

Universidad Católica de Valparaíso
Escuela de Ingeniería Industrial

**Perfeccionamiento de un Proceso de
Modelado de Sistemas de Información
Dirigido por Comportamiento**

por

Guillermo Bustos R.

Ayudantes de investigación

Paulina Domínguez

Gabriela Figueroa

Sandra Frugone

Informe Técnico Final
Proyecto Dirección de Investigación
205.741/98

Enero, 1999

Índice

Glosario.....	3
Lista de Tablas	4
Lista de Tablas	5
Introducción.....	6
1 Visión general de la propuesta.....	7
1.1 Modelamiento inicial del sistema	8
1.1.1 Modelo de visión global	9
1.1.2 Modelo de procesos	9
1.1.3 Diccionario de datos	9
1.2 Ciclos de vida	9
1.3 Derivación del modelo estructural de objetos	10
2 Paralelo entre HLS y OMT.....	11
2.1 Diferencias formales.....	11
2.2 Diferencias estructurales.....	15
3 Paralelo entre HLS y Use Case.....	20
3.1 Diferencias formales.....	20
3.2 Diferencias funcionales.....	25
4 Conclusiones.....	28
ANEXO 1 Definición del Problema de la IFIP	29
ANEXO 2 Modelo Completo del Problema de la IFIP	31
ANEXO 3 Definición del Problema de la Biblioteca Universitaria.....	54
ANEXO 4 Modelo Completo del Problema de la Biblioteca Universitaria.....	57
ANEXO 5 Definición del Problema del Juzgado de Letras	83
ANEXO 6 Modelo Completo del Problema del Juzgado de Letras	87
Bibliografía	109

Glosario

acción: una actividad realizada durante una transición, que puede ser una operación específica dentro del proceso, un evento para otra transición o puede corresponder a una respuesta del proceso.

clase: un conjunto formado por varios objetos del mismo tipo, en donde la clase es un patrón para todos ellos.

condición: expresión lógica cuyo valor_verdad determina el disparo de la transición cuando el evento de esta transición ocurre. Si su valor_verdad fuese verdadero la transición ocurre en caso contrario no ocurre.

encapsulamiento: el encapsulamiento propone que sean separados, en los objetos, los aspectos internos de los externos, en donde los aspectos externos son conocidos por otros objetos y los internos son de conocimiento exclusivo del propio objeto.

estímulo: cualquier tipo de comunicación que algún emisor envía al sistema.

evento: hecho relevante para un proceso específico que dispara una transición (dependiente de una condición si es que ésta existe).

jerarquía de herencia: las jerarquías de herencia ocurren entre clases de objetos y pueden ser construidas por generalización (abstracción de propiedades comunes de una o más subclases para una superclase) o especialización (reutilización de las propiedades de una superclase en una o más subclases). Las propiedades heredadas entre un nivel y el siguiente pueden ser atributos/operaciones, relacionamientos, comportamiento y colaboración.

parámetro: datos que acompañan o eventualmente justifican el estímulo y que sirven de entrada para el proceso que trata particularmente este estímulo. Los parámetros también aparecen en los eventos como una derivación de los parámetros de los estímulos. En el caso de los parámetros de las respuestas, éstos corresponden a datos de salida del proceso y, eventualmente, justifican la respuesta o sólo la acompañan. Los parámetros de las acciones desempeñan la misma función que los de las respuestas.

proceso: componente del sistema encargado del tratamiento de un estímulo específico. Como consecuencia de la ejecución interna de este tratamiento, resulta la emisión de una respuesta también específica.

receptor: agente externo que desempeña el papel específico de recibir respuestas emitidas por el sistema.

respuesta: cualquier tipo de comunicación que el sistema emite para un receptor.

Lista de Tablas

Figura 2.1 – Jerarquías de herencia de la clase de objeto autor.....	18
---	----

Lista de Tablas

Tabla 2.1 - Análisis de clases para el problema de la IFIP	12
Tabla 2.2 - Análisis de clases para el problema de la Biblioteca Universitaria	15
Tabla 2.3 - Análisis de clases para el problema del Juzgado de Letras	16
Tabla 3.1 - Análisis de funciones entre procesos y use cases para la IFIP	21
Tabla 3.2 - Análisis de funciones entre procesos y use cases para la IFIP	22
Tabla 3.3 - Análisis de funciones entre procesos y use cases para la Biblioteca Universitaria	22
Tabla 3.4 - Análisis de funciones entre use cases y procesos para la Biblioteca Universitaria	23
Tabla 3.5 - Análisis de funciones entre use cases y procesos para el Juzgado de Letras	24
Tabla 3.6 - Análisis de funciones entre use cases y procesos para el Juzgado de Letras	25

Introducción

En este informe se presentan los resultados obtenidos del desarrollo y el estudio de diferentes modelos para la contribución en el perfeccionamiento del proceso previamente propuesto de modelado de sistemas de información dirigido por comportamiento.

Para comenzar, se presenta una visión general de la propuesta [Bustos96] que se utiliza en este informe, y que permite modelar sistemas desde una perspectiva dinámica, con el formalismo denominado *High-Level-Statecharts*. De esta manera, se señalan y explican los diferentes modelos y diagramas que se desarrollan, en las distintas etapas del modelado. Además, se realiza una breve explicación sobre la manera en que se deriva el modelo estructural orientado a objeto a partir de los HLS desarrollados.

A continuación se presentan comparaciones entre los modelos obtenidos según el desarrollo de la propuesta y otros existentes, tales como el modelo OMT (*Object Modeling Technology*) desarrollado por [Rumbaugh91] y el modelo de *Use Cases* desarrollado por [Jacobson92]. La finalidad de este paralelismo es encontrar diferencias y similitudes entre los modelos en cuanto, por ejemplo, a las clases, atributos y jerarquías obtenidas por OMT y a la funcionalidad entregada por el sistema hacia los agentes externos para el caso de los use cases.

En los anexos se encuentran los enunciados de los tres problemas desarrollados, así como los diferentes modelos obtenidos directamente de la propuesta, los modelos estructurales orientados a objeto derivados de los HLS, los modelos OMT y por último los Use Cases.

Por último, se presentan algunas conclusiones obtenidas del análisis comparativo entre modelos, cuya finalidad es entregar una guía de las áreas que la propuesta original no considera o que deberían ser perfeccionadas.

1 Visión general de la propuesta

La propuesta de modelamiento aquí presentada tiene como objetivo permitir modelar sistemas desde una perspectiva dinámica, postergando el encapsulamiento de la orientación a objeto hacia las etapas finales del proceso. En otras palabras, esta técnica es una estrategia dirigida por comportamiento que privilegia el aspecto dinámico de los sistemas y que posee suficiente poder de expresión para, al mismo tiempo, permitir el modelamiento de sistemas de información a nivel conceptual y derivar desde los modelos obtenidos una representación estructural orientada a objetos [Bustos96].

La propuesta es dividida conceptualmente bajo tres perspectivas:

- **Enfoque sistémico:** De modo de situar, contextualizar y concebir el sistema de información a ser modelado, de manera que se entienda como perteneciente a un sistema mayor o supra-sistema.
- **Modelamiento dinámico:** Para describir el comportamiento y la estructura interna del sistema. La estructura de los objetos puede ser determinada a partir del comportamiento externo e interno que el sistema debe presentar, permitiéndose así introducir tardíamente el encapsulamiento.
- **Modelamiento estructural orientado a objeto:** Utilizado para derivar el modelo estructural de objetos, el cual está ligado a las propiedades estáticas o pasivas de los sistemas. La estructura de los objetos incluye su identidad, su clasificación, su encapsulamiento (sus atributos y operaciones) y sus asociaciones estáticas (jerarquías de herencia, agregación, composición y asociaciones específicas derivadas del dominio del problema).

El sistema, en la concepción de esta propuesta, está compuesto de procesos concurrentes, cada uno de los cuales recibe un estímulo del ambiente y realiza un tratamiento específico sobre él generando una respuesta para el ambiente. Los estímulos externos son descompuestos en conjuntos de eventos concurrentes tratados en el interior del proceso. Las acciones realizadas en el interior del mismo se relacionan a las respuestas generadas para el exterior.

Los procesos son modelados utilizando el formalismo denominado *High-Level Statecharts (HLS)*. HLS es una extensión de los statecharts de Harel, con ciertas modificaciones como la introducción de estados “parametrizados” usando variables y la representación genérica de conjuntos de estados concurrentes y exclusivos.

De manera de comprender la propuesta, es necesario especificar algunas ideas del modelamiento de sistemas aquí expuestas:

- **Cambio del énfasis en los objetos hacia las relaciones dinámicas:** Los objetos pueden ser deducidos a partir de las relaciones dinámicas existentes en una red de eventos.

- Cambio del énfasis en la estructura hacia los procesos: La estructura de objetos puede ser derivada desde la red de procesos subyacente.
- Las propiedades de los objetos son determinadas por las relaciones: Los atributos y las operaciones de los objetos son determinados sólo por las asociaciones con otros objetos.

De esta manera es posible construir modelos dinámicos no orientados a objetos, en los cuales los objetos y sus respectivas estructuras son derivados en función de los patrones de comportamiento y de interacción. Este encapsulamiento tardío entrega más calidad, riqueza, eficacia y poder de expresión al modelo, sin forzar conceptos del modelamiento orientado a objetos en una etapa inicial. De esta manera, el modelamiento inicial se torna orientado al problema y no a la solución orientada a objeto.

El sistema de información a ser modelado es así un sistema que reacciona en un ambiente y que, por tanto, desempeña una función o papel definido dentro de un supra-sistema. La organización del sistema es hecha centrándose en las propiedades dinámicas del mismo, es decir, focalizando el comportamiento externo al sistema y derivando, posteriormente, unidades internas (subsistemas) que respondan específicamente a los estímulos provenientes de otros sistemas.

El proceso de la técnica puede ser dividido en tres grandes etapas, con diversas actividades, modelos y notaciones.

1.1 Modelamiento inicial del sistema

El modelamiento inicial permite una primera aproximación del sistema considerando su comportamiento.

En este caso son definidos el contexto en el cual opera el sistema y sus relaciones con otros sistemas. Así mismo, es descrito su comportamiento en la forma de procesos. Esta etapa se basa en el enfoque sistémico y en los conceptos de modelamiento dinámico.

La entrada a este paso está constituida por el propio problema, posiblemente en la forma de lenguaje natural de los requisitos que el sistema debe satisfacer.

Las salidas de esta etapa son un modelo de visión global, un modelo de procesos y un diccionario de datos.

1.1.1 Modelo de visión global

Modelo constituido por dos diagramas:

- Diagrama de contexto del sistema: Contextualiza el sistema situándolo dentro de un supra-sistema que contiene a su vez a otros sistemas, con los cuales el sistema a ser modelado se relaciona. Este diagrama incluye a los agentes externos relacionados con el sistema mediante estímulos y respuestas.
- Diagrama de visión global: Describe el sistema como un conjunto de procesos concurrentes, los cuales reciben estímulos y generan respuestas desde/para los agentes externos. Cada proceso es concebido como una caja negra y su comportamiento es descrito en los HLS de comportamiento. Todos los datos envueltos en el sistema son representados como variables cuyos dominios deben ser definidos.

1.1.2 Modelo de procesos

Constituido por un conjunto de diagramas que describen el comportamiento de cada uno de los procesos del diagrama de visión global. Es decir, cada una de las cajas negras son “abiertas” y cada proceso es mostrado como una red jerarquizada de estados y transiciones, representadas bajo la forma de los *High-Level Statecharts (HLS)*

1.1.3 Diccionario de datos

Conjunto de definiciones de la composición y el dominio, para todas las variables utilizadas en los diagramas de visión global y de procesos.

1.2 Ciclos de vida

La construcción de ciclos de vida es un proceso de encadenamiento de procesos, o partes de ellos, que traten el mismo tipo de dato, representado por una variable, de manera de configurar HLS mayores que describan toda la “historia” del ciclo de vida para aquel tipo de dato. Este es un proceso de desintegración de componentes organizados en los HLS de procesos y su posterior integración en diagramas HLS de ciclos de vida.

Esta etapa utiliza como entrada los tres documentos de salida del paso anterior, produciendo como salida un conjunto de diagramas HLS de ciclos de vida, denominado modelo de ciclos de vida.

Para obtener este modelo se deben realizar diversas actividades:

- Identificación de estados y variables comunes.
- Conexión de pre y post estados de los diferentes procesos que tratan del mismo tipo de dato.
- Verificación de consistencia en el uso de los nombres de estados y variables.

1.3 Derivación del modelo estructural de objetos

En este paso es introducido el encapsulamiento de la orientación a objetos. Utiliza como datos de entrada los documentos de la sección anterior.

En esta etapa es realizada una identificación, en los modelos construidos en los pasos anteriores, de todos los elementos que estructuran un modelo estático orientado a objetos. Por lo cual se hace necesario realizar las siguientes actividades:

- Identificación de objetos candidatos: A partir de los datos que poseen los HLS de ciclo de vida, de los datos de entrada para el sistema y de las acciones específicas realizadas en cada proceso.
- Derivación de jerarquías de herencia: Las subclases, dentro de una jerarquía de herencia, son identificadas por un proceso de especialización a partir de comportamientos alternativos en los ciclos de vida (subciclos).
- Determinación de asociaciones estáticas: Las asociaciones estáticas y los objetos participantes en ellas son determinados a partir de los estímulos que el sistema recibe y las propiedades que éstos presentan.
- Definición de atributos: Las descripciones de los datos, incluidas en el diccionario de datos, permiten definir los atributos de las clases.
- Definición de operaciones: Todas las acciones que aparecen en cada uno de los diagramas HLS de ciclo de vida sirven para definir las operaciones.

2 Paralelo entre HLS y OMT

Esta sección tiene como objetivo realizar un paralelo entre los modelos estructurales orientados a objeto obtenidos de la derivación de los HLS y los obtenidos de la técnica OMT desarrollada por Rumbaugh [Rumbaugh91]. La comparación es realizada para los tres problemas desarrollados (Ver Anexos 1 a 5) y se basa principalmente en desarrollar un paralelo entre las clases de objetos y sus atributos, así como entre las jerarquías de herencia obtenidas. Las operaciones no son comparadas pues el modelo OMT no las considera.

En la tablas 2.1, 2.2 y 2.3 se muestran las distintas clases de objetos con sus atributos, incluidas en el modelo estructural de objetos derivado de los HLS y en el modelo según la técnica OMT. El término “no aparece” es colocado cuando la clase de objeto que está siendo descrita no aparece en forma exacta en el otro modelo, notando que aún cuando la clase represente lo mismo, pero esté con otro nombre, esta clase es considerada como un objeto aparte.

Cabe considerar que en las tablas fueron colocados todos los objetos participantes en las jerarquías de herencia (tanto padres como hijos).

2.1 Diferencias formales

Se entenderá por diferencia formal aquella referida a los casos en los cuales atributos en un modelo corresponden a clases en el otro y viceversa, además de los casos en los cuales clases diferentes representan lo mismo.

Basándose en la Tablas 2.1 se puede concluir que:

- a) El miembro de grupo (16) y el representante nacional (26) son considerados como una persona (21) en el caso de HLS.
- b) La clase respuesta invitación (28) en OMT corresponde a la clase carta de aceptación (7) en el otro caso.
- c) En el caso de OMT se considera a un receptor de call for *papers* (25) y en el caso de HLS se considera al call for *papers* (6) (llamada para presentar *papers*) como una clase. Esto es sólo una diferencia en la manera de enfocar el problema.

		Modelo estructural derivado de HLS	Modelo según técnica OMT
	Clase de objeto	Atributos	Atributos
1.	autor	nombre institución título dirección	-
2.	autor ausente	-	no aparece
3.	autor invitado	-	no aparece
4.	autor participante	-	no aparece
5.	autor no participante	-	no aparece
6.	<i>call for papers</i>	-	no aparece
7.	carta de aceptación	-	no aparece
8.	carta de intención	-	fecha de envío de la intención intención
9.	decisión	-	no aparece
10.	evaluador	máximos revisiones	tema
11.	invitación	-	no aparece
12.	invitación autor	-	no aparece
13.	invitación prioritaria	-	no aparece
14.	invitado	no aparece	fecha envío de la invitación
15.	invitado prioritario	no aparece	fecha de envío de la invitación prioritaria
16.	miembro grupo	no aparece	nombre grupo
17.	moderador	-	tema
18.	<i>paper</i>	título tema	título tema fecha de envío estado
19.	<i>paper</i> aceptado	-	-
20.	<i>paper</i> rechazado	-	-
21.	participante	no aparece	papel
22.	persona	nombre institución título	nombre institución dirección
23.	persona ausente	-	no aparece
24.	persona invitada	-	no aparece
25.	receptor <i>call for papers</i>	no aparece	fecha envío de la llamada
26.	reporte	evaluador tema	evaluación
27.	representante nacional	no aparece	cargo
28.	respuesta invitación	no aparece	fecha de envío respuesta respuesta
29.	sesión	sala horario	tema fecha horario

Tabla 2.1 - Análisis de clases para el problema de la IFIP

En base a la Tabla 2.2, se concluye lo siguiente:

- a) La super clase préstamo (13) del modelo OMT no considera la fecha de devolución y la fecha de devolución real en sus atributos, ya que ellas son considerados en la clase préstamo casa satisfecho (16) (hija del anterior). Esto sucede ya que el modelador realizó una distinción entre préstamos casa y préstamos sala, en donde cada uno cuenta con préstamos satisfechos e insatisfechos. Dado que sólo los préstamos para la casa poseen fecha de devolución, pues los para sala deben ser devueltos dentro del mismo día, se consideró que sólo ellos requerían de los atributos nombrados anteriormente. Las clases préstamo rechazado por no disponibilidad (19), préstamo efectivo(18) y préstamo concluido (17) de la tabla consideradas según HLS representan las clases préstamo casa insatisfecho (15), préstamo casa satisfecho (16), préstamo sala insatisfecho (21) y préstamo sala satisfecho (22) según OMT.
- a) La clase lector (6) es considerada de manera similar en ambos casos.
- b) Las clases trabajador (29) con sus subclases académico (1) y funcionario (5) y la clase alumno (3) con sus subclases regular (23) y memorista (11) fueron consideradas por el modelador para representar que cada uno de los posibles lectores posee distintas cantidades de textos que puede solicitar y distintos plazos para devolverlos. Esto está considerado en el atributo máximos y fecha devolución de las clases lector y préstamo, según HLS.
- c) La clase pago (12) es considerada como un atributo, morosidad, en OMT.
- d) Las clases material bibliográfico (9) y texto (26) representan prácticamente lo mismo. Cabe notar que la clase texto según HLS (26) representa tanto a los libros de baja demanda, de alta demanda y material de consulta, en cambio la clase texto según OMT (25) representa el caso general de cualquier material bibliográfico y no debe ser considerada igual a la clase 26.

En relación a el Juzgado de Letras, analizado en la Tabla 2.3, se concluye lo siguiente:

- a) Para la clase causa (2) según HLS, sus atributos actuario, juzgado (considerando juez, jurisdicción y tipo de juzgado), tipo y partes involucradas son clases en el modelo OMT (1, 10, 16, 12).
- b) La clase diligencia (5) posee similares atributos, con excepción del atributo tipo_exhorto, el cual en el caso de los HLS marca la diferencia entre las diligencias que están siendo llevadas por el mismo juzgado y aquellas que son enviadas para ser realizadas en otro juzgado. En OMT esto es manejado en forma de clases.

		Modelo estructural derivado de HLS	Modelo según técnica OMT
	Clase de objeto	Atributos	Atributos
1.	académico	no aparece	-
2.	alta demanda	idioma	no aparece
3.	alumno	no aparece	-
4.	baja demanda	idioma	no aparece
5.	funcionario	no aparece	-
6.	lector	rut nombre dirección condición rol máximos prestados	rut nombre dirección teléfono cantidad préstamos casa morosidad
7.	lector incluido	-	no aparece
8.	lector descartado	-	no aparece
9.	material bibliográfico	no aparece	código título tema copias disponibilidad *****
10.	material consulta	-	no aparece
11.	memorista	no aparece	-
12.	pago	valor_multa	no aparece
13.	préstamo	fecha_préstamo fecha_devolución fecha_para_devol	texto lector fecha solicitud
14.	préstamo casa	no aparece	-
15.	préstamo casa insatisfecho	no aparece	causa
16.	préstamo casa satisfecho	no aparece	fecha devolución fecha devolución real
17.	préstamo concluido	-	no aparece
18.	préstamo efectivo	-	no aparece
19.	préstamo rechazado por no disponibilidad	-	no aparece
20.	préstamo sala	no aparece	-
21.	préstamo sala insatisfecho	no aparece	causa
22.	préstamo sala satisfecho	no aparece	-
23.	regular	no aparece	-
24.	revista	número	no aparece
25.	texto	no aparece	código autor título inventario tema año editora tipo texto n_copias n_disponibles

			fecha adquisición
26.	texto	código autor título inventario tema año editora tipo texto n_copias n_disponibles fecha_adquisición	no aparece
27.	texto incluido	-	no aparece
28.	texto descartado	-	no aparece
29.	trabajador	no aparece	-

Tabla 2.2 - Análisis de clases para el problema de la Biblioteca Universitaria

2.2 Diferencias estructurales

Se entenderá por diferencia estructurales al caso que ocurre cuando clases de objetos no son consideradas en uno de los modelos. Se considerará además en esta sección las observaciones existentes entre las diferencias que puedan existir en cuanto a las jerarquías de herencia.

En base a la Tabla 2.1 que ilustra el problema de la IFIP, se puede decir que:

- Las clases autor ausente (2), invitado (3), participante (4) y no participante (5), del modelo derivado de los HLS, pueden ser consideradas análogas a las clases participante (21) y autor (1) del modelo según OMT, con la salvedad que en este caso no se considera la ausencia de alguno de ellos. Esto es explicado con más detalle en el párrafo dedicado a las jerarquías de herencia.
- En el caso de OMT, las invitaciones (11 y 12) no son consideradas clases.

		Modelo estructural derivado de HLS	Modelo según técnica OMT
	Clase de objeto	Atributos	Atributos
1.	actuario	no aparece	nombre
2.	causa	actuario juzgado rol tipo materia principal fecha solicitud fecha préstamo materia secundaria partes involucradas	tipo materia principal materia secundaria diligencias actuario orden de no informar
3.	causa archivada	-	no aparece
4.	causa terminada	-	no aparece
5.	diligencia	tipo exhorto fecha solicitud fecha realización juzgado origen juzgado ejecutor	rol identificador fecha inicio tramitación fecha en que fue solicitada fecha de realización detalle tipo
6.	exhorto	fecha solicitud fecha realización juzgado origen juzgado ejecutor	fecha de ingreso juzgado rol de origen materia
7.	exhortos recibidos	no aparece	descripción
8.	juez	no aparece	nombre
9.	jurisdicción	no aparece	nombre
10.	juzgado	no aparece	nombre
11.	motivo término	no aparece	descripción
12.	partes involucradas	no aparece	descripción
13.	proceso judicial	no aparece	nombre
14.	sentencia	fecha emisión rol causa motivos término	no aparece
15.	tipo de causa	no aparece	nombre
16.	tipo de juzgado	no aparece	descripción
17.	tipo de motivo de término	no aparece	descripción
18.	tramitación	fecha inicio fecha término	denunciante

Tabla 2.3 - Análisis de clases para el problema del Juzgado de Letras

- c) La clase invitado prioritario (15) es considerada en HLS mediante la asociación entre la clase invitación prioritaria (13) y persona (21) y no como una clase en sí misma, como es el caso de OMT.

En lo relacionado al problema de la Biblioteca Universitaria y en base a la Tabla 2.2, se puede decir lo siguiente:

- a) Las clases lector incluido (7) y lector descartado (8) no fueron consideradas en OMT por el modelador, ya que no consideró el ingreso y la eliminación de lectores.
- b) Las clases, según OMT revista (24), baja demanda (4), alta demanda (2) y material consulta (10) no son consideradas en HLS, ya que aquí no se hace distinción entre tipos de lectores, en cambio, según OMT, fue necesario incluirlas debido a que existía discriminación entre tipo de lectores además del hecho de que no todos ellos pueden acceder a los mismos tipos de textos.
- c) Las clases texto incluido (27) y texto descartado (28) no fueron consideradas en OMT, ya que el modelador consideró que la función de descartes era relevante para el sistema de información, pues una vez que se descarta algún material este es eliminado del sistema.

A partir de la Tabla 3.3 que analiza el problema del Juzgado de Letras, se puede decir lo siguiente:

- a) La jerarquía de herencia que incluye a causa (2), causa terminada (4) y causa archivada (3) no es considerada en OMT.
- b) La clase sentencia es manejada en el modelo OMT mediante las clases motivo de término (11) y tipo de motivo de término (17), en donde se considera que los factores que pueden dar fin a una causa se encuentran estandarizados, lo cual no es necesariamente cierto, pero puede ser atribuirle esto a un error de interpretación del modelador.

En cuanto a las jerarquías de herencia se pueden encontrar algunas diferencias. En el Anexo 2, secciones 5 y 6 se puede encontrar el modelo estructural del problema de la IFIP derivado de los HLS y de la técnica OMT. En el primer caso existe la jerarquía que comprende a autor con sus descendientes autor participante y autor no participante, de la misma manera autor participante posee los descendientes autor ausente y autor invitado. Al derivar el modelo según la técnica OMT esta jerarquía fue considerada por decirlo de alguna manera en forma inversa pues se puede encontrar a participante como padre y a autor participante como descendiente, por lo cual, no pueden ser consideradas como análogas. Un extracto de estas jerarquías puede ser visto en la Figura 2.1.

La jerarquía de herencia derivada según la técnica OMT no considera una diferenciación entre personas y autores invitados y ausentes, esto debido a que no es un aspecto de especial relevancia para el problema.

El modelo estructural derivado de los HLS considera jerarquías de herencia más complejas, debido a que en él se pueden encontrar jerarquías con sobreposición (triángulos negros) lo que implica que la clase considerada puede cumplir el rol de sus descendientes en forma simultánea sin ningún problema. Por ejemplo, se puede ver que un participante puede ser al mismo tiempo moderador, autor evaluador o miembro de grupo. Esto puede ser visto en la Figura 2.1.

El tipo de jerarquía recién nombrada no es obtenida mediante la derivación directa de los HLS debido a la forma en que estas jerarquías son encontradas mediante los subciclos de los ciclos de vida. Además, OMT permite la sobreposición de clases, pero otras notaciones, tal como UML no, por lo que no se un asunto que merezca de una mayor consideración.

El modelo estructural para el caso de la biblioteca puede ser encontrado en el Anexo 3. Se puede ver que en el caso de OMT, el modelador no consideró diferencias entre textos y lectores incluidos, eliminados y descartados. Como ya fue explicado anteriormente, esto se debe a que el modelador consideró que una vez que el texto o el lector es descartado o eliminado, simplemente es eliminado del sistema, debido a que entre los indicadores necesarios para una buena gestión del sistema no se consideró alguno relativo a reportes históricos de los libros que la biblioteca maneja, por lo cual, no se hace estrictamente necesario poseer estos datos.

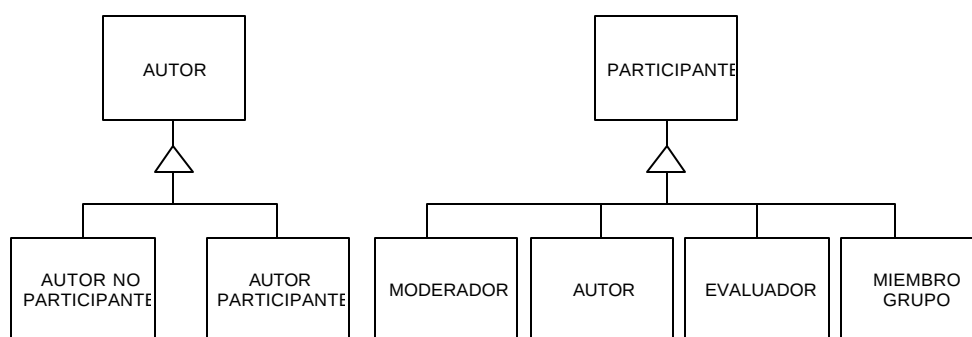


Figura 2.1 – Jerarquías de herencia de la clase de objeto autor

En cuanto a la jerarquía referida al préstamo, ésta es considerada en OMT, sólo que de una manera diferente, pues al estar los préstamos diferenciados en “para la casa” y “para sala”, cada una de estas clases posee sus respectivos descendientes insatisfechos y satisfechos, que podrían ser considerados análogos a préstamo rechazado por no disponibilidad y préstamo efectivo, respectivamente.

Al derivar el modelo estructural del problema de la biblioteca se obtuvieron, entre otras, las clases de objetos Lector, Texto y Préstamo a partir de variables de referencia. En los procesos `Solicitud_De_Préstamo_De_Texto`, `Devolución_de_Texto`, `Solicitud_De_Préstamo_De_Revista` y `Devolución_de_Revista`, los estímulos no consideran como parámetro al préstamo, con lo cual, no es un objeto candidato a participar en las asociaciones correspondientes, por el contrario, los objetos considerados en tales asociaciones son Lector y Texto pero, debido a que ninguno de los dos posee como atributos en sí mismos a la fecha en que se realiza el préstamo y la fecha de devolución, se debe crear el objeto préstamo para considerar esto. Es así como la asociación entre Texto y Lector, derivada de los estímulos, es considerada como una asociación entre Lector y Préstamo y entre Préstamo y Texto.

De esta manera, el objeto Préstamo pasa a ser una clase asociativa entre Lector y Texto. Aún así, la cardinalidad que se obtiene según la propuesta es válida para la asociación Lector-Texto, obviándose a Préstamo.

El modelo estructural para el caso de el juzgado de letras se encuentra en el Anexo 5 y en él se puede notar que en el caso de la derivación desde los HLS se encontró sólo una jerarquía que incluye a las causas con sus respectivos descendientes causas archivadas y causas terminadas. Esta jerarquía no fue considerada en el caso de OMT, ya que aquí el modelador considera dos jerarquías que incluyen a proceso judicial con sus descendientes exhorto y causa, y a exhorto con su descendiente exhorto recibido. Debido a la manera en como fue estructurado el problema, en el caso de los HLS, esta jerarquía no se hizo necesaria.

A modo de resumen cabe destacar que el modelo estructural derivado de los HLS es el mínimo que satisface los requerimientos del problema, por lo cual, puede ser considerado como un subconjunto de los modelos que puedan ser realizados directamente orientados a objetos (como la técnica OMT).

3 Paralelo entre HLS y Use Case

Al modelar según use cases el énfasis es puesto en la manera en la cual es sistema es utilizado por agentes externos (actores), sin importar cómo las funciones, a las cuales este actor tiene acceso, son desarrolladas internamente por el sistema. Desde este punto de vista, para realizar el análisis se debe comparar el use case con el diagrama de visión global y los modelos de procesos, debido a que es aquí en donde se puede ver cómo se relaciona el sistema con los agentes externos, en la forma de estímulos y respuestas enviadas entre unos y otros.

Las diferencias que interesan entre estos modelos deben ser enfocadas a las existentes entre los agentes externos y los actores del diagrama de contexto y del use case respectivamente. Esto con el fin de ver si son las mismas entidades externas las que se relacionan con el sistema y poseen requerimientos de él.

Por lo tanto, el paralelo se centra en comparar la descripción de los distintos use cases con los procesos modelados según HLS. Esto debido a que es en los procesos en donde se puede observar el tratamiento que reciben los diferentes requerimientos provenientes desde el exterior a través de estímulos, lo cual puede ser observado en la descripción de lo que el use case realiza.

3.1 Diferencias formales

Se entenderá como diferencias formales aquellas referentes a la manera en que las distintas funciones son realizadas en los procesos y en los use cases. Para encontrar estas diferencias se compararán los modelos recién nombrados para los casos de la IFIP (Anexos 1 y 2), Biblioteca Universitaria (Anexos 3 y 4) y el Juzgado de Letras (Anexos 5 y 6).

En la Tabla 3.1 se realiza un análisis de la correspondencia entre los procesos y los use cases, de manera de verificar si todos los procesos fueron considerados al modelar los use cases y cuáles son los use cases involucrados que pueden describir análogamente lo que el proceso realiza.

En el caso del proceso *Chairman_Assignment* (7), esto sólo corresponde a una parte del use case *Programar sesión*, esto debido a que en este use case además de asignar un moderador a cada sesión se escogen los artículos que serán incluidos en la misma.

El proceso *Session_Programming* (9) también corresponde a una fracción del use case *Programar sesión*, en lo referente a la creación de la sesión y la asignación de los artículos a sesiones en particular.

En todos los demás casos se puede considerar que los use cases reflejan lo que sucede en el proceso, aún cuando el use case considere funciones adicionales. Por este motivo, se rea-

liza el análisis inverso, es decir, se consideran los use cases y se analiza cuáles son los procesos que los pueden reflejar. Esto es mostrado en la Tabla 3.2.

	Proceso	Use case
1.	Paper_Submission	Registrar artículo
2.	People_To_Be_Sent_Call_For_Papers_Inclusion	Preparar lista
3.	Letter_Of_Intent_Registration	Ingresar carta de intención
4.	Call_For_Papers_Sending	Preparar lista
5.	Paper_Selection	Seleccionar artículos Listar
6.	Paper_Revision	Registrar evaluaciones
7.	Chairman_Assignment	Programar sesión Listar Revisión de evaluaciones
8.	Paper_Distribution	Distribuir artículos Listar
9.	Session_Programming	Programar sesión
10.	People_To_Invite_Inclusion	Listar personas a invitar Enviar invitaciones Listar
11.	Invitation_sending	Listar personas a invitar Enviar invitaciones Listar
12.	Priority_Invitation_Sending	Listar personas a invitar Enviar invitaciones Listar
13.	Acceptance_Of_Invitation_Registration	Registrar aceptación
14.	Author_Invitation_Sending	Listar personas a invitar Enviar invitaciones
15.	Finallist_Of_Attendees	Generar nómina final Listar
16.	Referee_Definition	-
17.	Chairman_Selection	-

Tabla 3.1 - Análisis de funciones entre procesos y use cases para la IFIP

Como se puede observar en la Tabla 3.2, todos los use cases considerados tiene su tratamiento respectivo en alguno de los procesos. El use case listar aparece entre paréntesis en alguno de los casos dado que el use case que está siendo considerado lo usa, pero no posee ningún proceso particular con el cual pueda ser comparado en forma directa.

En la Tabla 3.3 se realiza el análisis comparativo entre los procesos y los use cases que los describen, para el caso de la Biblioteca Universitaria.

	Use case	Proceso
1.	Preparar lista	People_To_Be_Sent_Call_For_Papers_Inclusion
2.	Ingresar carta de intención	Letter_Of_Intent_Registration
3.	Registrar artículo	Paper_Submission
4.	Ingresar autor	Paper_Submission
5.	Distribuir artículos (Listar)	Paper_Distribution
6.	Registrar evaluaciones	Paper_Revision
7.	Seleccionar artículos (Listar)	Paper_Selection
8.	Programar sesión (Listar)	Chairman_Assignment Session_Programming
9.	Preparar lista	Call_For_Papers_Sending
10.	Revisión de evaluaciones	Session_Programming
11.	Listar personas a invitar (Listar)	People_To_Invite_Inclusion
12.	Enviar invitaciones (Listar)	Invitation_sending Priority_Invitation_Sending Author_Invitation_Sending
13.	Registrar aceptación	Acceptance_Of_Invitation_Registration
14.	Generar nómina final (Listar)	Finallist_Of_Attendees

Tabla 3.2 - Análisis de funciones entre procesos y use cases para la IFIP

	Proceso	Use case
1.	Solicitud_De_Préstamo_De_Texto	Solicitar material Prestar material
2.	Devolución_De_Texto	Devolver material
3.	Solicitud_De_Préstamo_De_Revista	Solicitar material Prestar material
4.	Devolución_De_Revista	Devolver material
5.	Pago_De_Multa	Cobrar multas Pagar multas Reponer préstamos
6.	Inclusión_De_Lector	Agregar usuario
7.	Cambios_De_Datos_De_Lector	-
8.	Eliminación_De_Lector	Eliminar usuario
9.	Inclusión_De_Revista	Adquirir material
10.	Inclusión_De_Texto	Adquirir material
11.	Modificación_De_Datos_De_Texto	-
12.	Modificación_De_Datos_De_Revista	-
13.	Descarte_De_Texto	Descartar material
14.	Revisión_Volumen_De_Préstamos	Reportar resultados
15.	Revisión_Volumen_De_Adquisiciones	Reportar resultados
16.	Revisión_Volumen_De_Descartes	Reportar resultados
17.	Revisión_Préstamos_Satisfechos	Reportar resultados
18.	Revisión_Tiempo_Permanencia_Promedio	Reportar resultados
19.	Revisión_De_Atrasos	Reportar resultados

Tabla 3.3 - Análisis de funciones entre procesos y use cases para la Biblioteca Universitaria

El proceso Pago_De_Multa (5) corresponde a los use cases cobrar multas y pagar multa , en donde se le advierte al lector que posee una deuda, la cual debe ser cancelada para poder continuar solicitando material. El segundo caso corresponde a la situación en la cual el usuario toma la iniciativa para saber si posee deuda y cancelarla. Existe además el use case reponer préstamos, según el cual una vez pagada la multa se habilita al usuario para poder solicitar libros nuevamente.

