



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Ingeniería de Software I
Primer Cuatrimestre de 2016

Integrante	LU	Correo electrónico
Iván Arcuschin	678/13	iarcuschin@gmail.com
Federico De Rocco	408/13	fedede.183@hotmail.com
Martín Jedwabny	885/13	martiniedva@gmail.com
Alan Lebedinsky	802/11	alanlebe@gmail.com
José Massigoge	954/12	jmmassigoge@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introducción	3
2. Cambios con respecto al TP1	4
3. Casos de Uso	5
3.1. Interacciones con el sistema: representado con casos de uso	5
3.2. Relación con otros modelos	5
3.3. En las que actúa el Cliente	6
3.4. En las que actúa el Proveedor	9
3.5. En las que actúa el Gerente	12
3.6. En las que actúa el PM	16
3.7. En las que actúa el User	19
3.8. En las que actúa el Data Entry	20
3.9. En las que actúa el Contador	22
4. Diagrama de Actividad	24
4.1. Descripción general	24
4.2. Relación con otros modelos	24
4.3. Vistas	25
5. Maquinas de Estado Finito	30
5.1. Descripción general	30
5.2. Relación con otros modelos	30
5.3. Vistas	31
6. Modelo Conceptual	34
6.1. Descripción general	34
6.2. Relación con otros modelos	34
6.3. Observaciones de las Vistas	35
6.4. Vistas	36
6.5. OCL	39
7. Conclusiones	43

1. Introducción

En el anterior informe se especifico una lista de requerimientos y se establecieron objetivos, con los cuales, se llevaría acabo el sistema pedido por la empresa *DC Construcciones*. A partir de esto, elegiremos cuales de las alternativas propuestas en el modelo de objetivos utilizaremos para la construcción de este sistema. Una vez completada la elección, se realizaran una serie de documentos en base a distintas técnicas de especificación, entre las cuales contarán: modelo de casos de uso, modelo conceptual ampliados con OCL, máquinas de estados finitos (o LTSs/FSMs), y/o diagramas de actividad. Según consideremos conveniente elegiremos una o otra para representar determinados detalles de nuestro problema.

A continuación redactaremos las elecciones que tomamos a partir del modelo de objetivos:

- En el o-refinamiento del objetivo Mantener[Mostrar proveedores de forma inteligente] seleccionamos la opción que contiene los objetivos Mantener[Proveedores ordenados por ranking] y Lograr[Aplicar filtros para proveedores (rubro, ubicación, geografía)]. El ranking mencionado es generado a partir de las encuestas de fin de proyecto.
- En el o-refinamiento del objetivo Mantener[orden de PM según ranking y cantidad de proyectos actuales] seleccionamos la opción que ordena a los PMs por ranking y cantidad de proyectos. Este ranking es generado a partir de las encuestas de fin de proyecto.
- En el o-refinamiento del objetivo Mantener[Estado de proyectos actualizados] elegimos la opción que incluye como subobjetivo Mantener[PM asignado al proyecto actualiza el estado periódicamente en el sistema].
- En el o-refinamiento del objetivo Lograr[Mejorar finalización de los proyectos] elegimos la opción que incluye como subobjetivo Lograr[El proyecto se marca como finalizado =¿entonces se realizan las encuestas]. Esta elección es necesaria para que los dos primeros o-refinamientos se puedan utilizar.

Seleccionadas las alternativas, pasaremos establecer con que modelos representaremos cada detalle de nuestro problema.

- Casos de uso: usaremos esta técnica para representar las interacciones que ocurren entre los distintos actores que conforman la empresa(gerente, PM, proveedor, cliente, data entry) y el sistema. Completaremos el diagrama con los detalles de los casos de uso, donde presentaremos la serie de pasos que representan una determinada interacción con nuestro sistema.
- Diagrama de Actividad: usaremos esta técnica para representar el ciclo de vida de un proyecto, y casos extraordinarios en la etapa de seguimiento, cancelación del proveedor y cambio de PM.
- Máquinas de estados finitos: las utilizamos para mostrar como funcionan las actualizaciones de estado que realiza el PM para un proyecto durante la etapa de seguimiento (con y sin data entry). Además, mostramos como son las interacciones a través del sistema durante las negociaciones del alcance (PM con Cliente) y del presupuesto (Gerente con Cliente).
- Modelo Conceptual: usaremos esta herramienta para modelar los diferentes agentes relevantes desde el punto de vista del sistema así como también sus diferentes estados y relaciones entre los mismos. Es decir, con el modelo conceptual hablaremos sobre la estructura misma de las entidades del problema y como el sistema entiende todo lo involucrado según las características que los hacen relevantes.

En las siguientes secciones de este informe empezaremos por formular cambios respecto a la etapa anterior del proyecto, para luego desarrollar cada herramienta utilizada, partiendo de una descripción general de las mismas para luego aclarar las relación entre la herramienta analizada y el resto de las herramientas utilizadas. La necesidad de aclarar esta relación surge del hecho de que cada herramienta tiene distinto poder expresivo, por lo tanto para lograr comunicar el comportamiento del sistema a desarrollar debemos utilizar cada una para diversas situaciones, lo cual genera escenarios que son transversales a los distintos modelos generados, es decir, para especificarlos, es necesario utilizar más de una herramienta.

2. Cambios con respecto al TP1

En base al feedback obtenido de la empresa *DC Construcciones*, y un análisis detenido de los requerimientos, llegamos a la conclusión de que era necesario realizar los siguientes cambios a los objetivos planteados en la primera etapa:

- El procedimiento por el cual el sistema logra saber que un proyecto a comenzado, necesario para poder exigir, posteriormente, actualizaciones al PM, no estaba incluido en las asunciones. Establecemos que el PM es quien marca, en el sistema, como iniciado el proyecto.
- Para el caso en el que un proveedor cancela durante la realización de un proyecto, se había tomado la asunción de que el proveedor cancela a través del sistema, pero esto va en contra de las prácticas comerciales de sentido común. Por esta razón, ahora asumimos que el proveedor cancela comunicándose directamente con el PM, quien después refleja esto en el Sistema.
- Una vez que el PM informa que canceló un proveedor en el Sistema, este envía una notificación al Gerente. Dado que puede haber alguna parte del proyecto ya realizada, el PM redefine el alcance antes de elegir un nuevo proveedor. Previo a redefinir el alcance el PM evalúa negativamente en el Sistema al proveedor que canceló.
- El gerente debe validar nuevamente el proyecto ante el cambio de proveedor y la redefinición del alcance.
- El Cliente puede comunicar un malestar con el PM, en cuyo caso el PM será apartado del proyecto (este desplazamiento se realiza por fuera del sistema). El PM también puede apartarse por voluntad propia, acción que realiza comunicándose directamente con el gerente.
- El sistema notifica al Cliente del cambio de PM.
- Cuando el PM es apartado de un proyecto, el gerente lo evalúa negativamente.

Estos cambios serán reflejados en nuestras especificaciones.

3. Casos de Uso

3.1. Interacciones con el sistema: representado con casos de uso

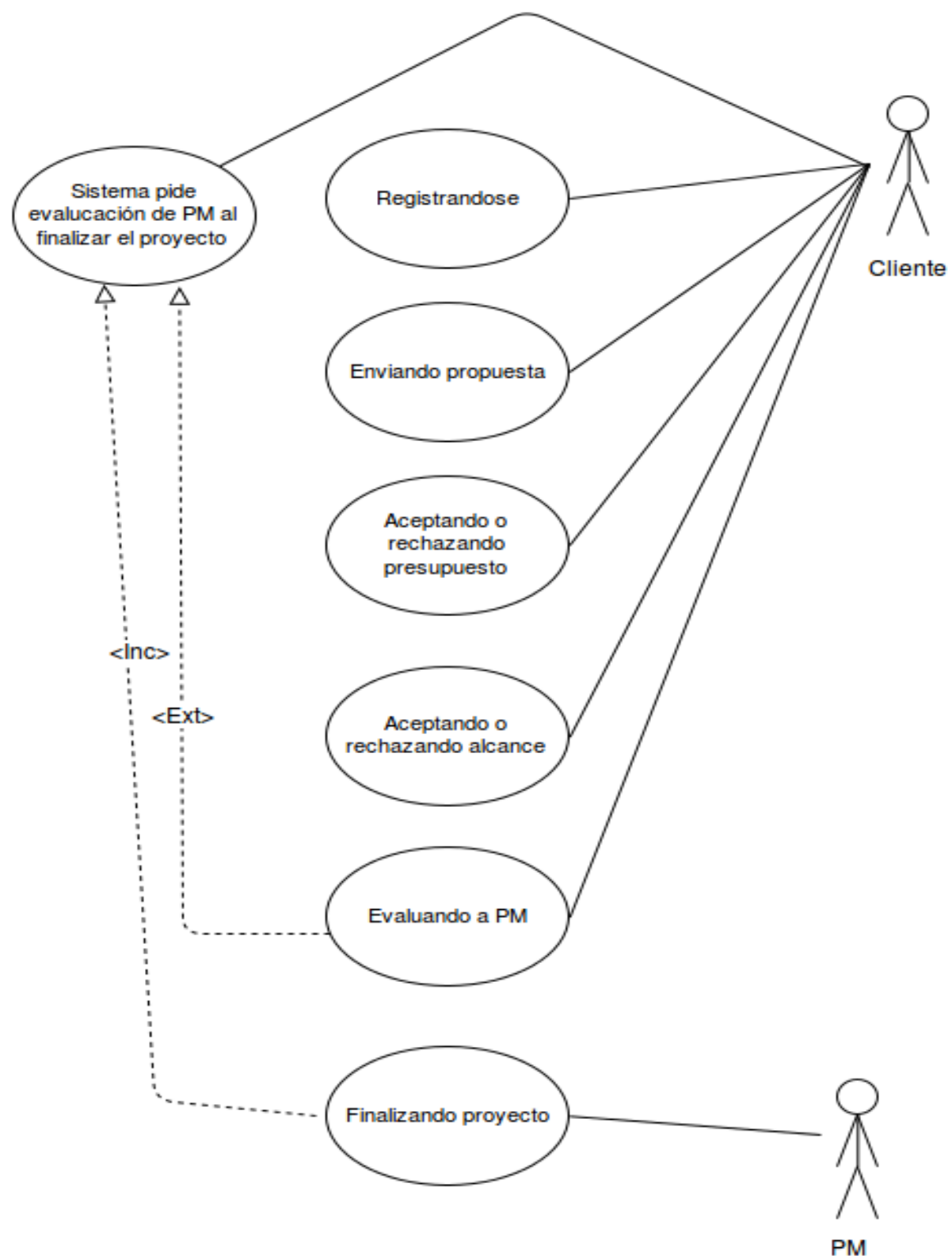
En primera instancia tenemos que representar las acciones que los diversos actores realizan en el sistema o a través de él. Para esto modelaremos nuestro proyecto usando casos de uso, y mostraremos los detalles de cada uno de estos.

3.2. Relación con otros modelos

Como mencionamos antes, el modelo de casos de uso nos permite representar las interacciones que existen entre el sistema y los actores externos a este. Utilizando esta técnica podemos definir el alcance del propio sistema en cuanto a que acciones toma ante cada interacción y, además, nos permite definir la interfaz que nuestro sistema tendrá.

A pesar de todo lo dicho, este modelo es incapaz de modelar todas las interacciones que ocurran fuera del sistema. Esta no es la única limitación que posee, dado que si bien podemos modelar un cierto orden en cuanto a los procesos que la empresa realiza en el sistema, utilizando las Pre y Post condiciones de los detalles, no podemos dejar claro ningún paralelismo que pueda existir entre ellas. Para eso y para otras cosas serán necesarios los demás modelos.

