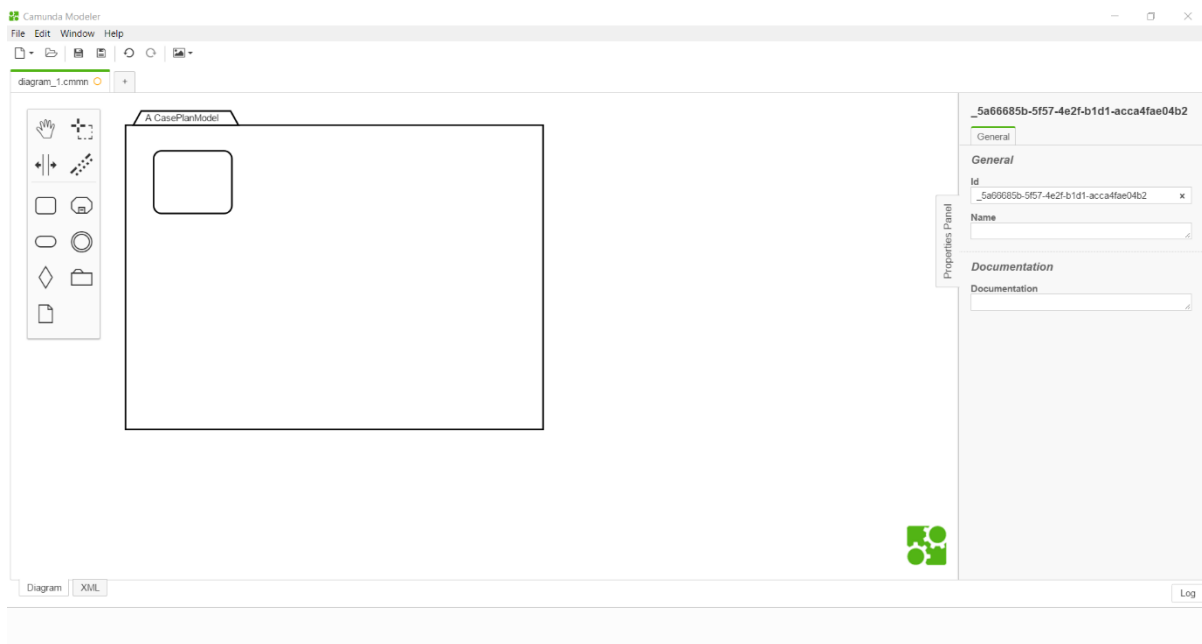


Figura 4-2 Interfaz de Camunda Modeler.

4.4 Etapa de Implementación: Núcleo del Motor

El núcleo de Camunda BPM es un motor de ejecución para BPMN, CMMN y DMN.

Es ligero y requiere menos de 3 MB de espacio en disco. Puede ejecutarse en cualquier Máquina Virtual Java (JVM) y viene con integración extendida para diferentes contenedores en tiempo de ejecución.

Se puede acceder al motor a través de la API REST o mediante la API Java y utilizar las integraciones existentes con Spring y Java EE. Se puede implementar tareas de servicio BPMN en código Java o hacer uso de los conectores REST y SOAP incorporados.

Además de Java, también se puede aplicar secuencias de comandos, expresiones y plantillas para la implementación de procesos.

El motor Camunda es rápido y el clustering para la escalabilidad horizontal es sencillo ya que el motor es apátrida: varias instancias pueden compartir la misma base de datos.

Al instalar y configurar la plataforma Camunda BPM que se explica en el apéndice C, se integra el núcleo del motor.

Sin embargo, no toda la notación del estándar CMMN 1.1 es soportada, en la Figura 4-3 se resalta en color naranja la notación que es soportada por el núcleo del motor de Camunda, dicha limitante será tomada en cuenta al modelar el ejemplo del caso de estudio del capítulo 5 del presente trabajo.

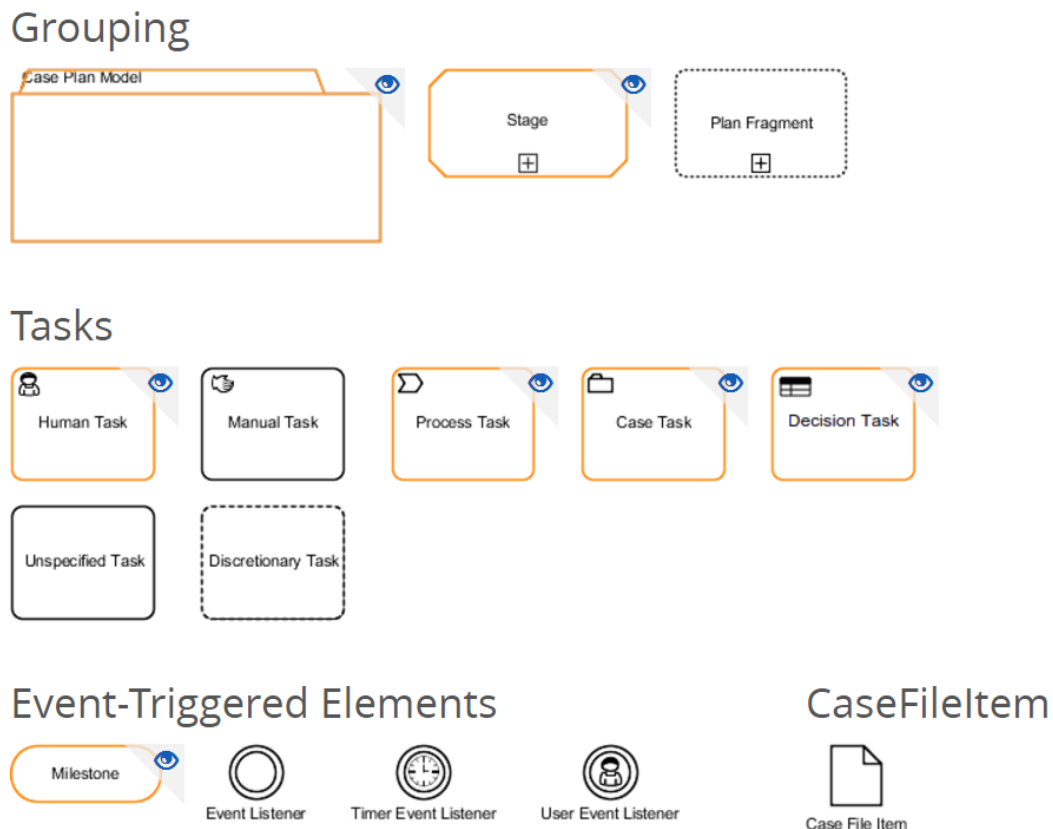


Figura 4-3 Notación CMMN 1.1 soportada [31].

4.5 Etapa de Ejecución: Lista de Tareas

La interfaz para la ejecución de un modelo es en la que interactúa el usuario final, puede realizarse a través de la aplicación de lista de tareas (TaskList) de Camunda o a través de una aplicación personalizada del usuario al consumir la API REST o la API Java del Motor de Camunda.

- Camunda TaskList

Los Usuarios finales usan Camunda Tasklist para organizar y trabajar en las tareas que se supone que deben completar. En BPMN, una tarea se describe como una tarea de usuario. En CMMN, el mismo concepto se llama tarea humana.

La lista de tareas Camunda se ejecuta completamente en el navegador web (Internet Explorer, Chrome, etc.). Puede personalizar el diseño de la Lista de tareas (colores, logotipo) en su diseño corporativo y ampliar su funcionalidad utilizando sus propios complementos. También se puede integrar a través de LDAP con Active Directory.

Durante la ejecución del proceso (BPMN) o del caso (CMMN), el motor Camunda creará tareas basadas en el modelo BPMN o CMMN predefinido. El motor asigna esas tareas a los usuarios finales según se definen en el modelo. Cuando los usuarios finales hacen clic en una tarea de la Lista de tareas de Camunda, verán un formulario de tarea que muestra información y se les pide que ingresen datos. Una vez que finalicen de trabajar en el formulario de tareas, lo completan haciendo clic en el botón "Completar", lo que le solicita al Motor de Camunda que continúe la ejecución del modelo.

La instalación y configuración de la TaskList de Camunda se encuentra detallada en el apéndice C de este documento, a continuación, en la figura 4-3 se presenta una imagen de la interfaz web de la aplicación.

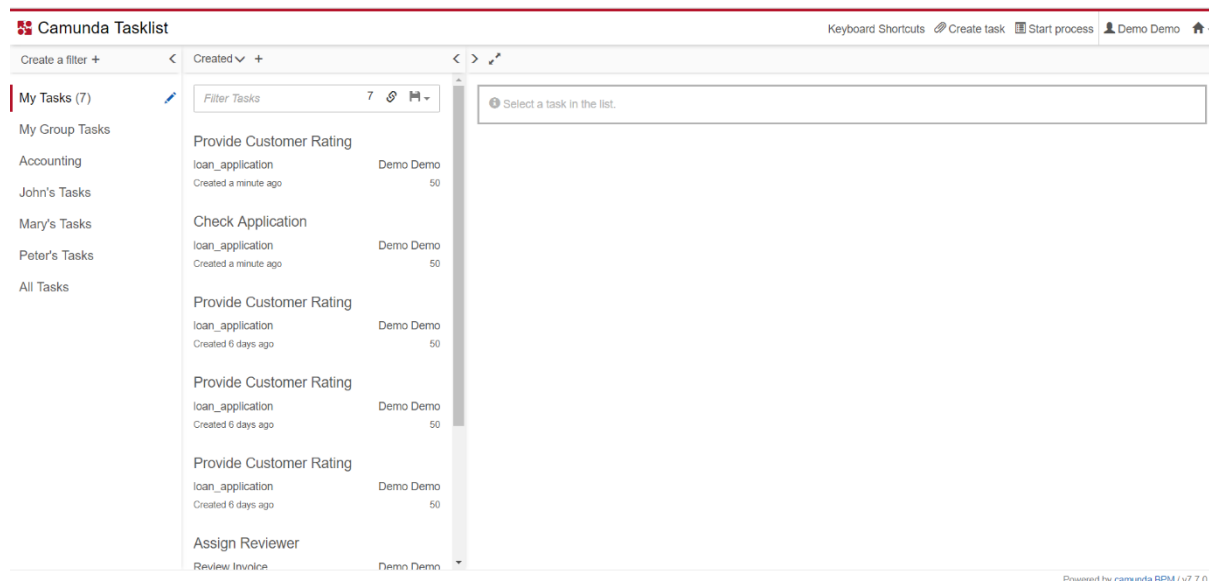


Figura 4-4 Interfaz de Camunda TaskList.

- Aplicaciones Personalizadas

Es un caso de uso muy común para mejorar las aplicaciones existentes con la plataforma Camunda BPM. Por ejemplo, si un Proveedor Independiente de Software (Independent Software Vendor, ISV), quiere ofrecer a sus clientes funciones BPMN, CMMN o DMN como parte de un producto.

Estos son algunos ejemplos de servicios que un ISV puede ofrecer en una interfaz de usuario personalizada, al incrustar Camunda:

- Una consola de administración de tareas.
- Un panel que muestra información en tiempo real sobre procesos o casos en curso.
- Visualizaciones BPMN o DMN con anotaciones específicas de la aplicación.
- Edición de diagramas BPMN o tablas de decisión DMN, por ejemplo, para la personalización de productos. Esto también puede incluir componentes reutilizables que agregue a la paleta de modelado.

Técnicamente, integrar una aplicación con el núcleo del motor de Camunda sería a través de la API REST o directamente en API Java, así mismo si la aplicación está escrita en Java.

Para casos de uso como la personalización de productos (es decir, permite a los usuarios editar BPMN o DMN), inserta las bibliotecas Javascript detrás de Camunda Modeler en su propia aplicación web.

Si se requiere ser un ISV y brindar servicios con la plataforma Camunda BPM, la empresa Camunda brinda los servicios de consultoría.

4.6 Etapa de Análisis: Cockpit y Admin

Las herramientas con las que dispone la plataforma Camunda BPM para la etapa de Análisis son Cockpit y el Administrador.

- Cockpit

Los operadores técnicos usan el Cockpit para inspeccionar las instancias de proceso en ejecución y completadas, lo anterior para cuidar los incidentes.

Un operador puede ser una persona técnica, como un administrador de sistemas o un ingeniero de Desarrollo, pero también puede ser alguien que trabaja en un departamento de negocios.

Debido a eso, Cockpit es altamente personalizable y permite crear una herramienta adaptada, o incluso diferentes versiones de Cockpit para casos de uso específicos y grupos de usuarios.

Además de las instancias de proceso, los operadores también pueden inspeccionar las instancias de decisión completas y cambiar las tablas de decisión DMN al instante ("Edición en vivo").

Cockpit también proporciona análisis sobre sus procesos, lo que le permite establecer un Supervisor de la Actividad Empresarial (Business Activity Monitoring, BAM) y crear informes KPI.

En la siguiente figura 4-4 se muestra la interfaz de la herramienta Cockpit en la que se puede monitorear los procesos en ejecución o completados.

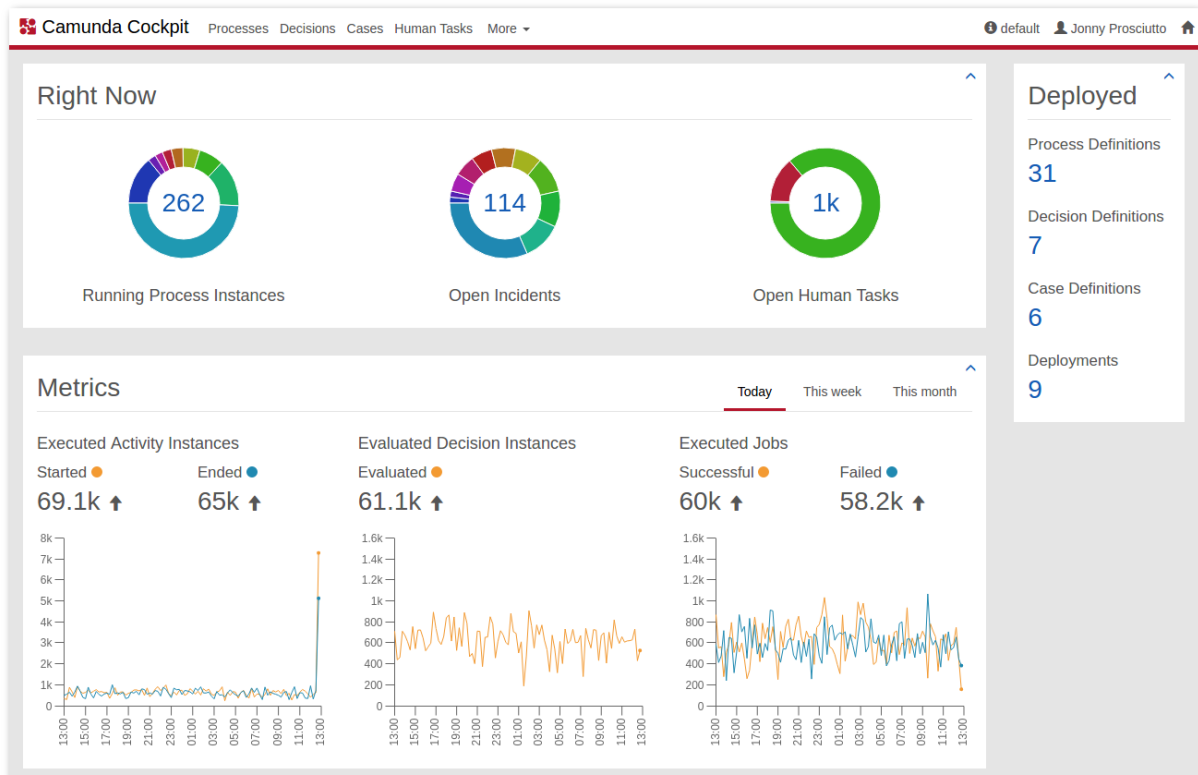


Figura 4-5 Interfaz de Camunda Cockpit.

- **Admin**

Los administradores utilizan Admin para administrar a sus usuarios, organizarlos en grupos y conceder permisos. Camunda desacopla la identificación de los usuarios de su autorización para ejecutar determinadas acciones.

Para la identificación, se puede utilizar la administración de usuarios que se suministra con Camunda o un sistema de administración de usuarios ya existente que puede ser integrado con Camunda a través de LDAP.

Una vez que un usuario se ha identificado (por ejemplo, presentando la pantalla de inicio de sesión en una aplicación web Camunda), Camunda los autorizará basándose en los permisos definidos en Camunda Admin.

Los permisos que se definan en Camunda Admin se considerarán en todos los niveles, es decir, al llamar a la API del motor principal, la API REST y al utilizar una aplicación web Camunda. La figura 4-5 muestra como es la interfaz de la herramienta Admin incluida en la plataforma Camunda BPM.

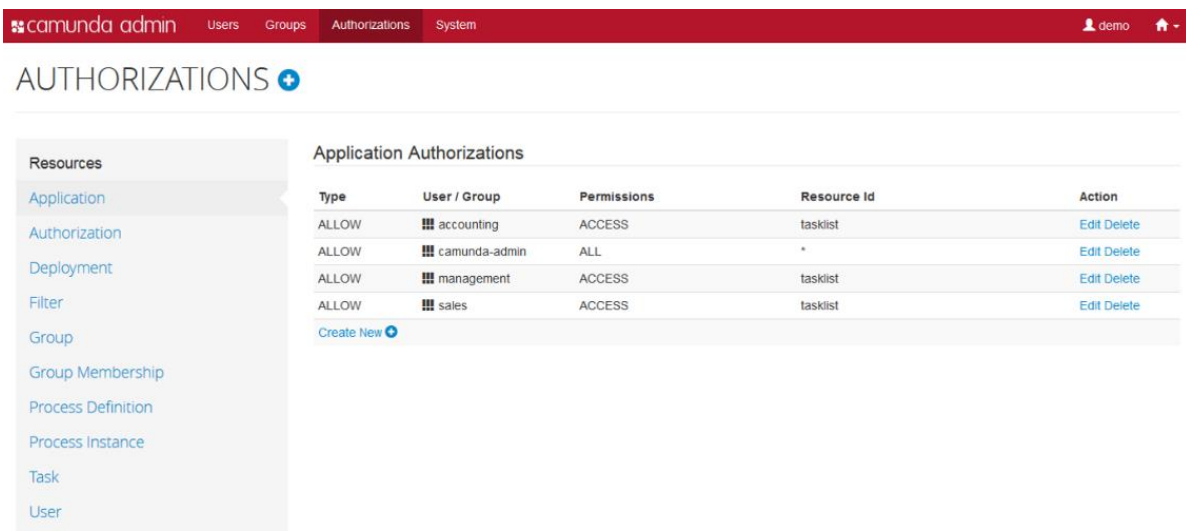


Figura 4-6 Interfaz de Camunda Admin.

En el siguiente capítulo 5 se realiza el caso de estudio en el cual tiene por objetivo que un procedimiento de respuesta expresado en lenguaje natural sea modelado en lenguaje CMMN para posteriormente ser ejecutado.

5. Caso de estudio del procedimiento de respuesta

5.1 Introducción

Con la finalidad de facilitar el planteamiento de los casos de estudio en los procedimientos de respuesta, es importante realizar una breve introducción acerca de los planes de autoprotección y la normatividad que los rige, así mismo en el Apéndice A se proporciona un glosario de las palabras claves de acuerdo con el contexto de la gestión de emergencia en materia de protección civil.

Posteriormente se aborda el caso de estudio de un procedimiento de respuesta de una emergencia de tipo incendio para un hospital, para ello primero se conocerá el procedimiento de respuesta en formato texto, tal cual como se presenta en el plan de autoprotección, después se modelará y se ejecutará el modelo que resulte de transformar el procedimiento de respuesta en formato texto a modelo en lenguaje CMMN.

Para finalizar se exponen los resultados de la ejecución del modelo del caso de estudio y las recomendaciones al respecto.

5.2 Planes de Autoprotección

De acuerdo a la legislación normativa reguladora de protección civil [32] en España, en donde debido a la magnitud y trascendencia de los valores que están en juego en las situaciones de emergencia exige poner a contribución los recursos humanos y materiales pertenecientes a todas las Administraciones Públicas, a todas las organizaciones y empresas, e incluso a los particulares, a los que, por tal razón, mediante ley, con arreglo al artículo 30.4 de la Constitución, podrán imponérseles deberes para hacer frente a los casos de grave riesgo, catástrofe o calamidad pública en materia de protección civil.

Esta ley tiene como tarea fundamental promover la autoprotección entre los ciudadanos ante situaciones de emergencia, por ello, en los supuestos de emergencia que requieran la actuación de la protección civil, una parte muy importante de la población depende, al menos inicialmente, de sus propios recursos.

De ahí que, como primera fórmula de actuación, haya de establecer un complejo sistema de acciones preventivas e informativas, al que contribuye en buena medida el cumplimiento de los deberes que se imponen a los propios ciudadanos, con objeto de que la población adquiera conciencia sobre los riesgos que puede sufrir y se familiarice con las medidas de protección que, en su caso, debe utilizar.

Por lo anterior, el ministerio del interior suscribió las normas básicas [33] que han de cubrir los centros, establecimientos y dependencias dedicados a actividades que puedan dar origen a situaciones de emergencia, en estas se define entre otras cosas que el plan de autoprotección es:

El documento que establece el marco orgánico y funcional previsto para un centro, establecimiento, espacio, instalación o dependencia, con el objeto de prevenir y controlar los riesgos sobre las personas y los bienes y dar respuesta adecuada a las posibles situaciones de emergencia, en la zona bajo responsabilidad del titular de la actividad, garantizando la integración de estas actuaciones con el sistema público de protección civil.

Así mismo, se indica que los centros, establecimientos y dependencias dispondrán de un sistema de autoprotección, dotado con sus propios recursos, y del correspondiente plan de autoprotección para acciones de prevención de riesgos, alarma, la intervención coordinada, el refugio, la evacuación, el socorro, la información en emergencia a todas aquellas personas que pudieran estar expuestas al riesgo, y la solicitud y recepción de ayuda externa de los servicios de emergencia.

Además, en el Anexo II, del mismo decreto, se especifica el contenido mínimo que debe contener un plan de autoprotección en España.

Como parte modular de un plan de autoprotección se tienen que definir los riesgos que pueden provocar una emergencia dentro del centro, establecimiento o dependencia.

Uno de los contenidos más importantes de los planes de autoprotección son los procedimientos de respuesta, en ellos encontramos las acciones a realizar por cada uno de los actores que intervienen ante una situación de emergencia en particular.

Además, el procedimiento de respuesta debe ser explicado de una manera clara para que los actores interpreten correctamente las actividades a realizar, para ello la descripción de las mismas debe dar respuesta a las siguientes preguntas:

- ¿Qué debe hacerse?
- ¿Quién debe actuar?
- ¿Cuándo se debe actuar?
- ¿Cómo debe actuarse?
- ¿Dónde debe actuarse?

5.3 Caso de estudio en el procedimiento de respuesta de un Incendio

Para realizar el siguiente caso de estudio, se va a utilizar del plan de autoprotección el procedimiento de respuesta de una emergencia de tipo incendio del Hospital Universitario y Politécnico La Fe ubicado en la ciudad de Valencia.

Así mismo, con la finalidad de facilitar el diseño del modelado del caso de estudio, utilizaremos el documento del Apéndice D “Formulario CMMN”, derivado del estándar CMMN, el cual nos ayuda a obtener una formulación de los requisitos basados en CMMN y a identificar los elementos del modelo de comportamiento / plan CMMN y las interrelaciones entre sus elementos.

5.3.1 Conocer el Procedimiento de Respuesta.

Es recomendable que antes de leer los procedimientos de respuestas de los planes de autoprotección se identifique a los actores que intervienen, así mismo sus abreviaturas, misiones y funciones a desempeñar.

Jefe de Emergencia (JE)	
Titular	Director(a) del Hospital
En su ausencia	Director Médico o Jefe de Guardia
Funciones específicas	<ul style="list-style-type: none"> • Será quien coordine las acciones necesarias para la implementación y mantenimiento del plan de autoprotección. • En caso de emergencia asumirá la coordinación y actuación de los equipos de emergencia, cumpliendo y haciendo cumplir el conjunto de consignas y actuaciones desarrolladas en el plan de autoprotección. • Para cumplir lo anterior deberá tener una formación adecuada respecto a: <ul style="list-style-type: none"> • Conocimiento exacto de los riesgos generales y particulares de cada zona del centro. • Conocimiento de la existencia y uso de los medios materiales de protección de los que se disponga. • Conocimiento de los medios de extinción de que se disponga. • En general, conocimiento exacto de todo el plan de autoprotección.

Jefe de Intervención (JI)	
Titular	Subdirector de Infraestructura
En su ausencia	Jefe de Servicio de ingeniería o Jefe de Mantenimiento
Funciones específicas	<ul style="list-style-type: none"> • Dependerá directamente del Jefe de Emergencia. • Valorará la emergencia y asumirá la dirección y coordinación de los equipos de intervención. • Estará siempre en contacto con el Jefe de Emergencia, facilitando información sobre la evolución de la emergencia y aconsejándole sobre las decisiones a tomar. • Estará perfectamente enterado de los medios de prevención y extinción de que dispone el Centro, así como de su uso.

Equipo de Primera Intervención (EPI)	
Integrantes	Todo el Personal del Hospital
Dependencia	Del Jefe de Intervención (JI)
Funciones específicas	<ul style="list-style-type: none"> • Estarán formados estos Equipos por absolutamente todo el personal que trabaje en el Centro. Para lo cual se facilitará una información mínima, con referencia a situación y ubicación de los medios de protección y extinción, así como a las acciones a desarrollar en caso de emergencia.

Equipo de Segunda Intervención (ESI)	
Jefe de Equipo	Maestro Industrial o Técnico de Mantenimiento.
Integrantes	Personal de Mantenimiento, Celadores, Vigilantes de Seguridad.
Dependencia	Del Jefe de Intervención (JI)
Funciones específicas	<ul style="list-style-type: none"> • En caso de que el Maestro Industrial sea el Jefe de Intervención, delegará el puesto de Jefe de Equipo de ESI en un mecánico. • Recibirá la instrucción necesaria y actuarán cuando la gravedad de la emergencia no pueda ser controlada por los Equipos de Primera Intervención. Aunque hay que tener presente que deberán ser avisados siempre. • En caso de requerirse ayuda exterior (bomberos), facilitarán información a éstos y les servirán de apoyo en su cometido.

Equipo de Alarma y Evacuación (EAE)	
Integrantes	Todo el Personal del Hospital
Dependencia	Del Jefe de Intervención (JI)
Funciones específicas	<ul style="list-style-type: none"> • Estarán formados estos Equipos por absolutamente todo el personal que trabaje en el Centro. Para lo cual se facilitará una información mínima, con referencia a situación y ubicación de los medios de protección y extinción, así como a las acciones a desarrollar en caso de emergencia.

Equipo de Primeros Auxilios (EPA)	
Integrantes	Todo el Personal del Servicio de Urgencias
Dependencia	Del Jefe de Intervención (JI)
Funciones específicas	<ul style="list-style-type: none"> • Prestar asistencia a los posibles afectados por la emergencia, para lo cual dejarán libre el Servicio, desviarán urgencia a otros hospitales y colaborarán en la evaluación.

Comité de Autoprotección del Centro (CAC)	
Integrantes	Director del hospital, Director económico, Director médico, Director de enfermería, Subdirector económico de infraestructura, Presidente de la comisión de salud y seguridad, Jefe de protección radiológica, Jefe del servicio de ingeniería, Jefe del personal subalterno.
Funciones específicas	<ul style="list-style-type: none"> • Se reunirán como máximo cada 3 meses y coordinarán y asesoran sobre la implantación y mantenimiento del plan de autoprotección.

Ahora que se tienen identificados a los actores, en el procedimiento de respuesta se describen las actividades a realizar en cada una de las fases de la emergencia de tipo Incendio.

ACTOR: PACIENTES Y VISITANTES	
Actividades	
❖	Mantendrá la calma, no gritará, no correrá ni hará ademanes bruscos.
❖	Pulsará el timbre de llamada a enfermera o al Centro de Control, número 8710012, desde cualquier teléfono interior del Hospital.
❖	Dará conocimiento del fuego.
❖	Si se prenden sus ropas se extenderá en el suelo y rodará.
❖	Si hay humo se dirigirá gateando a la salida.
❖	Atenderá y obedecerá sin discutir las instrucciones del personal del hospital.
❖	No utilizará los ascensores.

ACTOR: EQUIPO DE PRIMERA INTERVENCIÓN (EPI)	
Actividades	
❖	Alertará al Centro de Control: <ul style="list-style-type: none"> • Marcará el número 87100 ó 12. • Se identificará • Dará detalle del lugar, naturaleza y tamaño de la Emergencia. • Comprobará que su aviso ha sido recibido.
❖	Utilizará inmediatamente el extintor adecuado, no utilizará manguera de agua.
❖	Desconectará, sólo si sabe, la electricidad e instalaciones especiales.
❖	Esperará la llegada de los Equipos de Emergencia de Segunda Intervención (ESI), a los que les explicará la situación del fuego.

ACTOR: EQUIPO DE SEGUNDA INTERVENCIÓN (ESI)
Actividades
<ul style="list-style-type: none"> ❖ Se dirigirá rápidamente al lugar señalado. ❖ Utilizará inmediatamente los medios de primera intervención. ❖ Intentará extinguir el incendio. ❖ Informará al Jefe de Intervención (JI) y esperará en todo momento sus órdenes. ❖ Colaborará, sólo en caso de que le ordenen, con las ayudas externas en las tareas de extinción.

ACTOR: EQUIPO DE ALARMA Y EVACUACIÓN (EAE)
Actividades
<ul style="list-style-type: none"> ❖ A la señal de alarma acudirá a los lugares de trabajo. ❖ Preparará y ayudará a los enfermos para la posible evacuación. ❖ Colaborará en el mantenimiento del orden. ❖ Hará salir a las visitas que no puedan colaborar en la evacuación. ❖ Cerrará puertas, ventanas, etc. ❖ Desconectará los aparatos que no sean utilizables (televisores, neveras, etc). ❖ Recogerá y preparará el material inflamable para su traslado fuera del recinto afectado. ❖ Recogerá información y documentación médica (historias clínicas, informáticas, etc) a salvar. ❖ Impedirá el uso de ascensores de manera libre. ❖ Esperará ordenes de su Jefe Inmediato.

ACTOR: EQUIPO DE PRIMEROS AUXILIOS (EPA)
Actividades
<ul style="list-style-type: none"> ❖ Se desviarán las urgencias a otros hospitales. ❖ Se prepararán a los enfermos de Urgencias para su traslado fuera del Centro, si fuera necesario. ❖ Se desalojará al personal de la Sala de Espera. ❖ Se prepararán medios de reanimación y cura necesarios. ❖ Se prestará asistencia a los posibles heridos o afectados. ❖ Se prestará asistencia y colaboración en la evacuación de los enfermos del Hospital.

