



UNIVERSIDAD NACIONAL DE LOJA

**Área de la Energía las
Industrias y los Recursos
Naturales no Renovables**

INGENIERÍA EN SISTEMAS

**TEMA: "SISTEMA FINANCIERO-CONTABLE PARA EL
FONDO DE JUBILACIÓN, CESANTÍA
ADMINISTRATIVO DE LA UNIVERSIDAD
NACIONAL LOJA"**

**TESIS PREVIA A LA
OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS.**

AUTORES:

**PALACIOS MOROCHO MILTON RICARDO
TOLEDO JORGE SALVADOR**

DIRECTOR DE TESIS:

ING. MILTON LABANDA JARAMILLO

LOJA - ECUADOR

2007

CERTIFICACIÓN

Ingeniero MILTON LABANDA JARAMILLO

CATEDRÁTICO DE LA CARRERA DE INGENIERÍA EN SISTEMAS Y DIRECTOR DE TESIS.

Certifica haber participado en la revisión de la presente tesis, la misma que reúne los requisitos que exige el reglamento de la Universidad Nacional de Loja, por tal razón autorizo su presentación.

Loja, Abril del 2007

.....
Ingeniero Milton Labanda Jaramillo.

DIRECTOR DE TESIS

AUTORÍA

Quienes suscribimos Milton Ricardo Palacios Morocho, y Jorge Salvador Toledo declaramos ser corresponsables de las ideas, opiniones, conclusiones y recomendaciones del presente trabajo de tesis.

Milton Ricardo Palacios Morocho

AUTOR

Jorge Salvador Toledo

AUTOR

AGRADECIMIENTO

Agradecemos en primer lugar a Dios y a nuestros familiares, por habernos ayudado y permitido llegar a estas instancias de nuestra vida. Por habernos dado fuerzas y aliento cuando nos sentíamos derrotados y sin ganas de luchar. Por darnos los medios para salir adelante en nuestra carrera y por enseñarnos que la vida esta llena de momentos agradables y felices que hay que saber aprovechar.

A la Universidad, en cuyas aulas hemos aprendido a crecer ética e intelectualmente. A todos nuestros profesores de la Carrera de Ingeniería de Sistemas por brindarnos su amistad, apoyo y conocimientos que han hecho de nosotros unas personas perseverantes en la vida, y a través de ello llevar a cabo nuestras metas y lograr ser útiles a la sociedad en cualquier situación y lugar.

De igual forma nuestro agradecimiento y gratitud para con el personal que labora en el Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja, por brindarnos todos los medios para llevar a cabo nuestra investigación.

A nuestro director de Tesis que estuvo en todo momento brindándonos su apoyo incondicional para realizar el presente trabajo.

Los Autores.

DEDICATORIA

El presente trabajo de tesis lo dedico a todos quienes han sido motivadores de mi superación y me han ayudado a alcanzar con éxito mi más anhelado propósito. Lo dedico a toda mi familia pilares fundamentales en mi vida

Jorge

Dedico esta tesis a Dios, a mi padre que esta en el cielo, a mi madre, a mis hermanas, a mis familiares; y de manera muy especial a mi esposa e hija, quienes de forma desinteresada han sabido ayudarme en los momentos más difíciles para llevar a feliz término la presente investigación.

Milton

TITULO

**SISTEMA FINANCIERO-CONTABLE PARA EL FONDO DE
JUBILACIÓN, CESANTÍA ADMINISTRATIVO DE LA
UNIVERSIDAD NACIONAL LOJA.**

RESUMEN

La presente Tesis tiene como objetivo, la construcción de un Software Financiero-Contable que optimice y mejore los procesos internos del Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja.

Para el desarrollo del software se aplicó el Modelo de desarrollo de Software "Iconix"; sus diferentes etapas nos permitieron diseñar los principales diagramas UML; que facilitaron la comprensión de los diferentes requerimientos para la construcción del software deseado. La etapa de análisis y diseño nos permitió la depuración del Modelo de clases, el mismo que fue la base para la programación del software Financiero-Contable. El software fue implementado en lenguaje JAVA utilizando librerías adicionales de Java como son: Hibernate, jExcel, iReport, binding, swingLabs, lookandfields, los mismos que facilitan la programación en este lenguaje; asimismo, mejoran la presentación (visualización) y rendimiento del presente Software. Las librerías JAVA y las herramientas utilizadas son de Software Libre y tienen una amplia aceptación a nivel mundial.

Los diferentes diagramas fueron realizados en la herramienta de Modelado UML Poseidón, esta herramienta permite la generación de código java, con sus respectivas relaciones, herencias, etc. Como base de datos se utilizó MYSQL y como motor de persistencia de datos se utilizó Hibernate, que también son software Open Source.

Las pruebas se las realizó una vez instalado el programa en el Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja, permitiendo su uso a los diferentes usuarios del sistema.

INDICE

CONTENIDO	Página
CARATULA	i
CERTIFICACIÓN	ii
AUTORÍA	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
TITULO	vi
RESUMEN	vii
INDICE	viii
INTRODUCCIÓN	1
METODOLOGÍA	3
MARCO TEÓRICO	7
CAPITULO I: ASPECTOS GENERALES DEL FONDO DE CESANTÍA Y JUBILACIÓN ADMINISTRATIVO DE LA UNIVERSIDAD NACIONAL DE LOJA	7
1.1. Instituciones Financieras	7
1.2. Contabilidad Financiera	8
1.3.Fondo De Cesantía Y Jubilación Administrativo De La Universidad Nacional De Loja	10
CAPITULO II: MODELO DE DESARROLLO DE SISTEMAS INFORMÁTICOS “ICONIX”	12
2.1. Enfoque Del Modelo De Desarrollo Iconix	12
2.2. Recolección De Requerimientos	14
2.3. Técnicas Recolección De Requerimientos	17
2.4. Los Problemas Del Análisis De Requisitos	18
2.5. Pasos Importantes Para El Análisis De Requerimientos	19
2.6. Análisis Y Diseño Preliminar	25
2.6.1. Modelado de casos de uso	25
2.6.2. Análisis de robustez	28
2.6.3. Diagramas de clases	30
2.7. Modelado De Interacción	31
2.7.1. Diagramas De Secuencia	31
2.7.2. Complementar el diagrama de Clases.	33
2.8. IMPLEMENTACIÓN	35
2.8.1. Administración de Proyectos	36
2.8.2. Pruebas	37
CAPITULO III. HERRAMIENTAS DE DESARROLLO	38
3.1 Lenguaje de Programación JAVA	38
3.2. Librería JExcel (jxl.jar)	39
3.3. JGoodies	42
3.3.1. Binding	43
3.4. Herramienta para la Persistencia de Datos: Hibernate	46
3.5. Herramienta de programación Visual NETBEANS	53
3.6. Base de Datos MYSQL	54

3.7. Herramienta visual de Modelamiento UML. Visual Poseidón	56
3.8. IREPORT	58
CAPITULO IV: PRESENTACION DE RESULTADOS	61
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	218
TABLA DE ILUSTRACIONES	223
BIBLIOGRAFÍA	224
ANEXOS	225
ANEXO 1	226
ANEXO 2	228
ANEXO 3	229
ANEXO4	232

INTRODUCCIÓN

La presente investigación de Tesis, trata la construcción de un software Financiero-Contable. Este software es una solución a los requerimientos de información que posee el Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja.

La necesidad de informatizar los procesos Financieros y Contables; principales actividades que se desarrollan dentro del Fondo, nos permitieron plantear la construcción de un software que facilite y agilice los procesos que dentro de esta institución realizan diariamente.

Con la construcción de este software fortalecimos los conocimientos adquiridos durante el transcurso de la Carrera; asimismo, presentamos tecnologías adicionales de software libre y gratuito que facilitarán y mejorarán la construcción de software según sean los requerimientos de las empresas.

El trabajo ha sido dividido en cinco capítulos: *Aspectos Generales del Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja, Modelo de Desarrollo de Sistemas Informáticos Iconix, Herramientas de Desarrollo, Presentación de Resultados, Conclusiones y Recomendaciones.*

En el capítulo sobre *Aspectos Generales del Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja*, nos referimos el régimen político de las Instituciones Financieras, además damos a conocer el origen y misión del Fondo de Cesantía y Jubilación Administrativo de la UNL, también indicamos el tipo de contabilidad a la que se somete como institución financiera.

En el capítulo *Modelo de Desarrollo de Sistemas Informáticos "Iconix"*, manifestamos los pasos más relevantes que este modelo ofrece para el análisis, diseño, implementación y pruebas de un proyecto de software, así mismo como los principales diagramas UML que utiliza este modelo de desarrollo de software.

En el capítulo *Herramientas de Desarrollo*, indicamos las diferentes herramientas visuales y gratuitas que se usó para el desarrollo del software como: Poseidón, Netbeans, etc.; las mismas que fueron utilizadas para representar gráficamente las diferentes etapas del Desarrollo de Software. Además damos a conocer las diferentes librerías adicionales a java, que facilitan, mejoran y aceleran la programación y permiten una mejor interacción con otro software.

En el capítulo *Presentación de Resultados*, exponemos todo el proceso para el desarrollo de este software, desde la etapa de análisis, diseño implementación y pruebas del software.

Gracias a esta investigación hemos construido un software muy intuitivo para el uso del usuario, con una presentación visual agradable, gracias a los diferentes componentes java utilizados.

El proyecto informático diseñado, responde a los requerimientos presentados por el Fondo de Cesantía y Jubilación de la Universidad Nacional de Loja.

METODOLOGÍA

Esta investigación esta enmarcada dentro de la Ingeniería de software, ya que se utilizó características propias para construir una aplicación de calidad; entre las características utilizadas están: fiabilidad, facilidad de mantenimiento y de uso, eficiencia, etc.

En la construcción y desarrollo de este proyecto se aplicaron métodos y técnicas para resolver los problemas, la informática aportó con las herramientas y procedimientos sobre los que se apoya la ingeniería de software.

Las capas de la ingeniería del software que se abarcó en esta investigación son: Enfoque de calidad del producto, Proceso, Métodos, Herramientas.

Además se utilizó la Metodología de Desarrollo Orientado a Objetos con la que se identificaron y organizaron los conceptos del dominio de la aplicación; y, no tanto su representación final en un lenguaje de programación.

En la Metodología de Desarrollo Orientada a Objetos utilizamos el lenguaje UML, que posee una notación gráfica muy expresiva, notación gráfica que permitió representar en mayor o menor medida todas las fases del proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de paquetes.

En el modelado UML utilizamos tres clases de bloques de construcción: **Elementos, Relaciones, Diagramas.**

En el Desarrollo Orientado a Objetos con UML se siguió las siguientes fases:

- Construcción:
 - Diseño de Alto Nivel: Se abordó el problema viendo al sistema a construir como una caja negra, centrándonos en la visión que desde el exterior tienen los actores, esto es, en los casos de uso. Se analizó el problema construyendo un modelo conceptual.
 - Diseño de Bajo Nivel: El sistema definido en la fase anterior se especificó en detalle, se describió todas las operaciones que el sistema va a tener que realizar internamente para satisfacer lo especificado en el diseño de alto nivel.
 - Implementación: Se llevó al lenguaje de programación JAVA, lo especificado anteriormente en el Diseño.
 - Pruebas: Se ejecutaron una serie de pruebas para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación y Especificación de Requisitos.

- Instalación: Se puso en marcha el sistema en el entorno de uso previsto.

Con la Metodología de Desarrollo Orientada a Objetos se tuvo la posibilidad de aplicar un Modelo de Desarrollo de software para la construcción de la aplicación, este Modelo de Desarrollo es "ICONIX", modelo que utiliza etapas de Desarrollo orientado a objetos y UML.

Las fases de "ICONIX" para el desarrollo del software son:

- Definición del proyecto y planificación: aquí se definió el alcance y límites del proyecto. Se realizó los estudios de factibilidad y relaciones costo/beneficio.

- Análisis
 - *Análisis del Fondo de Cesantía y Jubilación Administrativo*: en esta etapa se modeló el Fondo de Cesantía y Jubilación Administrativo o parte del mismo, en orden de comprender su naturaleza; como se realizan actualmente las actividades, y como los usuarios desean que se realicen en el futuro. Provee una comprensión preliminar de las áreas específicas del negocio a ser informatizadas. Esta etapa también es conocida en otras metodologías como estudio del sistema actual.
 - *Análisis de requerimientos del sistema*: se estableció con claridad las capacidades requeridas para el nuevo sistema.

- Diseño
 - *Diseño Lógico*: se identificó los componentes de software/hardware necesarios para satisfacer los requerimientos, así también se especificó las relaciones arquitecturales entre dichos componentes. En el diseño lógico se obvió los detalles técnicos específicos requeridos para mapear el diseño en un entorno de implementación específico.
 - *Diseño Físico*: Se tomó decisiones técnicas considerando arquitecturas de hardware específicas, sistemas de bases de datos, lenguajes de programación, utilización de paquetes, etc. Aquí también se tomó decisiones con respecto a características de implementación como arquitectura cliente/servidor, distribución de objetos, etc.

- Construcción
 - *Desarrollo*: El diseño físico se implementó en un lenguaje de programación, o entorno específico de desarrollo (JAVA).

- Prueba: se realizaron pruebas del software para validar su correcto funcionamiento y detectar fallas que deberán ser depuradas.
- Documentación: Se realizó la documentación técnica sobre la aplicación, manual de usuario, etc.

- Aprobación

- Operación y Mantenimiento
 - ✓ Hardware y Redes: También se realizó una evaluación previa del hardware disponible en esta entidad, con la finalidad de establecer los mejores mecanismos para su configuración visual y de rendimiento de cada equipo de la red en donde operará el sistema.

En el proceso de investigación se necesitó información acerca del problema a investigar, para lo cual se utilizó el método científico *“es el conjunto de procedimientos lógicos que sigue la investigación para descubrir las relaciones internas y externas de los procesos de la realidad natural y social”*¹ que nos permitió conocer de mejor manera el problema.

También se utilizó el método analítico sintético *“que es la desarticulación práctica o mental del todo en sus partes y de reunificación del todo a base de sus partes”*², que nos permitió conocer de mejor manera el problema y estudiarlo para dar la solución más optima. Además se llevo a cabo la observación científica, y como instrumento de Investigación se aplicó la entrevista y el cuestionario.

¹ GUTIÉRRES, Abraham. Métodos de Investigación. Novena Edición. Quito-Ecuador, 2000. Pág.38.

² Ibid

MARCO TEÓRICO

CAPITULO I

ASPECTOS GENERALES DEL FONDO DE CESANTÍA Y JUBILACIÓN ADMINISTRATIVO DE LA UNIVERSIDAD NACIONAL DE LOJA

1.1. INSTITUCIONES FINANCIERAS.

Las Instituciones financieras del país se rigen por la Ley General de Instituciones Financieras que señala en su Art. 2. Decreto Ejecutivo No. 1852. RO/ 475 de 4 de julio de 1994:

“ARTICULO 2.- Son instituciones financieras privadas:

- a. Bancos;
- b. Sociedades Financieras;
- c. Asociaciones Mutualistas de Ahorro y Crédito para la Vivienda; y,
- d. Cooperativas de Ahorro y crédito que realizan intermediación financiera con el público”³.

Por lo cual tienen todos los derechos y obligaciones establecidas en la Ley General de Instituciones del Sistema Financiero del Ecuador y particularmente a Ley de Cooperativas del Ecuador, Ley Nro. 1.031.

³ LEY GENERAL DE INSTITUCIONES DEL SISTEMA FINANCIERO.

Para propósitos de esta Ley se aplica el ARTÍCULO 2.- “son instituciones financieras privadas los bancos, las sociedades financieras o corporaciones de inversión y desarrollo, las asociaciones mutualistas de ahorro y crédito para la vivienda y las cooperativas de ahorro y crédito que realizan intermediación financiera con el público. Los bancos y las sociedades financieras o corporaciones de inversión y desarrollo se caracterizan principalmente por ser intermediarios en el mercado financiero, en el cual actúan de manera habitual, captando recursos del público para obtener fondos a través de depósitos o cualquier otra forma de captación, con el objeto de utilizar los recursos así obtenidos, total o parcialmente, en operaciones de crédito e inversión”⁴.

Las instituciones financieras son las encargadas de manejar los dineros de las personas afiliadas a estas, y su razón de existir es brindar beneficios a sus accionistas a través de la correcta inversión de los fondos económicos de estas instituciones financieras, por lo que es indispensable que diariamente se interactúe entre instituciones tipo financiera de ahorro y crédito con las personas con lo que dichas empresas lograrían coexistir y no se eliminarían.

1.2. CONTABILIDAD FINANCIERA

1.2.1. Concepto.- “Es la técnica mediante la cual se recolectan, se clasifican, se registran, se suman y se informa de las operaciones cuantificables en dinero, realizadas por una entidad económica”⁵.

Es el arte de usar ciertos principios al registrar, clasificar y sumarizar en términos monetarios datos financieros y económicos, para informar en forma oportuna y fehaciente de las operaciones de la vida de una empresa.

⁴ Ibid.

⁵ CASAS SANTASUSANA, Enrique. Contabilidad General. 10 edición. Barcelona, 2000, Pág. 21

La función principal de la contabilidad financiera es llevar en forma histórica la vida económica de una empresa, los registros de cifras pasadas sirven para tomar decisiones que beneficien al presente o al futuro.

También proporciona estados financieros que son sujetos al análisis e interpretación, informando a los administradores, a terceras personas, y a oficinas gubernamentales del desarrollo de las operaciones de la empresa.

Los criterios que deben seguir en los registros de la contabilidad financiera son:

- ✍ Aplicación correcta de conceptos contables clasificando lo que debe quedar incluido dentro del activo y lo que incluirá el pasivo.
- ✍ Presentar la diferencia entre un gasto y un desembolso.
- ✍ Decidir lo que debe constituir un producto o un ingreso, dentro de las cuentas de resultados.
- ✍ Esta contabilidad se concreta en la custodia de los activos confiados a la empresa.

Fundamentalmente se ocupa de la preparación de informes y datos para personas distintas de los directivos de la empresa.

Incluye la preparación, presentación de información para acciones, acreedores, instituciones bancarias y del Gobierno.

Características de la Contabilidad Financiera:

- ✍ Rendición de informes a terceras personas sobre el movimiento financiero de la empresa.

- ✍ Cubre la totalidad de las operaciones del negocio en forma sistemática, histórica y cronológica.
- ✍ Debe implantarse necesariamente en la compañía para informar oportunamente de los hechos desarrollados.
- ✍ Se utiliza de lenguaje en los negocios.
- ✍ Se basa en reglas, principios y procedimientos contables para el registro de las operaciones financieras de un negocio.
- ✍ Describe las operaciones en el engranaje analítico de la teneduría de libros por partida doble.

1.3. FONDO DE CESANTÍA Y JUBILACIÓN ADMINISTRATIVO DE LA UNIVERSIDAD NACIONAL DE LOJA

Las empresas financieras tienen la necesidad de llevar una contabilidad Financiera, sino también las entidades públicas y privadas se manejan a lo interior con este tipo de recursos para sus trabajadores, creando fondos de ahorro y crédito o cooperativas de ahorros en la que sus trabajadores tengan la oportunidad de ahorrar parte de su sueldo ganado mensualmente. Una de estas instituciones es la Universidad Nacional de Loja.

El origen de la Universidad Nacional de Loja se remonta al gobierno federal dirigido por don Manuel Carrión Pinzano, quien en decreto expedido al 31 de diciembre el 1859, dispone que adscritos al Colegio "San Bernardo", se impartan estudios de Jurisprudencia, Filosofía, Letras, Teología y Medicina.

A mediados de siglo, mediante decreto Ejecutivo expedido por el primer magistrado de la nación, Dr. Carlos Alfredo Arroyo del Río, el 9 de octubre de 1943, la Junta Universitaria de Loja es elevado a la categoría de Universidad, con lo cual este centro de estudios comenzó a recibir el mismo trato de otras universidades del país.

La Universidad Nacional de Loja desde sus inicios ha venido trabajando con los diferentes estamentos universitarios entre estos están los Trabajadores Administrativos, los Docentes, y los Estudiantes.

Los trabajadores universitarios decidieron formar el Fondo Administrativo Cesantía y Jubilación del Trabajador Universitario, que proporcione servicios cooperativistas como ahorro, préstamos, aperturas de libretas de ahorros a sus socios, pago de su mensualidad de trabajo, etc.

Este Fondo se formó por resolución del Honorable Consejo Universitario del 31 de Octubre de 1996; inició sus actividades el 1ro de enero de 1997; es una asociación autónoma de trabajadores administrativos que se reúne de forma voluntaria para satisfacer sus necesidades y aspiraciones económicas y sociales mediante una entidad conjunta y de gestión democrática sin fines de lucro, el fondo brinda servicios como aperturas de cuentas de aportaciones, prestamos a bajo interés a sus asociados y a las personas que laboren dentro de la universidad siempre cuando tengan el nombramiento. Su capital va de acuerdo al ingreso y egreso de los asociados en donde cada trabajador administrativo aporta con el 10% de su sueldo y la Universidad aporta con otro 10% como aporte patronal.

CAPITULO II

MODELO DE DESARROLLO DE SISTEMAS INFORMÁTICOS

“ICONIX”

El modelo de desarrollo Iconix tiene las siguientes características:

- ✍ Es un modelado de objetos conducido por casos de Uso.
- ✍ Centrado en datos: se descompone en fronteras de datos
- ✍ Basado en escenarios que descomponen los casos de uso.
- ✍ Enfoque iterativo e incremental.
- ✍ Ofrece trazabilidad.
- ✍ Uso directo de UML.

2.1. ENFOQUE DEL MODELO DE DESARROLLO ICONIX

“El proceso de ICONIX maneja casos de uso, como el RUP, pero le falta mucho para llegar al nivel del RUP. También es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño que hace XP. Este proceso también hace uso aerodinámico del UML mientras guarda un enfoque afilado en el seguimiento de requisitos. Y, el proceso se queda igual a la visión original de Jacobson del “manejo de casos de uso”, esto produce un resultado concreto, específico y casos de uso fácilmente entendible, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real⁶.”