Los procesos Solicitud_De_Préstamo_De_Texto (1) y Solicitud_De_Préstamo_De_Revista (3) corresponden a los use cases solicitar material y prestar material, en donde el lector entrega sus datos y los datos del material que solicita, una vez esto se le presta el material dependiendo si es que no posee multas y el material está disponible.

En todos los demás casos se puede considerar que los use cases reflejan lo que sucede en el proceso, aún cuando el use case considere funciones adicionales. Por este motivo, y al igual que en caso de la IFIP, se realiza el análisis inverso, es decir, se consideran los use cases y se analiza cuáles son los procesos que los pueden reflejar. Esto es mostrado en la Tabla 3.4.

	Use case	Proceso
1.	Prestar material	Solicitud_De_Préstamo_De_Texto Solicitud_De_Préstamo_De_Revista
2.	Descartar material	Descarte_De_Texto
3.	Cobrar multas	Pago_De_Multa
4.	Solicitar material (Consultar material)	-
5.	Devolver material	Devolución_De_Texto Devolución_De_Revista
6.	Pagar multas (Consultar multa)	Pago_De_Multa
7.	Adquirir material	Inclusión_De_Revista Inclusión_De_Texto
8.	Reponer préstamos	Pago_De_Multa
9.	Suspender préstamo	-
10.	Renovar suscripción	-
11.	Eliminar usuario	Eliminación_De_Lector
12.	Reportar resultados	Revisión_Volumen_De_Préstamos Revisión_Volumen_De_Adquisiciones Revisión_Volumen_De_Descartes Revisión_Préstamos_Satisfechos Revisión_Tiempo_Permanencia_Promedio Revisión_De_Atrasos
13.	Ordenar material	-
14.	Agregar usuario	Inclusión_De_Lector
15.	Eliminar usuario	Eliminación_De_Lector

Tabla 3.4 - Análisis de funciones entre use cases y procesos para la Biblioteca Universitaria

Las diferencias que se pueden encontrar son las debidas a que en el modelo de procesos se consideró a la solicitud y devolución de textos (baja demanda, alta demanda y material

de consulta) en forma separada a la de revista, por lo cual, se puede ver que en el modelo de use cases, esto es realizado a través de Prestar material (1) y Devolver material (5).

En la Tabla 3.5 se realiza el análisis comparativo entre los procesos y los use cases que los describen para el problema del Juzgado de Letras..

	Proceso	Use case
1.	Ingreso_De_Causa	Ingresar causa Ingresar tramitación
2.	Apertura_De_Diligencia	Ingresar diligencia
3.	Cierre_De_Diligencia	Terminar proceso
4.	Recepción_De_Exhorto	Ingresar exhorto
5.	Término_De_Exhorto	Terminar proceso
6.	Término_De_Tramitación	Terminar proceso
7.	Nueva_Tramitación_Por_Nuevos_Antecedentes	-
8.	Generación_De_Informe_De_Estadísticas	Informes estadísticos
9.	Elaboración_Informe_Bimensual	Informe bimensual
10.	Elaboración_Informe_Causas_Terminadas	Informe causas terminadas
11.	Generación_Informe_Corte_De_Apelaciones	-

Tabla 3.5 - Análisis de funciones entre use cases y procesos para el Juzgado de Letras

El proceso Ingreso_De_Causa (1) corresponde al use case ingresar causa e ingresar tramitación, en los cuales la causa y la tramitación son abiertas, pero con las diferencias de que al ingresar una causa en el modelo de visión global el motivo de término y las partes involucradas no son ingresadas automáticamente al sistema una vez que se ingresa el tipo de la causa, sino que debe ser ingresado por el actuario. Otra diferencia es que en el modelo use se considera el acto de abrir una causa como independiente al de la apertura de la tramitación, al contrario en el modelo de visión global se considera un acto simultáneo.

En todos los demás casos se puede considerar que los use cases reflejan lo que sucede en el proceso, aún cuando el use case considere funciones adicionales. Por este motivo, y al igual que en los dos casos anteriores, se realiza el análisis inverso, el cual es mostrado en la Tabla 3.6.

Como se puede ver la mayoría de las diferencias se refieren a aquellas relacionadas con la manera en la cual los problemas son enfocados e interpretados por las personas que los modelan. Por ejemplo, en el caso del Juzgado de Letras se dio un enfoque mucho más orientado al usuario externo, el cual no fue dado en el caso de la otra propuesta.

Se pudo ver que también existen discrepancias en cuanto a la manera misma de implementar cada proceso o función que el sistema debe manejar, lo cual no es de gran relevancia ya que sólo representa el hecho de que existen varias maneras de hacer ciertas cosas, aún cuando unas pueden ser más eficientes que las otras. Además, del hecho que los modelos use cases tienen como finalidad mostrar la manera cómo los agentes externos se relacionan con el

sistema de información, pero sin prestar relevancia en cómo estas tareas son realizadas en forma interna, es decir, implementadas.

	Use case	Proceso
1.	Ingresar causa (Identificar usuario)	Ingreso_De_Causa
2.	Ingresar tramitación (Identificar usuario)	Ingreso_De_Causa
3.	Ingresar diligencia (Identificar usuario)	Apertura_De_Diligencia
4.	Ingresar exhorto (Identificar usuario)	Recepción_De_Exhorto
5.	Terminar causa	Término_De_Tramitación
6.	Terminar diligencia	Cierre_De_Diligencia
7.	Terminar tramitación	Término_De_Tramitación
8.	Informes estadísticos (Ordenar causa)	Generación_De_Informe_De_Estadísticas
9.	Informe bimensual	Elaboración_Informe_Bimensual
10.	Informe causas terminadas	Elaboración_Informe_Causas_Terminadas
11.	Ingresar usuario (Asignar tipo usuario)	-
12.	Modificar usuario	Término_De_Exhorto
13.	Modificar/Ingresar atribuciones	-
14.	Cambiar materia de una causa (Identificar usuario, Buscar causa)	-
15.	Listar causa (Ordenar causa)	-
16.	Ordenar causa	-
17.	Emitir expediente de una causa (Identificar usuario)	-

Tabla 3.6 - Análisis de funciones entre use cases y procesos para el Juzgado de Letras

3.2 Diferencias funcionales

Se entenderá por diferencias funcionales a aquellas relacionadas con que una función tal no es considerada ya sea en el modelo de procesos o en el modelo de use cases. Se refiere también a los casos en que existan actores o agentes externos que aparecen en un caso pero en otro no, con lo cual existen funciones que dejan de ser consideradas. Cabe considerar que si el agente o el actor no se encuentra pero la función que le correspondía es considerada para otro agente o actor se trata de una diferencia formal. Estas diferencias serán desarrolladas utilizando las mismas tablas de la sección 3.1.

En el caso de la IFIP, los procesos Referee_Definition (16) y Chairman_Selection (17) que cambian los estados de los evaluadores y moderadores a disponibles y no asignados, no fueron considerados en los modelos use cases, debido a que el modelador no lo consideró un aspecto de especial relevancia.

En el caso de la Biblioteca Universitaria, los procesos Cambios_De_Datos_De_Lector (7), Modificación_De_Datos_De_Texto (11) y Modificación_De_Datos_De_Revista (12) no fueron considerados al modelar use cases, dado que el modelador no lo consideró importante.

El caso de Solicitar material (4) no es considerado en algún proceso, debido a que este use case considera que un lector cualquiera puede acceder al sistema para consultar la disponibilidad de cierto texto, lo cual es realizado automáticamente en los procesos *Solicitud_De_Préstamo_De_Texto* y *Solicitud_De_Préstamo_De_Revista*, en donde al solicitar un préstamo de algún material bibliográfico, la revisión de la disponibilidad se realiza automáticamente.

Los use cases *Suspender préstamos* y *Renovar suscripción* no son considerados como procesos dado que, en el primer caso no se consideró que el bibliotecario revisa personalmente cada uno de los lectores verificando su morosidad de libros, sino que esto es realizado simultáneamente al momento de solicitar y devolver algún material. En el segundo caso, no se consideró que en el sistema existía el registro de las suscripciones a revistas.

En el caso del Juzgado de Letras, el proceso *Nueva Tramitación Por Nuevos Antecedentes* (7) que corresponde a la reapertura de una causa debida a la entrega de nuevos antecedentes, no fue considerado en el modelo de use cases. Probablemente esto se debió a un problema de interpretación del enunciado o simplemente a que el modelador olvidó considerarlo.

Debido a que la corte de apelaciones no fue considerada como un actor en el modelo de use cases, el proceso *Generación Informe Corte De Apelaciones* (11) no posee su equivalente en algún use case.

Los use cases *ingresar usuario* (11), *modificar usuario* (12), *modificar/ingresar atribuciones* (13) e *identificar usuario* no son considerados en los procesos, esto debido a que no se consideró que un administrador del sistema ingresara tales datos, dado que el modelador no tomó en cuenta el hecho de que usuarios externos al sistema pueden utilizar el sistema del juzgado.

Los use cases *listar causas* (16) y *emitir expediente de una causa* (17) no fueron considerados para el modelo de visión global debido a que el modelo de use cases considera el ingreso de usuarios al sistema para que realice posibles consultas de procesos judiciales.

En cuanto a la consideración de los agentes externos y actores, en el caso de la IFIP, los mismos son considerados tanto en el diagrama de contexto como en los use cases, es decir, comité de programa, comité de organización y miembro de comité de programa y miembro de comité de organización, respectivamente.

En el problema de la Biblioteca Universitaria, los agentes externos encontrados en el diagrama de contexto del modelo de visión global son los bibliotecarios y el jefe de biblioteca. Los actores considerados en los use cases son el jefe de biblioteca, el usuario y el empleado. Se puede ver que el empleado corresponde al bibliotecario, por lo cual, ésta es sólo una diferencia formal.

En cuanto al usuario, al modelar con HLS no se consideró que el usuario puede relacionarse directamente con el sistema de información, es decir, se modeló pensando que es el bibliotecario con quién se relaciona el lector, siendo así el bibliotecario la persona encargada

de lidiar con el sistema. Esto se debe a posibles carencias en el enunciado del problema en donde no se distingue si el usuario puede o no acceder directamente al sistema para consultas.

En el caso del Juzgado de Letras, los agentes externos encontrados en el diagrama de contexto del modelo de visión global son el actuuario, otro juzgado, el juez y la corte de apelaciones. En el caso del modelo use case se encuentra a los actores juez, usuario, actuuario y administrador del sistema. Existe coincidencia entre el juez y el actuuario, en cuanto a los demás se puede decir que el usuario no aparece en el modelo de visión global solamente debido a una interpretación del problema, es decir, el modelador no consideró relevante modelar lo relativo a consultas de agentes externos al sistema judicial, aún cuando éstas sí pueden ser realizadas.

En lo referente al administrador del sistema, este agente externo es totalmente prescindible para el contexto en el cual se está modelando el sistema dado que no se está considerando como una parte de él a la mantención en sí del sistema, por lo cual, no interesa quién y cómo se ingresa a los usuarios al sistema, por ejemplo.

El agente externo denominado otro juzgado no aparece como un actor en los use cases, esto debido a un problema de interpretación, ya que si bien es cierto que es otro juzgado el cual envía exhortos para ser realizados en el juzgado que se está considerando, no es éste el que interactúa directamente con el sistema de información, sino que esto es realizado a través del actuuario o de un usuario del sistema. De esta manera, en términos estrictos la forma en la cual está estructurado el use case es más precisa.

En cuanto a la Corte de Apelaciones, ésta fue omitida al desarrollar el modelo use case aún cuando existen informes que se le deben ser entregados. Esto puede tener su origen en que el modelador no consideró a la corte de apelaciones como un agente de especial relevancia para el funcionamiento del Juzgado de Letras.

4 Conclusiones

Las conclusiones aquí presentadas tiene como objetivo señalar áreas en las cuales la propuesta tratada merece cierta atención, ya sea por alguna carencia o alguna observación que ella merezca.

Al realizarse el análisis comparativo entre los modelos HLS y el modelo OMT (sección 2) se pudo notar que para el caso de la Biblioteca Universitaria fue necesario agregar un ciclo de vida para el préstamo, el cual no entra al sistema como un parámetro del estímulo, por lo que según la propuesta, no debería merecer su propio ciclo de vida ni participar en las asociaciones del modelo estructural orientado a objetos. Aún así fue necesario agregarlo en tales modelos, debido a que si esto no era realizado el modelo se encontraría incompleto y no se reflejarían aspectos propios del préstamo, que ni los lectores ni el material bibliográfico pudiesen haber considerado.

Por lo tanto, se abre una ventana hacia la incorporación de variables de referencia que no correspondan a parámetros en los estímulos, para que de esta manera las caracterice un ciclo de vida y puedan así participar en asociaciones estáticas con otras variables.

En cuanto al análisis comparativo realizado entre los use cases y la propuesta se puede notar que en los modelos use cases se presenta la funcionalidad del sistema de una manera más orientada al usuario externo del sistema, que a la forma en que las funciones en sí son realizadas.

La propuesta fue comparada con el modelo OMT, el cual utiliza una notación que no es de las más recientes, por lo cual, el análisis comparativo podría ser efectuado en relación a UML (*Unified Modeling Language*), que tiene como propósito general unificar el modelamiento de sistemas, a través de ciertas notaciones y diagramas característicos. Cabe considerar que, debido a que UML es sólo una notación, es decir, no presenta una metodología para la derivación de los modelos, lo que se debería hacer es modelar el sistema mediante el procedimiento que dicta OMT, pero utilizando la notación de UML.

De esta manera, al utilizar la notación de UML se pueden incluir más modelos con los cuales extender la propuesta, como por ejemplo los diagramas de interacción (de secuencia o colaboración), los cuales representan el intercambio de mensajes entre los objetos.

Por último, la composición entre objetos que puede ser modelada mediante UML y OMT, no es considerada en la propuesta. Por composición se entiende que cierto objeto puede estar compuesto de otros, de modo que si una instancia de tal objeto es destruida, todas las instancias de los objetos que lo componen también son destruidas. Un ejemplo de composición se encuentra en el Anexo 2, sección 6, en donde la sesión está compuesta de artículos aceptados. De esta forma, se hace necesario encontrar alguna manera mediante la cual esta característica pueda ser derivada de los HLS y ser incluida en el modelo estructural orientado a objeto.

ANEXO 1 Definición del Problema de la IFIP

1 Antecedentes

Un Congreso de Trabajo de la IFIP (*International Federation for Information Processing*) es un congreso internacional que se propone reunir expertos de todos los países de la IFIP para discutir temas técnicos de interés específico para uno o más Grupos de Trabajo de la IFIP. El procedimiento más común, que será considerado para estos fines, es un congreso invitado que no es abierto para cualquiera. Para tales congresos, parte del problema es asegurar que todos los miembros relacionados con los Grupos de Trabajo y Comités Técnicos de la IFIP sean invitados, aunque no puedan asistir. Además, es importante asegurar suficientes asistentes al congreso de tal manera que la factibilidad financiera sea alcanzada sin exceder el máximo indicado por las facilidades disponibles.

Las políticas de la IFIP sobre Congresos de Trabajo sugieren la designación de un Comité de Programa para tratar el contenido técnico del congreso y un Comité de Organización para tratar las materias financieras, preparación del local, invitaciones y/o publicidad. Claramente, estos comités trabajan juntos, poseen necesidades comunes de información y necesitan mantener registros de información consistentes y actualizados.

2 Sistema de información a ser modelado

El sistema de información a ser modelado debe soportar las actividades de los Comités de Programa y Organización vinculados a la preparación de un Congreso de Trabajo de la IFIP. Es papel de estos dos comités es considerado análogo al de dos unidades organizacionales en una estructura corporativa que usan información en común.

Deben ser soportadas las siguientes actividades de los comités:

Comité de Programa:

1. Preparar una lista de personas para enviar el *call for papers*.
2. Mantener un catastro con las cartas de intención en respuesta al *call for papers*
3. Registrar los artículos de contribución recibidos.
4. Distribuir los artículos entre los que realizan la evaluación.
5. Recoger los informes de los evaluadores y seleccionar los artículos para incluirlos en el programa.
6. Agrupar los artículos aceptados en sesiones para presentación y seleccionar el moderador para cada sesión.

Comité de Organización:

1. Preparar una lista de personas que recibirán invitación para el congreso.

2. Imprimir invitaciones prioritarias para Representantes Nacionales, miembros del Grupo de trabajo y miembros e grupos de trabajos asociados.
3. Asegurar que todos los autores de artículos aceptados reciban una invitación.
4. Asegurar que todos los autores de artículos rechazados reciban una invitación.
5. Evitar el envío de invitaciones duplicadas para cualquier persona.
6. Registrar la aceptación de las invitaciones.
7. Generar la nómina final de los participantes.

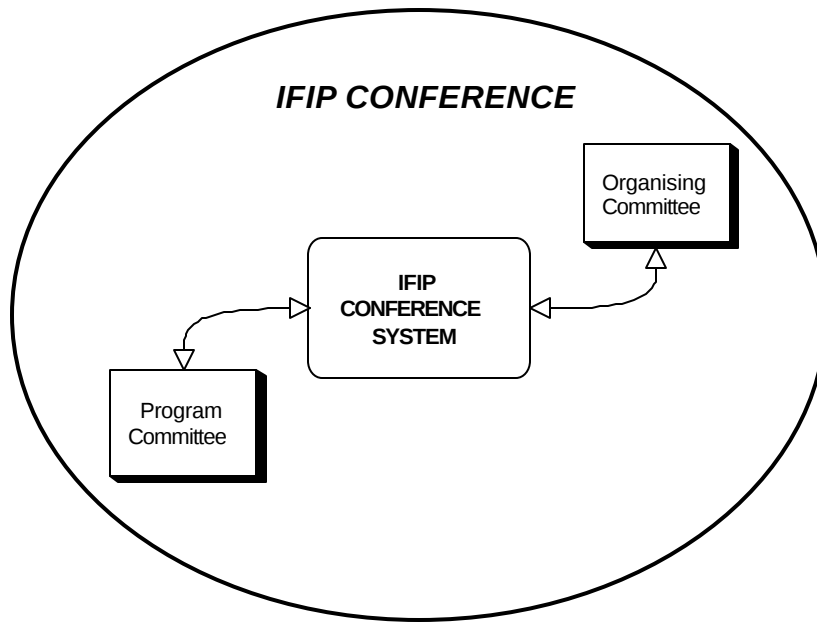
3 Fronteras del sistema

Se debe notar que los aspectos financieros y de calendario del trabajo del Comité de Organización, la planificación de ambos comités, las reservas de hoteles para los participantes y las materias relativas a la preparación de copias de los anales del congreso han sido omitidos intencionalmente de esta definición del problema.

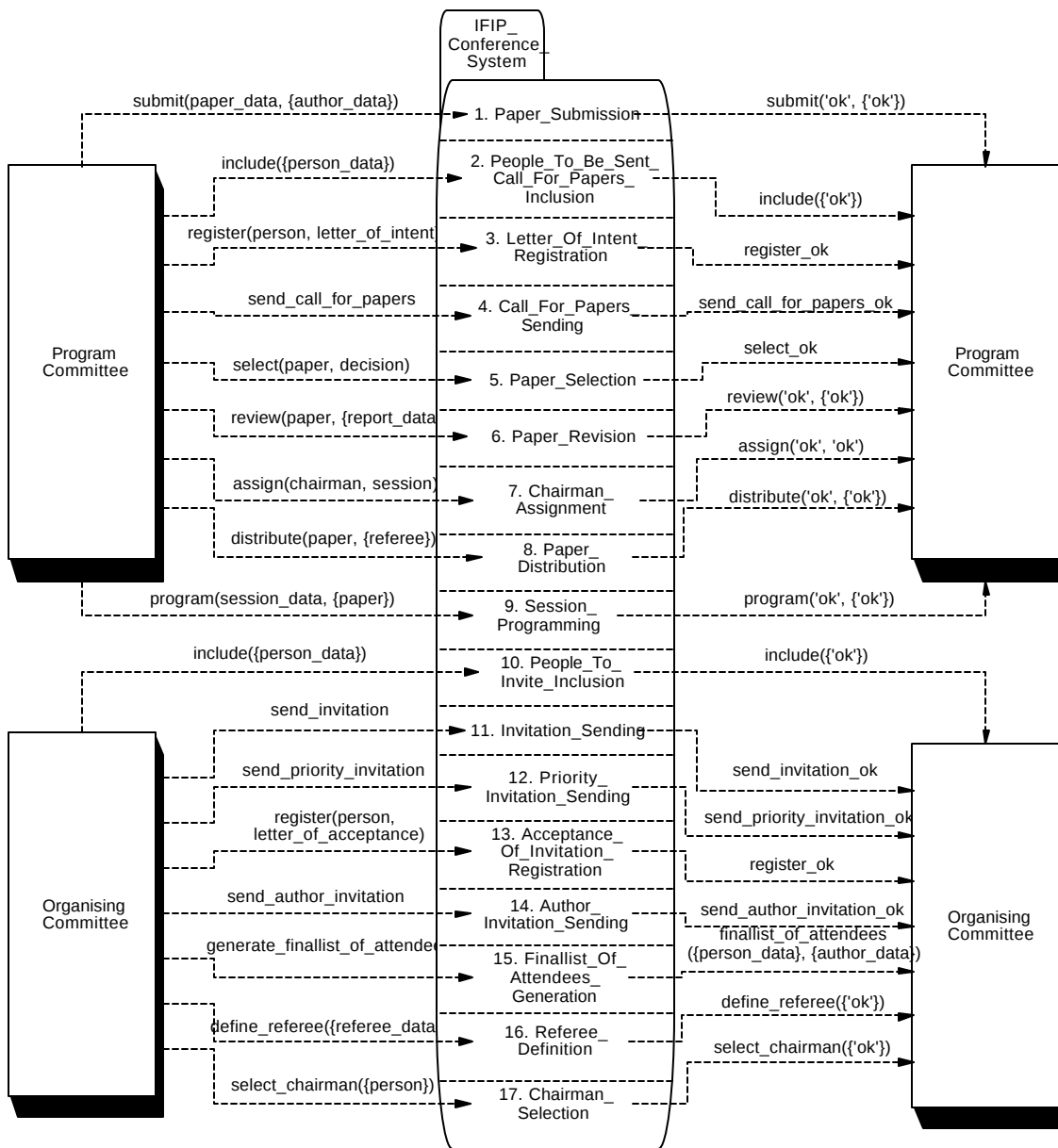
ANEXO 2 Modelo Completo del Problema de la IFIP

1 Modelo de Visión Global

1.1 Diagrama de Contexto del Sistema (en inglés)

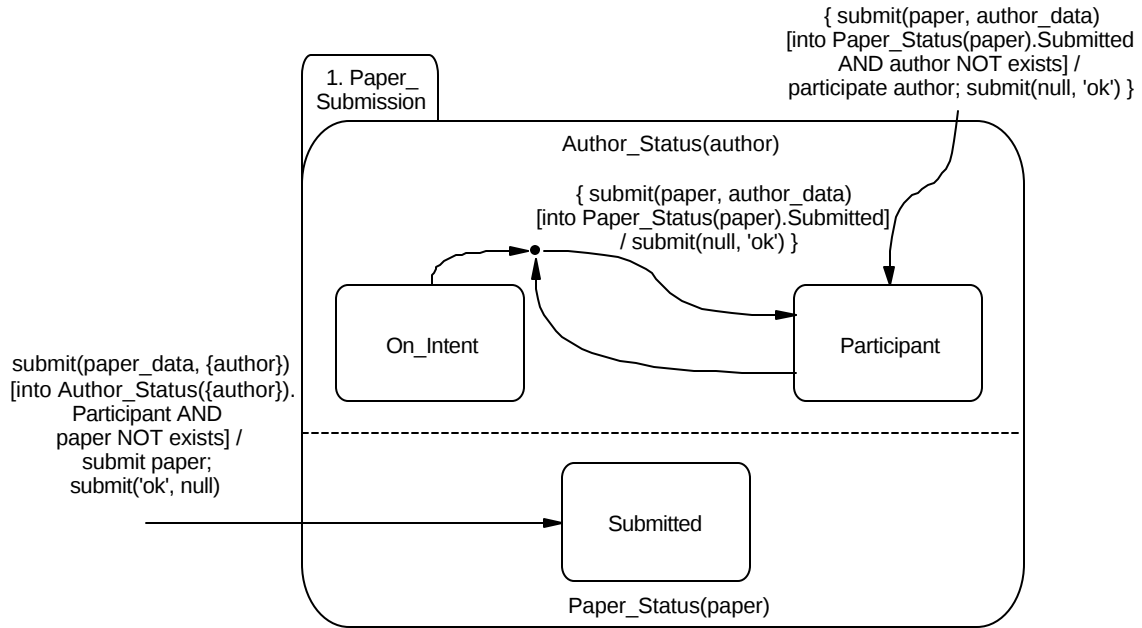


1.2 Diagrama de Visión Global del Sistema

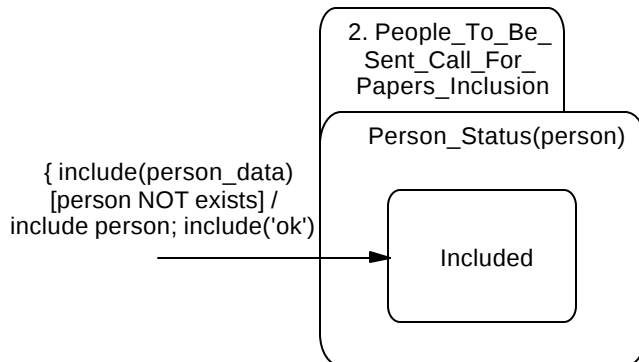


2 Modelo de Procesos

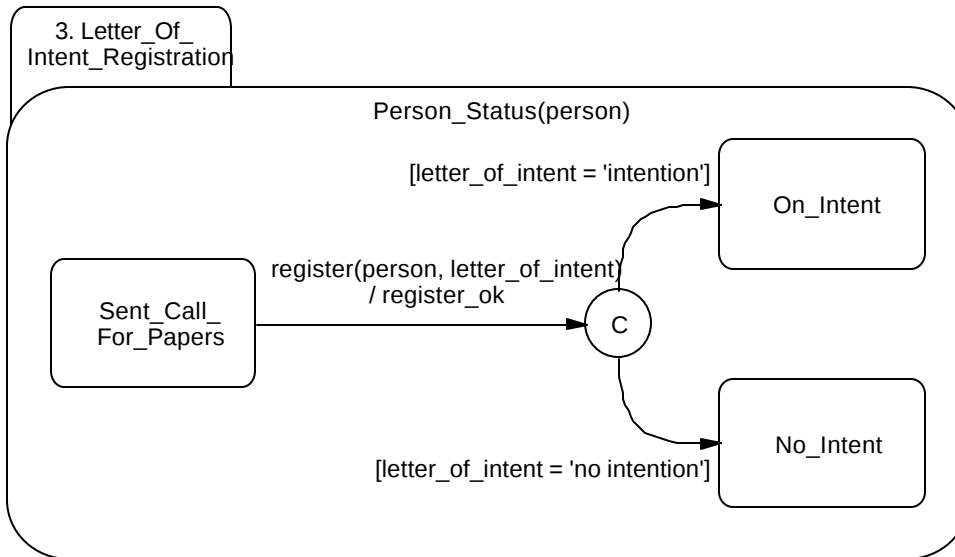
2.1 Diagrama HLS del proceso “Paper_Submission”



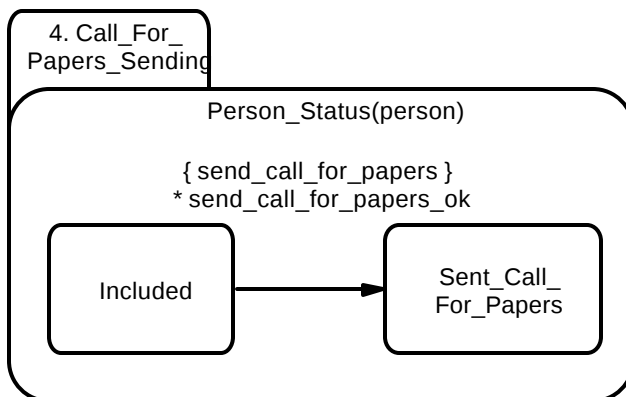
2.2 Diagrama HLS del proceso “People_To_Be_Sent_Call_For_Papers_Inclusion”



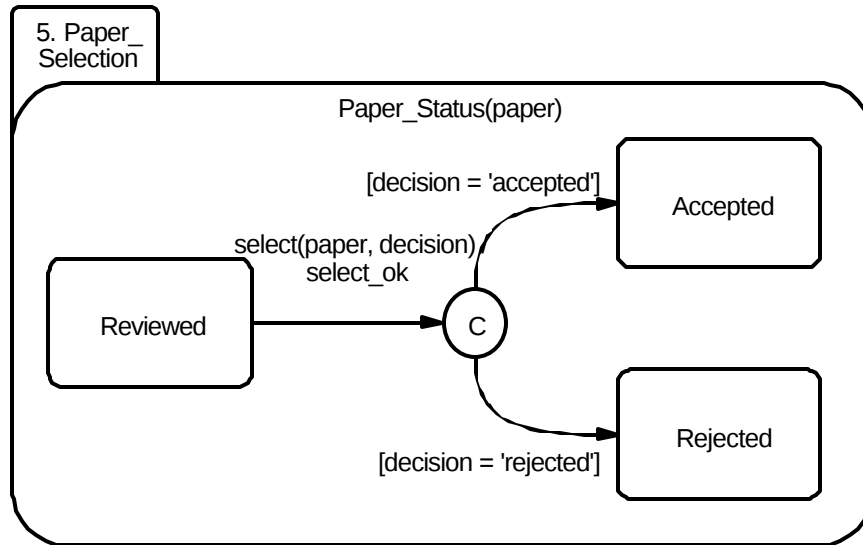
2.4.2.3 Diagrama HLS del proceso “Letter_Of_Intent_Registration”



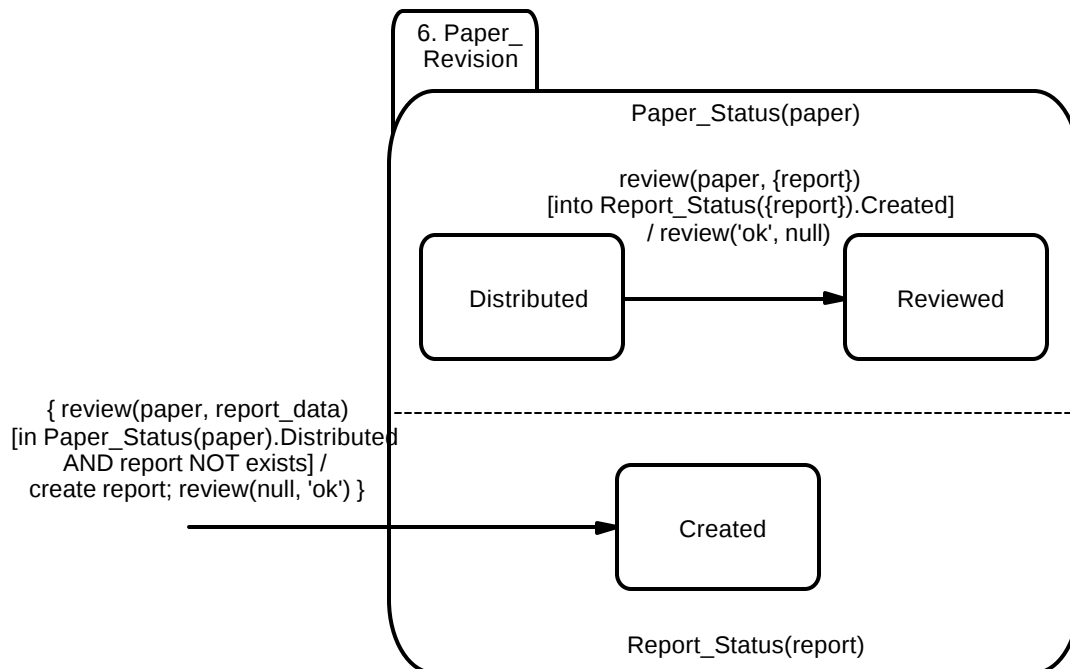
2.4 Diagrama HLS del proceso “Call_For_Papers_Sending”



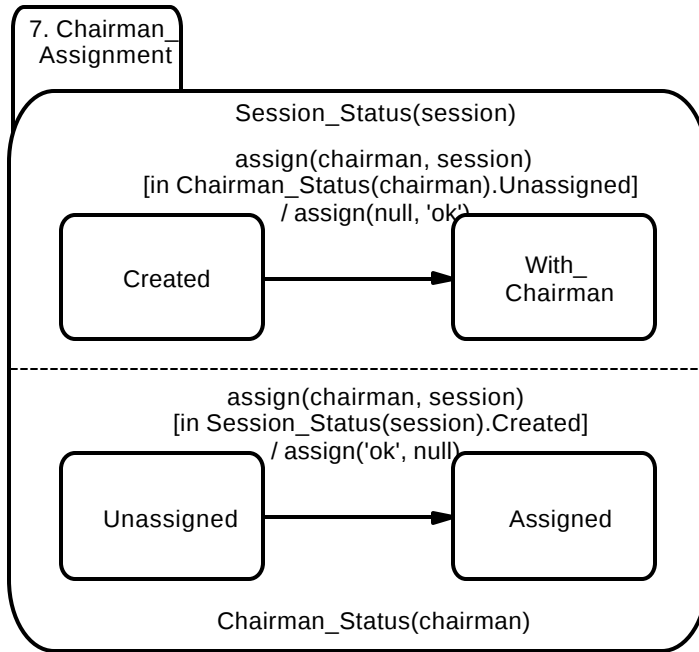
2.5 Diagrama HLS del proceso “Paper_Selection”



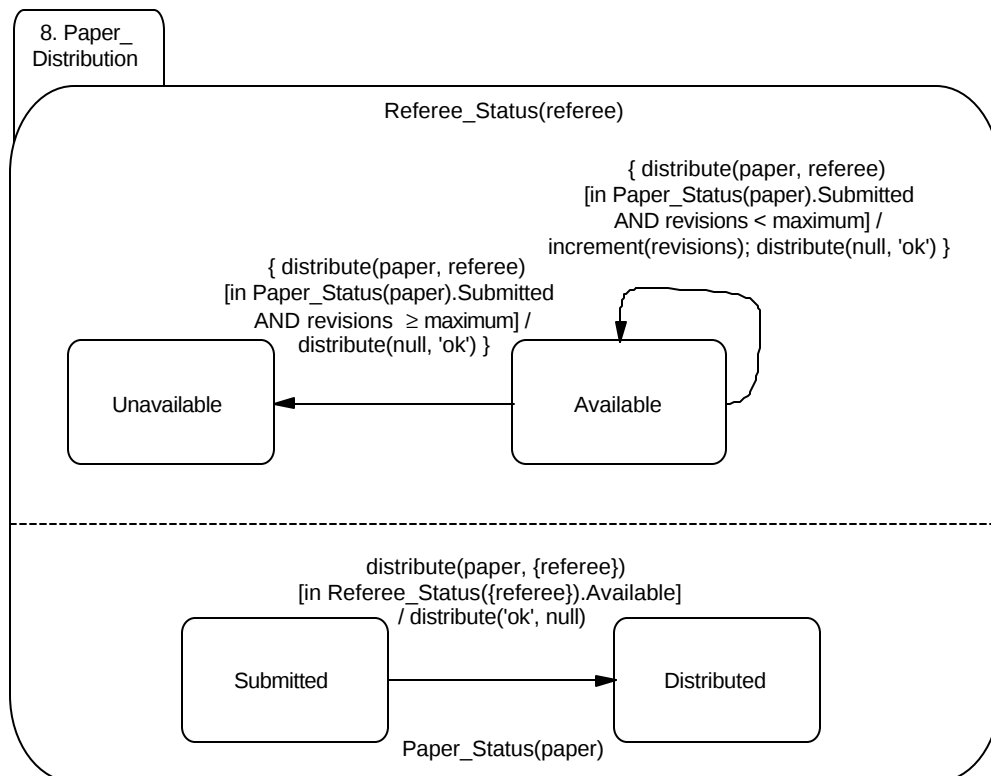
2.6 Diagrama HLS del proceso “Paper_Revision”