ACTOR: JEFE DE INTERVENCIÓN (JI)
Actividades
<ul style="list-style-type: none"> ❖ Valorará la emergencia, así como su evolución. ❖ Coordinará y dirigirá la lucha contra la emergencia al frente de los equipos de intervención. ❖ Informará en todo momento al Jefe de Emergencia (JE) de la evolución de la emergencia. ❖ Estará continuamente a la espera de las decisiones del Jefe de Emergencia (JE).

ACTOR: JEFE DE EMERGENCIA (JE)
Actividades
<ul style="list-style-type: none"> ❖ Recibirá la información y valorará la necesidad de alarma general. ❖ Ordenará la evacuación. ❖ Ordenará que se ponga en marcha el organigrama de alerta exterior. ❖ Dará órdenes y coordinará a todos los equipos de emergencia. ❖ Recibirá u ordenará que se reciba y se proporcione información a las ayudas externas (Bomberos, Policía Nacional, Policía Local, etc). ❖ Recibirá información de los grupos de alarma, primera intervención y evacuación. ❖ Redactará un informe de las causas, desarrollo y consecuencias de la emergencia.

5.3.2 Diagrama del Procedimiento de Respuesta.

A continuación, para facilitar la comprensión del procedimiento de respuesta (véase 5.3.1) se presenta el Diagrama de Actuación, en el cual se resumen las posibles secuencias de actuación por parte de cada uno de los actores del procedimiento de respuesta del caso.

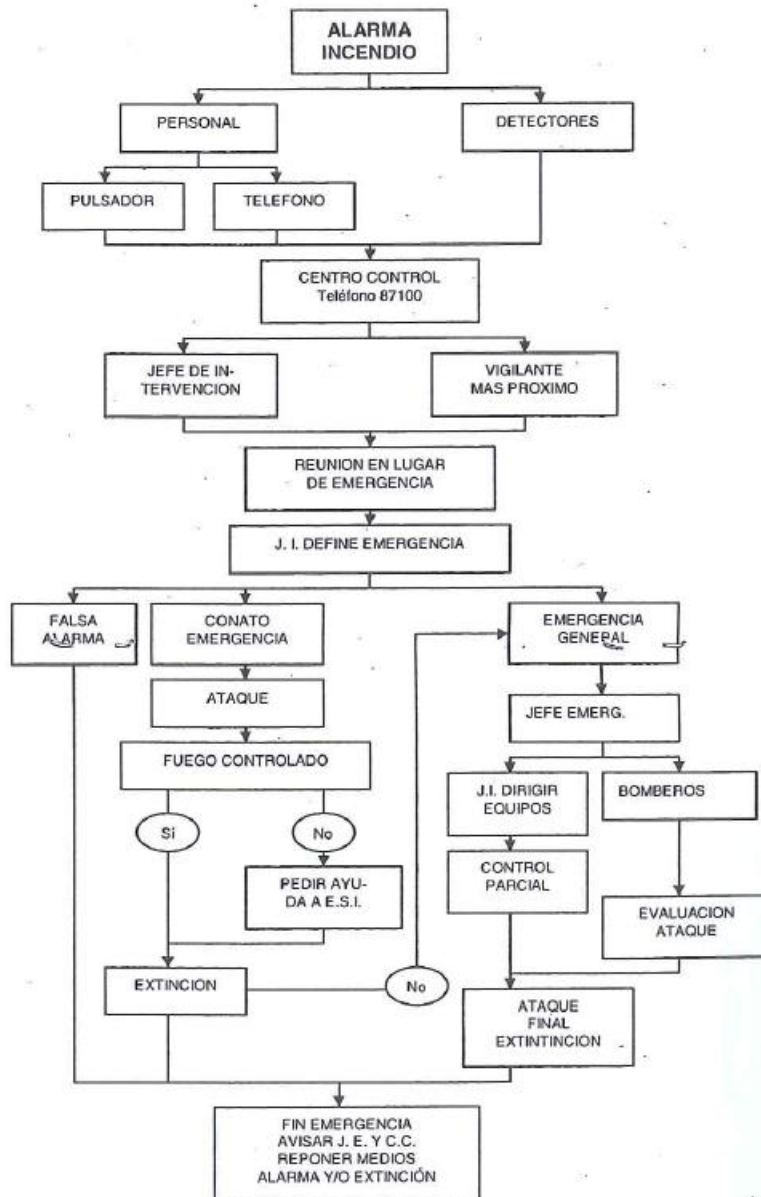


Figura 5-1 Diagrama de Actuación del procedimiento de respuesta.

Como hemos observado, en el Diagrama de Actuación anterior, nos facilita la comprensión del procedimiento de respuesta, sin embargo, existen actividades en el que no se tiene clara la secuencia a realizar y en que momento se debe realizar, esta representación de un procedimiento de respuesta crea ambigüedades y por ello resulta poco confiable la secuencia en el flujo de control al intentar que el procedimiento de respuesta sea ejecutable en tiempo real.

5.3.3 Identificación de los elementos críticos de negocio

Esta sección proporciona la respuesta a la tercera pregunta de investigación.

RQ3: ¿Cómo puede aplicarse el lenguaje CMMN para analizar y diseñar modelos de respuesta partiendo de procedimientos de respuesta de los planes de autoprotección?

Una vez conocido y comprendido el contenido del procedimiento de respuesta, es recomendable identificar los elementos claves de negocio y la relación entre ellos para facilitar el diseño del modelo en lenguaje CMMN.

De acuerdo con el contenido de respuesta y al diagrama de actuación del mismo, se identifica que las etapas claves son la **Detección del incendio**, **Fijar el nivel del Incidente**, Mitigar en el caso de que sea un **conato de emergencia**, resolver si fuera una **emergencia parcial**, en caso de que fuera un nivel de emergencia general prepara la **pre- evacuación** y la **evacuación**. Estos elementos identificados con negritas serán las etapas (Stage) de nuestro modelo.

La **Detección** de un incendio es activado cuando se recibe una llamada dando la alerta, cuando algunos de los detectores se disparan cuando alguien activa manualmente la alarma, estos serán los eventos de entrada de la etapa **Detección**. Dentro de la etapa de detección se identifican las tareas (Task) de Inicio de Incidente, Recibir Aviso, Avisar al vigilante y Evaluar el Aviso. La salida de la etapa deriva en una *Falsa Alarma* o *Incidente Real*; estos últimos elementos identificados con letra cursiva son los hitos con los que el trabajador del caso evalúa el avance del flujo durante la ejecución.

La etapa **Fijar nivel de Incidente** contiene la tarea Evaluar Nivel de Incidente, la salida de esta tarea deriva en los hitos *Conatos de Emergencia*, *Emergencia Parcial* y *Emergencia General*.

La etapa **Mitigar Conato de Emergencia** contiene la tarea Avisar a Jefe de Intervención y Mitigar Conato de Emergencia, dado que el personal de seguridad del Hospital para este tipo de nivel de incendio cuenta con los recursos necesarios para combatirlo, la salida de esta etapa se conecta con el hito *Emergencia Resuelta*.

Para la etapa **Resolver Emergencia Parcial** se tienen las tareas Avisar la Jefe de Intervención, Activar Alarma Interior, Avisar a Equipos de Segunda Intervención, Desalojar visitas de la zona, Evacuar Zona, Mitigar Incendio, Evaluar Situación; así mismo un hito *Emergencia Parcial*. La salida de esta etapa puede ser hacia el hito *Emergencia Resuelta* o *Emergencia General*.

Al declararse una Emergencia General se cuenta con la etapa **Pre-Evacuación** en la cual se tienen las tareas Avisar a servicios Externos de Extinción, Recibir Información Externa, Procedimiento Pre-Evacuación, Decidir Evacuación. La salida puede ser la etapa **Evacuación** o el hito *Emergencia Resuelta*. La etapa **Evacuación** contempla las tareas Procedimiento de Evacuación, Recibir Información Extra, Evaluar Situación. La salida de la etapa es hacia el hito *Emergencia Resuelta*.

5.3.4 Diseño del modelo en Lenguaje CMMN

Con base al análisis anterior, se utiliza Camunda Modeler (ver Apéndice C, sección 1) para diseñar el modelo en Lenguaje CMMN 1.1 del procedimiento de respuesta, es importante señalar que se han tomado en cuenta las limitantes de la notación CMMN 1.1 soportada por el Camunda Modeler, por lo que se ha adecuado el diseño del modelo, quedando de la siguiente forma:

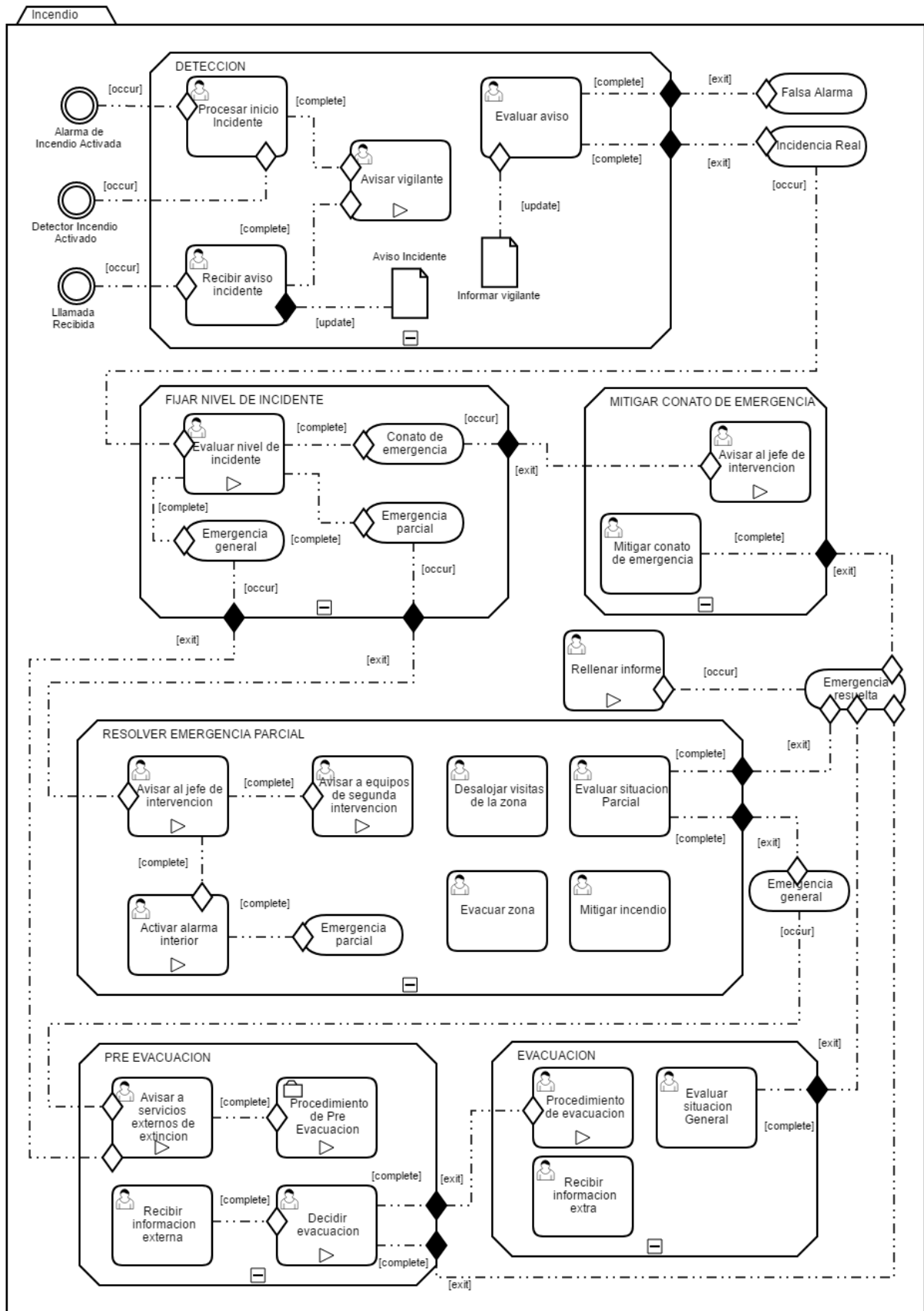


Figura 5-2 Modelo CMMN del Procedimiento de Respuesta de Incendio.

Después de haber conocido el procedimiento de respuesta de un incendio expuesto en el caso de estudio y aplicar los conceptos del lenguaje CMMN, se ha obtenido como resultado un modelo; con ello se ha dado respuesta a la pregunta de investigación RQ3, y se alcanza el objetivo específico SO3 del trabajo de investigación:

SO3: Aplicar el lenguaje CMMN para analizar y diseñar modelos de respuesta flexibles partiendo de procedimientos de respuesta de los planes de autoprotección.

5.3.5 Ejecución del modelo CMMN del Procedimiento de Respuesta

En esta sección se dará respuesta a la cuarta pregunta de investigación:

RQ4: ¿Cómo se puede ejecutar modelos basados en lenguaje CMMN que sean adaptables al entorno y las decisiones tomadas por el trabajador del caso?

Una vez diseñado el modelo se debe tener la plataforma Camunda BPM instalada y configurada con Eclipse Mavel para poder ejecutar el modelo. En el Apéndice C se presenta una guía que indica paso a paso la manera de instalar la plataforma de Camunda BPM y la configuración del entorno para ejecutar un modelo en lenguaje CMMN.

En la siguiente Tabla 5-1 se definen las propiedades de los elementos del modelo Incendio, los cuales fueron ingresados en el Camunda Modeler.

Elemento	Tipo	Propiedades
Procesar inicio Incidente	Tarea	EntryCriterion_04khtcu: If part Condition = \${alarmaIncendio} EntryCriterion_1d6eh00: If part Condition = \${detectorIncendio}
Recibir aviso incidente	Tarea	EntryCriterion_11dqre8: If part Condition = \${llamadaRecibida}
Avisar vigilante	Tarea	Manual Activation Rules = EntryCriterion_01gwz4l: If part Condition = \${inicioIncidente} EntryCriterion_12rtqci: If part Condition = \${avisoIncidente}
DETECCIÓN	Etapa	EntryCriterion_1jshiul: If part Condition = \${!incidenteReal} EntryCriterion_0y6o2dj: If part Condition = \${incidenteReal}
Falsa Alarma	Hito	EntryCriterion_1ragos3: If part Condition = \${!incidenteReal}
Incidencia Real	Hito	EntryCriterion_03cxnw0: If part Condition = \${incidenteReal}
FIJAR NIVEL DE INCIDENTE	Etapa	EntryCriterion_1dew2c4: If part Condition = \${conatoEmergencia} EntryCriterion_1pj67xv: If part Condition = \${emergenciaParcial} EntryCriterion_1erk4hx: If part Condition = \${emergenciaGeneral}
Evaluar nivel de incidente	Tarea	Manual Activation Rules = \${false} EntryCriterion_1i2stg0: If part Condition = \${incidenteReal}
Emergencia general	Hito	EntryCriterion_1hvx2n8: If part Condition = \${emergenciaGeneral}
Conato de emergencia	Hito	EntryCriterion_1i4rxrw: If part Condition = \${conatoEmergencia}
Emergencia parcial	Hito	EntryCriterion_0tx87fz: If part Condition = \${emergenciaParcial}
MITIGAR CONATO DE EMERGENCIA	Etapa	EntryCriterion_0qmdf31: If part Condition = \${conatoMitigado}
Avisar al jefe de intervención	Tarea	Manual Activation Rules = EntryCriterion_06dh2ns: If part Condition = \${conatoEmergencia}
RESOLVER EMERGENCIA PARCIAL	Etapa	EntryCriterion_1khvo5t: If part Condition = \${emergenciaParcialResuelta} EntryCriterion_1ywe5b0: If part Condition = \${!emergenciaParcialResuelta}
Avisar al jefe de intervención	Tarea	Manual Activation Rules = \${false}
Activar alarma interior	Tarea	Manual Activation Rules = \${false} EntryCriterion_08w7ub7: If part Condition = \${alarmaInterior}

Elemento	Tipo	Propiedades
Emergencia Parcial	Hito	EntryCriterion_134okme: If part Condition = \${emergenciaParcialOK}
Avisar a equipos de segunda intervención	Tarea	Manual Activation Rules = EntryCriterion_15lv65d: If part Condition = \${avisarESI}
Emergencia general	Tarea	EntryCriterion_0eqmvia: \${!emergenciaParcialResuelta}
PRE EVACUACIÓN	Etapa	EntryCriterion_1drzyc3: If part Condition = \${siEvacuacion} EntryCriterion_144kdk5: If part Condition = \${!siEvacuacion}
Avisar a servicios externos de extinción	Tarea	Manual Activation Rules = \${false} EntryCriterion_0kh0b9e: If part Condition = \${!emergenciaParcialResuelta} EntryCriterion_16fkn5l: If part Condition = \${emergenciaGeneral}
Procedimiento de Pre Evacuación	Tarea	Manual Activation Rules = EntryCriterion_17c769s: If part Condition = \${preEvacuacion}
Decidir evacuación	Tarea	Manual Activation Rules = \${false} EntryCriterion_1g1gkub: If part Condition = \${evaluarEvacuacion}
EVACUACIÓN	Etapa	EntryCriterion_09kiftb: If part Condition = \${emergenciaGeneralResuelta}
Procedimiento de evacuación	Tarea	Manual Activation Rules = EntryCriterion_1anbrp7: If part Condition = \${siEvacuacion}
Emergencia resuelta	Hito	EntryCriterion_058qxtg: If part Condition = \${emergenciaParcialResuelta} EntryCriterion_1n10tg0: If part Condition = \${emergenciaGeneralResuelta} EntryCriterion_09t222k: If part Condition = \${!siEvacuacion} EntryCriterion_1p2ix0f: If part Condition = \${conatoMitigado}
Rellenar informe	Tarea	Manual Activation Rules = \${false}

Tabla 5-1 Propiedades de los elementos del modelo Incendio.

A continuación, se carga el siguiente contenido en el archivo IncendioApplication.java del proyecto Maven en la ruta src/main/java dentro del paquete:
org.camunda.bpm.getstarted.cmmn.incendio

```

package org.camunda.bpm.getstarted.cmmn.incendio;
import org.camunda.bpm.application.PostDeploy;
import org.camunda.bpm.application.ProcessApplication;
import org.camunda.bpm.application.impl.ServletProcessApplication;
import org.camunda.bpm.engine.CaseService;
import org.camunda.bpm.engine.ProcessEngine;
import org.camunda.bpm.engine.variable.Variables;

@ProcessApplication("Incendio App CMMN")
public class IncendioApplication extends ServletProcessApplication {

    @PostDeploy
    public void startCaseInstance(ProcessEngine processEngine) {
        CaseService caseService = processEngine.getCaseService();
        caseService.createCaseInstanceByKey("incendio",
            Variables.createVariables()
                .putValue("alarmaIncendio", Variables.booleanValue(true))
                .putValue("detectorIncendio", Variables.booleanValue(false))
                .putValue("llamadaRecibida", Variables.booleanValue(false))
                .putValue("inicioIncidente", Variables.booleanValue(false))
        );
    }
}

```

```

        .putValue("avisoincidente", Variables.booleanValue(false))
        .putValue("incidentereal", Variables.booleanValue(false))
        .putValue("emergenciageneral", Variables.booleanValue(false))
        .putValue("conatoemergencia", Variables.booleanValue(false))
        .putValue("emergenciaparcial", Variables.booleanValue(false))
        .putValue("conatomitigado", Variables.booleanValue(false))
        .putValue("alarmainterior", Variables.booleanValue(false))
        .putValue("avisarESI", Variables.booleanValue(false))
        .putValue("emergenciaparcialOK", Variables.booleanValue(false))
        .putValue("emergenciaparcialresuelta", Variables.booleanValue(false))
        .putValue("emergenciageneralresuelta", Variables.booleanValue(false))
        .putValue("preevacuacion", Variables.booleanValue(false))
        .putValue("evaluarEvacuacion", Variables.booleanValue(false))
        .putValue("siEvacuacion", Variables.booleanValue(false))
    );
}
}

```

Debido a que actualmente el núcleo del motor de Camunda no soporta la notación de eventos del Listener (ver figura 4-3), la variable *alarmaIncendio* tipo booleana del archivo *IncendioApplication.java* la inicializamos a *true* para que se active el inicio de la tarea *Procesar Inicio Incidente*.

Así mismo, como se indica en el capítulo 4 de este trabajo, se agrega el archivo *LyfecycleListener.java* en la ruta *src/main/java* dentro del paquete: *org.camunda.bpm.getstarted.cmmn.incendio*

Al abrir la plataforma Camunda BPM, ingresamos a la aplicación del TaskList y observamos el listado de tareas del modelo de Incendio en el filtro "My Task". De acuerdo con el modelo y la limitante de que el motor no soporta las tareas discretas, se ha optado por instanciar las tareas desde el inicio, sin que ello signifique su ejecución, únicamente se permite que el trabajador del caso disponga de ellas para ejecutarlas en el momento en que lo requiera. Las tareas son: Evaluar situación general, Recibir información extra, Recibir información externa, Mitigar incendio, Evacuar zona, Desalojar visitas de la zona, Evaluar situación parcial, Mitigar conato de emergencia, Evaluar aviso.

Se elige la tarea **Procesar inicio Incidente** la cual se ha instanciado al colocar en *True* la variable *alarmaIncendio* en el archivo *IncendioApplication.java*. Posteriormente, se agrega la variable *inicioIncidente* de tipo *Boolean* con el valor *True*, tal como se muestra en la figura 5-3; para finalizar se da clic en el botón *Complete*, al realizar esto el motor instancia la tarea **Avisar al vigilante**.

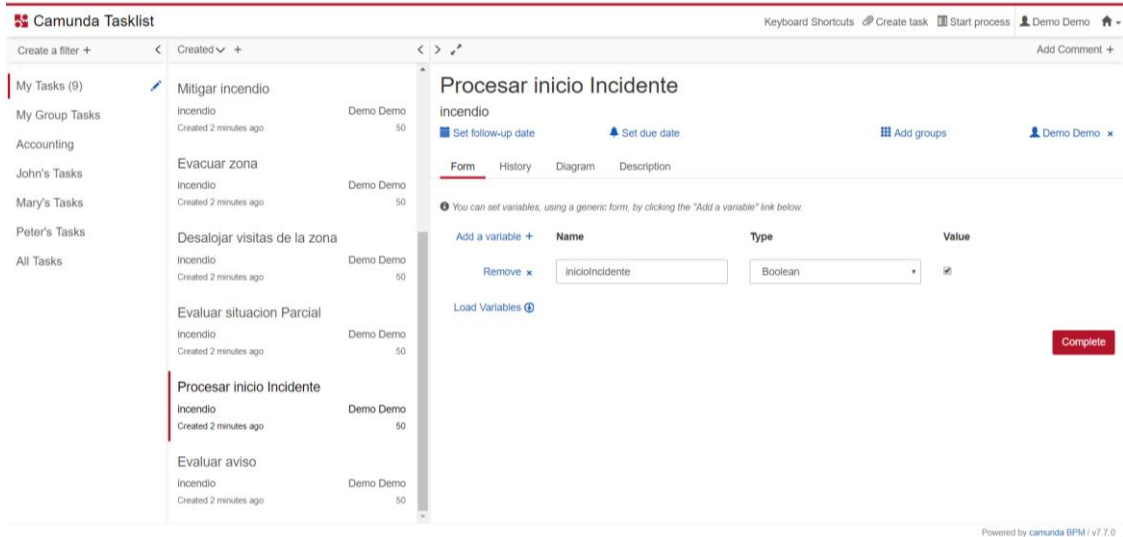


Figura 5-3 Tarea Procesar inicio Incidente.

A continuación, se va a representar las diferentes situaciones y para ello se va a elegir distintos flujos de ejecución. El primer flujo elegido es denominado “Falsa Alarma”.

5.3.5.1 Flujo de Ejecución: Falsa Alarma

En el flujo de ejecución “Falsa Alarma” se busca que el hito alcanzado sea el denominado *Falsa Alarma*. En la siguiente Tabla 5-2 se muestra las variables que se han de agregar en cada uno de los elementos para recorrer el flujo de ejecución (ver figura 5-2). Recordar que en cada tarea dar clic al botón *Complete* para que el motor procese la información y continúe con el flujo de ejecución.

Tarea	Nombre	Tipo	Valor
Evaluar aviso	incidenteReal	Boolean	False

Tabla 5-2 Valores de las variables del flujo de trabajo Falsa Alarma.

En la siguiente figura se resalta en naranja los elementos que fueron ejecutados en el flujo.

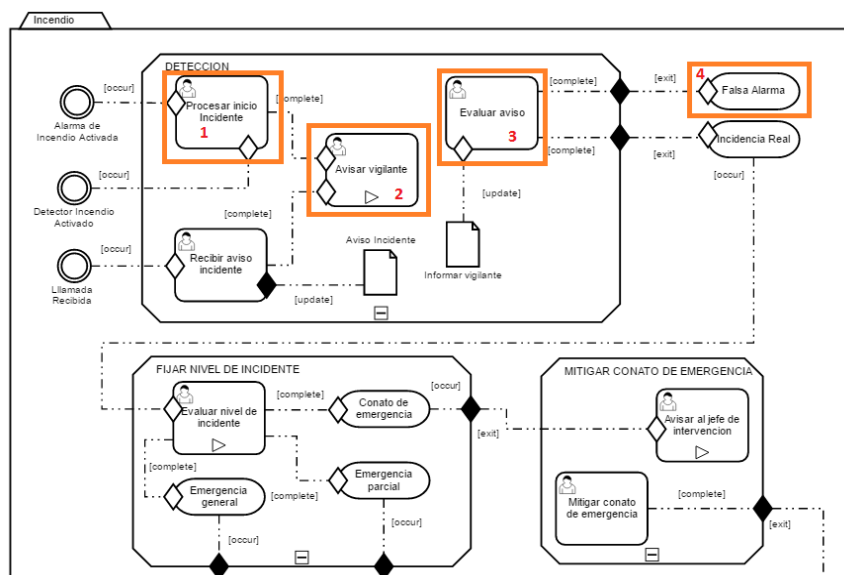


Figura 5-4 Flujo de ejecución de Falsa Alarma.

5.3.5.2 Flujo de Ejecución: Conato de Incendio

En el flujo de ejecución “Conato de Incendio” se tiene como objetivo alcanzar el hito *emergencia resuelta* y posteriormente ejecutar la tarea *Rellenar Informe*. En la siguiente Tabla 5-3 se detallan las variables que se han de ingresar en cada una de las tareas para reproducir el flujo de ejecución.

Tarea	Nombre	Tipo	Valor
Evaluar aviso	incidenteReal	Boolean	True
Evaluar nivel de Incidente	conatoEmergencia	Boolean	True
Mitigar conato de emergencia	conatoMitigado	Boolean	True

Tabla 5-3 Valores de las variables del flujo de trabajo Conato de Incendio.

En la siguiente figura 5-5 se resalta en naranja los elementos que fueron ejecutados.

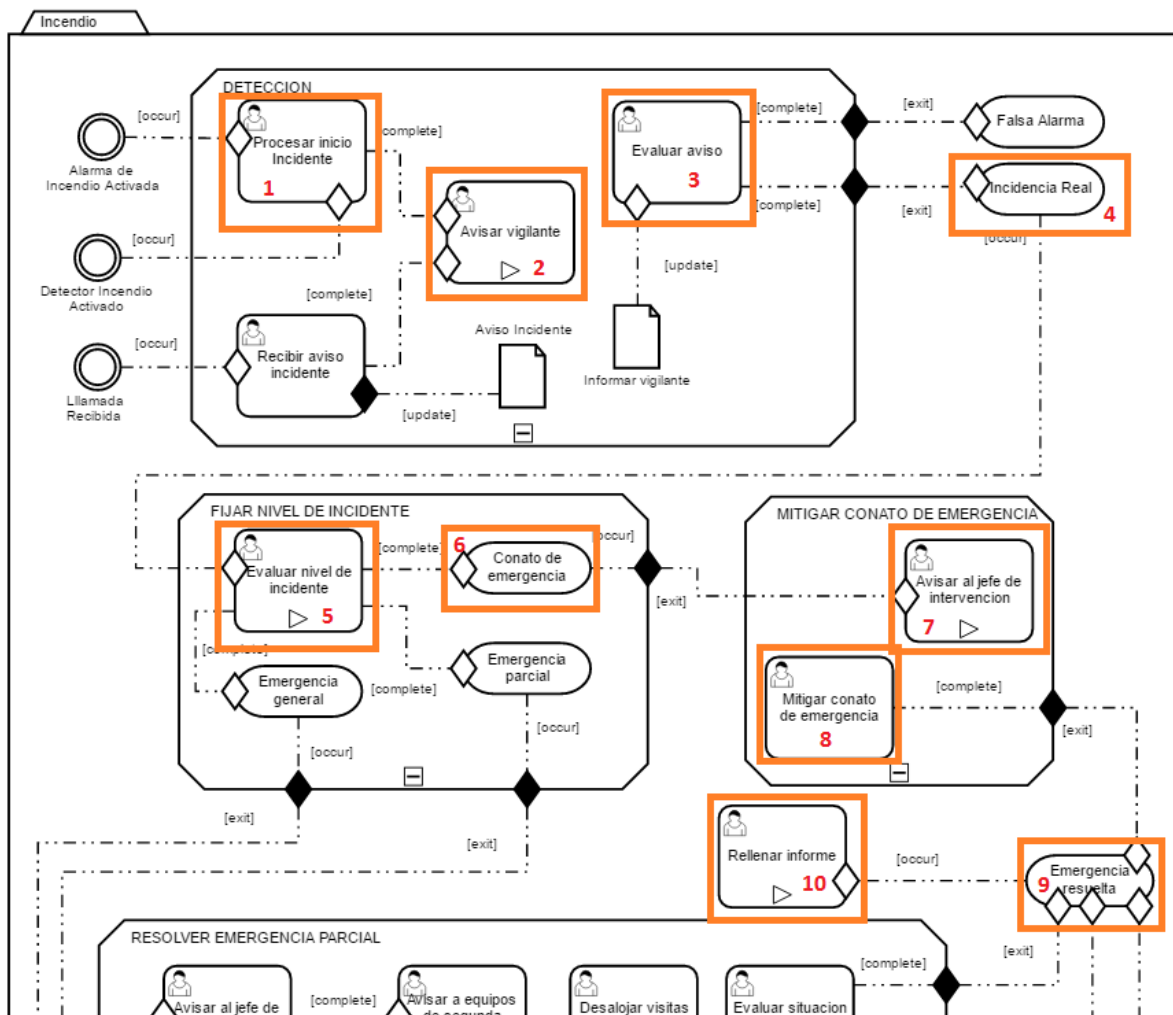


Figura 5-5 Flujo de ejecución de Conato de Emergencia.

5.3.5.3 Flujo de Ejecución: Emergencia Parcial

A continuación, se presenta el flujo de ejecución “Emergencia Parcial” el cual tiene como objetivo alcanzar el hito *emergencia resuelta* y posteriormente ejecutar en automático la tarea *Rellenar Informe*. Así mismo, se va a elegir ejecutar las tareas discretas: *Desalojar visitas de la zona* y *Mitigar Incendio*. En la siguiente Tabla 5-4 se explican las variables que se han de ingresar en cada una de las tareas para reproducir el flujo de ejecución.