⁶ www.iconix.net

La Figura 1 muestra el cuadro del proceso. El diagrama retrata la esencia del enfoque aerodinámico al desarrollo del software, que incluye un juego mínimo de diagramas de UML y algunas valiosas técnicas que se toman de los casos del uso para codificar rápida y eficazmente. El enfoque es flexible y abierto.

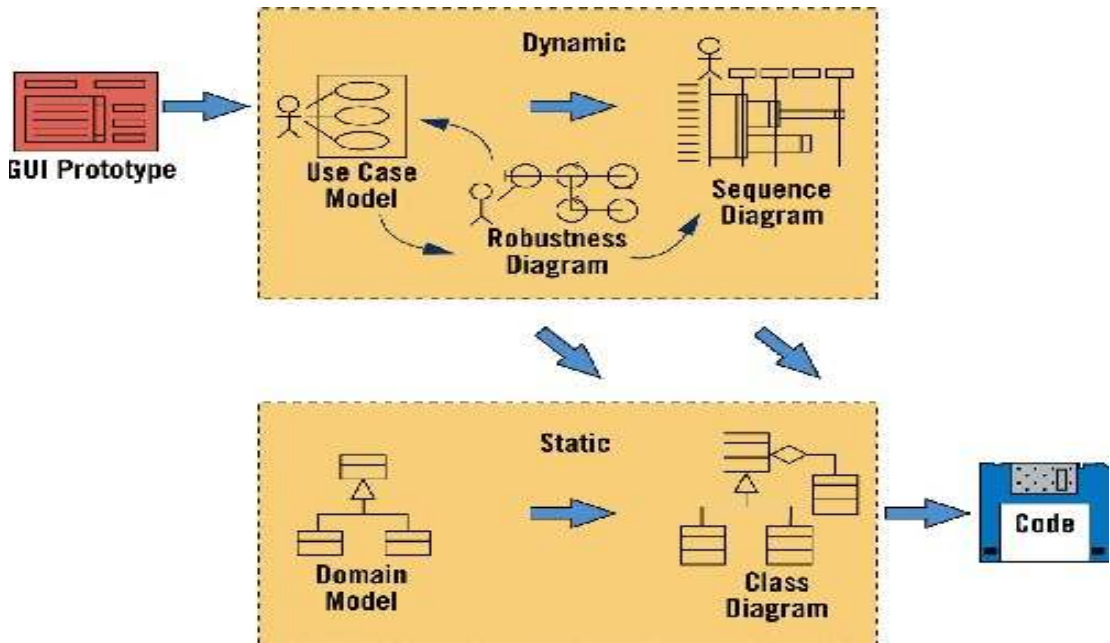


Figura 1: Modelo de desarrollo Iconix

Iconix es reiterativo e incremental. Las iteraciones múltiples ocurren entre el desarrollo del modelo del dominio e identificar y analizar los casos de uso. El modelo estático se refina incrementalmente durante las iteraciones sucesivas a través del modelo dinámico (compuesto de casos de uso, análisis de robustez y el diagrama de secuencia).

Iconix ofrece un alto grado de seguimiento. Por el camino, a cada paso usted consultara de alguna manera los requisitos anteriores. Nunca hay un punto en que el proceso le permita desviarse lejos de las necesidades del usuario.

Iconix ofrece uso aerodinámico del UML. Los pasos que se utilicen representan un mínimo del acercamiento, estos comprenden el juego mínimo de pasos que nosotros

hemos encontrado para ser necesarios y suficientes en el desarrollo de un proyecto Orientado a Objetos exitoso.

2.2. RECOLECCIÓN DE REQUERIMIENTOS

Los requerimientos de un sistema de software, cuando se ven en su conjunto son extensos y detallados, y además contienen múltiples relaciones entre si. Obtenemos la posibilidad de especificar sistemas complejos al documentar especificaciones simples y concisas para el sistema. Esto se logra mediante el *clasificar, estructurar y organizar* todo lo que el sistema debe de hacer. En otras palabras al analizar sus requerimientos.

El análisis de requerimientos consiste brevemente en los siguientes pasos:

1. Obtener información acerca de lo que los usuarios desean.
2. Clasificar esos deseos para comenzar a estructurar los requerimientos del sistema.
3. Identificar los niveles de jerarquía del sistema y empezar a alojarlos ya clasificados los requerimientos en cada nivel.
4. Especificar formalmente los requerimientos de acuerdo al nivel de audiencia que se desea⁷.

1. Obteniendo la información

Los requerimientos se enfocan a describir las necesidades del cliente, entonces es lógico que para recabarlos haya que obtener la información de primera mano. Esto es, mediante entrevistas con el cliente o recabando documentación

⁷ <http://www.geocities.com/txmetsb/compleji.htm>

que describa la manera que el cliente desea que funcione el sistema de software⁸.

Como cada necesidad del cliente es tratada de diferente forma, es necesario clasificar estas necesidades para saber cuales de ellas serán satisfechas por el software y cuales por algún otro producto del sistema.

2. Clasificación de los requerimientos

El clasificar requerimientos es una forma de organizarlos, hay requerimientos que por sus características no pueden ser tratados por igual. La siguiente es una recomendación de como pueden ser clasificados los requerimientos aunque cada proyecto de software pueda usar sus propias clasificaciones⁹.

- *Requerimientos del "entorno"*.- El entorno es todo lo que rodea al sistema. Existen cierto tipo de requerimientos que se clasifican en esta categoría por que el sistema usa el entorno y lo necesita como una fuente de los servicios necesarios para que funcione. Ejemplos del entorno podemos mencionar: sistemas operativos, sistema de archivos, bases de datos.¹⁰
- *Requerimientos "ergonómicos"*.- El más conocido de los requerimientos ergonómicos es la interface con el usuario o GUI (Graphic User Interface). En otras palabras, los requerimientos ergonómicos son la forma en que el ser humano interactúa con el sistema¹¹.

⁸ Ibid

⁹ Ibid

¹⁰ Ibid

¹¹ Ibid

- *Requerimientos de Interface.*- La interface es como interactúa el sistema con el ser humano o con otros sistemas (enfoque opuesto a los requerimientos ergonómicos), La interface es la especificación formal de los datos que el sistema recibe o manda al exterior¹².
- *Requerimientos funcionales.*- Estos son los que describen lo que el sistema debe de hacer. Es importante que se describa el ¿Que? Y no el ¿Como? Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema¹³.
- *Requerimientos de desempeño.*- Estos requerimientos nos informan las características de desempeño que deben de tener el sistema. ¿Que tan rápido?, ¿Que tan seguido?, ¿Cuántos recursos?, ¿Cuántas transacciones? Este tipo de requerimientos es de especial importancia en los sistemas de tiempo real en donde el desempeño de un sistema es tan crítico como su funcionamiento.
- *Disponibilidad* (en un determinado periodo de tiempo).- Este tipo de requerimientos se refiere a la durabilidad, degradación, portabilidad, flexibilidad, contabilidad y capacidad de actualización. Este tipo de requerimientos es también muy importante en sistemas de tiempo real puesto que estos sistemas manejan aplicaciones críticas que no deben de estar fuera de servicio por periodos prolongados de tiempo.
- *Restricciones de diseño.*- Muchas veces las soluciones de un sistema de software son normadas por leyes o estándares, este tipo de normas caen como "restricciones de diseño".

¹² Ibid

¹³ Ibid

3. Manejo de requerimientos

Para lograr el control de los requerimientos, se revisa los requerimientos antes de que estos sean incorporados al proyecto de software y cada vez que los requerimientos cambian los planes, productos, y actividades son ajustados para quedar en línea con los nuevos requerimientos de software".

Para obtener el nivel que requiere el CMM (Capability Maturity Model) en manejo de requerimientos debemos de tomar en cuenta dos cosas.

- Que los requerimientos deben de ser **revisados (y aprobados)** por el grupo de requerimientos, y **no son impuestos** en su totalidad por presiones externas ajenas al proyecto.
- Los requerimientos técnicos y no técnicos forman un conjunto entre si, si cambia uno forzosamente deberán cambiar los demás. Esto es: más contenido técnico implica o más costo, o menos calidad o mas tiempo estimado de entrega.

2.3. TÉCNICAS RECOLECCIÓN DE REQUERIMIENTOS

Las técnicas más habituales en la elicitación de requerimientos son las entrevistas, el desarrollo conjunto de aplicaciones, la tormenta de ideas y la utilización de escenarios más conocidos como casos de uso.

A estas técnicas se las suele apoyar con otras técnicas complementarias como la observación in situ, el estudio de documentación, los cuestionarios, la inmersión en el

negocio del cliente o haciendo que los ingenieros de requerimientos sean aprendices del cliente.

2.4. LOS PROBLEMAS DEL ANÁLISIS DE REQUISITOS

El primer problema que se presenta es la *captura de los requisitos del usuario*. Para empezar, necesitamos recoger los requisitos de los usuarios o clientes de una manera sistemática y organizada. Para ello precisamos de unas directrices o líneas guía, ya que en general los usuarios expresan los requerimientos de la aplicación de forma muy variable, tanto en la forma como en el contenido. Nos interesa pues sistematizar la captura, con el fin de hacer los requisitos manejables y analizables.

Una vez conseguidos los requisitos, pasamos a la fase de análisis. En ella, lo que haremos es analizar los requisitos obtenidos de los usuarios con el fin de comprenderlos, y a partir de ellos desarrollar una especificación de la aplicación, que deberá ser completa y consistente, y deberá estar expresada de una manera al menos semiformal, no simplemente textual. En este proceso, encontraremos habitualmente gran cantidad de problemas en los requisitos, áreas no especificadas, requisitos contradictorios, y afirmaciones (aparentemente) vagas e irrelevantes. Eso nos llevará de vuelta a los usuarios con el fin de mejorar la calidad de los requisitos: pero debemos abordarles sabiendo lo que queremos conseguir, qué aspectos de los requisitos obtenidos inicialmente nos interesa aclarar, y el porqué.

Después de haber obtenido una buena lista de requisitos, comenzamos con el análisis y el diseño de la aplicación, Aquí surgieron dos nuevos problemas:

1. *La trazabilidad de los requisitos*: cómo seguir un requisito de usuario por el análisis, el diseño y el código, que nos permita comprobar que el requisito ha sido tenido en cuenta y cómo lo ha sido. Y esto enlaza directamente con el problema de la mantenibilidad

2. *La Mantenibilidad*: cuando los requisitos comienzan a evolucionar -como sin duda lo harán-, cómo podemos ir evolucionando el diseño y el código consistentemente con ello, y cómo seguir manteniendo la trazabilidad¹⁴.

2.5. PASOS IMPORTANTES PARA EL ANÁLISIS DE REQUERIMIENTOS

- ✍ Identificar Objetos del dominio y relaciones de agregación y generalización.
- ✍ Prototipo Rápido
- ✍ Identificar Casos de Uso
- ✍ Organizar Casos de uso en grupos (paquetes)
- ✍ Asignar requerimientos funcionales a casos de uso y objetos del dominio.

2.5.1 Construir un Modelo del Dominio del Problema

A continuación se va a construir un modelo descriptivo del problema que queremos resolver. Para ello vamos a utilizar dos técnicas de forma complementaria. En primer lugar, vamos a construir un modelo de objetos del dominio del problema. Vamos a ir identificando todos los conceptos característicos del problema, susceptibles de expresarse como clases de objetos (es decir, conceptos que tienen propiedades -o atributos- y relaciones con otros objetos del dominio).

Vamos a construir un modelo estático de objetos del dominio del problema, añadiéndole las operaciones características. En él expresaremos las clases de objetos del problema, y las describiremos; las características o atributos de esas clases de objetos, de nuevo con su descripción en relación a la clase

¹⁴ CHOQUE, Guillermo. *Ingeniería de Requerimientos*. [www.inf.puc-rio.br/~wer98/artigos/79.html]

correspondiente; las relaciones entre esas clases de objetos: las asociaciones y las agregaciones (asociaciones con dependencias), y eventualmente las clasificaciones (si bien éstas también las podemos expresar en el modelo siguiente); y finalmente, las operaciones que, en el dominio del problema, se realizan sobre esas clases de objetos.

“Si el modelo de objetos que hemos construido se corresponde bien con el sistema, cualquier afirmación que hagamos sobre éste deberá ser posible expresarla en términos de los objetos-atributos-relaciones-operaciones de nuestro modelo (que incluirá, todo el léxico específico del dominio), junto con otra serie de expresiones o términos del lenguaje habitual.

Hasta aquí, en el modelo de dominio hemos recogido sólo aspectos "naturales" del dominio del problema (es decir, aspectos que forman parte del modelo del problema que tienen los clientes o usuarios). Sin embargo, dado que se trata en definitiva de una técnica de clasificación multidimensional, no hay ninguna razón para no incluir, en la lista de facetas, todos aquellos aspectos que nos puedan interesar (como por ejemplo, aspectos de arquitectura, diseño o implementación)”¹⁵.

2.5.2. Fuentes de Información

Las principales fuentes de información para construir el modelo de dominio son:

- ✍ Descripción de alto nivel del problema.
- ✍ Requerimientos de bajo nivel.
- ✍ Conocimiento de expertos.
- ✍ Literatura que proporcione el Cliente.

15

www.itlp.edu.mx/publica/tutoriales/analisis/index.htm

2.5.3. Clases y objetos.

Concepto de Clase.- Una clase es una categoría o grupo de cosas que tienen atributos y acciones similares. Ejemplo: Cualquier cosa dentro de la clase Lavadoras: tienen atributos como son la marca, el modelo, el número de serie y la capacidad. Entre las acciones de las cosas de esta clase se encuentran: “agregar ropa”, “agregar detergente”, “activarse” y “sacar ropa”¹⁶.

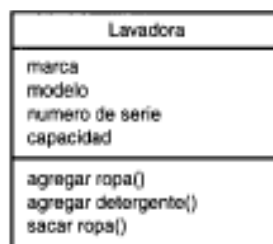


Figura 2: Símbolo UML de la Representación de una clase

Concepto de Objeto.- Un objeto es una instancia de la clase (una entidad que tiene valores específicos de los atributos y acciones). Por ejemplo: Su Lavadora, podría tener la marca LG, el modelo Washmeister, el número de Serie GL57774 y una capacidad de 7Kg.

A continuación el ejemplo muestra la forma en que el UML representa a un objeto. Vea que el símbolo es un rectángulo, como en una clase, pero el nombre está subrayado. El nombre de la instancia específica se encuentra a la izquierda de los dos puntos; y el nombre de la clase a la derecha.

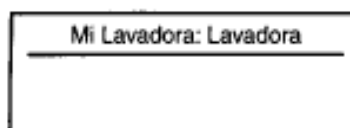


Figura 3: Símbolo UML de la Clase Lavadora

¹⁶ SCHMULLER, Joseph. Apreniendo UML en 24 horas. Pág. 19

2.5.4. Modelando el Dominio.

Procedimiento para el modelado del dominio del problema:

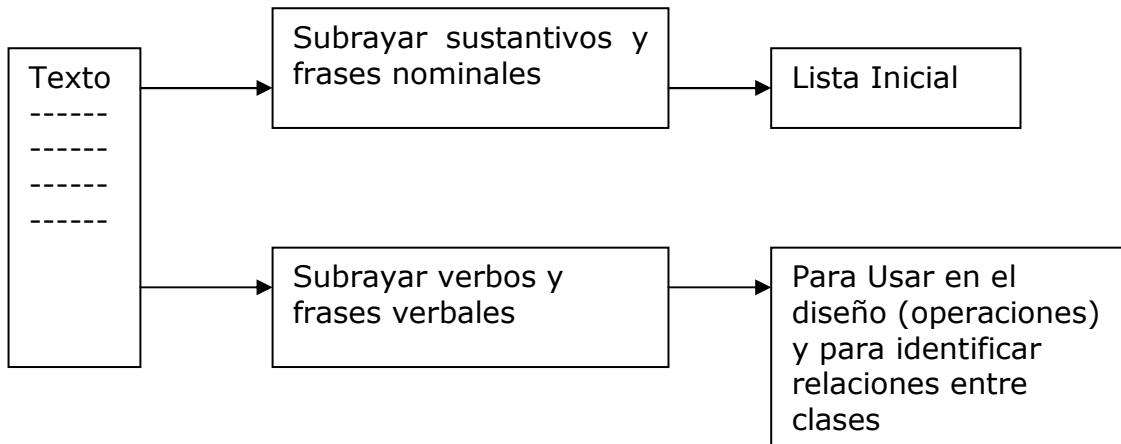


Figura 4: Procedimiento del Modelado del Dominio (Parte1)

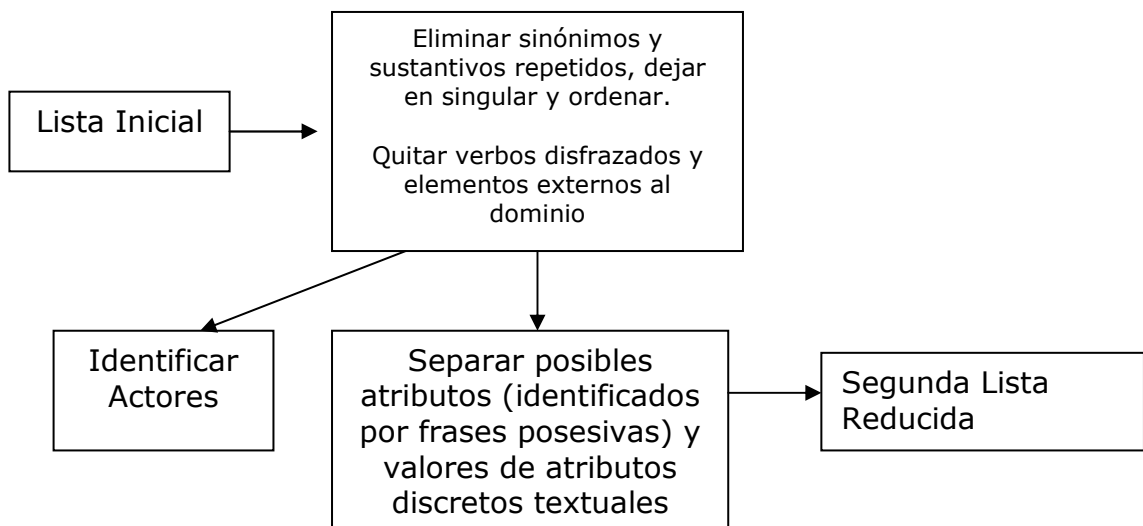


Figura 5: Procedimiento para construir el modelo de dominio (Parte 2)

- Tomar documentos disponibles y hacer una lectura rápida, subrayando los sustantivos y anotando frases posesivas y verbos (estos verbos nos van a servir en el futuro).
- Los sustantivos y frases nominales se convertirán en objetos y atributos.
- Los verbos y frases verbales se convierten en operaciones y relaciones.
- Las frases posesivas indican los sustantivos que son atributos y no objetos.
- Formar una lista con los sustantivos y frases nominales identificados, evitando los plurales y las repeticiones y ordenándola alfabéticamente.
- Revisar la lista eliminando los elementos innecesarios (irrelevantes o redundantes) o incorrectos (vagos o conceptos fuera del alcance del modelo o representan acciones aún cuando parezcan sustantivos).

- Construir relaciones de generalización:
 - Una generalización es una relación en la cual una clase es una generalización de otra. También se le llama “tipo-de” o “es-una”. La clase más general se llama Antecesor o Superclase y la otra (refinamiento de la Primera) descendiente o Subclase.
 - La subclase hereda los atributos y métodos de la superclase y las asociaciones en que participa.

- Establecer asociaciones entre clases:
 - Una asociación es una relación estática entre dos clases; indican dependencia, pero no acción (aunque se las nombre con un verbo).
 - Deben ser persistentes (es modelo estático).

- Establecer relaciones de agregación:
 - Una agregación es una relación en la cual una clase está formada por otras (sus partes).
 - A veces se le llama “parte-de”.
 - En UML se distingue una forma más fuerte llamada Composición, pero para este método no se hará diferencia.

- Clases de asociación:
 - Una clase de asociación es una variante de las asociaciones muy útil cuando hay relaciones muchas a muchas entre clases.

Nota. Pueden conseguirse clases de dominio a partir de entidades en bases de datos preexistentes. Cuando una clase tiene demasiados atributos, conviene dividirla en clases auxiliares y usar agregación para reunir las.

2.5.5. Los Elementos importantes del modelo del Dominio

La primera cosa que usted debe hacer cuando este construyendo un modelo estático de su sistema es el hallar las clases apropiadas que con precisión representan las abstracciones reales de los problemas que se presentan en el modelo del dominio.

Es probable que los mejores recursos de clases sean la declaración del problema de alto nivel. Para empezar, ponga todas las declaraciones pertinentes de estas áreas (e incluso otros) como pueda encontrar, y entonces señale o resalte, todos los sustantivos de la frase. Refine las listas gradualmente, los sustantivos de la frases se volverán objetos y atributos, mientras los verbos se volverán funcionamientos y asociaciones. Lo posesivo ("su", "nuestro" y "suyo") tiende a indicar que los sustantivos deben ser los atributos, en lugar de los objetos.

Luego, seleccione de su lista de clases candidatas y elimine los artículos innecesarios. Busque las clases que son redundantes, no pertinentes, incorrectas o vagas.

También se debe tomar algunas decisiones de la inicial sobre la generalización (el "tipo de" o "es un" relación entre las clases) mientras construye su diagrama de clases. Generalice a más de un nivel de subclase. Recuerde buscar el tipo de declaraciones que son verdad en el mundo real. El modelamiento del dominio también es el área apropiada para las decisiones sobre las agregaciones ("parte de" o "tiene" relaciones entre clases).

Finalmente, tal como muchos diagramas de relación de entidad (ERD), su modelo del dominio, pone al día para mostrar las asociaciones (las relaciones estáticas entre los pares de clases) debe ser una verdadera declaración sobre el espacio del problema, independiente del tiempo (es decir, estática). Este modelo sirve como la construcción de su modelo de la clase estático.

2.6. ANÁLISIS Y DISEÑO PRELIMINAR:

- ✍ Escribir descripciones de casos de uso.
 - Cursos Normales de eventos y cursos alternos.
- ✍ Análisis de Robustez
 - Identificar grupos de objetos que realizan el escenario.
 - Actualizar diagramas de clases del dominio.
- ✍ Finalizar Diagramas de Clases.