3.3. En las que actúa el Cliente



Sección 3.3 En las que actúa el Cliente

CU1: Registrándose	
Actor: Cliente	
Pre: true	
Post: El cliente está registrado en el sistema	
Curso normal	Curso alternativo
1- El cliente solicita al sistema que le permita registrarse	
2- El sistema confirma y pide al cliente que ingrese su nombre y clave	
3- El cliente ingresa su nombre y clave	3.1- El cliente ingresa un nombre o clave inválidos. Ir a Fin CU1
4- El sistema guarda el nombre y la clave del cliente	3.2- El cliente ya estaba registrado. No lo registra
5- Fin CU1	

CU2: Enviando propuesta	
Actor: Cliente	
Pre: El cliente está registrado en el sistema	
Post: El sistema guarda la propuesta del cliente, para que después la obtenga el gerente	
Curso normal	Curso alternativo
1- El cliente solicita al sistema que le permita enviar una propuesta	
2- El sistema confirma y pide al cliente que ingrese su nombre y clave	
3- El cliente ingresa su nombre, clave y su propuesta	3.1- El cliente ingresa un nombre o clave incorrectos. Ir a Fin CU2
4- El sistema guarda la propuesta del cliente	
5- Fin CU2	

CU3: Aceptando o rechazando presupuesto	
Actor: Cliente	
Pre: Gerente propuso presupuesto	
Post: El sistema guarda la aceptación del cliente	
Curso normal	Curso alternativo
1- El cliente recibe el presupuesto	
2- El cliente acepta la propuesta	2.1- El cliente rechaza el presupuesto. Ir a CU19
3- El sistema guarda la aceptación del cliente	
4- Fin CU3	

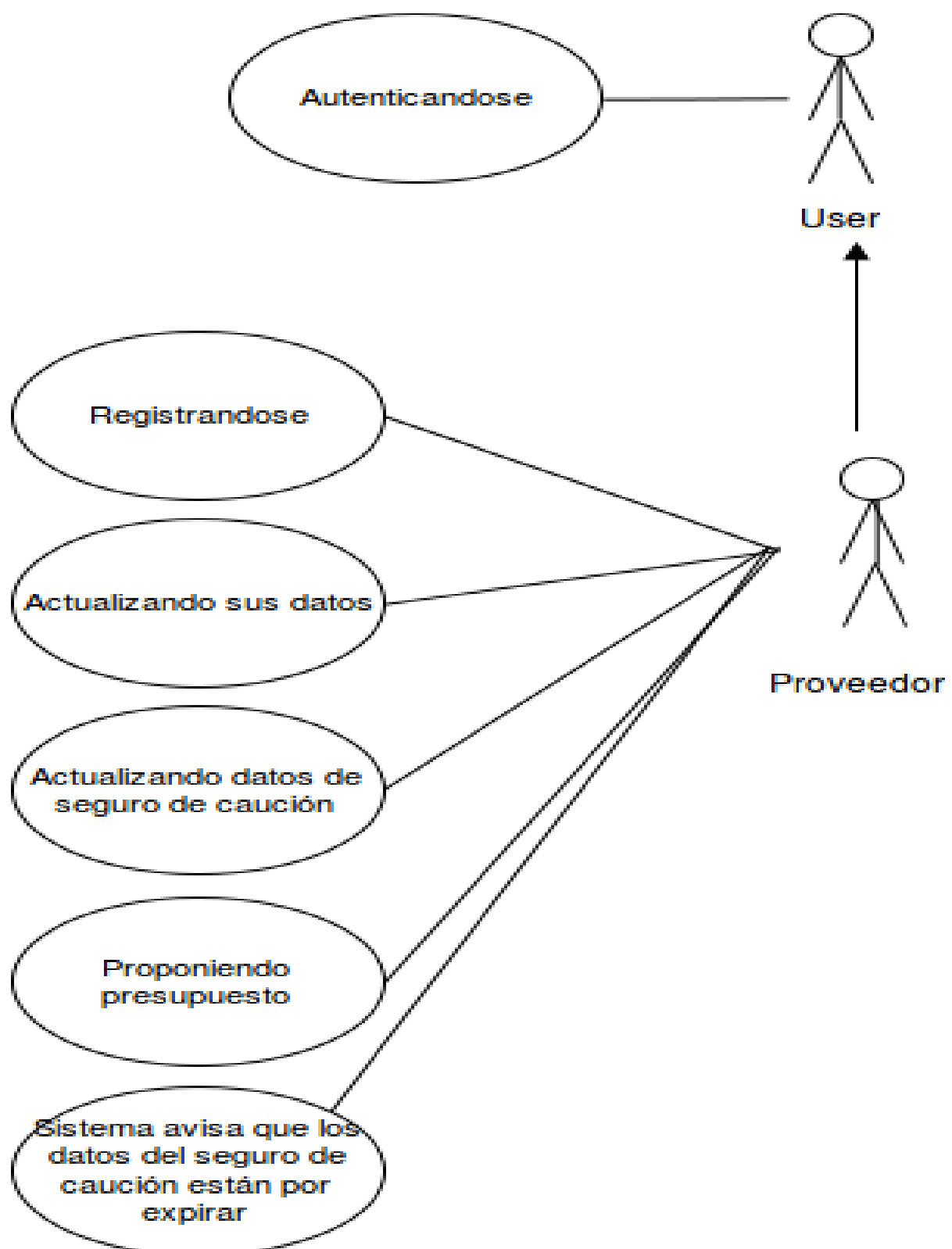
CU4: Aceptando o rechazando alcance	
Actor: Cliente	
Pre: PM propuso alcance	
Post: El sistema guarda la aceptación del cliente	
Curso normal	Curso alternativo
1- El cliente recibe el alcance	
2- El cliente acepta el alcance	2.1- El cliente rechaza el alcance. Ir a CU22
3- El sistema guarda la aceptación del cliente	
4- Fin CU4	

CU5: Sistema pide evaluación de PM al finalizar el proyecto	
Actor: Cliente	
Pre: El proyecto finalizo	
Post: El cliente recibe el pedido de evaluación de un PM	
Curso normal	Curso alternativo
1- El sistema se actualiza y observa que tiene un proyecto como finalizado	
2- El sistema guarda el pedido de evaluación del PM para que el cliente lo observe.	
3- Usa CU5	
4- Fin CU5	

Sección 3.3 En las que actúa el Cliente

CU6: Evaluando PM	
Actor: Cliente	
Pre: El cliente recibió del sistema un pedido de evaluación para el PM	
Post: El cliente evalúa al PM	
Curso normal	Curso alternativo
1- El cliente pide al sistema que le permita evaluar al PM	
2- El sistema confirma y pide al cliente su nombre y clave.	
3- El cliente ingresa su nombre y clave	3.1- El cliente ingresa un nombre o clave incorrectos. Ir a FinCU6
4- El sistema confirma al cliente y muestra la encuesta para evaluar al PM	
5- El cliente completa la encuesta y pide al sistema que la guarde	
6- El sistema guarda la encuesta	
7- Fin CU6	

3.4. En las que actúa el Proveedor



Sección 3.4 En las que actúa el Proveedor

CU7: Registrándose	
Actor: Proveedor	
Pre: true	
Post: El proveedor queda registrado en el sistema	
Curso normal	Curso alternativo
1- El proveedor solicita al sistema que le permita registrarse	
2- El sistema confirma y pide al proveedor que ingrese su nombre y clave	
3- El proveedor ingresa su nombre y clave	3.1- El proveedor ingresa un nombre o clave inválidos. Ir a Fin CU7
4- El sistema guarda el nombre y la clave del proveedor	
5- Fin CU7	

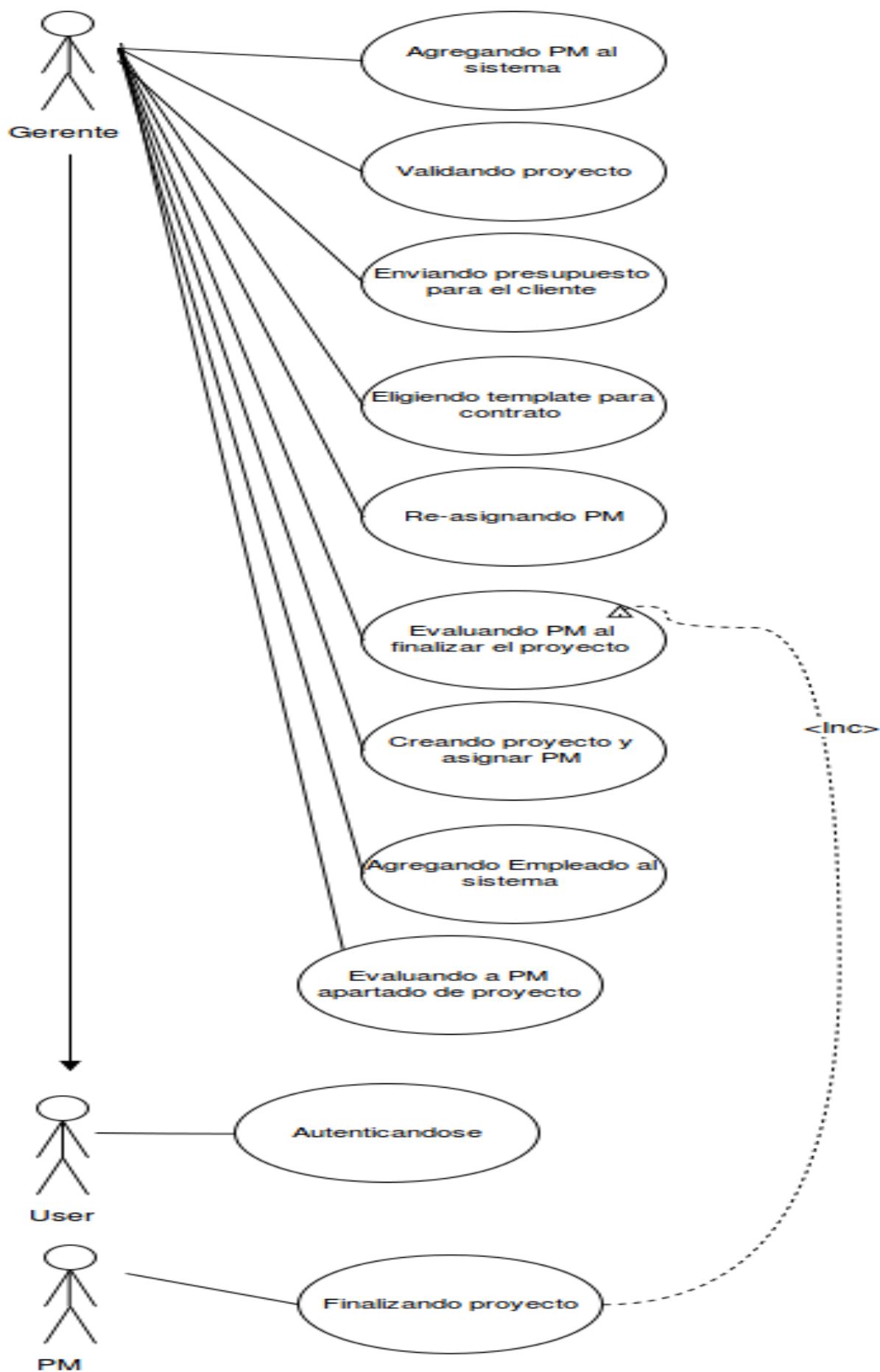
CU8: Actualizando sus datos	
Actor: Proveedor	
Pre: El proveedor está autenticado	
Post: El proveedor tiene actualizada su información en el sistema	
Curso normal	Curso alternativo
1- El proveedor solicita al sistema que le permita que actualice sus datos	
2- El sistema confirma y muestra los datos actuales del proveedor	
3- El proveedor modifica sus datos	
4- El sistema guarda los nuevos datos del proveedor	
5- Fin CU8	

CU9: Actualizando datos de seguro de caución	
Actor: Proveedor	
Pre: El proveedor está autenticado	
Post: El proveedor tiene actualizado los datos del seguro de caución	
Curso normal	Curso alternativo
1- El proveedor solicita al sistema que le permita que actualice sus datos de seguro de caución	
2- El sistema confirma y muestra los datos actuales del proveedor	
3- El proveedor carga su nuevo seguro de caución	
4- El sistema guarda los nuevos datos del proveedor	
5- Fin CU9	

CU10: Propone presupuesto	
Actor: Proveedor	
Pre: El proveedor está autenticado	
Post: El proveedor le envía el presupuesto al PM	
Curso normal	Curso alternativo
1- El proveedor solicita al sistema que le permita cargar el presupuesto para un proyecto	
2- El sistema confirma y le pide al proveedor cargar el presupuesto	
3- El proveedor carga el presupuesto	
4- El sistema guarda los datos y notifica que se guardaron correctamente	
5- Fin CU10	

CU11: Sistema avisa que seguro de caución está a punto de expirar	
Actor: Proveedor	
Pre: El proveedor está autenticado	
Post: El proveedor tiene un mensaje del sistema que le pide que actualice su seguro de caución	
Curso normal	Curso alternativo
1- El sistema detecta que el seguro de caución del proveedor está próximo a expirar	
2- El sistema envía al proveedor un mensaje pidiéndole que actualice su seguro de caución	
3- Fin CU11	

3.5. En las que actúa el Gerente



Sección 3.5 En las que actúa el Gerente

CU12: Agregando PM al sistema	
Actor: Gerente	
Pre: El gerente está autenticado	
Post: El PM es agregado al sistema	
Curso normal	Curso alternativo
1- El gerente solicita al sistema ingresar un nuevo PM	
2- El sistema confirma y pide al gerente que ingrese los datos	
3- El gerente ingresa los datos correspondientes al sistema	
4- El sistema valida que los datos sean correctos	
5- El sistema informa que se ingreso el PM al sistema correctamente	5.1-El sistema informa que la información es incorrecta o que los datos corresponden a un PM existente
	5.2- Ir a 2
6- Fin CU12	