Tarea	Nombre	Tipo	Valor
Evaluar aviso	incidenteReal	Boolean	True
Evaluar nivel de Incidente	emergenciaParcial	Boolean	True
Avisar al jefe de intervención	avisarESI	Boolean	True
Desalojar visitas de la zona			
Mitigar incendio			
Evaluar situación Parcial	emergenciaParcialResuelta	Boolean	True

Tabla 5-4 Valores de las variables del flujo de trabajo Emergencia Parcial.

En la figura 5-6 se resalta los elementos en color naranja que fueron ejecutados.

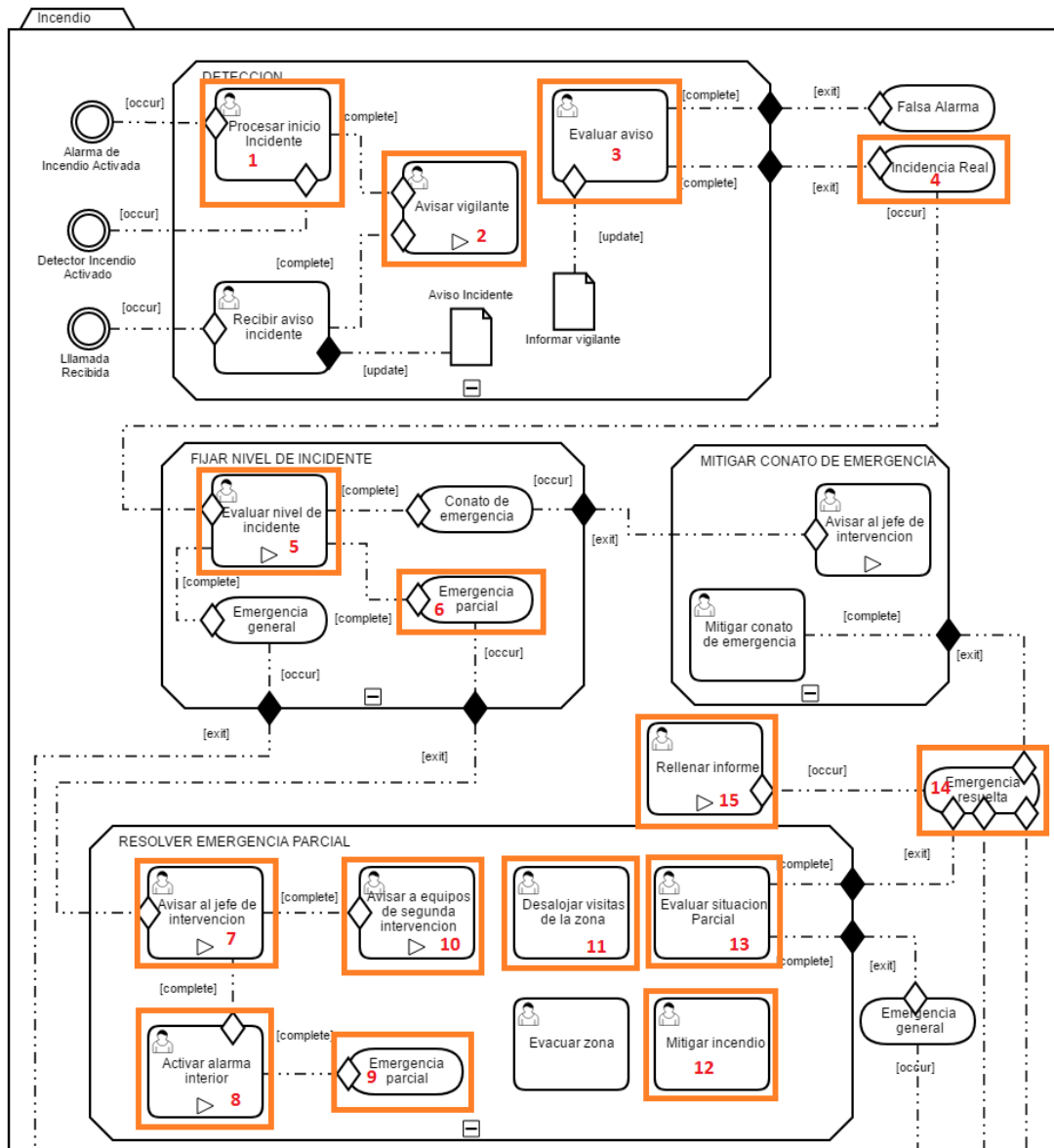


Figura 5-6 Flujo de ejecución de Emergencia Parcial.

5.3.5.4 Flujo de Ejecución: Emergencia General

Por último, se detalla el flujo de ejecución “Emergencia General” en el cual se alcanza el hito *emergencia resuelta* y posteriormente se ejecuta automáticamente la tarea *Rellenar Informe*. En la Tabla 5-5 se enlistan las variables que se agregan en cada una de las tareas para ejecutar el flujo.

Tarea	Nombre	Tipo	Valor
Evaluar aviso	incidenteReal	Boolean	True
Evaluar nivel de Incidente	emergenciaGeneral	Boolean	True
Avisar a servicios externos de extinción	preEvacuacion	Boolean	True
Recibir información externa	evaluarEvacuacion	Boolean	True
Decidir evacuación	siEvacuacion	Boolean	True
Evaluar situación General	emergenciaGeneralResuelta	Boolean	True

Tabla 5-5 Valores de las variables del flujo de trabajo Emergencia General.

En la siguiente figura 5-7, nuevamente se señalan en color naranja los elementos fueron ejecutados durante el flujo.

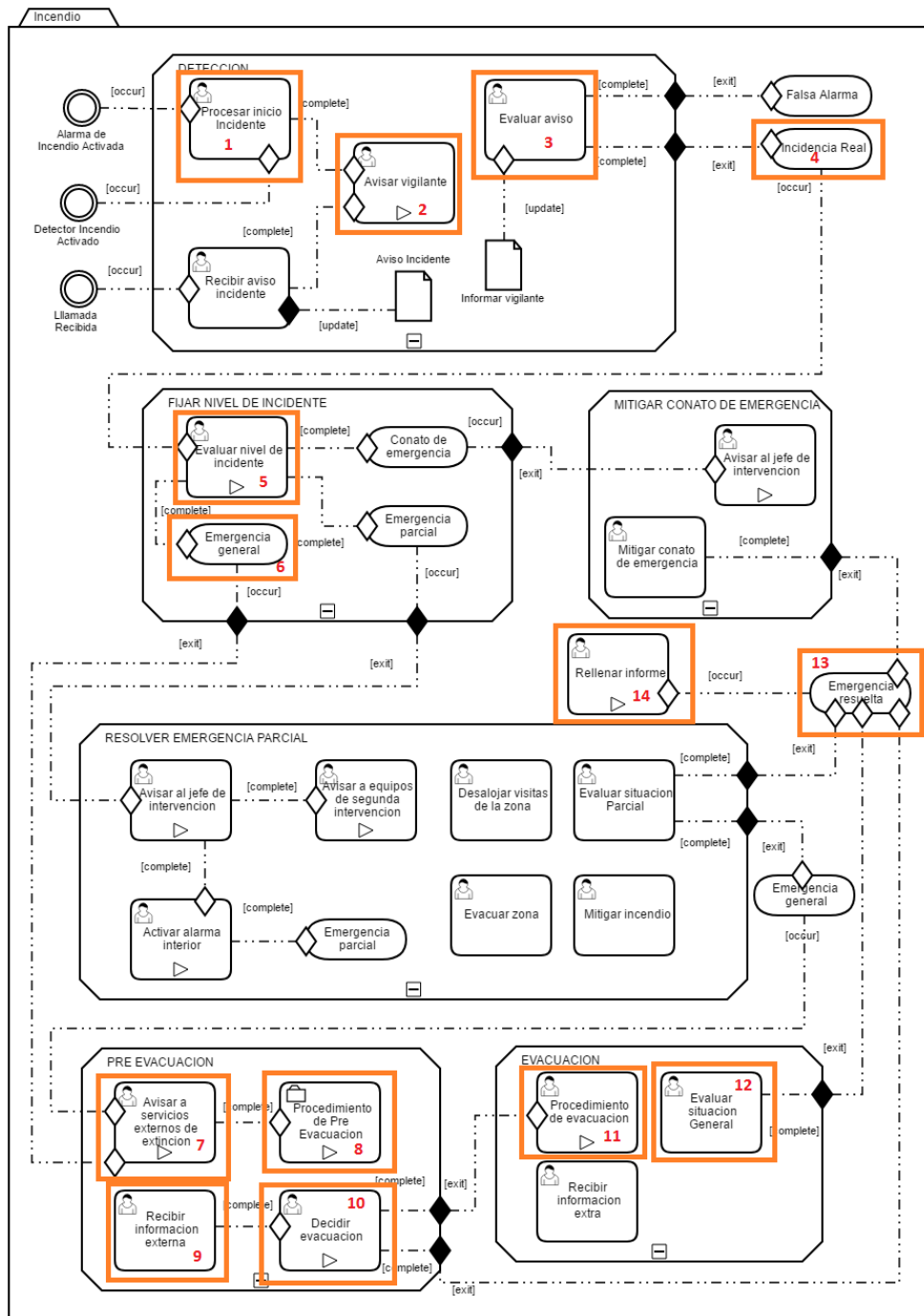


Figura 5-7 Flujo de ejecución de Emergencia General.

Utilizando los posibles valores cambiantes de entorno, se ha ejecutado por distintos flujos el modelo del procedimiento de respuesta de Incendio, se han ilustrado los análisis de cada una de las ejecuciones; con ello se ha dado respuesta a la pregunta de investigación RQ4, y se alcanza el objetivo específico SO4 del trabajo de investigación:

SO4: Ejecutar un modelo CMMN y que sea adaptable al entorno y a las decisiones elegidas por el trabajador del caso.

5.3.6 Observaciones

Durante el caso de estudio se hicieron las siguientes observaciones con respecto al modelado, tanto en tiempo de diseño como en tiempo de planificación y ejecución:

1. Reglas de Sintaxis.

Las reglas de sintaxis para los elementos de diseño CMMN no son descritas de forma clara en la especificación del estándar. Las reglas de sintaxis y las restricciones se distribuyen entre las especificaciones, lo anterior deriva en menos visibilidad y fragmentación de las reglas. No existe una gestión de reglas central basada en un lenguaje de restricciones detallado.

2. Notación.

Durante el diseño de la aplicación práctica como parte del experimento, se ha omitido la representación de las tareas a sus respectivos roles usando un enfoque de diseño de tipo carril de natación en CMMN, ya que no existe la representación visual de roles. Aunque la aplicación práctica como parte del experimento está limitada a sólo unos pocos roles, suele existir confusión al momento de interpretar el modelo y asignar la ejecución de tareas.

3. Limitantes del modelador de Camunda.

Actualmente la versión web del modelador de Camunda no cuenta con la opción del panel de propiedades, por lo tanto, el diseño del modelo si se realiza en esa versión queda incompleto, además cuando el diseño del modelo es robusto y se realiza en el modelador del entorno web, existe incompatibilidad al momento de abrir o importar el modelo en formato XML. En la versión web no se pueden duplicar (copiar y pegar) esto alarga el tiempo de diseño.

4. Validación de un modelo.

La validación de los modelos CMMN se realiza de manera manual, no existe una herramienta como parte de la plataforma Camunda BPM que automatice esta labor. Por lo tanto, el trabajador del caso debe realizar varios flujos de trabajo para validar el modelo y con ello buscar mejorar el diseño del mismo. Así mismo si existen inconsistencias en el diseño del modelo con respecto a las reglas del estándar, el modelador no las valida y falla al intentar realizar el despliegue del modelo en la plataforma Camunda BPM. Además, uno de los beneficios que aporta el contar con los modelos de los procedimientos de respuesta es buscar mejorar las acciones de los mismos procedimientos de respuesta, sin embargo, estas mejoras el trabajador del caso las puede identificar de forma manual con forme se van realizando distintos flujos de ejecución y se detectan situaciones en las que el modelo no proporciona una respuesta satisfactoria.

5. Manejo de Errores.

Si existe problemas al momento de desplegar un modelo en la plataforma Camunda BPM, los errores son presentados mediante un archivo de texto log, sin embargo, la información presentada no es detallada y es difícil identificar las inconsistencias para dar solución a las mismas.

6. Interfaz de ejecución del modelo.

La interfaz que se utiliza en la plataforma Camunda BPM es la aplicación TaskList, la cual tiene la ventaja de ser en ambiente web. Esta aplicación permite al trabajador del caso seleccionar las tareas que se irán ejecutando con base a los criterios de los sentries, reglas de negocio o instanciación de tareas, sin embargo, el flujo de trabajo se va siguiendo a través de un listado y no mediante la representación gráfica del modelo CMMN, como ocurre con los modelos BPMN. Así mismo cada vez que se ejecuta un caso es necesario volver a definir e ingresar los valores de las variables de los sentries. A pesar de que el núcleo del motor de Camunda puede ser consumido a través de la API REST o la API Java, la interfaz en la que interactúa el trabajador del caso puede ser personalizable únicamente mediante páginas web. Sin embargo, la observación más importante es que la versión actual no implementa el uso de toda la notación del lenguaje CMMN 1.1, es por ello que es necesario adaptar el modelo en tiempo de diseño para posteriormente ser ejecutado.

7. Limitada interoperabilidad de la plataforma Camunda BPM.

Si bien el núcleo del motor de Camunda puede ser consumido por API REST o API Java, la plataforma de Camunda BPM no facilita la interoperabilidad con otros entornos de trabajo que tengan que ser utilizados para realización de tareas y con ello mejorar la interacción con los usuarios finales.

6. Propuesta del Marco de trabajo

Con base a las observaciones realizadas como producto del modelado y ejecución de un modelo a través de la plataforma Camunda BPM en este capítulo se propone el diseño de dos aplicaciones personalizadas que mejoren por un lado el diseño del modelado de los casos de estudio, denominada “Modelador CMMN” y por otro la interoperabilidad al momento de ejecutar los modelos permitiendo interactuar con diferentes entornos de trabajo, denominada “Ejecutor de Modelos”; lo anterior aprovechando que el motor del núcleo de Camunda permite conectividad API REST y que su arquitectura contempla la incorporación de aplicaciones personalizadas.

En el presente capítulo se realiza una propuesta del marco de trabajo integrado por las dos aplicaciones de interfaz web que sirvan al trabajador del caso para modelar y ejecutar modelos, y con ello atender la quinta pregunta del presente trabajo de investigación.

RQ5: ¿Cuál sería el marco de trabajo a utilizar cuya interfaz facilite al trabajador del caso el diseño y ejecución de procedimientos de respuesta en modelos basados en lenguaje CMMN?

6.1 Modelador CMMN

Si bien el modelador de Camunda tiene algunas limitantes como son el rol de usuarios y la falta de validación del modelo, también es de resaltar que para la edición de modelos proporciona una interfaz eficiente. Sin embargo, se propone la incorporación de un panel en donde el usuario con rol diseñador pueda cargar el texto del procedimiento de respuesta y guardar en un repositorio el diseño del modelo. A continuación, se detalla la herramienta propuesta denominada “Modelador CMMN”.

6.1.1 Arquitectura de Modelador CMMN

Se propone una herramienta con interfaz web que brinde servicio para el diseño de modelos en lenguaje CMMN 1.1. En la figura 6-1 se presenta la arquitectura de la herramienta.

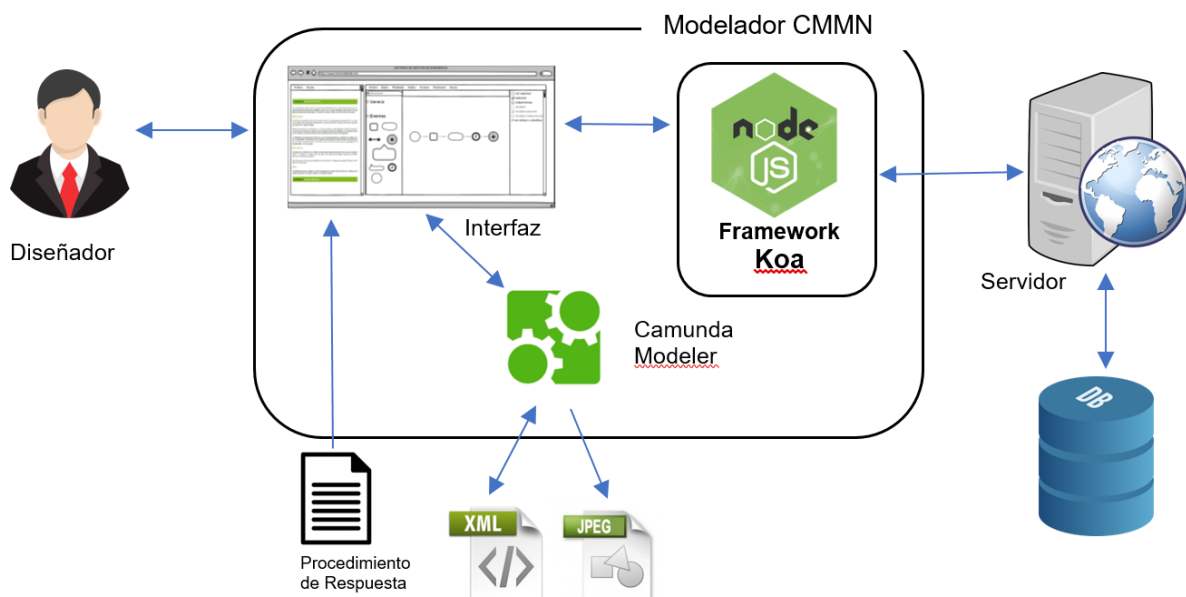


Figura 6-1 Arquitectura de Modelador CMMN.

El usuario con rol diseñador podrá acceder a la herramienta mediante la identificación de sus credenciales, la herramienta incluirá el modelador de Camunda y adicionará un panel donde el usuario podrá cargar el procedimiento de respuesta en formato texto, así mismo se permite al usuario almacenar en un repositorio de PostgreSQL los modelos y sus propiedades.

La arquitectura de esta herramienta estará soportada por el Framework Koa de Node.js que brinda mediante el patrón de diseño Modelo–Vista–Controlador (Model–View–Controller, MVC) la plataforma necesaria para gestionar servicios, interfaz, código fuente, librerías y acceso a base de datos PostgreSQL alojada en un servidor web.

6.1.2 Análisis y diseño conceptual del Modelador CMMN

En esta sección se describe el análisis y diseño conceptual del modelador CMMN. En primer lugar, se establecen las características funcionales y no funcionales que se eligen para la realización del modelador CMMN, a fin de cubrir con todos los requerimientos necesarios para poder dar solución a la funcionalidad requerida y cumplir con los objetivos planteados en la herramienta propuesta. Posteriormente el diagrama de contexto representa a los actores y su interacción con el sistema. La funcionalidad se muestra con la utilización de casos de uso y su respectiva especificación.

6.1.2.1 Requisitos del Modelador CMMN

Esta sección enlista los requerimientos funcionales que posteriormente serán representados con mayor nivel de detalle en los casos de uso, así mismo, los requerimientos no funcionales orientados a aspectos técnicos fundamentados en la operatividad del modelador CMMN.

- Requisitos Funcionales

- El modelador CMMN permite la gestión de usuarios por medio de un rol administrativo con privilegios globales.
- El modelador CMMN permite el diseño de modelo CMMN basado en la versión actualizada de Camunda Modeler.
- El modelador permite al usuario almacenar y recuperar en un repositorio los modelos que diseñe junto con sus propiedades.
- El modelador CMMN permite la exportación de los modelos en formato XML o en imagen.
- El modelador CMMN permite cargar el procedimiento de respuesta en un panel emergente para facilitar el modelado.

- Requisitos No Funcionales

- Almacenamiento en un gestor de base de datos relacional como lo es PostgreSQL.
- Autenticación de usuario para el acceso a la herramienta web.
- Interfaz de la herramienta web desarrollada en lenguaje Java Script.
- Se utiliza el Framework Koa, el cual implementa el patrón de diseño MVC.
- La interfaz de la herramienta web estará desarrollada mediante scripts de JQuery y hojas de estilo CSS que estarán insertados en páginas HTML.
- Visualización de la arquitectura web en cualquier navegador web de reciente actualización.

6.1.2.2 Diagrama del contexto

En esta sección se presentan los actores y su interacción con el modelador CMMN por medio del diagrama de contexto de la Figura 6-2. Principalmente se trabaja con dos tipos de usuarios cuya función se detalla en las Tablas 6-1 y 6-2.



Figura 6-2 Diagrama del contexto del Modelador CMMN.

Nombre	Administrador del Modelador CMMN
Descripción	Representa al usuario administrador, con roles y privilegios globales sobre el modelador CMMN. Encargado de la gestión de usuarios.

Tabla 6-1 Usuario Administrador del Modelador CMMN.

Nombre	Usuario Diseñador
Descripción	Representa al usuario diseñador experto en el modelado CMMN. Encargado de gestionar los modelos que realice con en los cuales puede cargar en formato texto los procedimientos de respuesta.

Tabla 6-2 Usuario Diseñador del Modelador CMMN.

6.1.2.3 Diagrama estructurado de casos de uso

La representación general de la funcionalidad del modelador CMMN está representada en el diagrama general de casos de uso de la Figura 6-3, en el cual se enseña la participación de cada actor definido en la sección anterior, con los diferentes casos generales de gestión de modelos. El administrador gestiona a los usuarios de rol diseñador. Así mismo, el usuario diseñador se encarga de crear, recuperar, actualizar o eliminar (CRUD) los modelos que sean de su propiedad, además se le permite cargar en formato tipo texto los procedimientos de respuesta, y exportar o importar modelos en formato XML o exportar en imagen.

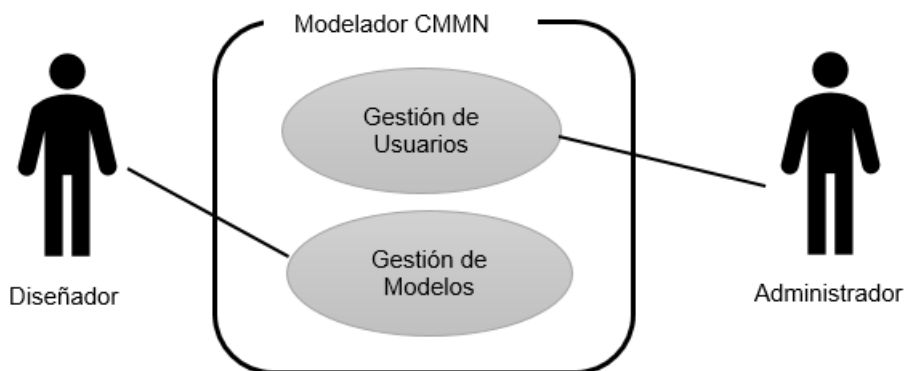


Figura 6-3 Diagrama estructurado de casos de uso.

Una vez que han sido definidos de manera general los casos de uso que integran el modelador CMMN, se procede con el detalle y especificación de cada uno de los componentes y sus relaciones con mayor nivel de detalle. Para una mejor explicación se divide en dos subsistemas (administrador y diseñador) y dentro de cada uno se describe la funcionalidad agrupada por características o gestores según su comportamiento dentro del modelador CMMN.

- Subsistema Administrador

Este subsistema describe las características en las que participa el usuario de tipo administrador y que involucra la gestión de los usuarios de tipo diseñador y de tipo experto, este último definido en la herramienta Ejecutor de modelos (ver sección 6.2.2.2). La funcionalidad está representada por 5 casos de uso (crear, modificar, eliminar, listar y búsqueda) que se describen a continuación en la figura 6-4.

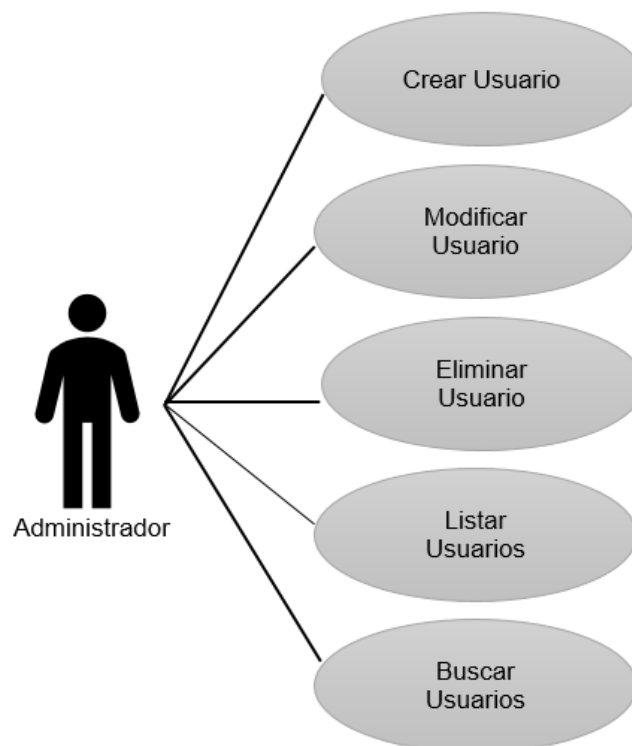


Figura 6-4 Diagrama de casos de uso de Gestión de usuarios.

Referencia	CU01
Nombre	Crear usuario.
Descripción	Gestionar la creación de un nuevo usuario con rol diseñador o experto.
Actores	Administrador, diseñador, experto
Relaciones	
Precondición	Que el usuario no esté registrado.
Postcondición	El usuario quedará registrado con estado activo.
Flujo básico	1.- El administrador asigna un nombre de usuario y contraseña temporal. 2.- Asigna el rol de diseñador o experto. 3.- Almacena el registro en la base de datos. 4.- Envía correo electrónico la notificación al usuario.
Flujo alternativo	3 o 4.- Si existe una excepción con el registro del usuario o alguno de los datos, se notifica con un mensaje de error.

Tabla 6-3 Caso de uso de crear usuario.

Referencia	CU02
Nombre	Modificar usuario.
Descripción	Gestionar el proceso de modificar los datos de un usuario existente con rol diseñador o experto.
Actores	Administrador, diseñador, experto
Relaciones	
Precondición	El usuario diseñador debe estar registrado.
Postcondición	Las modificaciones de los datos del usuario quedarán almacenadas.
Flujo básico	1.- El administrador busca e identifica al usuario requerido. 2.- Realiza las modificaciones en los campos correspondientes. 3.- Actualiza y almacena la información.
Flujo alternativo	3.- Si existe una excepción con la modificación del usuario, se notifica con un mensaje de error.

Tabla 6-4 Caso de uso de modificar usuario.

Referencia	CU03
Nombre	Eliminar usuario.
Descripción	Gestionar el proceso de eliminar un usuario existente con rol diseñador o experto.
Actores	Administrador, diseñador, experto
Relaciones	
Precondición	El usuario diseñador debe estar registrado.
Postcondición	El usuario quedará inactivo.
Flujo básico	1.- El administrador busca e identifica al usuario requerido. 2.- Asignar el estado de inactivo al usuario. 3.- Actualiza y almacena la información.
Flujo alternativo	3.- Si existe una excepción con la eliminación del usuario, se notifica con un mensaje de error.

Tabla 6-5 Caso de uso de eliminar usuario.

Referencia	CU04
Nombre	Listar usuarios.
Descripción	Obtener una lista de usuarios de tipo diseñador o tipo experto.
Actores	Administrador
Relaciones	
Precondición	Deben existir usuarios tipo diseñador o tipo experto registrados.
Postcondición	
Flujo básico	1.- El administrador ingresa al sistema en el área correspondiente. 2.- Accede a formulario de consultas. 3.- Elige los criterios de búsqueda de los usuarios.
Flujo alternativo	3.- Si existe una excepción con el listado de usuarios, se notifica con un mensaje de error.

Tabla 6-6 Caso de uso de listar usuarios.

Referencia	CU05
Nombre	Búsqueda de usuario.
Descripción	A través de una búsqueda obtener los datos de un usuario elegido.
Actores	Administrador
Relaciones	
Precondición	Deben existir usuarios tipo diseñador o experto registrados.
Postcondición	
Flujo básico	1.- El administrador ingresa al sistema en el área correspondiente. 2.- Accede a formulario de consultas. 3.- Elige los criterios de búsqueda del usuario y selecciona. 4.- Visualiza los datos del usuario elegido.
Flujo alternativo	3.- Si existe una excepción con la búsqueda de usuario, se notifica con un mensaje de error.

Tabla 6-7 Caso de uso de buscar usuario.

- Subsistema Diseñador

En el presente subsistema se detallan las características en las que participa el usuario de tipo diseñador y que involucra la gestión de los modelos en lenguaje CMMN. La funcionalidad está representada por 8 casos de uso (crear, editar, eliminar, listar modelos, búsqueda, exportar modelo, importar modelo, leer documento,) los cuales se describen en la figura 6-5 de casos de uso.

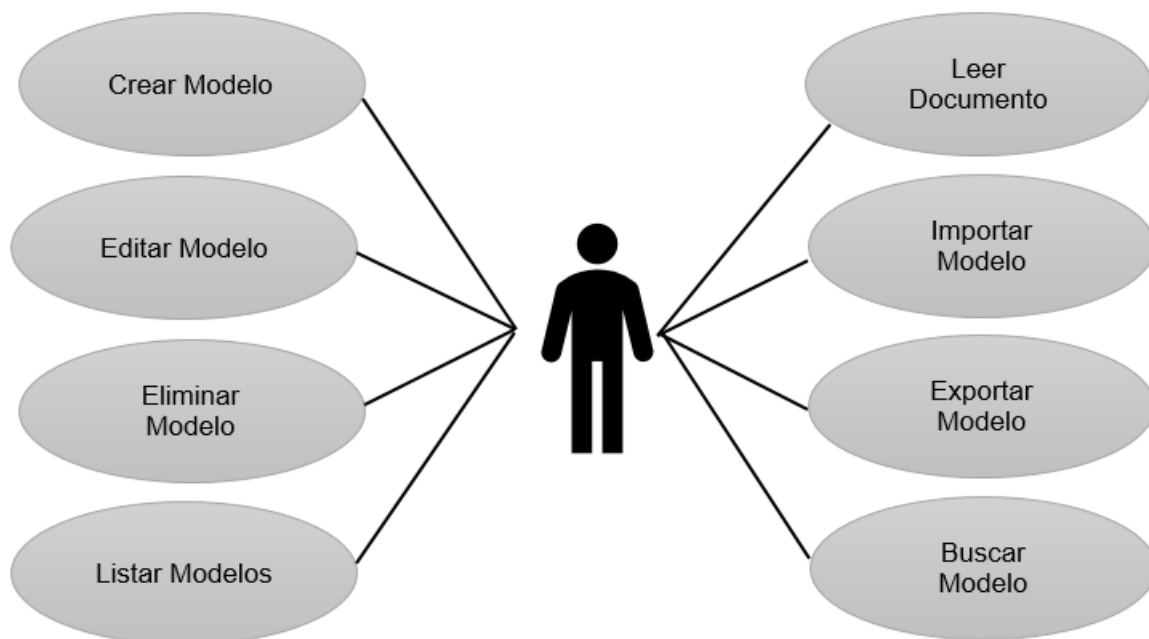


Figura 6-5 Diagrama de casos de uso de Gestión de modelos.