2.6.1. Modelado de Casos de Uso

2.6.1.1. Concepto

Un caso de uso es una secuencia de acciones que un actor (usualmente una persona, pero que puede ser una entidad externa, como otro sistema o un elemento de hardware) realiza dentro del sistema.

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, esta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es

crear un sistema que pueda ser utilizado por la gente en general (no solo por expertos en computación)¹⁷.

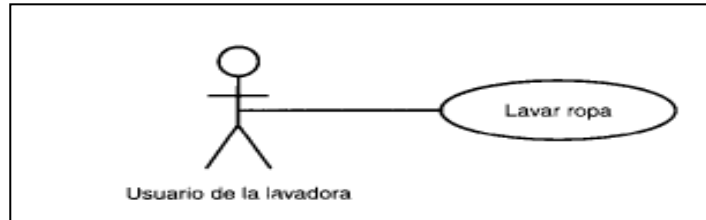


Figura 6: Representación de un Caso de Uso

2.6.1.2. Características de los Casos de uso.

- ✍ Los casos de uso buscan capturar los requerimientos del usuario para el nuevo sistema.
- ✍ Los casos de uso puede empezar desde cero o tomando como inicio un sistema anterior.
- ✍ Los casos de uso especifican escenarios detallados de lo que hace el usuario para lograr sus fines.
- ✍ Los casos de uso son la base de todo el proceso de desarrollo basado en el Modelo Iconix.

Así pues, los objetivos principales de la elaboración de casos de uso son:

- ✍ Definir el límite entre el sistema a desarrollar y los elementos externos a ese sistema (actores usuarios del sistema).

¹⁷ SCHMULLER, Joseph. Apreniendo UML en 24 horas. Pág. 33

- ✎ Capturar el conjunto de funcionalidades y comportamientos del sistema a desarrollar.

Cada caso de uso se documenta mediante una representación gráfica y un texto con la descripción de las situaciones o escenarios ante los que el usuario se pueda encontrar en su interacción con el sistema.

En la Figura se muestra un ejemplo de Diagrama de Casos de Uso del Sistema Financiero Contable del Fondo de Cesantía y Jubilación de la Universidad Nacional de Loja.

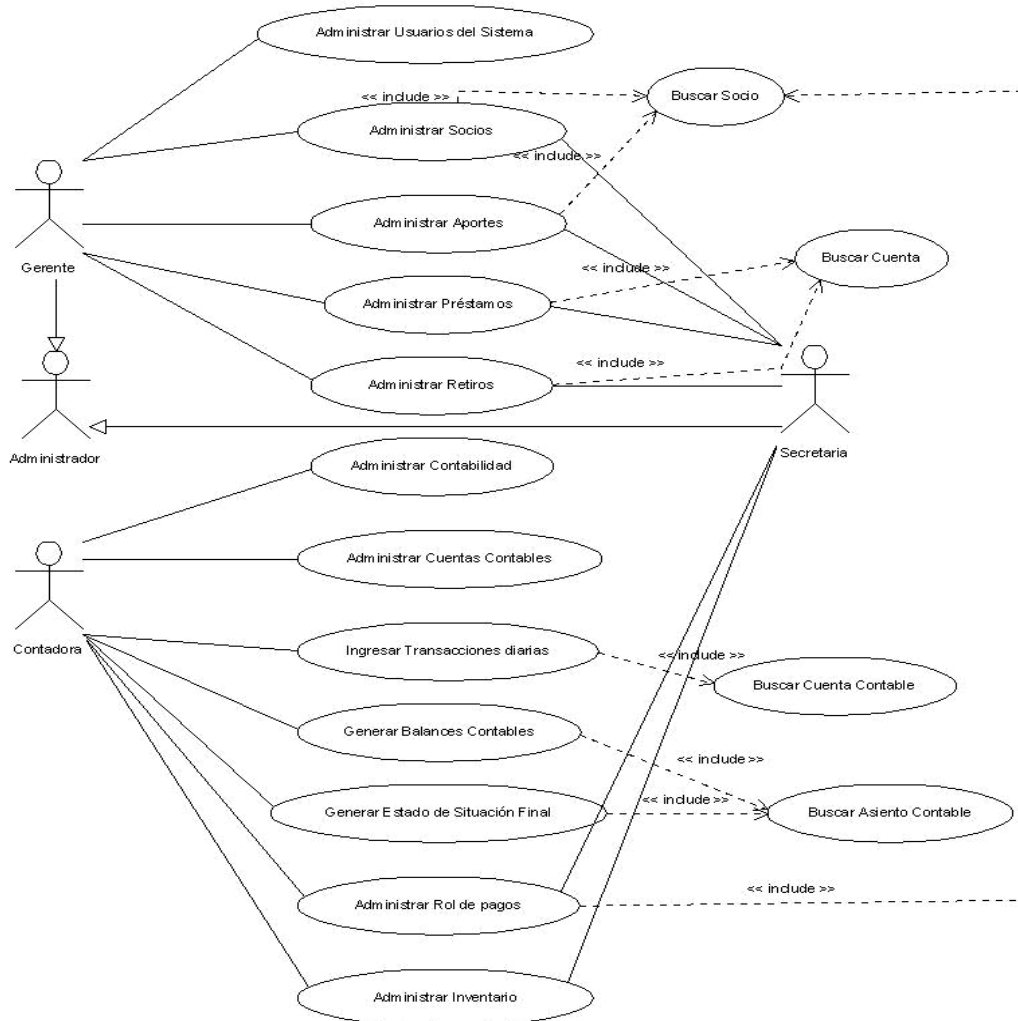


Figura 7: Diagrama de Casos de Uso

2.6.1.3 Como escribir los Casos de Uso

Para describir los casos de uso se puede tener en cuenta:

- ✎ Escriba un párrafo o más para cada caso de uso, describiendo su comportamiento
- ✎ Si sólo hay una frase, quizá dividió demasiado el caso de uso y debería reunirlos en varios.
- ✎ Si es demasiado extenso o complicado, quizá debe subdividirlo.
- ✎ Deben guardar estrecha correlación: el manual de usuario y la interfaz gráfica de usuario (GUI).
- ✎ Primero escribimos el manual y luego se trabaja en el código (como sea: dibujos, prototipos rápidos, objetos de utilería, etc.)
- ✎ Asegurarse que los casos de uso corresponden a lo que efectivamente hacen los usuarios.

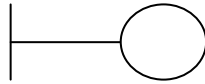
2.6.2. Análisis de Robustez

Los diagramas de robustez tienen los siguientes usos:

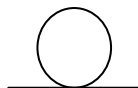
- ✎ **Comprobación de Sanidad.**- Nos permite revisar las ideas de los casos de Uso (Revisar su comportamiento razonable).
- ✎ **Comprobación de entereza.**- Nos permite asegurar que en los casos de uso de cubra el camino básico y los posibles caminos alternos.
- ✎ **Descubrir objetos.**- si son estrictamente necesarios.
- ✎ **Diseño Preliminar.**- Los diagramas de robustez son la primera vista del nuevo sistema.

Los Objetos que intervienen dentro del diagrama de Robustez son:

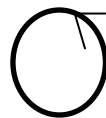
- ✎ **Objetos Fronterizos:** o de límite; son objetos con los cuales puede interactuar el usuario con la Interfaz. Su representación gráfica es la siguiente.



- ✎ **De Entidad.-** Son generalmente los objetos del modelo de dominio; son objetos de almacenamiento.



- ✎ **De Control.-** (Controles) Intermediarios entre los fronterizos y de entidad. Son métodos o comportamientos de los objetos.



2.6.2.1. Reglas para construir los diagramas de robustez

- ✎ Un actor únicamente puede comunicarse con un objeto *De límite*.
- ✎ Los objetos *De límite* pueden interactuar con objetos controladores u objetos *De Control*.
- ✎ Los objetos *De Control* pueden interactuar con objetos *De límite*, *De Entidad* u otros objetos *De Control*¹⁸.

¹⁸ <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x320.html>

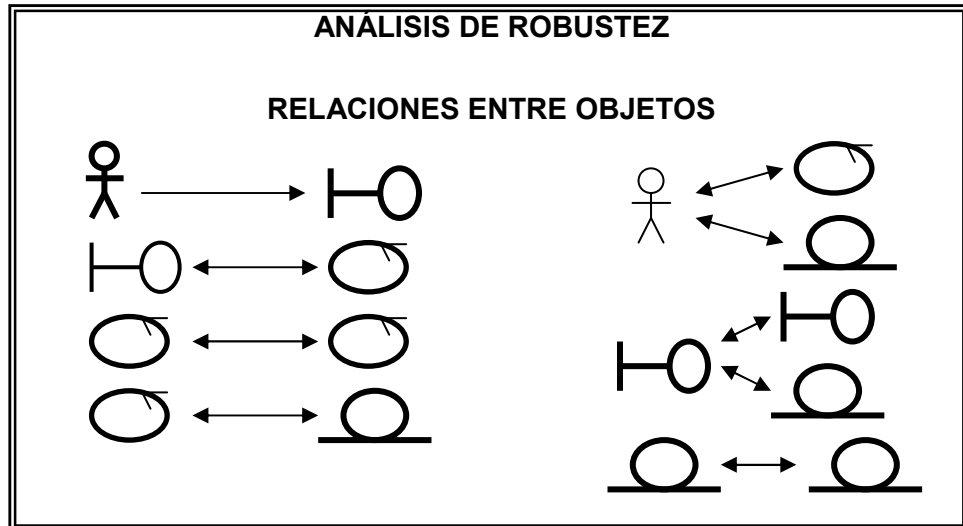


Figura 8: Relaciones entre objetos

Un diagrama de robustez representa el curso básico y los alternos de cada caso de uso.

Los diagramas de robustez deben tener entre 2 y 6 objetos de control por cada caso de uso. Deben conectarse entre si mediante flechas que van en una o dos direcciones.

2.6.3. Diagramas de Clases

Un diagrama de clases es un tipo de modelo estático. Un diagrama de clases describe la vista estática del sistema. Aunque tiene similitudes con un modelo de datos (entidad-relación), recuerde que las clases no solo muestran la estructura de la información, sino que describen también el comportamiento. Un propósito de los diagramas de clases es definir una base para otros diagramas donde otros aspectos del sistema son mostrados tales como los estados de objetos o la colaboración entre ellos mostrados en los modelos dinámicos. Una clase en un diagrama de clases puede ser directamente implementada en un lenguaje de programación orientado a objetos, en nuestro caso JAVA.

A medida que más y más clases son añadidas al modelo, una representación textual de las clases no es suficiente. Los diagramas de clases son creados para proporcionar vista de algunas o todas las clases en el modelo.

2.7. MODELADO DE INTERACCIÓN

- ✎ Construcción de hilos sobre el comportamiento de los objetos en los casos de uso.
- ✎ Asignar el comportamiento de los objetos (Fronterizos, entidades y de control).
- ✎ Detallar la interacción entre los objetos (por medio de mensajes).
- ✎ Ubicar los métodos correspondientes a cada clase (responsabilidades).

2.7.1. Diagramas de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la Interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Un diagrama de secuencia consta de 4 elementos:

- ✎ El texto de caso de uso que muestra el orden del curso de acción en forma cronológica.
- ✎ Objetos.- se representan con el nombre del objeto (opcional) y la clase.
- ✎ Mensajes.- fechas entre los objetos.
- ✎ Métodos: Operaciones (objetos *De Control*) representados por rectángulos.

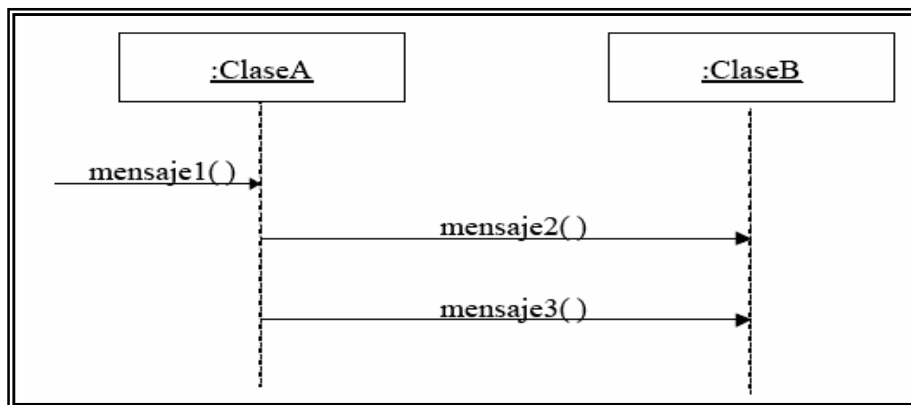


Figura 9: Diagrama de Secuencia

Como construir un diagrama de secuencia:

1. Revisar el texto descrito en el Caso de Uso.
2. Agregar objetos *entidad* del diagrama de robustez (parte superior - derecha).
3. Agregar objetos *fronterizos* y actores (parte superior - izquierda).
4. Asignar métodos y mensajes: Los objetos de control pasan a ser métodos de entidades o de objetos fronterizos (Responsabilidad).
5. Si un objeto de control se necesita, se agrega (Cuando sólo es intermediario sin actividad propia, se funde con fronterizo o entidad).

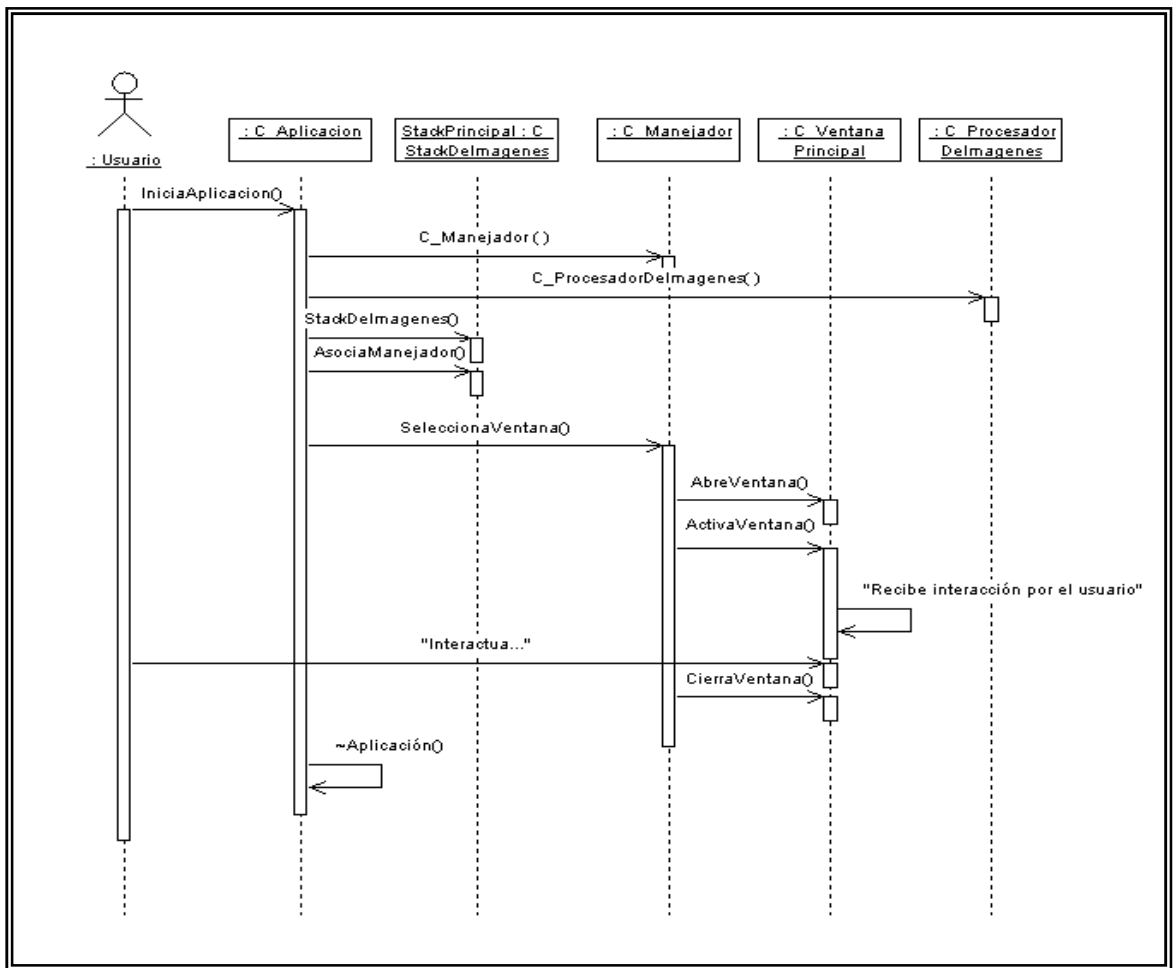
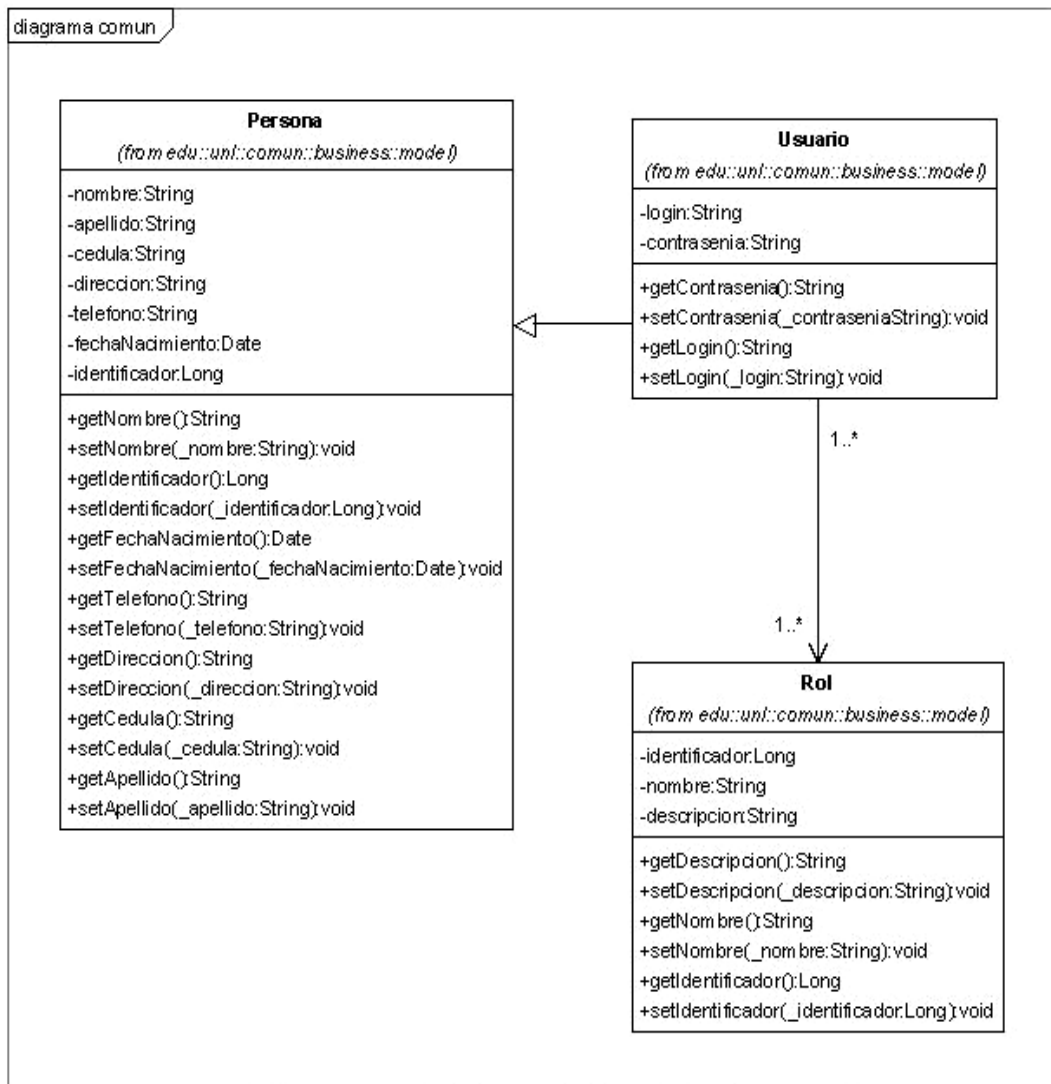


Figura 10: Diagrama de Secuencia Detallado

2.7.2. Complementar el diagrama de Clases.

De la misma manera que se cerró el ciclo de análisis, se termina la etapa de diseño con los diagramas de secuencia y con un diagrama de clases más completo que se presenta a continuación.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Figura 11: Diagrama de Clases

Al terminar la etapa de diseño, hemos refinado suficientemente el diagrama de clases y las relaciones entre estas. También conocemos mejor los mensajes que se intercambian los objetos para realizar las tareas necesarias. Durante la siguiente etapa nuevamente entraremos en otro ciclo para mejorar algunos detalles que se hacen visibles sólo al momento de la implementación.

2.8. IMPLEMENTACIÓN

La etapa que normalmente se espera más dentro de la creación de una aplicación es la de la implementación, sin embargo es importante haber llegado a un buen avance de las partes que la preceden para evitar 'improvisar' el código.

Normalmente no se debe crear una aplicación desde cero, se puede utilizar alguna herramienta que escriba el código a partir de los diagramas de clases (por ejemplo Visual Poseidón) y una herramienta gráfica para realizar ventanas como NetBeans, o bien un entorno de desarrollo que cree un 'esqueleto' de la aplicación.

Durante la etapa de implementación, se hizo énfasis en los siguientes puntos principalmente:

Crear clases en la 'punta' de la jerarquía para permitir paso de referencias a los objetos.

- ✍ Optimizar el despliegue de gráficos.
- ✍ Permitir el uso de múltiples formatos de imágenes.
- ✍ Aprovechar la estructura de datos en los lugares donde se requerían arreglos.

En la etapa de diseño definimos la estructura de clases que se acerca a lo que podemos llamar 'definitivo'. Sin embargo, existen algunos problemas que surgen en el momento de la implementación a nivel del paso de referencias a objetos.