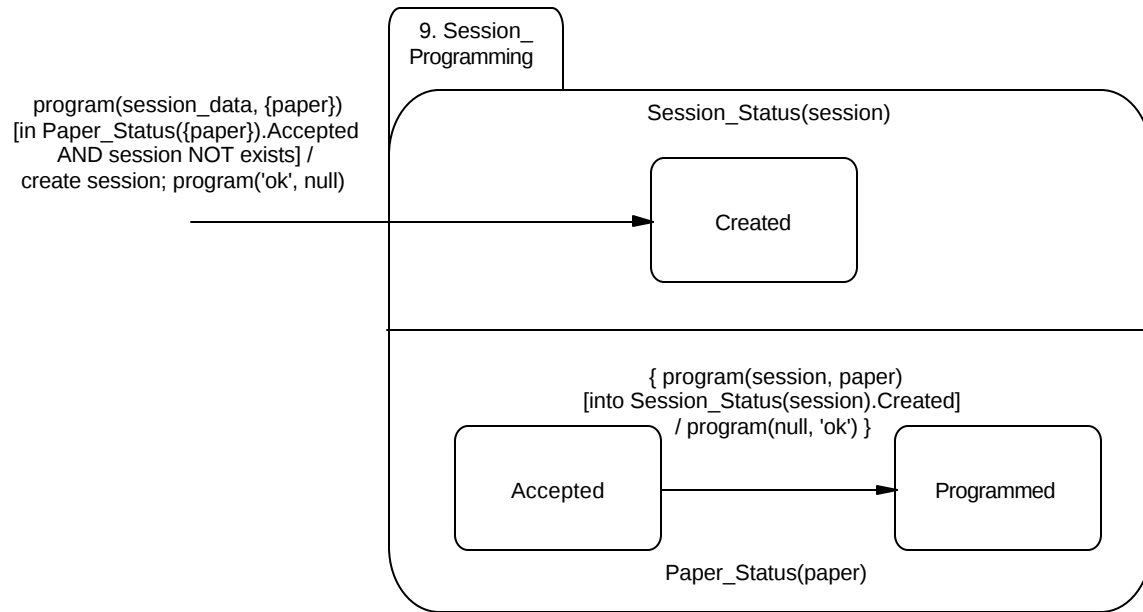
2.7 Diagrama HLS del proceso “Chairman_Assignment”



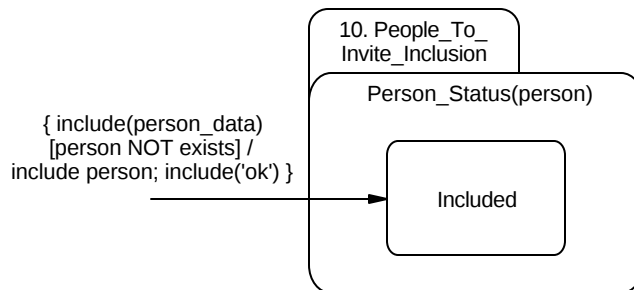
2.8 Diagrama HLS del proceso “Paper_Distribution”



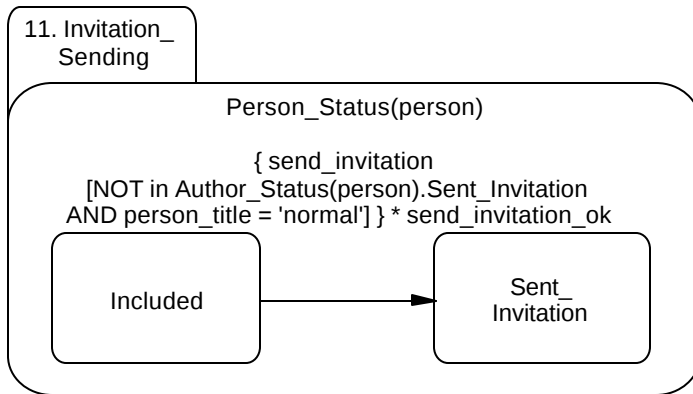
2.9 Diagrama HLS del proceso “Session_Programming”



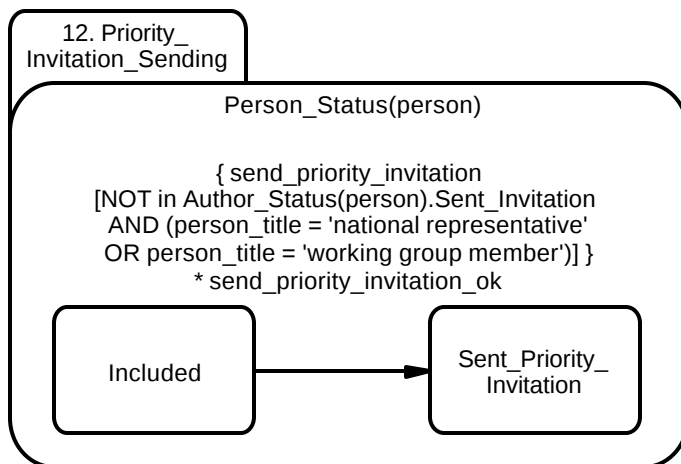
2.10 Diagrama HLS del proceso “People_To_Invite_Inclusion”



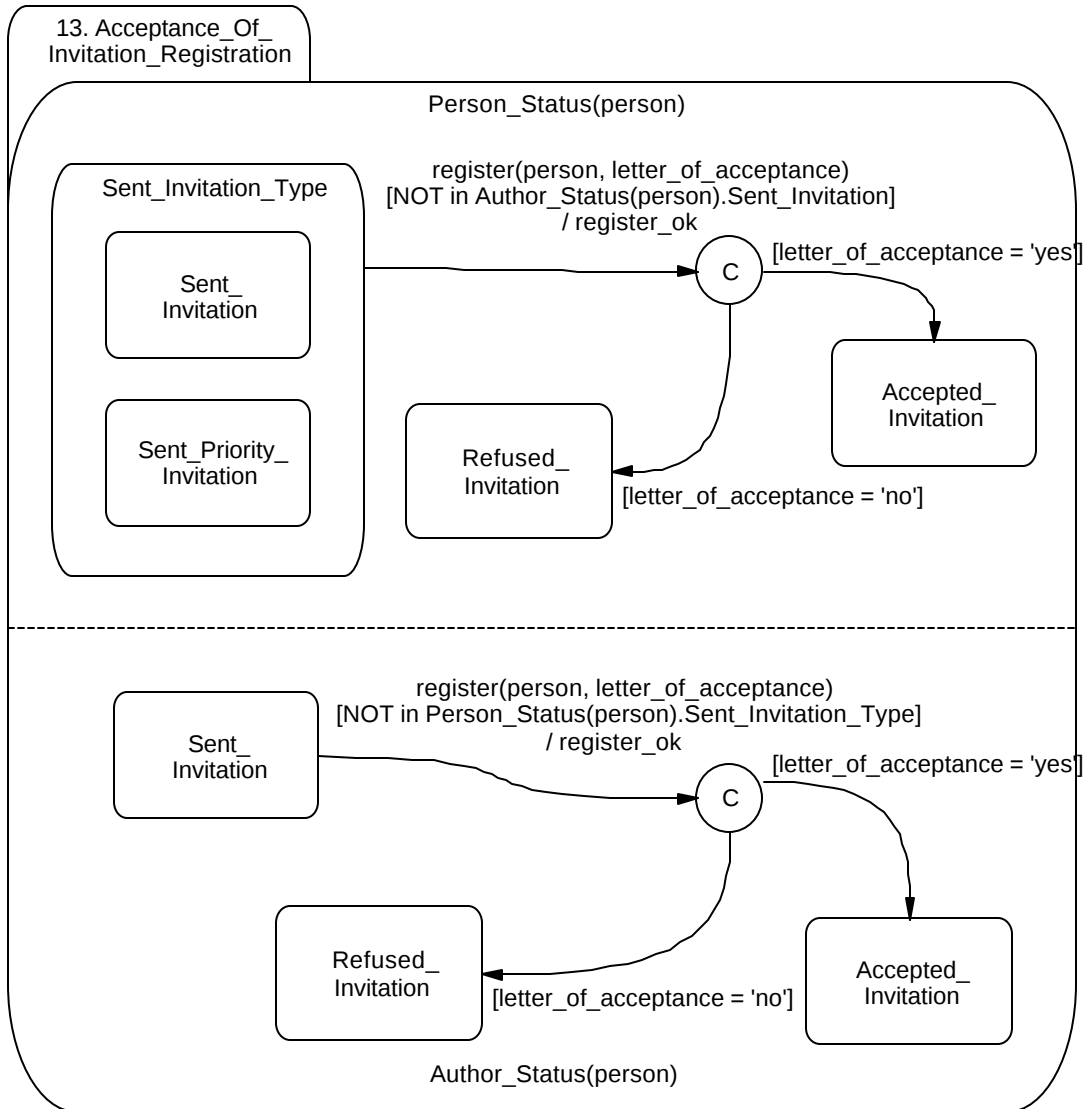
2.11 Diagrama HLS del proceso “Invitation_Sending”



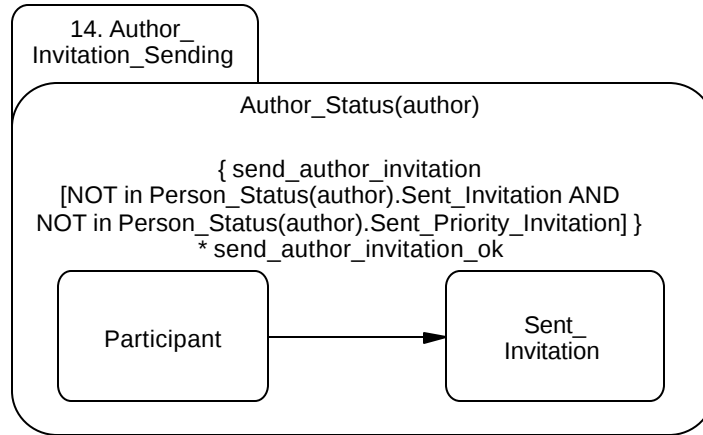
2.12 Diagrama HLS del proceso “Priority_Invitation_Sending”



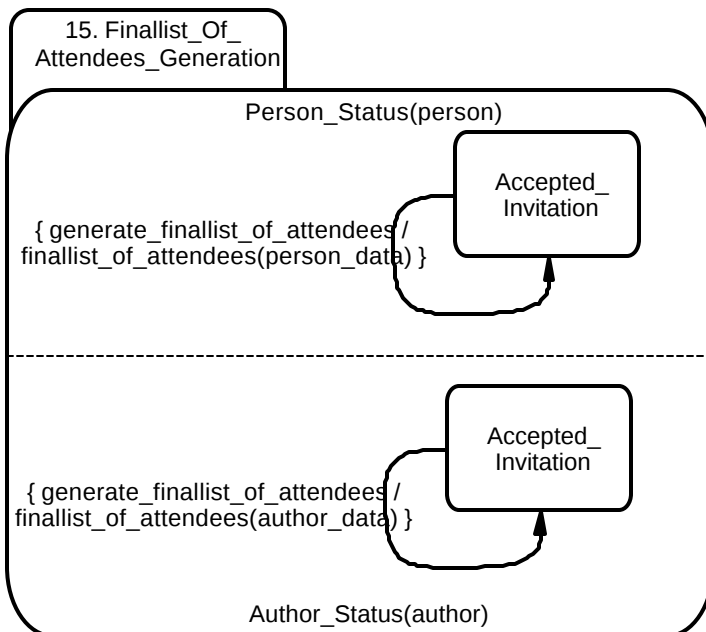
2.13 Diagrama HLS del proceso “Acceptance_Of_Invitation_Registration”



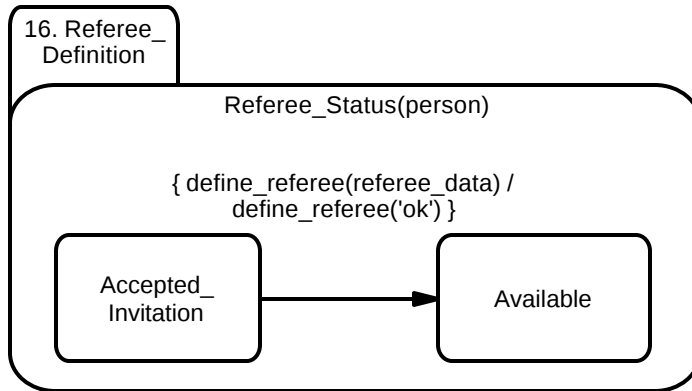
2.14 Diagrama HLS del proceso “Author_Invitation_Sending”



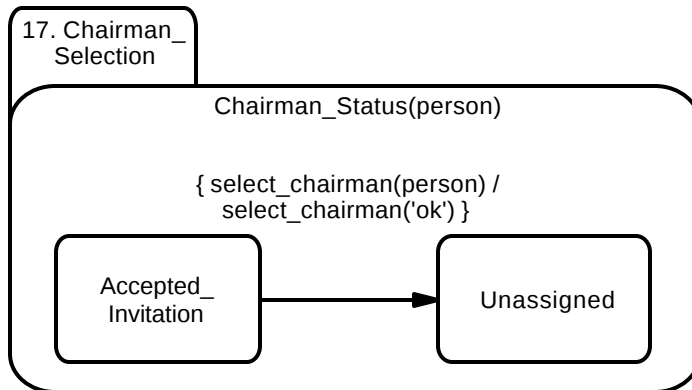
2.15 Diagrama HLS del proceso “Finallist_Of_Attendees_Generation”



2.16 Diagrama HLS del proceso “Referee_Definition”



2.17 Diagrama HLS del proceso “Chairman_Selection”



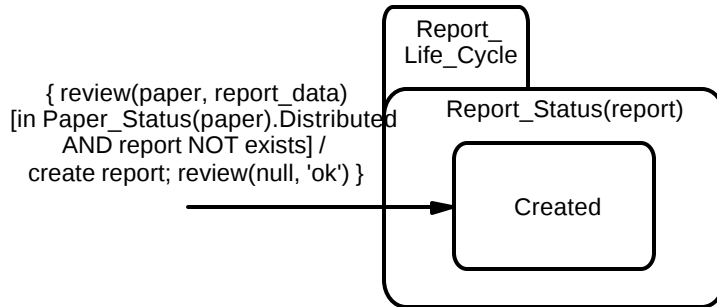
3 Diccionario de datos

DATA DICTIONARY

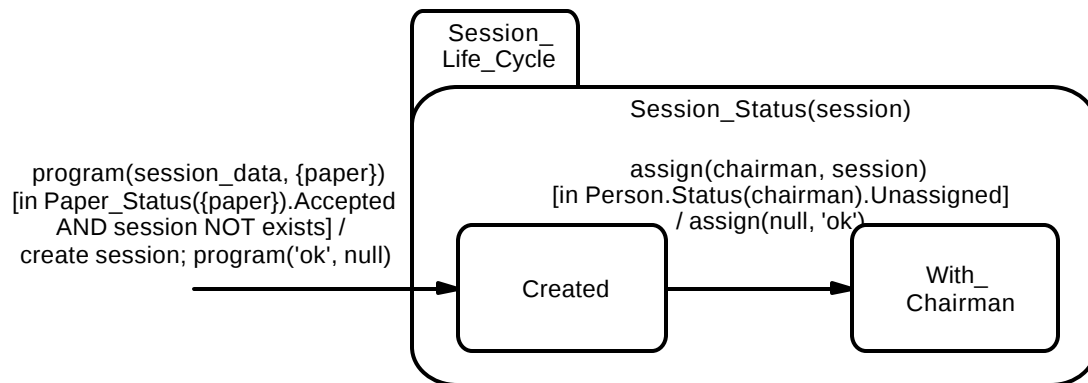
author_data = @author + name + institution + address
decision = ['accepted' | 'rejected']
letter_of_acceptance = ['yes' | 'no']
letter_of_intent = ['intention' | 'no intention']
paper_data = @paper + title + subject
person_data = @person + name + institution + person_title
referee_data = @person + maximum
report_data = @report + referee + concept
session_data = @session + room + time
person_title = ['normal' | 'national representative' | 'working group member']

4 Diagramas HLS de Ciclos de Vida

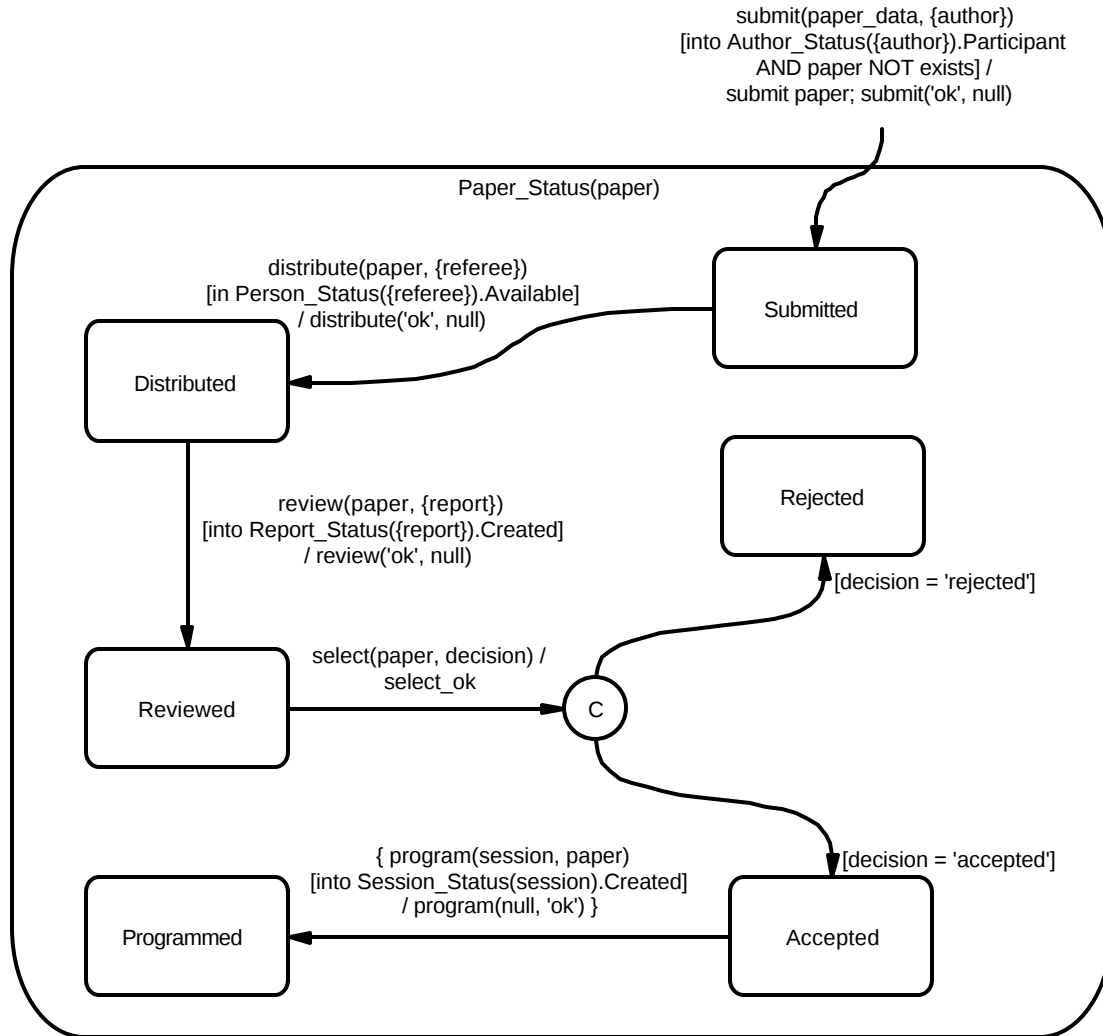
4.1 Diagrama HLS de ciclo vida de la variable report



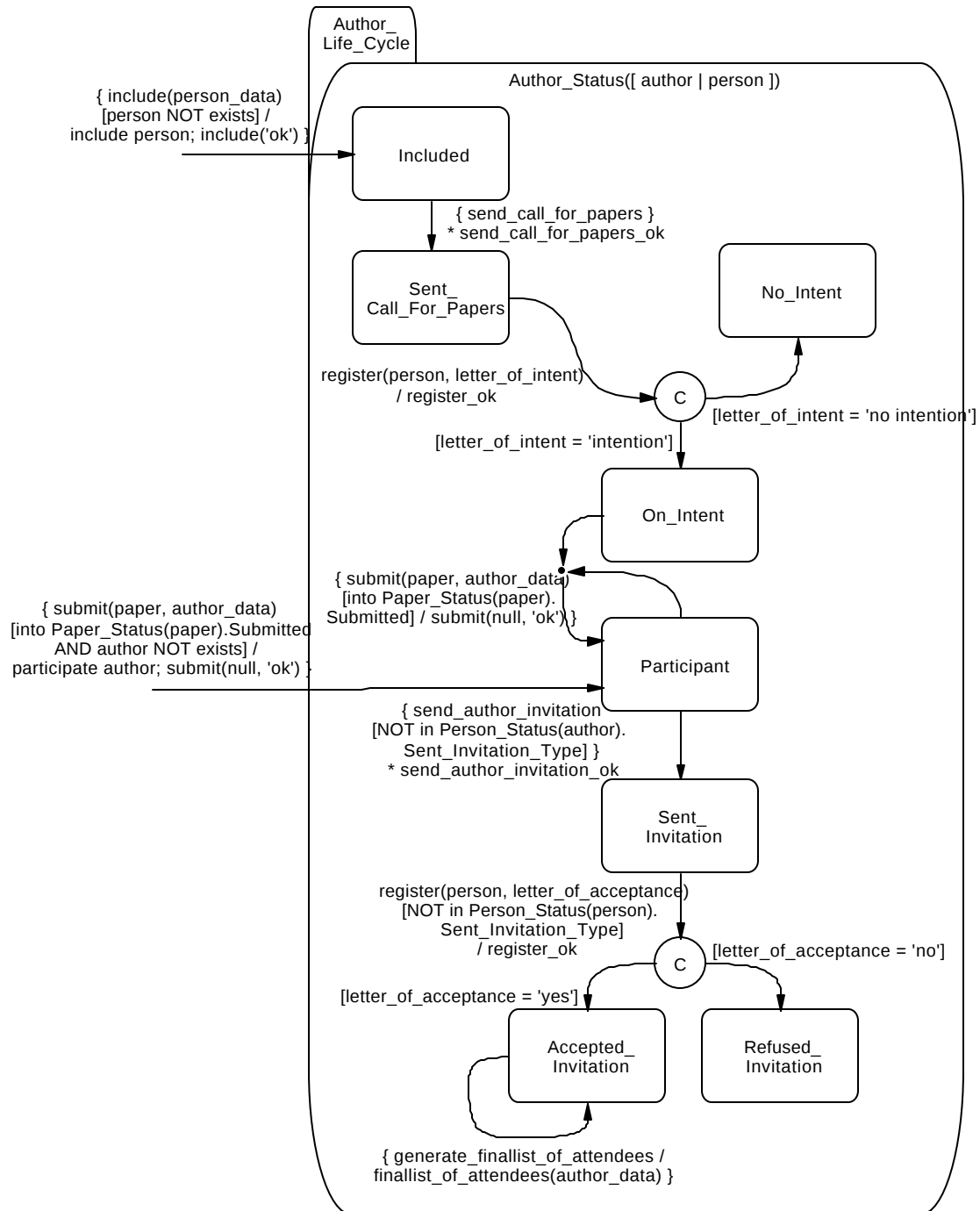
4.2 Diagrama HLS de ciclo vida de la variable session



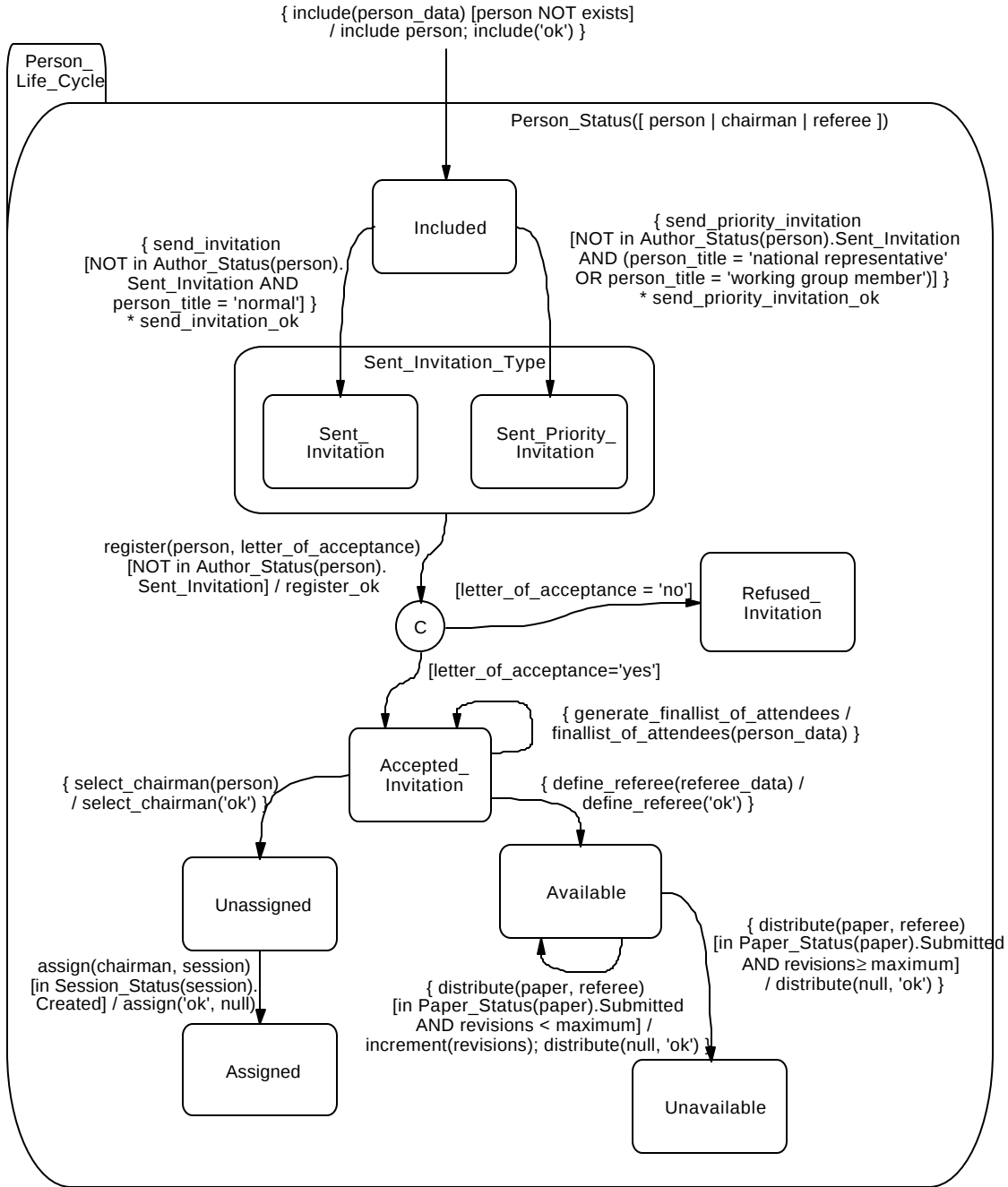
4.3 Diagrama HLS de ciclo vida de la variable paper



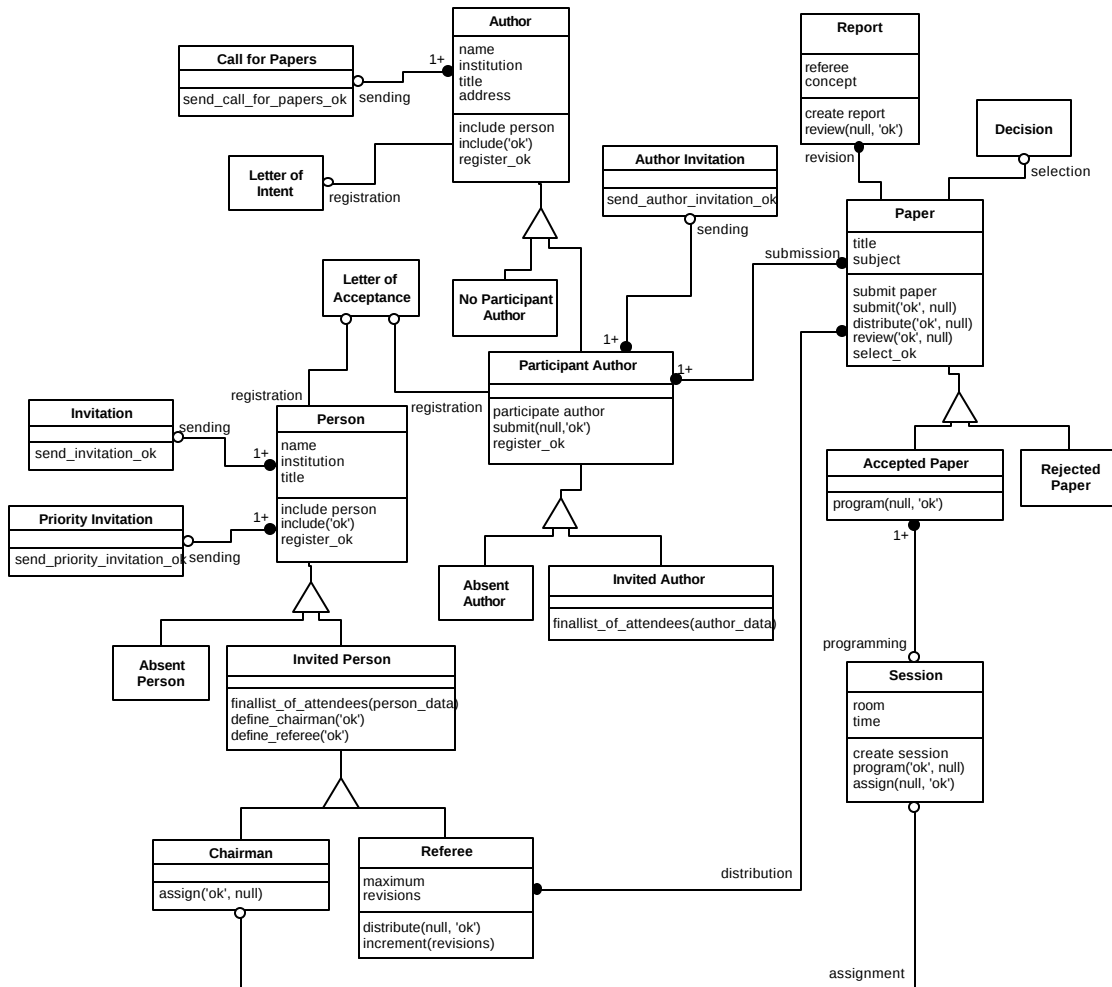
4.4 Diagrama HLS de ciclo vida de la variable author/person



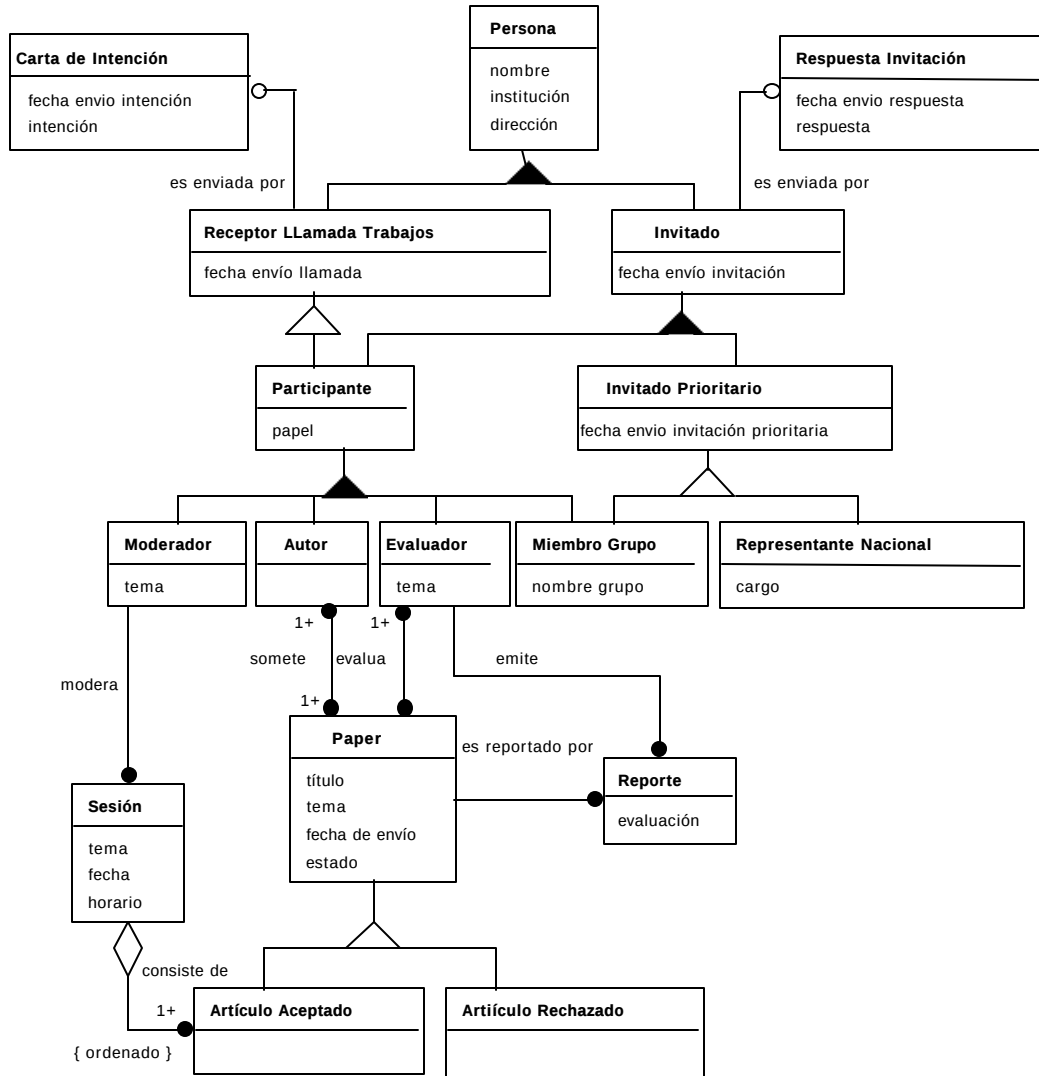
4.5 Diagrama HLS de ciclo vida de la variable person/referee/chairman



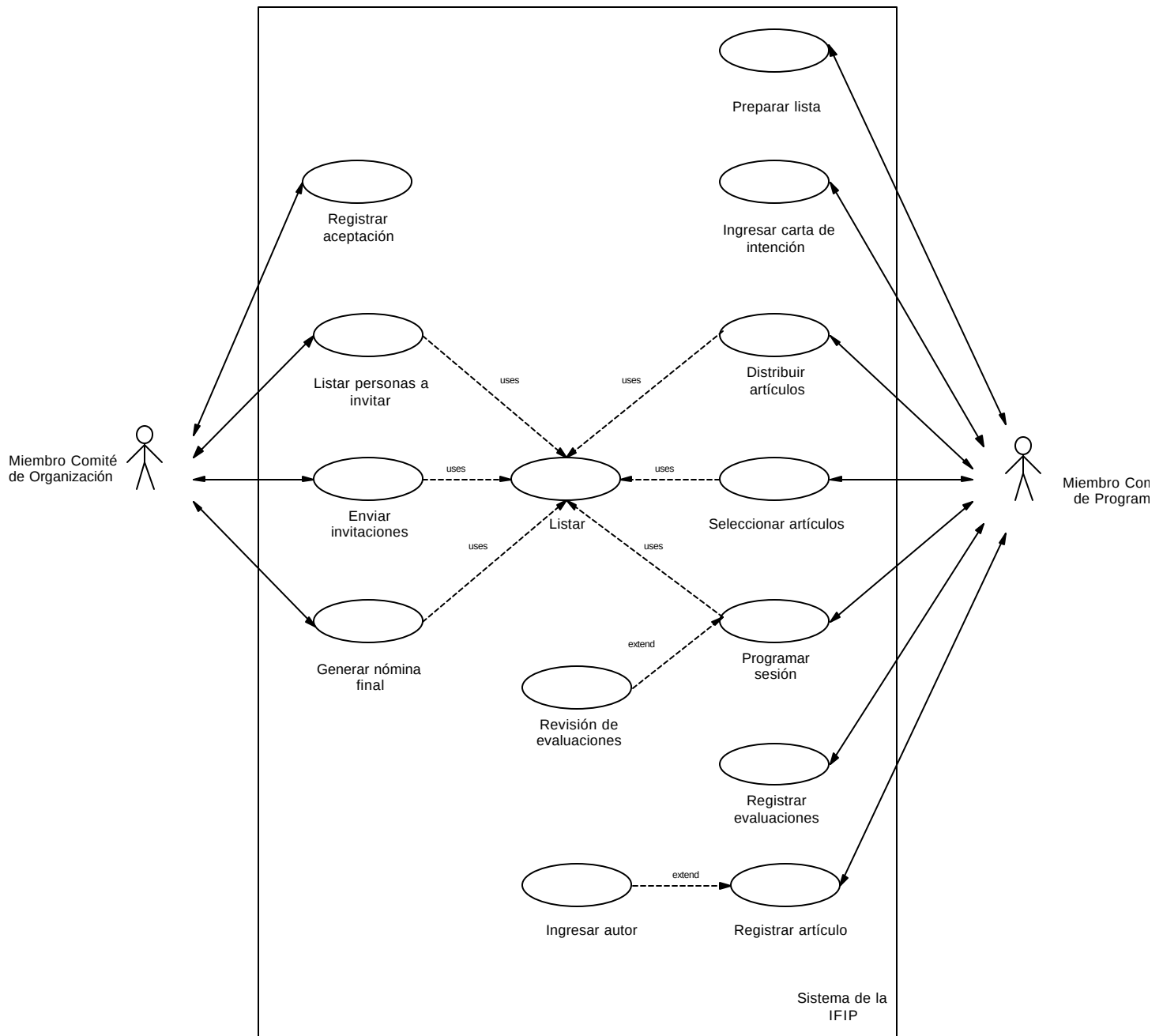
5 Modelo Estructural Orientado a Objetos



6 Modelo estructural de objetos según técnica OMT



7 Modelo use case



1. Preparar lista.

Se debe ingresar al sistema el nombre y dirección de todas las personas que ingresan a la lista de personas (participantes) a las cuales se les enviará un *call for papers*, debiéndose ingresar también un identificador del congreso en particular.