CU13: Eligiendo template para contrato	
Actor: Gerente	
Pre: Cliente acepto presupuesto o PM re-asigno proveedor y el gerente está autenticado	
Post: Se eligió un template	
Curso normal	Curso alternativo
1- El gerente solicita al sistema templates para el contrato	
2- El sistema muestra los templates para el contrato y le pide al gerente que seleccione un template	
3- El gerente selecciona un template	
4- El sistema guarda la opción elegida	
5- Fin CU13	

CU14: Re-assignando PM	
Actor: Gerente	
Pre: Hay PM asignado a proyecto y el gerente está autenticado	
Post: Se modifico el PM asignado al proyecto	
Curso normal	Curso alternativo
1- El gerente solicita al sistema reasignar un PM	
2- El sistema confirma, muestra una lista de los proyectos en curso del gerente y le pide seleccionar el proyecto en el cual quiere reasignar al PM	
3- El gerente selecciona un proyecto	
4- El sistema muestra información del proyecto, al PM asignado y la lista de PM's (sin el asignado actualmente), le pide al gerente seleccionar un nuevo PM	
5- El gerente selecciona un PM	
6- El sistema informa que se reasigno correctamente	
7- Ir a CU20	
8- Fin CU14	

CU15: Creando proyecto y asignar PM	
Actor: Gerente	
Pre: El gerente está autenticado y Cliente envió una propuesta	
Post: Se creo un nuevo proyecto en el sistema con un PM asignado	
Curso normal	Curso alternativo
1- El sistema informa al gerente de la nueva propuesta y le pide que cree un proyecto	
2- El gerente crea un nuevo proyecto con los detalles de la propuesta	
3- El sistema muestra la lista de PM's y le pide al gerente elegir uno	
4- El gerente selecciona un PM	
5- El sistema informa que se creo el proyecto con el PM asignado exitosamente	
6- Fin CU15	

CU16: Evaluando PM al finalizar el proyecto	
Actor: Gerente	
Pre: El gerente está autenticado y Finalizo el proyecto	
Post: El gerente evalúo al PM	
Curso normal	Curso alternativo
1- El sistema avisa al gerente que termino el proyecto	
2- El gerente evalúa al PM	
3- El sistema guarda la evaluación	
4- Fin CU16	

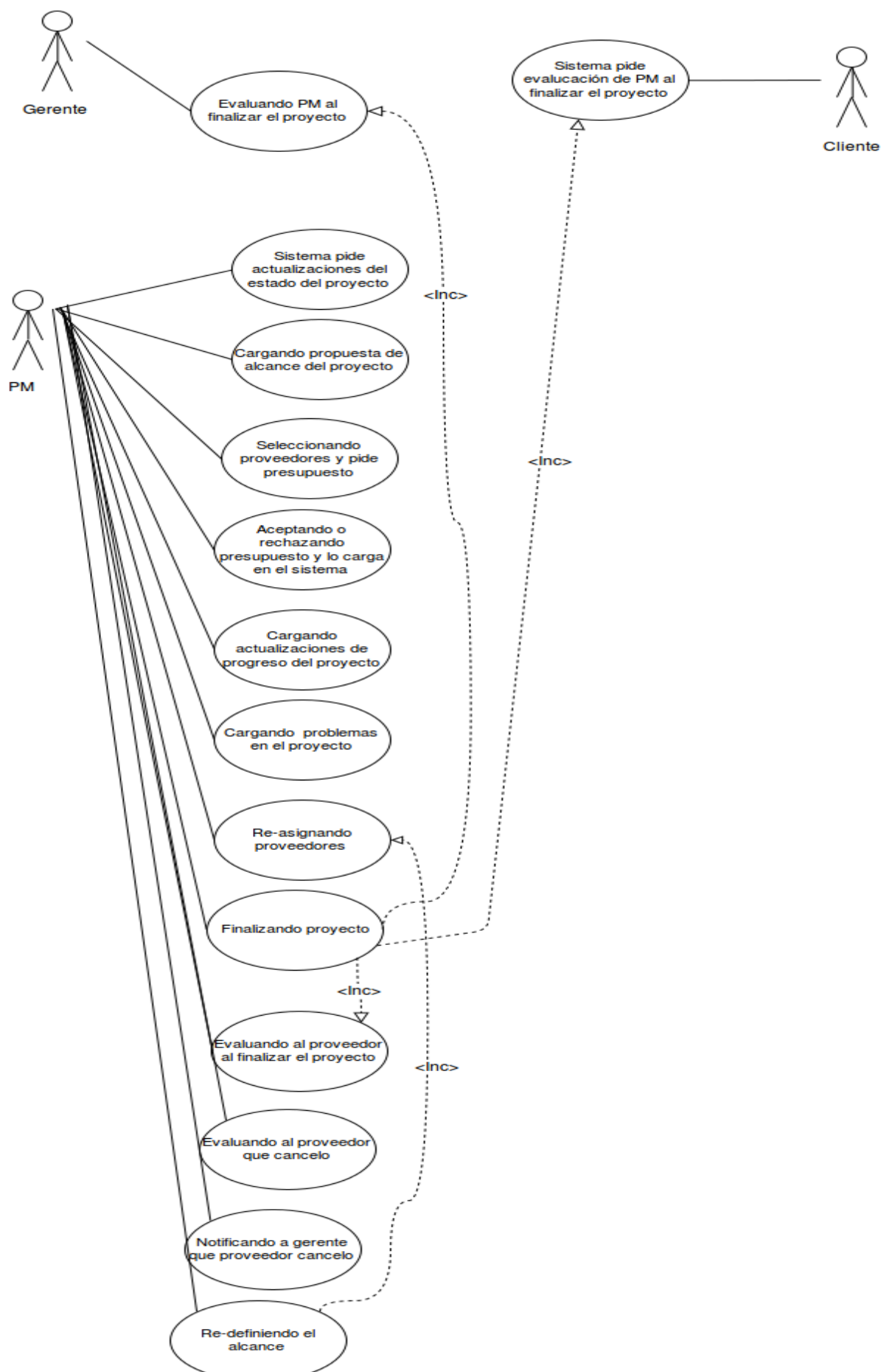
CU17: Agregando Empleado al sistema	
Actor: Gerente	
Pre: El gerente está autenticado	
Post: El empleado queda registrado en el sistema	
Curso normal	Curso alternativo
1- El gerente solicita al sistema que agregue un nuevo empleado	
2- El sistema confirma y pide los datos del empleado	
3- El gerente ingresa los datos del empleado	3.1- El gerente ingresa datos erróneos. Ir a Fin CU17
4- El sistema guarda los datos del empleado	
5- Fin CU17	

CU18: Validando proyecto	
Actor: Gerente	
Pre: El Cliente acepto presupuesto o PM redefinió alcance por proveedor cancelado y el gerente está autenticado	
Post: El proyecto queda validado en el sistema	
Curso normal	Curso alternativo
1- El sistema avisa a gerente que el Cliente aceptó el presupuesto	
2- El gerente revisa el proyecto y lo valida	
3- Fin CU18	

CU19: Enviando presupuesto para el Cliente	
Actor: Gerente	
Pre: El PM aceptó presupuesto de proveedor y lo cargo en el sistema y el gerente está autenticado	
Post: El Cliente tiene el presupuesto del gerente para aceptar o rechazar	
Curso normal	Curso alternativo
1- El sistema avisa a gerente que el PM acepto presupuesto del proveedor	
2- El gerente arma el presupuesto total del proyecto, usando este dato, y lo envía al Cliente	
3- Fin CU19	

CU20: Evaluando a PM apartado de proyecto	
Actor: Gerente	
Pre: El gerente está autenticado	
Post: El PM queda evaluado negativamente y el gerente está autenticado	
Curso normal	Curso alternativo
1- El Gerente evalúa al PM	
2- El sistema guarda la evaluación	
3- Fin CU20	

3.6. En las que actúa el PM



Sección 3.6 En las que actúa el PM

CU21: Sistema pide actualizaciones del estado del proyecto	
Actor: PM	
Pre: El gerente está autenticado y valido el proyecto	
Post: El PM recibe el pedido de actualización del estado del proyecto, por parte del sistema	
Curso normal	Curso alternativo
1- El sistema se actualiza y observa que tiene un proyecto como pendiente de actualización de estado	
2- El sistema pide al PM asignado al proyecto que actualice el estado del mismo.	
3- Fin CU21	

CU22: Cargando propuesta de alcance del proyecto	
Actor: PM	
Pre: El Gerente creó un proyecto y asignó al PM y el PM está autenticado	
Post: La propuesta de alcance para el proyecto queda cargada en el sistema para que el Cliente la pueda ver	
Curso normal	Curso alternativo
1- El PM analiza la propuesta, arma el alcance y lo guarda en el sistema para que sea observable para el Cliente	
2- Fin CU22	

CU23: Seleccionando proveedores y pide presupuesto	
Actor: PM	
Pre: El Cliente aceptó el alcance y el PM está autenticado	
Post: Proveedor queda asignado al proyecto	
Curso normal	Curso alternativo
1- El sistema notifica al PM que el Cliente aceptó el alcance	
2- El PM pide al sistema la lista de proveedores especificando el rubro	
3- El sistema muestra la lista de proveedores, filtrada por rubro y ordenada por ranking de proveedores	
4- El PM elige una cantidad de proveedores de la lista	
5- El sistema manda un mail a todos los proveedores seleccionados con la propuesta de alcance y pidiendo presupuesto	
6- Fin CU23	

CU24: Aceptando o rechazando presupuesto y lo carga en el sistema	
Actor: PM	
Pre: El Proveedor propuso presupuesto y el PM está autenticado	
Post: El PM acepta el presupuesto del proveedor y lo carga en el sistema, o lo rechaza y no lo carga	
Curso normal	Curso alternativo
1- El sistema notifica al PM que el Proveedor propuso un presupuesto	
2- El PM analiza el presupuesto enviado, lo acepta y lo carga en el sistema	2.1- El PM rechaza el presupuesto. Ir a 4
3- El sistema guarda el presupuesto aceptado	
4- Fin CU24	

CU25: Evaluando al proveedor que cancelo	
Actor: PM	
Pre: El gerente valido el proyecto y el PM está autenticado	
Post: El proveedor que cancelo queda evaluado negativamente en el sistema	
Curso normal	Curso alternativo
1- El PM evalúa al proveedor negativamente	
2- El sistema guarda la evaluación	
3- Fin CU25	

CU26: Notificando a gerente que proveedor cancelo	
Actor: PM	
Pre: El PM está autenticado	
Post: El gerente es avisado que el proveedor cancelo	
Curso normal	Curso alternativo
1- El PM carga un aviso de que el proveedor cancelo	
2- El sistema guarda el aviso para que el gerente lo pueda ver	
3- Fin CU26	

CU27: Re-definiendo el alcance	
Actor: PM	
Pre: El PM está autenticado	
Post: El alcance nuevo queda guardado en el sistema	
Curso normal	Curso alternativo
1- El PM analiza el alcance previo del proyecto y lo cambia	
2- El sistema guarda el nuevo alcance para el proyecto	
3- Fin CU27	

CU28: Evaluando al proveedor al finalizar el proyecto	
Actor: PM	
Pre: Finalizo el proyecto y el PM está autenticado	
Post: El PM evalúo al proveedor	
Curso normal	Curso alternativo
1- El PM evalúa al proveedor	
2- El sistema guarda la evaluación	
3- Fin CU28	

CU29: Finalizando proyecto	
Actor: PM	
Pre: El PM está autenticado	
Post: El proyecto queda como finalizado en el sistema y comienza el proceso de evaluaciones	
Curso normal	Curso alternativo
1- El PM marca proyecto como finalizado en el sistema	
2- El sistema guarda esto, avisa a gerente que puede evaluar a PM. USA CU5, USA CU16, USA CU28	
3- Fin CU29	

CU30: Cargando propuesta de alcance del proyecto	
Actor: PM	
Pre: El PM está autenticado y el gerente creó el proyecto y asigno PM	
Post: La propuesta de alcance del proyecto queda guardada en el sistema para que el cliente la vea	
Curso normal	Curso alternativo
1- El sistema notifica al PM que se lo asigno a un proyecto y le manda los detalles de la propuesta	
2- El PM los recibe y formula el alcance para el proyecto. Una vez terminado, lo carga en el sistema	
3- El sistema lo guarda para que el cliente lo pueda ver cuando entre en el sistema	
4- Fin CU30	