2.8.1. Administración de Proyectos.

La administración del proyecto nos permite llevar de mejor manera la construcción del software deseado. Para ello se tomaron en cuenta las siguientes condiciones:

- ✍ Ser listo e ingenioso en todo el transcurso de la implementación del proyecto.
- ✍ No desconcentrarse y perder el enfoque en el proceso del proyecto.
- ✍ No utilizar herramientas visuales para generar pruebas tontas o simples del código.
- ✍ Apreciar más la calidad del código que la cantidad.

Revisar el modelo de dominio

En esta etapa se realizó:

- ✍ Finalización de los diagramas de secuencia y se refinará el modelo de dominio (Se verificó que los métodos sean concisos y atómicos).

A continuación:

- ✍ Se definió una lista de argumentos para las operaciones.
- ✍ Se definió operaciones lógicas
- ✍ Se asignó clases a componentes si es posible.

2.8.2. Pruebas.

- ✎ Para saber si el sistema es aceptable, se realizó pruebas de.
 - Caja Negra – casos de uso.
 - Prueba en tiempo real.

- ✎ Las pruebas se involucraron grupos lógicos (paquetes) de casos de uso, pruebas de unidad, de integración y de sistema.

CAPITULO III

HERRAMIENTAS DE DESARROLLO

3.1. LENGUAJE DE PROGRAMACIÓN JAVA.

Es probablemente uno de los mejores lenguajes de programación que podemos aprender hoy; por lo siguiente:

- ✍ Java puede correr en dispositivos portátiles gracias a la plataforma J2ME.
- ✍ Java es Orientado al Objeto.
- ✍ Java es “compatible” con Internet: Corre en navegadores, sus programas pueden ser transmitidos y correr en “cualquier” computador.
- ✍ Es de propósito general
- ✍ Es independiente de la plataforma
- ✍ Tiene muchas bibliotecas: para manejo gráfico, acceso a Internet, interfaces gráficas de usuarios (GUI's) etc.
- ✍ Surge por la necesidad de construir software para la electrónica de consumo (VCR's, TV, teléfonos,...) La dificultad de C y C++ fue el transporte a otros procesadores. Objetivo: crear programas pequeños, rápidos, transportables.

Los programas Java no son ejecutables, no se compilan como los programas en C o C++. En su lugar son interpretados por una aplicación conocida como la máquina virtual de Java (JVM). Gracias a ello no tienen que incluir todo el código y librerías propias de cada sistema.

3.2. LIBRERÍA JEXCEL (JXL.JAR)

La Librería JExcel es una poderosa herramienta, muy útil para manejar los parámetros y valores de la conocida hoja Excel desde un programa Java. La amplia difusión de este tipo de documento, hacen de Excel uno de los formatos mas reconocidos de intercambio de información de datos en todas las áreas, desde el ámbito laboral, hasta el ámbito educacional.

Utilizando las potencialidades de Java, se pueden mezclar el uso de bases de datos, cuyo uso tiene una poca justificada utilización, con la presentación amigable y conocida de Excel por medio de una aplicación JAVA¹⁹.

Así mismo se pueden diseñar entornos gráficos para Excel que hagan más amigable la presentación de datos, o la introducción de los mismos, usando las conocidas herramientas gráficas de Java.

Requerimientos para el desarrollo utilizando JExcel

Para desarrollar aplicaciones se requiere, primero que todo, tener instalado la máquina virtual Java, Java Runtime Environment. Si no se tiene, se puede descargar desde: http://java.com/en/download/windows_xpi.jsp.

Se requiere también por supuesto, la plataforma de desarrollo de Java, Java 2 Platform, Standard Edition (J2SE), que está disponible para su descarga desde <http://java.sun.com/>. En la página también está la opción de descargar la herramienta netbeans integrada con el J2SE. Se recomienda instalar esta aplicación en esta distribución si se desean hacer aplicaciones java, pero se recomienda también que si

¹⁹ <http://profesores.elo.utfsm.cl/~agv/elo330/2s05/projects/CeronSilva/index.htm>

se quiere trabajar con aplicaciones en servidores, posterguen su descarga para descargar la versión integrada con el Sun Java System Application Server.

El siguiente código permite leer el archivo de Excel:

```
private Workbook workbook;
public Read() {
    try {
        this.workbook = Workbook.getWorkbook(new File("read.xls"));
        //podemos escoger la ubicación del archivo en un lugar del equipo.

    } catch (java.io.IOException ioe) {
        System.out.println("No se pudo abrir el archivo");
    } catch (jxl.read.biff.BiffException jrbe) {
        System.out.println("No se pudo crear el libro");
    }
}
```

Con el siguiente código leemos cada campo del archivo:

```
Sheet sheet = workbook.getSheet(0);
Cell a1 = sheet.getCell(0,0);
Cell b2 = sheet.getCell(1,1);
Cell c2 = sheet.getCell(2,2);

String stringa1 = a1.getContents();
String stringb2 = b2.getContents();
String stringc2 = c2.getContents();
```

Ejemplo: Leer un archivo de Excel:

Clase Principal de Java

```
/*
 * Main.java
 */

public class Main {
    /** Creates a new instance of Main */
    public Main() {
    }
    public static void main(String[] args) {
        new Read();
    }
}
```

Clase Para leer las filas y columnas del Archivo Excel

```
/*
 * Read.java
 */

package excel;
```

```

import java.io.File;
import java.util.Date;
import jxl.Cell;
import jxl.CellType;
import jxl.DateCell;
import jxl.LabelCell;
import jxl.NumberCell;
import jxl.Sheet;
import jxl.Workbook;

public class Read {
    private Workbook workbook;
    public Read() {
        try {
            this.workbook = Workbook.getWorkbook(new File("read.xls"));
        } catch (java.io.IOException ioe) {
            System.out.println("No se pudo abrir el archivo");
        } catch (jxl.read.biff.BiffException jrbe) {
            System.out.println("No se pudo crear el libro");
        }
    }

    public Read(String file) {
        try {
            this.workbook = Workbook.getWorkbook(new File(file));
        } catch (java.io.IOException ioe) {
            System.out.println("No se pudo abrir el archivo");
        } catch (jxl.read.biff.BiffException jrbe) {
            System.out.println("No se pudo crear el libro");
        }
    }

    public void readBook() {
        Sheet sheet = workbook.getSheet(0);
        Cell a1 = sheet.getCell(0,0);
        Cell b2 = sheet.getCell(1,1);
        Cell c2 = sheet.getCell(2,2);

        String stringa1 = a1.getContents();
        String stringb2 = b2.getContents();
        String stringc2 = c2.getContents();

        workbook.close();

        System.out.println(stringa1 + "\n" + stringb2 + "\n" + stringc2);
    }

    public void readType() {
        String string = null;
        double number = 0;
        Date date = null;

        Sheet sheet = workbook.getSheet(0);
        Cell a1 = sheet.getCell(0,0);
        Cell b2 = sheet.getCell(1,1);
        Cell c2 = sheet.getCell(2,2);
    }
}

```

```

    if (a1.getType() == CellType.LABEL) {
        LabelCell lc = (LabelCell) a1;
        string = lc.getString();
    }

    if (b2.getType() == CellType.NUMBER) {
        NumberCell nc = (NumberCell) b2;
        number = nc.getValue();
    }

    if (c2.getType() == CellType.DATE){
        DateCell dc = (DateCell) c2;
        date = dc.getDate();
    }

    workbook.close();
}

public static void main (String[] args) {

    if (args.length == 0) {
        Read book = new Read();
        book.readBook();
    }
    else if (args.length == 1) {
        Read book = new Read(args[0]);
        book.readType();
    }
    else {
        System.out.println("USSAGE:  java Read");
        System.out.println("        java Read <file.xls>");
        System.out.println("");
        System.out.println("by default file = read.xls");
        System.out.println("");
    }
}
}
}

```

3.3. JGOODIES

JGoodies es un proyecto de desarrollo JAVA, que facilita la conexión con de los diferentes elementos swing con bases de datos. Este proyecto posee soporte mundial es decir cientos de programadores en el mundo participan de este proyecto.

El proyecto JGoodies posee varias librerías que pueden ser utilizadas en Java como:

- Validation
- Binding
- Looks
- Forms
- Imagen

3.3.1. Librería Binding

El *framework binding* actúa como un intermediario entre la vista y el modelo de aplicación. Usa las mismas técnicas en ambos lados, conecta los valores en el lado del modelo usando oyentes de cambio de propiedad (property change listeners), y en la vista adaptadores de presentación del modelo (presentation model). La diferencia es que usa un objeto (o cadena de objetos) entre ellos como un “traductor universal” para ambos lados. Binding crea un solo intermediario (la interface `com.jgoodies.binding.value.ValueModel`), ambos lados actúan recíprocamente con él de una manera común.

La meta principal del *framework binding* es simplificar y unificar el proceso de conexión de los componentes visuales (como los botones) con los valores en su modelo de dominio de la aplicación.

El *framework binding* intenta resolver el problema de ligamiento de los componentes con los objetos de la aplicación. Típicamente si usted quisiera recibir los valores de un GUI que controle un objeto de la aplicación, usted usa un oyente (como un `ActionListener` de un checkbox). Lo mismo se puede realizar para conseguir los cambios de un objeto de dominio controlado por un GUI. Típicamente allí se usaría un `PropertyChangeListener` o un Observador. De una u otra manera, la meta es que los cambios en un lado sean propagados hacia el otro lado. Este modelo se lo resuelve típicamente con múltiples oyentes por cada propiedad. Cuando se desarrolla *beans* más complicados, éste es un proceso difícil. Si, a veces únicamente sólo quiere mover

los valores en una sola dirección o cuando el usuario realiza una acción en particular (como apretar un botón “Guardar” o un botón “Recargar”).

A continuación se muestra como enlazar una propiedad de un bean de tipo boolean a un checkbox.

```
// Example Bean... MyBean.java
import java.beans.PropertyChangeListener;
import com.jgoodies.binding.beans.*;
public class MyBean {
    private ExtendedPropertyChangeSupport changeSupport = new
ExtendedPropertyChangeSupport(this);
    private boolean booleanValue;
    public boolean getBooleanValue() {
        return booleanValue;
    }
    public void setBooleanValue(boolean newValue) {
        boolean oldValue = booleanValue;
        booleanValue = newValue;
        changeSupport.firePropertyChange("booleanValue", oldValue, newValue);
    }
    public void addPropertyChangeListener(PropertyChangeListener x) {
        changeSupport.addPropertyChangeListener(x);
    }
    public void removePropertyChangeListener(PropertyChangeListener x) {
        changeSupport.removePropertyChangeListener(x);
    }
}
}
```

```
// Example test class - BindingTest.java
import javax.swing.*;
import com.jgoodies.binding.adapter.BasicComponentFactory;
import com.jgoodies.binding.beans.PropertyAdapter;

public class BindingTest {
    public static void main(String[] args) {
        MyBean bean = new MyBean();
        PropertyAdapter adapter = new PropertyAdapter(bean, "booleanValue", true);
        JCheckBox box = BasicComponentFactory.createCheckBox(adapter, "Boolean Value");
        JFrame frame = new JFrame();
        frame.getContentPane().add(box);
        frame.pack();
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

En el bloque anterior de código, se crea un adaptador de propiedad (property adapter) para la propiedad que nosotros queremos conectar, creamos un componente para ese adaptador de propiedad, y se lo coloca en el GUI. Ahora, cada vez que el checkbox sea activada o desactivada, el objeto "bean" se actualizará automáticamente, e igualmente, el objeto del bean se actualizará todo el tiempo.

Además del API de PropertyAdapter, existe también el API de BeanAdapter que es más eficaz al tratar con propiedades múltiples en un solo bean. A continuación un ejemplo extendido usando el adaptador del bean (bean adapter):

```
// Example Bean... MyBean.java
import java.beans.PropertyChangeListener;
import com.jgoodies.binding.beans.*;
public class MyBean {
    private ExtendedPropertyChangeSupport changeSupport = new
    ExtendedPropertyChangeSupport(
        this);
    private boolean booleanValue;
    private String stringValue;
    public boolean getBooleanValue() {
        return booleanValue;
    }
    public void setBooleanValue(boolean newValue) {
        System.out.println("Boolean value set: " +newValue);
        boolean oldValue = booleanValue;
        booleanValue = newValue;
        changeSupport.firePropertyChange("booleanValue", oldValue, newValue);
    }
    public void addPropertyChangeListener(PropertyChangeListener x) {
        changeSupport.addPropertyChangeListener(x);
    }
    public void removePropertyChangeListener(PropertyChangeListener x) {
        changeSupport.removePropertyChangeListener(x);
    }
    public String getStringValue() {
        return stringValue;
    }
    public void setStringValue(String newValue) {
        System.out.println("String value set: "+ newValue);
        String oldValue = stringValue;
        this.stringValue = newValue;
        changeSupport.firePropertyChange("stringValue", oldValue, newValue);
    }
}

// Example test class - BindingTest.java
import java.awt.GridLayout;
import javax.swing.*;
import com.jgoodies.binding.adapter.BasicComponentFactory;
```

```

import com.jgoodies.binding.beans.BeanAdapter;
import com.jgoodies.binding.value.ValueModel;
public class BindingTest {
    public static void main(String[] args) {
        MyBean bean = new MyBean();
        BeanAdapter adapter = new BeanAdapter(bean, true);
        ValueModel booleanModel = adapter.getValueModel("booleanValue");
        ValueModel stringModel = adapter.getValueModel("stringValue");
        // creates a JCheckBox with the property adapter providing the underlying model.
        JCheckBox box = BasicComponentFactory.createCheckBox(booleanModel, "Boolean Value");
        JTextField field = BasicComponentFactory.createTextField(stringModel);
        JFrame frame = new JFrame();
        frame.getContentPane().setLayout(new GridLayout(2,1));
        frame.getContentPane().add(box);
        frame.getContentPane().add(field);
        frame.pack();
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        frame.setVisible(true);
    }
}

```

En este ejemplo se agrega las declaraciones de impresión al bean sólo para visualizar como trabaja el *framework binding*.

La razón por la que planteamos esto, es porque puede ser importante para usted ligar los beans y los GUI ya construidos. Si una aplicación requiere usar su propio GUI, usted puede usar la clase Binding para enlazar a algunos modelos.

3.4. HERRAMIENTA PARA LA PERSISTENCIA DE DATOS: HIBÉRNATE

Para la mayoría de las aplicaciones, almacenar y recuperar información implica alguna forma de interacción con una base de datos relacional.

Las bases de datos relacionales están estructuradas en una configuración tabular y los ejemplares orientados a objetos normalmente están relacionados en forma de árbol. Esta “diferencia de impedancia” ha llevado a los desarrolladores de varias tecnologías de persistencia de objetos a intentar construir un puente entre el mundo relacional y el mundo orientado a objetos.

Persistencia de Objetos

La tarea de persistir objetos Java en una base de datos relacional actualmente está siendo facilitada por un gran número de herramientas que permiten a los desarrolladores, dirigir motores de persistencia para convertir objetos Java a columnas/registros de una base de datos y viceversa. Esta tarea implica serializar objetos Java, estructurados en forma de árbol a una base de datos relacional estructurada de forma tabular y viceversa. Esencial para este esfuerzo es la necesidad de mapear los objetos Java a columnas y registros de la base de datos de una manera optimizada en cuanto a velocidad y eficiencia.

El marco de trabajo Hibernate se enfrenta al problema "objeto-java-a-base-de-datos" de forma tan elegante como cualquier otro marco de trabajo disponible. Hibernate funciona persistiendo y restaurando viejos objetos Java (POJOs) utilizando un modelo de programación muy transparente y poco exigente.

Introducción a Hibernate

Hibernate es un marco de trabajo Java que proporciona mecanismos de mapeo objeto/relacional para definir cómo se almacenan, eliminan, actualizan y recuperan los objetos Java. Además, Hibernate ofrece servicios de consulta y recuperación que pueden optimizar los esfuerzos de desarrollo dentro de entornos SQL y JDBC. Por último, Hibernate reduce el esfuerzo necesario para convertir hojas de resultados de la base de datos relacional en gráficos de objetos Java.

Una de las características únicas de Hibernate es que no requiere que los desarrolladores implementen interfaces propietarios o extiendan clases base propietarias para poder persistir las clases. En vez de eso, Hibernate trata con la *reflection* de Java y el aumento de clases en tiempo de ejecución utilizando una librería de generación de código Java muy poderosa y de alto rendimiento llamada

CGLIB. CGLIB se utiliza para extender clases Java e implementar interfaces Java en tiempo de ejecución.

El Fichero de Configuración de Hibernate

Se puede configurar el entorno Hibernate en un par de formas. Una forma estándar que se declara como muy flexible y conveniente es almacenar la configuración en un fichero llamado hibernate.cfg.xml. Este fichero se sitúa en la raíz del classpath del contexto de la aplicación (por ejemplo: src/clases). Se puede acceder a este fichero utilizando la clase net.sf.hibernate.cfg.Configuration en tiempo de ejecución.

El fichero hibernate.cfg.xml define la información sobre la conexión a la base de datos, la clase factoría de transacciones, los recursos de mapeo, etc. El siguiente código muestra una configuración típica de este fichero:

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 2.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">
<hibernate-configuration>
  <session-factory>

    <property name="connection.driver_class"> com.mysql.jdbc.Driver</property>
    <property name="connection.url"> jdbc:mysql://localhost/financier</property>
    <property name="connection.username">root</property>
    <property name="connection.password">root</property>
    <property name="show_sql">true</property>
    <property name="hbm2ddl.auto">update</property>
    <property name="cglib.use_reflection_optimizer">true</property>
    <property name="dialect">org.hibernate.dialect.MySQLInnoDBDialect</property>
    <property name="transaction.factory_class">
      net.sf.hibernate.transaction.JDBCTransactionFactory
    </property>
    <property name="current_session_context_class">thread</property>

    <!-- common mapping files -->
    <mapping resource="edu/unl/comun/data/mappings/Rol.hbm.xml"/>
    <mapping resource="edu/unl/comun/data/mappings/Persona.hbm.xml"/>
    <mapping resource="edu/unl/financiero/data/mappings/Prestamo.hbm.xml"/>
    <mapping resource="edu/unl/financiero/data/mappings/AporteMensual.hbm.xml"/>
    <mapping resource="edu/unl/financiero/data/mappings/PagoPrestamo.hbm.xml"/>
    <mapping resource="edu/unl/financiero/data/mappings/Cuenta.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

```

<mapping resource="edu/unl/financiero/data/mappings/DetallePrestamo.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/EstadoDeCuenta.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/PagoDeSueldo.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/Retiro.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/Servicio.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/Socio.hbm.xml"/>
<mapping resource="edu/unl/financiero/data/mappings/Cheque.hbm.xml"/>
</session-factory>
</hibernate-configuration>

```

El Archivo de Configuración de Mapeo de Hibernate

Las aplicaciones Hibernate hacen uso de ficheros de mapeo que contienen metadatos que definen los mapeos objeto/relacional para las clases Java. Un fichero de mapeo tiene el sufijo .hbm.xml. Dentro de cada fichero de configuración, se mapean a tablas de la base de datos las clases que se van a persistir y las propiedades se definen con mapeos de campo/columna y claves primarias. El siguiente código ilustra un típico fichero de configuración de Hibernate llamado UserInfo.hbm.xml:

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping package="edu.unl.financiero.business.model">
  <class name="Cuenta" table="Cuenta">
    <id name="identificador" type="java.lang.Long">
      <column name="idCuenta" not-null="true" sql-type="int(7)"/>
      <generator class="native"/>
    </id>
    <many-to-one name="socio" column="idSocio" not-null="true"/>
    <set name="estadoDeCuenta" table="estadoDeCuentaCuenta" inverse="true" lazy="true"
    cascade="save-update">
      <key column="idCuenta"/>
      <many-to-many column="idEstadoDeCuenta" class="EstadoDeCuenta"/>
    </set>
    <property name="numeroDeCuenta" type="java.lang.String"
    column="numeroDeCuenta" />
    <property name="fechaApertura" type="java.util.Date" column="fechaApertura"/>
    <property name="interes" type="java.lang.Double" column="interes"/>
    <property name="tipo" type="java.lang.String" column="tipo" />
    <property name="descripcion" type="java.lang.String" column="descripcion" />

    <joined-subclass name="CuentaDeCesantia" table="CuentaDeCesantia"
    extends="edu.unl.financiero.business.model.Cuenta">
      <key>
        <column name="idCuentaDeCesantia" sql-type="int(7)" not-null="true"/>
      </key>
    </joined-subclass>

```

```

    <joined-subclass name="CuentaDeJubilacion" table="CuentaDeJubilacion"
extends="edu.unl.financiero.business.model.Cuenta">
    <key>
    <column name="idCuentaDeJubilacion" sql-type="int(7)" not-null="true"/>
    </key>
    </joined-subclass>

</class>
</hibernate-mapping>

```

Sesiones Hibernate

Para poder utilizar los mecanismos de persistencia de Hibernate se debe inicializar el entorno Hibernate y obtener un objeto Session utilizando la clase SessionFactory de Hibernate. El siguiente fragmento de código ilustra este proceso:

```

// Initialize the Hibernate environment
Configuration cfg = new Configuration().configure();

// Create the session factory
SessionFactory factory = cfg.buildSessionFactory();

// Obtain the new session object
Session session = factory.openSession();

```

La llamada a Configuration().configure() carga el fichero de configuración hibernate.cfg.xml e inicializa el entorno de Hibernate. Una vez inicializada la configuración, se puede hacer cualquier modificación adicional de forma programática. Sin embargo, estas modificaciones se deben hacer antes de crear el ejemplar de SessionFactory.

Normalmente, el ejemplar de SessionFactory sólo se crea una vez y luego se utiliza para crear todas las sesiones relacionadas con un contexto dado.