2. Ingresar carta de intención.

Una vez recibida una respuesta a una carta de intención de una persona en particular, a la cual le fue enviado un *call for papers*, se debe ingresar al sistema el deseo o no de la persona de querer participar en el congreso.

3. Registrar artículo.

Una vez recibido un artículo de contribución se debe buscar en el sistema el autor de él e ingresar el nombre del artículo. Si es que el autor no se encuentra dentro del sistema ir al use case Ingresar Autor.

3.1 Ingresar autor.

Se debe ingresar al sistema los datos personales del autor tales como nombre dirección y teléfono, entre otros.

4. Distribuir artículos.

Una vez ingresados todos los artículos al sistema deben ser distribuidos entre los evaluadores disponibles para ello. Primero listar los artículos y los evaluadores disponibles en el sistema mediante el use case Listar y la opción artículos y evaluadores. Luego mediante algún criterio asignar a cada artículo por lo menos dos evaluadores, no importando que queden evaluadores sin artículos, ingresándose al sistema que evaluador(es) evalúa cual(es) artículos(s).

5. Registrar evaluaciones.

Una vez recibidas las evaluaciones de los evaluadores ponderarlas mediante algún criterio obteniéndose una nota que lo caracteriza (1 a 5). Buscar el artículo, por su nombre o nombre de su autor, e ingresar al sistema la nota que recibió.

6. Seleccionar artículo.

Listar los artículos ya evaluados mediante el use case Listar y la opción artículos. A través de la nota que recibió decidir si se selecciona o no para el programa en curso, si esto es así se debe ingresar al sistema el artículo como aceptado en la lista de artículos aceptados. Los artículos no seleccionados son dejados momentáneamente sin estado (ni rechazados ni aceptados) hasta programar las sesiones.

7. Programar sesión.

Se debe ingresar al sistema el identificador del congreso y crear sesiones de él a las cuales son ingresados los artículos catalogados como aceptados. Ingresándose al sistema el nombre de uno o dos moderadores, dentro de los ya disponibles en el sistema y listados a través del use case Listar y la opción moderadores, para cada sesión. Los artículos no programa-

dos son catalogados como rechazados. Si es que no existieran suficientes artículos como para completar la sesión ir al use case Revisión de evaluaciones.

7.1 Revisión de evaluaciones.

A partir de los artículos evaluados y sin catalogar, escoger aquellos con mejor nota y programarlos en la sesión hasta que ésta se complete.

8. Listar personas a invitar.

Se debe ingresar a una lista de personas a invitar todas las personas listadas por el sistema, según el use case Listar, que correspondan a las siguientes opciones: representantes nacionales, miembros grupo de trabajo, miembros grupos de trabajos asociados, cartas de intención, artículos aceptados y artículos rechazados.

9. Enviar invitaciones.

Se deben listar todos los datos de las personas que se encuentran dentro de la lista de personas a invitar mediante el use case Listar y la opción persona a invitar, para luego imprimir invitaciones para cada una de ellas y enviarlas.

10. Registrar aceptación.

Una vez que se reciben respuestas a las invitaciones para participar en el congreso, se debe ingresar a la lista de participantes el nombre de la misma y la aceptación o rechazo de participar en el congreso.

11. Generar nómina final.

e debe listar los nombres de todos las personas presentes en la lista de personas a invitar que aceptaron tal invitación para participar en el congreso, mediante el use case Listar y la opción participantes.

12. Listar

Se debe ingresar al sistema el identificador del congreso y la opción a listar de entre las disponibles (*call for papers*, artículos, evaluadores, moderadores, personas a invitar, representantes nacionales, miembros grupo de trabajo, miembros grupos de trabajos asociados, artículos aceptados, artículos rechazados, cartas de intención, personas a invitar y participantes), listándose por pantalla o por impresora, los resultados.

ANEXO 3 Definición del Problema de la Biblioteca Universitaria

1 Antecedentes

Una biblioteca es una instancia de servicio que tiene un papel fundamental en instituciones de educación superior. Su propósito puede concebirse desde una simple entidad que disponibiliza información para los usuarios, hasta uno en que desempeña un rol activo en la formación de personas.

El recurso bibliográfico debe ser administrado por la biblioteca con conceptos de equidad y eficiencia, es decir, que pueda entregarse a todo aquel que lo requiera y al mismo tiempo que tal recurso no sea desperdiciado. No es de extrañar entonces que los dos principales objetivos que pueden asignarse a una biblioteca son los de maximizar la satisfacción de todos los usuarios, independientemente de sus necesidades bibliográficas, como así también el de maximizar la utilización de este recurso.

Desde el punto de vista del usuario, parece ser la función de préstamo la única que justifica la existencia de una biblioteca, sin embargo, existen diversas otras funciones que la apoyan y que no por ser menos visibles son menos importantes. Entre estas funciones se encuentran las de compra y clasificación del material nuevo, como así el análisis y descarte del material no utilizado.

2 Sistema de información a ser modelado

El sistema de información a ser modelado considera principalmente las funciones de préstamo de libros de alta y baja demanda, material de consulta y la hemeroteca (préstamo de revistas). También se consideran las funciones de adquisición y descarte del material bibliográfico.

Deben mantenerse los datos sobre cada libro en particular que forma parte del acervo. Consignando todos sus datos se facilita posteriormente la consulta de los mismos ya sea por código (usando el sistema decimal de materias de Dewey), autor, título, número de inventario, tema, año, editora, otros y combinación de las anteriores. Todo esto es válido para los libros de alta y baja demanda y el material de consulta.

Aunque no es muy común en los libros de baja demanda y en el material de consulta, existen varios títulos que poseen más de una copia. También es posible encontrar versiones del mismo libro en diferentes idiomas (lo más común es en inglés y en español) o aún contar con diferentes ediciones del título. Eventualmente existen por ejemplo, títulos traducidos de una edición anterior simultáneamente con la nueva edición en el idioma original.

Los plazos de préstamo para los libros de baja demanda varían según el tipo de usuario.

Para académicos, los libros se prestan por 30 días corridos, siendo que no tienen límites de cuántos libros pueden tener en préstamo. Los alumnos y funcionarios en general tienen un plazo de entrega de 7 días corridos, siendo que no pueden tener en su poder más de 5 libros al mismo tiempo. La excepción son los alumnos memoristas a los que se le quita la restricción del número de libros.

Bajo el régimen de préstamo de libros de alta demanda, éstos deben ser devueltos hasta el mediodía del día hábil siguiente y sólo se prestan “para la casa” a partir de esa misma hora. En las mañanas el préstamo es en sala. Este tipo de préstamo es sólo para los alumnos (incluyendo memoristas).

Con respecto a los libros en este régimen de préstamo, siempre se tiende a tener la última versión en español, sin embargo, si existe una versión más reciente en el idioma original se incorporan algunas copias a este tipo de préstamo.

Dado el intenso uso de libros, éstos se deben mantener en general con tapas duras y encuadernación cada vez que lo requieran. Esto les permite una mayor durabilidad. Lo mismo ocurre con aquellos libros de baja demanda pero mal tratados.

Las multas por atraso de entrega para todos los libros se fijan en base al valor de 2 pasajes de locomoción colectiva. A partir del tercer día de atraso se cobra una multa adicional fija de 10 pasajes. La multa adicional no se aplica para el caso de los libros de baja demanda. Mientras un alumno no haya devuelto sus libros ni cancelado sus multas tiene los préstamos suspendidos en la biblioteca, salvo indicación de lo contrario del jefe de biblioteca.

Existe también el denominado material de consulta que sólo se presta en sala. Este material corresponde generalmente a catálogos. Diccionarios, directorios, índices y afines que sirven como medio rápido de consulta.

La hemeroteca permite la libre consulta de material en sus aulas (como en realidad sucede con todo el material con la única restricción de los libros de alta demanda). Sin embargo, las revistas se prestan (hasta un máximo de 10 ejemplares) bajo el mismo régimen de los libros de alta demanda.

En la hemeroteca existen diversas publicaciones periódicas: diarios, semanales, quincenales, mensuales, bimensuales, etc. Existen algunas publicaciones que eventualmente funden dos o más ediciones en un solo ejemplar de mayor porte. La biblioteca debe mantener toda la información de la publicación y debe estar atenta a los plazos para renovar las suscripciones.

El orden de todo el material es hecho por tipo:

- los libros de alta demanda son ordenados por autor en una sección separada y sin acceso al público;

- los libros de baja demanda son distribuidos, de acuerdo a la clasificación de materias y dentro de cada materia por autor, en estanterías de acceso libre al público;
- el material de consulta es ordenado por título en una sección separada, también de acceso libre al público; y
- las revistas son separadas por tema y ordenadas por título y en forma cronológica en estanterías de acceso restringido al público.

La adquisición de nuevos textos y el descarte anual de textos se hace en base al movimiento histórico de préstamos. Adicionalmente, académicos pueden indicar la compra de textos nuevos y también sugerir la inclusión de textos bajo el régimen de alta demanda. La biblioteca se reserva el derecho de ampliar el número de copias de algún texto si así lo amerita la demanda. Las compras son hechas directamente por la biblioteca con indicación a contabilidad para realizar los pagos. Lo mismo sucede con la suscripción de nuevas revistas y las renovaciones periódicas.

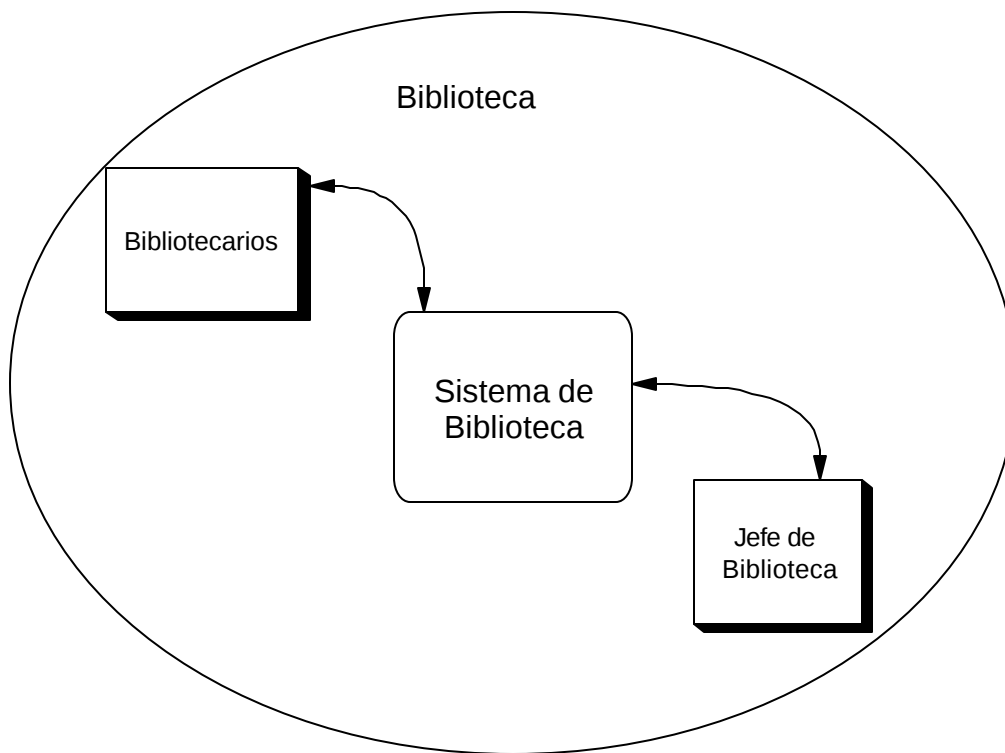
Libros candidatos a descarte son aquéllos que no han sido prestados durante el último año. Sin embargo, se consulta a los académicos del tema en cuestión para confirmar si pueden descartarse. Los libros descartados que aún se mantienen en buen estado son donados a bibliotecas públicas de la región. Las revistas no son dadas de baja salvo que su estado general así lo exija.

El jefe de biblioteca debe conocer la actividad global que se lleva a cabo en biblioteca. En este sentido le sería muy útil contar con información de cuál es el volumen de préstamos, adquisiciones y descartes, basándose en el tipo de usuario, tipo de préstamo y para cualquier periodo que solicite. Asimismo le interesa conocer la proporción de solicitudes de préstamo efectivamente satisfechas y el tiempo de permanencia promedio de los distintos tipos de recurso bibliográfico y para diferentes periodos de tiempo. Con esta información le es posible asignar mejor los recursos humanos, financieros y materiales con que cuenta la biblioteca para el logro de sus objetivos.

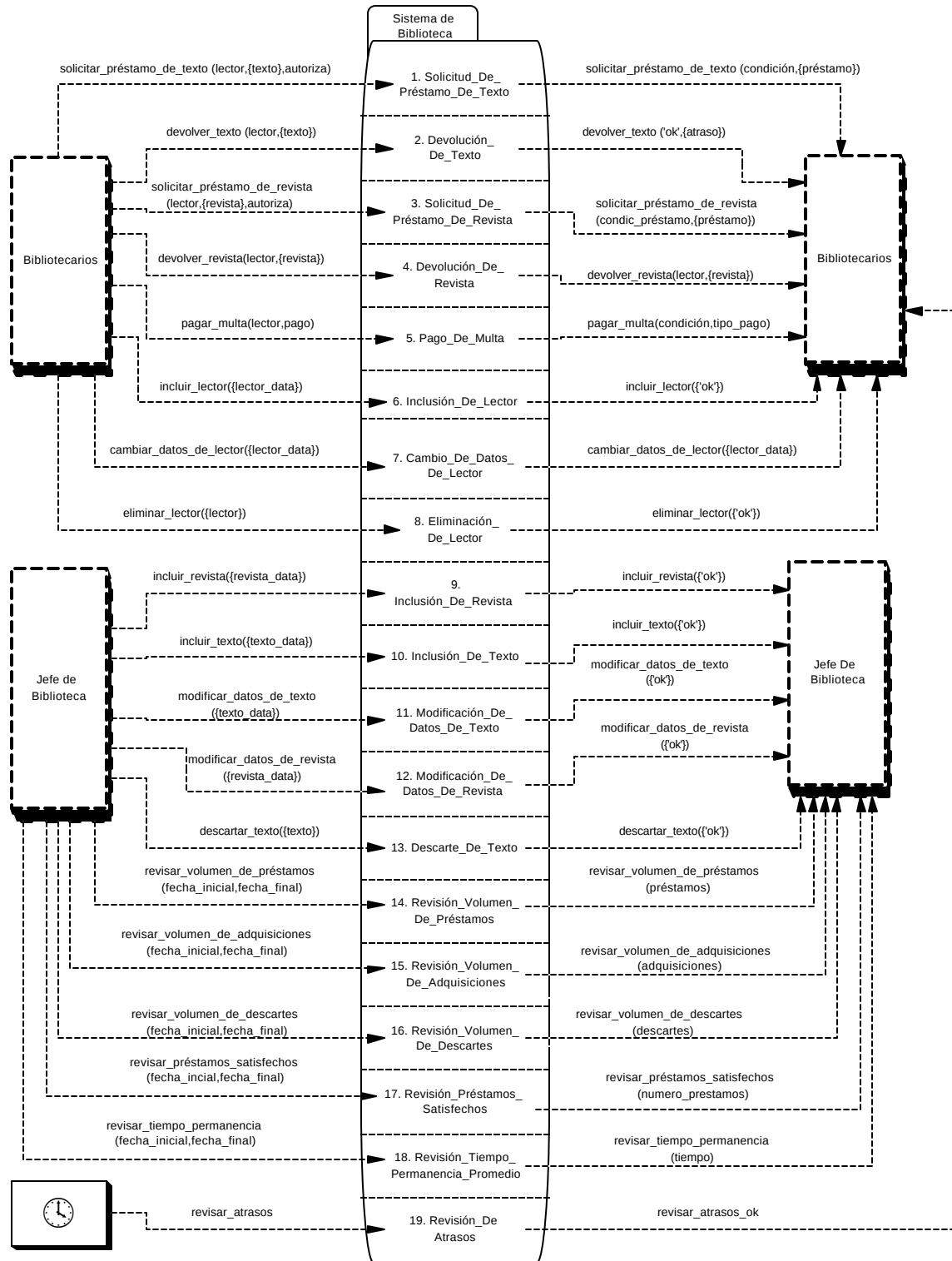
ANEXO 4 Modelo Completo del Problema de la Biblioteca Universitaria

1 Modelo de Visión Global

1.1 Diagrama de Contexto del Sistema

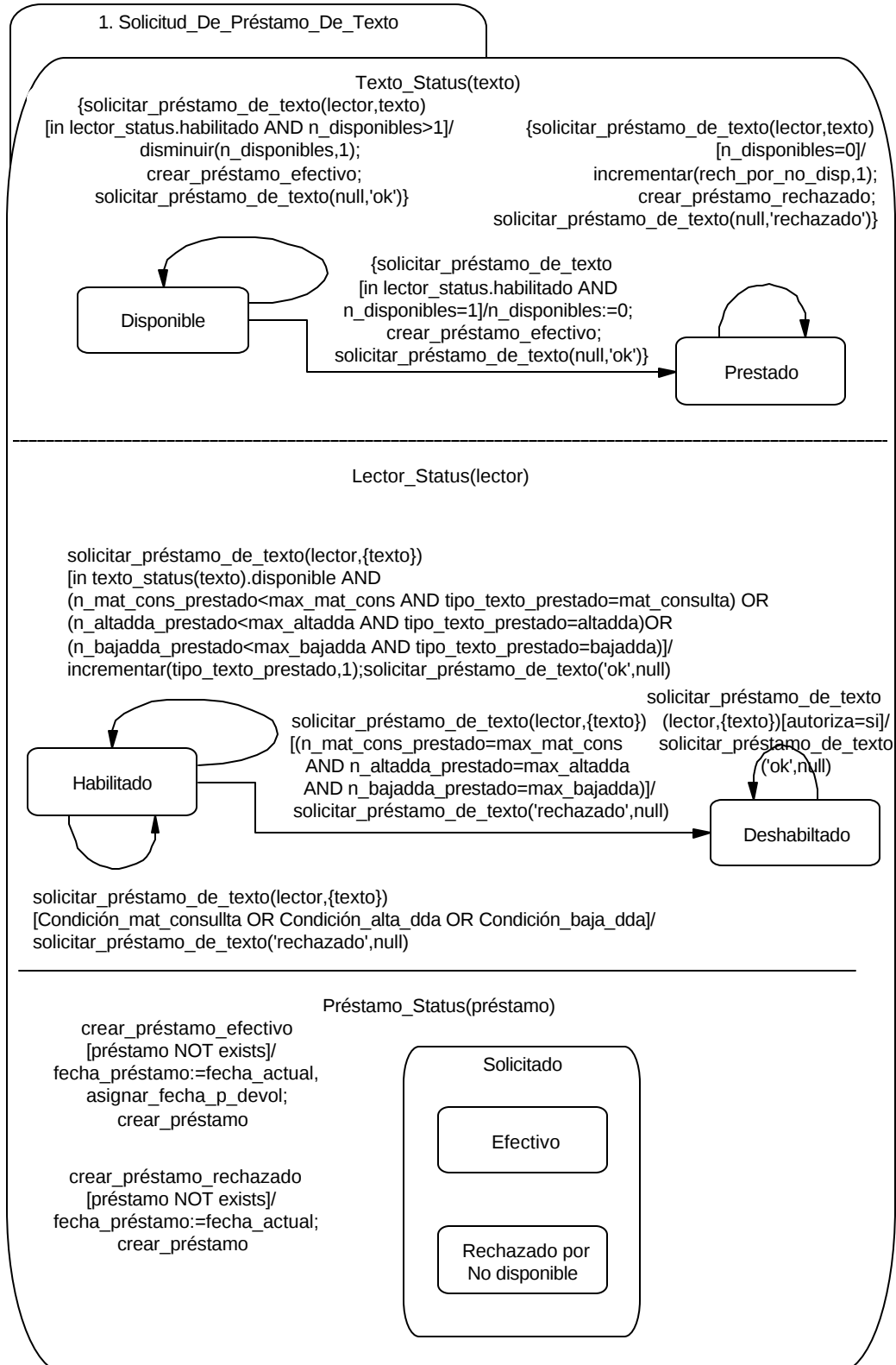


1.2 Diagrama de Visión Global del Sistema

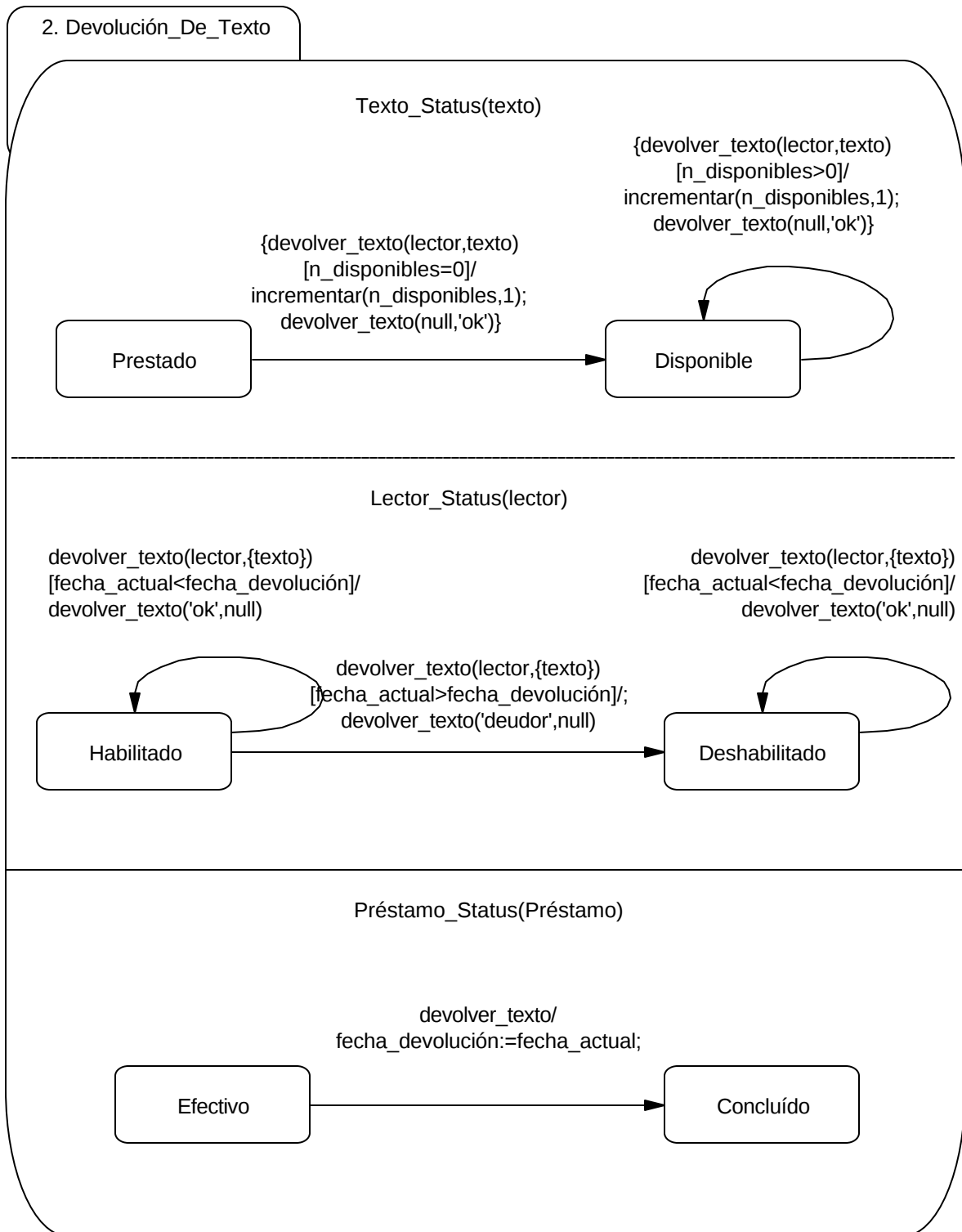


2 Modelo de Procesos

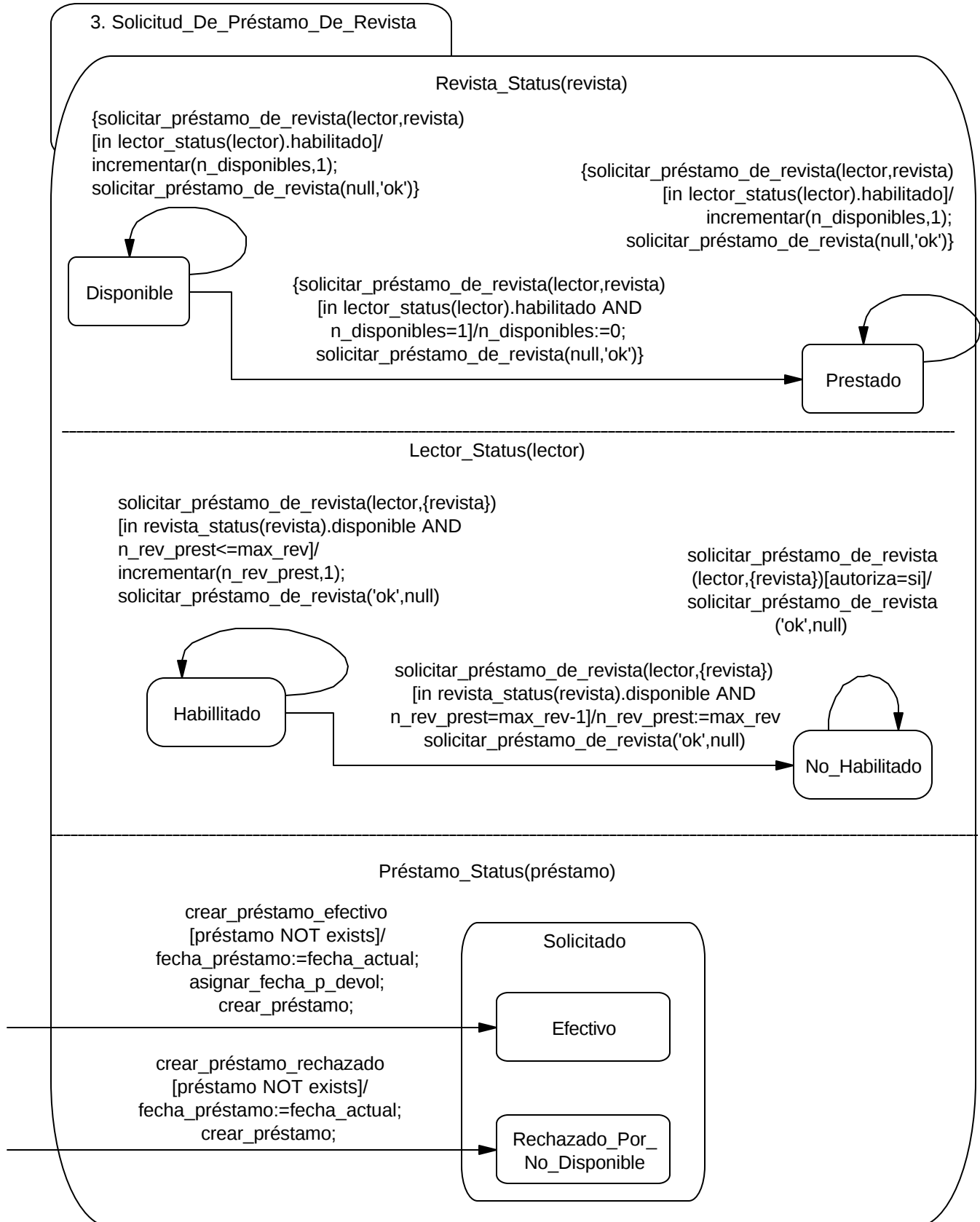
2.1 Diagrama HLS del proceso “Solicitud_De_Préstamo_De_Texto”



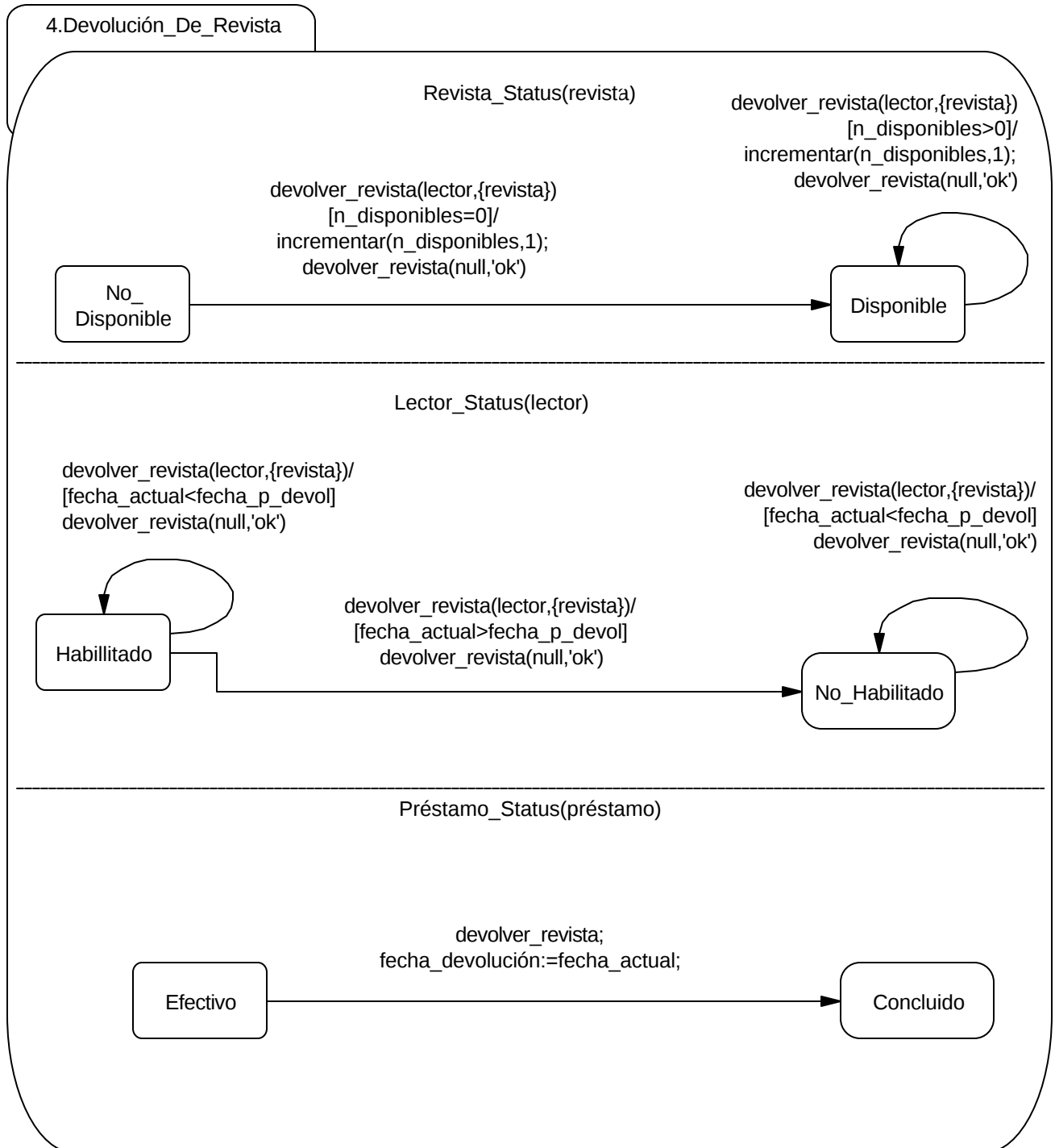
2.2 Diagrama HLS del proceso “Devolución_Texto”



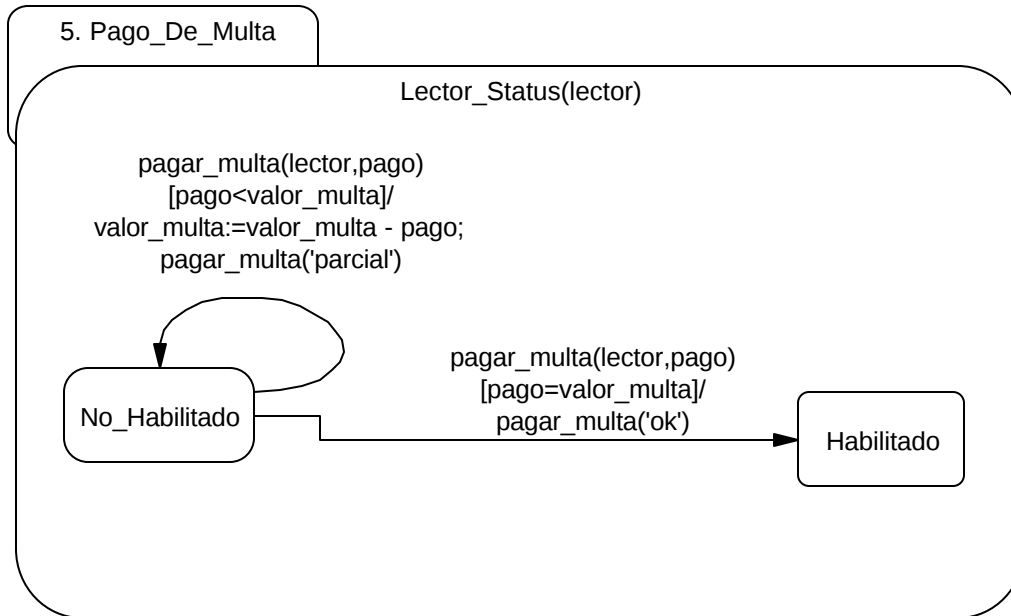
2.3 Diagrama HLS del proceso “Solicitud_De_Préstamo_De_Revista”