Referencia	CU06
Nombre	Crear modelo.
Descripción	Gestionar la creación de un nuevo modelo.
Actores	Diseñador
Relaciones	
Precondición	
Postcondición	Se crea un nuevo modelo.
Flujo básico	1.- El diseñador selecciona la opción nuevo modelo. 2.- Se crea un nuevo modelo con base a la plantilla de modelo inicial.
Flujo alternativo	2.- Si existe una excepción con la creación del modelo, se notifica con un mensaje de error.

Tabla 6-8 Caso de uso de crear modelo.

Referencia	CU07
Nombre	Editar modelo.
Descripción	Gestionar la edición de un nuevo modelo.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un modelo propiedad del usuario
Postcondición	Se almacena la modificación de un modelo.
Flujo básico	1.- El usuario diseñador busca y selecciona un modelo de su propiedad. 2.- El diseñador puede editar un modelo modificando tamaño, posición, elementos que lo conforman, relaciones entre sus elementos, gestionar las propiedades de los elementos, gestionar las propiedades del modelo. 3.- Se almacenan las modificaciones del modelo.
Flujo alternativo	2 o 3.- Si existe una excepción con la edición del modelo, se notifica con un mensaje de error.

Tabla 6-9 Caso de uso de editar modelo.

Referencia	CU08
Nombre	Eliminar modelo.
Descripción	Gestionar el proceso de eliminar un modelo.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un modelo propiedad del usuario
Postcondición	Se elimina un modelo.
Flujo básico	1.- El usuario diseñador busca y selecciona un modelo de su propiedad. 2.- El diseñador elige eliminar el modelo seleccionado. 3.- Se elimina el modelo de base de datos.
Flujo alternativo	2 o 3.- Si existe una excepción con el proceso de eliminar el modelo, se notifica con un mensaje de error.

Tabla 6-10 Caso de uso de editar modelo.

Referencia	CU09
Nombre	Listar modelos.
Descripción	Obtener una lista de los modelos propiedad del usuario.
Actores	Diseñador
Relaciones	
Precondición	Debe existir al menos un modelo propiedad del usuario
Postcondición	
Flujo básico	1.- El usuario diseñador ingresa al sistema en el área correspondiente. 2.- Accede al listado de modelos de su propiedad.
Flujo alternativo	2.- Si existe una excepción con el listado de modelos, se notifica con un mensaje de error.

Tabla 6-11 Caso de uso de listar modelos.

Referencia	CU10
Nombre	Buscar modelo.
Descripción	A través de una búsqueda obtener la información de un modelo elegido.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un modelo propiedad del usuario
Postcondición	
Flujo básico	1.- El usuario diseñador ingresa al sistema en el área correspondiente. 2.- Accede al listado de modelos de su propiedad. 3.- Selecciona un modelo del listado. 4.- Visualiza el modelo elegido.
Flujo alternativo	4.- Si existe una excepción con la búsqueda del modelo, se notifica con un mensaje de error.

Tabla 6-12 Caso de uso de buscar modelo.

Referencia	CU11
Nombre	Exportar modelo.
Descripción	El usuario diseñador elige exportar en formato XML o imagen el modelo en proceso de edición.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un modelo elegido propiedad del usuario
Postcondición	El modelo elegido es exportado a la carpeta Descargas.
Flujo básico	1.- El usuario diseñador busca un modelo de su propiedad. 2.- Visualiza el modelo elegido. 3.- Selecciona la opción exportar del menú. 4.- Elige el formato de salida del modelo los cuales pueden ser XML, JPEG, PNG o SVG.
Flujo alternativo	4.- Si existe una excepción con el proceso de exportar modelo, se notifica con un mensaje de error.

Tabla 6-13 Caso de uso de exportar modelo.

Referencia	CU12
Nombre	Importar modelo.
Descripción	El usuario diseñador elige importar en formato XML un modelo.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un modelo en formato XML
Postcondición	El modelo importado se crea como nuevo.
Flujo básico	1.- El usuario diseñador elige la opción importar modelo. 2.- Indica la ruta en donde se encuentra el modelo. 3.- Selecciona el modelo a importar en formato XML. 4.- Visualiza el modelo elegido. 5.- Se crea un nuevo modelo con base al XML del modelo importado.
Flujo alternativo	4 o 5.- Si existe una excepción con el proceso de importar modelo, se notifica con un mensaje de error.

Tabla 6-14 Caso de uso de importar modelo.

Referencia	CU13
Nombre	Leer documento.
Descripción	El usuario diseñador elige cargar un documento PDF con el procedimiento de respuesta a modelar.
Actores	Diseñador
Relaciones	
Precondición	Debe existir un documento en formato PDF.
Postcondición	El archivo es visualizado en el panel de documento.
Flujo básico	1.- El usuario diseñador elige la opción cargar documento. 2.- Indica la ruta en donde se encuentra el archivo. 3.- Selecciona el archivo a cargar en formato PDF. 4.- Visualiza el archivo elegido en el panel de documento.
Flujo alternativo	4.- Si existe una excepción con el proceso de leer documento, se notifica con un mensaje de error.

Tabla 6-15 Caso de uso de leer documento.

6.1.3 Diseño de la presentación del Modelador CMMN

El diseño mockups para la interfaz de la herramienta web Modelador CMMN, en donde el usuario construirá los modelos a partir de los procedimientos de respuesta, se ha diseñado a partir de los casos de uso descritos con anterioridad.

En la Figura 6-6 se muestra la interfaz de ingreso en la que el usuario debe validar su identidad y acceder a la herramienta o puede optar por crear una cuenta al rellenar los campos solicitados.



Figura 6-6 Prototipo de Interfaz de ingreso.

En la Figura 6-7 se muestra la interfaz principal de la herramienta en la cual el usuario diseña los modelos de los procedimientos de respuesta, carga los documentos, gestiona los modelos que ha diseñado.

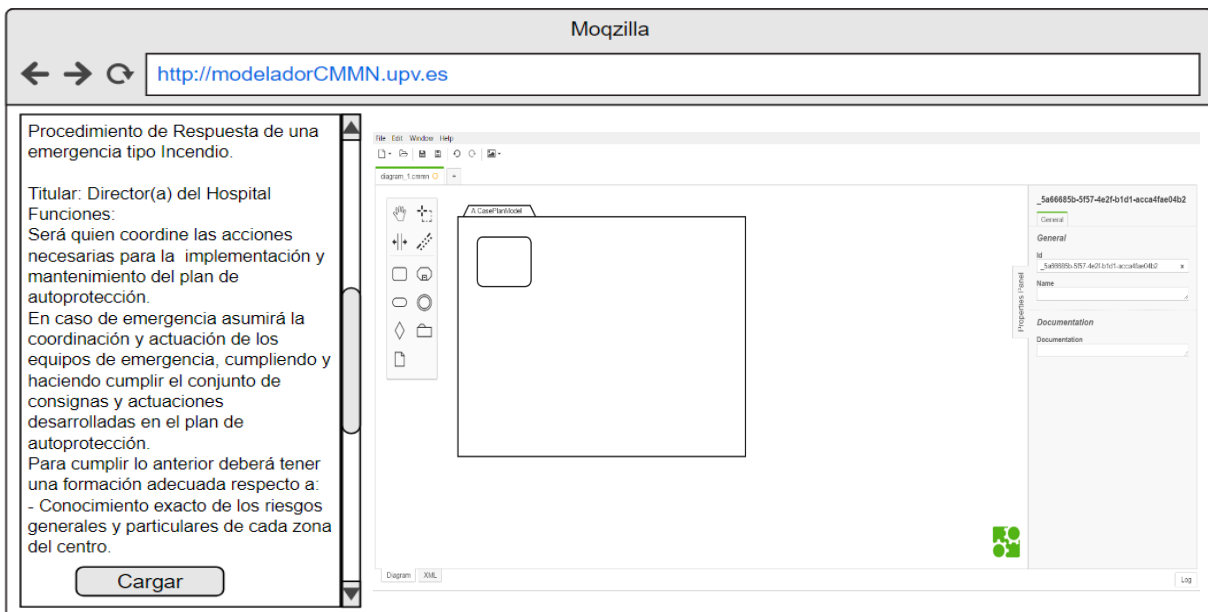


Figura 6-7 Prototipo de Interfaz de área de trabajo.

6.2 Ejecutor de Modelos

El núcleo del motor la plataforma Camunda BPM permite en su arquitectura que una aplicación personalizada puede consumir a través del servicio de API REST sus componentes, con base a esto se propone el diseño de una herramienta denominada “Ejecutor de Modelos” que mejore la interfaz en la que interviene el trabajador del caso y brinda interoperabilidad para que se pueda colaborar de manera bidireccional con otros entornos, los cuales pueden estar en distintas arquitecturas.

6.2.1 Arquitectura del Ejecutor de Modelos

Se propone una herramienta web que proporcione una adecuada interfaz al trabajador del caso al momento de ejecutar un modelo para gestionar una emergencia. En la figura 6-8 se presenta la arquitectura de la herramienta.

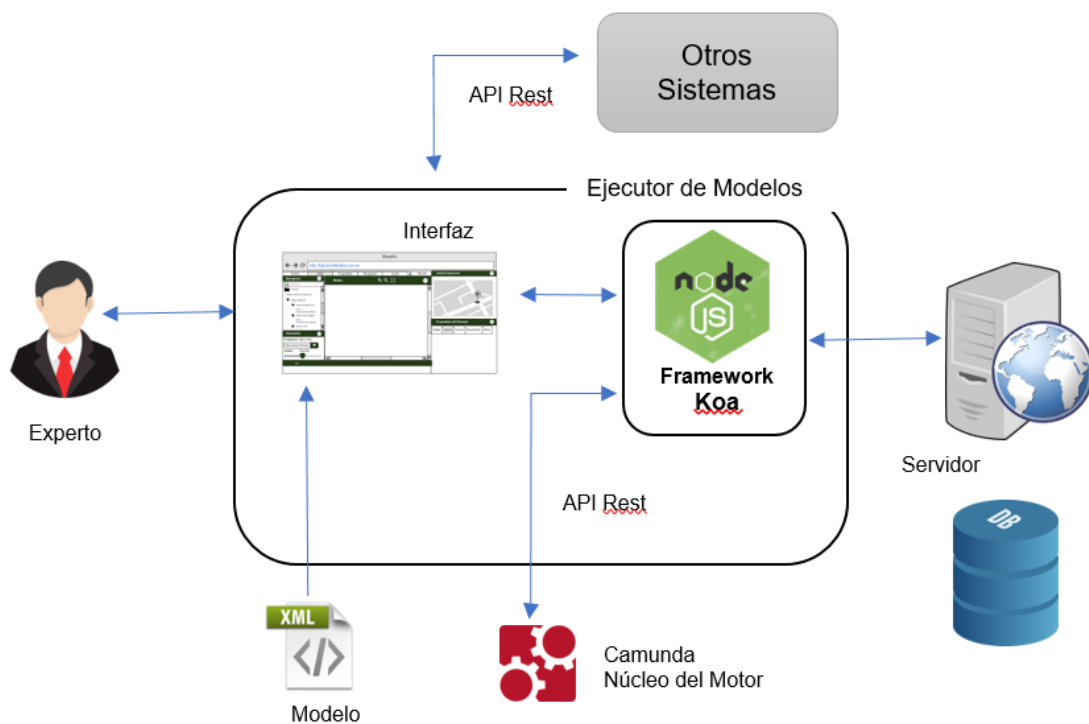


Figura 6-8 Arquitectura del Ejecutor de Modelos.

El trabajador del caso se le asigna el rol de usuario Experto y es quien podrá acceder a la herramienta mediante la identificación de sus credenciales, la herramienta permite al usuario experto interactuar con los paneles Navegación, Información, Modelo, Interfaz interacción, Propiedades del elemento y Log de actividades; lo cuales se describen más adelante. Así mismo se permite al usuario almacenar en un repositorio de PostgreSQL las propiedades del modelo y la bitácora de sus interacciones.

La arquitectura de esta herramienta estará soportada por el Framework Koa de Node.js que brinda mediante el patrón de diseño Modelo–Vista–Controlador (Model–View–Controller, MVC) la plataforma necesaria para gestionar servicios, interfaz, código fuente, librerías y acceso a base de datos PostgreSQL alojada en un servidor web. La herramienta sirve de puente como aplicación personalizada entre el motor del núcleo de la plataforma Camunda BPM, la interfaz de formularios web diseñada para los elementos elegidos por el experto y la interoperabilidad con otros sistemas, si así se requiere; ambas comunicaciones mediante API Rest.

6.2.2 Análisis y diseño conceptual del Ejecutor de Modelos

En esta sección se describe el análisis y diseño conceptual del ejecutor de modelos. En primer lugar, se establecen las características funcionales y no funcionales que se eligen para la realización del ejecutor de modelos, con la finalidad de abarcar todos los requerimientos necesarios para poder dar solución a la funcionalidad requerida y cumplir con los objetivos planteados en la herramienta propuesta. Posteriormente el diagrama de contexto representa a los actores y su interacción con el sistema. La funcionalidad se muestra con la utilización de casos de uso y su respectiva especificación.

6.2.2.1 Requisitos del Ejecutor de modelos

Esta sección detalla los requerimientos funcionales que posteriormente serán representados con mayor detalle en los casos de uso, así mismo, los requerimientos no funcionales orientados a aspectos técnicos fundamentados en la operatividad del ejecutor de modelos.

- Requisitos Funcionales

- El ejecutor de modelos permite la gestión de usuarios por medio de un rol administrativo con privilegios globales.
- El ejecutor de modelos permite la ejecución de modelos CMMN mediante el consumo de API Rest del núcleo del motor de Camunda BPM.
- El ejecutor de modelos permite al usuario almacenar y recuperar en un repositorio las propiedades del modelo y la bitácora de actividades de los registros de cada una de las ejecuciones realizadas.
- El ejecutor de modelos permite la importación de los modelos en formato XML.
- El ejecutor de modelos permite mostrar a través de su panel de *Navegación* el listado en forma de árbol jerárquico elementos ejecutados y la búsqueda de los elementos.
- El ejecutor de modelos mediante el panel de *Información* muestra al usuario la información del caso como son el usuario que ha diseñado el modelo, el listado de ejecuciones previas, el estado actual del modelo, y mediante una barra desplazadora se puede ir reproduciendo, con base a la unidad de medida tiempo, una ejecución previa.
- En el panel *Modelo* el usuario puede visualizar la imagen del modelo y mediante colores identificar los elementos instanciados disponibles para su ejecución y los que ya han sido ejecutados.
- En el panel de *Interfaz de Interacción* el usuario experto dispone de la interfaz denominada como propia, los cuales son los formularios web de los elementos elegidos para ser ejecutados dentro de la misma aplicación, o una interfaz externa que le permite interactuar de manera amigable con otros sistemas mediante API Rest y con ello identifica los valores enviados y recibidos en formato JSON.
- El ejecutor de modelos mediante el panel de *Propiedades* define para cada uno de los elementos las reglas de negocio, los valores predefinidos, el manejo de errores, la información de los documentos, las credenciales de los roles. En este panel se definen la información necesaria para interactuar con otros sistemas.
- En el panel de Log de Actividades el usuario visualiza en forma de lista una bitácora de las actividades que van ocurriendo durante la gestión del modelo. Dicha información es almacenada en el repositorio de base de datos o exportada en formato texto.

- **Requisitos No Funcionales**

- Almacenamiento en un gestor de base de datos relacional como lo es PostgreSQL.
- Autenticación de usuario para el acceso a la herramienta web.
- Interfaz de la herramienta web desarrollada en lenguaje Java Script.
- Se utiliza el Framework Koa, el cual implementa el patrón de diseño MVC.
- La interfaz de la herramienta web estará desarrollada mediante scripts de JQuery y hojas de estilo CSS que estarán insertados en páginas HTML.
- Visualización de la arquitectura web en cualquier navegador web de reciente actualización.

6.2.2.2 *Diagrama del contexto*

En esta sección se presentan los actores y su interacción con el ejecutor de modelos por medio del diagrama de contexto de la Figura 6-9. Principalmente se trabaja con dos tipos de usuarios cuya función se detalla en las Tablas 6-16 y 6-17.



Figura 6-9 Diagrama del contexto del Ejecutor de modelos.

Nombre	Administrador del Ejecutor de modelos
Descripción	Representa al usuario administrador, con roles y privilegios globales sobre el ejecutor de modelos. Encargado de la gestión de usuarios.

Tabla 6-16 Usuario Administrador del Ejecutor de modelos.

Nombre	Usuario Experto
Descripción	Representa al trabajador del caso el cual tiene un alto nivel de conocimiento en el lenguaje CMMN, definición de servicios de API Rest y la ejecución de modelos.

Tabla 6-17 Usuario Experto del Ejecutor de modelos.

6.2.2.3 *Diagrama estructurado de casos de uso*

La representación general de la funcionalidad del ejecutor de modelos está representada en el diagrama general de casos de uso de la Figura 6-10, en el cual se muestra la participación de cada actor definido en la sección anterior, con los diferentes casos generales de gestión de modelos. El administrador gestiona a los usuarios de rol experto. Así mismo, el usuario experto se encarga de crear, recuperar, actualizar o eliminar (CRUD) casos que sean de su propiedad, además se le permite importar en formato XML modelos previamente definidos en otros entornos que soporten el estándar CMMN.

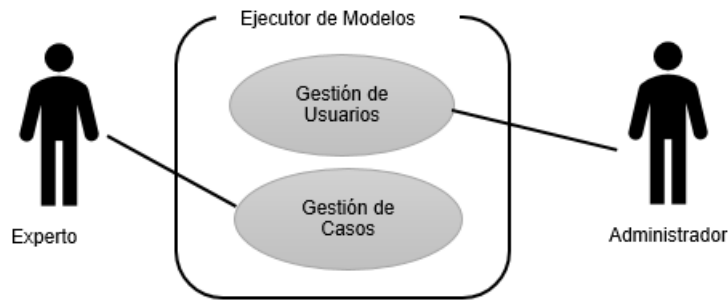


Figura 6-10 Diagrama estructurado de casos de uso.

Una vez que han sido definidos de manera general los casos de uso que integran el modelador CMMN, se procede con el detalle y especificación de cada uno de los componentes y sus relaciones con mayor nivel de detalle. Para una mejor explicación se divide en dos subsistemas (administrador y diseñador) y dentro de cada uno se describe la funcionalidad agrupada por características o gestores según su comportamiento dentro del modelador CMMN.

- Subsistema Administrador

Este subsistema describe las características en las que participa el usuario de tipo administrador y que involucra la gestión de los usuarios de tipo experto. La funcionalidad está representada por 5 casos de uso (crear, modificar, eliminar, listar y búsqueda) que fueron descritas en los casos de uso CU01, CU02, CU03, CU04, CU05 y CU06 en la sección 6.1.2.3. Se define que un usuario dado de alta en la herramienta Modelador CMMN puede loguearse en la herramienta Ejecutor de modelos y viceversa; siempre y cuando en su perfil active la opción del rol necesario.

- Subsistema Experto

La definición del subsistema experto se deja pendiente como parte de los trabajos futuros a complementar y desarrollar del presente trabajo fin de master.

6.2.3 Diseño de la presentación del Ejecutor de Modelos

El diseño mockups para la interfaz de la herramienta web Ejecutor de modelos, en donde el usuario experto ejecuta los modelos previamente construidos y le permite interactuar de manera amigable con el núcleo del motor de la plataforma Camunda BPM y comunicarse con otros sistemas con los que tenga que intervenir durante el proceso de ejecución; lo anterior mediante el servicio de API Rest.

En la Figura 6-11 se enseña la interfaz de ingreso en la que el usuario debe validar su identidad y acceder a la herramienta o puede optar por crear una cuenta al rellenar los campos solicitados.



Figura 6-11 Prototipo de Interfaz de ingreso.

En la Figura 6-12 se muestra la interfaz principal de la herramienta en la cual el usuario ejecuta los modelos de los procedimientos de respuesta, asigna las propiedades a los casos y gestiona los casos que son de su propiedad.

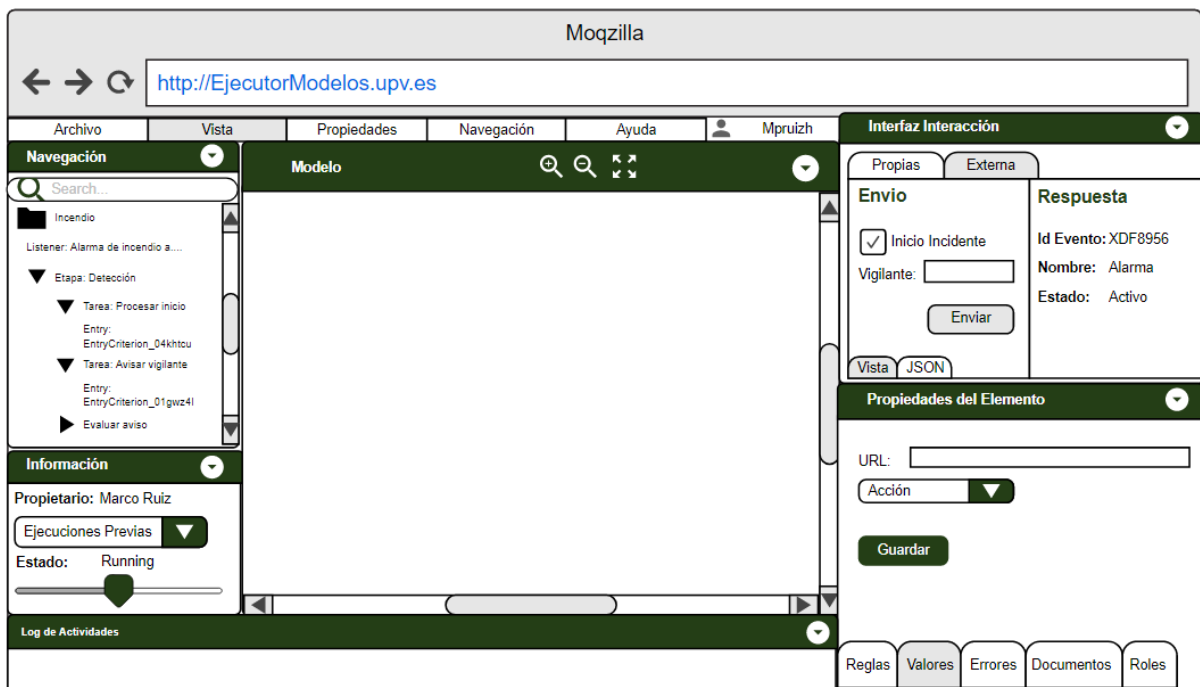


Figura 6-12 Prototipo de Interfaz de área de trabajo.

Una vez que se ha presentado la propuesta del marco de trabajo en la que se mejora la herramienta para el diseño de modelos basados en procedimientos de respuesta en el dominio de la gestión de emergencias, y la herramienta que permite ejecutar los modelos e interactuar con otros sistemas, se brinda una solución a la pregunta de investigación RQ5, así mismo, se alcanza el objetivo específico SO5 del trabajo de investigación:

SO5: Proponer el diseño de aplicaciones web basadas en la plataforma Camunda BPM que sirva de Framework para el trabajador del caso con la finalidad de facilitar el diseño y ejecución de un modelo CMMN, así como la interacción con otros entornos.

7. Conclusiones y trabajos futuros

En esta sección se aborda la sexta pregunta de investigación:

RQ6: ¿Qué conclusiones y recomendaciones pueden realizarse sobre la aplicación del Lenguaje CMMN en el dominio de casos de emergencia?

7.1 Conclusiones

A medida que aumenta la necesidad de soluciones de TI que soporten casos empresariales complejos y basados en el conocimiento, surgen nuevos enfoques y tecnologías. La gestión flexible de casos aparece como una de las innovaciones más atractivas y viables en ese contexto, por ello se ha decidido aplicar al dominio de la gestión de emergencias.

Como se ha expuesto, un modelo de lenguaje CMMN durante su ejecución puede modificar su flujo de trabajo mediante los valores de las variables de entorno y la instanciación de tareas discretas que decida ejecutar el trabajador del caso con base a su experiencia, esta característica es lo que permite dar atención adecuada mediante la gestión flexible de casos a problemas clasificados como negocios intensivos.

En el presente trabajo de fin de master se ha realizado un caso de estudio en el cual de manera satisfactoria un procedimiento de respuesta de tipo incendio se ha analizado y con base al estudio del lenguaje CMMN se ha diseñado un modelo para posteriormente ser ejecutado en distintos escenarios cambiantes con la finalidad de recrear diversos flujos de ejecución y de esa manera dar atención satisfactoria.

Las conclusiones derivadas de los objetivos específicos de este trabajo se describen a continuación:

Se realizó un profundo estudio del estado del arte de la gestión flexible de casos y de los distintos lenguajes que pueden ser aplicados para darle atención. De esto se concluye que el lenguaje CMMN es la mejor opción para brindar atención a problemas relacionados a la gestión flexibles de casos.

Se estudió a detalle los conceptos y elementos que conforman el lenguaje de CMMN para contar con la base teórica necesaria para comprender la utilización de sus elementos, la relación que se establecen entre ellos y las reglas de sintaxis que lo definen.

Con base a lo anterior, se ha logrado transformar un procedimiento de respuesta de tipo incendio expresado en lenguaje natural a un modelo en lenguaje CMMN y con ello tener un objeto digital capaz de ser interpretado y ejecutado por un computador a través de la plataforma Camunda BPM.

Una vez que se cuenta con el modelo en CMMN y se configura el entorno de trabajo, se puede ejecutar a través de la plataforma Camunda BPM e ingresar valores de entorno distintos en tiempo de ejecución para ir recreando diversas situaciones, con ello se logra validar el modelo ante situaciones cambiantes lo cual es característico en la gestión de emergencias.

Si bien la plataforma Camunda BPM es una herramienta capaz de ejecutar modelos en CMMN y que su núcleo de motor permite ser consumido mediante API REST, la interfaz para un trabajador del caso no es la adecuada en situaciones de emergencia, es por ello por lo que se ha definido un marco de trabajo cuya interfaz facilite al trabajador del caso el diseño de modelos de los procedimientos de respuesta, la ejecución de los modelos y la interoperabilidad con otros sistemas.

7.2 Trabajos futuros

En esta sección se describen algunos temas que pueden desarrollarse con mayor detalle o que sirvan para complementar el trabajo que se ha realizado, considerando la aplicación de la gestión flexible de casos al dominio de la gestión de emergencias, se pueden considerar las siguientes:

- Como la Gestión flexible de casos es una disciplina bastante nueva el método se desarrolla bajo las características de un enfoque de cascada. Una vez que se consigue la madurez necesaria se recomienda un enfoque de desarrollo ágil.
- La propuesta de marco de trabajo debe ser desarrollada con mayor nivel de detalle y considerando la interacción con otros sistemas que pueden estar definidos en distintas arquitecturas.
- Si bien CMMN permite atender la gestión de casos flexibles, no todos los problemas conservan durante todas sus etapas la misma clasificación, por lo tanto, se debe atender cada etapa o grupo de tareas de acuerdo con la necesidad de intervención del trabajador del caso. Tanto el modelador de Camunda, utilizado como base en la propuesta del nuevo entorno de modelado, como el núcleo del motor de la plataforma Camunda BPM interpretan adecuadamente las notaciones de DMN, BPMN y CMMN; las cuales a su vez pueden combinarse en un mismo modelo [24], por ello se tendrá un mejor resultado si el marco de trabajo propuesto se desarrolla partiendo de esta idea.
- Considerar la incorporación de Machine Learning en la propuesta del marco de trabajo para construir un sistema de recomendación que brinde apoyo durante la fase de toma de decisiones al trabajador del caso, con base a los resultados de modelos previamente entrenados en situaciones similares al que se esté presentando durante el desarrollo de la emergencia.
- Complementar, desarrollar e implementar el diseño del marco de trabajo propuesto con la finalidad de obtener una herramienta que sirva de base para que los trabajadores del caso realicen sistemas que brinden atención a las situaciones del dominio de la gestión de emergencias; y con ello colaborar en materia de protección civil.

En la nueva era del conocimiento, los usuarios expertos son quienes hacen la diferencia con su experiencia. Depende de nosotros apoyarlos con herramientas TI que brinden una plataforma adecuada al momento de intervenir.

Las TI tienen un sinnúmero de aplicaciones para mejorar la calidad de vida de las personas, en esta ocasión el objetivo principal del presente trabajo fin de master es brindar una propuesta que permita desarrollar una herramienta que colabore en materia de protección civil al momento de gestionar una emergencia y con ello salvaguardar la mayor cantidad de vidas y recursos posibles.

Una vez que se han presentado las conclusiones y trabajos futuros se atiende la pregunta de investigación RQ6 y con ello se alcanza el objetivo SO6 de actual trabajo de investigación.

SO6: Presentar conclusiones y recomendaciones que aporten futuras líneas de investigación sobre la aplicación del lenguaje CMMN en el dominio de casos de emergencia.

Referencias Bibliográficas

- [1] Sánchez, José., Carsí, José. y Penadés M. Carmen, “Un entorno de gestión de casos para la resolución flexible de emergencias”, ISSI-DSIIC, Universidad Politécnica de Valencia, 2015. [Online] Available: https://biblioteca.sistedes.es/wp-content/uploads/2015/08/3_paper_581.pdf
- [2] OMG. “Case Management Model and Notation, version 1.0, 2014. Document formal/2014-05-05”. [Online] Available: <http://www.omg.org/spec/CMMN/1.0/>
- [3] Penadés M. Carmen, “Tema 02 – Dimensiones de los SGE”, Material académico de la asignatura Sistemas de Gestión de Emergencias del master MUITSS 2016-2017 de la Universidad Politécnica de Valencia, 2017.
- [4] Gobierno de España, “Deber de colaboración”, Ley 17/2015, de 9 de julio, del Sistema Nacional de Protección Civil, artículo 7 bis”, Agencia Estatal Boletín Oficial del Estado, Referencia BOE-A-2015-7730, BOE Núm 164, pp 12. [Online] Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2015-7730>
- [5] Knut Hinkelmann, “Case Management Model and Notation-CMMN”, 2013. [Online] Available: http://knut.hinkelmann.ch/lectures/bpm2013-14/06_CMMN.pdf
- [6] Camunda, “Core Engine, Camunda features, Execute models”. [Online] Available: <https://camunda.org/features/engine/>
- [7] Camunda, “Camunda CMMN 1.1 Engine”, 2016. [Online] Available: <https://camunda.org/cmmn/tool/>
- [8] Sampieri, R., Collado C., Lucio P., “Metodología de la Investigación”, McGraw-Hill Inc, 4ta Edición, ISBN: 970-10-5753-8, pp. 3-29, 2006.
- [9] Shaw, M., “What makes Good Research in Software Engineering?”, European Joint Conference on Theory and Practice of Software ETAPS 2002, April 2002. [Online] Available: <http://www-2.cs.cmu.edu/~Compose/ftp/shaw-fin-etaps.pdf>
- [10] Apache 2.0, “Library mxGraph”, Github, 2016. [Online] Available: <https://github.com/jgraph/mxgraph>
- [11] Vanderfeesten, I., Reijers, H.A., Wil M.P. van der Aalst. Case Handling Systems as Product Based Workflow Design Support. ICEIS 2007, LNBIP 12, pp. 187–198, 2008.
- [12] Wil M.P. Van der Aalst, Mathias Weske, Dolf Grünbauer. Case handling: a new paradigm for business process support. Data & Knowledge Engineering pp. 129–162, 2005.
- [13] MarketsandMarkets, “Business Process Management Market by Solutions (Process Modeling, Automation, Integration, Content & Document Management, and Monitoring & Optimization Management), by End User (SMBs, Enterprises & Large Enterprises)” Global Forecast to 2019, April 2016, [Online] Available: <http://www.marketsandmarkets.com/PressReleases/business-process-management.asp>
- [14] Gartner Inc., “Gartner's 2013 BPM Summit”, Maryland, US: Gartner Inc., 2013.