Un objeto Session Hibernate representa una única unidad-de-trabajo para un almacén de datos dado y lo abre un ejemplar de SessionFactory. Se deben cerrar las sesiones cuando se haya completado todo el trabajo de una transacción. El siguiente código ilustra una sesión típica de Hibernate:

```
Session session = null;
UserInfo user = null;
Transaction tx = null;

try {
    session = factory.openSession();
    tx = session.beginTransaction();

    user = (UserInfo)session.load(UserInfo.class, id);

    tx.commit();
}
catch(Exception e) {
    if (tx != null) {
        try {
            tx.rollback();
        }
        catch (HibernateException e1) {
            throw new DAOException(e1.toString());
        }
    }
    throw new DAOException(e.toString());
}
finally {
    if (session != null) {
        try {
            session.close();
        }
        catch (HibernateException e) {
        }
    }
}
}
```

El Lenguaje de Consultas de Hibernate

Hibernate ofrece un lenguaje de consultas que agrupa un potente y flexible mecanismo de consulta, almacenamiento, actualización y recuperación de objetos desde una base de datos. Este lenguaje, el *Hibernate Query Language (HQL)*, es una extensión orientada a objetos de SQL. HQL permite acceder a los datos de varias formas,

incluyendo consultas orientadas a objetos, como en el método find() del siguiente ejemplo:

```
List users = session.find("from UserInfo as u where u.fullName = ?",
    "John Doe", Hibernate.STRING );
```

Se pueden construir consultas dinámicas utilizando el API criteria de Hibernate:

```
Criteria criteria = session.createCriteria(UserInfo.class);
criteria.add(Expression.eq("fullName", "John Doe"));
criteria.setMaxResults(20);
List users = criteria.list();
```

Si se prefiere, se puede utilizar SQL o expresar una consulta SQL, utilizando createSQLQuery():

```
List users = session.createSQLQuery("SELECT {user.*} FROM USERS AS {user}",
    "user", UserInfo.class).list();
```

Cuando se devuelven muchos objetos desde una consulta, los objetos serán cargados según se necesite utilizando uno de los métodos iterate(). Los métodos iterate() ofrecen un mejor rendimiento porque cargan los objetos bajo demanda:

```
Iterator iter = session.iterate("from UserInfo as u where u.city = New York");
while (iter.hasNext()) {
    UserInfo user = (UserInfo)iter.next();
    // process the user object here
}
```

3.5. HERRAMIENTA DE PROGRAMACIÓN VISUAL NETBEANS

Entorno gratuito de código abierto para la generación de código en diversos lenguajes (especialmente pensado para Java). Contiene prácticamente todo lo que se suele pedir a un IDE, editor avanzado de código, depurador, diversos lenguajes, extensiones de todo tipo (CORBA, Servlets,...). Incluye además un servidor de aplicaciones Tomcat para probar aplicaciones de servidor. Se descarga en www.netbeans.org.

El IDE NetBeans es un entorno de desarrollo - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso²⁰.

NetBeans es un entorno de desarrollo o IDE (integrated development environment) para todo tipo de tecnologías Java e incluso permite la codificación de programas en C, C++ y otros (aunque está pensado para Java).

Sus funciones son:

- ✎ Editor de código sensible al contenido. Con soporte para auto completar el código, coloreado de etiquetas, auto tabulación y uso de abreviaturas para varios lenguajes de programación.
- ✎ Soporte para Java, C, C++, XML y lenguajes HTML.
- ✎ Soporte para JSP, XML, RMI, CORBA, JINI, JDBC y tecnologías Servlet.
- ✎ Incluye CVS (control de versiones) y Ant (compilación avanzada)
- ✎ Posibilidad de utilizar otras versiones de compiladores, depuradores,...
- ✎ Creación visual de componentes gráficos.

²⁰ www.netbeans.org/products/ide/index.html

- ✎ Herramientas con asistentes para facilitar la escritura de código

3.6. BASE DE DATOS MYSQL

Las principales características de MySQL

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL.

- ✎ Interioridades y portabilidad
 - Escrito en C y en C++
 - Probado con un amplio rango de compiladores diferentes
 - Funciona en diferentes plataformas.
 - Usa GNU Automake, Autoconf, y Libtool para portabilidad.
 - APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
 - Proporciona sistemas de almacenamiento transaccional y no transaccional.
 - Un sistema de reserva de memoria muy rápido basado en threads.
 - El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- ✎ Tipos de Datos
 - Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT,

BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS.

- Registros de longitud fija y longitud variable.

Seguridad

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

Localización

- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
- Soporte completo para distintos conjuntos de caracteres, incluyendo latin1 (ISO-8859-1), german, big5, ujis, y más. Por ejemplo, los caracteres escandinavos 'â', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para Unicode está disponible.

Lo mejor de MySQL

“Es evidente que la gran mayoría de gente usa este gestor en Internet, por lo que encontrar opiniones favorables no ha resultado en absoluto complicado:

1. Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
3. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
4. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está²¹.

3.7. HERRAMIENTA VISUAL DE MODELAMIENTO UML. VISUAL POSEIDÓN-UML

“Poseidón para UML puede simplificar la compleja tarea de desarrollo de software ayudando a estructurar pensamientos, a clarificar la comunicación, y a encontrar la correcta abstracción. La incorrecta implantación de la herramienta UML, le sumergirá en detalles y diagramas llenos de funciones extrañas y excesivamente complicadas, lo que le evitara el ahorro de tiempo y esfuerzo.

Poseidón para UML evita estas distracciones. La intuitiva interfaz hace de Poseidón la herramienta más rápida de UML para dominar el análisis orientado a objetos, liberando al diseñador para centrarse solamente en su modelo. La abundancia de características disponibles realza la productividad desde el comienzo sin ser un obstáculo en el camino²².

Características Principales:

²¹ www.netpecos.org/docs/mysql_postgres/x56.html
²² http://es.wikipedia.org/wiki/Poseidon_for_UML

Entre las principales características de Poseidón UML tenemos:

Plataforma	JAVA
Licencia :	Libre
Diagramas que se pueden realizar:	De Clases, de Estado, de Actividad, de Casos de Uso, de Colaboración, Deployment (combinación de diagramas de Objetos/Componentes/Despliegue)
Generación de código	Genera a partir del diagrama de Clases.
Generación de documentación	JAVA, HTML
Apoyo al repositorio:	Esta en construcción
Ingeniería inversa:	SI
Navegación por el modelo:	SI (distintas perspectivas de navegación)
Exportación de diagramas:	GIF, PostScript, EPS, PGML, SVG
Métricas:	SI
XMI	SI
Otras características	Control de errores (checklist) automático

IREPORT

“La herramienta iReport es un constructor/diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc.

iReport está además integrado con JFreeChart, una de las bibliotecas gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, etc.”²³.

iReport es una herramienta visual que sirve para generar ficheros XML (plantillas de informe) que se puedan utilizar con la herramienta de generación de informes JasperReports.

iReport inicialmente se ha desarrollado en J++ por lo que su entorno de ejecución se limita a plataformas Microsoft. En sus últimas versiones se ha publicado una versión 100% java con lo cual se limita la restricción existente.

Características de iReport

La lista siguiente describe algunas de las características importantes de iReport:

- ✍ 100% escrito en JAVA y además OPENSOURCE y gratuito.
- ✍ Maneja el 98% de las librerías y etiquetas de JasperReports
- ✍ Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports (subreportes).
- ✍ Soporta internacionalización nativamente.
- ✍ Browser de la estructura del documento.
- ✍ Recopilador y exportador integrados.
- ✍ Soporta JDBC.
- ✍ Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).

²³

<http://www.adictosaltrabajo.com/tutoriales/retornapdf.php?pdf=ireport>

- ✎ Incluye Wizard's (asistentes) para crear automáticamente informes.
- ✎ Tiene asistentes para generar los subreportes.
- ✎ Tiene asistentes para las plantillas.
- ✎ Facilidad de instalación.

JasperReports

JasperReports, es una poderosa herramienta para generar reportes en java, con la habilidad de producir contenido completo para la pantalla, directo para impresora o en diferentes formatos de archivo (PDF, XLS, CSV, XML, entre otros).

La librería es 100% Java, y puede reutilizarse tanto en aplicaciones cliente/servidor, como en aplicaciones web, J2EE, etc. Permite organizar la información obtenida desde una base de datos relacional, a través de conectores JDBC, Hibernate, etc., en diseños de reportes predefinidos en un formato XML.

La estructura de paquetes y las clases más importantes que utilizan iReport y JasperReports son:

- ✎ *net.sf.jasperreports.engine.JasperExportManager*
- ✎ *net.sf.jasperreports.engine.JasperPrintManager*
- ✎ *net.sf.jasperreports.engine.JasperFillManager*
- ✎ *net.sf.jasperreports.engine.JasperCompileManager*
- ✎ *net.sf.jasperreports.view.JasperViewer*
- ✎ *net.sf.jasperreports.view.JasperDesignViewer*

Estas clases presentan una abstracción para facilitar la vista del reporte, el diseño, la impresión, el llenado, la compilación, etc. Las primeras cuatro permiten manejar el motor de generación del reporte, y las últimas dos (las pertenecientes al paquete view), la visualización de los reportes.

CAPITULO IV

PRESENTACIÓN DE RESULTADOS

REQUERIMIENTOS DEL SISTEMA

Determinación de Requerimientos del sistema.

Es la parte más importante dentro del desarrollo de sistemas, para la ejecución del presente *SISTEMA FINANCIERO CONTABLE PARA EL FONDO DE JUBILACIÓN, CESANTÍA ADMINISTRATIVO*; se ha considerado los siguientes requerimientos.

REQUERIMIENTOS

REQUERIMIENTOS FUNCIONALES

CÓDIGO	REQUERIMIENTO
RQ01	El sistema debe permitir mantener una Base de Datos con toda la Información Personal de los usuarios
RQ02	El sistema debe permitir asignar cuentas de aportaciones para los usuarios de Fondo del Cesantía.
RQ03	El sistema debe permitir buscar un socio por nombre, numero de cuenta, cedula.
RQ04	Asignar cuentas de aportaciones para los usuarios del Fondo de Jubilación.
RQ05	El sistema debe calcular los intereses por aportaciones tanto para el fondo de Cesantía como para el de Jubilación.
RQ06	El sistema debe diferenciar socios activos y socios pasivos.
RQ07	El sistema debe registrar los préstamos realizados por los usuarios.
RQ08	El sistema debe calcular dos tipos de amortizaciones según elija el usuario: <ul style="list-style-type: none">- Amortizaciones de interés fijo.- Calcular los intereses para todos los meses iguales.- Amortizaciones sobre el saldo.- Calcular los interés sobre el saldo actual.
RQ09	El sistema debe permitir la actualización de intereses para las aportaciones y para préstamos.
RQ10	El sistema debe calcular el 10% de aporte personal del socio a partir del sueldo del trabajador administrativo que se obtiene en una cinta magnética.
RQ11	El sistema debe asignar el 10% de aporte patronal para el fondo de jubilación de manera personal y grupal.
RQ12	El sistema debe calcular del 10% de aporte personal; un 9% para el fondo de cesantía y el 1% para el fondo de jubilación del trabajador administrativo.

RQ13	<p>El sistema debe de permitir al usuario escoger los montos de los préstamos que están presupuestados dentro del fondo administrativo de acuerdo a la siguiente tabla</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>MONTO</th> <th>PLAZO (MESES)</th> </tr> </thead> <tbody> <tr> <td>6000</td> <td>48</td> </tr> <tr> <td>5000</td> <td>42</td> </tr> <tr> <td>4000</td> <td>36</td> </tr> <tr> <td>2500</td> <td>18</td> </tr> </tbody> </table>	MONTO	PLAZO (MESES)	6000	48	5000	42	4000	36	2500	18
MONTO	PLAZO (MESES)										
6000	48										
5000	42										
4000	36										
2500	18										
RQ14	El sistema debe imprimir reportes de las actividades realizadas en el día, así como también la tabla de amortizaciones de los usuarios del Fondo Administrativo.										
RQ15	El sistema debe calcular el 50% del total de los aportes realizados por cada Socio para el retiro de fondos de cesantía y restringir este retiro cada 8 años o cada 36 aportaciones mensuales por Socio.										
RQ16	<p>El sistema debe de calcular el porcentaje total del retiro aplicando la siguiente fórmula:</p> $(\text{Capital} + \text{Rentabilidad}) - 10\% \text{ Rentabilidad} = \text{Total Retiro}$										
RQ17	<p>Calcular el retiro de los fondos de jubilación de acuerdo a la siguiente reglamentación:</p> <ul style="list-style-type: none"> - Jubilación General: Mínimo 360 aportaciones y ser mayor de 65 años. - Jubilación Reducida: Mínimo 300 aportaciones. - Jubilación especial (accidente): 30% de los 6 últimos sueldos. 										
RQ18	El sistema debe registrar el servicio de pago de sueldos a los socios.										
RQ19	El sistema debe registrar el pago de sueldos a partir de una cinta magnética.										
RQ20	El sistema debe llevar la Contabilidad General.										
RQ21	El sistema debe ingresar actividades contables diarias.										
RQ22	Generar Mayores auxiliares, Balances de sumas y saldos, Balances de resultados.										
RQ23	El sistema debe ingresar Auxiliar de bancos.										

RQ24	<p>El sistema debe generar reportes de los actividades contables realizadas como:</p> <ul style="list-style-type: none"> - Asientos Contables - Balances de resultados diarios - Balances de Resultados para cierre del periodo Económico. - Registros de asientos contables - Balance de comprobación. - Balance de pérdidas y ganancias.
------	--

REQUERIMIENTOS NO FUNCIONALES

CÓDIGO	REQUERIMIENTO
RQNF01	El sistema debe tener un ambiente amigable para el trabajo
RQNF02	Ingresar datos de Usuario.
RQNF03	El sistema a construir deberá correr en varios sistemas operativos como Windows y Linux
RQNF04	El sistema a construirse deberá poseer una arquitectura cliente servidor.
RQNF05	El sistema debe poseer pantallas amigables que le permita al usuario adaptarse fácilmente al trabajo.

GLOSARIO DE TÉRMINOS

GLOSARIO DE TÉRMINOS

TERMINO	DEFINICIÓN	CATEGORÍA
Usuario	Persona que haga uso del Sistema.	Actor
Administrador	Actor del Sistema que tiene los atributos de manejar todos los procesos del Sistema.	Actor
Socio	Persona que realiza aportaciones y mantiene una cuenta dentro del fondo de cesantía y jubilación.	Actor
Presidente	Persona que realiza labores administrativas del Fondo de Jubilación	Actor
Gerente	Persona que ejerce labores administrativas y de dirección del Fondo de Jubilación y Cesantía.	Actor
Secretaria	Persona que controla, verifica, ingresa y actualiza los datos personales de cada socio.	Actor
Contadora	Persona que realiza la contabilidad de la institución.	Actor
Auxiliar de Servicios	Persona que realiza y entrega tramites del fondo de Cesantía.	Actor
Fondo de Jubilación	Activos destinados para la jubilación de los socios de la institución.	Concepto
Fondo de Cesantía	Activos destinados para el pago de la cesantía de los socios de la institución.	Concepto
Servicios	Prestaciones o beneficios que recibe cada socio del fondo de jubilación y cesantía.	Concepto
Socio activo	Socio que realiza aportaciones mensuales para el fondo de cesantía y jubilación	Actor
Socio Pasivo	Socio que únicamente utiliza los servicios que presta el fondo.	Actor
Cuenta de Aportaciones	Registro de aportaciones que realiza cada socio del Fondo Administrativo.	Concepto
Préstamo	Proceso en el que un socio del Fondo Administrativo accede a un determinado monto de dinero, que posteriormente lo tiene que cancelar en cuotas mensuales.	Concepto
Aportación Mensual	Valor aportado mensualmente por cada socio destinado para los fondos de jubilación y cesantía.	Concepto
Rol de Pago	Registro de los salarios pagados al trabajador.	Concepto

Aporte Patronal	Aportación que realiza la Universidad por cada trabajador para el fondo de cesantía.	Concepto
Intereses de Amortización	Lista de valores y porcentajes mensuales cobrados por cada préstamos. Pueden ser en dividendos iguales o en intereses sobre el saldo pendiente.	Concepto
Cuenta de Aportaciones de Cesantía	Registro de aportaciones que realiza cada socio destinada para la cesantía.	Concepto
Cuenta de Aportaciones de Jubilación	Registro de aportaciones que realiza cada socio para la jubilación.	Concepto
Retiros	Es el retiro de los fondos de cesantía o de jubilación de cada socio.	Concepto
Estado de Cuenta	Valores de los aportes realizados por cada socio, donde consta: la fecha, No de aporte, valor aporte, interés, acumulado.	Concepto
Detalle de Préstamo	Tabla que registra los valores a pagar por cada préstamo realizado por cada socio donde constan: No cuenta, Apellidos y Nombres del Socio, No de cuota, dividendo y capital.	Concepto
Transacciones Diarias	Proceso contable de todas las transacciones diarias realizadas.	Concepto
Jubilación General	Es el retiro de la totalidad de los fondos de cada socio del fondo de jubilación, para ello cada socio debe cumplir con un mínimo de 360 aportaciones y ser mayor de 65 años de edad	Concepto
Jubilación Especial	Es el retiro de los fondos de cada socio del fondo de jubilación a la que puede acceder el socio en caso de accidente, es equivalente al 30% de los 6 últimos sueldos recibidos.	Concepto
Jubilación Reducida	Es el retiro de los fondos de cada socio del fondo de jubilación, para lo cual cada socio debe tener un mínimo de 300 aportaciones	Concepto
Balance de comprobación	Libro contable utilizado en la contabilidad de la institución en el cual se realiza la verificación de las sumas del debe y haber y saldos deudor y acreedor de las transacciones realizadas dentro de un determinado periodo de tiempo.	Concepto
Balance de Resultados	Libro contable utilizado en la contabilidad. Estos pueden ser mensuales.	Concepto

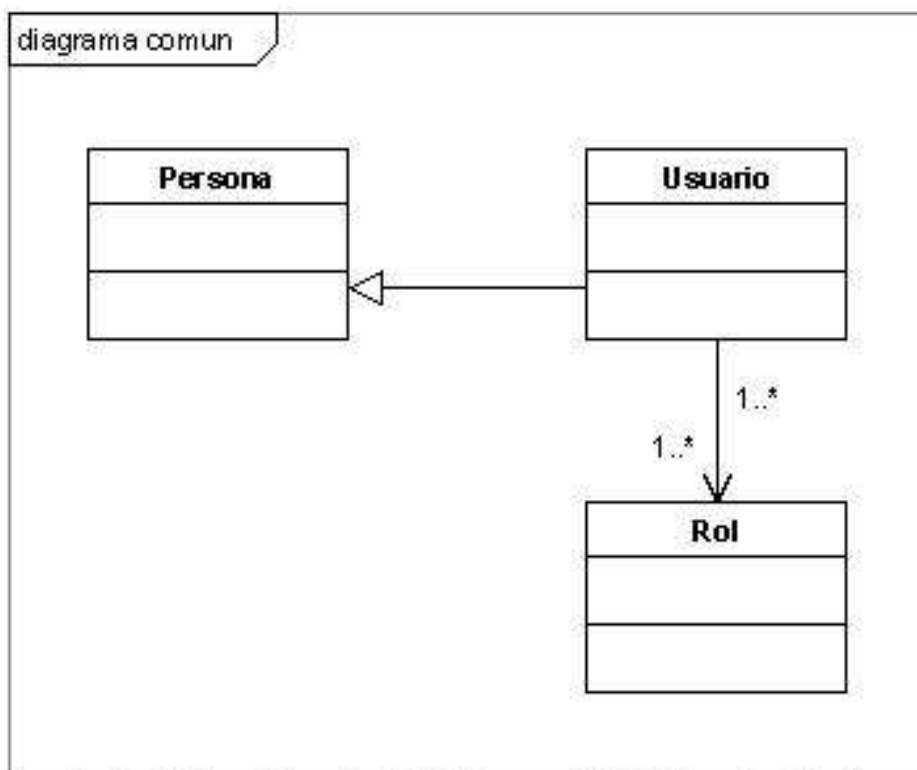
Balance de Situación Final	Proceso contable utilizado para la verificación de los valores de las transacciones realizadas en un periodo de un año completo.	Concepto
Auxiliares de Bancos	Proceso contable en el cual se registran las transacciones realizadas con instituciones bancarias.	Concepto
Asiento Contable	Detalle de la transacción que ha realizado de fondo administrativo.	Concepto
Estado de Situación Final	Libro de la contabilidad que permite saber el estado de pérdidas y ganancias de la empresa	Concepto
Libro Contable	Denominado Libro Contable a los diferentes tipos de formatos o rayados que se utiliza para llevar la contabilidad de la empresa.	Concepto
Cuenta Contable	Nombre que se asigna a un grupo de valores de una misma naturaleza.	Concepto
Balance Contable	Agrupación de las cuentas contables con sus respectivos saldos para determinar su total.	Concepto
Rol de Pagos	Listado de socios del fondo en la que constan los parámetros para el pago de su sueldo.	Concepto
Auxiliar de Bancos	Registro en la que constan los listados de los cheques que han sido emitidos por el fondo	Concepto
Inventario	Toda relación valorada y con detalle de los bienes que constituyen valores con que cuenta la empresa.	Concepto
Plan de Cuentas	Lista de las cuentas contables utilizadas dentro del fondo administrativo.	Concepto
Libro Diario	Registro riguroso y sistemático en orden cronológico las operaciones que se realizan dentro del fondo administrativo.	Concepto
Libro Mayor	Clasificador de de las anotaciones o asientos del libro diario, se lo realiza en las Cuentas Contables que se registra todas las operaciones de idéntica naturaleza o característica.	Concepto
Cuenta de Detalle	Cuenta en la que se pueden almacenar valores económicos	Concepto
Contiene Subcuentas	Es una cuenta Contable de la que se derivan varias Cuentas de Detalle	Concepto
Producto	Bien económico que debe constar dentro del inventario	Concepto
Cierre del Período	Se lo realiza a final de año con el resultado de toda la contabilidad realizada por el fondo administrativo.	Concepto
Activo	Todo lo que posee o esté a favor del fondo administrativo.	Concepto

Pasivo	Todas las deudas con terceras personas del fondo administrativo	Concepto
Capital	El valor del derecho del propietario, ó su inversión en el negocio, se conoce como Capital o patrimonio.	Concepto
Fecha Final de cierre	Fecha final en que se realizarán los balances como: Balance general, Balance de Comprobación, de las operaciones contables del Fondo Administrativo	Atributo
Fecha de Inicio	Fecha de inicio de las operaciones de un periodo contable	Atributo
Fecha Final	Fecha a la que se cierra un periodo contable	Atributo
Tipo de Jubilación	Opciones que tiene el socio del Fondo para poder retirar su jubilación.	Concepto
Período Contable	Libro contable que posee una fecha de inicio y una fecha de cierre	Concepto