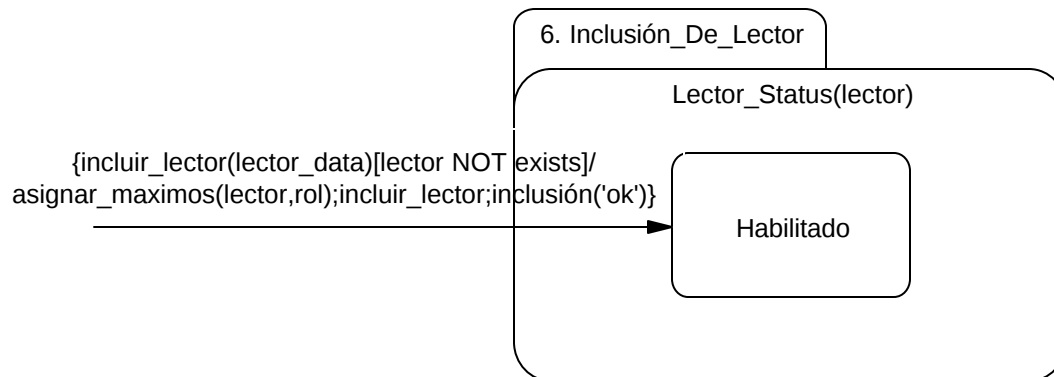
4 Diagrama HLS del proceso “Devolución_De_Revista”



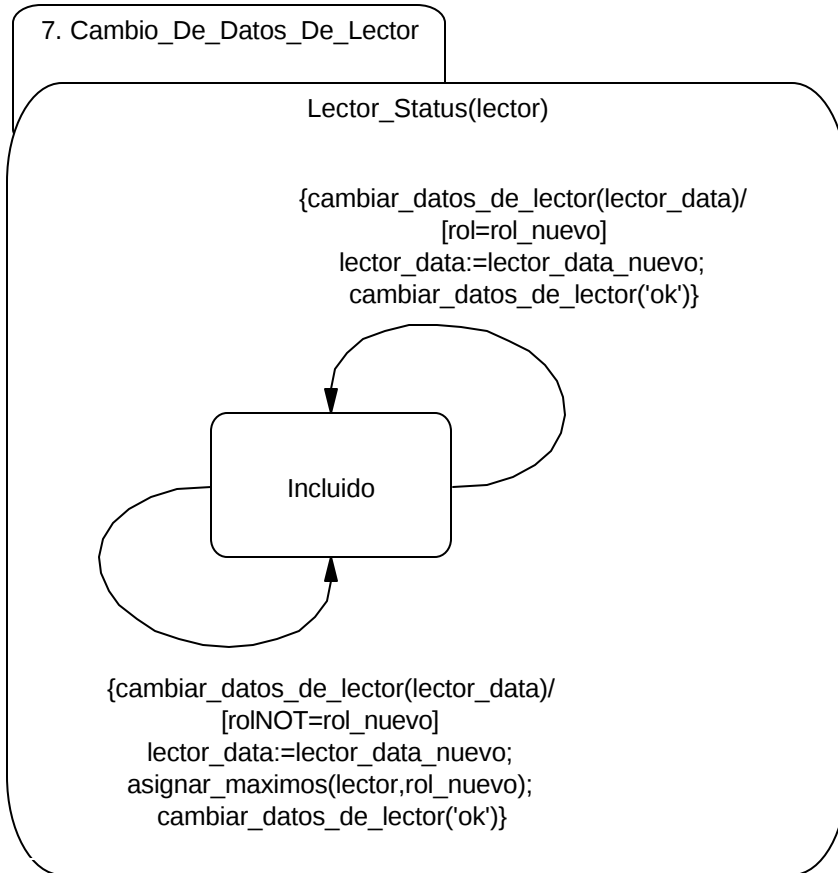
2.5 Diagrama HLS del proceso “Pago_De_Multa”



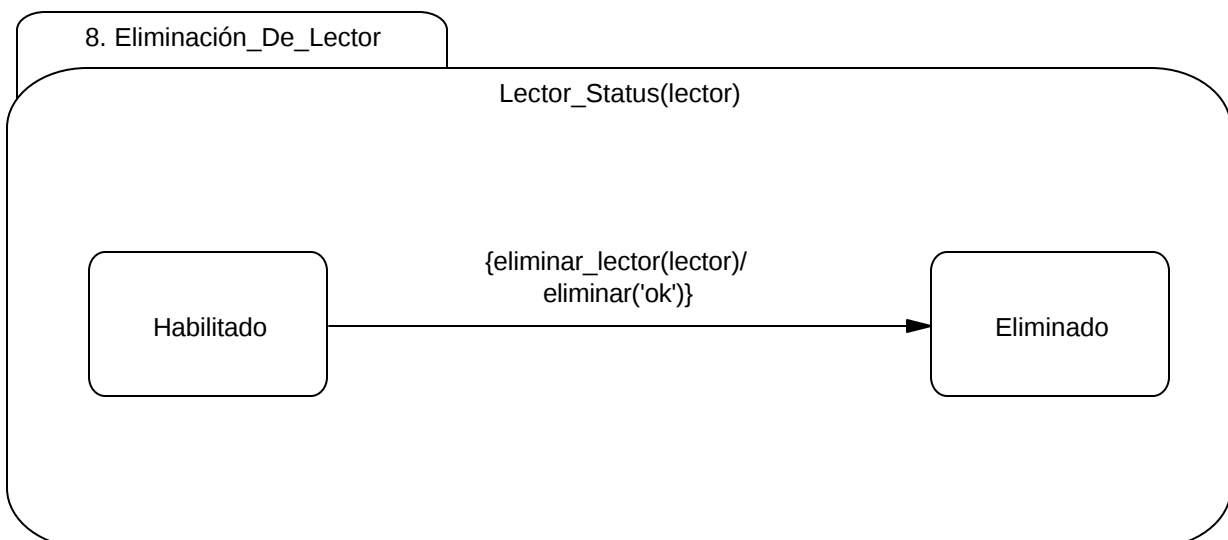
2.6 Diagrama HLS del proceso “Inclusión_De_Lector”



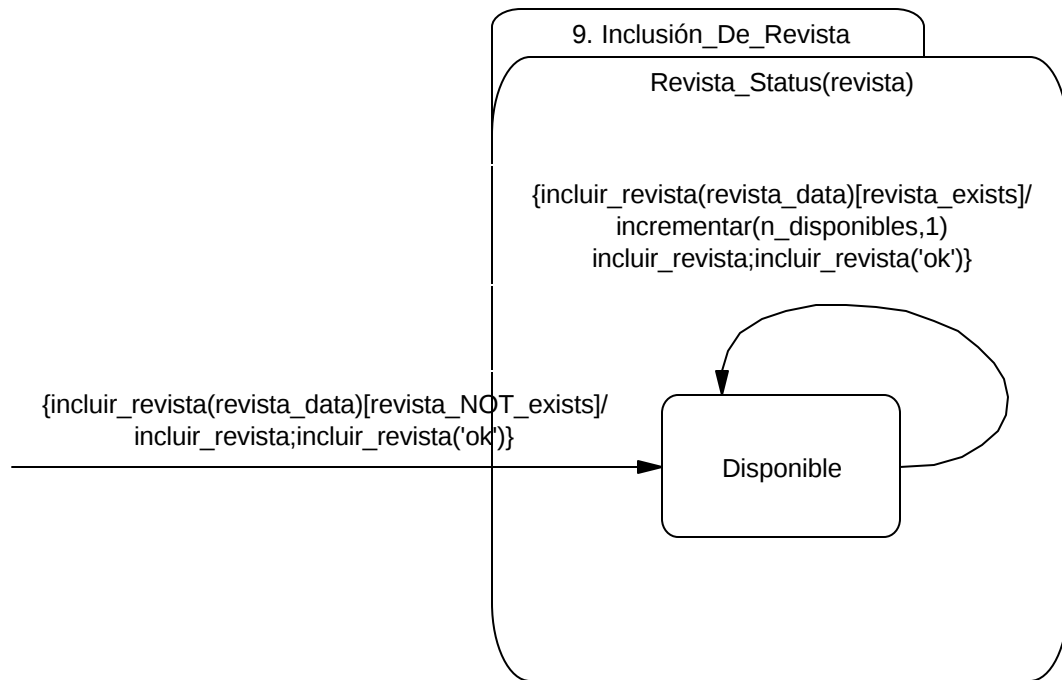
2.7 Diagrama HLS del proceso “Cambio_De_Datos_De_Lector”



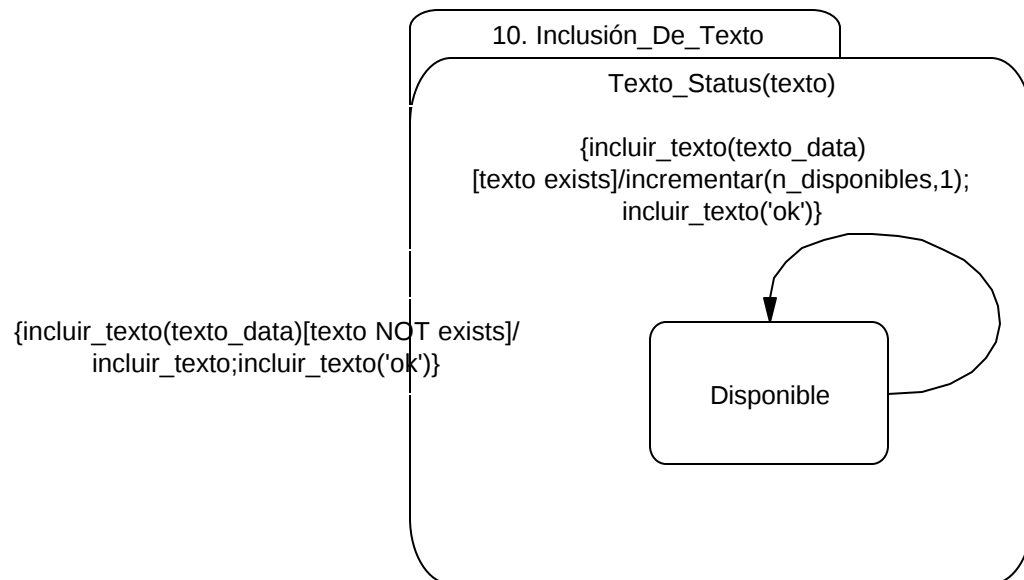
2.8 Diagrama HLS del proceso “Eliminación_De_Lector”



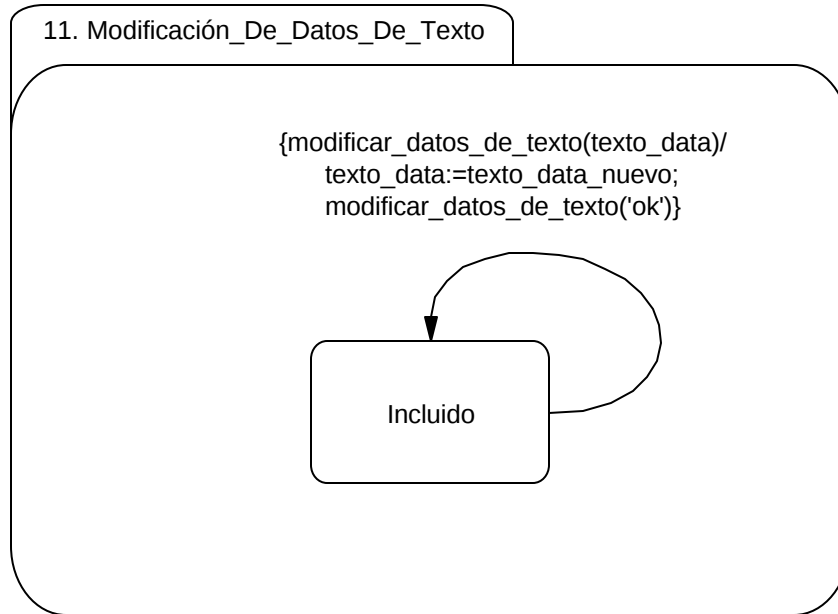
2.9 Diagrama HLS del proceso “Inclusión_De_Revista”



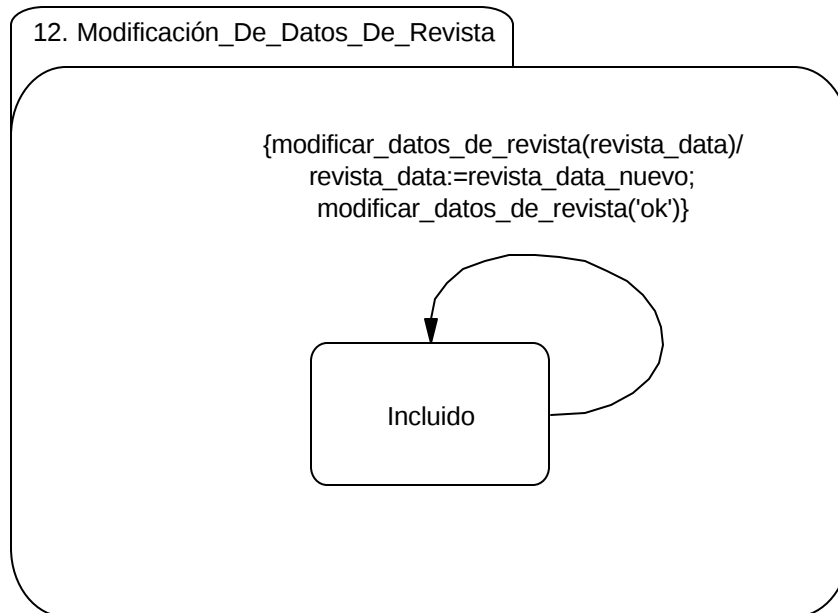
2.10 Diagrama HLS del proceso “Inclusión_De_Texto”



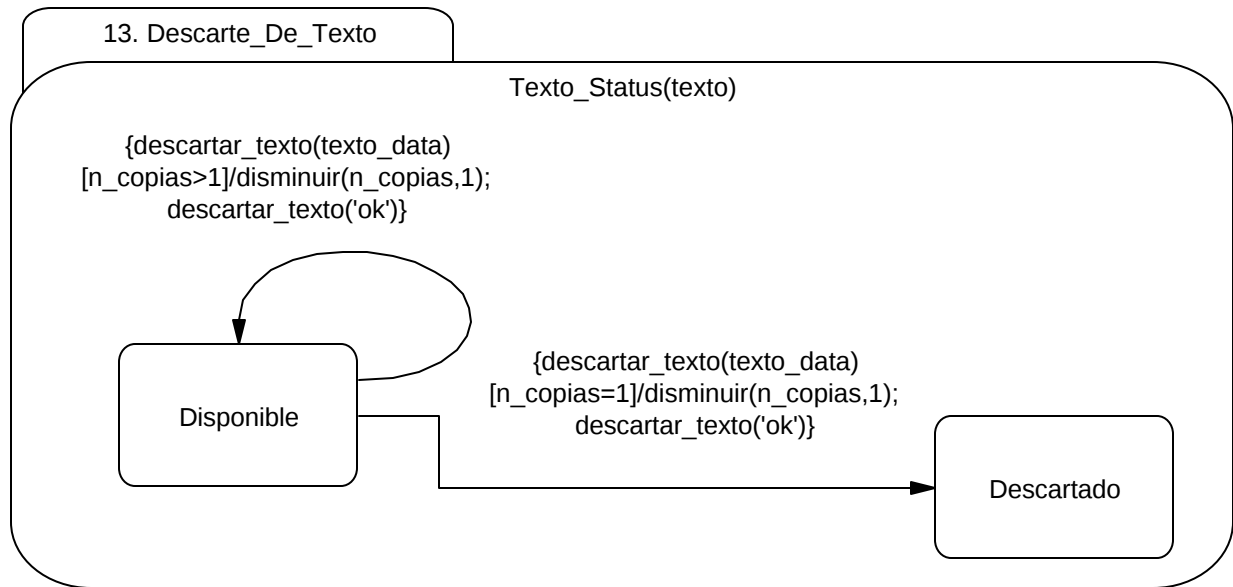
2.11 Diagrama HLS del proceso “Modificación_De_Datos_De_Texto”



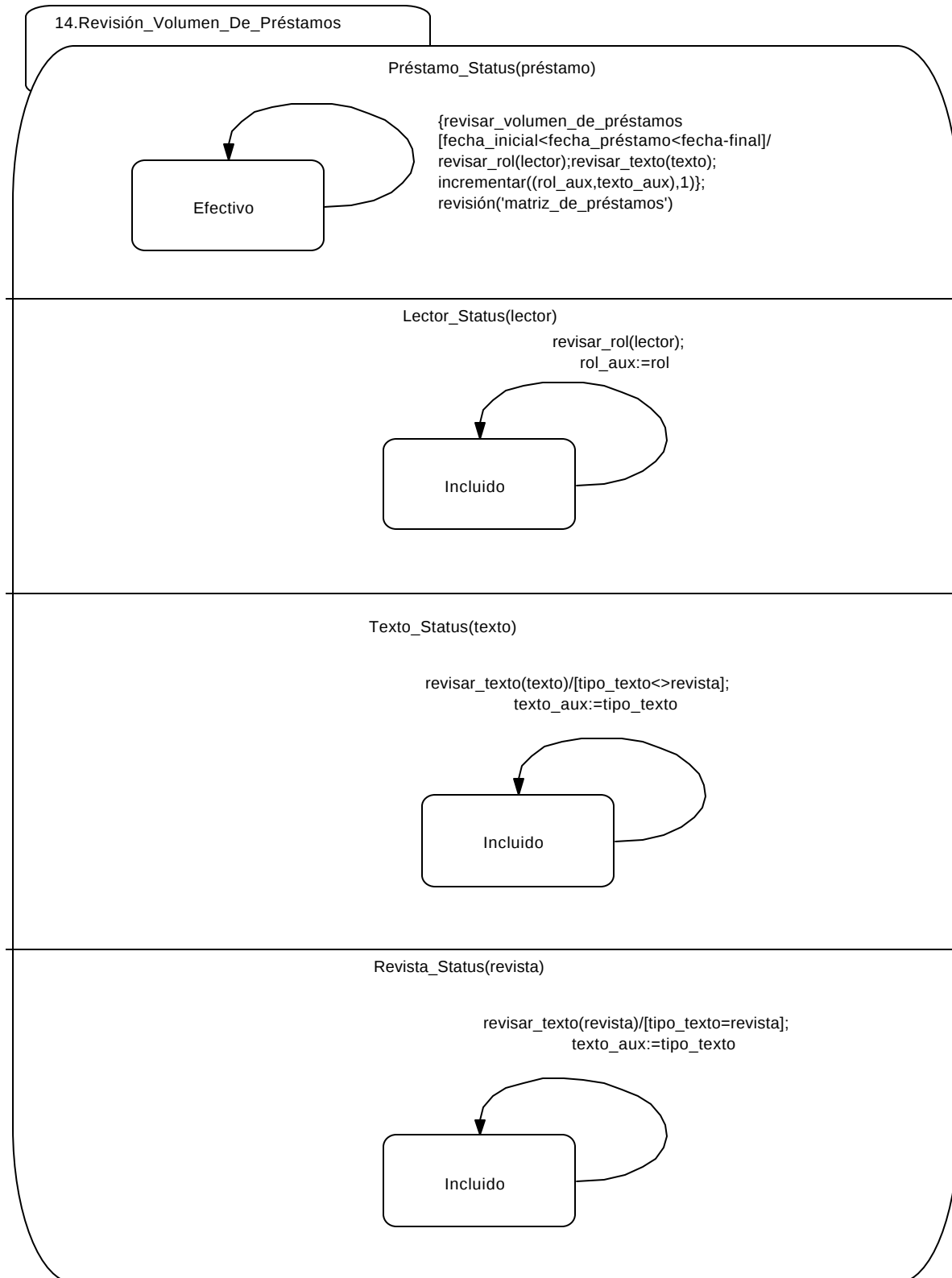
2.12 Diagrama HLS del proceso “Modificación_De_Datos_De_Revista”



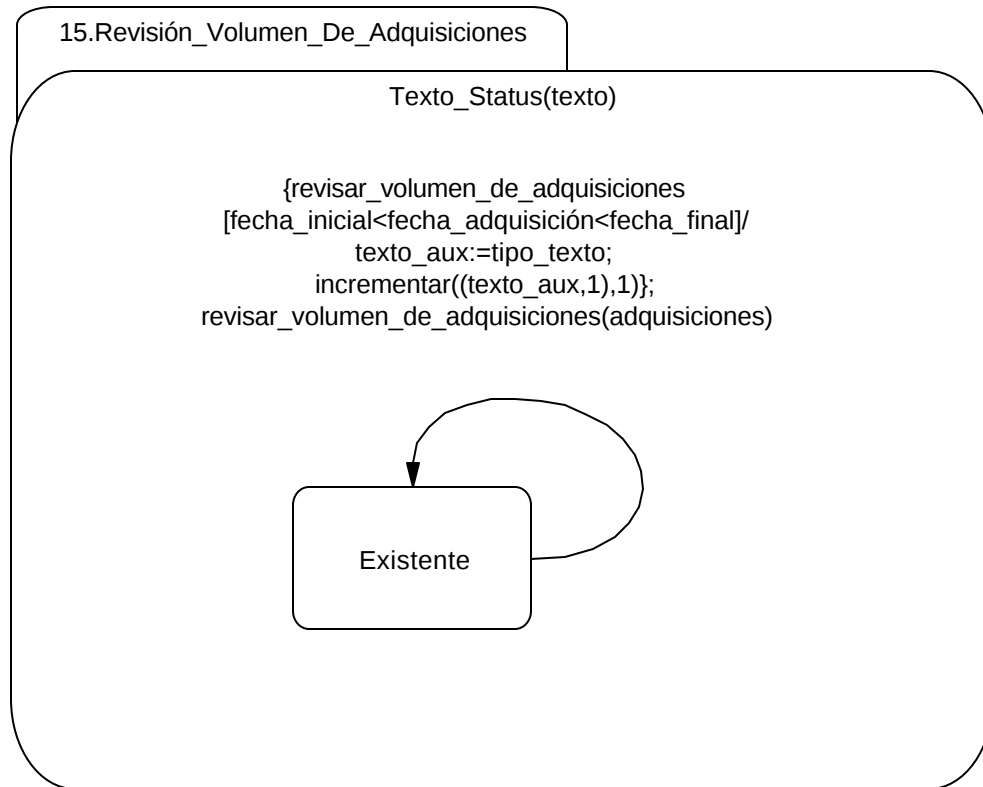
2.13 Diagrama HLS del proceso “Descarte_De_Texto”



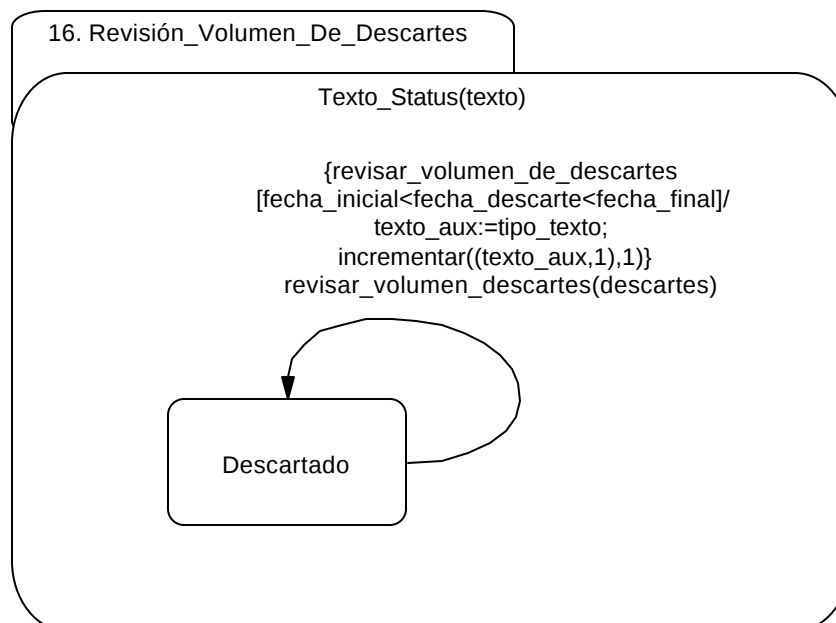
2.14 Diagrama HLS del proceso “Revisión_Volumen_De_Préstamos”



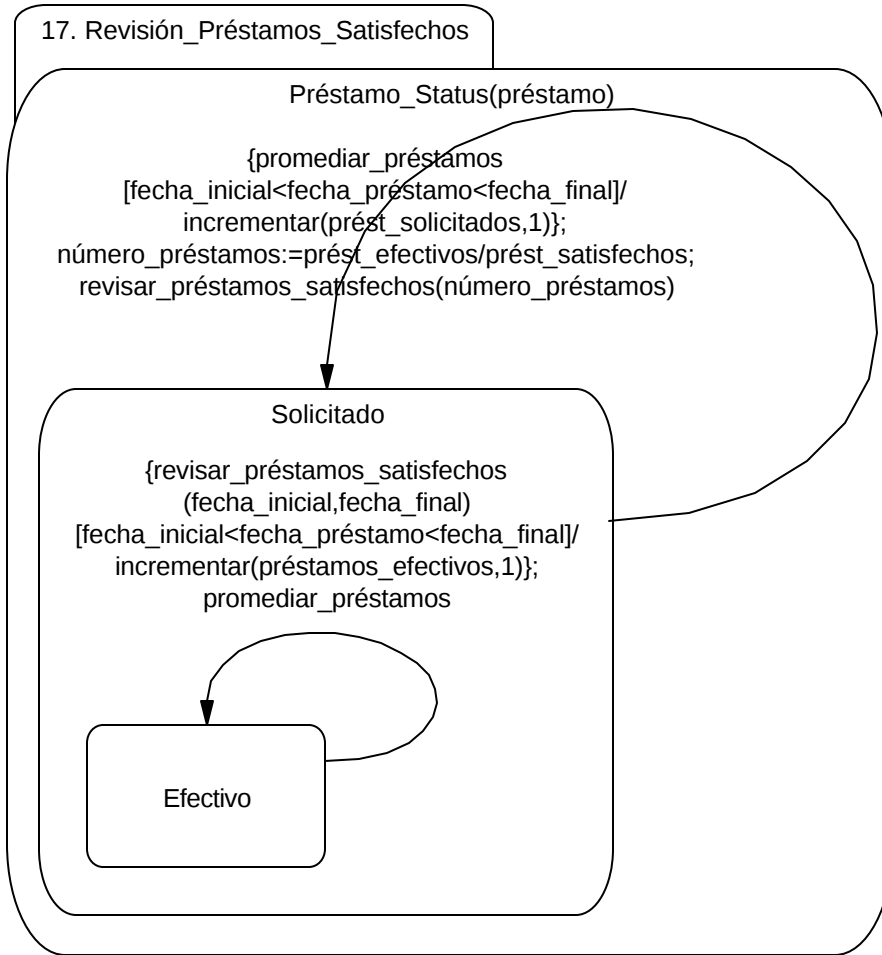
2.15 Diagrama HLS del proceso “Revisión_Volumen_De_Adquisiciones”



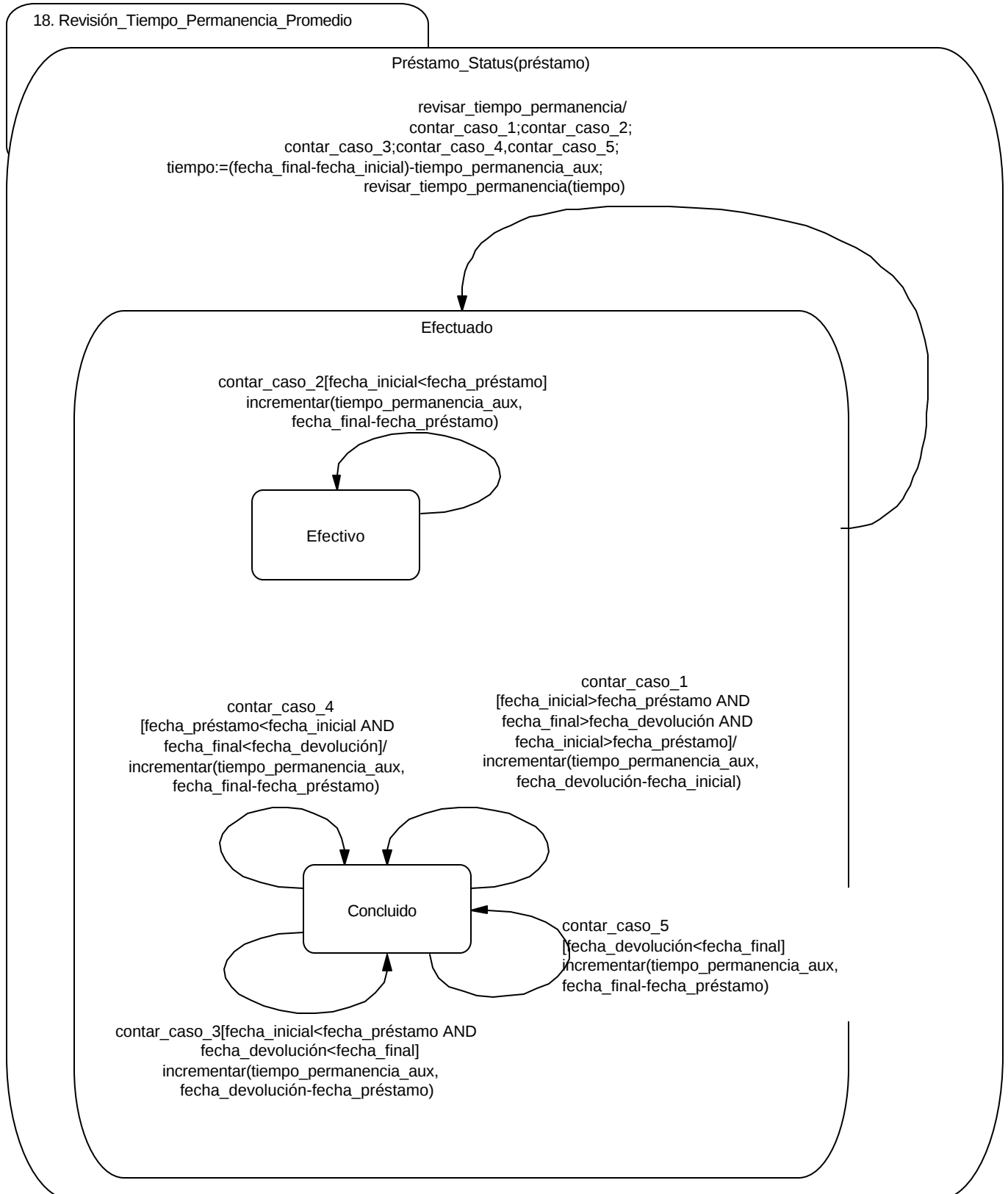
2.16 Diagrama HLS del proceso “Revisión_Volumen_De_Descartes”



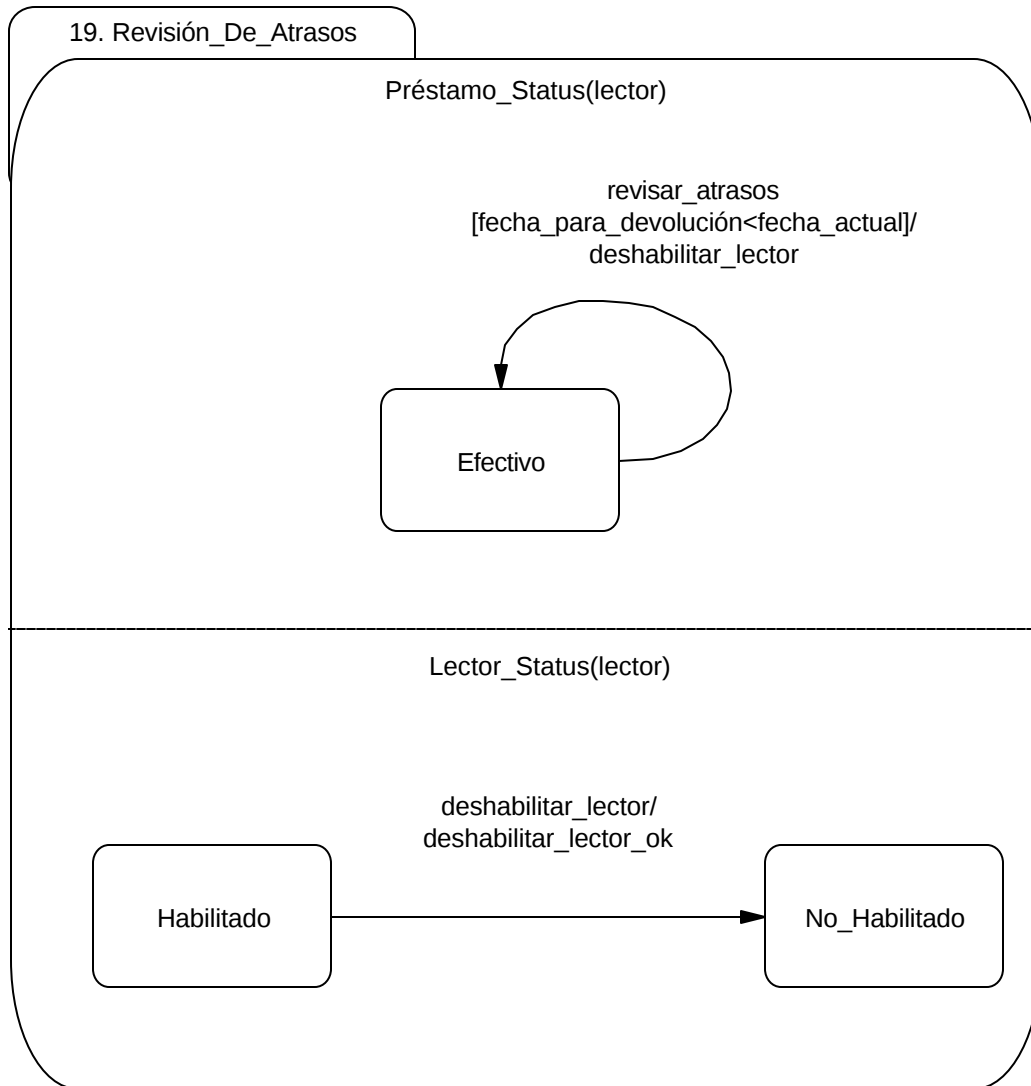
2.17 Diagrama HLS del proceso “Revisión_Préstamos_Satisfechos”