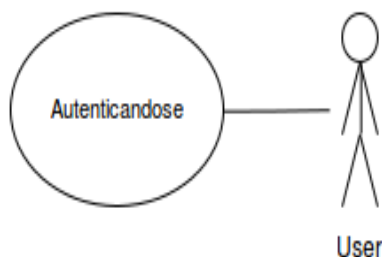
CU31: Cargando actualizaciones de progreso del proyecto	
Actor: PM	
Pre: El PM está autenticado y esta asignado a un proyecto	
Post: Queda cargada en el sistema una actualización del estado del proyecto para que el gerente la vea	
Curso normal	Curso alternativo
1- El PM actualiza el estado del proyecto ingresando los detalles del progreso	
2- El sistema lo guarda para que el gerente lo pueda ver cuando entre en el sistema	
3- Fin CU31	

Sección 3.7 En las que actúa el User

CU32: Cargando problemas en el proyecto	
Actor: PM	
Pre: El PM está autenticado y esta asignado a un proyecto	
Post: Queda cargado en el sistema un aviso de que el proyecto tiene problemas y detalla cuales para que el gerente la vea	
Curso normal	Curso alternativo
1- El PM redacta una actualización en el estado del proyecto informando que esta pasando por problemas	
2- El sistema lo guarda para que el gerente lo pueda ver cuando entre en el sistema	
3- Fin CU32	

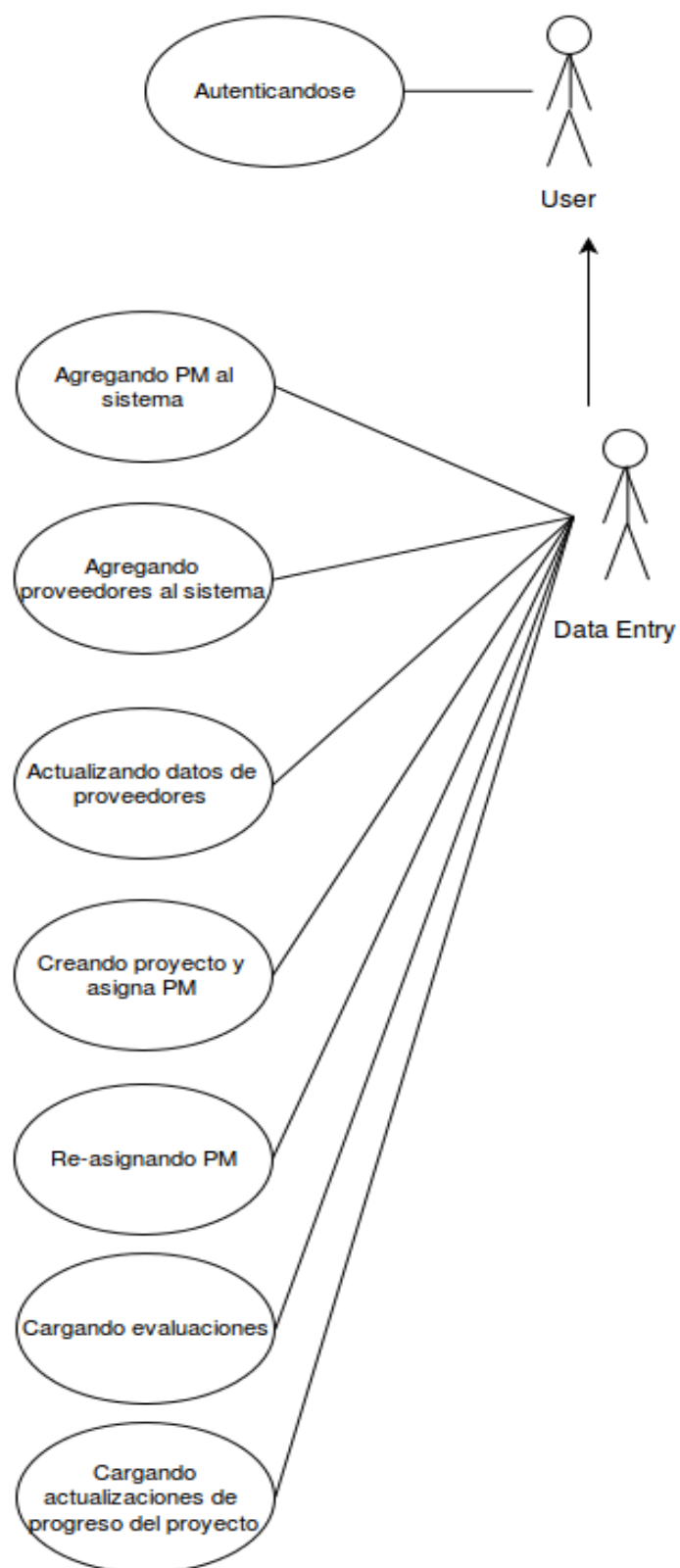
CU33: Re-asignando proveedores	
Actor: PM	
Pre: El PM está autenticado y el PM redefinió el alcance	
Post: Se cambia el proveedor asignado al proyecto por otro	
Curso normal	Curso alternativo
1- El PM pide al sistema la lista de proveedores especificando el rubro	
2- El sistema muestra la lista de proveedores, filtrada por rubro y ordenada por ranking de proveedores	
3- El PM elige una cantidad de proveedores de la lista, los cuales no incluyen al anterior proveedor	
4- El sistema manda un mail a todos los proveedores seleccionados con la propuesta de alcance y pidiendo presupuesto	
5- Fin CU33	

3.7. En las que actúa el User



CU34: Autenticandose	
Actor: User	
Pre: True	
Post: El User queda autenticado en el sistema	
Curso normal	Curso alternativo
1- El User pide al sistema que le permita autenticarse	
2- El sistema confirma y pide al User que ingrese su nombre y contraseña	
3- El User ingresa su nombre y contraseña	3.1- El User ingresa un nombre o contraseña incorrecta, o el User no está registrado en el sistema. Ir a Fin CU34
4- El sistema el sistema confirma y autentica al User	
5- Fin CU34	

3.8. En las que actúa el Data Entry



Sección 3.8 En las que actúa el Data Entry

CU35: Agregando PM al sistema	
Actor: Data Entry	
Pre: El data entry está autenticado	
Post: El PM es agregado al sistema	
Curso normal	Curso alternativo
1- El data entry solicita al sistema ingresar un nuevo PM	
2- El sistema confirma y pide al data entry que ingrese los datos	
3- El data entry ingresa los datos correspondientes al sistema	
4- El sistema valida que los datos sean correctos	
5- El sistema informa que se ingreso el PM al sistema correctamente	5.1-El sistema informa que la información es incorrecta o que los datos corresponden a un PM existente 5.2- Ir a 2
6- Fin CU35	

CU36: Agregando proveedores al sistema	
Actor: Data Entry	
Pre: El data entry está autenticado	
Post: El proveedor es agregado al sistema	
Curso normal	Curso alternativo
1- El data entry solicita al sistema ingresar un nuevo proveedor	
2- El sistema confirma y pide al data entry que ingrese los datos	
3- El data entry ingresa los datos correspondientes al sistema	
4- El sistema valida que los datos sean correctos	
5- El sistema informa que se ingreso al proveedor al sistema correctamente	5.1-El sistema informa que la información es incorrecta o que los datos corresponden a un proveedor existente 5.2- Ir a 2
6- Fin CU36	

CU37: Actualizando datos de proveedores	
Actor: Data Entry	
Pre: El data entry está autenticado	
Post: La información de los proveedores se actualiza en el sistema	
Curso normal	Curso alternativo
1- El data entry solicita al sistema que actualice los datos de proveedores	
2- El sistema confirma y muestra el listado de los proveedores pidiéndole al data entry que seleccione los que quiera modificar	
3- El data entry selecciona uno o mas proveedores	
4- El sistema muestra los datos actuales de un proveedor seleccionado	
5- El data entry modifica los datos del proveedor	
6- El sistema guarda los nuevos datos del proveedor	
7- Si quedan proveedores por modificar Ir a 4 sino Fin CU37	

CU38: Creando proyecto y asignar PM	
Actor: Data Entry	
Pre: El data entry está autenticado y el cliente envió una propuesta	
Post: Se creo un nuevo proyecto en el sistema con un PM asignado	
Curso normal	Curso alternativo
1- El data entry solicita al sistema que cree un nuevo proyecto	
2- El sistema confirma y pide al data entry que ingrese los datos	
3- El data entry crea un nuevo proyecto con los detalles de la propuesta	
4- El sistema muestra la lista de PM's y le pide al data entry elegir uno	
5- El data entry selecciona un PM	
6- El sistema informa que se creo el proyecto con el PM asignado exitosamente	
7- Fin CU38	

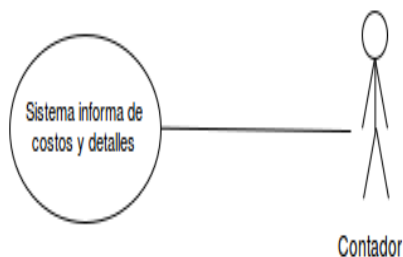
Sección 3.9 En las que actúa el Contador

CU39: Re-assignando PM	
Actor: Data Entry	
Pre: Hay PM asignado a proyecto y el Data Entry está autenticado	
Post: Se modifico el PM asignado al proyecto	
Curso normal	Curso alternativo
1- El data entry solicita al sistema reasignar un PM	
2- El sistema confirma, muestra una lista de los proyectos en curso y le pide seleccionar el proyecto en el cual quiere reasignar al PM	
3- El data entry selecciona un proyecto	
4- El sistema muestra información del proyecto, al PM asignado y la lista de PM's (sin el asignado actualmente), le pide al data entry seleccionar un nuevo PM	
5- El data entry selecciona un PM	
6- El sistema informa que se reasigno correctamente	
7- Ir a CU20	
8- Fin CU39	

CU40: Cargando evaluaciones	
Actor: Data Entry	
Pre: El Data Entry está autenticado	
Post: Se cargaron las evaluaciones en el sistema	
Curso normal	Curso alternativo
1- El data entry solicita al sistema cargar evaluaciones	
2- El sistema confirma, y le pide al data entry que elija entre PM y proveedor	
3- El data entry selecciona uno de los actores	
4- El sistema muestra la encuesta del actor seleccionado y pide al data entry completarla	
5- El data entry completa la encuesta y pide al sistema que la guarde	
6- El sistema guarda la evaluación y notifica que se guardo correctamente	
7- El sistema le pregunta al data entry si desea cargar mas evaluaciones	
8- Si el data entry quiere completar mas evaluaciones Ir a 2 sino Fin CU40	

CU41: Cargando actualizaciones de progreso del proyecto	
Actor: Data Entry	
Pre: El Data Entry está autenticado	
Post: Queda cargada en el sistema una actualización del estado del proyecto para que el gerente la vea	
Curso normal	Curso alternativo
1- El data entry solicita al sistema que actualice el estado de un proyecto	
2- El sistema confirma y pide al data entry que ingrese el proyecto y los detalles del progreso	
3- El data entry ingresa el proyecto y actualiza el estado ingresando los detalles del progreso	
4- El sistema lo guarda para que el gerente lo pueda ver cuando entre en el sistema	
5- Fin CU41	

3.9. En las que actúa el Contador



Sección 3.9 En las que actúa el Contador

CU42: Sistema informa de costos y detalles	
Actor: Contador	
Pre: El Contador está autenticado	
Post: El sistema le brinda los costos y detalles al contador	
Curso normal	Curso alternativo
1- El contador solicita al sistema que le de la información de los costos y detalles de una semana/mes/año	
2- El sistema confirma y le proporciona al contador la información solicitada	
3- Fin CU42	

4. Diagrama de Actividad

4.1. Descripción general

Los diagramas de actividad presentan varias características que resultan ventajosas a la hora de especificar el comportamiento del sistema a desarrollar. Principalmente la capacidad de los diagramas de actividad de especificar una secuencia de eventos resulta útil para describir los procesos que ocurren en el ciclo de vida de los proyectos. Otra característica relevante de esta herramienta es su capacidad de poder especificar procesos que suceden simultáneamente, es decir de forma paralela.