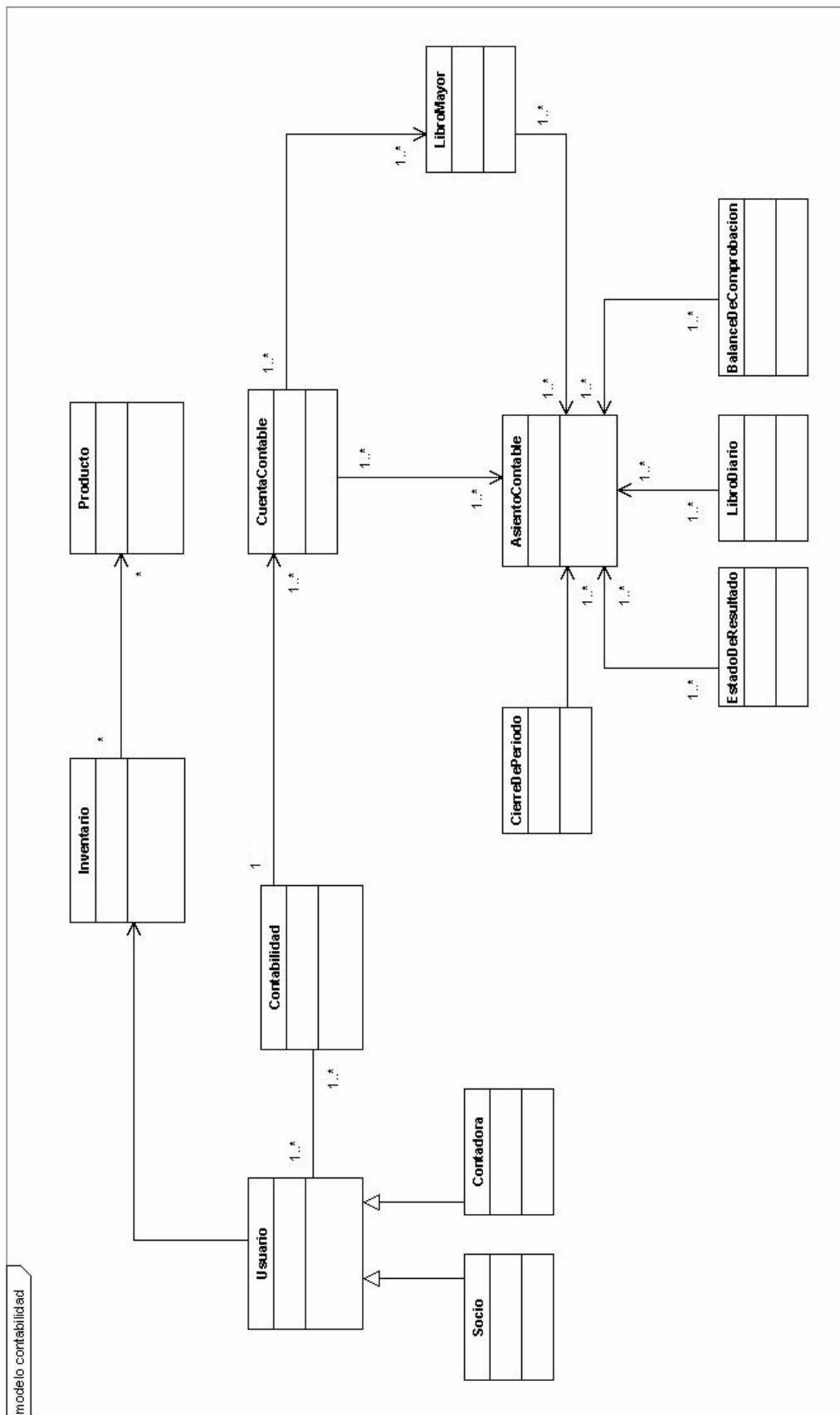
MODELO DE DOMINIO

Modelo del dominio del Sistema

El modelo del dominio se dividió en tres partes para mejorar su entendimiento, a continuación se describen los modelos.



Created with Poseidon for UML Community Edition. Not for Commercial Use.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

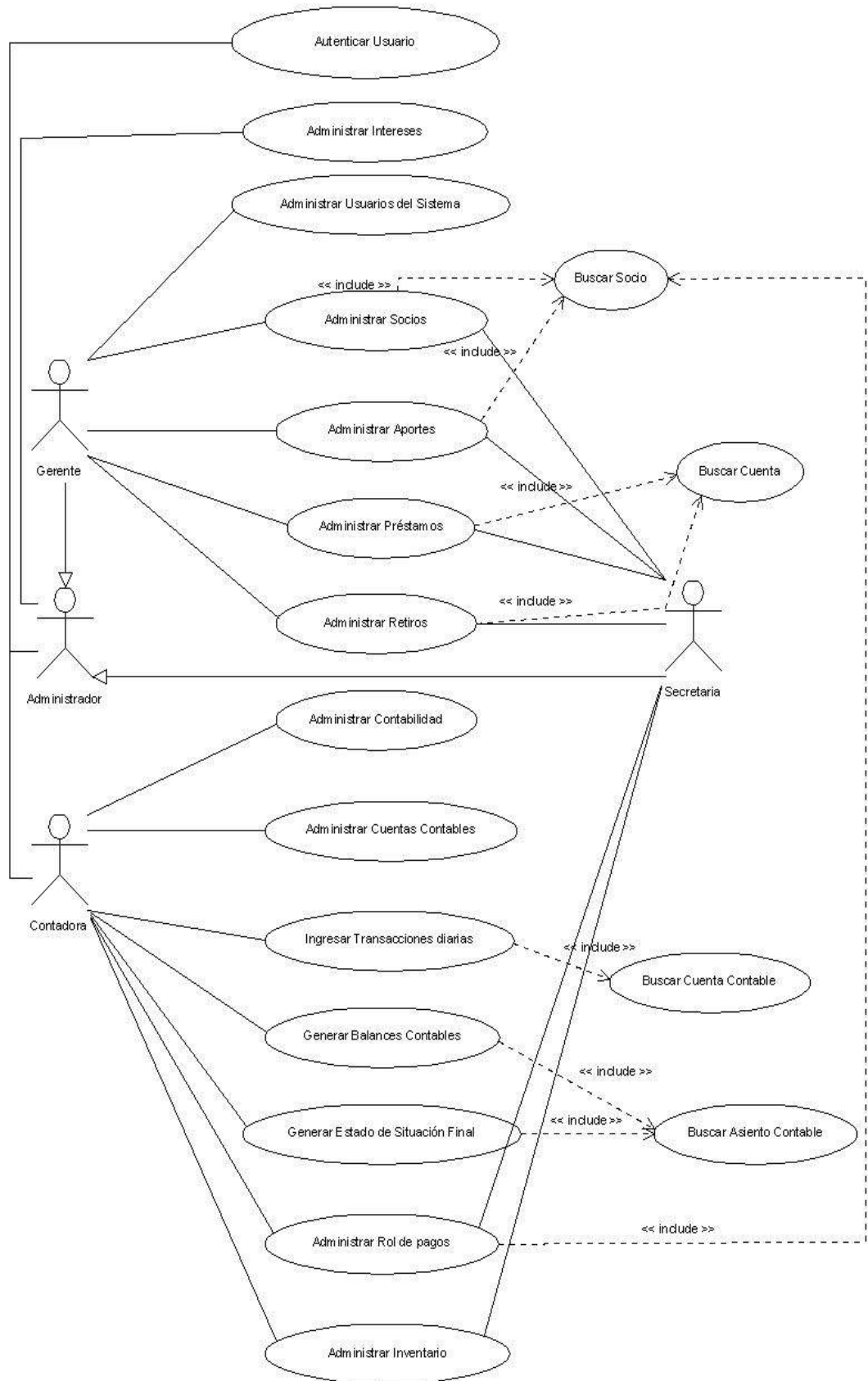
Identificar Use cases y Esquematizar Diagramas Use cases

El actor representa el rol genérico de usuario del sistema. El nombre que se le dé a un actor deberá reflejar el papel que tendrá para el sistema.

Los use cases son una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Reflejan el uso que harán los actores del sistema; se muestran a través de ellos tanto las funcionalidades que ofrecerá el sistema, como los diferentes comportamientos posibles inherentes a las situaciones contempladas para cada una de estas.

DIAGRAMA DE CASOS DE USO

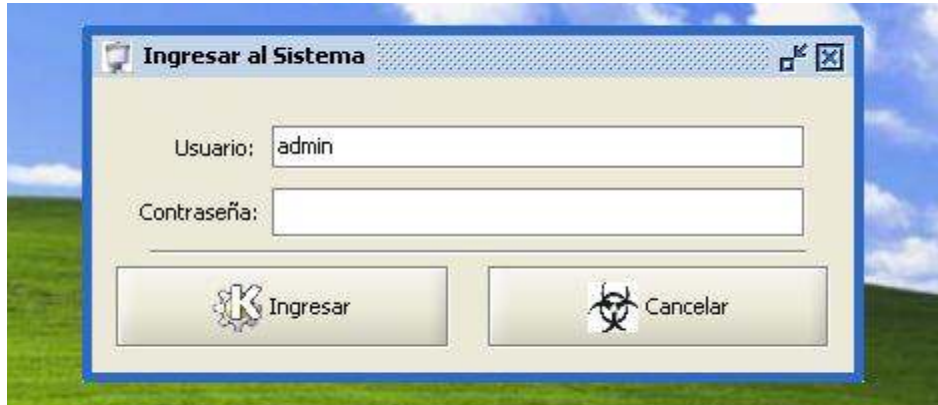
Diagram de Casos de Uso



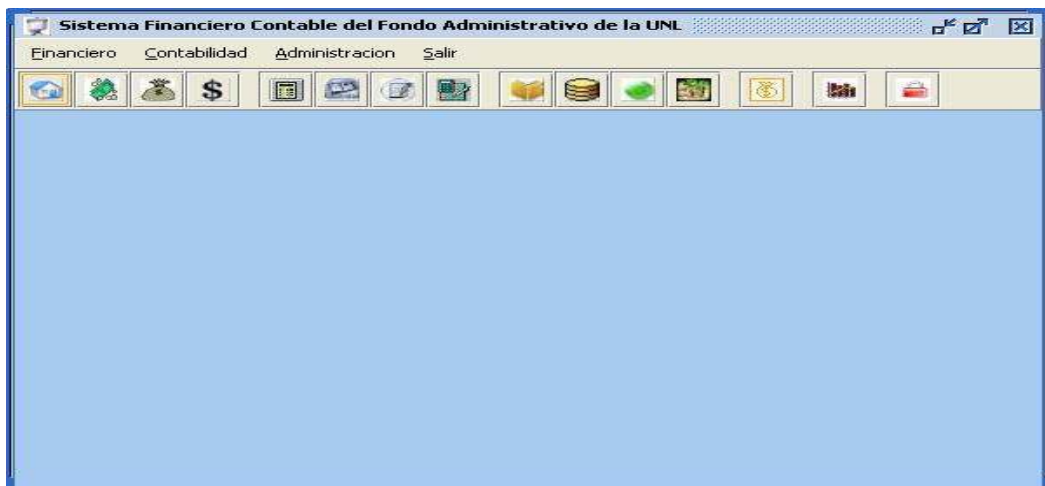
PROTOTIPOS

Caso de Uso: Administrar Socio

Ventana Ingresar al Sistema



Ventana principal.



Menú Financiero.



Ventana Administrar Socios.

Administrar Socios

Búsqueda del Socio

Criterio de Búsqueda:

Opcion:

Lista de Socios:

Apellidos	Nombres	Cédula
arias arias	elizabeth	1103805569

Administración:

Ventana Ingresar Nuevo Socio.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administración Salir

Ingresar nuevo socio

Datos socio

Fecha: mar, 28 nov 2006

Datos del socio:

Apellidos: Benitez Jara

Nombres: Maria Beatriz

Cargo: Auxiliar

Cedula: 1103985121

Area Academica: Educación Arte y Comunicación

Dirección: Ciudadela Daniel Alvarez no 854

Fecha de Nacimiento: 28/11/1970

Años de servicio: 6

Telefono: 2589654

Sueldo Mensual: 715.00

Nro Nombramiento: E-213123

Ventana Modificar Socio: Ventana para modificar los datos del socio

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administración Salir

Modificar socio

Datos socio

Fecha: mar, 28 nov 2006

Datos del socio:

Apellidos: Alvarez Intriago

Nombres: Nestor Fernando

Cargo: estudiante

Cedula: 1715544421

Area Academica: Energia

Dirección: Av de los Paltas y paraguay

Fecha de Nacimiento: 25/07/1979

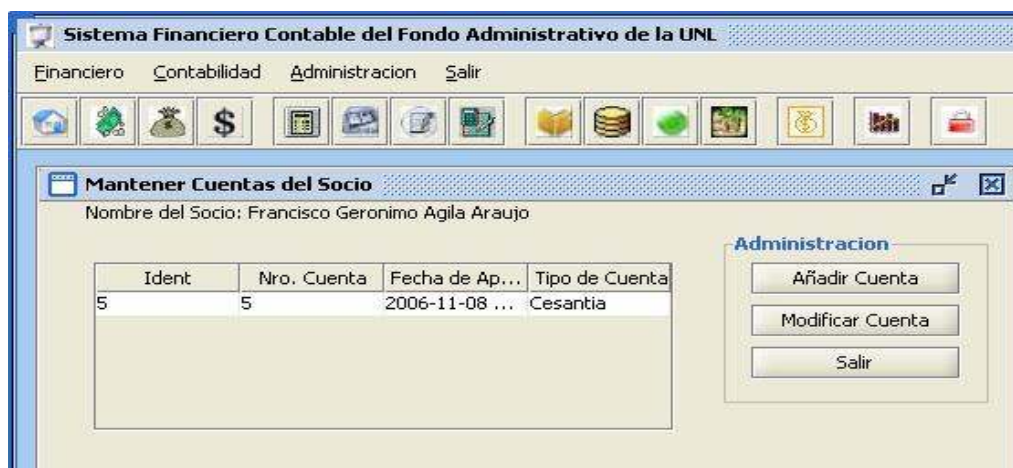
Años de servicio: 4

Telefono: 2589455

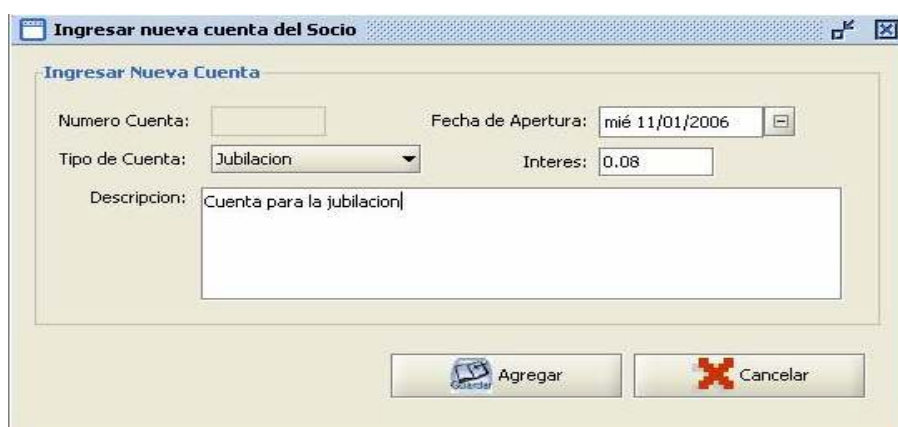
Sueldo Mensual: 600.00

Nro Nombramiento: 4455

Ventana Mantener Cuentas del Socio:



Ventana Ingresar Nueva Cuenta del socio:



Ventana Modificar Cuentas:.



Caso de Uso: Administrar Aportes

Ventana Administrar Aportes (Cesantía).

Administrar Aportes Fecha: 002, 14 nov...

Tipo de Cuenta:

- Cesantía
- Jubilacion

Datos del aporte Individual

Nro. Cuenta: 1

Cedula: 1103805569

Nombres: elizabeth

Apellidos: arias arias

Sueldo: 865,30

Nro.de Aportacion: 24

Total Aportaciones: 1.990,19

Cesantía

Aporte 10%: 086,53 U.S.D

interes: 000,15 U.S.D

Guardar Imprimir Cancelar

Ventana Administrar Aportes (Jubilación)

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Administrar Aportes Fecha: 002, 14 nov...

Tipo de Cuenta:

- Cesantía
- Jubilacion

Datos del aporte Individual

Nro. Cuenta: 2

Cedula: 1103805569

Nombres: elizabeth

Apellidos: arias arias

Sueldo: 865,30

Nro.de Aportacion: 5

Total Aportaciones: 311,51

Jubilacion

Aporte Patronal 10%: 077,88 U.S.D

interes: 000,02 U.S.D

Guardar Imprimir Cancelar


Ventana Administrar Aportes (Grupal).

Administrar Aportes



Aportes Individuales \ Aportes desde un archivo \

Datos del Aporte Grupal

Fecha de Aportación: sáb 04/07/2007

 Generar

Apellidos	Nombres	Cedula	Valor de Cesantía	Valor de Jubilación

 Guardar  Cancelar


Caso de Uso: Administrar Préstamos


Ventana Administrar Préstamos:

Administrar Prestamos

Buscar Socios ...

Criterio de Búsqueda: Apellido

Opcion:  Buscar



Lista de Socios:

Apellidos	Nombres	Cédula
ABENDANO HERRERA	CARMEN MATILD	1101445284
APOLO APOLO	VICENTE ENRIQUE	0700866684
AGUILAR JARAMILLO	BEATRIZ ELAD	1100611035
ARROYO CABRERA	SONIA IRAIDA	1100014941
ARIAS BURNEO	ELIDA BERTHA	1100003969
AGUIRRE RIOFRIO	SILVANA INES	1101927083
ANDRADE FLORES	LUZ AMERICA	1102076591
ALVARADO ESPINOZA	HERNAN BOLIV	1100316049
ANGUISACA MENDOZA	VICTOR MANUE	1100284288

Administración:

Ventana Registrar Nuevo Préstamo:

Registrar Nuevo Préstamo

Datos del Socio

Nro. Cuenta:

Nro. Cedula:

Nombres:

Apellidos:

Datos del Préstamo

Codigo Préstamo:

Monto: USD

Monto Liquido: USD

Numero de Préstamo:

Interes: %

Desgravamen: %

Sector Economico:

Plazo: meses

Tipo de Amortización:

Destino:

Tipo:

Garantía:

Garantes

Nombres	Apellidos	Cedula	Direccion
Garante 1: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Garante 2: <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Detalle del Préstamo

Title 1	Title 2	Title 3	Title 4

Ventana Actualizar Préstamo.

Actualizar Préstamo Fecha: lun, 7 nov 2007

Datos del Socio

Nro. Cuenta:

Nro. Cedula:

Nombres:

Apellidos:

Datos del Préstamo

Codigo Préstamo:

Monto: USD

Monto Liquido: USD

Numero de Préstamo:

Interes: %

Desgravamen: %

Sector Economico:

Plazo: meses

Tipo de Amortización:

Destino:

Tipo:

Garantía:

Detalle del Préstamo

Title 1	Title 2	Title 3	Title 4

Saldo:

Ventana Registrar Pago de Préstamo:

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Registrar Pago de Préstamo

Fecha: 002, 14 nov 2006

Datos del Pago de Préstamo

Nro. Cuenta: Monto Total del Préstamo: **2.500,00**

Apellidos: Nro. Cuota: De

Nombres: Monto de la cuota:

Cedula:

Detalle del Préstamo

Numero de Pa...	Saldo Inicial	Interes	Amortizacion	Pago de Cuotas	Saldo Final	Fecha de Pago	Est. Pago
1	2500.0	302.5	392.74	695.24	2107.26	2006-12-09 0...	Pagado
2	2107.26	254.98	440.26	695.24	1667.0	2007-01-09 0...	En espera
3	1667.0	201.71	493.53	695.24	1173.47	2007-02-09 0...	En espera
4	1173.47	141.99	553.25	695.24	620.22	2007-03-09 0...	En espera
5	620.22	75.05	620.19	695.24	0.03	2007-04-09 0...	En espera

Ventana Pago de Cuota Mensual.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Pago de Cuota mensual

14 nov 2006

Detalle de Pago de Cuota Préstamo

Fecha de Vencimiento:

Nro. Cuota	Saldo Inicial	Amortizacion	Interes	Cuota Mensual	Saldo Final
2	2.107,26	440,26	254,98	695,24	1.667

Estado del Pago

Pagado
 En espera

Numero de Pa...	Saldo Inicial	Interes	Amortizacion	Pago de Cuotas	Saldo Final	Fecha de Pago	Est. Pago
4	1173.47	141.99	553.25	695.24	620.22	2007-03-09 0...	En espera
5	620.22	75.05	620.19	695.24	0.03	2007-04-09 0...	En espera

Ventana Pago de cuotas por Préstamo

Lista de Préstamos Pendientes

Seleccione la fecha a la que desea consultar el pago de Préstamos pendientes:

Numero de...	Saldo Inicial	Interes	Desgravamen	Amortizacion	Pago de C...	Saldo Final	Fecha de P...	Est. Cuota

Caso de Uso: Administrar Retiros


Ventana Administrar Retiros.

Administrar Retiros

Bucar socios ...

Criterio de Búsqueda:

Opcion:



Lista de Socios:

Apellidos	Nombres	Cédula
arias arias	elizabeth	1103805569
arciniegas	jorge salvador	1210865239

Administración:

Ventana Realizar Nuevo Retiro (pestaña Cesantía)

Realizar nuevo retiro

Fecha hoy: 14 nov 2006

Retiros Fondo de Cesantía (Retiros Fondo Jubilación)

codigo de Retiro Cesantía: RC54

Datos del Socio:

Nro. Cuenta:

Nro. Cédula:

Apellido:

Nombre:

Dirección:

Nro. Nombramiento:

Cargo:

Área:

Fecha de Nacimiento:

Salario Mensual: U.S.D.