2.18 Diagrama HLS del proceso “Revisión_Tiempo_Permanencia_Promedio”



2.19 Diagrama HLS del proceso “Revisión_De_Atrasos”

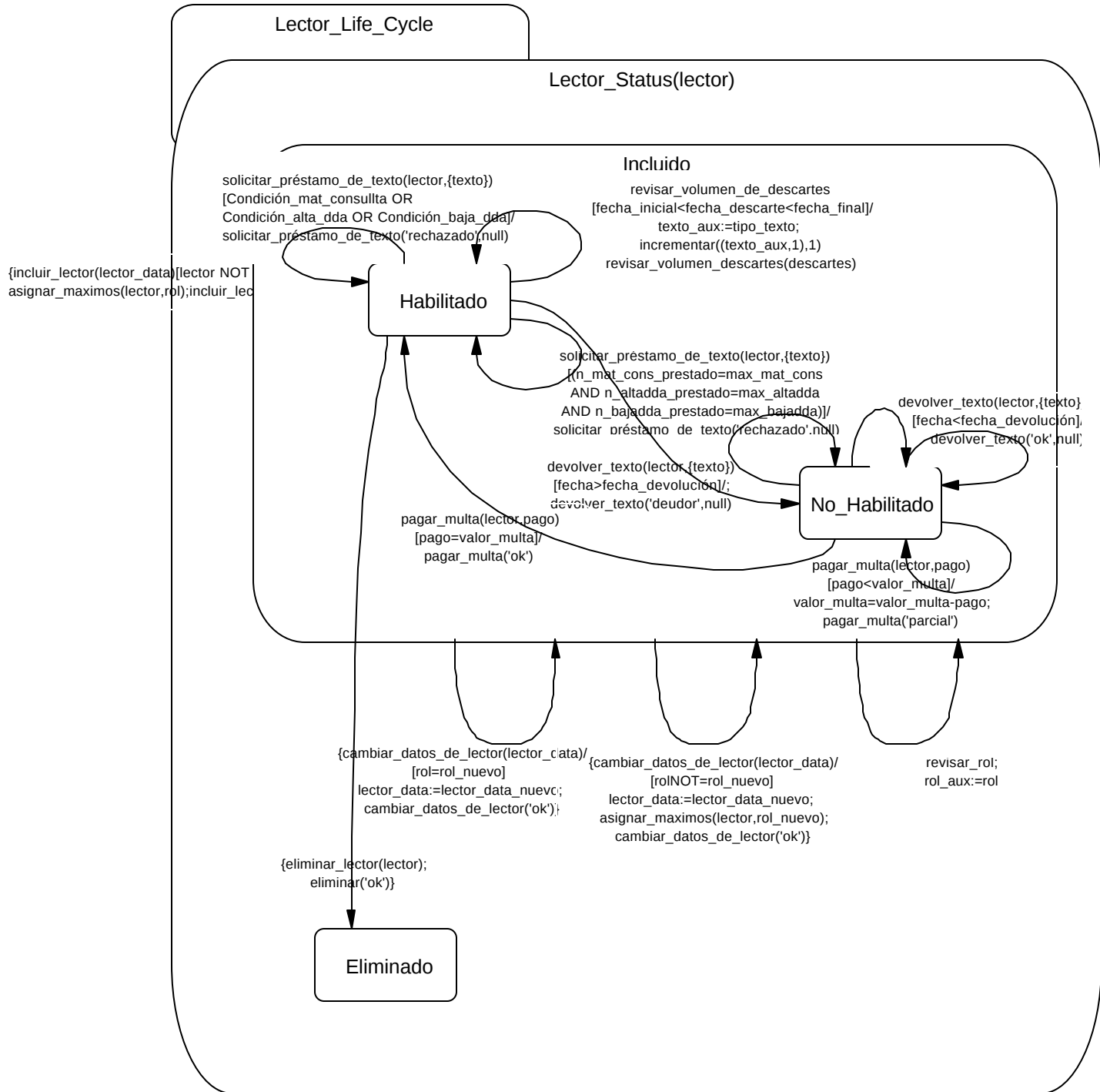


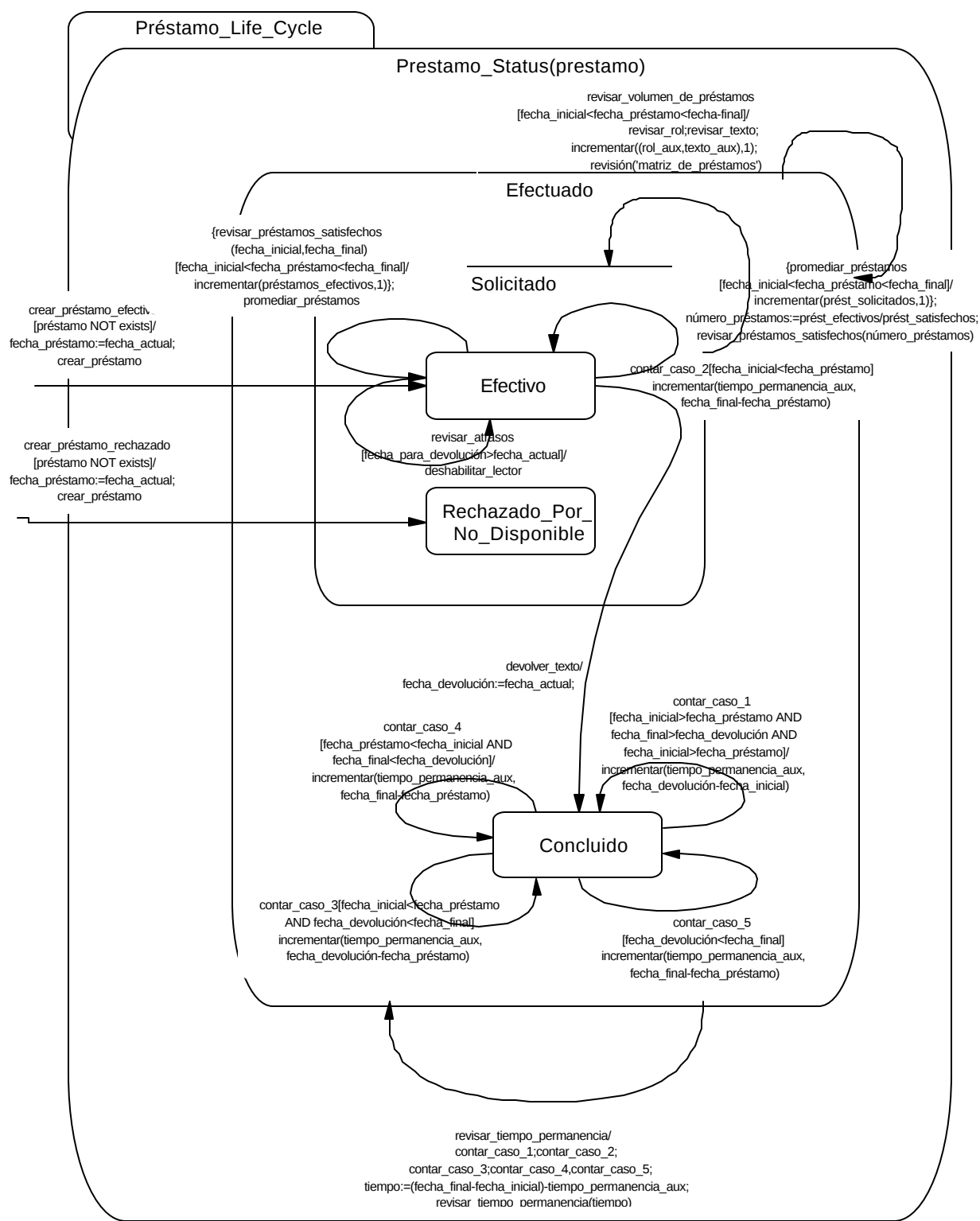
3 Diccionario de datos

lector_data	= @lector+rut+nombre+dirección+condición+rol +maximos+prestados
rol	= [alumno académico funcionario memorista]
condición	= [deudor nodeudor]
maximos	= (max_rev, max_mat_cons, max_alta_dda, max_baja_dda)
prestados	= (n_rev_prestadas, n_matcons_prestados, n_altadda_prestados, n_bajadda_prestados)
texto_data	= @texto+código+autor+título+nºinventario+tema +año+editora+tipo_texto+n_copias+n_disponibles+fecha_adquisición
tipo_texto	= [libro_alta_dda libro_baja_dda material_de_consulta]
revista_data	= @revista+código+título+tema+número+n_disponibles
valor_multa	= [100..10000]
pago	= [100..10000]
condic_préstamo	= [aceptado rechazado]
fecha_devolución	= *fecha en que efectivamente se devuelve el libro*
fecha_para_devolución	= *fecha en la que se debe devolver el libro*
texto_data_nuevo	= *nuevos datos de texto*
lector_data_nuevo	= *nuevos datos de lector*
préstamo	= @préstamo+fecha_préstamo+fecha_devolución+fecha_p_devol
Autoriza	= [si no]

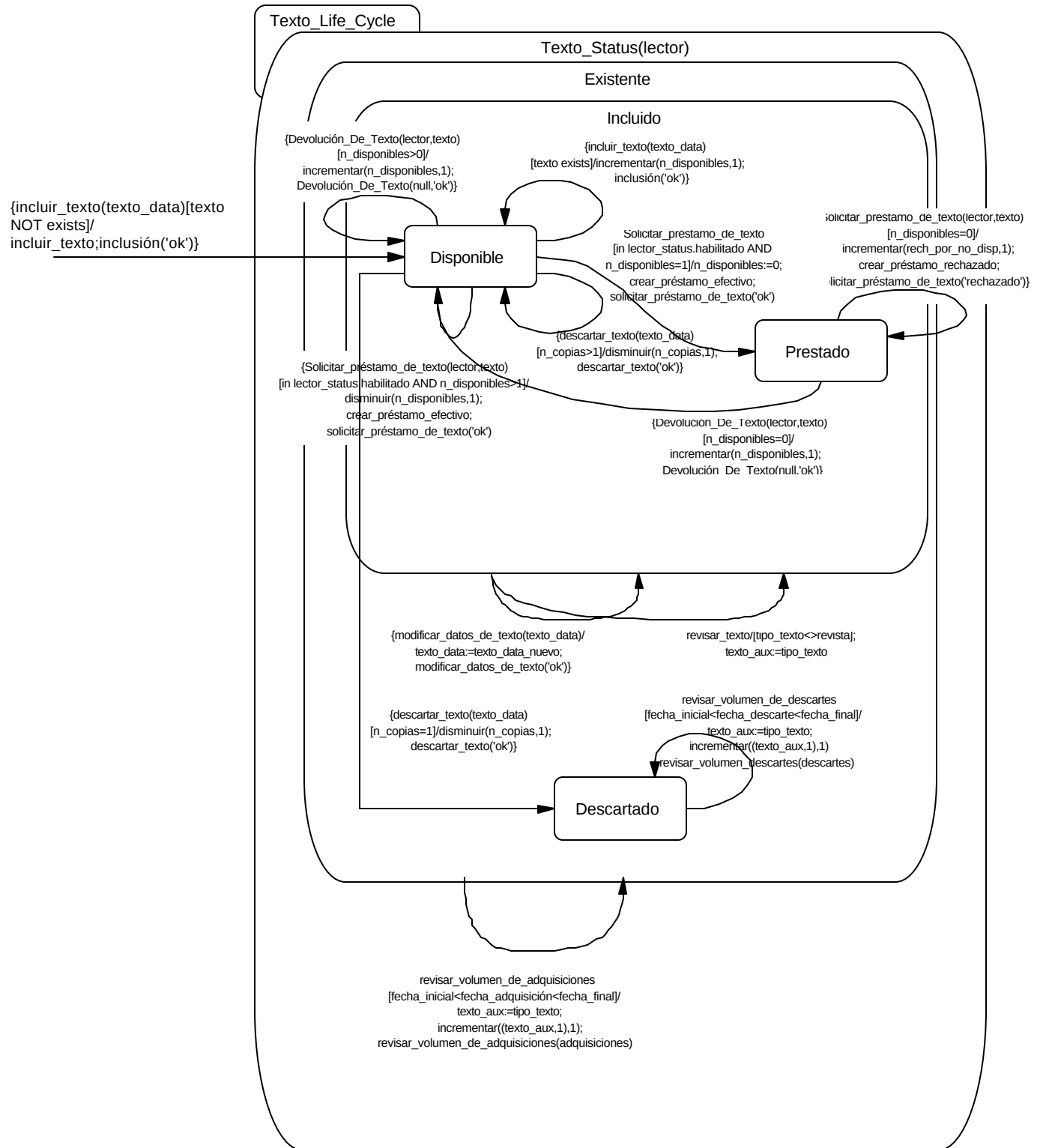
4 Diagramas HLS de Ciclos de Vida

4.1 Diagrama HLS de ciclo vida de la variable lector

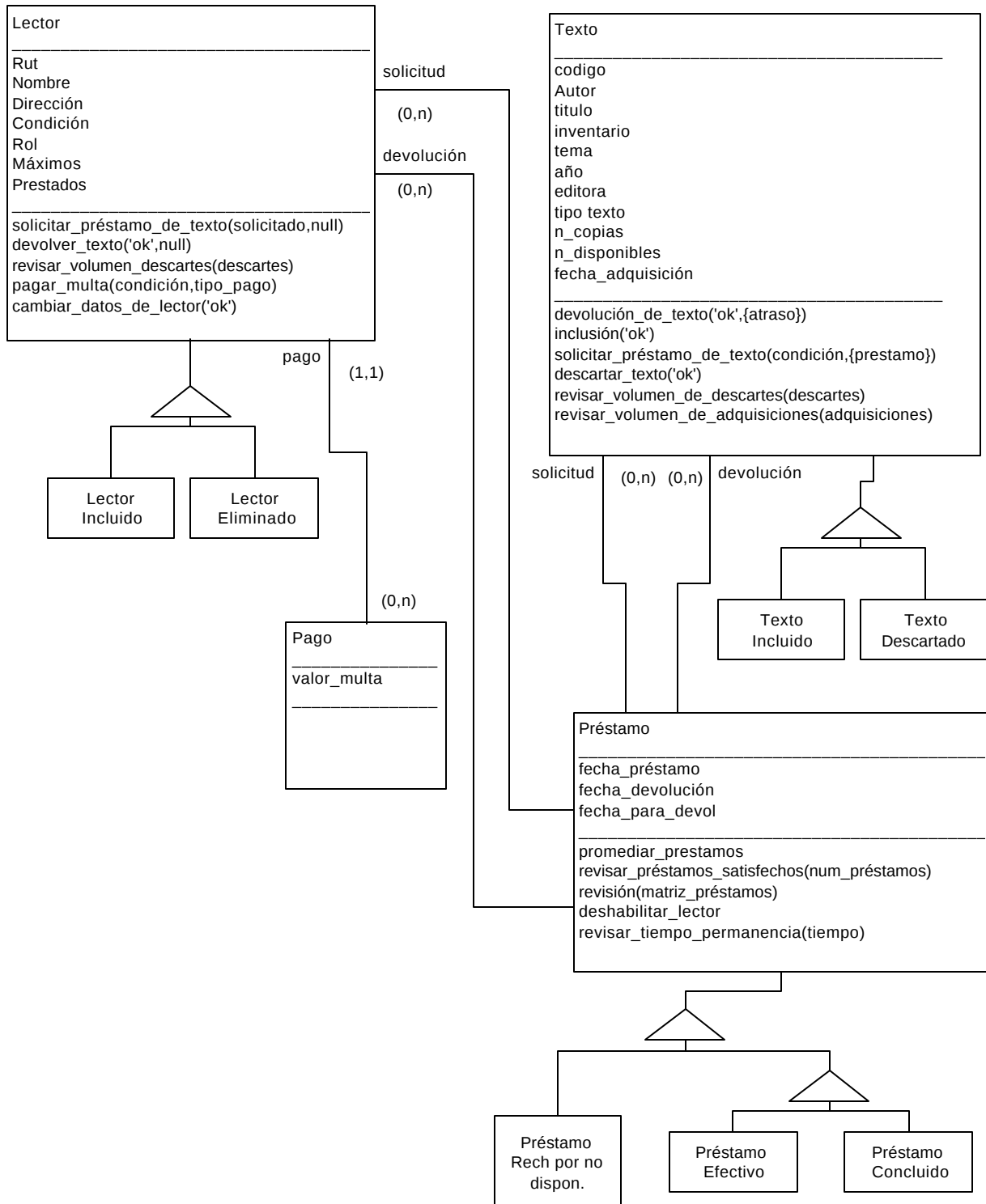




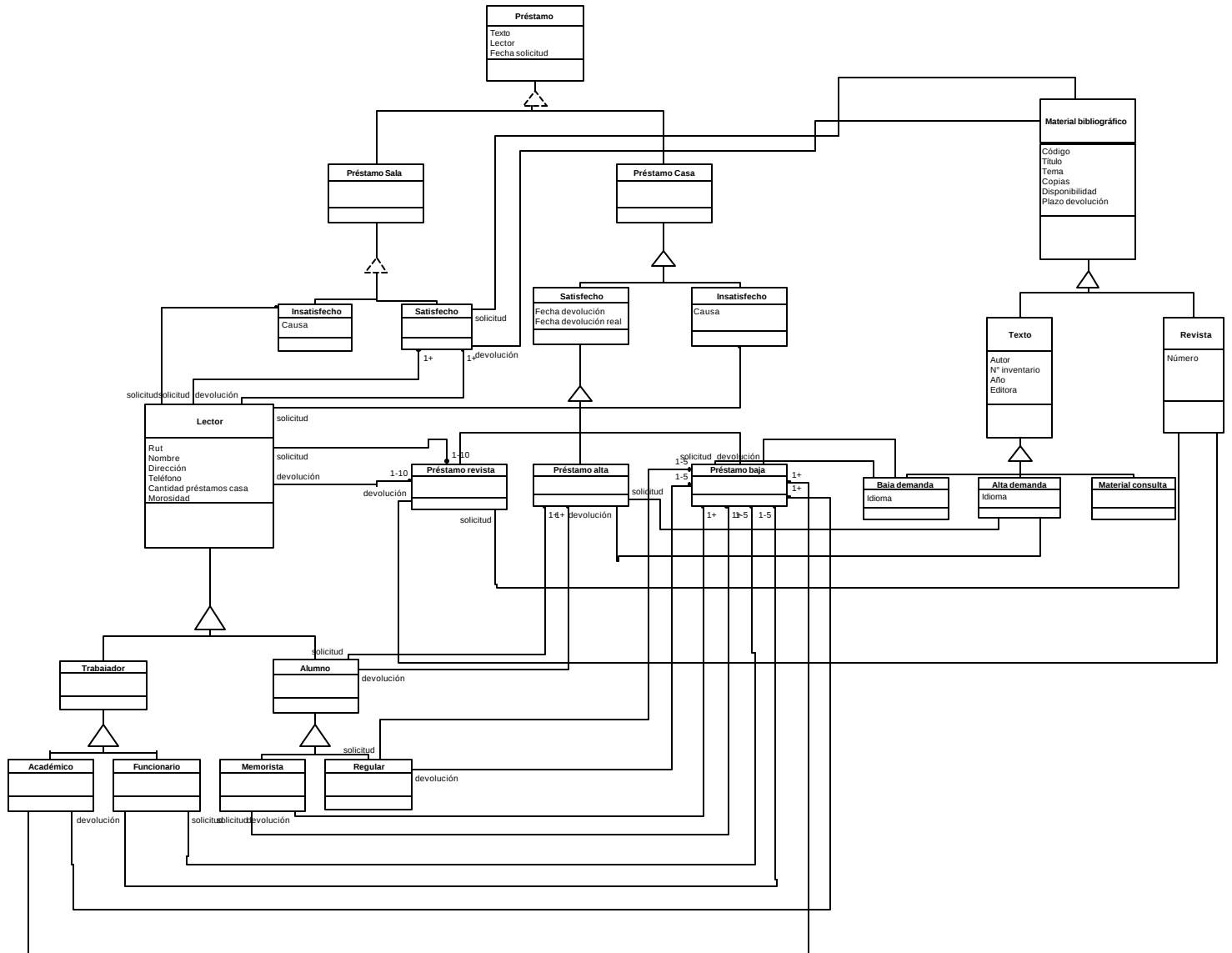
4.3 Diagrama HLS de ciclo vida de la variable texto



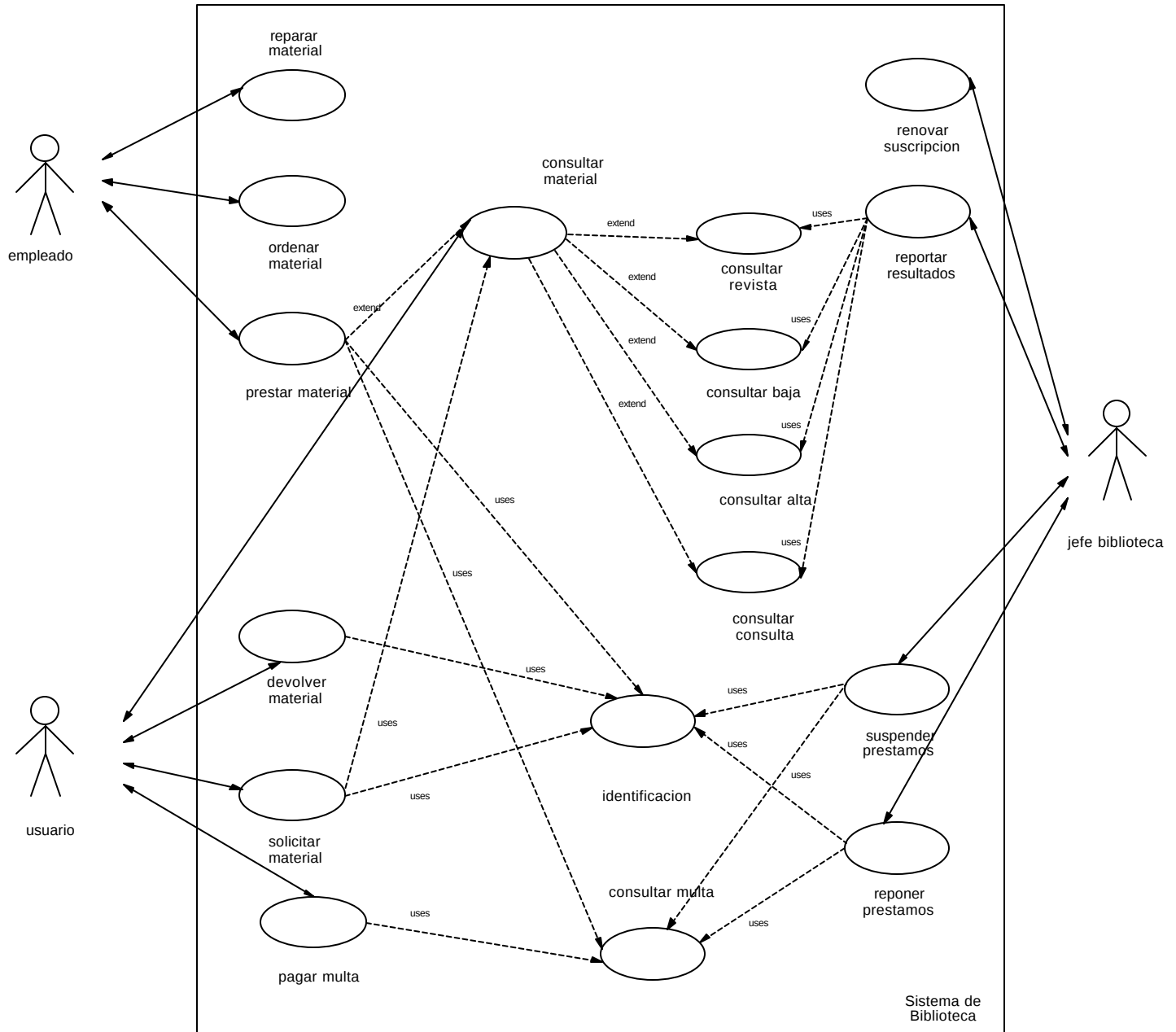
5 Modelo Estructural Orientado a Objetos



6 Modelo estructural de objetos según técnica OMT



7 Modelo use case



1. Descartar material

El empleado consulta el material, revisando el estado que presenta. Si está en mal estado o algún profesor ha dado orden para darlo de baja, aparta el material en mal estado y lo descarta inmediatamente.

2. Cobrar multas

El empleado consulta el estado de multas del usuario que se identifica (mediante el use case Identificación), y si el estado de las multas indica que hay una sin pago, indica al usuario que hay una multa pendiente y no puede solicitar mas materiales hasta que esta sea cancelada.

3. Solicitar material

El usuario se identifica, indicando nombres, tipo de usuario (funcionario, alumno, memorista, profesor) según el use case identificación, y consulta si existe disponibilidad de un material, indicando autor y/o título y tipo de material (libro, consulta, revista) según el use case Consultar material, o bien, materia de interés a través del catálogo de material disponible en biblioteca.

3.1 Consultar material

Consiste en la búsqueda en el sistema por tipo de material bibliográfico, siendo éste de revista, de baja demanda, de alta demanda o de material de consulta, esto según los use cases Consultar revista, Consultar baja, Consultar alta o Consultar consulta.

3.1.1 Consultar revista

El sistema entrega la disponibilidad o no de la revista consultada.

3.1.2 Consultar baja

El sistema entrega la disponibilidad o no del texto de baja demanda consultado.

3.1.3 Consultar alta

El sistema entrega la disponibilidad o no del texto de alta demanda consultado.

3.1.4 Consultar consulta

El sistema entrega la disponibilidad o no del material de consulta consultado.

4. Identificación

El usuario se identifica, indicando nombres, tipo de usuario (funcionario, alumno, memorista, profesor) y el sistema despliega la ficha del lector por pantalla o entrega un mensaje de error en caso de que el solicitante no exista en el sistema.

5. Devolver material

Una vez concluido el periodo de préstamo del material, el usuario restituye el mismo a la biblioteca, ingresando al sistema el identificador del texto y del autor para que se restituya al texto como disponible y quede memorizado en la ficha del lector la devolución del texto.

6. Pagar multa

El usuario consulta (mediante el use case Consultar multa), sabe o es informado por el empleado sobre una deuda pendiente impaga, el valor y motivo de deuda, y procede a hacer la cancelación y recibir el talón de comprobante de pago quedando habilitado para retirar material de la biblioteca.

6.1 Consultar multa

Se ingresa al sistema el identificador del lector y él entrega el valor y motivo de la deuda o la ausencia de la misma.

7. Adquirir material

El Jefe de biblioteca elige un determinado material de un catálogo de ventas del proveedor, lo ingresa al acervo mientras el proveedor emite una factura a ser cancelada a cobranzas.

8. Reponer préstamos

El Jefe de biblioteca es informado que un usuario pagará una multa, solicita su identificación (mediante el use case Identificación), verifica la multa por monto, fecha y motivo (mediante el use case Consultar multa), recibe el dinero del usuario, completa el talón de comprobante de pago y actualiza el estado del usuario, habilitándolo para solicitar material nuevamente.

9. Suspender préstamos

El Jefe consulta los usuarios deudores (mediante el use case Identificación) y mediante el use case Consulta multa ve cuales son sus deudas impagas y, acto seguido, suspende a cada uno de ellos la capacidad para solicitar material hasta que regularicen la situación.

10. Renovar suscripción

El Jefe comprueba que ha llegado la fecha de vencimiento de la suscripción de un material y la renueva por el siguiente periodo.

11. Reportar resultados

EL Jefe consulta la existencia total de material disponible en la biblioteca en todas las categorías de material (mediante los use cases Consultar revista, Consultar baja, Consultar alta y Consultar consulta), obtiene índices respecto de nuevos volúmenes adquiridos; además, materiales deteriorados, más solicitados, menos solicitados, número de volúmenes por cada uno de ellos y confecciona un informe de desempeño del periodo de servicio.

12. Prestar material

El empleado requiere que el usuario solicitante se identifique (mediante el use case Identificación); acto seguido, confirma que el identificado no tenga multas impagas pendientes (mediante el use case Consultar multa). Aclarado el estado de las multas, consulta el material

(mediante el use case Consultar material), para determinar si hay disponibilidad de lo solicitado por el usuario, de acuerdo al tipo de usuario, anota la información del volumen a prestar (título, autor, edición, código de biblioteca, fecha de préstamo, fecha devolución), lo entrega al solicitante y le comunica la fecha de devolución.

13. Ordenar material

El empleado consulta el material, revisando el título, autor, edición e idioma de cada volumen y los ordena alfabéticamente agrupándolos por misma letra de inicio.

14. Agregar usuario

El empleado solicita al nuevo usuario que se identifique, y anota el nombre, status (memorista, profesor, alumno, funcionario), fecha de apertura de la ficha en el registro de usuarios.

15. Eliminar usuario

El empleado constata cuales usuarios no han hecho uso del sistema en cuatro semestres, los registra en " usuarios en pasivo " y los elimina de los usuarios activos.

ANEXO 5 Definición del Problema del Juzgado de Letras

1 Antecedentes

La Constitución Política del Estado de Chile establece que la función del Poder Judicial es la de “...conocer las causas civiles y criminales, de resolverlas y de hacer ejecutar lo juzgado...” (Artículo 73 de la Constitución Política del Estado de Chile). Esta función es delegada en forma exclusiva en el Poder Judicial, prohibiéndose la intervención de los demás poderes del estado en ella.

La constitución establece también la obligación del Poder Judicial de resolver todos los asuntos que se le presenten (principio de *inexclusividad*), y confiere al Poder Judicial *potestad de imperio* sobre las autoridades y la fuerza para hacer cumplir sus resoluciones.

El sistema judicial chileno se organiza en tres niveles jerárquicos. En el nivel superior se encuentra la Corte Suprema, bajo cuya supervisión se estructura todo el sistema judicial. Asume las funciones de hacer efectiva la supremacía constitucional, proteger las garantías constitucionales, uniformar la jurisprudencia y, en general, servir de tribunal de segunda instancia en las causas en que la Corte de Apelaciones funcionan como tribunales de primera. Le corresponde la superintendencia directiva, correccional y económica sobre todos los Tribunales de la República.

Las Cortes de Apelaciones son tribunales dependientes directamente de la Corte Suprema. En segunda instancia conocen las causas civiles y criminales, así como las no contenciosas que hayan resuelto en primera instancia los jueces de letras de la jurisdicción. Le corresponden también funciones administrativas, económicas y disciplinarias sobre los tribunales de su territorio jurisdiccional, que coincide con el de una región o parte de ella.

Bajo la supervigilancia de las Cortes de Apelaciones se encuentran los Juzgados de Letras, que son tribunales ordinarios de primera instancia. Están constituidos por un juez de letrado, un secretario y un número variable de funcionarios. Se organizan en función del territorio jurisdiccional, esto es, ejercen sus atribuciones en una posición del territorio de la Nación, que, por lo general, coincide con una comuna o agrupación de comunas. Tienen habitualmente jurisdicción común (conocen todo tipo de asuntos), pero en algunos centros urbanos se han creado juzgados especializados en materias penales, civiles, de menores y del trabajo.

A estos Tribunales les corresponden diversas funciones, distinguiéndose dos grandes clases de actividades: jurisdiccionales y administrativas.

La actividad jurisdiccional dice relación con el ejercicio de la jurisdicción que tienen los tribunales. Es decir, representa el poder de administrar justicia que poseen los tribunales judiciales. Específicamente, consiste en las actuaciones procesales y decisiones de orden jurídico adoptadas por el personal del tribunal durante la tramitación procesal de causas judiciales.

Por actuaciones procesales se entienden las diligencias realizadas en cumplimiento de los procedimientos establecidos por ley para regular la tramitación de los procesos judiciales. Algunos ejemplos de diligencias son los siguientes:

- realizar comparendos,
- realizar pruebas testimoniales,
- efectuar testificaciones,
- dictar providencia correspondiente a escritos ingresados al tribunal,
- acoger o rechazar lo pedido de las partes, y
- dictar sentencias.

Por su parte la actividad administrativa dice relación con la manutención y funcionamiento del tribunal. Consiste en la asignación, manutención y control de los recursos materiales, financieros y humanos de que dispone un tribunal para funcionar y cumplir sus objetivos.

2 Sistema de información a ser modelado

El sistema de información a ser modelado consiste en un conjunto de funciones que permiten operar más eficientemente un juzgado de letras desde el punto de vista del seguimiento de las causas llevadas en él. Además, debe permitir obtener una buena visión global de la actividad realizada en el mismo, la cual puede ser detallada en el grado en que se desee.

El modelo de sistema de información debe tener amplia aplicabilidad, logrando una independencia del tipo de juzgado en que se implante, y por ende, del tipo de causas judiciales que son procesadas en él.

Las causas son tipificadas en criminales, laborales, alcoholes, civil voluntaria, civil contenciosa, menores civil y menores crimen. Dependiendo del tipo de juzgado, es el(los) tipo(s) de causa(s) que se lleva(n). Sin embargo, en el caso de juzgados mixtos (que son los más numerosos), éstos pueden recibir causas de todo tipo.

Las causas en general pueden contemplar varias materias (o delitos), pero por definición deben poseer una materia principal que tipifica la causa, por ejemplo: una causa puede contemplar un homicidio y una lesión menor, siendo la primera materia la que tipifica la causa como criminal y como homicidio. Las materias o delitos están relativamente estandarizados y pueden ser codificados para facilitar su clasificación y posterior toma de estadísticas. No es poco frecuente que una causa cambie de materia o que la materia principal pase a ser secundaria y una de las secundarias pase a ser principal.

Cuando se recibe una causa, ésta es ingresada, tipificada y mantenida posteriormente por un funcionario denominado actuario, el cual le asigna un rol identificador, anotando el inicio de su tramitación y todas las diligencias que se llevan a cabo dentro de ésta y eventuales futuras tramitaciones. Cada diligencia que registra el actuario debe considerar la fecha en que

fue solicitada, la fecha en que fue efectivamente realizada, el responsable por la misma y el detalle correspondiente. La forma en que se llevan las diligencias en una causa es la misma que en un exhorto con la diferencia que no existe un actuario designado a cargo del exhorto.

Las diligencias están también tipificadas, por tanto, y dependiendo del tipo de causa, cada nueva diligencia debe enmarcarse dentro de los tipos que correspondan.

Cabe hacer notar que una diligencia solicitada en un juzgado para ser realizada fuera de su jurisdicción se transforma en un exhorto para el juzgado receptor. Por tanto, cada juzgado solicita también diligencias que serán exhortos para otros juzgados.

Cada tramitación de una causa debe tener un término o sentencia de responsabilidad del juez. Cada tipo de causa tiene motivos de término propios que el juez puede invocar. Por ejemplo, una causa puede ser archivada por falta de antecedentes, esto provoca el término de una tramitación, sin embargo, tiempo después puede ser aportado un nuevo antecedente que obliga a iniciar una nueva tramitación. Cada tramitación es siempre iniciada por un denunciante (por ejemplo Carabineros o Investigaciones).

La sentencia es redactada por el juez considerando todos los antecedentes y diligencias de la causa. Es importante entonces contar con una buena disponibilidad para consultar todos los datos que el juez requiera sobre la causa, así por ejemplo, debe ser posible acceder a:

- tramitaciones,
- diligencias,
- materias que trata, y
- partes que involucra (con su RUT, nombre completo, dirección, profesión, estado civil y nacionalidad).

Esta misma facilidad de consulta de datos que dispone el juez puede entregarse a quien lo solicite con expresas excepción de aquellas causas que por su naturaleza tengan orden de no informar. Los actuarios son los únicos responsables por cualquier adición, modificación o eliminación de registros de una causa.

Los tipos de partes involucrados en una causa dependen del tipo de esta última.

En el juzgado las causas son indistintamente identificadas por el rol o por el apellido de ambas partes. Por ejemplo, una causa puede ser identificada como la “438” o como la “Perez y García”. Dependiendo de la modalidad que adopte el juzgado, el expediente que contiene toda la documentación de la causa puede ordenarse por cualquiera de estos dos criterios. En el caso de juzgados mixtos, las causas deben identificarse primero por el tipo.

Los actuarios deben también elaborar dos informes periódicos estandarizados denominados “Informe Bimensual” e “Informe de Causas Terminadas”. El primero muestra la última diligencia realizada para cada causa además de la diligencia consignada como última en el Informe Bimensual anterior. De esta forma, es posible determinar si existe avance en la trami-

tación procesal de las causas. El Informe de Causas Terminadas es emitido mensualmente y muestra el motivo de término de cada una de las causas terminadas en el mes considerado. Ambos informes son de responsabilidad del juez y por tanto él debe preocuparse por su veracidad y correctitud.

Los ministros de la Corte de Apelaciones a la cual está subordinado el juzgado, solicitan esporádicamente informes estadísticos que muestran los ingresos y términos de causas para un periodo de tiempo cualquiera. Estos informes pueden ser solicitados por:

- tipo de causa,
- materia,
- motivo de término, y
- combinación de las anteriores.

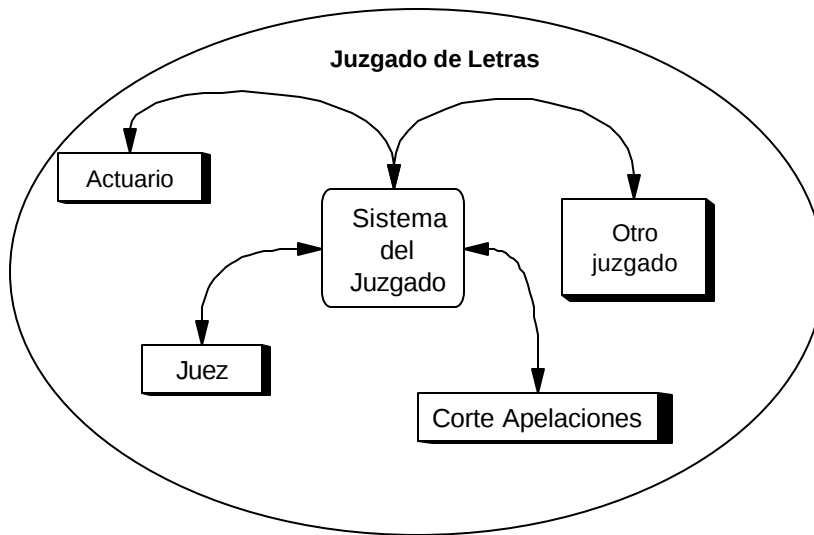
Con esta información la Corte de Apelaciones puede formarse una idea de la actividad de los juzgados de su jurisdicción. El propio juez también solicita informes estadísticos como los elaborados para la Corte de Apelaciones pero con un mayor detalle, incluyendo también los actuarios y las diligencias.

Para una mejor gestión de su juzgado, el juez debe contar también con estadísticas que muestren la duración promedio en días de las causas llevadas (por ejemplo, bajo la forma de histograma de frecuencias) y los exhortos, que representan una carga adicional recibida de otros juzgados.