- [15] Jakob Freund, “CMMN – The BPMN for Case Management?”, December 2014, [Online] Available: <http://www.bpm-guide.de/2014/12/20/cmmn/>
- [16] de Man, H., “Case Management: A Review of Management Approaches”, BPTrends, 2009.
- [17] Tscheschner, W., “Transformation from EPC to BPMN”, Potsdam, Germany: Hasso-Plattner-Institute, 2008.
- [18] Franke, J., “Case Management Process Modeling (CMPM) – An Introduction”, February 2015. [Online] Available: <http://scn.sap.com/people/joern.franke/blog/2011/07/27/case-management-process-modeling-cmpm-an-introduction>
- [19] Cohn, D., & Hull, R., “Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes”. IBM T.J. Watson Research Laboratory. Hawthorne: IBM, 2009
- [20] IBM., “What is Business Rules Management?”, February 2015, [Online] Available: <http://www-01.ibm.com/software/websphere/products/business-rule-management/whatis/>
- [21] Swenson, K. D., Ukelson, J. P., Shepherd, T., Matthias, J. T., Pucher, M. J., Khoiyi, D., . . . Lee, C., “Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done”, Megan-Kiffer, 2010.
- [22] IBM, “Business process management and case management”, IBM Knowledge Center. [Online] Available: <https://www.ibm.com/support/knowledgecenter/en/SSV2LR/com.ibm.wbpm.wle.editor.doc/topics/cbpmmandcasemgmt.html>
- [23] OMG, “About OMG”. [Online] Available: <http://www.omg.org/about/index.htm>
- [24] OMG, “BPMN, CMMN and DMN specifications at OMG”, [Online] Available: <http://www.omg.org/intro/TripleCrown.pdf>
- [25] KRESS, J. M., “Adaptive Case Management in Practice. Dissertação de Mestrado em Engenharia Elétrica”, Publicação PPPGEE.DM-632/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 77p, 2016.
- [26] Clair, C. L., & Miers, D., “The Forrester Wave: Dynamic Case Management”, Q1 2014. Forrester.
- [27] Hill, J. B., Chin, K., & Dunie, R., “Critical Capabilities for Case Management”, Gartner, 2014.
- [28] Joan C. Fuster, “Case Management versus Business Process Management”, April 2016. [Online] Available: <http://blog.auraportal.com/es/case-management-vs-business-process-management/>
- [29] Peter J., “Case Management Model and Notation, analyse of a practical application”, Masteropleiding Business Process Managementand IT, December 2015, pp. 19.
- [30] Camunda, “Repositories Camunda BPM”, GitHub. [Online] Available: <https://github.com/camunda>

- [31] Camunda, “CMMN 1.1 Implementation Reference”. [Online] Available: <https://docs.camunda.org/manual/7.7/reference/cmmn11/>
- [32] Gobierno de España, “Ley 2/1985 de 21 de enero sobre protección civil”, Agencia Estatal Boletín Oficial del Estado, Referencia BOE-A-1985-02092, BOE Núm. 22, pp. 2092 – 2095. [Online] Available: <https://www.boe.es/boe/dias/1985/01/25/pdfs/A02092-02095.pdf>
- [33] Gobierno de España, “Real Decreto 393/2007 de 23 de marzo sobre norma básica de autoprotección”, Agencia Estatal Boletín Oficial del Estado, Referencia BOE-A-2007-12841, BOE Núm. 72, pp. 12841 – 12850. [Online] Available: <https://www.boe.es/boe/dias/2007/03/24/pdfs/A12841-12850.pdf>
- [34] Camunda, “Get started with Camunda and CMMN 1.1”. [Online] Available: <https://docs.camunda.org/get-started/cmmn11/>

Apéndice A - Definiciones

Accesibilidad: Datos e información relevante sobre el acceso. Características de los accesos de vehículos a las fachadas del establecimiento. Número de fachadas accesibles a bomberos.

Actividad: Conjunto de operaciones o tareas que puedan dar origen a accidentes o sucesos que generen situaciones de emergencia.

Aforo: Capacidad total de público en un recinto o edificio destinado a espectáculos públicos o actividades recreativas.

Alarma: Aviso o señal por la que se informa a las personas para que sigan instrucciones específicas ante una situación de emergencia.

Alerta: Situación declarada con el fin de tomar precauciones específicas debido a la probable y cercana ocurrencia de un suceso o accidente.

Altura de evacuación: La diferencia de cota entre el nivel de un origen de evacuación y el del espacio exterior seguro.

Autoprotección: Sistema de acciones y medidas, adoptadas por los titulares de las actividades, públicas o privadas, con sus propios medios y recursos, dentro de su ámbito de competencias, encaminadas a prevenir y controlar los riesgos sobre las personas y los bienes, a dar respuesta adecuada a las posibles situaciones de emergencia y a garantizar la integración de estas actuaciones en el sistema público de protección civil.

Centro, establecimiento, espacio, dependencia o instalación: La totalidad de la zona, bajo control de un titular, donde se desarrolle una actividad.

Confinamiento: Medida de protección de las personas, tras un accidente, que consiste en permanecer dentro de un espacio interior protegido y aislado del exterior.

Entorno: Información sobre el entorno (urbano, rural, proximidad a ríos, a rutas por las que transitan vehículos con mercancías peligrosas, a industrias, a zonas forestales, edificio aislado o medianero con otras actividades. Tipo de actividades del entorno y sus titulares.) Vulnerables existentes en el entorno.

Evacuación: Acción de traslado planificado de las personas, afectadas por una emergencia, de un lugar a otro provisional seguro.

Focos de peligro y vulnerables: Tipo de riesgo más significativo que emana del edificio. Tipo y cantidad de productos peligrosos que se almacenan y/o procesan.

Intervención: Consiste en la respuesta a la emergencia, para proteger y socorrer a las personas y los bienes.

Medios: Conjunto de personas, máquinas, equipos y sistemas que sirven para reducir o eliminar riesgos y controlar las emergencias que se puedan generar.

Ocupación: Máximo número de personas que puede contener un edificio, espacio, establecimiento, recinto, instalación o dependencia, en función de la actividad o uso que en él se desarrolle. El cálculo de la ocupación se realiza atendiendo a las densidades de ocupación indicadas en la normativa vigente. No obstante, de preverse una ocupación real mayor a la resultante de dicho cálculo, se tomará esta como valor de referencia. E igualmente, si

legalmente fuera exigible una ocupación menor a la resultante de aquel calculo, se tomará esta como valor de referencia.

Órgano competente para el otorgamiento de licencia o permiso para la explotación o inicio de actividad: El Órgano de la Administración Pública que, conforme a la legislación aplicable a la materia a que se refiere la actividad, haya de conceder el título para su realización.

Peligro: Probabilidad de que se produzca un efecto dañino específico en un periodo de tiempo determinado o en circunstancias determinadas.

Plan de Autoprotección: Marco orgánico y funcional previsto para una actividad, centro, establecimiento, espacio, instalación o dependencia, con el objeto de prevenir y controlar los riesgos sobre las personas y los bienes y dar respuesta adecuada a las posibles situaciones de emergencias, en la zona bajo responsabilidad del titular, garantizando la integración de estas actuaciones en el sistema público de protección civil.

Plan de actuación en emergencias: Documento perteneciente al plan de autoprotección en el que se prevé la organización de la respuesta ante situaciones de emergencias clasificadas, las medidas de protección e intervención a adoptar, y los procedimientos y secuencia de actuación para dar respuesta a las posibles emergencias.

Planificación: Es la preparación de las líneas de actuación para hacer frente a las situaciones de emergencia.

Prevención y control de riesgos: Es el estudio e implantación de las medidas necesarias y convenientes para mantener bajo observación, evitar o reducir las situaciones de riesgo potencial y daños que pudieran derivarse. Las acciones preventivas deben establecerse antes de que se produzca la incidencia, emergencia, accidente o como consecuencia de la experiencia adquirida tras el análisis de las mismas.

Procedimientos de respuesta: Las acciones detalladas a realizar por cada uno de los actores ante una situación de emergencia en particular.

Puertos comerciales: Los que en razón a las características de su tráfico reúnen condiciones técnicas, de seguridad y de control administrativo para que en ellos se realicen actividades comerciales portuarias, entendiéndose por tales las operaciones de estiba, desestiba, carga, descarga, transbordo y almacenamiento de mercancías de cualquier tipo, en volumen o forma de presentación que justifiquen la utilización de medios mecánicos o instalaciones especializadas.

Recursos: Elementos naturales o técnicos cuya función habitual no está asociada a las tareas de autoprotección y cuya disponibilidad hace posible o mejora las labores de prevención y actuación ante emergencias.

Rehabilitación: Es la vuelta a la normalidad y reanudación de la actividad.

Riesgo: Grado de pérdida o daño esperado sobre las personas y los bienes y su consiguiente alteración de la actividad socioeconómica, debido a la ocurrencia de un efecto dañino específico.

Titular de la actividad: La persona física o jurídica que explote o posea el centro, establecimiento, espacio, dependencia o instalación donde se desarrollen las actividades

Apéndice B – Forma CMMN

Este apéndice describe un formulario (plantilla), que puede usarse para preparar una especificación de acuerdo con la versión estándar 1.0 del lenguaje CMMN [2].

El documento consta de una especificación de plantilla y diagramas, que se estructura de acuerdo con el agrupamiento de elementos de diseño CMMN como se define en el capítulo 3 de este documento de la norma de notación CMMN como se mencionó anteriormente.

El alcance del formulario se limita a los elementos que se ajustan al modelo de comportamiento / modelo de plan de CMMN.

En este trabajo y en el propio estándar CMMN se puede encontrar otra terminología y una tabla resumen de los diferentes elementos de la CMMN.

1. Declaración Base de CMMN

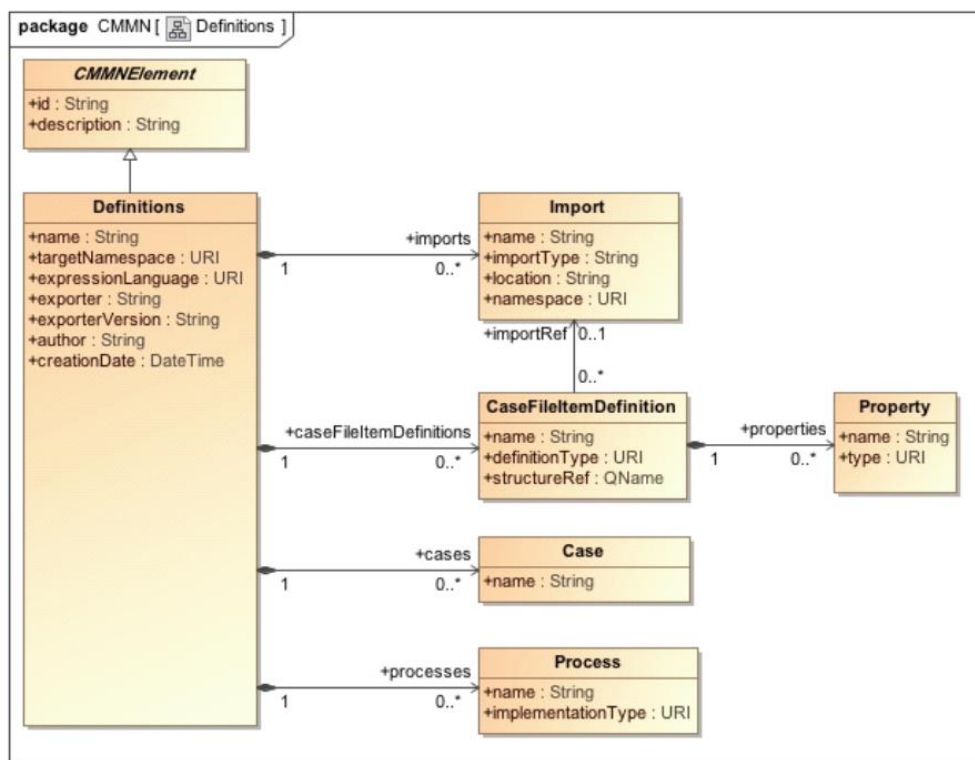


Figura B-1 Definición del diagrama de clases.

2. Elementos del Caso de CMMN

Case (Caso) es un concepto de nivel superior que combina todos los elementos que constituyen un modelo de caso. El siguiente diagrama ilustra el metamodelo del Caso y sus clases asociadas.

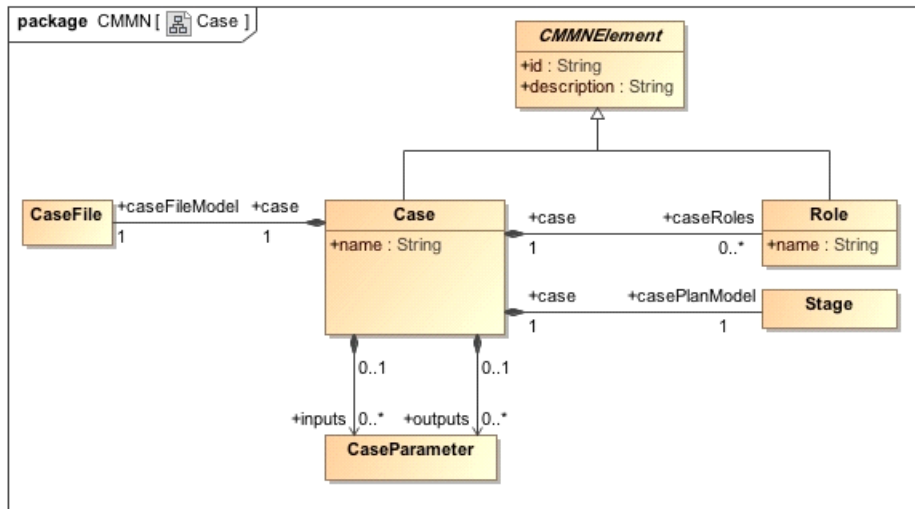


Figura B-2 Diagrama de la clase del caso.

Un caso consiste en un caseFileModel, un casePlanModel y un conjunto de caseRoles. También contiene entradas y salidas, para permitir la interacción del Caso con su entorno.

- Case

La clase Case hereda de la clase CMMNElement y se compone de los siguientes atributos adicionales:

Atributo	Descripción
name : String	El nombre del Caso
caseRoles : Role[0..*]	Este atributo enumera los objetos de rol asociados con el caso. Estos roles son específicos del caso, y no se conocen fuera del contexto del caso.
caseFileModel: CaseFile[1]	Un objeto CaseFile. Cada caso DEBE ser asociado con exactamente un CaseFile.
casePlanModel: Stage[1]	El modelo del plan del caso. Cada caso DEBE estar asociado con exactamente un modelo de plan. Se define por asociación a Etapa (Stage) representa un concepto recursivo, usadas como contenedor de elementos de los cuales el plan del caso está construido y puede evolucionar más, y tener un ciclo de vida que puede ser Rastreado en tiempo de ejecución. La etapa "más externa" se asocia al caso como su modeloPlanModel.
inputs: CaseParameter[0..*]	Parámetros de entrada del caso. Un caso puede tener parámetros de entrada para que pueda ser llamado desde fuera.
outputs: CaseParameter[0..*]	Parámetros de salida del caso. Un caso puede tener parámetros de salida para que pueda devolver un resultado.

Tabla B-1 Atributos del caso

- Rol

CaseRoles autorizan a los trabajadores de casos o equipos de trabajadores del caso a realizar HumanTasks (especificados en 3.3.2.3), plan basado en DiscretionaryItems, y elevar eventos de usuario (activando UserEventListeners, como se especifica en 3.3.2.5).

Ejemplo de Roles de un caso puede ser:

- Doctor - Una función del médico puede contener uno o más participantes que están autorizados a realizar HumanTasks, activar UserEventListeners, o hacer la planificación que requiere conocimientos de médico.
- Paciente - Un caso puede proporcionar una interfaz para que los pacientes realicen una planificación que pueda corresponder a citas de programación, HumanTasks completos que puedan corresponder a proporcionar información sobre su salud, etc. En una aplicación típica, un caso puede limitar al paciente. Participante único.
- Enfermera - Una función de la enfermera puede representar a uno o más participantes con las habilidades de un proveedor de cuidado de la enfermera.

La asignación de funciones a los participantes, como individuos o equipos, no está incluida en el ámbito de la CMMN.

Atributo	Descripción
name : String	El nombre del Rol
case : Case[1]	El caso que contiene el caseRoles.

Tabla B-2 Atributos del Rol

3. Elementos del Plan de Control

PlanItemControls definen aspectos del control de instancias de Tareas, Etapas, EventListeners e Hitos. Se definen en relación con sus "orígenes" en el modelo - PlanItems y DiscretionaryItems - y pueden ser omitidos por PlanItemControls que se definen en relación con las PlanItemDefinitions a las que los PlanItems y DiscretionaryItems se refieren a través de su definitionRef.

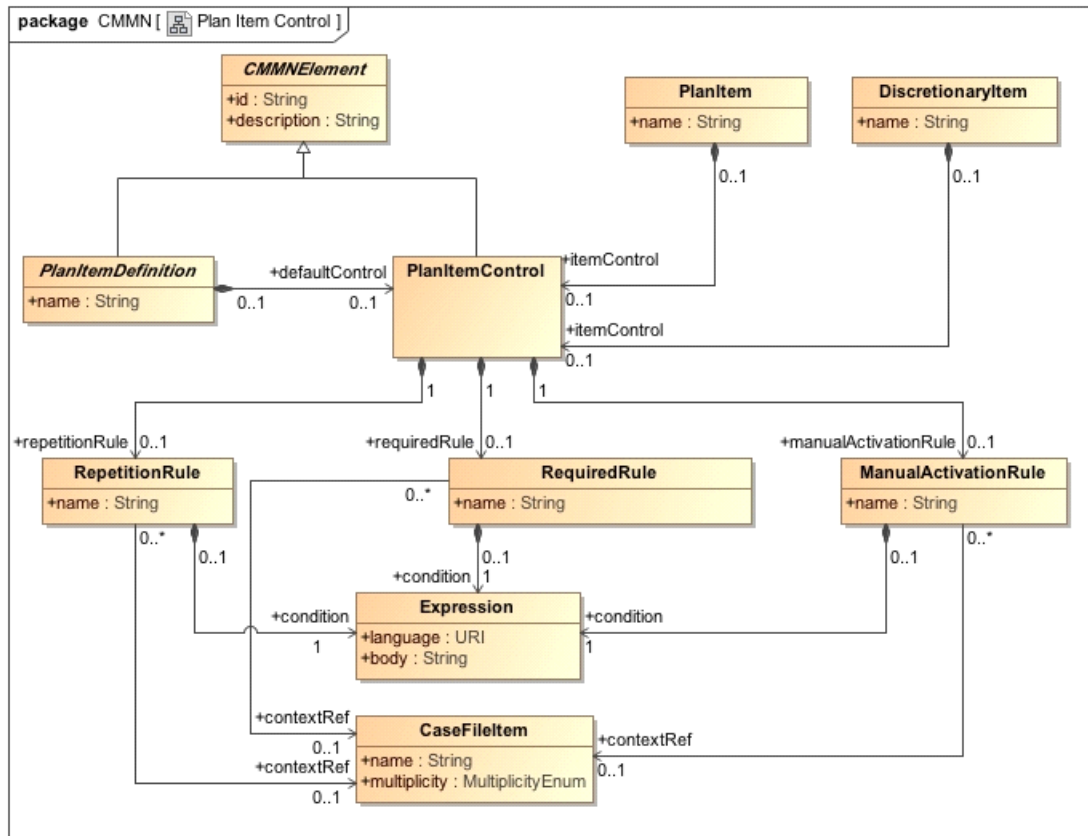


Figura B-3 Diagrama de clases del PlanItemControl.

La clase PlanItemControl hereda de CMMNElement y tiene los siguientes atributos.

Atributo	Descripción
manualActivationRule :ManualActivationRules[0..1]	(a) condición: Expresión [1] que DEBE evaluar un Boolean. True: la instancia de Task o Stage se activa automáticamente cuando está en estado Disponible. False: la instancia de Task o Stage se activa manualmente cuando está en estado Activado. (b) contextRef: CaseFileItem [0..1] CaseFileItem que sirve como punto de partida para la evaluación de la expresión que se especifica por la condición de ManualActivationRule. Si no se especifica, la evaluación comienza en el CaseFile objeto al que hace referencia el caso como su caseFileModel.
requiredRule: RequiredRule[0..1]	(a) condición: Expresión [1] que DEBE evaluar un Boolean. True: instancia de la Task o Stage o MileStone es obligatorio y DEBE estar en estado Desactivado, Completado, Terminado o Fallido antes de que se pueda completar el escenario (instancia) que contiene.

	<p>False: la instancia de la Task o Stage o MileStone se considera opcional antes de que se pueda completar el escenario (instancia) que contiene.</p> <p>(b) contextRef: CaseFileItem [0..1]</p> <p>CaseFileItem que sirve como punto de partida para la evaluación de la expresión que se especifica por la condición de RequiredRule. Si no se especifica, la evaluación comienza en el objeto CaseFile al que hace referencia Case como su caseFileModel.</p>
<p>repetitionRule: RepetitionRule[0..1]</p>	<p>(a) condición: Expresión [1] que DEBE evaluar un Boolean.</p> <p>True: La instancia de Task o Stage o MileStone puede repetirse. Falso: la instancia de Task o Stage o MileStone puede no repetirse.</p> <p>(b) contextRef: CaseFileItem [0..1]</p> <p>CaseFileItem que sirve como punto de partida para la evaluación de la expresión que se especifica por la condición de la RepetitionRule. Si no se especifica, la evaluación comienza en el CaseFile Objeto al que hace referencia el caso como su CaseFileModel.</p>

Tabla B-3 Atributos de PlanItemControl y asociaciones de modelos

4. Información del Modelo

El modelo de información de un Caso comprende clases para el manejo de los aspectos de información (datos) de un Caso. Toda la información, o referencias a la información, que se requiere como contexto para manejar un Caso, es definida por un CaseFile.

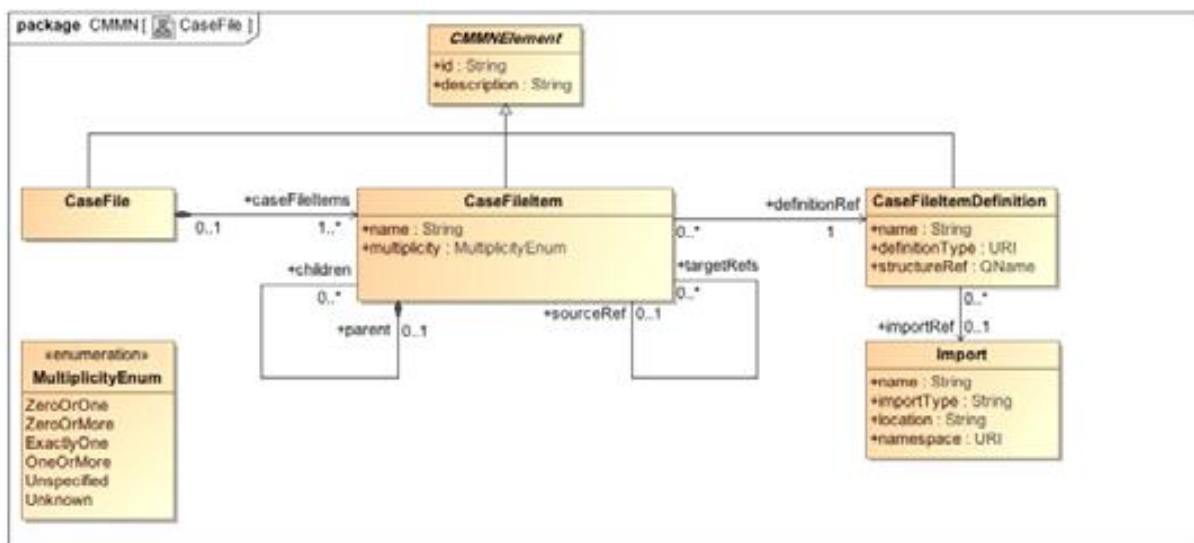


Figura B-4 Diagrama de clases del CaseFile.

La información en el CaseFile sirve como contexto para levantar eventos y evaluar expresiones, así como punto de referencia para CaseParameters, como entradas y salidas de tareas. CaseFile también sirve como contenedor de datos que es accesible por otros sistemas y personas fuera del caso, a través de CaseParameters. CaseFile se entiende como modelo lógico. No implica ninguna suposición sobre el almacenamiento físico de información.

Atributo	Descripción
caseFileItems: CaseFileItem[1..*]	Este atributo enumera los CaseFileItems de un CaseFile. Un CaseFile DEBE contener al menos un CaseFileItem

Tabla B-4 Atributos de CaseFile y asociaciones de modelos

5. Definición del Elemento del Plan

Los elementos PlanItemDefinition definen los bloques de construcción a partir de los cuales se construyen los planes Case (instancia). PlanItemDefinition es una clase abstracta que hereda de CMMNElement.

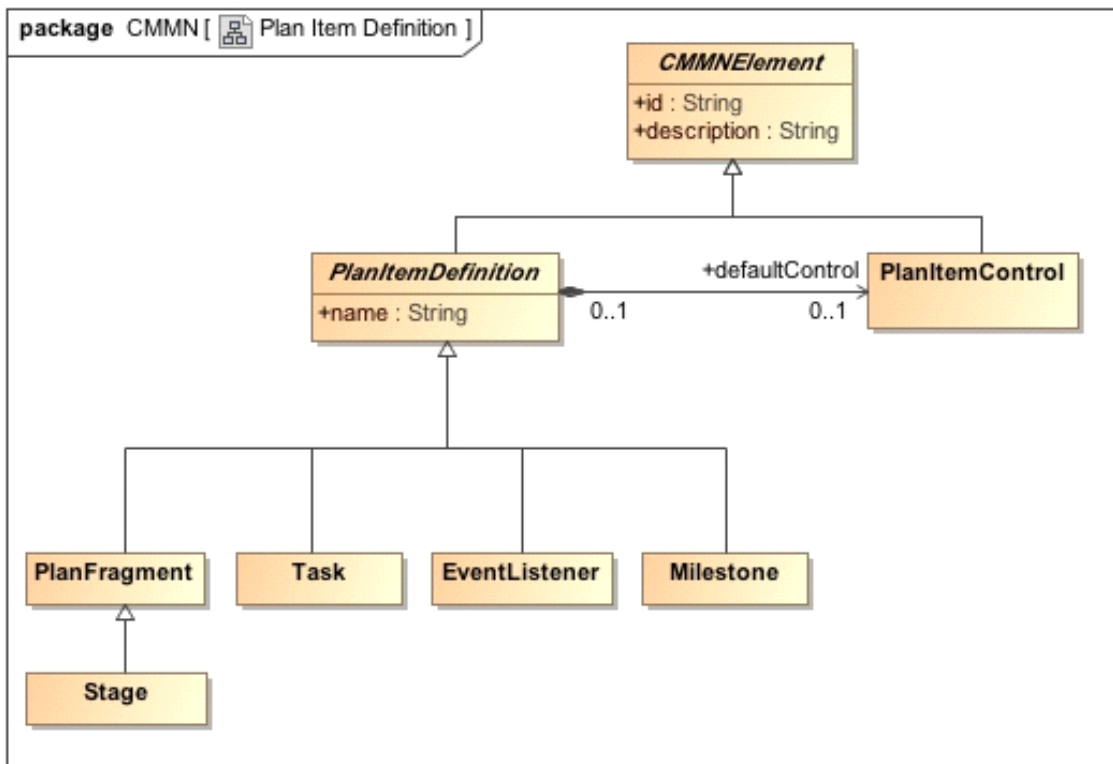


Figura B-5 Diagrama de clases de PlanItemDefinition.

PlanItemDefinition está especializada en varios conceptos que se especifican secciones posteriores en este documento: EventListener, Milestone, PlanFragment y Tarea.

La clase PlanItemControl especifica los valores predeterminados para los aspectos de control de PlanItemDefinitions, como si estas instancias tienen que completarse antes de que se pueda completar el caso o una etapa del caso que contiene las instancias.

PlanItemControl a diferencia de Etapas y los otros subtipos de PlanItemDefinition, PlanFragments (que no son etapas) no se instanciarán en tiempo de ejecución. PlanItemDefinition tiene los siguientes atributos:

Atributo	Descripción
name : String	El nombre del PlanItemDefinition.
defaultControl: PlanItemControl[0..1]	Elemento que especifica el valor predeterminado para los aspectos de control de PlanItemDefinitions. DefaultControl NO DEBE ser especificado para la etapa al que hace referencia Case como su casePlanModel.

Tabla B-5 Atributos de PlanItemDefinition

6. Task (Tarea)

Una Tarea es una unidad atómica de trabajo. Task es una clase base para todas las tareas en CMMN y hereda de PlanItemDefinition.