En particular, hemos confeccionado cuatro diagramas de actividad los cuales captan el ciclo de vida de los proyectos y las situaciones extraordinarias que suceden en la etapa de seguimiento, utilizando escenarios concretos. Detalladamente:

- El primer diagrama de actividad confeccionado especifica un escenario completo del ciclo de vida de un proyecto con la particularidad de que suponemos que tanto el cliente como los proveedores utilizan el sistema, que, tanto el alcance como el presupuesto, son aceptados en el primer intento y que no hay inconvenientes en la etapa de seguimiento de los proyectos, es decir: el proveedor no cancela, el PM no es reemplazado y no cancela, y el PM carga las actualizaciones periódicas. Aclaramos que en este escenario el PM pide presupuestos a solo dos proveedores (llamados proveedor 1 y 2). Tomamos esta decisión porque el comportamiento del sistema no difiere si los pedidos de presupuesto son a 2 o a N proveedores, mientras que con 2 proveedores ganamos claridad. Con respecto al resto de las características particulares de este escenario, consideramos que para lograr captar al máximo el comportamiento del sistema y a su vez lograr que el diagrama sea legible, asumir que el cliente y el proveedor utilizan siempre el sistema es lógico, lo mismo sucede con la definición del alcance y de presupuesto (proceso que se encuentra especificado en su totalidad en otra herramienta). Con respecto al seguimiento, consideramos que especificar todas las situaciones posibles en esta etapa en un diagrama resulta imposible por lo cual especificamos la situación ideal. El resto de los situaciones serán especificados en otros diagramas.
- El segundo diagrama de actividad especifica un escenario en donde el proveedor contratado (llamado proveedor 1 en el diagrama) cancela y se debe resolver esta situación extraordinaria. El mismo finaliza cuando un nuevo proveedor es contratado para llevar adelante la obra. Nuevamente limitamos la cantidad de proveedores a los que el sistema les pide presupuesto por las razones expuestas en el primer inciso. Este escenario representa una de las situaciones no especificadas en el primer diagrama, y, dado su importancia, consideramos necesario mostrarlo con un diagrama propio.
- El tercer diagrama de actividad especifica un escenario en donde el PM asignado es reemplazado debido a la inconformidad del cliente con respecto a su trabajo. El mismo finaliza con el reemplazo del PM por otro. Este escenario representa una de las situaciones no especificadas en el primer diagrama, y, dado su importancia, consideramos necesario mostrarlo con un diagrama propio.
- Por ultimo el cuarto diagrama de actividad especifica un escenario en donde el PM asignado cancela por decisión propia. El mismo finaliza con el cambio del PM. Este escenario representa una de las situaciones no especificadas en el primer diagrama, y, dado su importancia, consideramos necesario mostrarlo con un diagrama propio.

4.2. Relación con otros modelos

Como ya hemos mencionado, los casos de uso captan las interacciones entre el sistema y los agentes, sin embargo son incapaces de darle una secuencia a los distintos procesos, hecho que se logra con los diagramas de actividad presentados.

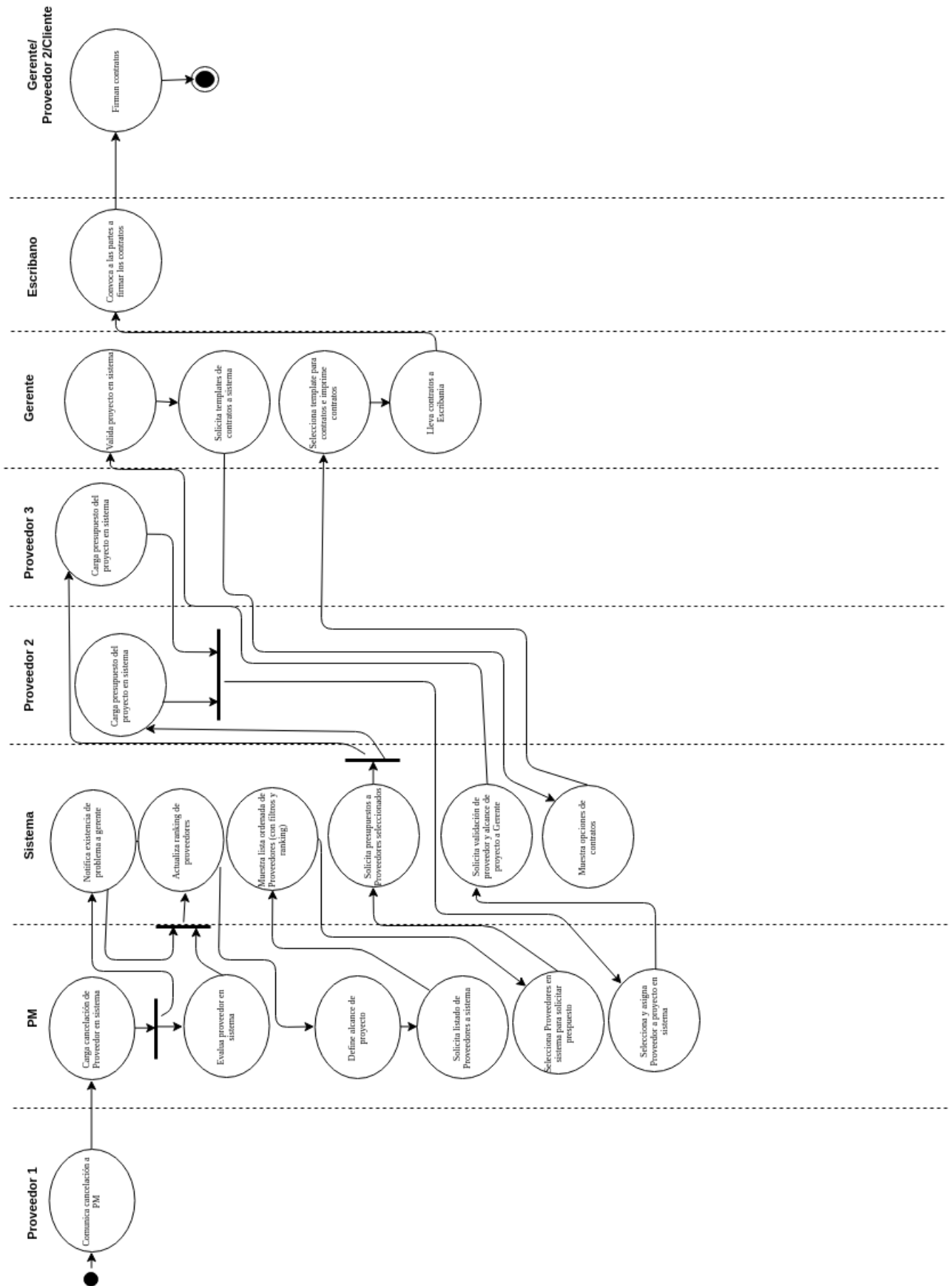
Por otro lado, los diagramas de actividad confeccionados se complementan con las Máquinas de Estados Finitos que serán presentados a continuación, ya que esta herramienta será utilizada para captar el proceso de actualizaciones en la etapa de seguimiento de proyecto en su totalidad, en contraposición con la mera presentación del mismo en los diagramas de actividad. También utilizaremos las Máquinas de

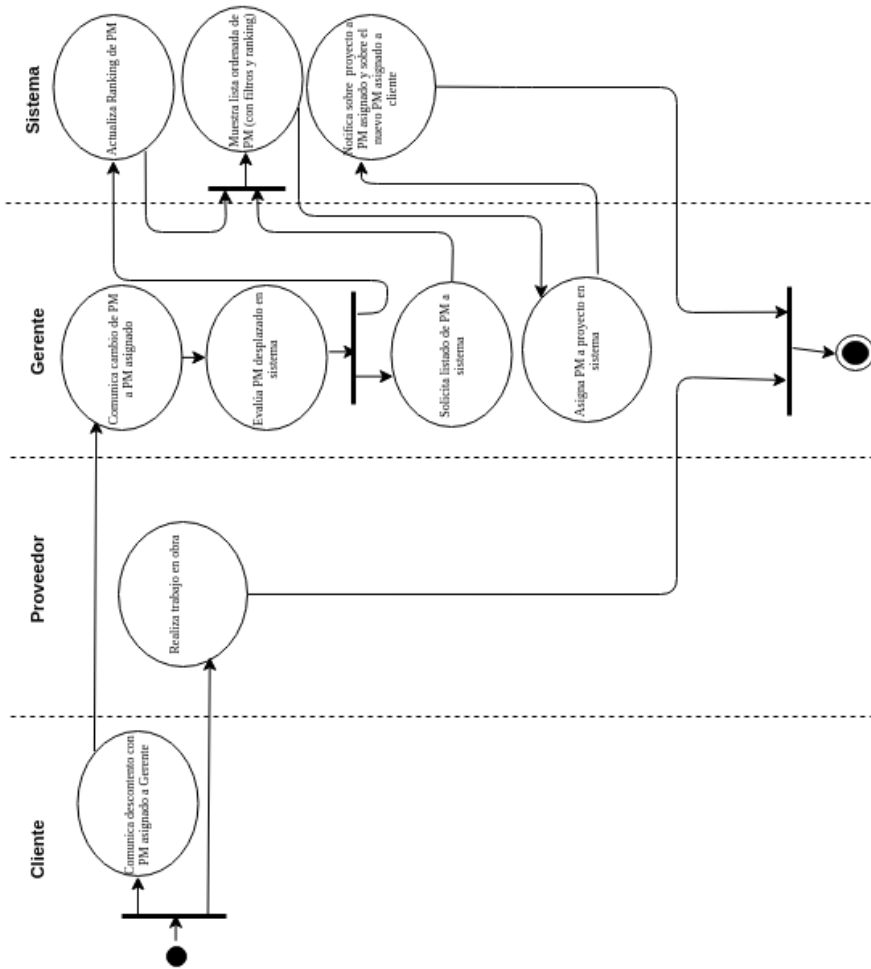
Estados Finitos para especificar los idas y vueltas de la negociación de alcance y presupuesto, cuestiones que no se especifican con la presente herramienta.

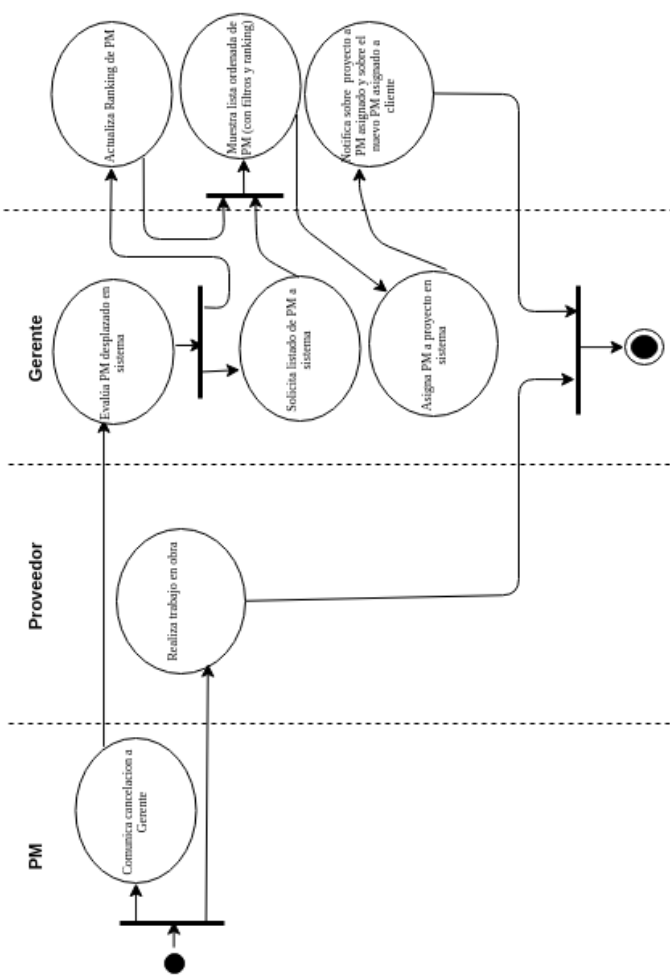
4.3. Vistas

A continuación mostramos los diagramas de actividad antes descritos, en el orden mencionado bajo el título de descripción general.









5. Maquinas de Estado Finito

5.1. Descripción general

El diagrama de Máquinas de Estados Finitos tiene como objetivo principal mostrar el detalle de ciertos procesos dinámicos que no se pueden apreciar correctamente en el Diagrama de Actividad. Los procesos que elegimos fueron:

- El funcionamiento de las actualizaciones durante la etapa de seguimiento de un proyecto.
- El proceso mediante el cual el Cliente y el Gerente acuerdan el presupuesto a través del Sistema.
- El proceso mediante el cual el Cliente y el PM acuerdan el alcance del proyecto a través del Sistema.