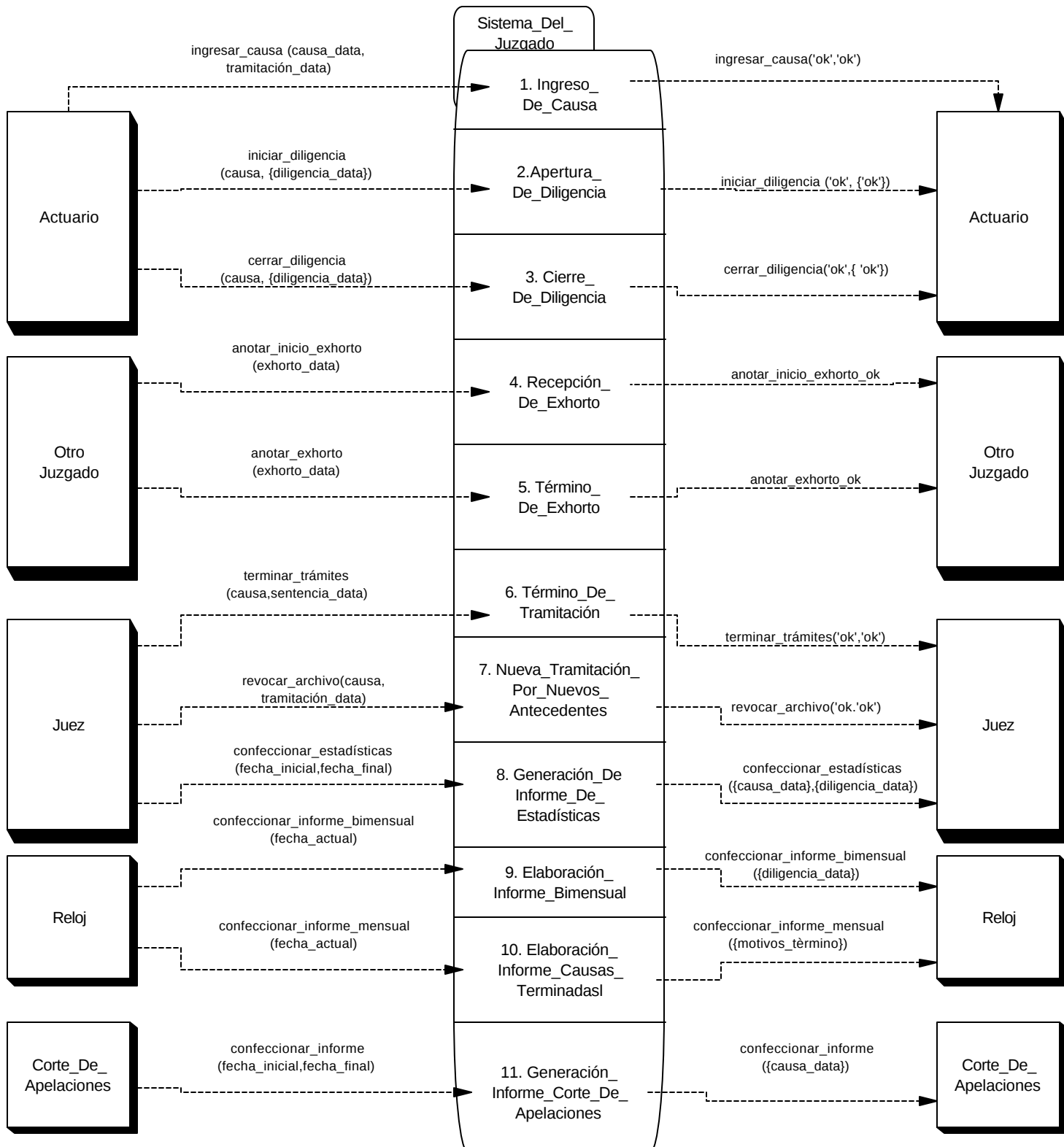
ANEXO 6 Modelo Completo del Problema del Juzgado de Letras

1 Modelo de Visión Global

1.1 Diagrama de contexto

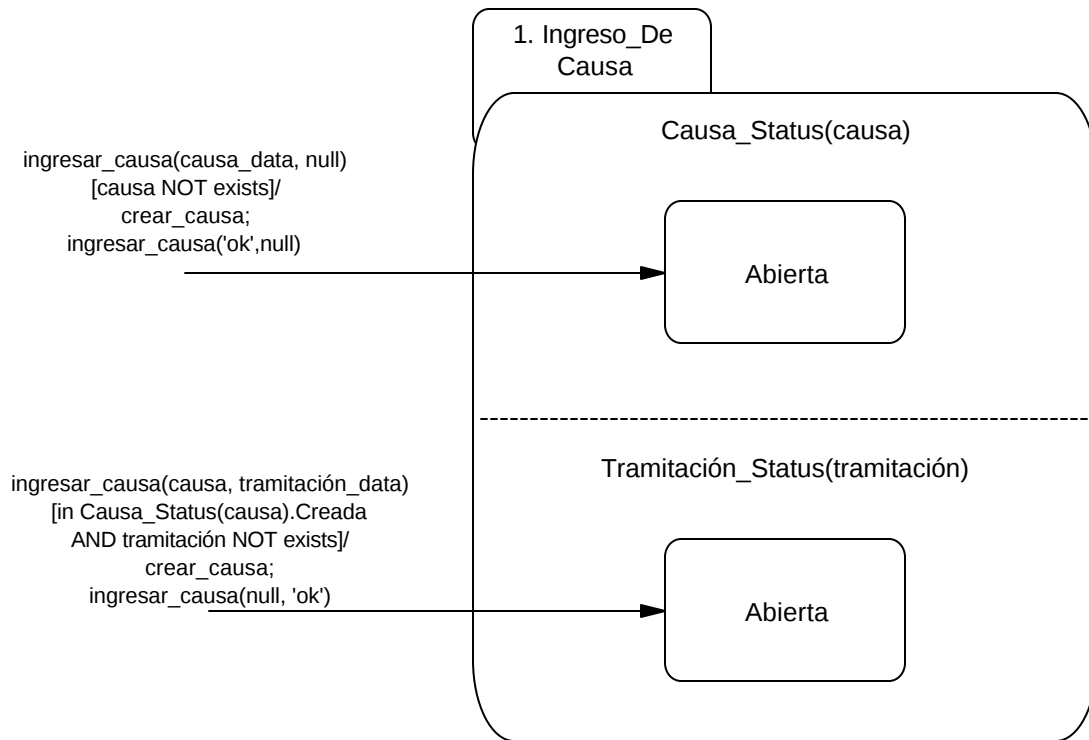


1.2 Diagrama de visión global

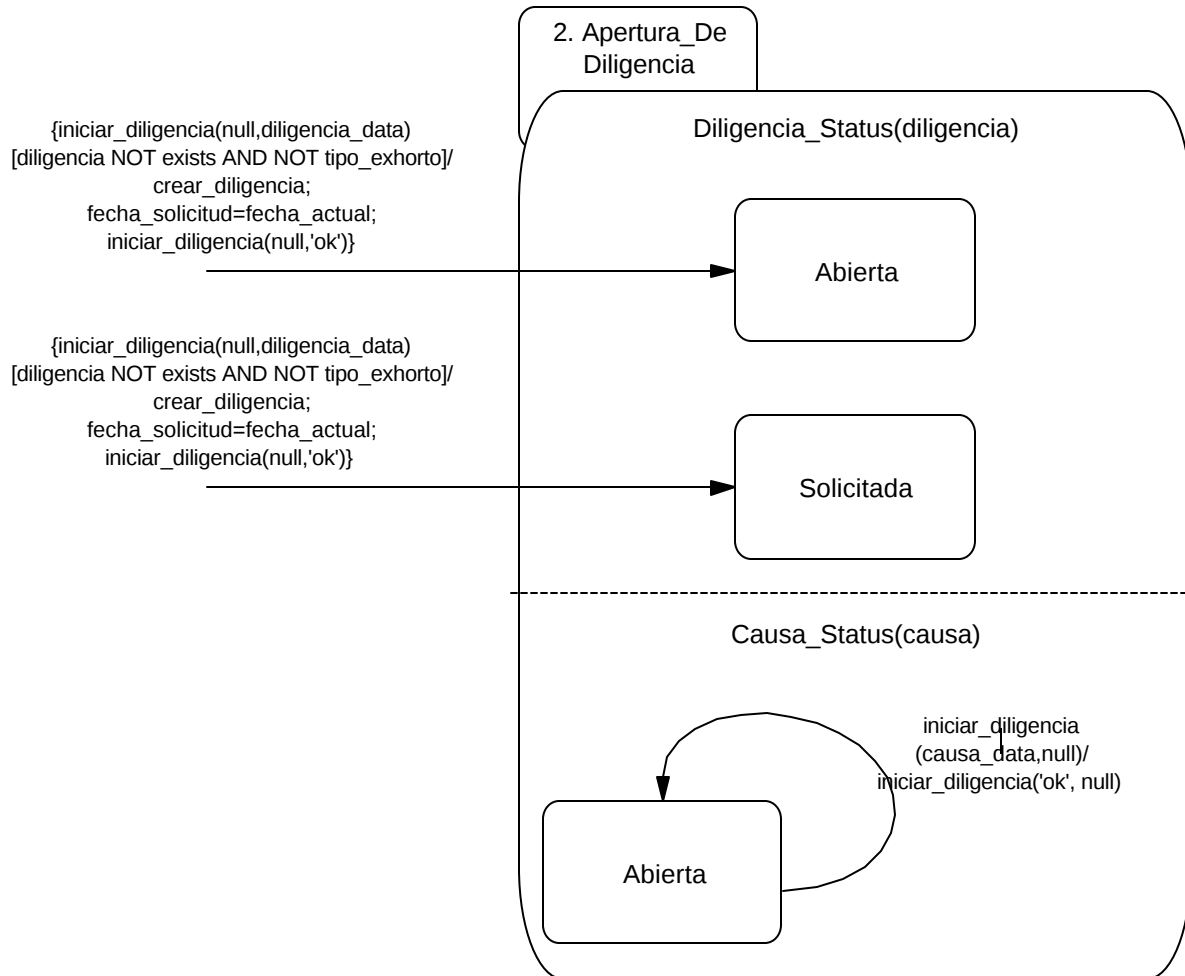


2 Modelo de Procesos

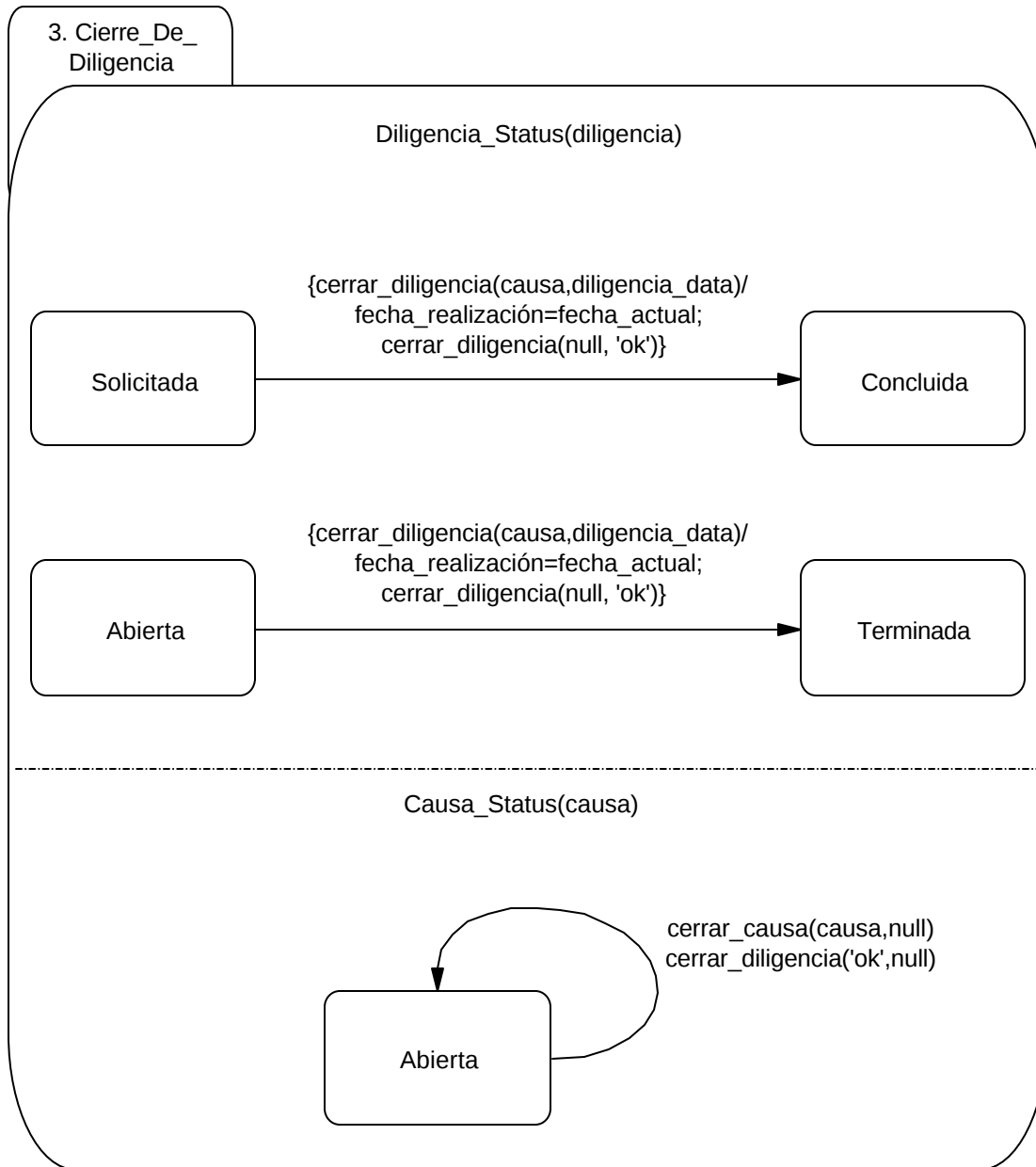
2.1 Diagrama HLS del proceso “Ingreso_De_Causa”



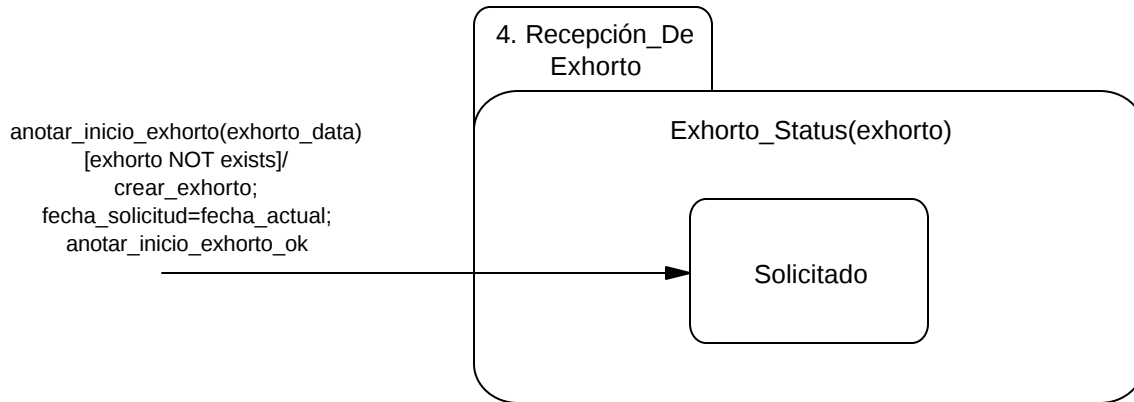
2.2 Diagrama HLS del proceso “Apertura_De_Diligencia”



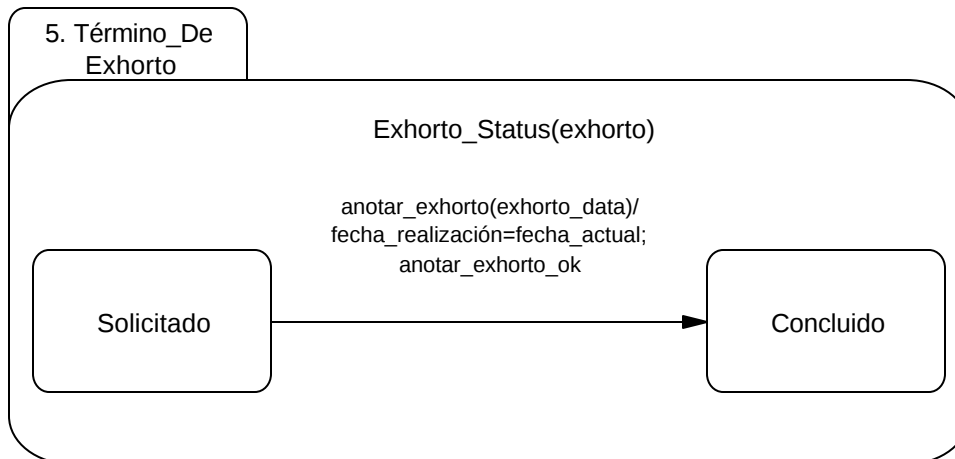
2.3 Diagrama HLS del proceso “Anotación_De_Diligencia”



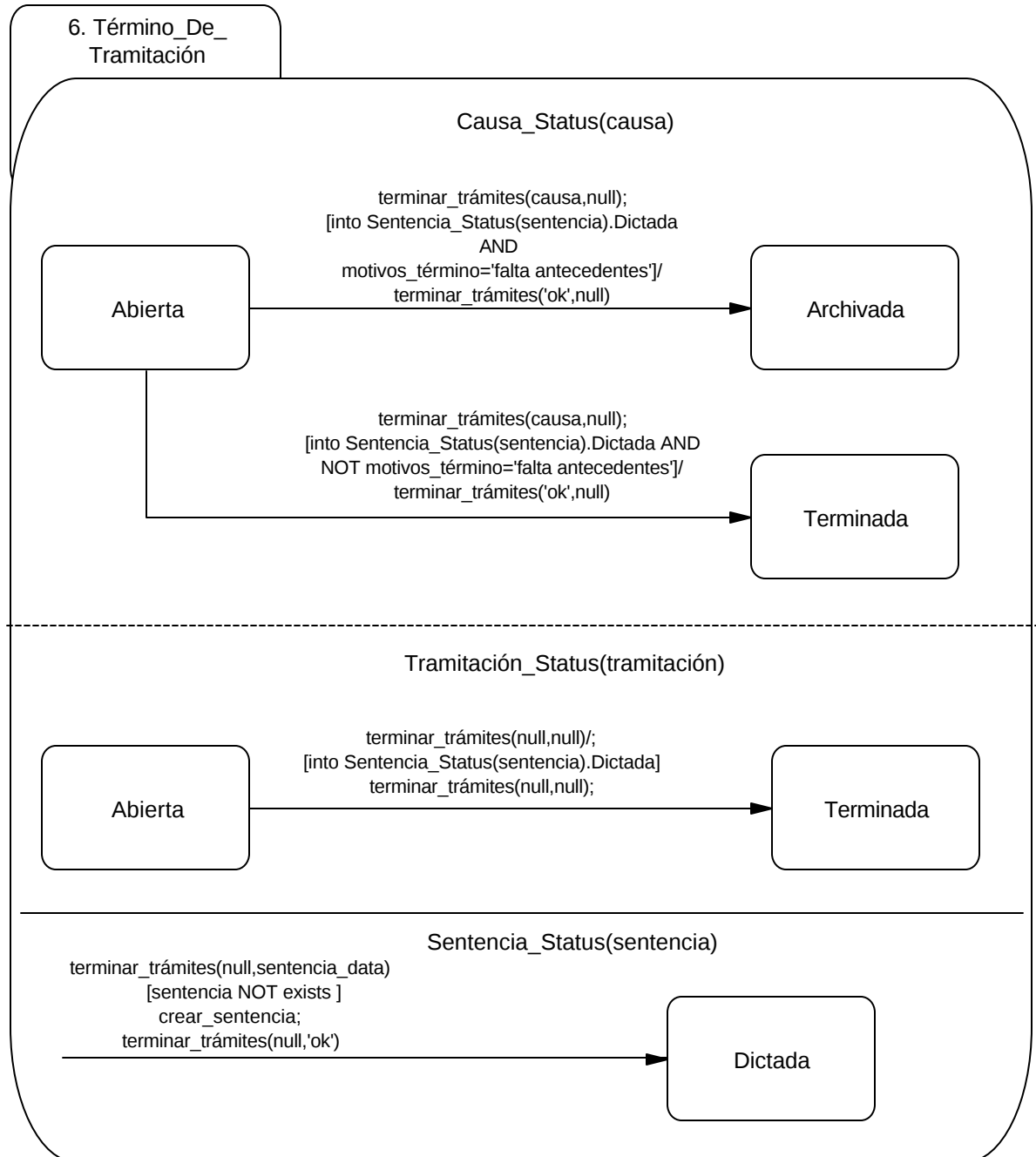
2.4 Diagrama HLS del proceso “Recepción_De_Exhorto”



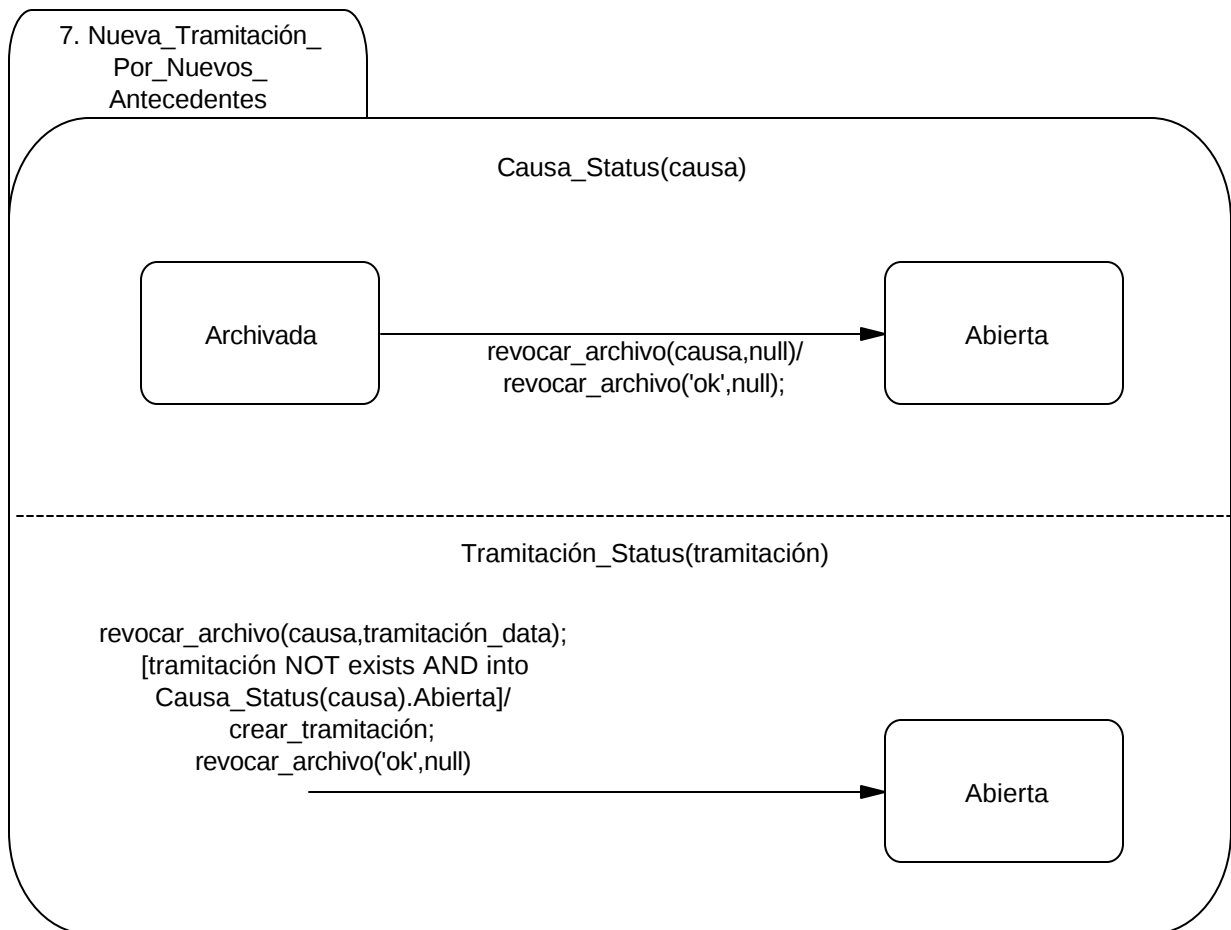
2.5 Diagrama HLS del proceso “Término_De_Exhorto”



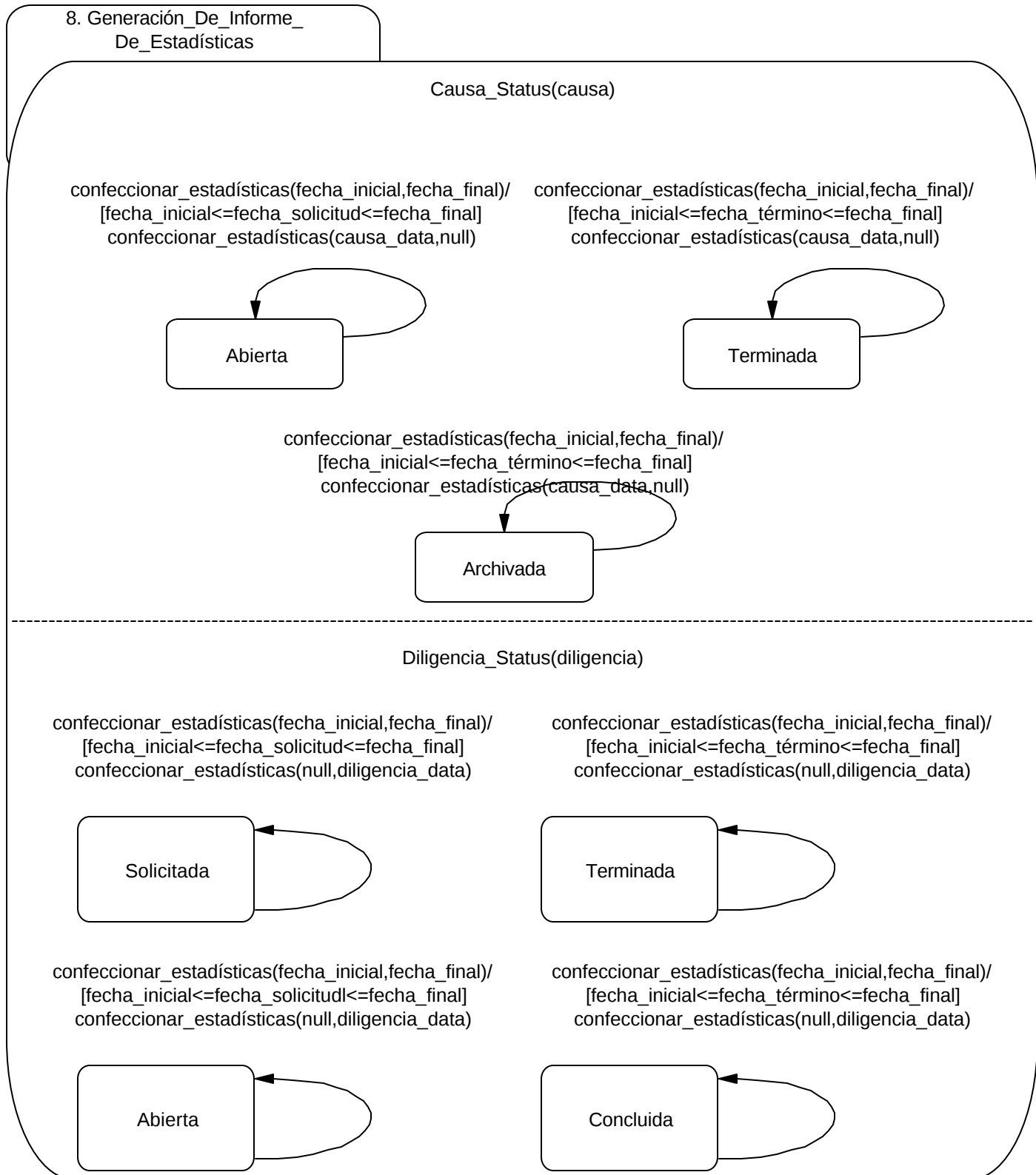
2.6 Diagrama HLS del proceso “Término_De_Tramitación”



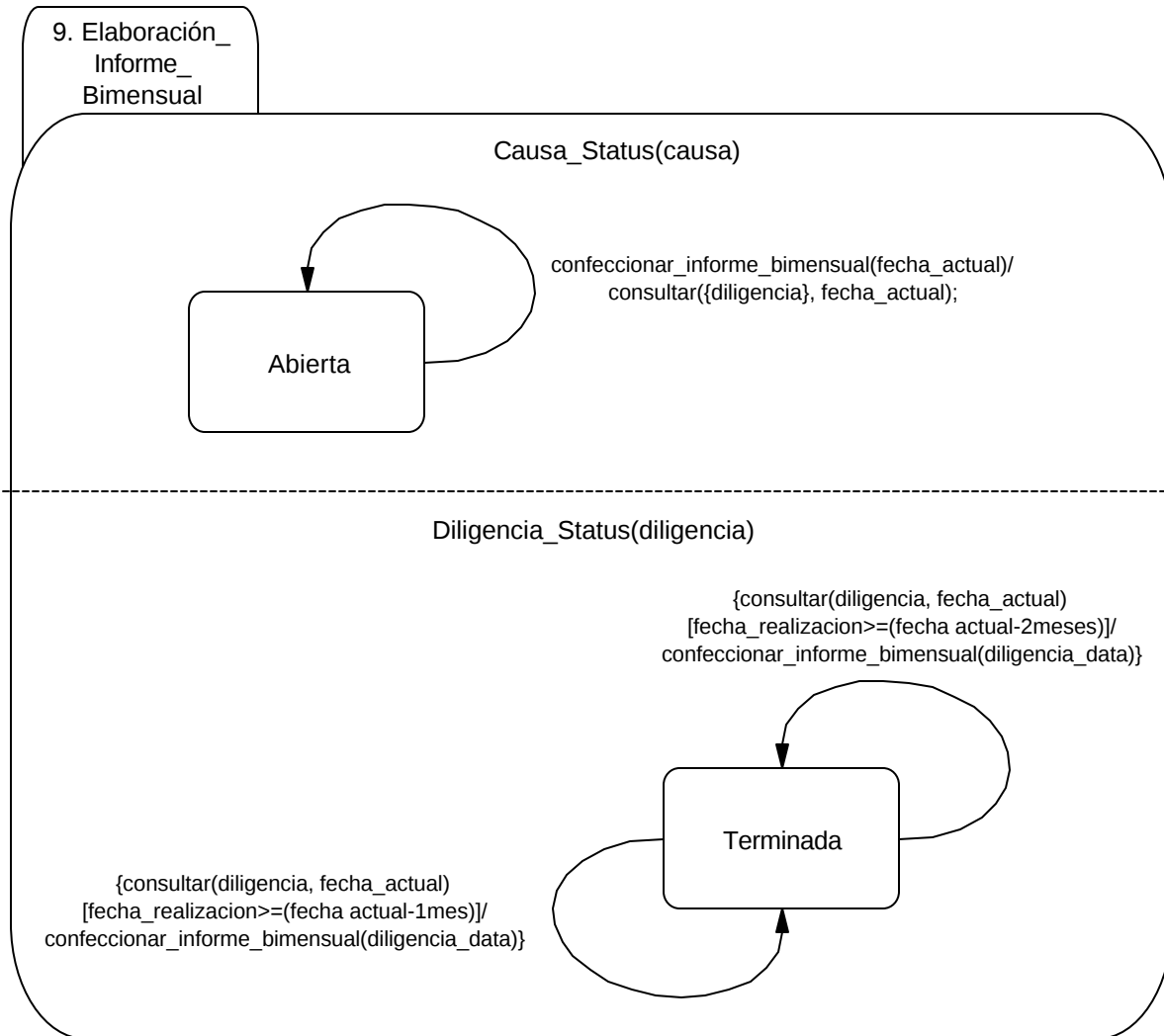
2.7 Diagrama HLS del proceso “Nueva Tramitación Por Nuevos Antecedentes”



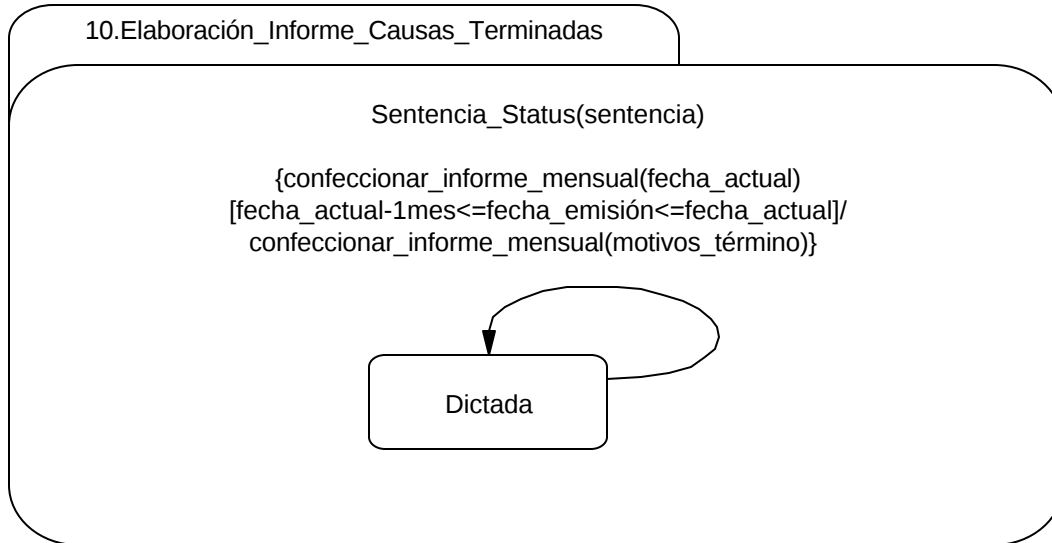
2.8 Diagrama HLS del proceso “Generación_De_Informe_De_Estadísticas”



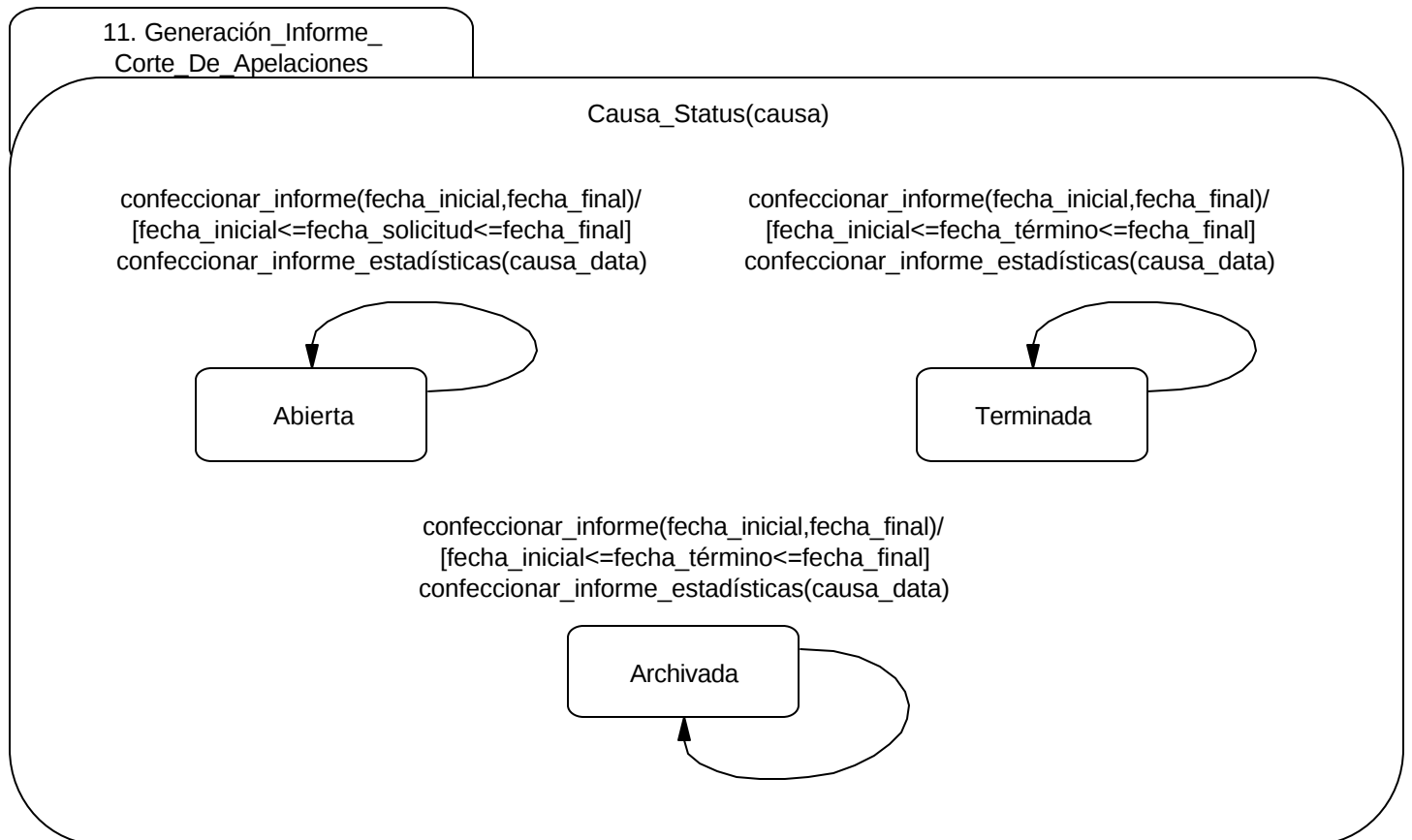
2.9 Diagrama HLS del proceso “Elaboración_Informe_Bimensual”