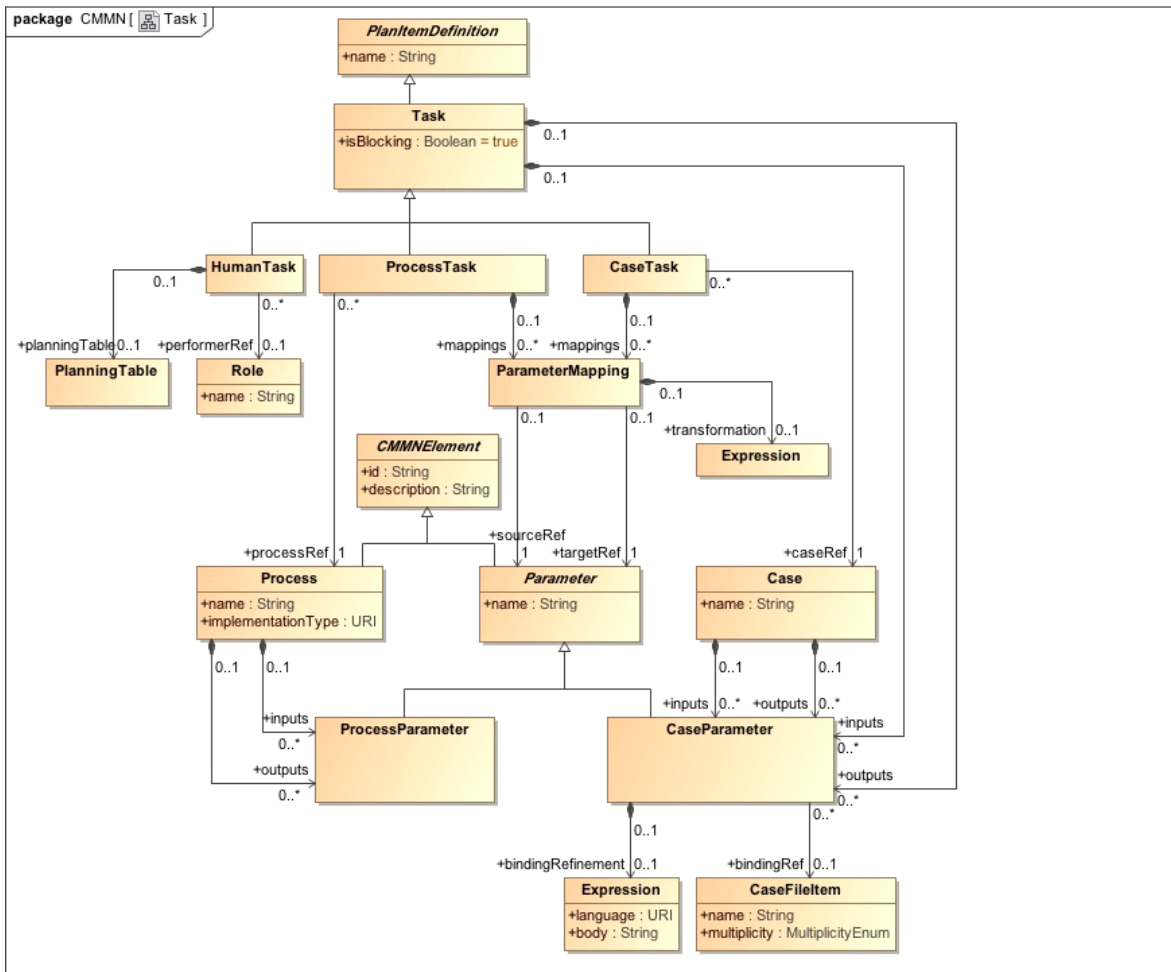


Figura B-6 Diagrama de clases de Tarea.

La clase Task tiene los siguientes atributos:

Atributo	Descripción
isBlocking : Boolean = true	Sí isBlocking está establecido en "true", la tarea está esperando hasta que se complete el trabajo asociado con la tarea. Sí isBlocking está establecido en "false", la tarea no está esperando que el trabajo se complete y se completa inmediatamente, al instanciar.

	El valor predeterminado del atributo isBlocking DEBE ser "true".
	Una tarea que no bloquea (isBlocking establecida en "false") NO DEBE tener salidas.
inputs: CaseParameter [0 .. *]	Cero o más objetos CaseParameter especifican la entrada de la Tarea.
outputs: CaseParameter[0..*]	Cero o más objetos CaseParameter que especifican la salida de la tarea.
planningTable: PlanningTable [0..1]	Opcional. HumanTask se puede utilizar para la planificación y PlanningTable puede contener TableItems que son útiles en el contexto de planificación particular.
processRef: Process[1]	Referencia a un proceso.
caseRef: Case[1]	Referencia al caso que se llama como parte de la CaseTask.

Tabla B-6 Atributos de Tarea

7. Elementos del Evento

En CMMN un evento es algo que "sucede" durante el curso de un Caso. Los eventos pueden desencadenar, por ejemplo, la habilitación, activación y finalización de Etapas y Tareas o el logro de Hitos. Cualquier evento tiene una causa.

CMMN predefine muchos eventos, y sus causas, tales como:

Cualquier cosa que pueda pasar a la información en el CaseFile. Esto se define por "eventos estándar" que indican transiciones en el ciclo de vida definido por CMMN de CaseFileItems.

- Cualquier cosa que pueda suceder a Etapas, Tareas e Hitos. Esto se define por "eventos estándar" que indican transiciones en el ciclo de vida definido por CMMN de éstos.

Sin embargo, el paso del tiempo no se puede capturar a través de estos "eventos estándar". También conducirá a menudo a modelado muy indirecto, cuando cualquier evento de usuario, como la aprobación o rechazo de algo, tiene que ser capturado a través del impacto en los datos en el CaseFile o mediante transiciones en ciclos de vida, por ejemplo, Tareas o Hitos.

Por esta razón, se introduce clase adicional, llamada EventListener, que se especializa en TimerEventListener y UserEventListener. EventListener tiene su propio ciclo de vida predefinido CMMN, de modo que también cualquier lapso de tiempo, así como cualquier evento de usuario, todavía pueden ser capturados como "eventos estándar", denotando transiciones en el ciclo de vida definido por CMMN de EventListener.

EventListener hereda de PlanItemDefinition, de modo que las instancias de EventListeners también pueden ser elementos de los planes de caso.

Esto permitirá a CMMN manejar cualquier evento de manera uniforme, es decir, como "eventos estándar" que denotan transiciones en ciclos de vida definidos por CMMN.

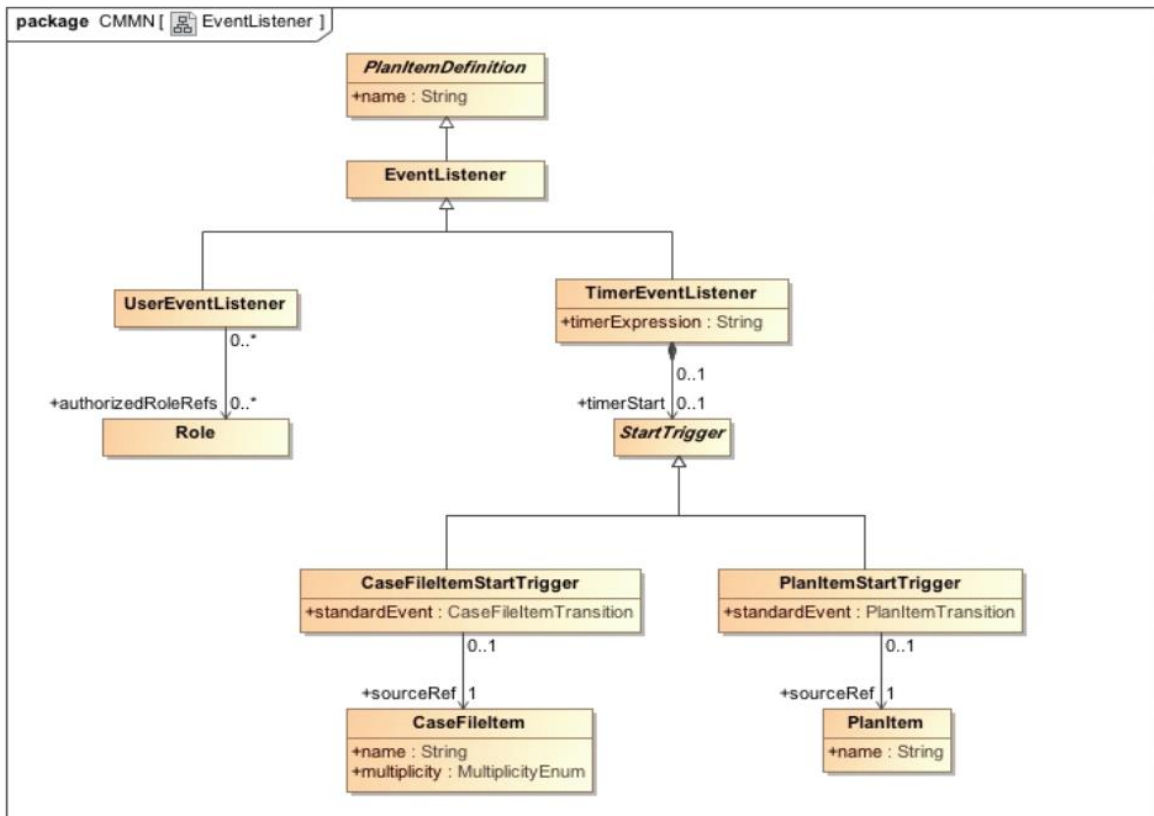


Figura B-7 Diagrama de clases de EventListener.

Un TimerEventListener es un PlanItemDefinition, cuyas instancias se utilizan para capturar intervalos de tiempo predefinidos. Hereda de EventListener. En la tabla siguiente se enumeran los atributos de la clase TimerEventListener:

Atributo	Descripción
TimerExpression: String	Una cadena de expresión que se ajusta al formato ISO-8601 para representaciones de fecha y hora, duración o intervalo.
timerStart : StartTrigger[0..1]	El disparador de inicio de TimerEventListener. Este atributo es opcional. Si se especifica timerStart, entonces en tiempo de ejecución, si el disparador se produce el tiempo de ocurrencia del disparador es capturado y el timerExpression DEBE ser relativo a la timestamp capturada cuando se produce el disparador timerStart.
	Clase abstracta que implementa 2 conceptos:
	- CaseFileItemStartTrigger, - PlanItemStartTrigger.

Tabla B-7 Atributos de EventListener

8. Sentry (Centinela)

Un Sentry mira hacia fuera para que ocurran situaciones importantes (o "acontecimientos"), que influyen en los procedimientos adicionales de un caso (y de ahí su nombre).

Un Sentry es una combinación de un "evento y / o condición". Cuando se recibe el evento, se puede aplicar una condición para evaluar si el evento tiene efecto o no. Los sentries pueden tomar la siguiente forma:

1. Una parte del evento y una parte de la condición en el formulario
On <evento> si <condición>
2. Una parte del evento en el formulario
On <evento>
3. Sólo una parte de la condición en la forma
Si <condición>

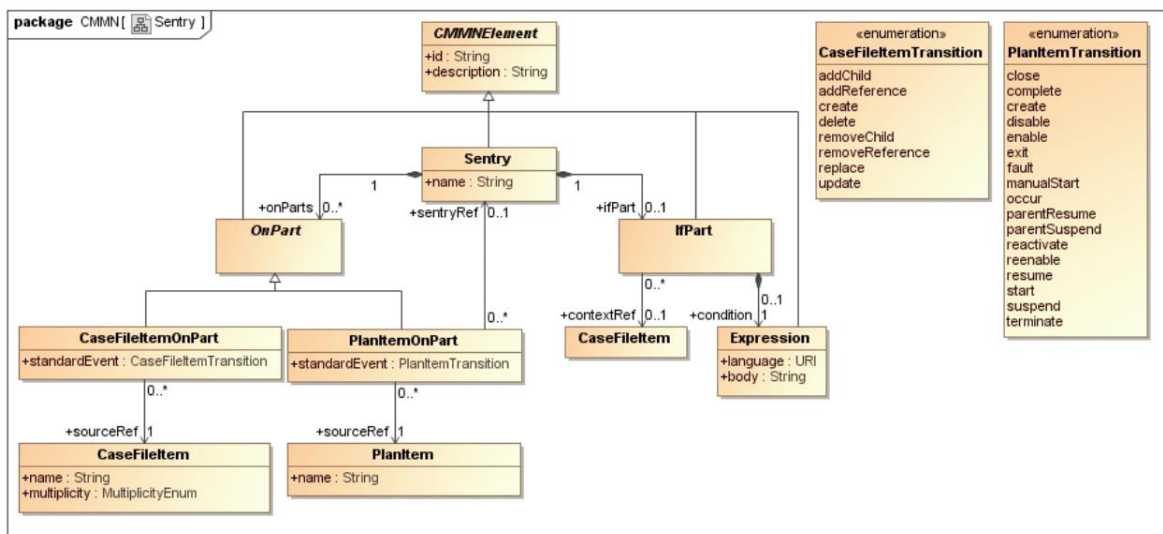


Figura B-8 Diagrama de clases de Sentry.

Se define un conjunto de "eventos estándar" basados en transiciones en ciclos de vida definidos por CMMN que son capaces de capturar cualquier evento que sea relevante en el contexto de un Caso. Esto incluye eventos de temporizador, eventos de información del caso y eventos de usuario.

Un sentry puede consistir en dos partes:

- Cero o más OnParts. Una OnPart especifica el evento que sirve como desencadenador. Cuando se captura el evento, se dice que la OnPart "ocurre".
- Cero o uno IfPart. El IfPart especifica una condición, como Expresión que evalúa sobre el CaseFile. Si todas las OnParts de un Sentry han ocurrido, y su IfPart (si existe) se evalúa como "true", se dice que el Sentry está "satisfecho".

Un Sentry que está satisfecho en realidad desencadena el PlanItem que se refiere a él:

- Cuando el Sentry sea referenciado por uno de los entryCriteriaRefs del PlanItem, el PlanItem (su instancia) transitará, basado en la transición relacionada con los criterios de entrada en su ciclo de vida: se habilitará una tarea o etapa y se logrará un hito.
- Cuando el Sentry sea referenciado por uno de los exitCriteriaRefs del PlanItem, el PlanItem transitará, en función de la transición relacionada con los criterios de salida en su ciclo de vida: se finalizará (saldrá) una Tarea o Etapa.

Sentry hereda de CMMNElement y tiene los siguientes atributos:

Atributo	Descripción
name : String	El nombre del Sentry.
onParts : OnPart[0..*]	Define la OnParts del Sentry. Clase abstracta, especializada en 2 conceptos: - CaseFileItemOnPart - PlanItemOnPart
ifPart : IfPart[0..1]	Define la ifPart del Sentry.
CaseFileItemTransition	sourceRef: CaseFileItem [1]. Referencia a un CaseFileItem sometidos a la transición de estado como se especifica, el OnPart DEBE ocurrir (en tiempo de ejecución).
planItemOnPart	Referencia a una transición de estado en PlanItem (casePlanModel, Stage, Task, EventListener o Hito). A medida que diferentes estados se aplican a diferentes PlanItems. (a) sourceRef: PlanItem[0..1] Referencia a un PlanItem sometido a la transición de estado como se especifica, el OnPart DEBE ocurrir (en tiempo de ejecución). SourceRef representa un PlanItem que DEBE ser contenido por el mismo PlanFragment (o Stage) que también contiene el Sentry que contiene el PlanItemOnPart. (b) sentryRef: Sentry[0..1] Referencia a un sentry. Hace que el PlanItemOnPart del Sentry ocurra cuando el PlanItem que es referenciado por sourceRef transita por la transición de salida en su ciclo de vida, debido a que el Sentry al que hace referencia sentryRef está satisfecho. SentryRef, si se especifica, DEBE referirse a un Sentry que es referenciado por un exitCriteriaRef del PlanItem que se conoce como sourceRef sourceRef del PlanItemOnPart. Cuando se especifica sentryRef, standardEvent DEBE tener el valor "exit".
ifPart	(a) contextRef: CaseFileItem [0..1] CaseFileItem que sirve de punto de partida para la evaluación de la expresión. Si no se especifica, la evaluación comienza en el objeto CaseFile al que hace referencia Case como su caseFileModel. (b) condición: Expresión [1] Debe evaluar a Boolean.

Tabla B-8 Atributos de Sentry

9. Plan Fragment and Stage

Un PlanFragmento es un conjunto de PlanItems, posiblemente dependientes entre sí, y que a menudo ocurren en Planos de Casos en combinación, representando un patrón.

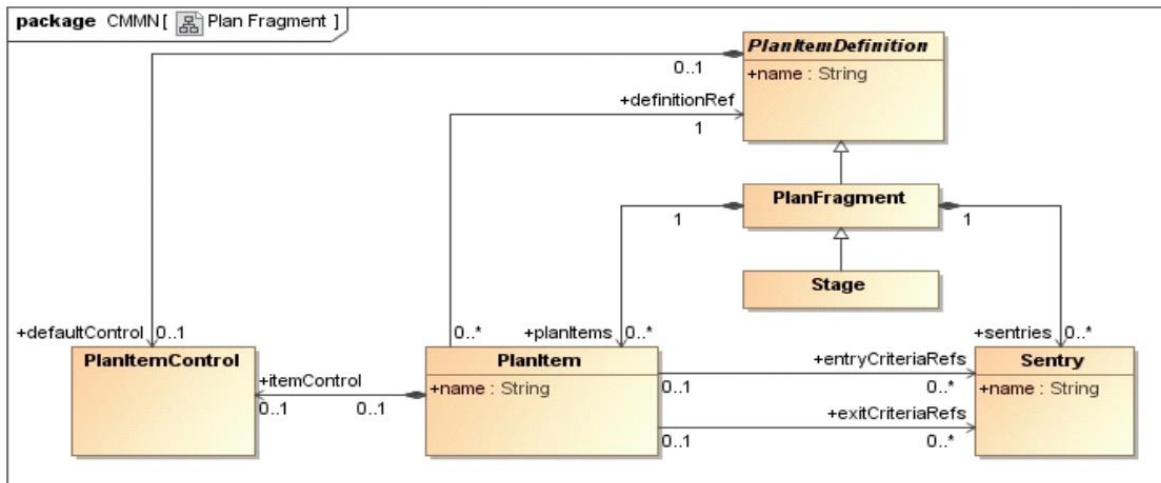


Figura B-9 Diagrama de clases de PlanFragment.

Las dependencias entre PlanItems, en PlanFragmentos, se definen como sentries. Un PlanFragmento es un contenedor de PlanItems y los Sentries que definen los criterios según los cuales los PlanItems están habilitados (o ingresados) y terminados (o salidos).

Ejemplos simples de PlanFragments son:

- Una combinación de dos Tareas, por lo que la finalización de una Tarea satisface al Sentry que permite el inicio del otro.
- Una combinación de un EventListener y una Task, por lo que la ocurrencia del evento satisface el Sentry que habilita el inicio de la Tarea.

La clase del PlanFragment tiene los siguientes atributos:

Atributo	Descripción
planItems : PlanItem[0..*]	Los PlanItems contenidos en el PlanFragmento.
sentries : Sentry[0..*]	The Sentry(ies) contenidos en el PlanFragment.
itemControl: PlanItemControl[0..1]	Si se especifica un objeto PlanItemControl para un PlanItem, debe sobrescribir el objeto PlanItemControl del elemento PlanItemDefinition asociado. De lo contrario, el objeto PlanItemControl especifica el comportamiento del objeto PlanItemControl de su PlanItemDefinition asociado.
definitionRef: PlanItemDefinition[1]	DefinitionRef NO DEBE representar: - La Etapa que es el casoPlanModel del Caso. - Un PlanFragmento que no es una Etapa. DefinitionRef de un PlanItem contenido en una Fase NO DEBE ser esa Fase o cualquier Fase en la que dicha Fase esté anidada.
entryCriteriaRefs: Sentry[0..*]	DEBE referirse a Sentries que están contenidos por la Etapa o PlanFragmento que contiene ese PlanItem. Un PlanItem que es definido por un EventListener NO DEBE tener entryCriteriaRefs.
exitCriteriaRefs: Sentry[0..*]	DEBE referirse a Sentries que están contenidos por la Etapa o PlanFragmento que contiene ese PlanItem.

Un PlanItem que es definido por un EventListener o definido por una tarea que no es bloqueante (isBlocking establecido en "false") NO DEBE tener exitCriteriaRefs.

Tabla B-9 Atributos de PlanFragment

10. Etapa (Stage)

Una etapa hereda de PlanFragment. Como PlanFragment es una PlanItemDefinition también, y sirve como base para los planes de Case (instancia).

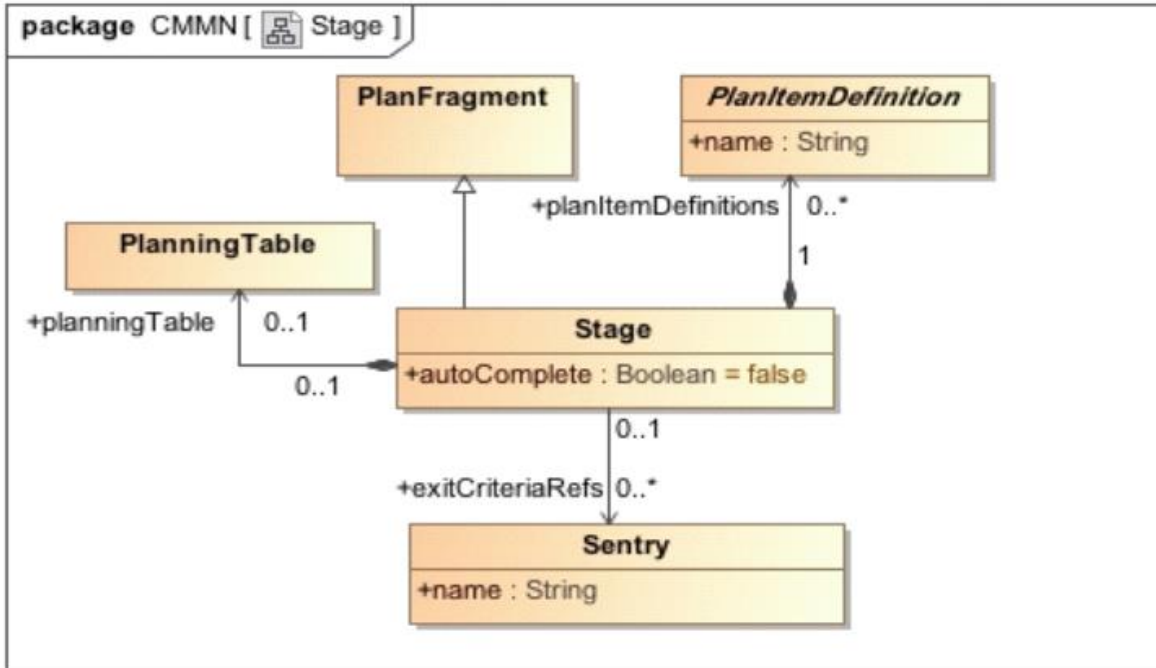


Figura B-10 Diagrama de clases de Stage.

Como PlanFragmento, una Etapa puede contener PlanItems y Sentries.

A diferencia de PlanFragments (que no son etapas), las etapas tienen representaciones en tiempo de ejecución en un caso (instancia) plan. Las instancias de etapas se rastrean a través del ciclo de vida de la etapa definida por CMMN. Las etapas pueden ser consideradas "episodios" de un caso, aunque los modelos de caso permiten definir etapas que se pueden planear en paralelo también.

Lo siguiente es compatible con una etapa, que no es compatible con un PlanFragmento (que no es una etapa):

- Una etapa se puede utilizar como PlanItem dentro de PlanFragmento u otras etapas.
- Una etapa (instancia) puede servir como contexto para la planificación (es decir, una etapa puede tener una tabla de planificación) para apoyar a los usuarios en la planificación de elementos adicionales ("discrecionales") en instancias de la etapa en tiempo de ejecución.
- El Caso se refiere a una Etapa como su casoPlanModel. Esto define la etapa "más externa" del caso.
 - Esta etapa "más externa" también contiene las PlanItemDefinitions que se utilizan en el caso.
 - Esta Etapa "más externa" del Caso también puede contener Centros que sirven como criterio de salida para esa Etapa, y por lo tanto para el caso.

La clase Stage tiene los siguientes atributos:

Atributo	Descripción
planItemDefinitions: PlanItemDefinition[0..*]	Este atributo enumera los objetos PlanItemDefinition disponibles en la etapa y sus etapas agrupadas. PlanItemDefinitions NO DEBE ser contenido por cualquier otra Etapa que el casePlanningModel del Caso.
autoComplete: Boolean = false	Este atributo controla la finalización de la etapa. Si es "falso", una etapa requiere que un usuario la complete manualmente, lo que a menudo es apropiado para etapas que contienen elementos "discrecionales" y / o Tareas o etapas no requeridas.
planningTable: PlanningTable[0..1]	Define el PlanningTable (opcional) de la Etapa.
exitCriteriaRefs: Sentry[0..*]	Referencia a cero o más Sentries que sirven como criterio de salida para la Etapa. ExitCriteriaRefs de una etapa DEBE referirse a los Sentries que están contenidos por esa etapa. Sólo la Etapa a la que hace referencia el caso como su casePlanningModel puede tener exitCriteriaRefs. Tenga en cuenta que sólo es útil que la etapa tenga directamente exitCriteriaRefs, ya que no puede anidarse más en otras Etapas (otras Etapas pueden contener ambos PlanItems que representan Etapas y los Sentries que les imponen criterios de entrada y / o salida).

Tabla B-10 Atributos de Stage

11. Tabla de Planificación (Planning Table)

La planificación es un esfuerzo en tiempo de ejecución. Un PlanningTable define el ámbito de la planificación, en términos de la identificación de un subconjunto de PlanItemDefinitions que se pueden considerar para la planificación en un determinado contexto. El contexto para la planificación podría ser:

- Una etapa. Cuando una etapa tiene una tabla PlanningTable, se puede utilizar PlanningTable para una instancia de esa etapa para planificar instancias de tareas y etapas en esa instancia de Stage.
- Una tarea humana. Cuando un HumanTask tiene un PlanningTable (véase 3.3.2.10), PlanningTable puede utilizarse, para una instancia de ese HumanTask, para planificar instancias de Tasks and Stages en la instancia de la etapa que contiene esa instancia del HumanTask.

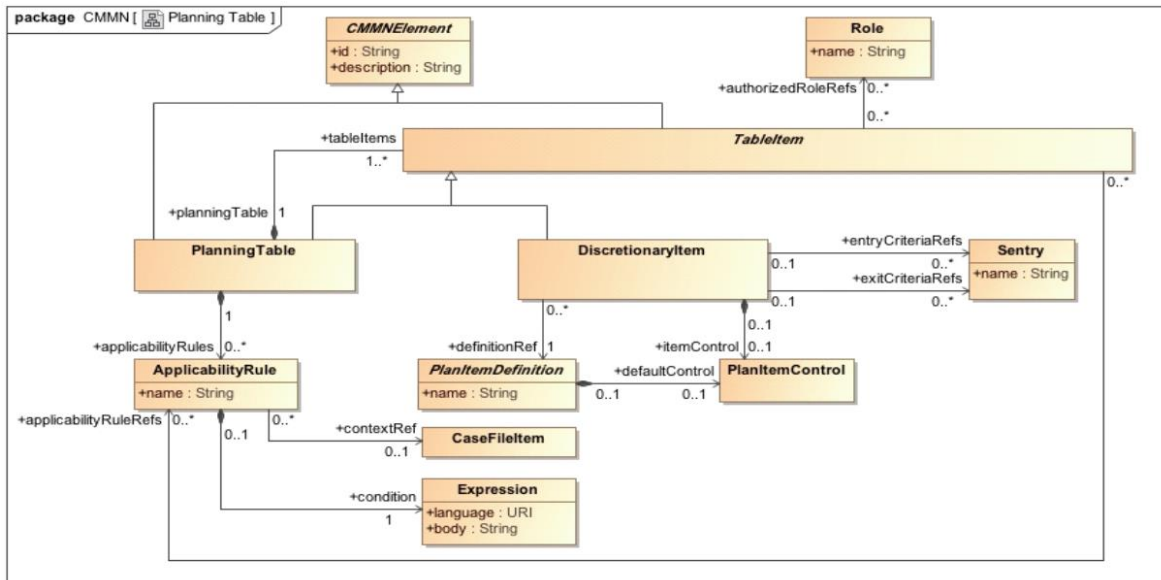


Figura B-11 Diagrama de clases de PlanningTable.

Las instancias de Tareas y Etapas que están definidas por el mismo PlanItemDefinition pueden planificarse basándose en posiblemente múltiples PlanningTables. Esto requirió una clase separada, DiscretionaryItem, que hace referencia a PlanItemDefinition. Multiple DiscretionaryItems puede referirse a la misma PlanItemDefinition. Un PlanItemDefinition se (re) utiliza en PlanningTables múltiples cuando estas PlanningTables contienen DiscretionaryItems que se refieren a o ("use") ese mismo PlanItemDefinition.

Por conveniencia, un DiscretionaryItem que se refiere a un PlanItemDefinition que es una tarea, podría llamarse una "Tarea discrecional". De manera similar podemos considerar "PlanFragments discretcionales" y "Etapas discretcionales". Note que PlanFragments que no son Etapas sólo pueden ser "discretcionales" ya que PlanItems no puede referirse a ellos. Observe de nuevo que cuando se usa un PlanFragmento (que no es una etapa) para la planificación, sólo los PlanItems que están contenidos en él se instancian y tienen sus ciclos de vida que se rastrean. El PlanFragmento (que no es una etapa) no se instancia en sí mismo.

Para mayor comodidad durante la planificación en tiempo de ejecución, en situaciones en las que PlanningTable podría contener potencialmente muchos DiscretionaryItems, es posible definir una PlanningTable recursivamente: PlanningTable que contiene otras PlanningTables. Se dice que los usuarios (trabajadores de caso) "planifican" (en tiempo de ejecución), cuando seleccionan DiscretionaryItems de PlanningTable y mueven instancias de PlanItemDefinitions asociadas al plan del caso (instance).

Es posible autorizar Roles para la planificación de ciertos DiscretionaryItems y sub-PlanningTables. También es posible hacer que DiscretionaryItems (y sub-PlanningTables) sean dinámicamente aplicables para la planificación, sobre la base de condiciones que evalúen sobre el CaseFile. Tanto las Autorizaciones de Roles como las Reglas de Aplicabilidad pueden controlar dinámicamente lo que los Elementos Discretcionales, posiblemente organizados a través de Sub-PlanningTables, están expuestos a los Trabajadores de Casos que están involucrados en la planificación. PlanningTable hereda de CMMNElement y tiene los siguientes atributos:

Atributo	Descripción
tableItems: TableItem[1..*]	<p>Una lista de objetos de TableItem disponible para la planificación. Un PlanningTable se dice que está "anidado" en otro PlanningTable, cuando PlanningTable es un TableItem que está contenido por ese otro PlanningTable directamente o recursivamente a través de incluso PlanningTables otros.</p> <p>El conjunto de tableItems de PlanningTable NO DEBE incluir ese PlanningTable o cualquier PlanningTable en el que PlanningTable esté anidado. Un PlanningTable DEBE contener al menos un TableItem.</p>
applicabilityRules: ApplicabilityRule[0..*]	<p>Cero o más Objetos de regla de aplicabilidad.</p> <p>Si la condición del objeto ApplicabilityRule se evalúa como "true", entonces el TableItem es aplicable para la planificación, de lo contrario no lo es. Si no se aplica una regla de aplicabilidad a un objeto de tabla, su aplicabilidad se considera "verdadera". Un PlanningTable que contiene un TableItem DEBE contener las ApplicabilityRules que representan el applicabilityRuleRefs de ese TableItem.</p>
authorisedRoleRefs: Role[0..*]	Referencias a cero o más Objetos de función que están autorizados a planificar basados en el TableItem.
planItemDefinitions: PlanItemDefinition [1]	Representa una tarea o un PlanFragmento (o una etapa).
itemControl: PlanItemControl[0..1]	Opcional. Sobrescribe el valor del atributo defaultControl de la PlanItemDefinition asociada a DiscretionaryItem.
entryCriteriaRefs: Sentry [0 .. *]	Referencia a cero o más Sentries que representan los criterios de entrada.
exitCriteriaRefs: Sentry [0 .. *]	Referencia a cero o más Sentries que representan los criterios de salida. Un DiscretionaryItem que se define por una tarea que no es bloqueante (isBlocking establecido en "false") NO DEBE tener exitCriteriaRefs.