5.2. Relación con otros modelos

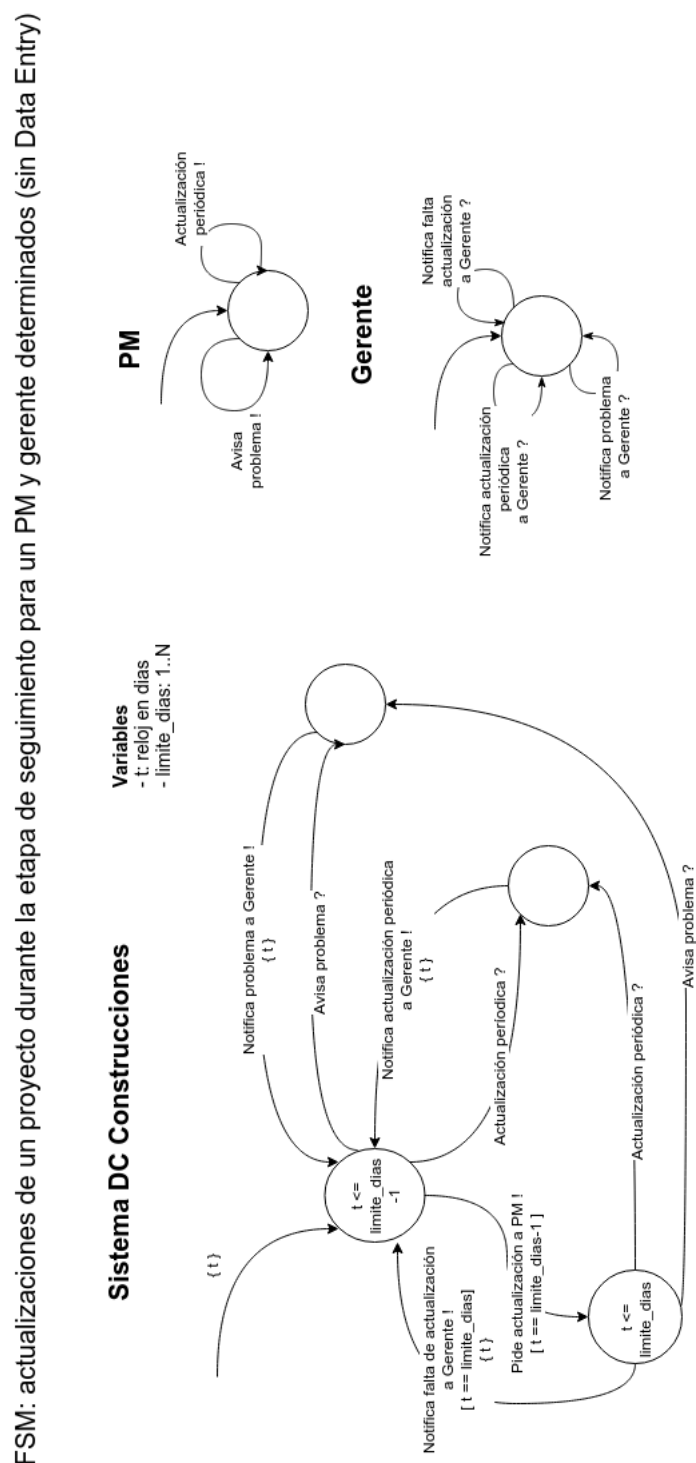
Los 3 procesos mencionados aparecen de alguna forma en el DA del ciclo de vida de un proyecto:

- Las actualizaciones de un proyecto en el seguimiento aparecen *encapsuladas* en un nodo, por lo que podríamos pensar que al modelar este escenario con FSM, estamos haciendo un *zoom* dentro de ese nodo en el DA.
- Los otros dos procesos aparecen como una secuencia de varios nodos, pero para no complicar demasiado el DA se omitieron los posibles ciclos. Por esta razón, modelarlos con FSM permite entender la dinámica real de dichos procesos.

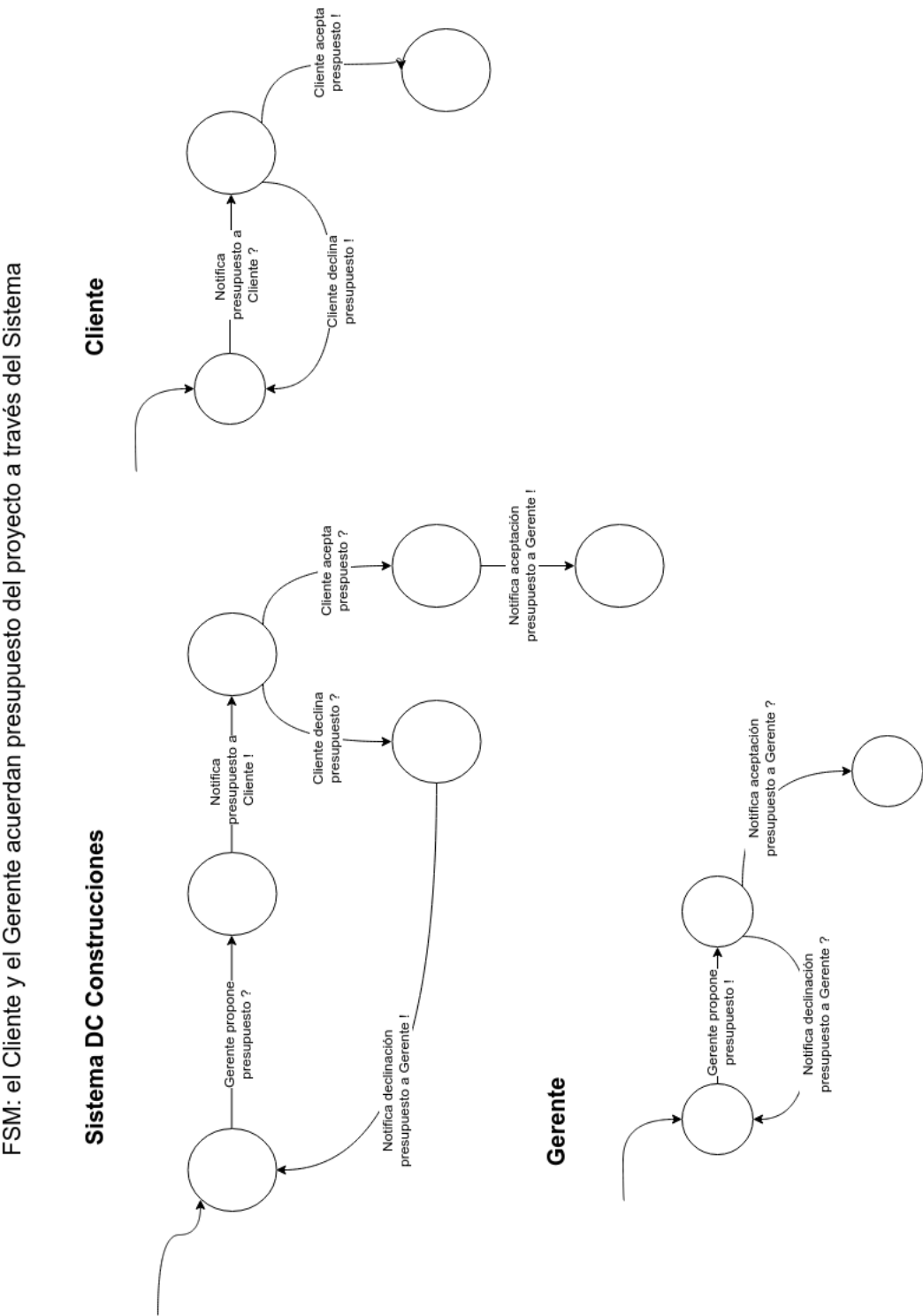
5.3. Vistas

El primer diagrama que mostraremos es el funcionamiento de las actualizaciones durante la etapa de seguimiento de un proyecto. Para simplificar, no consideramos en el escenario la posibilidad de que cambie el PM del proyecto, o cancele el proveedor, ya que son dos posibilidades que están explicadas con detalle en diversos Diagramas de Actividad.

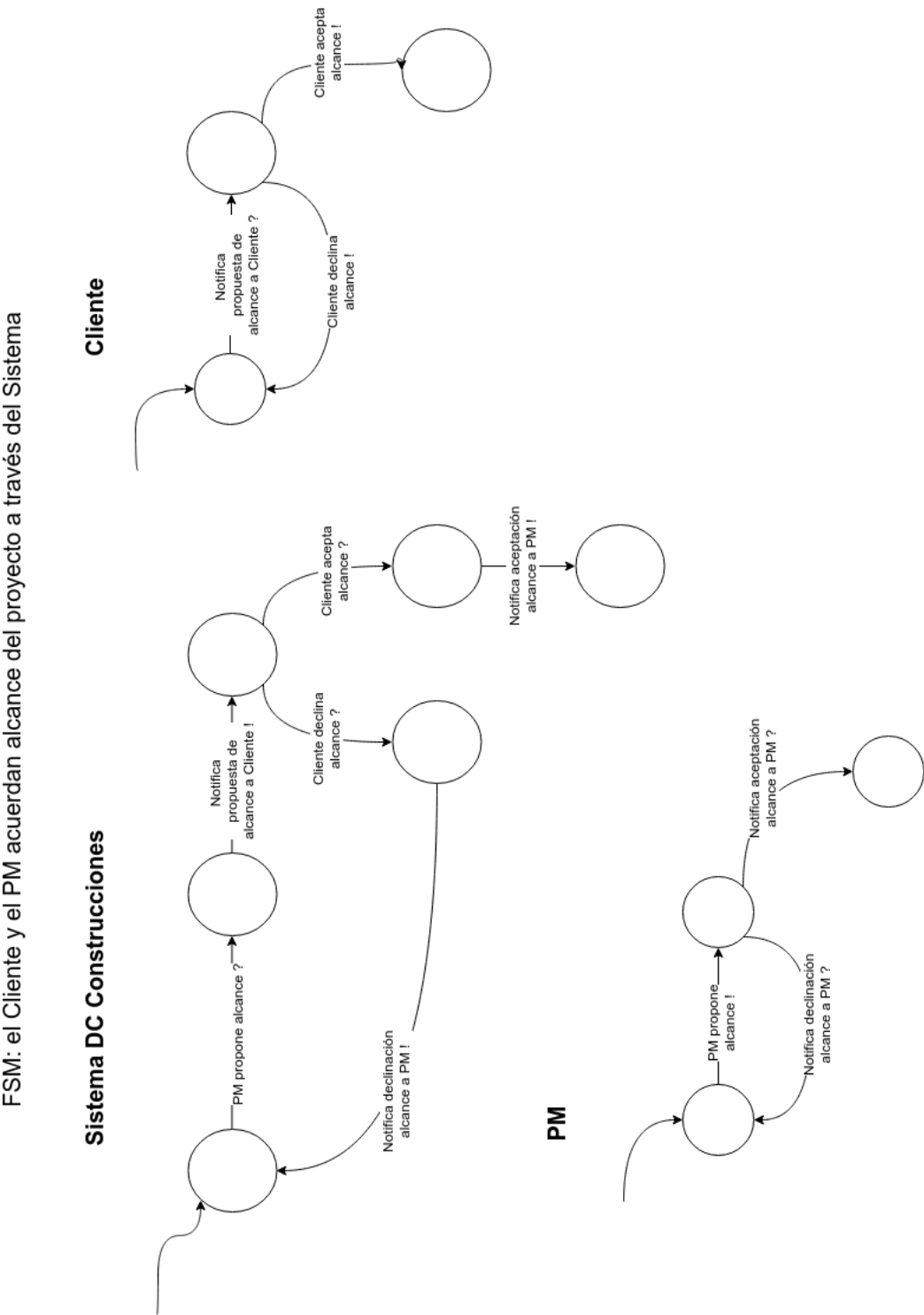
La variable *limite_dias* mencionada en la FSM, responde al intervalo de actualizaciones durante el cual debe haber al menos una actualización. Este limite es particular de cada proyecto y lo configura el Gerente al momento de la creación del mismo.



El segundo diagrama que mostraremos es el detalle del proceso mediante el cual el Cliente y el Gerente acuerdan el presupuesto a través del Sistema.



El tercer diagrama que mostraremos es el detalle del proceso mediante el cual el Cliente y el PM acuerdan el alcance del proyecto a través del Sistema.