2.10 Diagrama HLS del proceso “Elaboración_Informe_Causas_Terminadas”



2.11 Diagrama HLS del proceso “Generación_Informe_Corte_De_Apelaciones”



3 Diccionario de datos

causa_data=@causa+actuario+juzgado+rol+tipo_causa+materia_principal+materia_secundaria+partes_involucradas+fecha_solicitud+fecha_término

diligencia_data=@diligencia+tipo_exhorto+fecha_solicitud+fecha_realización+juzgado_origen+juzgado_ejecutor

exhorto_data=@exhorto+fecha_solicitud+fecha_realización+juzgado_origen+juzgado_ejecutor

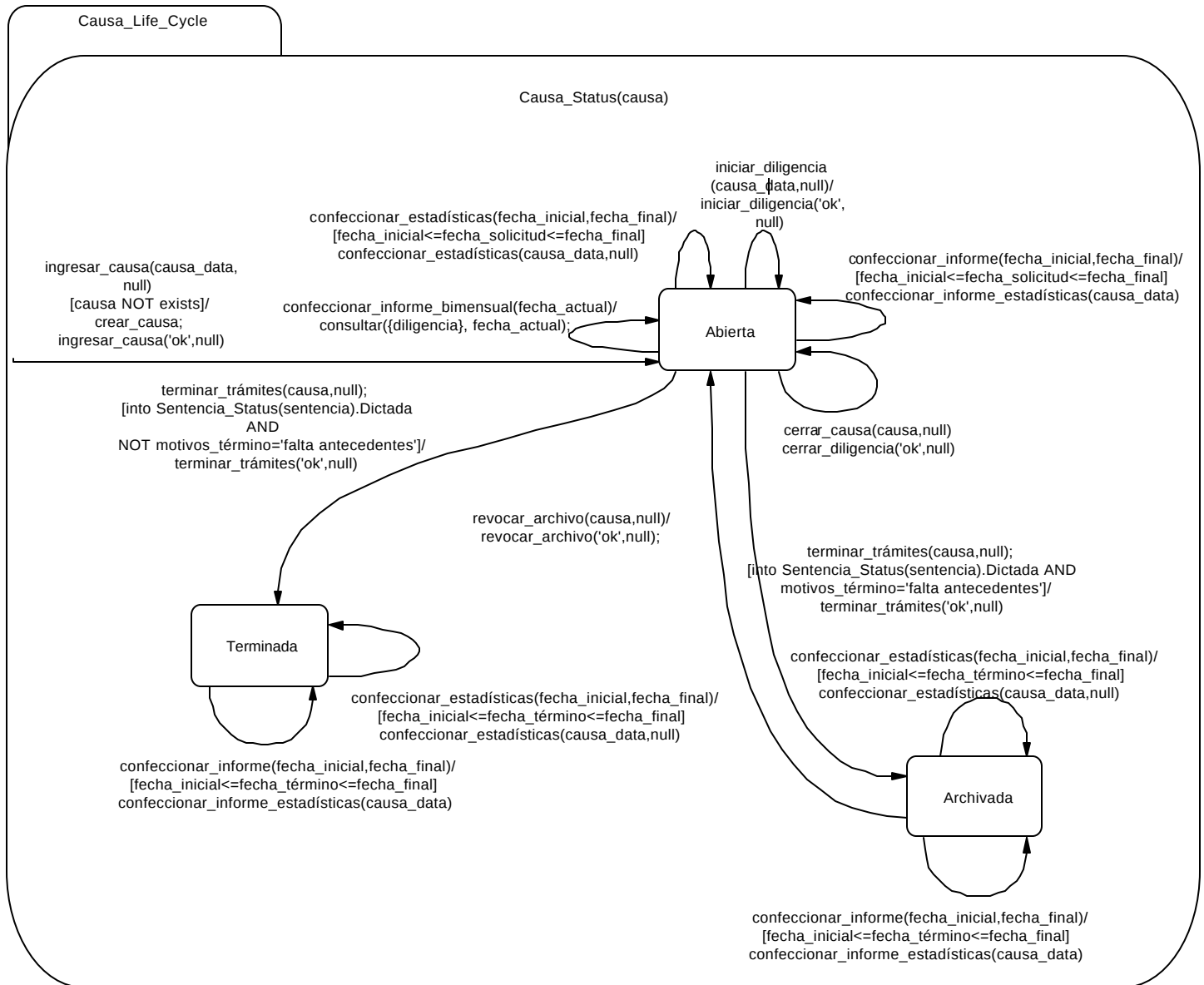
sentencia_data=@sentencia+rol_causa+motivos_término+fecha emisión

tipo_exhorto=[Verdadero, Falso]

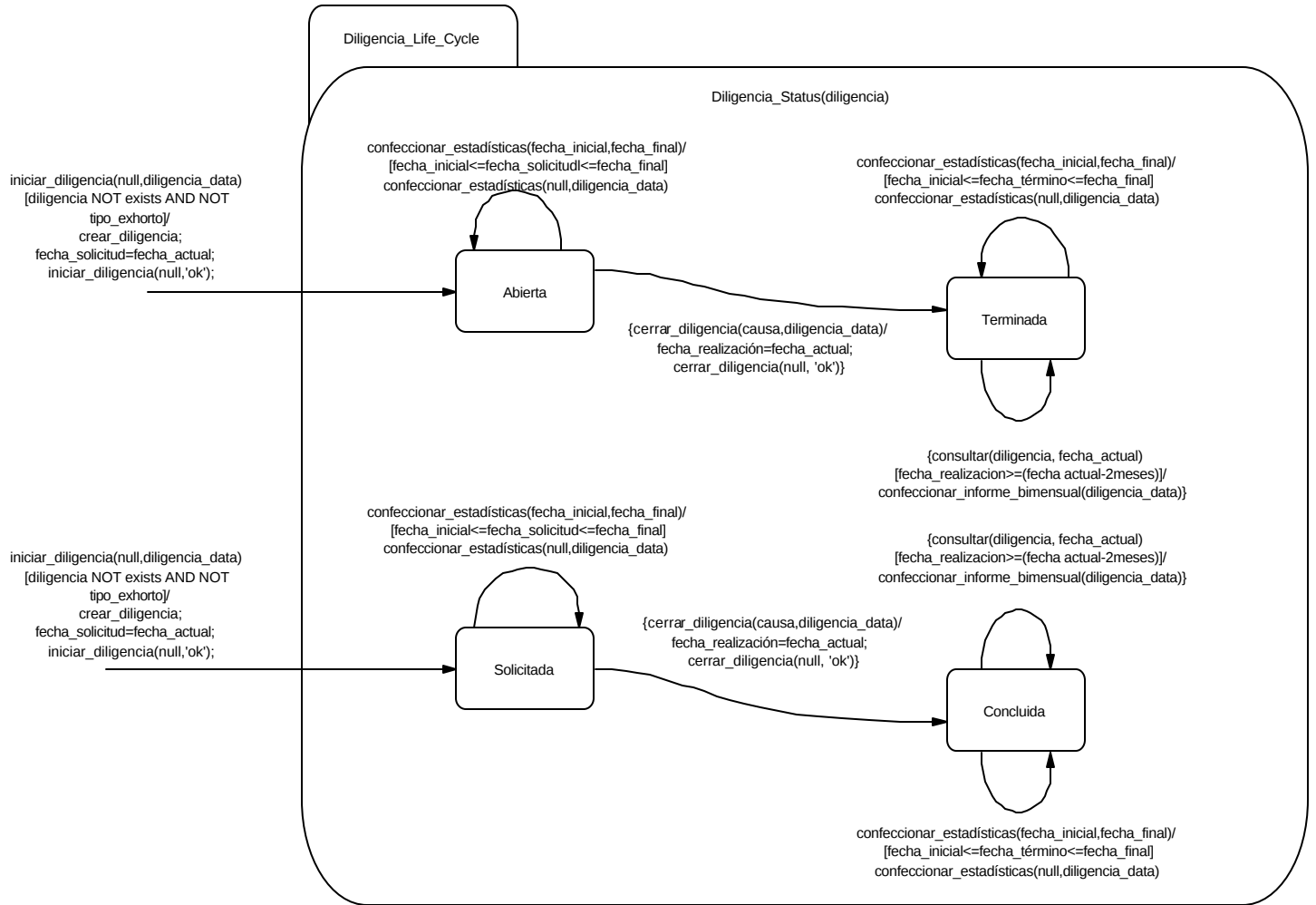
tramitación_data=@trámite+fecha_inicio+fecha_término

4 Diagramas HLS de Ciclos de Vida

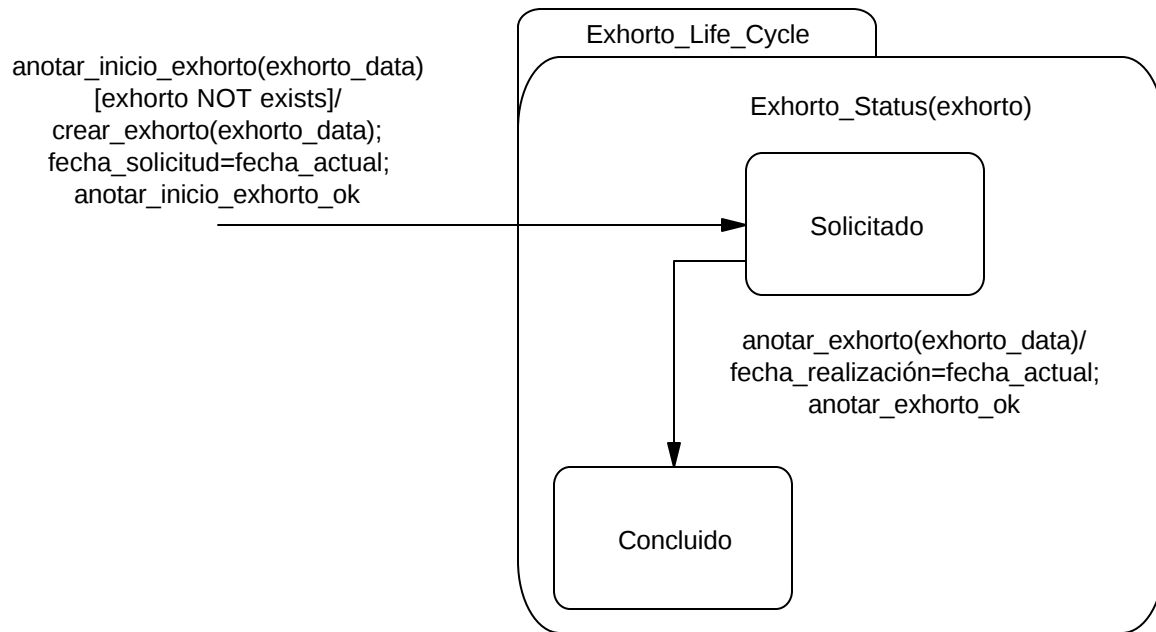
4.1 Diagrama HLS de ciclo vida de la variable causa



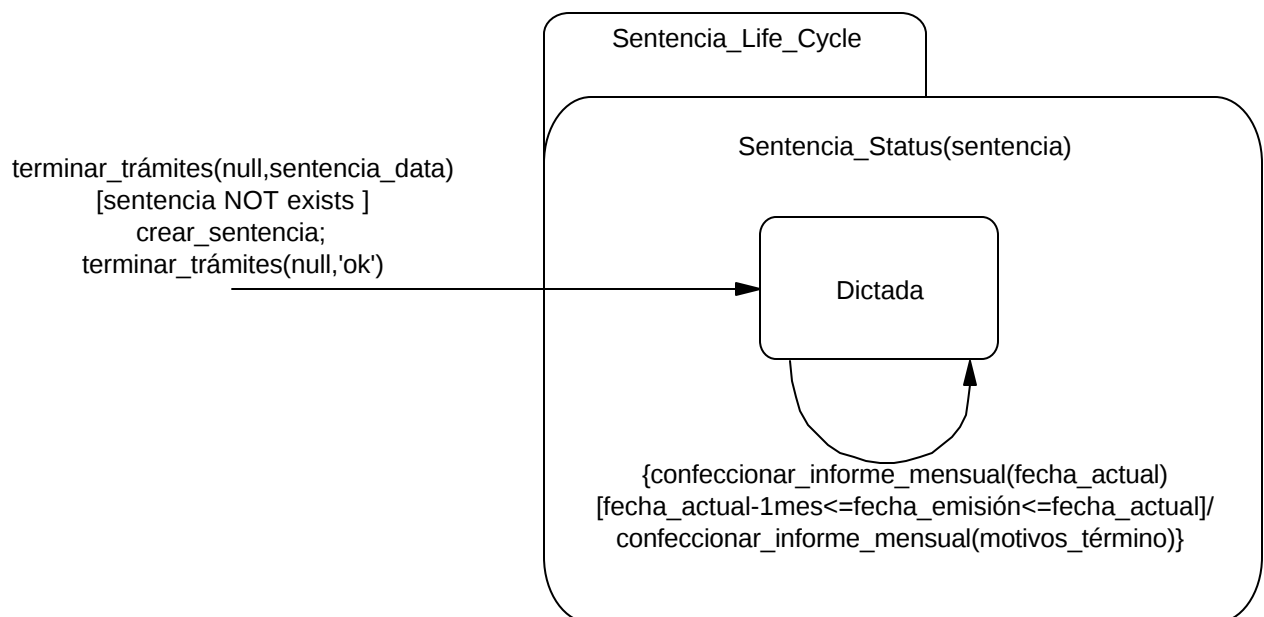
4.2 Diagrama HLS de ciclo vida de la variable diligencia



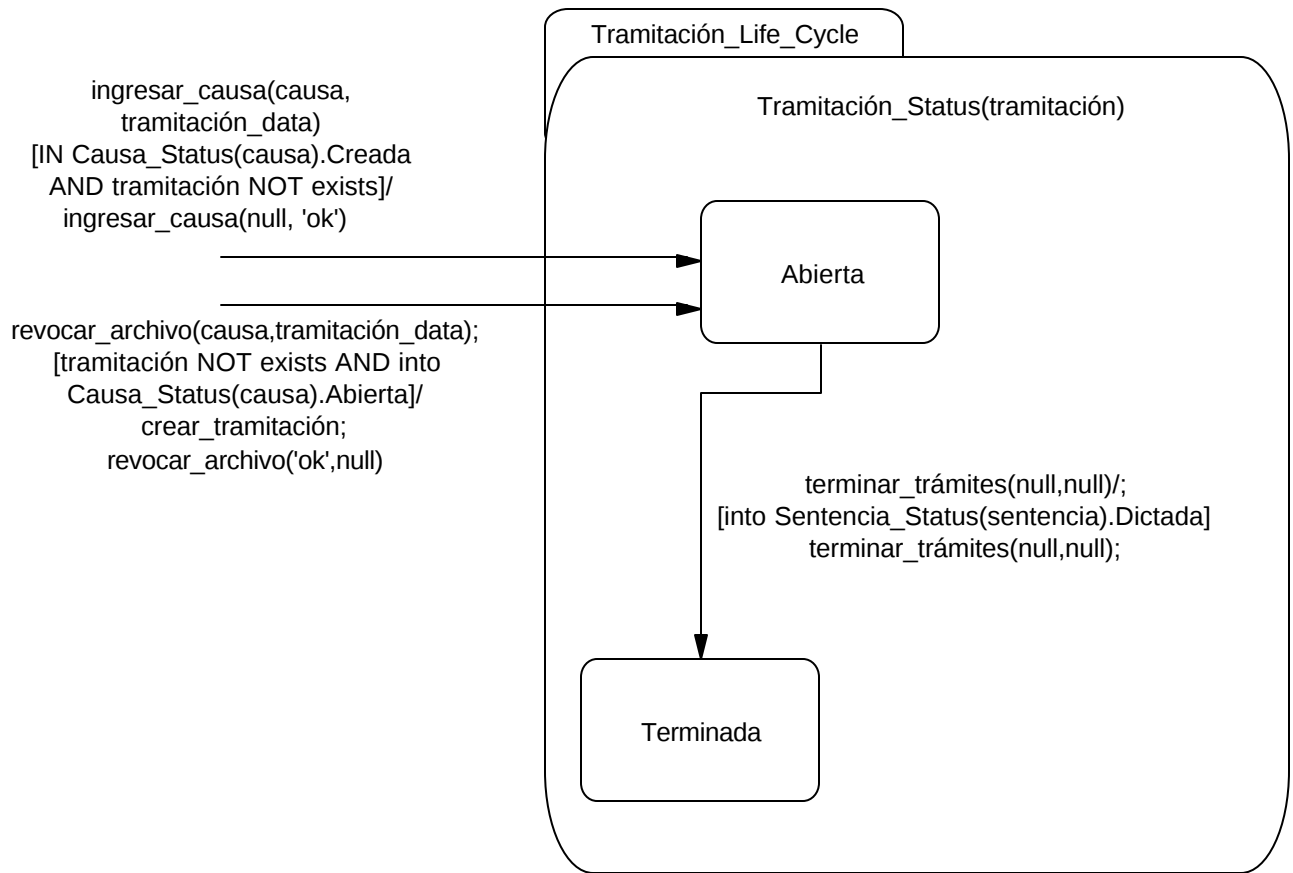
4.3 Diagrama HLS de ciclo vida de la variable exhorto



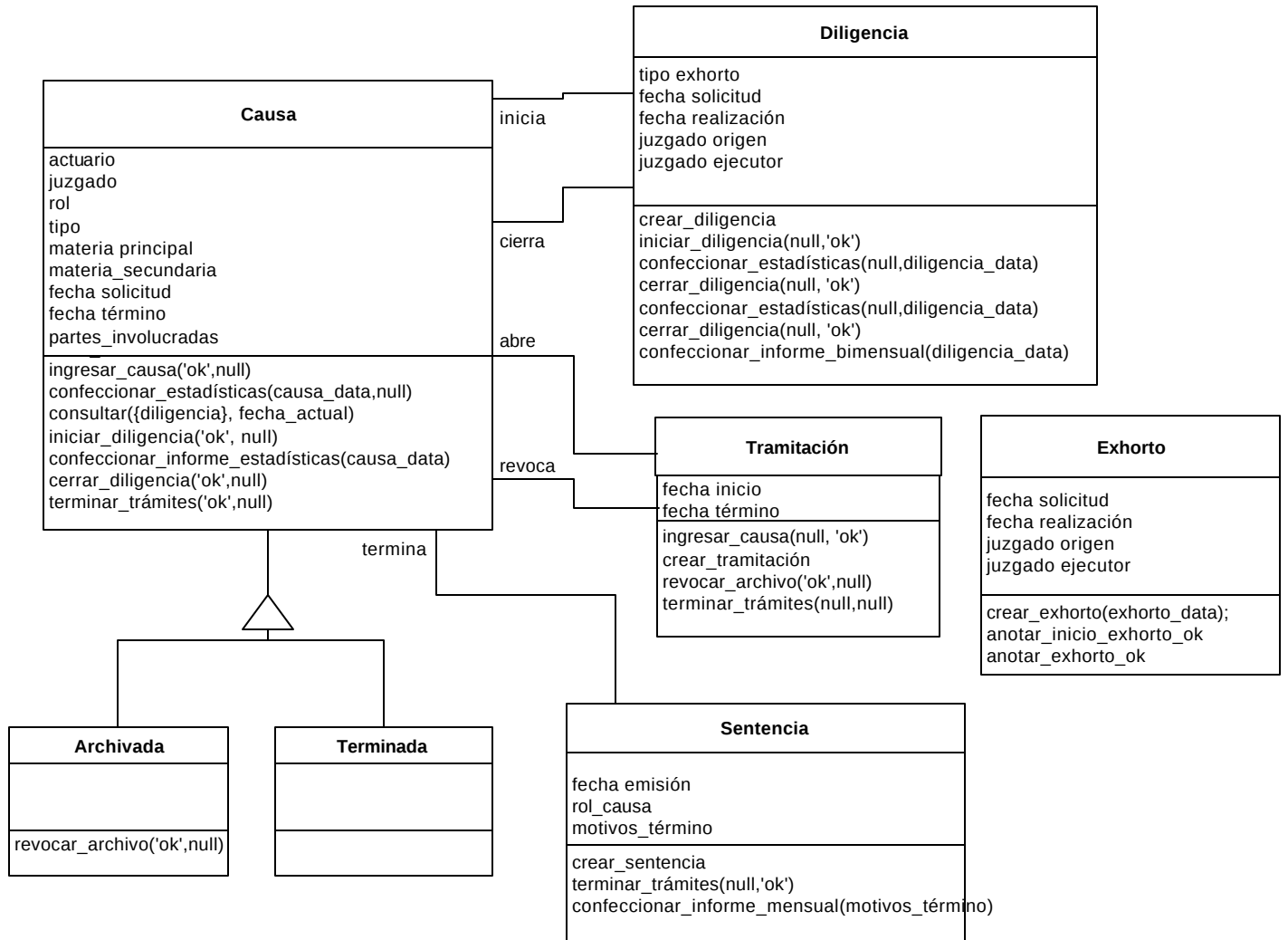
4.4 Diagrama HLS de ciclo vida de la variable sentencia



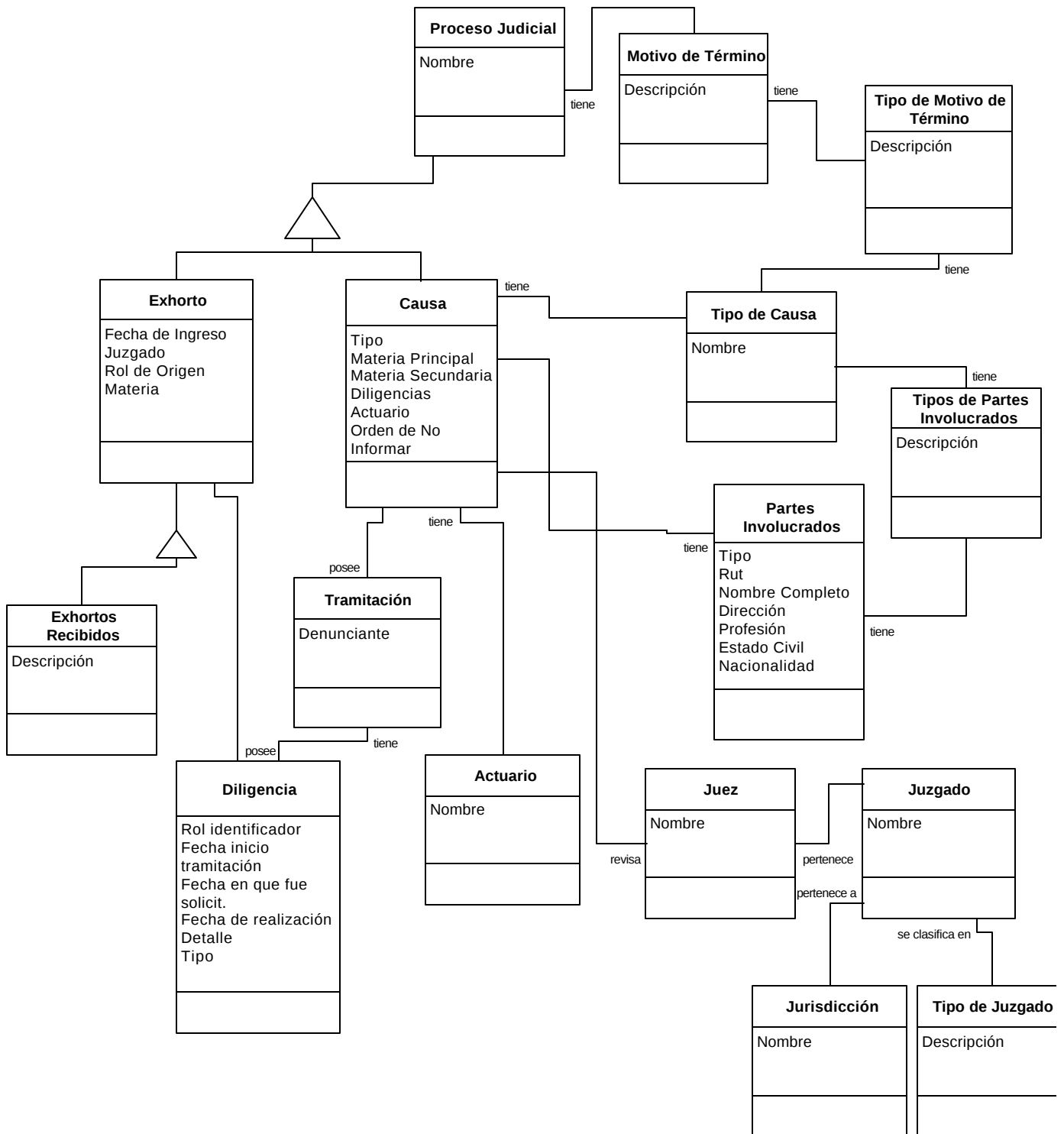
4.3.4.5 Diagrama HLS de ciclo vida de la variable tramitación



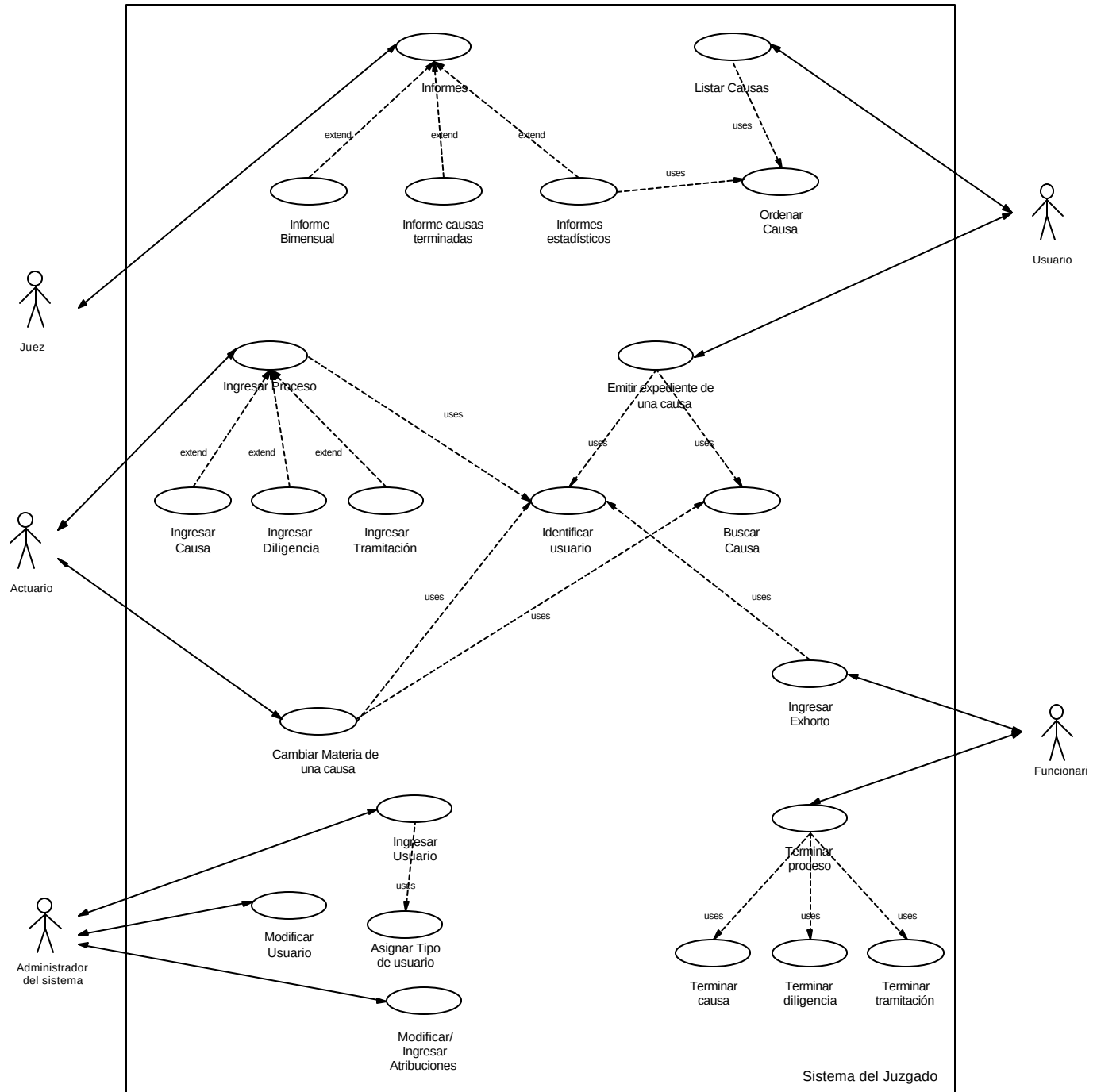
4.3.5 Modelo Estructural Orientado a Objetos



6 Modelo estructural de objetos según técnica OMT



7 Modelo use case



1. Ingresar usuario

Para ingresar un *usuario* al sistema, el *administrador* debe registrar el *nombre* del nuevo usuario que desea ingresar, además de asignarle un *tipo de usuario* mediante el use case Asignar Tipo Usuario, para que sea reconocido por las demás funciones del sistema.

1.1 Asignar tipo de usuario

El sistema posee diferentes *tipos de usuarios*, los cuales tienen *atribuciones* que le son propias a cada tipo, y que les permiten o no acceder a las diferentes opciones del sistema. Asignar tipo de usuario implica dar al usuario las atribuciones que le correspondan para operar dentro del sistema.

2. Modificar usuario

Si el *usuario* ya existe, el *administrador del sistema* puede modificar sus datos, como puede ser su *nombre* o *tipo*. Para ello se deben reemplazar los datos existentes por los nuevos.

3. Modificar/ Ingresar atribuciones

El *administrador del sistema* pueden ingresar o modificar las *atribuciones* existentes en el sistema, dependiendo de las opciones que éste presente, las que posteriormente podrán ser asignadas a un *tipo de usuario* en particular.

4. Identificar usuario

El *usuario* debe ingresar su identificación para que su *tipo* sea reconocido por el sistema, esto se debe a que cada *tipo de usuario* del sistema tiene distintas *atribuciones* para su uso.

5. Ingresar proceso

Al sistema se deben ingresar ya sea causas, diligencias o tramitaciones según sea el caso, mediante los use cases Ingresar causa, Ingresar diligencia e Ingresar tramitación.

5.1 Ingresar causa

Cada vez que llega una *causa* al juzgado, ésta debe ser ingresada al sistema por el *actuuario*, quien se encarga de mantener esta causa. Este debe asignar el *rol identificador* de la causa, seleccionar el *tipo de causa*, y anotar la *fecha de inicio* de su tramitación. Cuando se selecciona el *tipo de causa*, inmediatamente queda determinado el *motivo de término* de la causa y los *tipos de partes involucradas*.

5.2 Ingresar diligencia

Cuando una *diligencia* es realizada dentro de una *causa*, ésta debe ser registrada en el sistema, considerando la *fecha en que fue solicitada*, la *fecha en que fue efectivamente realizada* el *responsable* de la misma y el *detalle* correspondiente. Para esto se debe identificar y seleccionar la *causa* y anotar la nueva *diligencia*, luego de las que han sido anotadas anteriormente.

5.3 Ingresar tramitación

Cuando se inicia una *tramitación* para una *causa*, ésta debe consignarse dentro del sistema. Para esto se debe ubicar la *causa* y anotar la *fecha de inicio* de la tramitación.

6. Ingresar exhorto

Primero debe ser identificado el usuario que desea utilizar el sistema mediante el use case Identificar usuario. Al igual que para las causas, una vez que el juzgado recibe un exhorto proveniente de otro juzgado, este debe ser ingresado al sistema, asignándose la *fecha de ingreso*, *juzgado*, *rol de origen* y *materia*.

7. Cambiar materia de una causa

Primero debe ser identificado el usuario que desea utilizar el sistema mediante el use case Identificar usuario. Luego debe buscarse la causa a la cual se le cambiará la materia, según el use case Buscar causa. Una *causa* puede cambiar de *materia* principal.

8. Identificar usuario

El *usuario* debe ingresar su identificación para que su *tipo* sea reconocido por el sistema, esto se debe a que cada *tipo de usuario* del sistema tiene distintas *atribuciones* para su uso.

9. Buscar causa

El sistema debe revisar todas las *causas* hasta que encuentre la que requiere.

10. Listar causas

Este informe permite obtener la lista de *causas*. Para esto el solicitante debe especificar si la desea por *nombre* o por *tipo de causa*. Una vez hecho esto el sistema revisa cada una de las causas, consignándolas en la lista en el orden deseado, utilizando el use case Ordenar causa.

11. Ordenar causa

El sistema da al *usuario* las *alternativas de orden* para que esta elija entre ellas la que más se acomode a sus requerimientos. Una vez definido el orden por el *usuario*, el sistema entrega una lista ordenada bajo este criterio.

12. Emitir expediente de una causa

Primero debe ser identificado el usuario que desea utilizar el sistema mediante el use case Identificar usuario. El *usuario* debe especificar la *causa* que requiere, la cual es buscada según el use case Buscar causa. Luego verificar si tiene “orden de no informar”, en este caso si el *usuario* es el *juez*, se entrega toda la información relativa a la causa, si no lo es, el informe no es emitido.

13. Informes

Tiene como objetivo entregar al usuario del sistema los informes que requiera según los use cases Informe bimensual, Informe causas terminadas e Informes estadísticos.

13.1 Informe Bimensual

Al ser requerido este informe por el *actuuario*, el sistema debe revisar cada una de las causas y consignar la última diligencia para cada una. Esto debe ser entregado en una lista que contiene la identificación de la *causa*, la *diligencia* marcada como última en el informe Bimensual anterior y la *última diligencia* existente para este informe.

13.2 Informe Causas Terminadas

El informe de *causas terminadas* es emitido mensualmente. Una vez requerido este informe, el sistema debe revisar cada una de las causas y verificar si se trata de una causa terminada o no. El informe de *causas terminadas* muestra el *motivo de término* de las causas terminadas en el mes considerado.

13.3 Informes estadísticos

Cuando el *juez* requiere este informe, el sistema debe recorrer todas las causas y registrar la *fecha de ingreso* y la *fecha de término* de cada una de ellas, pudiendo ordenarlas por *tipo*, por *materia* o por *motivo de término*.

14. Terminar proceso

Cuando un *proceso* es finalizado esto se debe consignar en el sistema, mediante los use cases Terminar causa, Terminar diligencia y Terminar tramitación. Anotando los *motivos de término* necesarios.

14.1 Terminar causa

Tiene como objetivo dejar constancia en el sistema que la causa se ha dado por terminada.

14.2 Terminar diligencia

Tiene como objetivo dejar constancia en el sistema que la diligencia se ha dado por terminada.

14.3 Terminar tramitación

Tiene como objetivo dejar constancia en el sistema que la tramitación se ha dado por terminada.

Bibliografía

- [Bustos96] Bustos, Guillermo. **Uma Proposta de Modelagem Conceitual de Sistemas Dirigida por Comportamiento**. Tesis para optar al grado de Doctor en Ciencias de Computación. Porto Alegre: CPGCC-UFRGS, 1996. (en portugués)
- [Jacobson92] Jacobson, Ivar et al. **Object-Oriented Software Engineering: A Use Case Driven Approach**. Wokingham: Addison-Wesley, 1992.
- [Rumbaugh91] Rumbaugh, James et al. **Object-Oriented Modeling and Design**. Englewood Cliffs: Prentice Hall, 1991.