Tabla B-11 Atributos de PlanningTable

12. Descripción y definición de los elementos del modelo de ACM

Elemento del Modelo ACM	Descripción
applicabilityRule	Se utiliza para especificar si un TableItem es "aplicable" ("elegible", "disponible") para la planificación, en base a las condiciones que se evalúan sobre la información en el CaseFile.
caseFileItemOnPart	Se hereda de OnPart.
caseParameter	Se utiliza para modelar las entradas y salidas de Casos y Tasks.
caseTask	Se puede utilizar para llamar a otro caso. Un CaseTask activa la creación de una instancia de ese otro Caso, cuya creación denota la transición inicial en el ciclo de vida definido por CMMN de una instancia de Casos.
discretionaryItem	Identifica una PlanItemDefinition, de la cual las instancias pueden ser planificadas, a la "discreción" de un trabajador

	<p>de caso que está involucrado en la planificación, que las instancias se planifican en el contexto que está implícito en el PlanningTable que contiene el DiscretionaryItem, ya sea directamente o a través de un 'Anidado' PlanningTable. Las condiciones para que un PlanItemDefinition o Stage sean "discrecionales" se explican en la norma.</p>
eventListener	<p>Un evento es algo que "sucede" durante el curso de un Caso. Distinción se hace entre "eventos estándar" que denotan transiciones en el ciclo de vida definido por CMMN de CaseFileItems y transiciones en el ciclo de vida definido por CMMN de Stage, Task y Miletones.</p>
expression	<p>Especifica objetos de cadena que se evalúan sobre la información en el CaseFile.</p>
humanTask	<p>Unidad de trabajo realizada por un trabajador del caso.</p>
ifPart	<p>Se utiliza para especificar una condición (opcional).</p>
manualActivationRule	<p>Especifica en qué condiciones Task y Stage, una vez activadas, se inician manual o automáticamente. Determina si la instancia de Task o Stage debe pasar al estado Enabled o Active. Esta regla se evalúa cuando se satisface uno de los criterios de entrada de la task o stage. Si esta regla se evalúa como "true", la instancia Task o Stage pasa de Available a Enabled, de lo contrario cambia de Available a Active. Esta regla impacta las instancias de etapa o tarea en estado Disponible.</p>
mileStone	<p>Representa un objetivo alcanzable, definido para permitir la evaluación del progreso de un caso.</p>
onPart	<p>Aborda el aspecto "evento" de un Sentry.</p>
parameterMapping	<p>Se utiliza para la asignación de entrada / salida de CaseTasks y ProcessTasks.</p>
planFragment	<p>Elemento de raíz para PlanItems, posiblemente dependientes entre sí, que debe ir al plan como una unidad. P.ej. Una combinación de dos Tareas, por lo que la finalización de una Tarea satisface al Sentry que habilita el inicio del otro o una combinación de un EventListener y una Tarea, por lo que la ocurrencia del evento satisface el Sentry que habilita el inicio de la Tarea.</p>
planItem	<p>Una instancia planItem es un uso de un elemento PlanItemDefinition en un PlanFrama (o Stage). La misma PlanItemDefinition podría ser (re) usada varias veces como parte de diferentes combinaciones, es decir, como parte de diferentes PlanFragments (o Stages). Por lo tanto, una PlanItemDefinition (por ejemplo, una tarea o EventListener) se define una vez, y puede ser (re) utilizado en múltiples PlanFragments. Esto requirió una clase separada, PlanItem, que hace referencia a PlanItemDefintion. Multiple PlanItems puede referirse a la misma PlanItemDefinition. A PlanItemDefinition se (re)utiliza en varios PlanFragments (o Stages) cuando estos PlanFragments (o Stages) contienen PlanItems que se refieren a o ("use") que PlanItemDefinition mismo.</p>
planItemControl	<p>Define aspectos del control de instancias de Task, Stages, EventListeners y MileStones. Se definen en relación con</p>

	<p>sus "orígenes" en el modelo - PlanItems y DiscretionaryItems - y pueden ser predeterminados.</p>
planItemDefinition	<p>Es una clase abstracta. Define los bloques de construcción de los cuales se interpretan los planes de caso. Se especializa en varios conceptos como PlanFragment, Task, EventListener y MileStone.</p>
planItemOnPart	<p>Se hereda de OnPart.</p>
planningTable	<p>Define el ámbito de la planificación en tiempo de ejecución, en términos de un subconjunto de PlanItemDefinitions que se pueden considerar para la planificación en un determinado contexto. El contexto para la planificación podría ser (1) una stage (incluyendo stages y tasks) o (2) una HumanTask.</p> <p>Tanto las Autorizaciones de Roles como las Reglas de Aplicabilidad pueden controlar dinámicamente lo que los Elementos Discrecionales, posiblemente organizados a través de Sub-PlanningTables, están expuestos a los Trabajadores de Casos involucrados en la planificación.</p>
process	<p>Abstracción de procesos como se especifican en varias especificaciones de modelado de procesos.</p>
processParameter	<p>Parámetro de un proceso</p>
processTask	<p>Se puede utilizar en el caso para llamar a un proceso de negocio</p>
repetitionRule	<p>Especifica en qué condiciones (Task) Tareas, (Stages) Etapas e Hitos (Milestones) tienen repeticiones. Cada repetición es una nueva instancia de la misma. El desencadenante para la repetición es un Sentry, que se hace referencia como criterio de entrada, siendo satisfecho, por lo que se produce una OnPart de ese Sentry. Por ejemplo: Una task puede repetirse cada vez que se crea un determinado documento. La Task (como PlanItem) puede tener un criterio de entrada, refiriéndose a un Sentry, tener en OnPart por el que OnPart se refiere al CaseFileItem que representa el tipo de documento, y por lo que el estándar de la OnPart se especifica como "crear". Cuando la Regla de repetición contenida en PlanItemControl de la task (como PlanItem) también se evalúa como "true", la task se repite al crear el documento.</p> <p>Esta regla DEBE ser evaluada cuando la instancia Milestone, Stage o Task es instanciada y transiciones al estado Available, y su valor booleano DEBE ser mantenido durante el resto de la vida de la instancia Milestone, State o Task.</p> <p>Las instancias de la stage y de la task con una regla de repetición que evalúa a "verdadero" crearán una instancia cada vez que un criterio de la entrada con un onPart es Satisfecho Bajo esta condición se crea una nueva instancia y debido a que los criterios de entrada se cumplen, se mueve desde el estado Disponible al estado Activo o Habilitado dependiendo de la Regla ManualActivation.</p>

	<p>EventListeners no puede tener RepetitionRule. Sin embargo, para una repetición TimerEventListener se puede definir a través de un timerExpression basado en ISO-8601, definiendo intervalos repetidos en él usando la notación "R <n> /".</p>
requiredRule	<p>Especifica en qué condiciones "Obligatorio" se completarán o terminarán las Task, Stages y Milestone.</p> <p>RequiredRule determina si la instancia Milestone, Stage o Task que tiene esta condición DEBE estar en el estado Completed, Terminated, Failed o Disabled para que su instancia Stage primaria pase al estado Completed. Esta regla DEBE ser evaluada cuando la instancia Milestone, Stage o Task es instanciada y transiciones al estado Available, y su valor booleano DEBE mantenerse durante el resto de la vida de la instancia Milestone, Stage o Task.</p> <p>Si esta regla no está presente, entonces se considera "falso". Si esta regla se evalúa como "true", la instancia de Stage primaria NO DEBE pasar al estado Complete a menos que esta instancia de Milestone, Stage o Task esté en el estado Completed, Terminated, Failed o Disabled. Esta regla impacta las instancias de Stage en estado Disponible.</p>
Sentry (Centinela)	<p>"Observa" para que ocurran situaciones (o "acontecimientos") importantes, que influyen en los procedimientos adicionales de un Caso.</p> <p>Es una combinación de un "evento y / o condición". Un Sentry toma una de las siguientes formas:</p> <ol style="list-style-type: none"> 1) on <evento> si <condición> 2) on <evento> 3) si <condición> <p>Un centinela puede consistir en dos partes:</p> <ul style="list-style-type: none"> - cero o más OnParts especificando el evento que sirve como un disparador. Cuando se captura el evento, se dice que el OnPart "ocurre". - cero o más IfParts que especifica una condición como Expression que evalúa sobre CaseFile. <p>Un Sentry DEBE tener una IfPart o al menos una OnPart.</p> <p>Si todas las OnParts de un Sentry han ocurrido, y sus IfParts (si existen) se evalúan como "true", se dice que el Sentry está "satisfecho".</p> <p>Un Sentry que está satisfecho en realidad desencadena el PlanItem que se refiere a él.</p>
Stage (Etapa)	<p>Las stages pueden ser consideradas "episodios" de un caso, aunque los modelos de caso permiten definir etapas que se pueden planear en paralelo también. A diferencia de PlanFragments (que no son stages), las stages tienen</p>

	<p>representaciones en tiempo de ejecución en un caso (instancia) plan.</p> <p>1) Es un PlanItemDefintion también. Se puede utilizar como PlanItem dentro de Planfragmentos u otras stages.</p> <p>2) Representa una stage en el modelo de caso y consta de cero o un PlanningTable. Una stage (instancia) puede servir como contexto para la planificación para apoyar a los usuarios en la planificación de elementos adicionales ("discrecionales") en instancias de la stage en tiempo de ejecución.</p> <p>3) El Caso se refiere a una Stage como su casoPlanModel. Esto define la stage "más externa" del caso. Esta stage "más externa" también contiene las PlanItemDefinitions que se utilizan en el caso. Esta Stage "más externa" del Caso también puede contener Sentries que sirven como criterios de salida para esa Stage y, por tanto, para el Caso.</p>
tableItem	Puede ser un DiscretionaryItem o un PlanningTable
timerEventListener	Las instancias se utilizan para capturar intervalos de tiempo predefinidos.

Tabla B-12 Definición de elementos CMMN

Apéndice C – Guía para iniciar Camunda con CMMN 1.1

Este apéndice sirve de guía para la instalación, configuración y puesta en marcha del marco de trabajo del motor de ejecución BPM Camunda y el modelador Camunda para el lenguaje CMMN 1.1 así como el desarrollo de una primera aplicación [34].

1. Descarga e instalación.

Primero se configura el entorno de desarrollo e instala la plataforma de BPM Camunda, el mismo motor es utilizado para ejecutar modelos BPMN, DMN y CMMN, además se debe instalar y configurar el modelador Camunda.

- Requisitos previos instalados.

- Sistema Operativo Windows 7 Home o superior.
- Java JDK 1.7+ y configuración de la variable de entorno Java_Home, para esto se puede utilizar la guía oficial del sitio:
<https://www.java.com/en/download/help/path.xml>
- Un navegador web actualizado (Firefox, Chrome o Internet Explorer 9+).
- Entorno de desarrollo integrado de Eclipse (IDE) con Maven desde <https://www.eclipse.org/downloads/>

- Plataforma BPM Camunda.

Primero, descargue una distribución de la plataforma BPM Camunda. Puede elegir entre diferentes distribuciones para varios servidores de aplicaciones. En esta guía, se utiliza la distribución basada en Apache Tomcat, la cual puede descargar desde el sitio <https://camunda.org/download/> versión 7.7.0.

Después de descargar la distribución, se descomprime dentro de una carpeta elegida. Se renombra la carpeta con `$CAMUNDA_HOME`. A continuación, ingresamos a la carpeta y se ejecuta el script llamado `start-camunda.bat` (para usuarios de Windows).

Este script iniciará el servidor de aplicaciones y abrirá una pantalla de bienvenida en el navegador web. Si la página no se abre, se ingresa la siguiente dirección en el navegador web <http://localhost:8080/camunda-welcome/index.html>.

- Modelador Camunda.

Para diseñar modelos con el lenguaje CMMN utilizamos el modelador Camunda versión escritorio, existe la versión web, sin embargo, se encontraron detalles en su utilización las cuales son descritas en este trabajo (sección 5.3.6 Observaciones). La descarga e instalación del modelador Camunda versión escritorio se puede realizar desde el sitio oficial <https://camunda.org/download/modeler/>, en esta ocasión se utiliza la versión 1.10.0

2. Configurar un Proyecto Java para CMMN 1.1

Una vez que se tiene instalado entorno de desarrollo integrado de Eclipse (IDE) con Maven y configurada la variable de entorno de Java_Home en el sistema operativo Windows se procede a la creación del proyecto.

- Crear un nuevo proyecto de Maven en Eclipse

Se abre el entorno Eclipse, posteriormente damos clic en el menú *File / New / Other ...* esto abre el Asistente para un nuevo proyecto. En el Asistente para proyectos nuevos, se selecciona la carpeta *Maven* y se elige la opción *Maven Project*. Se hace clic en *Next*.

En la primera página del Asistente para proyectos de New Maven Project, se crea un proyecto sencillo para ello se selecciona *Create a simple project (skip archetype selection)*, posteriormente se da clic en *Next*.

En la segunda página del Asistente para proyectos New Maven Project se ingresan los datos con se indican en la Figura C-1, los datos de los campos *Group Id* y *Artifact Id* pueden ser modificados para crear otros proyectos.

Figura C-1 Asistente para proyectos nuevos Maven.

Cuando se complementa la información, se da clic en *Finish*. Eclipse establece un nuevo proyecto Maven. El proyecto aparece en la vista *Package Explorer* de proyectos del entorno Eclipse.

- Dependencias de Camunda Maven

El siguiente paso consiste en configurar las dependencias de Maven para la nueva aplicación. Para ello en la vista *Package Explorer* se da doble clic en el archivo *pom.xml*, a continuación, en la parte inferior damos clic en la pestaña *pom.xml* y se agregan las dependencias del motor de camunda, el contenido del archivo deberá ser el siguiente:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.camunda.bpm.getstarted</groupId>
  <artifactId>loan-approval-cmmn</artifactId>
  <version>0.1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.camunda.bpm</groupId>
        <artifactId>camunda-bom</artifactId>
        <version>7.7.0</version>
        <scope>import</scope>
        <type>pom</type>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.camunda.bpm</groupId>
      <artifactId>camunda-engine</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.3</version>
        <configuration>
          <failOnMissingWebXml>>false</failOnMissingWebXml>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Ahora puede realizar la primera compilación. Se selecciona el archivo *pom.xml* en el *Package Explorer*, se da clic derecho y se selecciona la opción *Run As / Maven Install*.

- El Descriptor del despliegue

El último paso para configurar la aplicación es agregar el archivo de despliegue *processes.xml* en la ruta *src/main/resources/META-INF* del proyecto Maven. Este archivo permite proporcionar una configuración declarativa del despliegue. El archivo *processes.xml* queda con el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8" ?>
<process-application
  xmlns="http://www.camunda.org/schema/1.0/ProcessApplication"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <process-archive name="Loan-approval-cmmn">
    <process-engine>default</process-engine>
    <properties>
      <property name="isDeleteUponUndeploy">false</property>
      <property name="isScanForProcessDefinitions">true</property>
    </properties>
  </process-archive>
</process-application>
```

En este punto, se ha configurado correctamente la aplicación y se puede comenzar a crear el primer caso CMMN.

3. Creación de un nuevo caso CMMN

En esta sección se crea un caso CMMN 1.1. se inicia el modelador de Camunda.

- Crear un nuevo diagrama CMMN 1.1

Se selecciona *File / New File / CMMN Diagram* en el menú de nivel superior del Modelador de Camunda, como se indica en la Figura C-2.

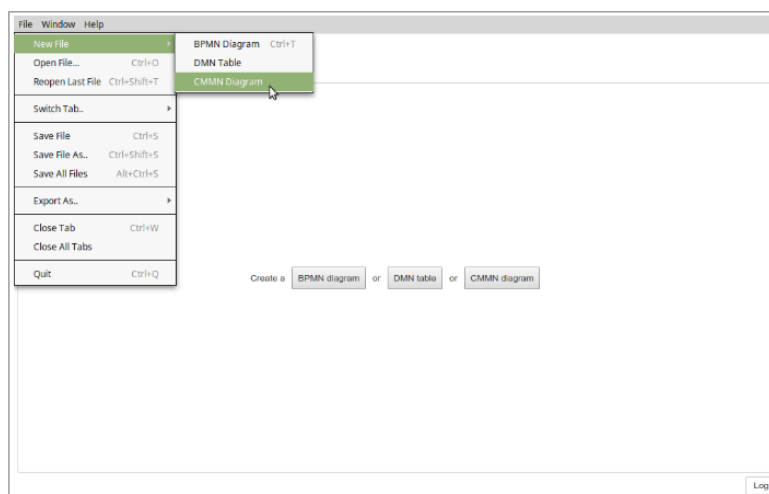


Figura C-2 Nuevo archivo CMMN del Modelador de Camunda.

El Modelador de Camunda crea un nuevo diagrama CMMN que contiene un modelo de plan de caso, la parte esencial de cualquier definición del caso CMMN. Observamos que una tarea sencilla se crea dentro del modelo de plan de caso.

- Definir el ID de caso

Puesto que se está modelando un caso ejecutable, se debe asignar un Id el cual es utilizado por el motor de proceso como identificador para el caso. Es recomendable asignar un nombre legible que tenga relación con la actividad del elemento. Se da clic en el modelo de plan de caso, posteriormente al lado derecho se tiene el panel de propiedades. En esta ocasión la identificación del caso se hace con la etiqueta *loan_application* en el campo *Case Id* debido a que el ejemplo trata acerca la Solicitud de un préstamo, así se muestra en la Figura C-3.

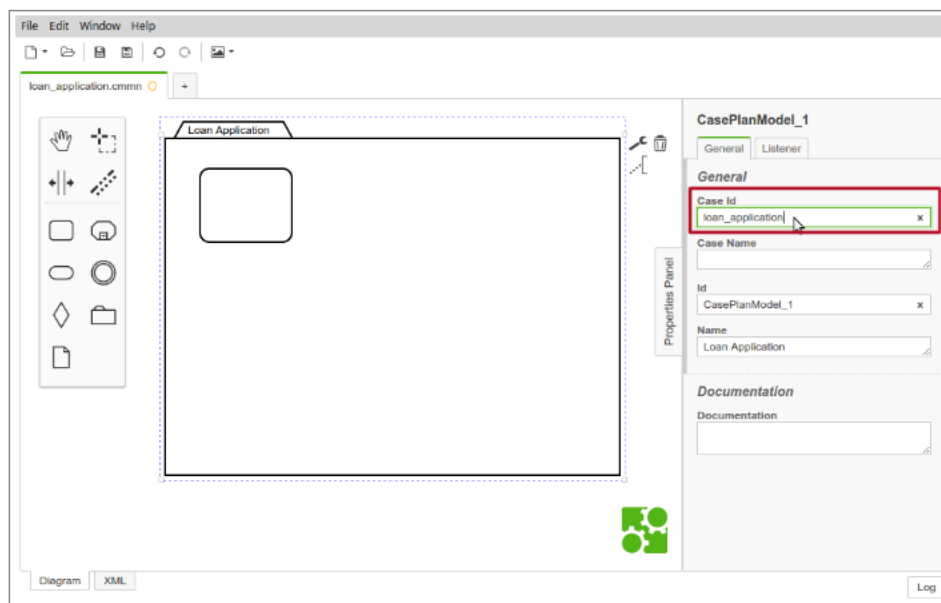


Figura C-3 Definir el Id del Caso.

Opcionalmente se puede cambiar el nombre del modelo de plan de caso. En el campo *Name*. Cuando se realiza esto el nuevo nombre también es visible en el diagrama en el pestillo superior del modelo de plan de caso.

- Añadir tareas humanas

Como primera parte de este caso, la solicitud del préstamo debe ser revisada por una persona por cualquier error que contenga la forma. Por lo tanto, necesitamos una tarea humana en este caso. Primero, en panel de notaciones ubicado en la parte izquierda, se da clic en el icono de la tarea sencilla (véase la notación de la sección 3.3.2.3), luego en el icono de llave inglesa junto a él. Se selecciona la *Human Task* para cambiar el tipo de la tarea. En la Figura C-4 se explica este paso.

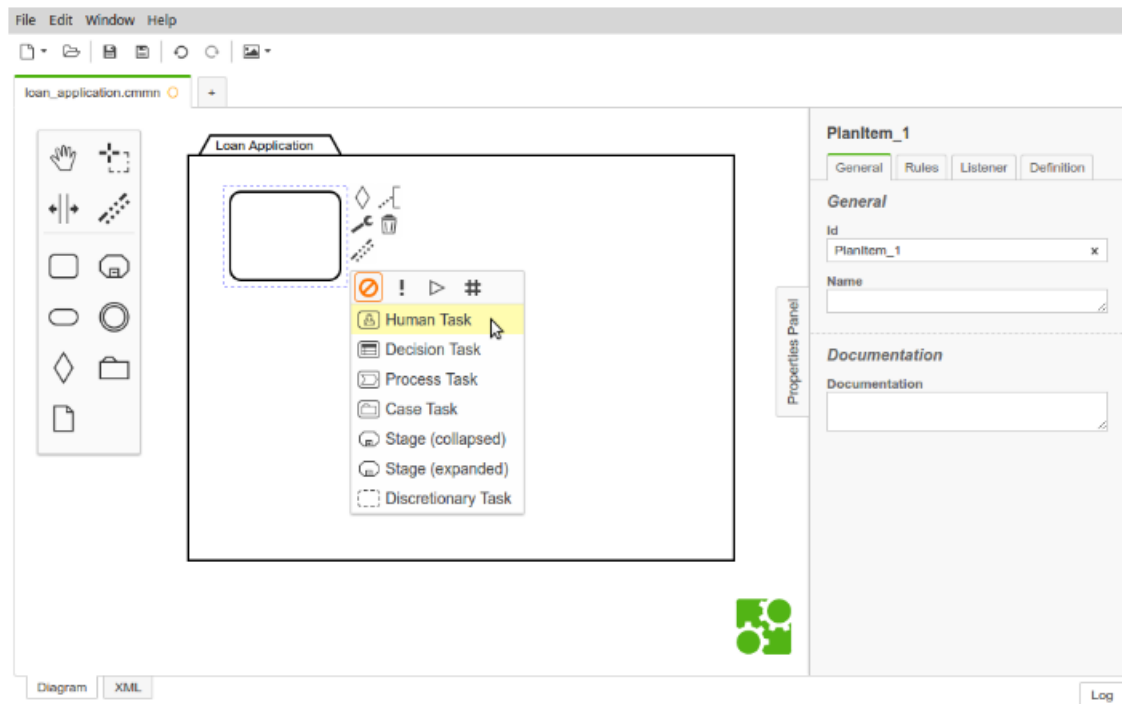




Figura C-4 Añadir una tarea al caso.

A continuación, se da doble clic en la tarea humana para editar el nombre, se escribe *Check Application* y se puede pulsar Enter o dar clic fuera de la tarea.

Además, se debe evaluar la solvencia del cliente. Agregar una segunda tarea dentro del modelo del plan del caso. Haga doble clic en él y asigne el nombre *Provide Customer Rating*.

- **Regla de activación manual**

El siguiente paso es definir que las tareas humanas no necesitan activación manual. Primero, haga clic en la tarea *Check Application*, luego nuevamente en el icono de la llave inglesa  y posteriormente en el icono de *Manual Activation Rule* (Regla de activación manual) .

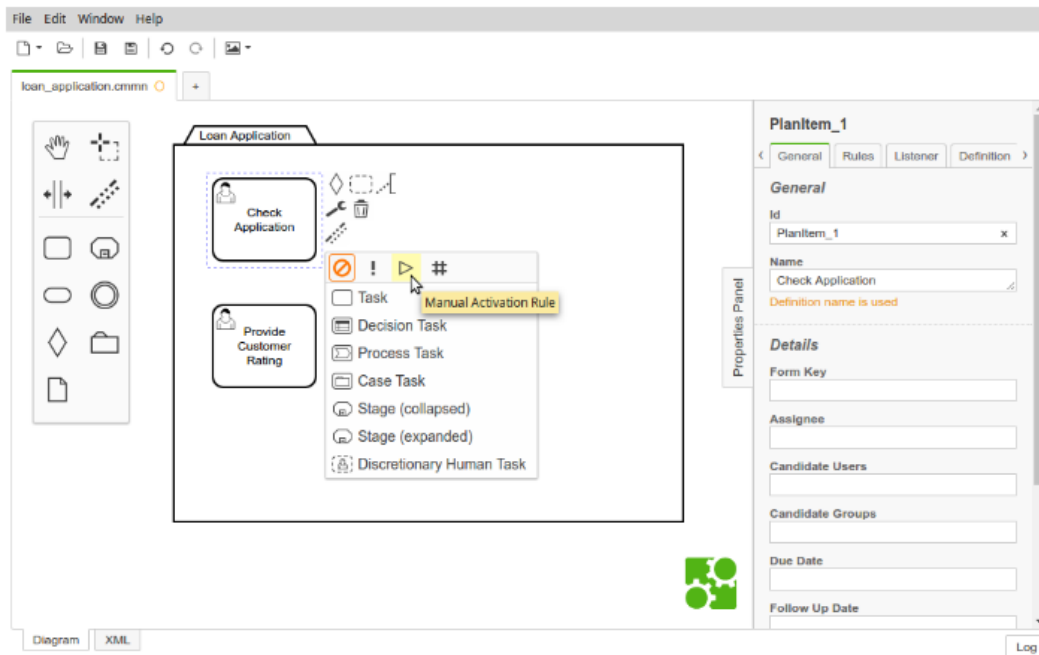


Figura C-5 Activación manual.

La regla de activación manual está activa. Se observa que se ha colocado un marcador en la tarea humana. Ahora se va a deshabilitar la activación manual y habilitar la activación automática. Para ello, se va al panel de propiedades y se da clic en la pestaña *Rules* en la parte superior del panel. A continuación, se define la condición `#{false}` en el campo *Manual Activation Rule*. Ver Figura C-6.

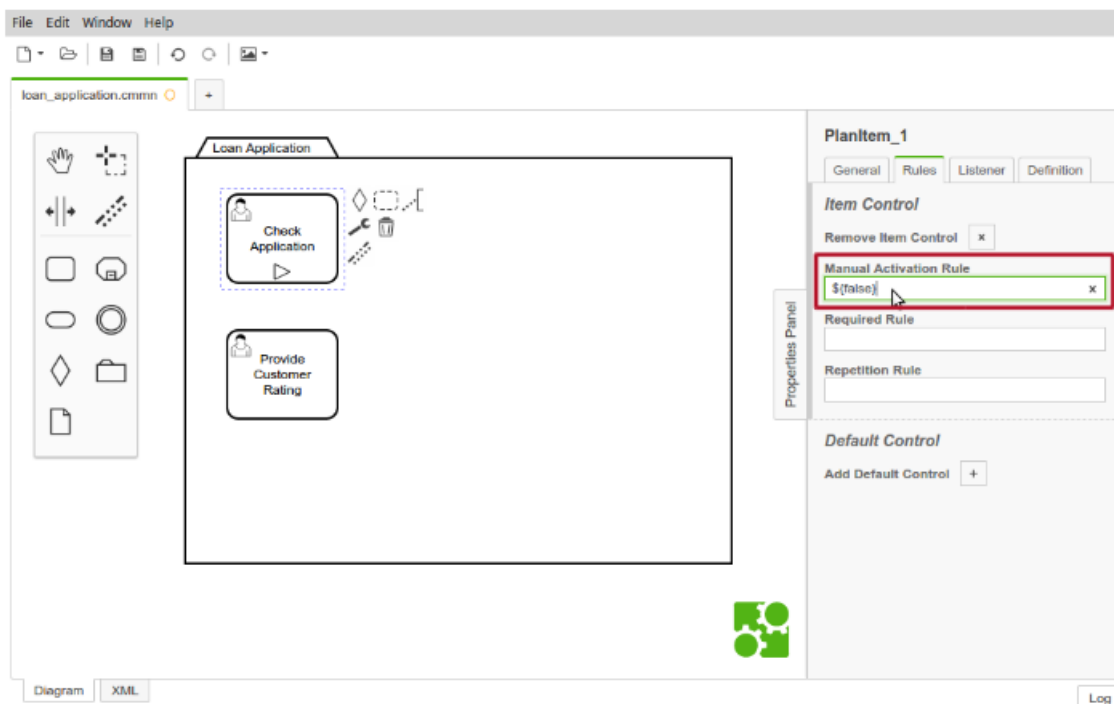


Figura C-6 Activación automática.

Se repite estos pasos para que la tarea *Provider Customer Rating* también active y configure la regla de activación.

- Asignación de usuario

En esta etapa se busca que ambas tareas se asignen a la demo del usuario, para que aparezcan más adelante en la sección de *My Tasks* (Mis tareas) de la Lista de tareas del motor de Camunda. Para asignar la demo del usuario a una tarea, se da clic en ella y se escribe *demo* en el campo de entrada *Assignee* en el panel de propiedades. Ver Figura C-7.

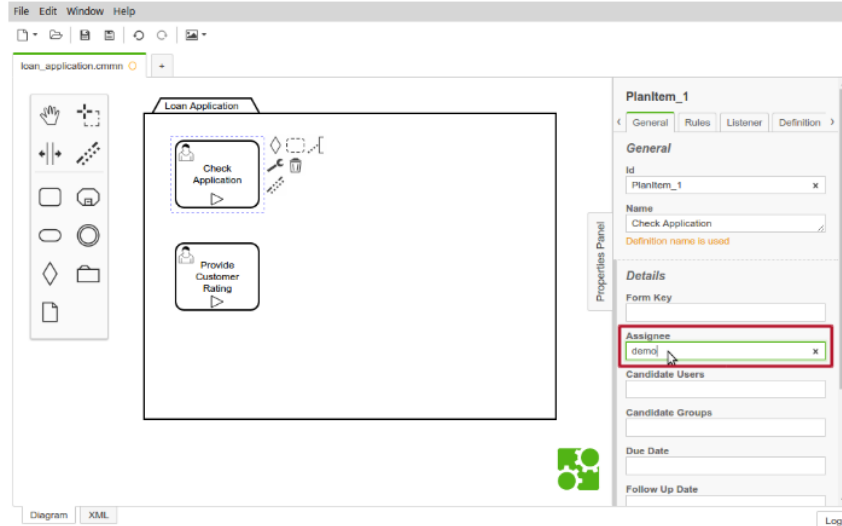


Figura C-7 Asignación de usuario.

Tenga en cuenta que no existe una relación directa entre los dos ítems del plan. No hay flujo de secuencia que conecte las dos tareas como en BPMN. En CMMN, esto expresa que las tareas se pueden ejecutar simultáneamente.

- Guardar el diagrama CMMN 1.1

Cuando haya terminado, guarde los cambios haciendo clic en *File > Save File as ...* En el cuadro de diálogo que aparece, se va al directorio del proyecto de Loan Application (de forma predeterminada, en la ruta del espacio de trabajo de Eclipse). En el directorio del proyecto, coloque el modelo en la carpeta *src/main/resources*.

Vuelve a Eclipse. Haga clic con el botón secundario en la carpeta del proyecto y haga clic en *Refresh*. Esto sincroniza el nuevo archivo CMMN con Eclipse.

4. Instanciar y probar un caso CMMN 1.1

En este paso, se utiliza código Java para instanciar la definición del caso, luego implementar y probar la aplicación.

- Código Java

Para crear directamente una instancia del caso después de la implementación, se agrega el siguiente método a la clase *LoanApprovalApplication* que se ubica en la ruta *src/main/java* y del paquete *org.camunda.bpm.getstarted.cmmn.loanapproval*:

```

package org.camunda.bpm.getstarted.cmmn.loanapproval;
import org.camunda.bpm.application.PostDeploy;
import org.camunda.bpm.application.ProcessApplication;
import org.camunda.bpm.application.impl.ServletProcessApplication;
import org.camunda.bpm.engine.CaseService;
import org.camunda.bpm.engine.ProcessEngine;
import org.camunda.bpm.engine.variable.Variables;

@ProcessApplication("Loan Approval App CMMN")
public class LoanApprovalApplication extends ServletProcessApplication {
    @PostDeploy
    public void startCaseInstance(ProcessEngine processEngine) {
        CaseService caseService = processEngine.getCaseService();
        caseService.createCaseInstanceByKey("loan_application", Variables.createVariables()
            .putValue("applicationSufficient", Variables.booleanValue(null))
            .putValue("rating", Variables.integerValue(null)));
    }
}

```

- Crear la aplicación Web con Maven

Una aplicación de proceso es una aplicación Web Java común y se implementa exactamente de la misma manera.

Seleccione el archivo pom.xml en el *Package Explorer*, dar un clic derecho y seleccione *Run As / Maven Clean* y, a continuación, *Run As / Maven Install*. Esto generará un archivo WAR denominado loan-approval-cmmn-0.1.0-SNAPSHOT.war en la carpeta target/ del proyecto Maven.

- Implementar en Apache Tomcat

Para implementar la aplicación de proceso, copie y pegue loan-approval-cmmn-0.1.0-SNAPSHOT.war desde el proyecto Maven al directorio \$CAMUNDA_HOME/server/apache-tomcat/webapps.

Comprobar que el archivo de registro del servidor Apache Tomcat en la ruta \$CAMUNDA_HOME/server/apache-tomcat/logs. Seleccionar el archivo con el nombre catalina.out. Desplázarse hasta el final del archivo y si se observa el siguiente mensaje de registro, la implementación tuvo éxito:

```

INFO org.apache.catalina.startup.HostConfig.deployWAR
Deploying web application archive ...webapps\loan-approval-cmmn-0.1.0-S
NAPSHOT.war
INFO org.camunda.commons.logging.BaseLogger.logInfo
ENGINE-07015 Detected @ProcessApplication class 'org.camunda.bpm.getstar
ted.cmmn.loanapproval.LoanApprovalApplication'
INFO org.camunda.commons.logging.BaseLogger.logInfo
ENGINE-08024 Found processes.xml file at ../webapps/loan-approval-cmmn-
0.1.0-SNAPSHOT/WEB-INF/classes/META-INF/processes.xml
INFO org.camunda.commons.logging.BaseLogger.logInfo

```

```
ENGINE-08023 Deployment summary for process archive 'loan-approval-cmmn'
:

loan-approval.cmmn11.xml

INFO org.camunda.commons.logging.BaseLogger.logInfo
ENGINE-08050 Process application Loan Approval App CMMN successfully dep
loyed
INFO org.apache.catalina.startup.HostConfig.deployWAR
Deployment of web application archive ...webapps\loan-approval-cmmn-0.1
.0-SNAPSHOT.war has finished in 1,335 ms
```

- **Verificar la lista de tareas de implementación con Camunda**

Ahora se usa el *Tasklist* de la plataforma BPM Camunda para comprobar si el caso fue instanciado. Para ir a Camunda Tasklist se teclea en el explorador web la dirección <http://localhost:8080/camunda/app/tasklist> recordar loguarse con las credenciales de usuario: demo y contraseña: demo. Ver la Figura C-8.

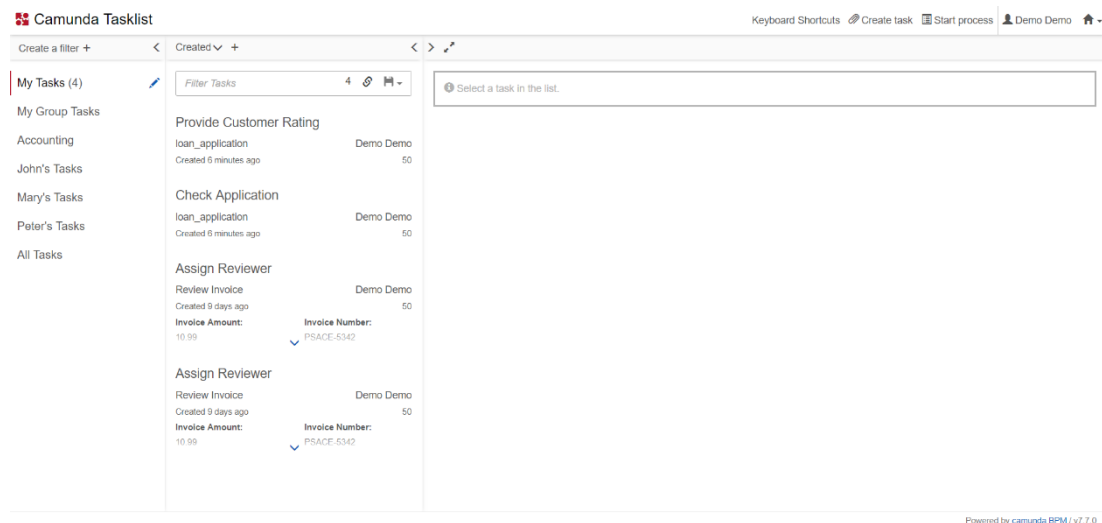


Figura C-8 Camunda Tasklist.

Después de haber iniciado sesión, haga clic en el filtro *My Task*. Debe haber dos tareas enumeradas con los nombres de *Check Application* y *Provide Customer Rating*. Seleccione la tarea *Check Application*.

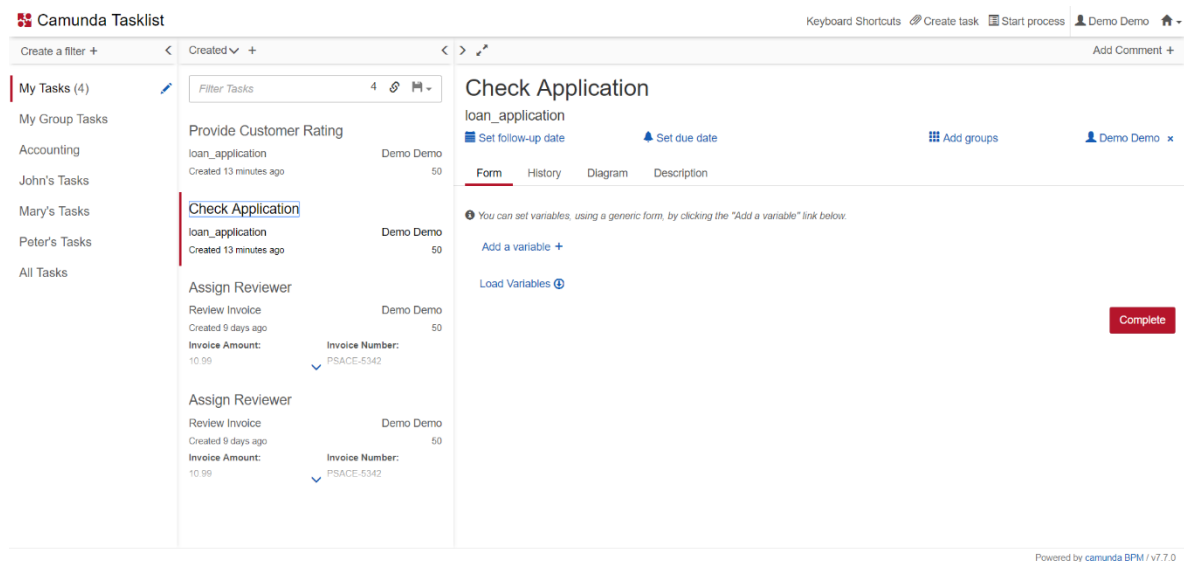


Figura C-9 Detalle de una tarea.

La lista de tareas muestra un formulario genérico que se puede utilizar siempre que no se haya agregado un formulario dedicado para una tarea humana. Puede usarse para agregar variables al caso. Por ahora, se deja vacío. Simplemente haga clic en *Complete*.

Hacer lo mismo con la tarea *Provide Customer Rating*. Ambas tareas ya se han completado y no hay más trabajo que hacer en este caso.

5. Uso de hitos (Milestones) y centinelas (Sentries) de CMMN

El siguiente paso consiste en añadir un hito. En CMMN, los hitos indican que se ha alcanzado un cierto objetivo intermedio en el caso. Las condiciones que definen cuándo se alcanza el hito se modelan usando Sentry.

- Definición de un hito

En el Modelador de Camunda, en el mismo caso *Loan Application*, crear un nuevo hito dentro del caso. Hacer doble clic en el hito y asignarle el nombre *Approved* (Aprobado).



Figura C-10 Añadir un hito al caso.

- Definición de un centinela

Los centinelas se utilizan para capturar las condiciones dentro de un caso y pueden desencadenar otros eventos. El hito aprobado se alcanza cuando ambas tareas se han completado con éxito, si la aplicación era suficiente y si el cliente recibió una buena calificación de solvencia.

En primer lugar, crear un criterio sobre el hito. Hacer clic en la tarea *Check Application task* y, a continuación, en el botón *Append Criterion* en el panel de contexto. Mover el centinela de entrada a la derecha y colóquelo en el borde del hito. Observe cómo se adjunta el criterio al hito y se crea una conexión onPart con el evento estándar completo.

Ahora se va a conectar la tarea *Provide Customer Rating* con el mismo centinela. Seleccionar la tarea y hacer clic en el botón *Connect using Discretionary/OnPart or Association button*. Mover el cursor del mouse y colocar el final de la conexión en el centinela.

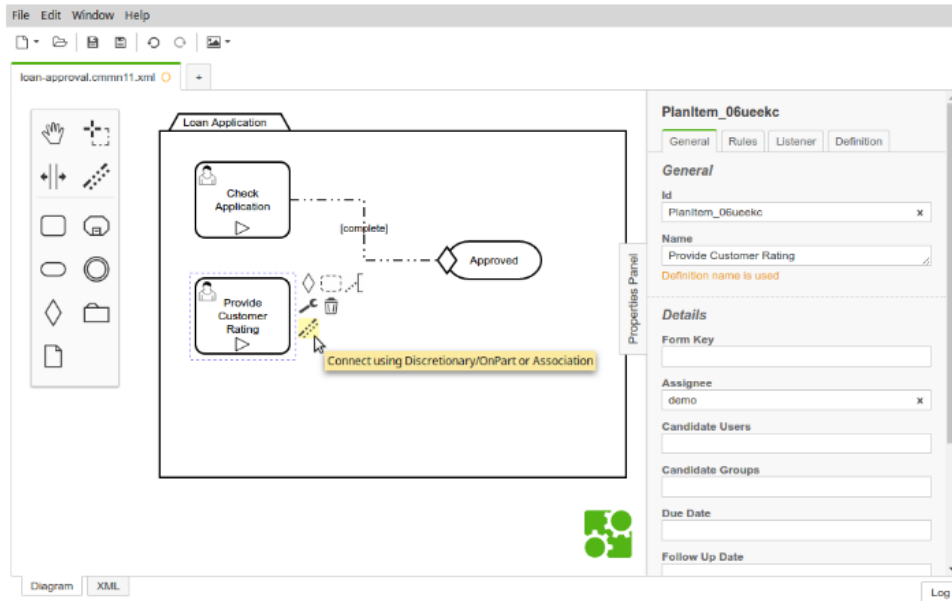


Figura C-11 Añadir un centinela al caso.

Ahora se puede configurar las dos condiciones (*application sufficient and good rating*) usando el panel de propiedades. Haga clic en el centinela y escribir lo siguiente en el campo *If Part Condition*: `${applicationSufficient && rating > 3}`

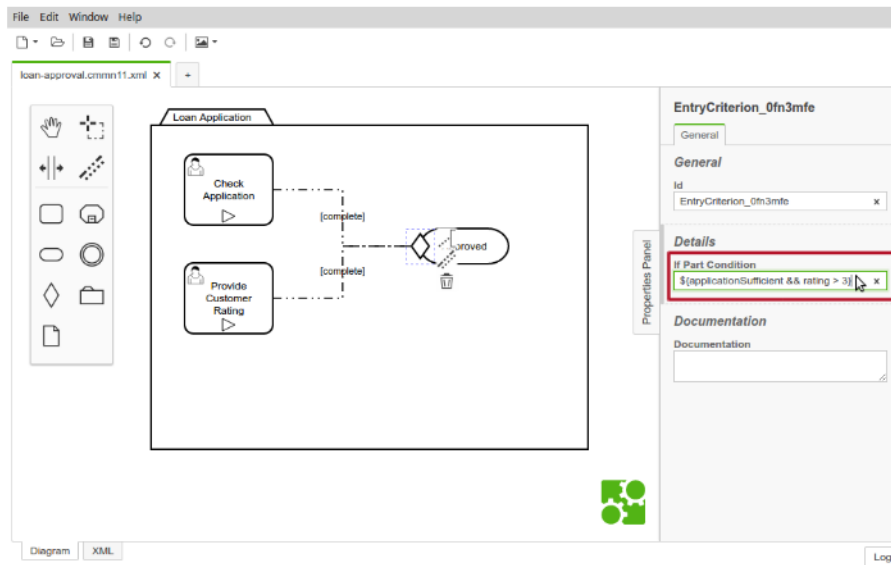


Figura C-12 Añadir una condición a un centinela.

- Oyente del hito (Milestone Listener)

Los hitos no se visualizan en las aplicaciones web de Camunda. Para ver que el hito se ejecuta se agrega un *CaseExecutionListener*. Se va al proyecto en Eclipse y se crea una nueva clase Java denominada *LifecycleListener* dentro del paquete existente *org.camunda.bpm.getstarted.cmmn.loanapproval* con el siguiente contenido:

```

package org.camunda.bpm.getstarted.cmmn.loanapproval;
import java.util.logging.Logger;
import org.camunda.bpm.engine.delegate CaseExecutionListener;
import org.camunda.bpm.engine.delegate DelegateCaseExecution;

public class LifecycleListener implements CaseExecutionListener {
    private final static Logger LOGGER = Logger.getLogger("LOAN-REQUESTS-CMMN");
    public void notify(DelegateCaseExecution caseExecution) throws Exception {
        LOGGER.info("Plan Item '" + caseExecution.getActivityId() + "' labeled '" +
caseExecution.getActivityName() + "' has performed transition: "
        + caseExecution.getEventName());
    }
}

```

Ahora este oyente tiene que ser registrado con el hito. Para hacer ese cambio se va al Modelador de Camunda. Hacer clic en el hito y, a continuación, en la pestaña *Listener* en el panel de propiedades. Ahora se puede crear un nuevo *Case Execution Listener* con un clic en el botón del signo + a la derecha. Se produce un nuevo *Case Execution Listener* con el tipo de evento y se ha creado el tipo Java Class. Configurar la clase Java insertando lo siguiente en el campo de *Java Class*:

org.camunda.bpm.getstarted.cmmn.loanapproval.LifecycleListener

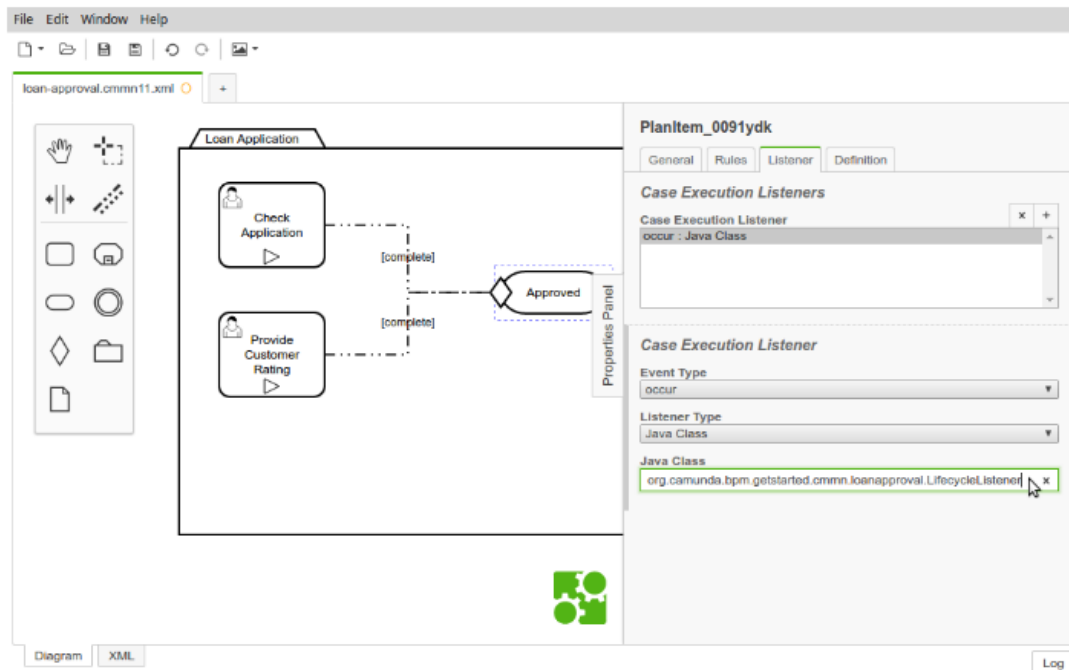


Figura C-13 Añadir un Oyente del Hito.

Es necesario guardar los cambios en el modelador antes de continuar.

- Reconstruir y desplegar.

Cuando se haya terminado, se guardan todos los recursos, se realiza una compilación Maven y se vuelve a implementar la aplicación de proceso (ver sección 4 de este apéndice).

Es recomendable primero seleccionar el archivo *pom.xml*, dar un clic derecho y elegir *Run as/Maven Clean* para asegurarse de que todos los recursos se reemplazan con su versión más reciente, posteriormente hacer clic derecho en el archivo *pom.xml* y elegir *Run as/Maven Install*.

A continuación, se abre la lista de tareas (Vease Figura C-8) y se accede a las tareas del usuario de la aplicación. Primero, se da clic en la tarea *Check Application*. Después, hace clic en *Add a variable* y en el campo *Name* escribir *applicationSufficient*, en el campo *Type* seleccionar *Boolean*, en *Value* marca la casilla de verificación que aparece, para finalizar dar clic en *Complete*.

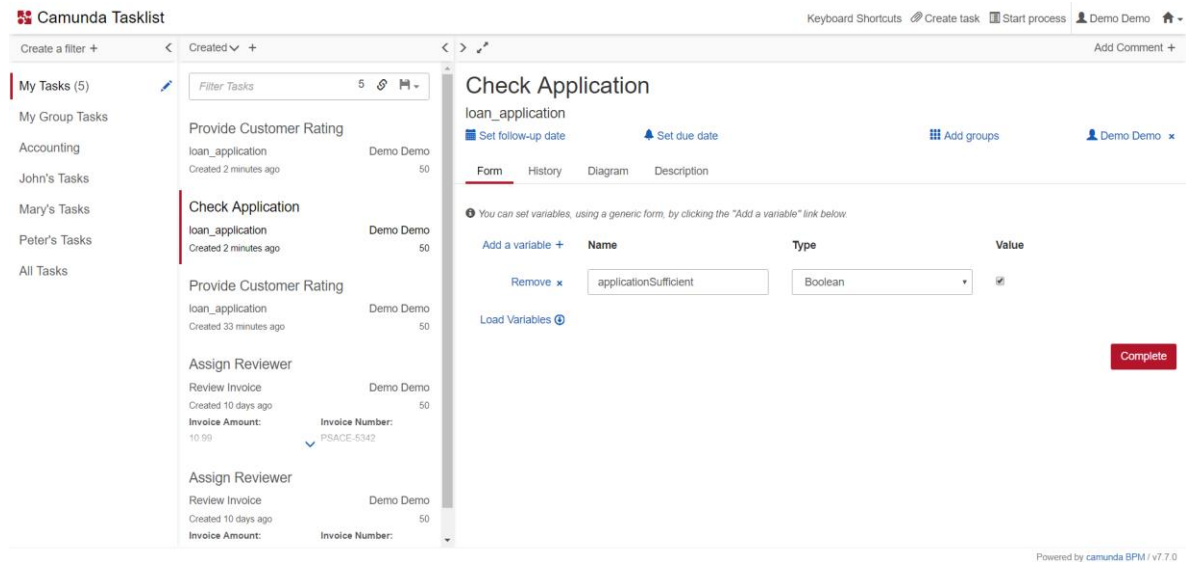


Figura C-14 Agregar variable a la Tarea Check Application.

A continuación, completar la tarea *Provide Customer Rating*. Una vez más, dar clic en *Add a variable* y el *Name* escribir *rating*, en el *Type* elegir *Integer* y en el *Value* asignar el 4, por último, dar clic en *Complete*. Ver Figura C-15.

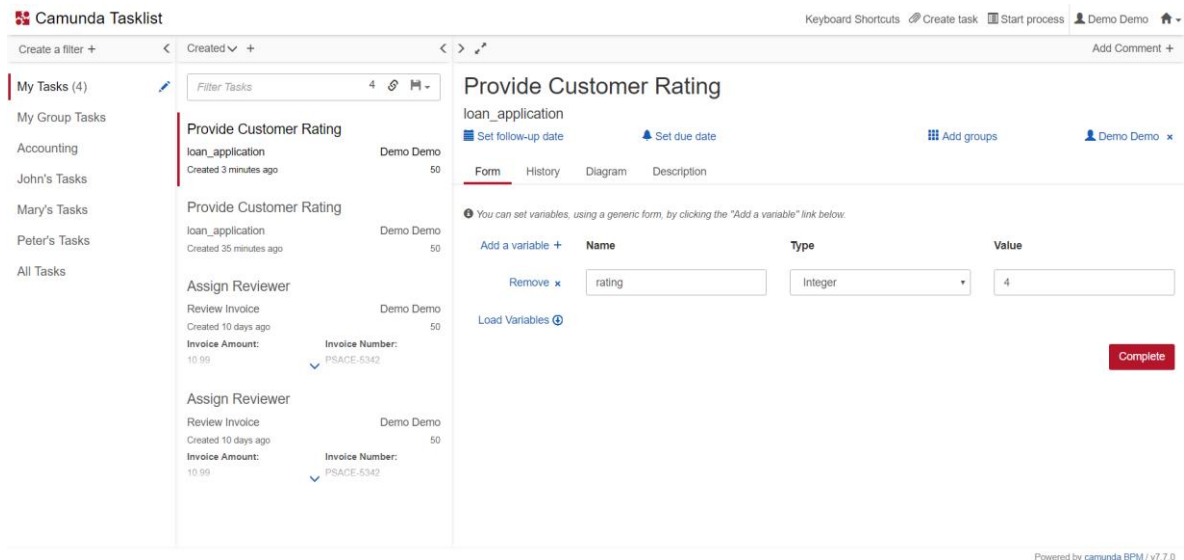


Figura C-15 Agregar variable a la Tarea Provide Customer Rating.

Después de haber completado las tareas y configurado las variables de forma que el hito haya ocurrido. En la consola, debería de observar las siguientes entradas de registro:


```
INFO org.camunda.bpm.getstarted.cmmn.loanapproval.LifecycleListener
Plan Item 'PI_Milestone_1' labeled 'Approved' has performed transit
ion: occur
```

El *lifecycle listener* ha sido notificado, mostrando que el hito realmente ha ocurrido.

6. Uso de un Sentry CMMN como criterio de salida.

Cuando una solicitud de préstamo no es suficiente, por ejemplo, porque tiene errores en la forma, no hay necesidad de proporcionar una calificación del cliente. Se puede expresar esto en CMMN agregando un sentry que actúa como criterio de salida.

- Agregar una sentry como criterio de salida.

En el Modelador de Camunda, dar clic en la tarea *Check Application* para abrir el panel de contexto y, a continuación, hacer clic en *Append Criterion*, mover el sentry al borde superior de la tarea *Provide Customer Rating* para adjuntarlo, después, hacer clic en el sentry de entrada y abrir el menú de reemplazo haciendo clic en el icono de llave inglesa , por último, hacer clic en *Exit Criterion* para transformar el sentry.

Mantener seleccionado el sentry de salida y configurar la *If Part Condition* con el panel de propiedades. Insertar el siguiente texto: `${!applicationSufficient}`.

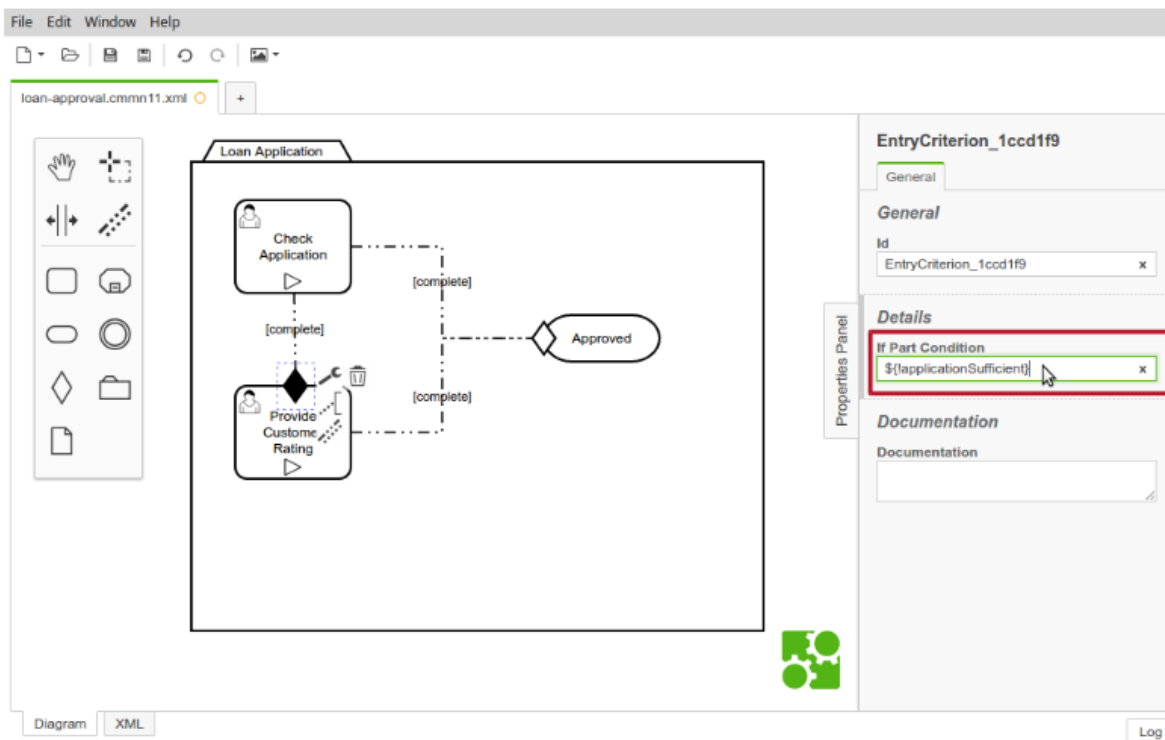


Figura C-16 Agregar una condición a un sentry de salida.

Guardar los cambios en el modelador antes de continuar.

- **Reconstruir y desplegar.**

Una vez más, se reconstruye y despliega (Como se realizó en la sección 5 de este apéndice).

En el *TaskList* de la Plataforma de Camunda acceder a las tareas del usuario de la aplicación ejemplo. Como antes, comprobar la tarea *Check Application*. Para activar el sentry, agregar la variable *applicationSufficient* de tipo Boolean, pero esta vez asegurarse de dejar la casilla de verificación desmarcada. Hacer clic en *Complete*.

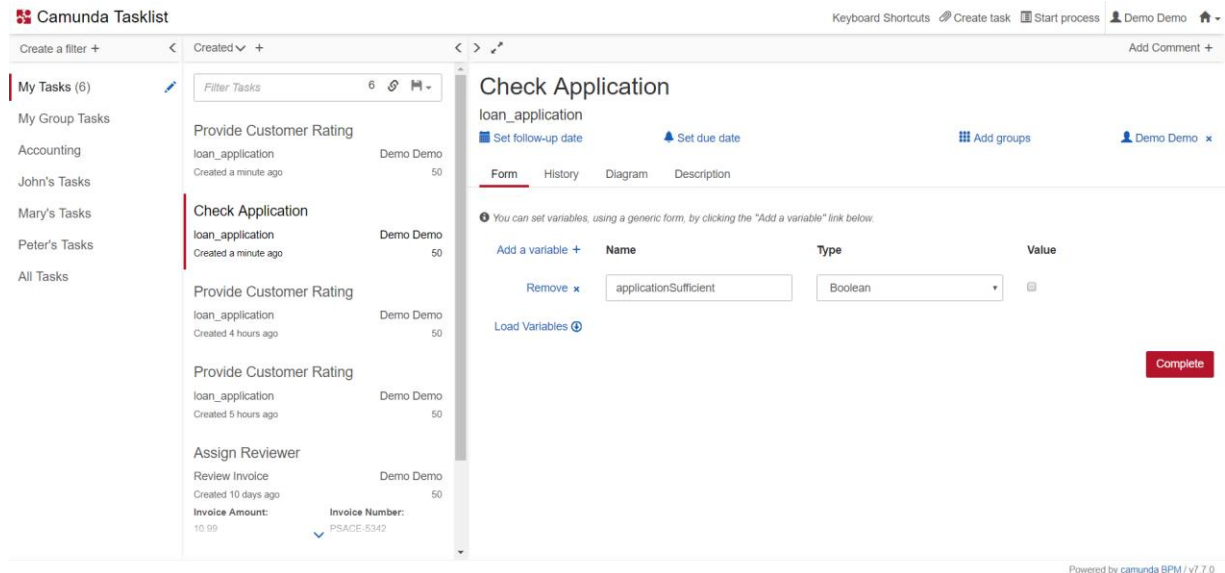


Figura C-17 Agregar variable con valor False a la Tarea Check Application.

Notará que la tarea *Provide Customer Rating* ha desaparecido de *TaskList*. Esto se debe a que el sentry ha sido activado y el criterio de salida de la tarea se ha cumplido. Además, se puede comprobar la consola de Tomcat. Esta vez, no hay entrada de registro para el hito.