Datos de la Cuenta:

Fecha de Ingreso:

Nro. Total de Aportaciones: 23

Aporta Mensual: 865,33

Valor por Intereses: 0,15

Total Aportaciones: 1.990,19

Saldo Total: 1.990,34

Total del Retiro: 597,10

Saldo Final: 1.393,24

Ventana Realizar Nuevo Retiro (pestaña Jubilación)

Retiros Fondo de Cesantía \ Retiros Fondo Jubilacion \

Fecha: jue, 2 nov

codigo de Retiro Jubilacion: RJ51

Datos del Socio:

Nro. Cuenta: 2
Nro. Cedula: 1105545421
Apellidos: Arias Arias
Nombres: Anibal Fernando
Dirección: Av Iberoamerica y 10 de Agosto
Nombramiento: 0012455552
Cargo: Secretario
Area Academica: Energia
Sueldo Mensual: 512.00 U.S.D.
Fecha de Nacimiento: dom 11/02/1969

Datos de la Cuenta:

Fecha de Ingreso: jue 11/02/2006
Nro. Total de Aportaciones: 7
Total de Aporte Mensual: 322.56

Tipode Jubilacion

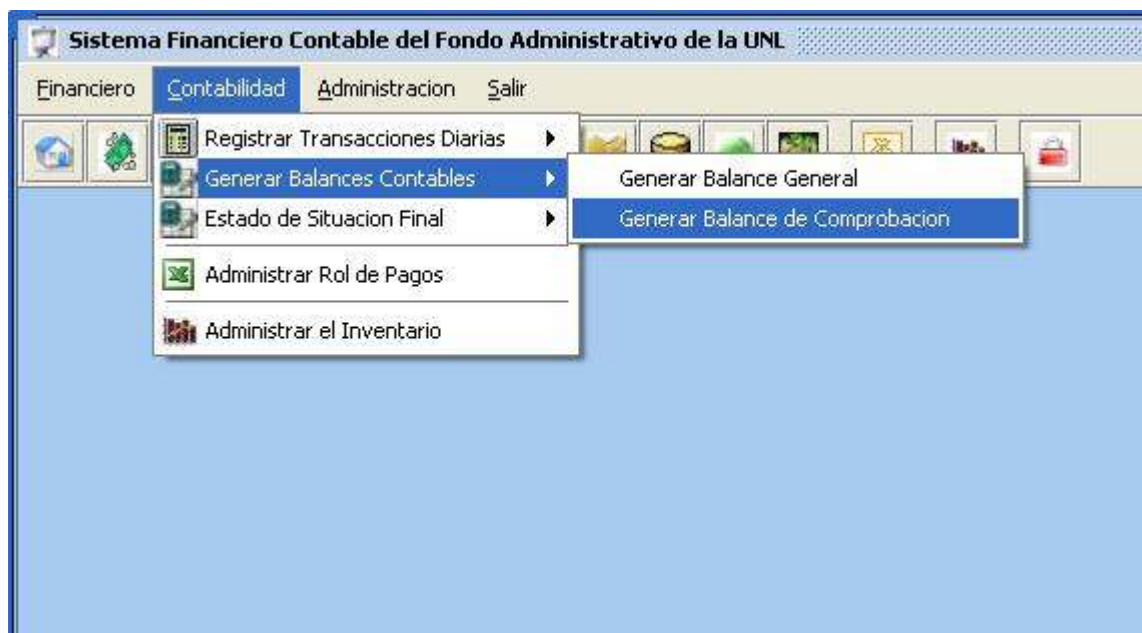
Jubilacion General
 Jubilacion Especial
 Jubilacion Reducida

Total de Retiro: 129.02

Guardar Imprimir Cancelar

Caso de Uso: Administrar Cuentas Contables.

Submenú Contabilidad



Ventana Administrar Cuentas Contables.

Administrar Cuentas Contables Fecha: mar, 28 nov 2006

Cuentas Contables

Cod Cuenta	Nom. Cuenta	Tipo de Cuenta	Clasificación de Cuenta
1	ACTIVO	Activo	Contiene Sub-Cuentas
1.1	ACTIVO CORRIENTE	Activo	Contiene Sub-Cuentas
1.1.1	Disponible	Activo	Contiene Sub-Cuentas
1.1.1.01	Caja Chica	Activo	De Detalle
1.1.1.02	Bancos	Activo	Contiene Sub-Cuentas
1.1.1.02.01	Filanbanco	Activo	De Detalle
1.1.1.02.02	Unibanco	Activo	De Detalle
1.1.1.02.03	Banco de Loja	Activo	De Detalle
1.1.1.02.04	Banco de Guayaquil	Activo	De Detalle
1.1.2	Inversiones	Activo	Contiene Sub-Cuentas
1.1.2.01	Inversiones Financieras	Activo	De Detalle
1.1.2.02	Bonos	Activo	De Detalle
1.1.3	Exigible	Activo	Contiene Sub-Cuentas
1.1.3.01	Cuentas por cobrar	Activo	De Detalle
1.1.3.02	Documentos por cobrar	Activo	De Detalle
1.1.3.03	Prestamo social	Activo	De Detalle

Administrar Cuentas Contables

Ventana Adicionar Cuenta Contable.

Adicionar Cuenta Contable Fecha: 004, 28 nov 06

Codigo de la Cuenta: 8.2 Nombre de la Cuenta: suministros computación

Descripcion: suministros de computacion como :disquetts, Cds, etc.

Cuenta del:

Activo Capital Gastos Gastos de Capital
 Pasivo Ingresos Cuentas de Orden

Esta cuenta es:

De Detalle Contiene Sub-Cuentas

Ventana Actualizar Cuenta Contable

Actualizar Cuenta Contable Fecha: 004, 28 nov 06

Codigo de la Cuenta: 1.1.1.02.03 Nombre de la Cuenta: Banco de Loja

Descripcion: Cuentas de Bancos

Cuenta del:

Activo Capital Gastos Gastos de Capital
 Pasivo Ingresos Cuentas de Orden

Esta cuenta es:

De Detalle Contiene Sub-Cuentas

Caso de Uso: Ingresar Transacciones diarias

Ventana Administrar Asientos Contables.

Administrar Asientos Contables

Fecha:

Asientos Contables

Nro Asiento	Cod. Cuenta	Cuenta	Debe	Haber	Referencia	Diario

Totales: 0.00 0.00

Administracion

Ventana Ingresar Nuevo Asiento Contable

Ingresar Nuevo Asiento Contable

Datos Principales

Nro de Asiento: 2 Libro Diario: 2006 Fecha: vie 11/03/2006 Referencia: Varios Comprobante: 123456 Fuente: Ingresos

Asientos Contables

Nro Asiento	Comentario	Debe	Haber	Registro Tributario
2	aportes de los socios	7523.0	0.0	
2	retiros de fondos	7523.0	0.0	

Transacciones

Cod. De Cuenta	Descripcion:	Debito:	Credito:	Reg. Tributario:
1.1.1.01	aportes	900.00	000.00	12555655

Caja Chica

Totales: 15,046.00 0.00 15,046.00

Ventana Modificar Asiento Contable.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Modificar Asiento Contable

Numero de Asiento: Modificar Asiento

Asientos Contables

Nro Asiento	Fecha	Referencia	Comentario	Debe	Haber	Registro Tributario
1	Fri Nov 03 16:37:5...	Varios	Deposito de aporte...	5222.0	0.0	
	Fri Nov 03 16:37:5...	Varios	retiro de fondos	0.0	5222.0	1234545

Modificar Item

Totales: Debito **5,222.00** Credito **5,222.00** Diferencia: **000.00**

Ventana Modificar Ítem Seleccionado del Libro Diario.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Modificar Item Seleccionado del Libro Diario

Datos Principales

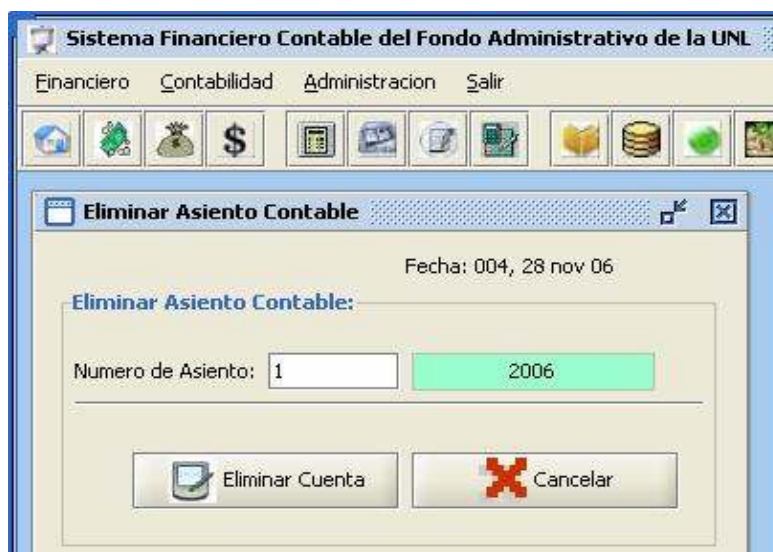
Nro de Asiento: Libro Diario: Fecha: Referencia: Comprobante: Fuente:

Transacciones

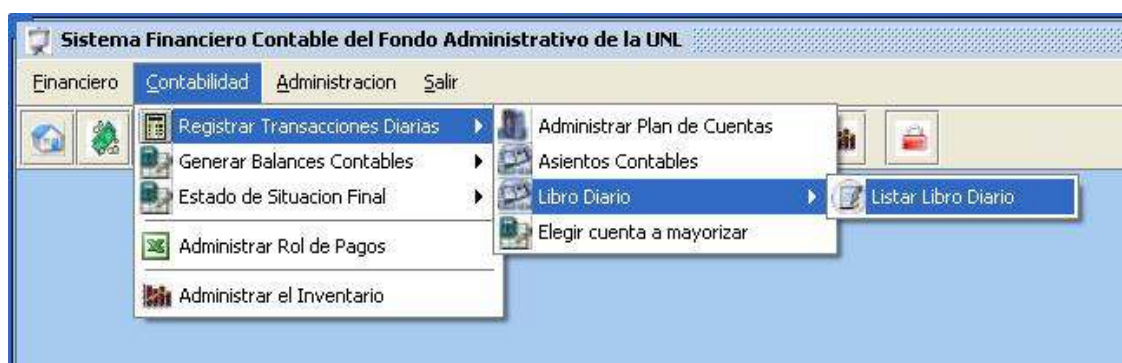
Cod. De Cuenta	Descripcion:	Debito:	Credito	Reg. Tributario:
1.1.1.01	retiro de fondos	000.00	5,222.00	1234545

Caja Chica

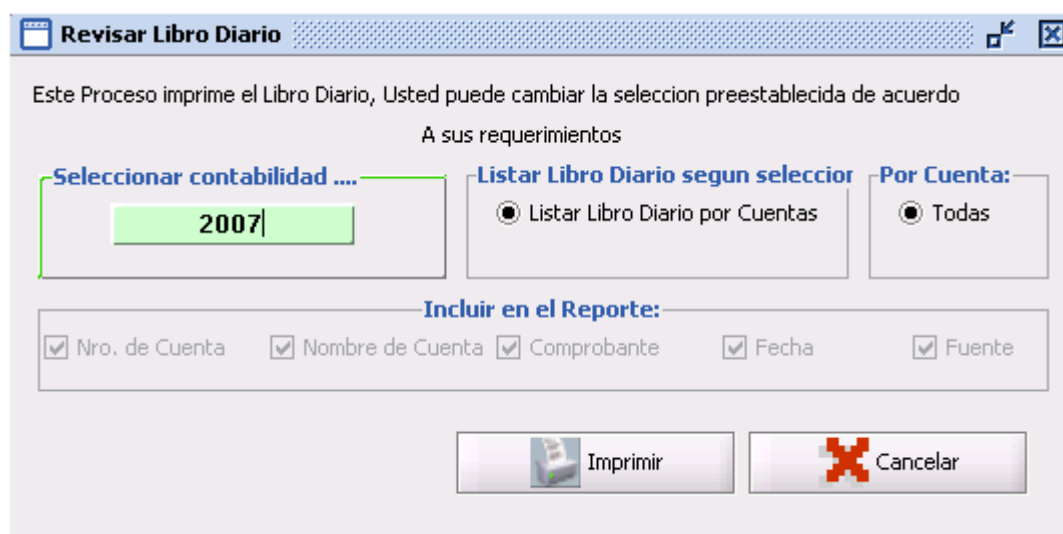
Ventana Eliminar Asiento Contable.



Submenú Libro Diario



Ventana Revisar Libro Diario.




Ventana Generar Libro Mayor.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Generar Libro Mayor

Fecha:



Generar Libro Mayor por Cuenta

Cuenta Contable:

Descripcion Cuenta:

Detalle Mayorización

Nro Asiento	Codigo de Cu...	Referencia	Comentario	Debe	Haber	Registro Trib...
1	1.1.1.01	Varios	Deposito de ...	5222.0	0.0	
1	1.1.1.01	Varios	retiro de fon...	0.0	5222.0	1234545
Totales:				5,222.00	5,222.00	

Caso de Uso: Generar Balances Contables

Ventana Generar Balance General.

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Generar Balance General



Balance General del Fondo Administrativo de la UNL

Este Reporte puede ser ejecutado en cualquier momento, no altera los asientos Contables

Balance Contable al mes:

Balance al mes:

Total del Balance General

Total Pasivo + Capital:

Ventana Generar Balance de Comprobación.




Submenú Estado de Situación Final.



Ventana Generar Estado de Pérdidas y Ganancias.

Generar Estado de Perdidas y Ganancias



ESTADO DE PÉRDIDAS Y GANANCIAS

Estado de Perdidas del diario 2007

Pérdida del Ejercicio Ganancia del Ejercicio


Total

Imprimir Generar Salir

Ventana Generar Cierre del Período Contable.

Generar Cierre del Periodo Contable

FONDO ADMINISTRATIVO DE LA UNL



Obtener reportes de:

- Balance General
- Pérdidas y Ganancias
- Balance de Comprobación

Seleccione el Periodo Contable a Cerrar:

Fecha de Apertura	Fecha de Cierre	Nombre de Contabil...	Estado
2007-05-04 00:00:...	2007-05-31 00:00:...	2007	abierto

Procesar Salir

Ventana Administrar Rol de Pagos

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Fecha:

Período

Desde: vie 11/03/2006

Lista del Rol de Pago

Fecha	Num. Cheque	Nombre	Apellido	Cantidad	Estado
Fri Nov 03 00:00:...	1	Juan Carlos	Arias Arias	350.0	Cobrado

Administrar Rol De Pag

Ventana Emitir cheques (pestaña Chequera).

Sistema Financiero Contable del Fondo Administrativo de la UNL

Financiero Contabilidad Administracion Salir

Emitir Cheques

Chequera \ Auxiliar de Bancos \

Fecha: vie 11/03/2006

Socio: La Cantidad de: 350.00

Pague a la orden de: Juan Carlos Arias Arias

La Cantidad de: trecientos cincuenta 00/100 USD

Referencia: pago de sueldo

Estado del Cheque

Cobrado

En espera

Relacion de Cheques:

Fecha	Num. Cheque	Nombre	Apellido	Cantidad	Estado
Fri Nov 03 00:00:...	1	Juan Carlos	Arias Arias	350.0	Cobrado

Ventana Lista De Socios del Rol de Pago

The screenshot shows a window titled "Lista de socios" within the "Sistema Financiero Contable del Fondo Administrativo de la UNL". The window has a menu bar with "Financiero", "Contabilidad", "Administracion", and "Salir". Below the menu is a toolbar with various icons. The main area contains a search field labeled "Buscar:" with the letter "a" entered and a "Buscar" button. Below this is a table titled "Lista de Socios" with the following data:

Apellidos	Nombres	Cedula	Direccion	Cargo	Sueldo M...
Arias Arias	Juan Carlos	1103367341	Av Ibero...	Secretario	740.0

At the bottom of the window are two buttons: "Seleccionar" and "Salir".

Ventana Emitir cheques (pestaña Auxiliar de Bancos).

The screenshot shows a window titled "Emitir Cheques" within the "Sistema Financiero Contable del Fondo Administrativo de la UNL". The window has a menu bar with "Financiero", "Contabilidad", "Administracion", and "Salir". Below the menu is a toolbar with various icons. The main area contains a breadcrumb trail "Chequera \ Auxiliar de Bancos \". Below this is a section titled "Lista de Cheques" with a "Fecha:" label. There are two date input fields: "Desde el:" with the value "vie 11/03/2006" and "Hasta el:" with the value "vie 11/03/2006". Below this is a table titled "Lista de cheques" with the following structure:

Title 1	Title 2	Title 3	Title 4

At the bottom of the window are two buttons: "Imprimir" and "Salir".

Caso de Uso: Administrar Inventario

Ventana Administrar Inventario

The screenshot shows the 'Administrar Inventario' window. At the top, there is a menu bar with 'Financiero', 'Contabilidad', 'Administracion', and 'Salir'. Below the menu is a toolbar with various icons. The main area is divided into several sections:

- Buscar Producto ...**: A search section with a dropdown menu for 'Criterio' set to 'Nombre', a text input for 'Buscar por', and a 'Buscar' button.
- Fecha:** 'vie, 3 nov 2...'
- Inventario**: A table with the following data:

Cod Producto	Nombre	Cantidad	Total	Tipo
AE-0015245	computador	2	1700.0	Fisico
- Administrador Inventario**: A vertical list of buttons: 'Agregar producto', 'Modificar Producto', 'Eliminar Producto', 'imprimir', 'Salir', and 'Agregar Inventario'.

Ventana Ingresar producto al Inventario.

The screenshot shows the 'Ingresar Producto al Inventario' window. At the top, there is a menu bar with 'Financiero', 'Contabilidad', 'Administracion', and 'Salir'. Below the menu is a toolbar with various icons. The main area contains the following fields and controls:

- Fecha:** 'vie 03/11/2006'
- Codigo:** 'AE-0015245'
- Nombre:** 'computador'
- Descripción:** 'computador de escritorio pentium 4'
- Costo:** '850.00' USD
- Proveedor:** 'Sisconet'
- Ubicación:** 'contabilidad'
- Cantidad:** '2'
- Producto:** Radio buttons for 'Físico' (selected), 'Servicio', and 'Uso Ocasional'.
- Costo Total del Producto:** '1,700.00' (highlighted in green)
- Buttons:** 'Aceptar' and 'Cancelar'.

Ventana Modificar producto del Inventario.

The screenshot shows a web application window titled "Sistema Financiero Contable del Fondo Administrativo de la UNL". The main menu includes "Financiero", "Contabilidad", "Administración", and "Salir". A toolbar contains various icons for navigation and actions. The active window is "Modificar Producto del Inventario".

Form fields and values:

- Código: AE-0015245
- Fecha: vie 03/11/2006
- Nombre: computador
- Descripción: computador de escritorio pentium 4
- Costo: 850.00 USD
- Proveedor: Sisconet
- Ubicación: contabilidad
- Cantidad: 2

Product type selection:

- Físico
- Servicio
- Uso Ocasional

Costo Total del Producto: **1,700.00**

Buttons: Aceptar, Cancelar

Ventana Ingresar un nuevo Inventario

The screenshot shows a web application window titled "Sistema Financiero Contable del Fondo Administrativo de la UNL". The main menu includes "Financiero", "Contabilidad", "Administración", and "Salir". A toolbar contains various icons for navigation and actions. The active window is "Ingresar un nuevo Inventario".

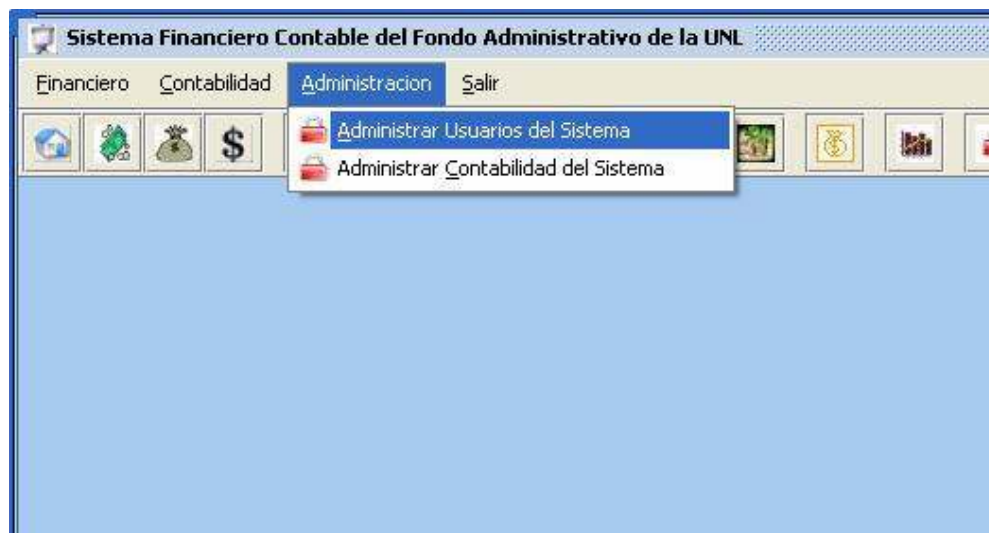
Form fields and values:

- Fecha: vie 11/03/2006
- Nombre del Inventario: 2006
- Descripción: inventario del año 2006

Buttons: Guardar, Salir

The window also features a logo on the left side, which is the seal of the Universidad Nacional de La Plata.

Menú Administración



Ventana Administrar Usuarios del Sistema.

The screenshot shows the 'Administrar Usuarios del Sistema' window. It includes a search section with a dropdown for 'Buscar por:' set to 'Nombre' and a 'Buscar' button. Below this is a table titled 'Usuarios del Sistema' with columns 'Nombres' and 'Cedula'. The table contains one entry: 'Administrador'. To the right of the table is a sidebar titled 'Administracion Usuarios' with buttons for 'Ingresar Usuario', 'Actualizar Usuario', 'Eliminar Usuario', 'Agregar Rol', 'Imprimir Reporte', and 'Salir'. A logo is visible in the top right corner of the window.

Nombres	Cedula
Administrador	

Ventana Ingresar Nuevo Usuario.

The screenshot shows a software window titled "Sistema Financiero Contable del Fondo Administrativo de la UNL". The menu bar includes "Financiero", "Contabilidad", "Administracion", and "Salir". Below the menu is a toolbar with various icons. The main window is titled "Ingresar nuevo Usuario" and contains two sections: "Datos del Usuario" and "Variables de Acceso".