6. Modelo Conceptual

6.1. Descripción general

Este modelo especifica de forma estática los diferentes estados válidos de un proyecto según el sistema. Con eso en mente, nuestro modelo realizado contempla:

- Todo los agentes relacionados con el proyecto desde el punto de vista del Sistema
- Los presupuestos clientes y proveedores y como se relacionan con los contratos que luego se firman
- Comisiones de los PMs
- Los rankings de PMs y Proveedores
- Seguros de caución de los Proveedores y su validez
- Representa estados que muestran cuando hay cambios de PMs y Proveedores
- Actualizaciones de Proyecto
- Notificaciones periódicas y por problemas
- Evaluaciones de proyecto cuando el mismo finaliza
- Los estados y propiedades de un proyecto y todos sus agentes relacionados según el Sistema

Nuestro modelo conceptual tiene como principal clase el Proyecto, que es el eje de todo el Sistema y el agente con mas relaciones con los demás. A su vez, se modelan los presupuestos y contratos con los proveedores y clientes. Determinamos que cada presupuesto de un Proveedor es para un proyecto diferente pero que un Cliente puede tener varios presupuestos propuestos para un mismo proyecto. En caso de que esos presupuestos sean aprobados, se crea un Contrato para ese presupuesto (cosa que marcamos con una relación 0..1 contratoCliente/contratoProveedor).

Cada proyecto en sí tiene un Gerente que lo supervisa y un PM actual (pueden haber varios PMs relacionados mediante la clase Comisión, pero solo uno es el actual, los demás están porque fueron PMs y en algún momento fueron cancelados). Este PM puede subir actualizaciones de Proyecto y por cada una se genera una actualización.

También modelamos los rankings de Proveedores y PMs, evaluaciones al terminar el proyecto y notificaciones según el agente que la recibe (es decir, si hay una notificación para, por ejemplo, un Gerente y un PM, se generan dos notificaciones distintas en este modelo, una para el Gerente y otra para el PM). El caso de los Contadores decidimos modelarlo relacionándolo al Gerente y no al Proyecto porque el Contador ayuda al Gerente y las cobranzas y pagos lo realiza todo externamente al Sistema.

El caso en que un nuevo PM es asignado para un proyecto, se marca la Comisión del viejo PM como cancelada (mediante una propiedad de estado) y se agrega una Comisión a otro PM. En cambio, cuando hay un cambio de Proveedor, el Contrato de ese Proveedor es el que se marca como cancelado (también mediante una propiedad de estado).

Consideramos que tanto los contratos de los Proveedores pueden ser cancelados como los de los Clientes. En el primer caso porque un Proveedor puede ser removido de la obra y cambiado por otros. En el segundo porque al renegociar con nuevos Proveedores (cuando estos cambian), también hay que firmar un nuevo contrato con el cliente.

Notar que los Data Entries no los modelamos pues no eran relevantes para el Proyecto en sí.

6.2. Relación con otros modelos

Ahora bien, como explicábamos antes, el modelo conceptual modela los estados, propiedades y relaciones estáticas entre los agentes de un proyecto según el Sistema. Por esto mismo, hay cosas que los demás modelos especifican de forma mas precisa:

Las etapas previas a la creación de un proyecto, el acuerdo de alcance y duración del proyecto con el PM se ven mucho mejor en los diagramas de FSM y Actividad, de hecho, el alcance lo modelamos

asumiendo que guardar todos los alcances acordados y re-acordados del Proyecto para el Modelo Conceptual es irrelevante y solo nos interesa el alcance actual, es decir el último acordado. La autenticación y los procedimientos de un Usuario en el Sistema y cómo se usa el Sistema en si, está explicado más claramente en el Diagrama de Casos de Uso.

Las notificaciones, como esto afecta al sistema y qué eventos las causan se pueden ver explícitamente en los Diagramas de FSM.

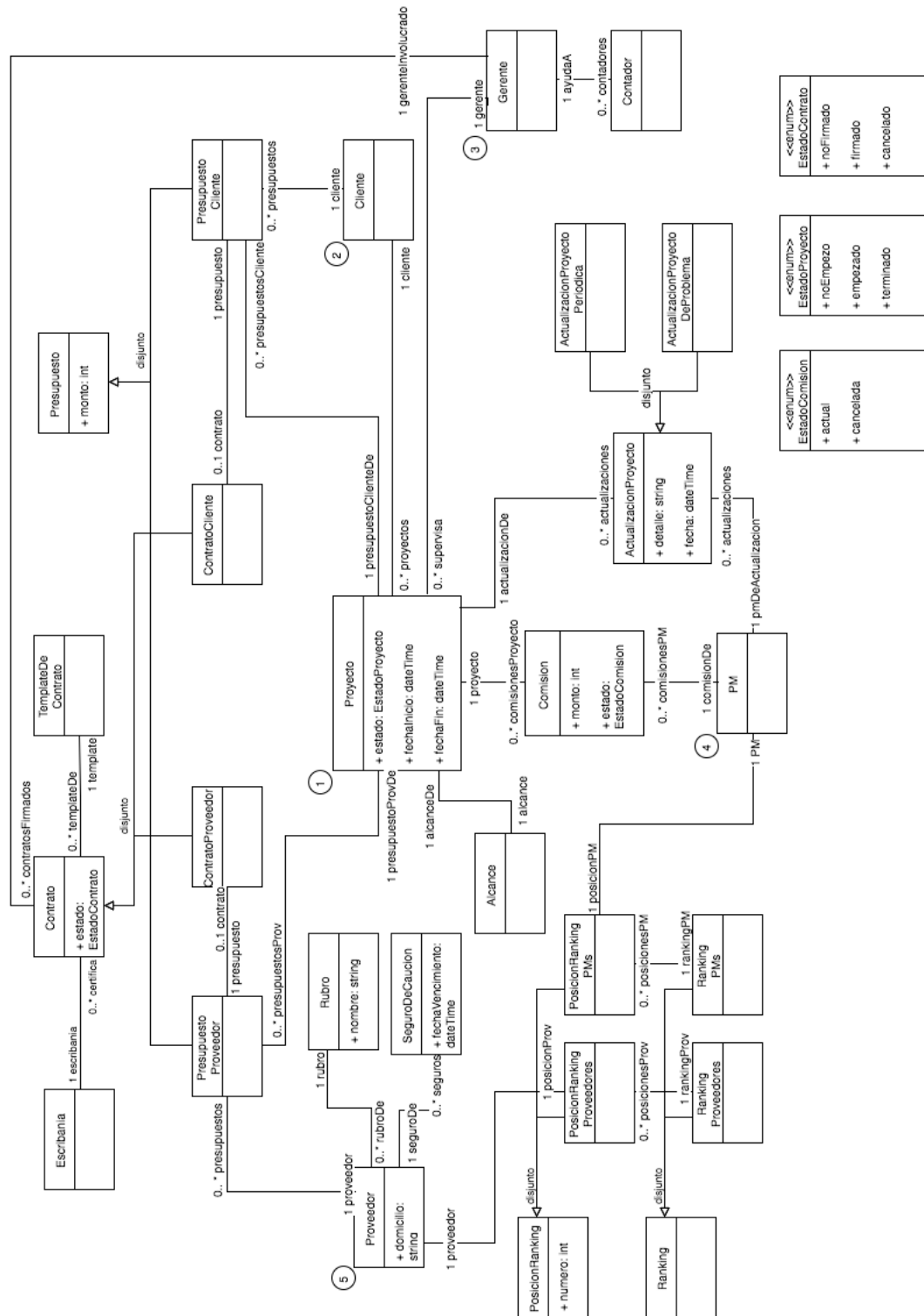
Los Data Entries también son modelados en los demás Diagramas.

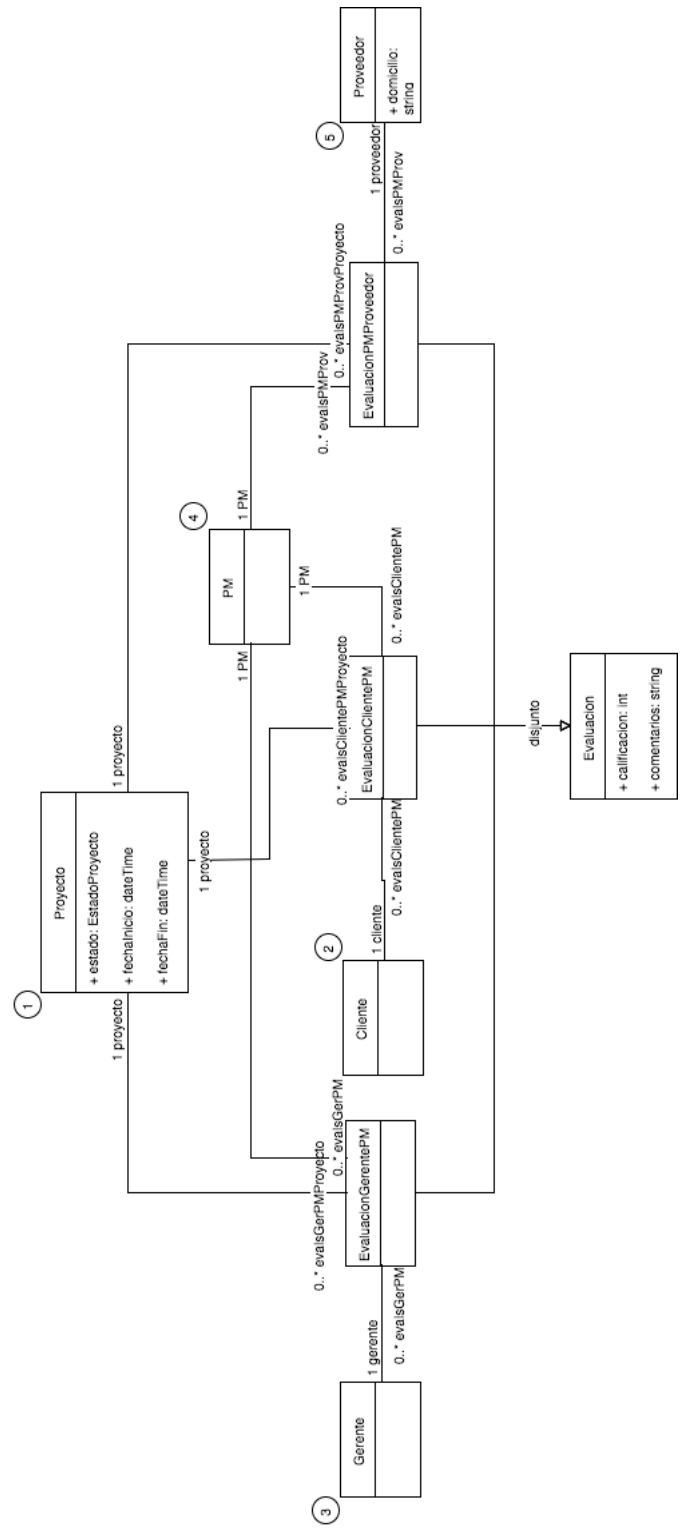
La cronología en general de todos los eventos del proyecto se explica mejor en el Diagrama de Actividad.

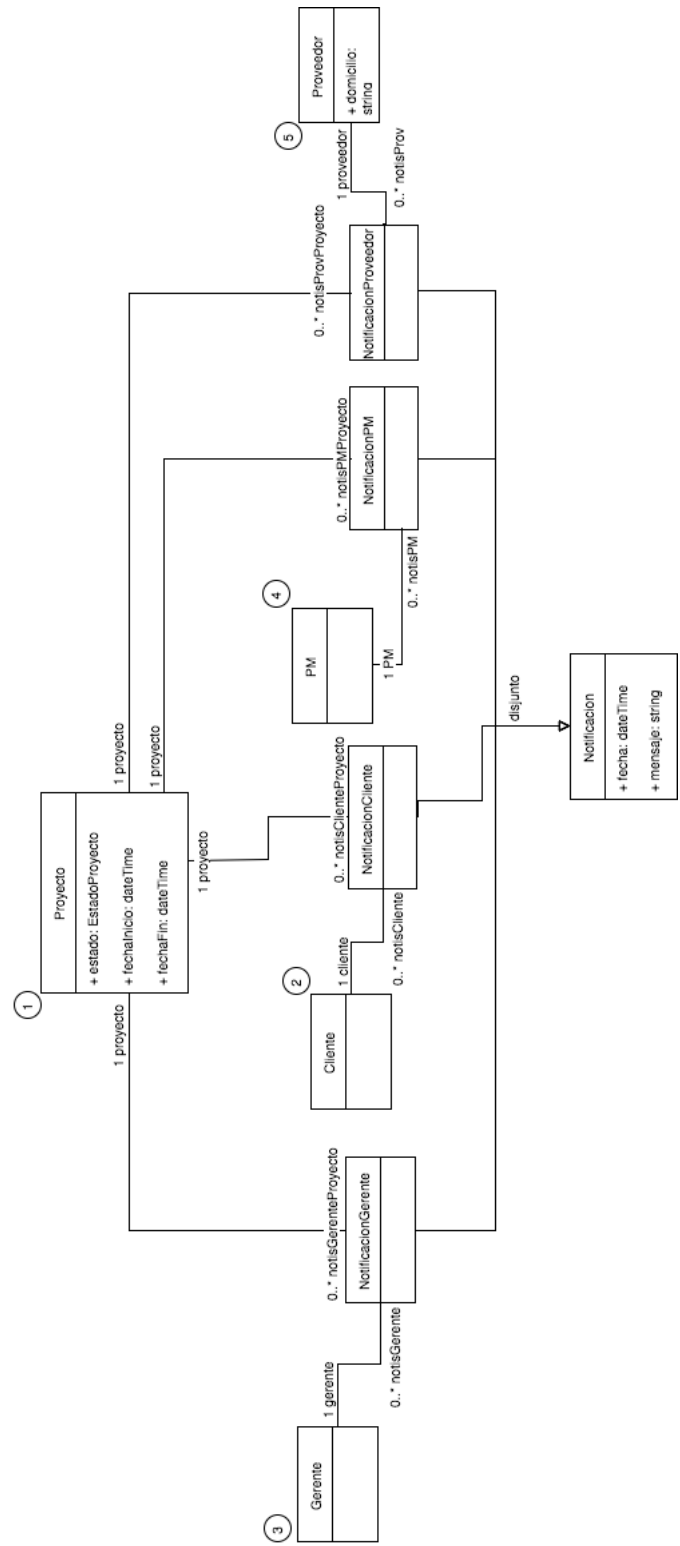
6.3. Observaciones de las Vistas

Como se puede ver, hay tres vistas para el Diagrama de Clases, pero estas tres vistas forman en realidad parte de un solo diagrama. Las separamos por motivos de claridad pero tienen que ser leídas e interpretadas como parte de un mismo Diagrama de Clases. Por este motivo, marcamos con números las clases que se repiten en las vistas para decir que en realidad son las mismas clases y que todo es parte de un solo y mismo diagrama.

6.4. Vistas







6.5. OCL

Nuestro modelo tendrá las siguientes restricciones:

1. Los rankings de Proveedor están bien formados (tienen del número 1 a cantidad(Proveedores)).

```
Context: Proveedor
Inv: Proveedor.allInstances()->isUnique(posicionProv.numero) and Proveedor.
    allInstances()->exists(p | p.posicionProv.numero = 1) and Proveedor.
    allInstances()->exists(p | p.posicionProv.numero = Proveedor.
    allInstances()->size())
```

2. Los rankings de PMs están bien formados (tienen del número 1 a cantidad(PMs)).

```
Context: PM
Inv: PM.allInstances()->isUnique(posicionPM.numero) and PM.allInstances()->
    exists(p | p.posicionPM.numero = 1) and PM.allInstances()->exists(p | p.
    posicionPM.numero = PM.allInstances()->size())
```

3. Cada proveedor tiene un seguro de caución que vence después de la finalización de cada proyecto al que le mandó un presupuesto.

```
Context: Proveedor
Inv: self.presupuestos->collect(presupuestoProvDe)->forAll(proy | self.
    seguros->exists(seg | seg.fechaVencimiento >= proy.fechaFin))
```

4. Los contratos de Proveedores están asociados a un gerente que supervisa el proyecto del contrato.

```
Context: ContratoProveedor
Inv: self.gerenteInvolucrado.supervisa->includes(self.presupuesto.
    presupuestoProvDe)
```

5. Los contratos de Clientes están asociados a un gerente que supervisa el proyecto del contrato.

```
Context: ContratoCliente
Inv: self.gerenteInvolucrado.supervisa->includes(self.presupuesto.
    presupuestoClienteDe)
```

6. Cada proveedor manda como máximo un presupuesto por proyecto.

```
Context: Proveedor
Inv: self.presupuestos->isUnique(presupuestoProvDe)
```

7. Hay como máximo un presupuesto del cliente con contrato firmado (y no cancelado) para cada proyecto.

```
Context: Proyecto
Inv: self.presupuestosCliente->select(pres | pres.contrato->notEmpty() and
    pres.contrato.estado = firmado)->size() <= 1
```

8. Si el proyecto no empezó, no hay contrato de clientes firmados, si empezó hay exactamente un contrato de cliente firmado y al menos un contrato de Proveedor firmado.

```
Context: Proyecto
```

```
Inv: self.estado = noEmpezo implies self.presupuestosCliente->select(pres |
    pres.contrato->notEmpty() and pres.contrato.estado = firmado)->size() =
    0 and self.estado <> noEmpezo implies (self.presupuestosCliente->select
    (pres | pres.contrato->notEmpty() and pres.contrato.estado = firmado)->
    size() = 1 and self.presupuestosProv->select(pres | pres.contrato->
    notEmpty() and pres.contrato.estado = firmado)->size() > 0)
```

9. La suma de los presupuestos de los Proveedores que firmaron un contrato es menor al presupuesto que se le manda al cliente.

Context: Proyecto

```
Inv: self.presupuestosCliente->select(presCliente | presCliente.contrato->
    notEmpty() and presCliente.contrato.estado = firmado)->forall(
    presCliente | presCliente.monto < self.presupuestosProv->select(presProv
    | presProv.contrato->notEmpty() and presProv.contrato.estado = firmado)
    ->collect(monto)->sum())
```

10. El cliente de un proyecto coincide con el cliente asociado a cada presupuesto del proyecto.

Context: Proyecto

```
Inv: self.presupuestosCliente->forall(pres | pres.cliente = self.cliente)
```

11. Hay como máximo un PM actual por proyecto (puede ser momentáneamente ninguno cuando se cambia de PM).

Context: Proyecto

```
Inv: self.comisionesProyecto->select(c | c.estado = actual) <= 1
```

12. Cada comisión de un PM es para un proyecto distinto.

Context: PM

```
Inv: self.comisionesPM->isUnique(proyecto)
```

13. Todas las actualizaciones de Proyecto pertenecen a un PM que es o fue el PM asignado del proyecto de la actualización.

Context: PM

```
Inv: self.actualizaciones->forall(a | self.comisionesPM->exists(c | c.
    proyecto = a.actualizacionDe))
```

14. Las evaluaciones de Gerentes a PMs están en un ciclo válido (es decir, el PM y el Gerente estuvieron asignados al proyecto de la evaluación).

Context: EvaluacionGerentePM

```
Inv: self.PM.comisionesPM->collect(proyecto)->includes(self.proyecto) and
    self.gerente.supervisa->includes(self.proyecto)
```

15. Las evaluaciones de Clientes a PMs están en un ciclo válido (es decir, el Cliente y el Gerente estuvieron asignados al proyecto de la evaluación).

Context: EvaluacionClientePM

```
Inv: self.PM.comisionesPM->collect(proyecto)->includes(self.proyecto) and
    self.cliente = self.proyecto.cliente
```


16. Las evaluaciones de Gerentes a PMs están en un ciclo válido (es decir, el PM y el Gerente estuvieron asignados al proyecto de la evaluación).

```
Context: EvaluacionPMProveedor
Inv: self.PM.comisionesPM->collect(proyecto)->includes(self.proyecto) and
    self.proyecto.presupuestosProv->select(p | p.contrato->notEmpty())->
    includes(self.proveedor)
```

17. Por cada PM cancelado hay una evaluación del Cliente al PM y otra del Gerente al PM.

```
Context: Comisión
Inv: self.estado = cancelada implies (self.proyecto.evalsGerPMProyecto->
    exists(e | e.PM = self.comisionDe) and self.proyecto.
    evalsClientePMProyecto->exists(e | e.PM = self.comisionDe))
```

18. Por cada Proveedor cancelado hay una evaluación del PM a ese Proveedor.

```
Context: ContratoProveedor
Inv: self.estado = cancelado implies self.presupuesto.presupuestoProvDe.
    evalsPMPProvProyecto->exists(e | e.proveedor = self.presupuesto.proveedor
    )
```

19. Si el proyecto terminó, se evaluaron a todos los PMs y Proveedores involucrados.

```
Context: Proyecto
Inv: self.estado = terminado implies (self.comisionesProyecto->collect(
    comisionDe)->forAll(pm | self.evalsGerPMProyecto->includes(e | e.PM = pm
    ) and self.evalsClientePMProyecto->exists(e | e.PM = pm)) and self.
    presupuestosProv->select(pres | pres.contrato->notEmpty())->collect(
    proveedor)->forAll(prov | self.evalsPMPProvProyecto->exists(e | e.
    proveedor = prov)))
```

20. Fechas correctas: la fecha de inicio de un proyecto es menor que la de finalización, las fechas de las actualizaciones y notificaciones son mayores o iguales a la fecha de inicio del proyecto

```
Context: Proyecto
Inv: self.fechaFin < self.fechaInicio and self.actualizaciones->forAll(a |
    a.fecha >= self.fechaInicio) and self.notisGerenteProyecto->forAll(n | n.
    fecha >= self.fechaInicio) and self.notisClienteProyecto->forAll(n | n.
    fecha >= self.fechaInicio) and self.notisPMPProyecto->forAll(n | n.fecha
    >= self.fechaInicio) and self.notisProvProyecto->forAll(n | n.fecha >=
    self.fechaInicio)
```

21. Las notificaciones a Gerentes son al asignado del proyecto.

```
Context: NotificacionGerente
Inv: self.gerente = self.proyecto.gerente
```

22. Las notificaciones a Clientes son al del proyecto.

```
Context: NotificacionCliente
Inv: self.cliente = self.proyecto.cliente
```

23. Las notificaciones a PMs son a alguno asignado en algún momento al proyecto.

```
Context: NotificacionPM
Inv: self.proyecto.comisionesProyecto->collect(comisionDe)->exists(pm | pm
    = self.PM)
```

24. Las notificaciones a Proveedores son a alguno asignado alguna vez al proyecto.

```
Context: NotificacionProveedor
Inv: self.proyecto.presupuestosProv->select(p | p.contrato->notEmpty())->
    collect(proveedor)->exists(prov | prov = self.proveedor)
```

Aclaraciones: según nuestro modelo, no es necesario restringir con OCL que ciertos eventos causan un envío de notificaciones porque es un comportamiento deseado pero no un comportamiento necesario para que la instancia del sistema sea válida. Mediante los otros Diagramas (como el de Actividad) especificamos que ciertos eventos como la creación de un proyecto causa un envío de notificación al Gerente, esto es algo deseado pero no necesario de restringir con OCL.

7. Conclusiones

El presente Trabajo Práctico se propuso, a partir de los objetivos y requerimientos realizados en el trabajo anterior, realizar una serie de modelos que permitan:

- Entender en detalle cuales son las interacciones que tendrá el Sistema con el resto de los actores.
- Entender cuales son los conceptos que se tienen en cuenta dentro del Sistema, y cuales son las relaciones entre ellos.
- Entender cual es el ciclo de vida de un proyecto cuando los diferentes actores utilizan el Sistema.

Los cuales creemos que se pudieron realizar de manera adecuada. Encontramos que cada modelo tiene ventajas para ciertos escenarios:

- El Diagrama de Casos de Uso (junto con el detalle de cada uno de ellos) es esencial para entender cual es la interfaz del Sistema, y ver cual es el detalle fino de las interacciones definidas en el Diagrama de Contexto.
- El Diagrama de Actividad es muy útil para entender las interacciones a lo largo del tiempo de varios actores en un proceso determinado. En el caso de este trabajo, se utilizó para mostrar el ciclo de vida completo de un proyecto, y como influye cada actor en las diferentes etapas de este.
- El Diagrama de Maquinas de Estado permite dar el detalle de procesos que son muy dinámicos y no son sencillos de expresar con el Diagrama de Actividad. En el caso de este trabajo, se utilizó para mostrar los procesos de: actualizaciones del PM en el seguimiento de un proyecto, y la negociación a través del Sistema del presupuesto (entre Gerente y Cliente) y el alcance del proyecto (entre PM y Cliente).
- El Modelo Conceptual muestra cuales son los conceptos a tener en cuenta en un problema, y como se relacionan entre ellos. En el caso de este trabajo, se utilizó para mostrar las relaciones que tienen los distintos actores alrededor del concepto “Proyecto”, ya que es un concepto central del problema y merece ese nivel de detalle.

En cuanto a la trazabilidad entre los distintos diagramas planteados, vimos que muchas veces es difícil mantener la coherencia entre estos, pero que es fundamental para lograr una comprensión entera del problema. Incluso en el caso en el que algunos diagramas definían escenarios similares y se superponían, utilizar diferentes diagramas nos permitió ver estos escenarios desde distintos niveles de perspectiva.

La experiencia que se lleva el grupo de las distintas herramientas utilizadas es que cada una debe ser usada en el contexto adecuado, ya que la realización de los diagramas lleva un tiempo considerable que no parecería justificarse si el escenario no es lo suficientemente complicado, o si el diagrama no aporta mucho más que leer el problema en lenguaje natural.