Datos del Usuario

Nombres:	<input type="text" value="Milton Ricardo"/>
Apellidos:	<input type="text" value="Palacios Morocho"/>
Cedula:	<input type="text" value="1103985541"/>
Direccion:	<input type="text" value="Ciudadela La banda"/>
Fecha de Nacimiento:	<input type="text" value="lun 12/08/1980"/>
Telefono:	<input type="text" value="097421454"/>

Variables de Acceso

Nombre de Usuario:	<input type="text" value="milton"/>
Contraseña:	<input type="text" value="milton21"/>

Buttons:

Ventana Modificar Usuario

The screenshot shows a software window titled "Sistema Financiero Contable del Fondo Administrativo de la UNL". The menu bar includes "Financiero", "Contabilidad", "Administracion", and "Salir". Below the menu is a toolbar with various icons. The main window is titled "Modificar Usuario" and contains two sections: "Datos del Usuario" and "Variables de Acceso".

Datos del Usuario

Nombres:	<input type="text" value="Administrador"/>
Apellidos:	<input type="text"/>
Cedula:	<input type="text"/>
Direccion:	<input type="text"/>
Fecha de Nacimiento:	<input type="text"/>
Telefono:	<input type="text"/>

Variables de Acceso

Nombre de Usuario:	<input type="text" value="admin"/>
Contraseña:	<input type="text" value="admin21"/>

Buttons:

Ventana Adicionar Rol al Usuario



Caso de Uso: Administrar Contabilidad

Ventana Administrar Contabilidad



Ventana Adicionar Período de Contabilidad

The screenshot shows the 'Nuevo periodo Contable' window. The title bar reads 'Sistema Financiero Contable del Fondo Administrativo de la UNL'. The menu bar includes 'Financiero', 'Contabilidad', 'Administracion', and 'Salir'. A toolbar contains various icons. The window content is titled 'Nuevo periodo Contable' and has a 'Configuracion' section. The fields are: 'Libro Diario del Periodo' with a text box containing '2006'; 'Fecha de Inicio Periodo Contable:' with a date picker showing '03/11/2006'; 'Fecha de CierrePeriodo Contable:' with a date picker showing '03/11/2006'; and 'Estado del Periodo:' with a dropdown menu showing 'abierto'. At the bottom are 'Guardar' and 'Cancelar' buttons.

Ventana Modificar Contabilidad.

The screenshot shows the 'Modificar periodo Contable' window. The title bar reads 'Sistema Financiero Contable del Fondo Administrativo de la UNL'. The menu bar includes 'Financiero', 'Contabilidad', 'Administracion', and 'Salir'. A toolbar contains various icons. The window content is titled 'Modificar periodo Contable' and has a 'Configuracion' section. The fields are: 'Libro Diario del Periodo' with a text box containing '2006'; 'Fecha de Inicio Periodo Contable:' with a date picker showing '03/11/2006'; 'Fecha de CierrePeriodo Contable:' with a date picker showing '03/11/2006'; and 'Estado del Periodo:' with a dropdown menu showing 'abierto'. At the bottom are 'Guardar' and 'Cancelar' buttons.

DESCRIPCIÓN DE CASOS DE USO

Narración de los use cases encontrados

Es la descripción del conjunto de secuencias de acciones que el sistema ejecuta y que produce un resultado observable de interés para un actor particular, un caso de uso se utiliza para estructurar los aspectos de comportamiento en un modelo, los casos de uso se escriben con el fin de expresar lo que debe hacer el sistema a desarrollar, sin tener en cuenta cómo debe hacerlo.

DESCRIPCIÓN DE LOS CASOS DE USO

Nombre:	Autenticar Usuario												
Meta:	Validar un usuario e ingresar al Sistema del Fondo de Cesantía y Jubilación del trabajador administrativo de la UNL.												
Descripción:	El Administrador ingresa su nombre de usuario y contraseña para ingresar al Sistema.												
Precondiciones:	El Administrador deberá haber presionado “ Enter ” el icono de acceso al sistema.												
Poscondiciones:	Validar a un usuario, el sistema presenta la ventana Principal .												
Actores:	Administrador												
Tipo:	-Primario -Esencial												
Flujo Normal de eventos: Ingresar al sistema													
<table border="1" style="width: 100%;"> <thead> <tr> <th colspan="2">ACCIONES DEL ACTOR</th> <th colspan="2">RESPUESTA DEL SISTEMA</th> </tr> </thead> <tbody> <tr> <td style="width: 5%;">01</td> <td style="width: 45%;">El Administrador ingresa el nombre de usuario y contraseña, luego presiona el botón Ingresar en la ventana Ingresar al Sistema.</td> <td style="width: 5%;">02</td> <td style="width: 45%;">El Sistema valida el nombre de usuario y contraseña, luego presenta la ventana Principal.</td> </tr> <tr> <td></td> <td></td> <td>03</td> <td>El caso de uso finaliza.</td> </tr> </tbody> </table>		ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA		01	El Administrador ingresa el nombre de usuario y contraseña, luego presiona el botón Ingresar en la ventana Ingresar al Sistema .	02	El Sistema valida el nombre de usuario y contraseña, luego presenta la ventana Principal .			03	El caso de uso finaliza.
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA											
01	El Administrador ingresa el nombre de usuario y contraseña, luego presiona el botón Ingresar en la ventana Ingresar al Sistema .	02	El Sistema valida el nombre de usuario y contraseña, luego presenta la ventana Principal .										
		03	El caso de uso finaliza.										

Curso alternativo de eventos:

Curso alternativo A: nombres mal ingresados

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
02	El Administrador presiona el botón Aceptar del mensaje de error.	03	El Sistema presenta un mensaje de error de ingreso de nombre de usuario y contraseña.
04	El caso de uso continúa con el paso 1 del curso normal de eventos.		

Nombre:	Administrar Usuarios del Sistema
Meta:	Habilitar a los nuevos usuarios para que ingresen al sistema, de acuerdo a los roles establecidos.
Descripción:	Registrar los usuarios que operan el sistema
Precondiciones:	<ul style="list-style-type: none"> - El Administrador debe estar dentro del sistema. - El Administrador debe estar autenticado. - El Administrador debe seleccionar el menú Administración en la ventana Principal. - El Administrador selecciona el ítem Administrar Usuarios del Sistema. - El Sistema presenta la ventana Administrar Usuarios del Sistema.
Poscondiciones:	- El Sistema muestra la ventana Principal .
Actores:	Administrador, Contadora
Tipo:	-Primario -Esencial

Flujo Normal de eventos: Agregar Usuario

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón Ingresar Usuario en la ventana Administrar Usuarios del Sistema .	02	El Sistema muestra la ventana Ingresar Nuevo Usuario .
03	El Administrador ingresa los datos en los campos activos de la ventana Ingresar Nuevo Usuario .		
04	El Administrador presiona el botón Guardar en la ventana Ingresar Nuevo Usuario .	05	El Sistema guarda el nuevo usuario y presenta un mensaje de confirmación.
06	El Administrador presiona el botón Aceptar del mensaje de confirmación.	07	El sistema presenta la Administrar Usuarios del Sistema .
		08	El caso de uso finaliza.

Curso alterno de eventos:

Curso alterno A: Actualizar Usuario

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el usuario lo selecciona, y presiona el botón Actualizar Usuario en la ventana Administrar Usuarios del Sistema .	02	El Sistema presenta el usuario en la ventana Modificar Usuario .
03	El caso de uso continúa con el paso 03 del curso normal de eventos.		

Curso alterno B: Eliminar Usuario

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el usuario, lo selecciona y presiona el botón Eliminar Usuario en la ventana Administrar Usuarios del Sistema .	02	El Sistema presenta un mensaje de confirmación.
03	El Administrador presiona SI en el mensaje de confirmación.	04	El sistema borra el usuario del Sistema y presenta un mensaje de notificación.
05	El Administrador presiona Aceptar en el mensaje de notificación.		
06	El caso de uso finaliza		

Curso alterno C: Asignar Rol

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el usuario, lo selecciona y presiona el botón Agregar Rol en la ventana Administrar Usuarios del Sistema .	02	El Sistema presenta la ventana Adicionar Rol al Usuario .
03	El Administrador selecciona el rol del Usuario y su descripción, luego presiona el botón Añadir en la ventana Adicionar Rol al Usuario .	04	El sistema guarda el usuario del Sistema y presenta un mensaje de notificación.
05	El Administrador presiona Aceptar en el mensaje de notificación.	06	El sistema presenta la ventana Administrar Usuarios del Sistema .
07	El caso de uso finaliza		

Nombre:	Administrar Socios	
Meta:	Registrar los datos del socio del Fondo de Cesantía y Jubilación del trabajador administrativo de la UNL	
Descripción:	El Administrador registra los datos personales y asigna el número de cuenta al nuevo socio que desea ingresar al Fondo de Cesantía y Jubilación Administrativo de la UNL.	
Precondiciones:	<ul style="list-style-type: none"> - El Administrador debe ingresar al sistema - El Administrador debe estar autenticado con usuario y contraseña. - El Administrador debe estar en la ventana Principal. - El Administrador deberá haber presionado Administrar socios del menú Financiero en la ventana Principal. - La ventana Administrar Socio debe estar activa. 	
Poscondiciones:	<ul style="list-style-type: none"> - Guardar los datos personales del nuevo socio. - El sistema retorna a la ventana Sistema del Fondo Administrativo de la UNL. 	
Actores:	Administrador	
Tipo:	-Primario -Esencial	
Flujo Normal de eventos: Ingresar Nuevo Socio		
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA
01	El Administrador presiona el botón Ingresar Socio de la ventana Administrar socios .	02 El Sistema presenta la ventana Ingresar Nuevo Socio .
03	El Administrador ingresa los datos en los campos activados en la ventana Ingresar Nuevo Socio .	04 El sistema valida los datos ingresados en la ventana Ingresar Nuevo Socio .
05	El Administrador presiona el botón Guardar de la ventana Ingresar Nuevo Socio .	06 El Sistema guarda los datos del socio y presenta un mensaje de notificación.
07	El Administrador presiona el botón Aceptar del mensaje de notificación.	08 El caso de uso finaliza

Curso alternativo de eventos:

Curso alternativo A: Datos Mal Ingresados

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
06	El Administrador modifica los datos de los campos en la ventana Ingresar Nuevo Socio .	05	El Sistema presenta un mensaje de error en el campo en el que se encuentran los datos mal ingresados.
07	El caso de uso continua con el paso 5 del curso normal de eventos		

Curso alternativo B: Modificar datos del socio

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el socio en la ventana Administrar Socios .		El sistema presenta los datos del socio en la ventana Modificar Socio . El Sistema continúa con el paso 06 del curso normal de eventos
02	El Administrador selecciona el socio de la tabla Lista de Socios y presiona el botón Modificar Socio en la ventana Administrar Socios .	03	
04	El Administrador modifica los datos requeridos del socio y presiona el botón Guardar en la ventana Modificar Socio .	05	

Curso alternativo C: Buscar Socios

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador selecciona un criterio de búsqueda en la ventana Administrar Socios .		El Sistema presenta los resultados de la búsqueda en el panel Lista de Socios de en la ventana Administrar Socios .
02	El Administrador ingresa el valor en el campo Opción .		
03	El Administrador presiona el botón Buscar en la ventana Administrar Socios .	04	
05	El caso de uso finaliza.		

Curso alternativo D: Asignar Cuentas al Socio

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el socio en la ventana Administrar Socios.		
02	El Administrador selecciona el socio y presiona el botón Mantener Cuenta en la ventana Administrar Socios.	03	El Sistema presenta la ventana Mantener Cuentas del Socio.
04	El Administrador presiona el botón Añadir Cuenta en la ventana Mantener Cuentas del Socio.	05	El Sistema presenta la ventana Ingresar Nueva Cuenta del socio.
06	El Administrador llena los datos de cuenta en la ventana Ingresar Nueva Cuenta del socio.		
07	El Administrador presiona el botón Agregar de la ventana Ingresar Nueva Cuenta del socio.	08	El Sistema registra la nueva cuenta y presenta un mensaje de notificación
09	El Administrador presiona el botón Aceptar del mensaje de notificación.		
10	El caso de uso finaliza		

Curso alternativo E: Modificar Cuentas al Socio

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador busca el socio en la ventana Administrar Socios.		
02	El Administrador selecciona el socio y presiona el botón Mantener Cuenta en la ventana Administrar Socios.	03	El Sistema presenta la ventana Mantener Cuentas del Socio.
04	El Administrador presiona el botón Modificar Cuenta en la ventana Mantener Cuentas del Socio.	05	El Sistema presenta la ventana Modificar Cuentas.
06	El Administrador modifica los datos de cuenta en la ventana Modificar Cuentas.		
07	El Administrador presiona el botón Agregar de la ventana Modificar Cuentas.	08	El Sistema actualiza la cuenta y presenta un mensaje de notificación
09	El Administrador presiona el botón Aceptar del mensaje de notificación.		
10	El caso de uso finaliza		

Nombre:	Administrar Aportes
Meta:	Ingresar los aportes mensuales de el/los socio(s) del Fondo de Cesantía y Jubilación del trabajador administrativo de la UNL.
Descripción:	El Administrador ingresa los aportes mensuales que realizan los socios del Fondo de Jubilación y Cesantía del trabajador administrativo de la UNL.
Precondiciones:	<ul style="list-style-type: none"> - El Administrador debe ingresar al sistema - El Administrador debe estar autenticado con nombre de usuario y contraseña. - El Administrador debe estar ubicado en la ventana Principal. - El Administrador debe seleccionar Administrar Aportes en el menú Financiero. - El Sistema presenta la ventana Administrar Aportes. - El sistema presenta la pestaña Aportes Individuales.
Poscondiciones:	Después de guardar el aporte mensual de el/los socio(s); el sistema retorna a la ventana Sistema del Fondo Administrativo de la UNL .
Actores:	Administrador
Tipo:	-Primario -Esencial

Flujo Normal de eventos: Aporte de Cesantía.

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador escoge las opciones: Cesantía del panel Tipo de Cuenta en la ventana Administrar Aportes .	02	El Sistema visualiza el panel Cesantía en la ventana Administrar Aportes .
03	El Administrador ingresa el número de Cuenta y presiona Enter en la ventana Administrar Aportes .	04	El Sistema llena los campos con los datos del socio y del aporte individual en la ventana Administrar Aportes .
		05	El sistema calcula el 10% del sueldo del socio para Cesantía y Jubilación y lo presenta en el campo Aporte 10% de la ventana Administrar Aportes .
06	El Administrador presiona el botón Imprimir de la ventana Administrar Aportes .	07	El Sistema visualiza el comprobante de depósito en la ventana del comprobante de depósito .

08	El Administrador presiona el icono de Imprimir de la ventana del comprobante de depósito .	09	El Sistema imprime el comprobante
10	El Administrador cierra la ventana del comprobante de depósito presionando el icono de cierre de dicha ventana.		
11	El Administrador presiona el Guardar de la ventana Administrar Aportes .	12	El Sistema guarda el aporte y presenta un mensaje de notificación.
		13	El Sistema muestra un mensaje de notificación.
14	El Administrador presiona el botón No del mensaje de notificación.	15	El caso de Uso finaliza.

Curso alternativo A: Aportes Jubilación

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador escoge selecciona la pestaña Aportes en Grupo en la ventana Administrar Aportes	02	El sistema presenta la pestaña Aportes en Grupo de la ventana Administrar Aportes .
		03	El sistema continúa con el paso 3 del curso normal de eventos.
		04	El sistema calcula el 10% del sueldo y lo fija en el campo Aporte Patronal 10% de la ventana Administrar Aportes .
		05	El Sistema continúa con los pasos 06 – 15 del curso normal de eventos.

Curso alternativo B: Aportes Grupales

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador selecciona la pestaña Aportes desde un Archivo en la ventana Administrar Aportes	02	El Sistema presenta la pestaña Aportes en Grupo en la Administrar Aportes
03	El Administrador presiona el botón Abrir Archivo del panel Datos del Aporte Grupal en la ventana Administrar Aportes .		
04	El Administrador busca el archivo xls (Excel) y presiona el botón Abrir de la ventana Abrir .	05	El sistema presenta la ventana abrir .
		06	El Sistema presenta los datos obtenidos del archivo xls .
		07	El sistema continúa con el paso 11 del curso normal de eventos.

Nombre:	Administrar Préstamos	
Meta:	Ingresar en el Sistema los datos de los préstamos que se conceden a los socios del Fondo de Cesantía y Jubilación del Trabajador Administrativo de la UNL.	
Descripción:	El Gerente aprueba un préstamo al socio y este préstamo es ingresado al sistema del Fondo de Cesantía y Jubilación del Trabajador Administrativo	
Precondiciones:	<ul style="list-style-type: none"> - El Gerente aprueba un préstamo al socio. - El Administrador debe ingresar al sistema - El Administrador debe estar autenticado con un nombre de usuario y contraseña. - El Administrador debe estar en la ventana Principal. - El Administrador debe presionar el Administrar Préstamos del menú Financiero en la ventana Principal. - El Sistema presenta la ventana Administrar Prestamos. 	
Poscondiciones:	Guardar el préstamo concedido al socio, el sistema se ubica en la ventana Sistema del Fondo Administrativo de la UNL .	
Actores:	Administrador	
Tipo:	-Primario -Esencial	
Flujo Normal de eventos: Registrar Préstamo		
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA
01	El Administrador presiona el botón Nuevo Préstamo en la ventana Administrar Prestamos .	02 El Sistema presenta la ventana Registrar Nuevo Préstamo .
03	El Administrador ingresa el número de cuenta y presiona enter en la ventana Registrar Nuevo Préstamo .	04 El Sistema muestra un mensaje de notificación informando que el socio puede acceder a un préstamo.
05	El Administrador presiona el botón Aceptar del mensaje de notificación	06 El Sistema presenta los datos del socio en los campos del panel Datos del socio en la ventana Registrar Nuevo Préstamo .

07	El administrador verifica los datos del socio en el panel Datos del socio en la ventana Registrar Nuevo Préstamo .	
08	El Administrador ingresa el monto, plazo y tipo de amortización del panel Datos del préstamo de la ventana Registrar Nuevo Préstamo .	
09	El Administrador presiona el botón Generar Detalle en la ventana Registrar Nuevo Préstamo .	10 El Sistema presenta los datos en el panel Detalle del préstamo en la ventana Registrar Nuevo Préstamo .
11	El Administrador presiona el botón Imprimir del panel Detalle del Préstamo en la ventana Registrar Nuevo Préstamo .	12 El Sistema visualiza el detalle del préstamo en la ventana del detalle del préstamo .
13	El Administrador presiona el icono de Imprimir de la ventana del detalle del préstamo .	14 El Sistema imprime el informe del detalle del préstamo.
15	El Administrador cierra la ventana del informe del detalle del préstamo presionando el icono de cierre de dicha ventana.	
16	El Administrador presiona el botón Guardar en la ventana Registrar Nuevo Préstamo .	17 El Sistema guarda el préstamo que se le realiza al socio y presenta un mensaje de notificación.
18	El Administrador presiona el Aceptar del mensaje de notificación.	19 El caso de uso finaliza

Curso alterno de eventos:

Curso alterno A: Registrar Pago de Préstamos

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón Pago de Préstamo en la ventana Administrar Prestamos .	02	El Sistema presenta la ventana Registrar Pago de Préstamo .
03	El Administrador ingresa el número de cuenta y presiona Enter en la ventana Registrar Pago de Préstamo .	04	El Sistema presenta los datos del socio y el detalle de las cuotas en la ventana Registrar Pago de Préstamo .
05	El Administrador selecciona la cuota de pago del panel Detalle del Préstamo en ventana Registrar Pago de Préstamo .		
06	El Administrador presiona el botón Editar Pago en la ventana Registrar Pago de Préstamo .	07	El Sistema presenta los datos de la Cuota de pago en la ventana Pago de Cuota Mensual .
08	El Administrador selecciona Pagado del panel Estado del pago en la ventana Pago de Cuota Mensual .		
09	El Administrador presiona el botón Guardar en la ventana Pago de Cuota Mensual .	10	El Sistema guarda los datos de la cuota pagada y presenta un mensaje de notificación
11	El Administrador presiona el Aceptar del mensaje de notificación	12	El Sistema presenta la ventana Registrar Pago de Préstamo .
13	El Administrador presiona el botón Imprimir del panel Detalle del Préstamo en la ventana Registrar Pago de Préstamo .	14	El Sistema presenta el Comprobante de Pago en la ventana del Comprobante de Pago .
15	El Administrador presiona el icono de Imprimir de la ventana del Comprobante de Pago .	16	El Sistema imprime del Comprobante de Pago .
17	El Administrador cierra la ventana del Comprobante de Pago .presionando el icono de cierre de dicha ventana.	18	El Sistema presenta la ventana Registrar Pago de Préstamo .
		19	El caso de uso finaliza

Curso alterno B: ActualizarPréstamos

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón ActualizarPréstamo en la ventana Administrar Prestamos .	02	El Sistema presenta la ventana ActualizarPréstamos .
03	El Administrador ingresa el número de cuenta y presiona Enter en la ventana ActualizarPréstamos .	04	El Sistema presenta los datos del socio y del préstamo en la ventana ActualizarPréstamos .
05	El Administrador presiona el botón Renovar en la ventana ActualizarPréstamos	06	El Sistema presenta un mensaje de confirmación
07	El Administrador presiona el botón Sí del mensaje de confirmación.	08	El sistema actualiza el préstamo, genera un nuevo préstamo y presenta un mensaje de notificación.
09	El Administrador presiona el botón Aceptar del mensaje de notificación.		
10	El sistema continua con el paso 8 del curso normal de eventos	11	El caso de uso finaliza

Curso alterno C: Pago de cuotas por Préstamo

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón Administrar Pago de cuotas por Préstamo del menú Financiero en la ventana Principal .	02	El Sistema presenta la ventana Lista de préstamos pendientes con los datos del socio y de préstamos pendientes para la fecha actual.
03	El Administrador selecciona la cuota pendiente a cancelar de la tabla de prestamos pendientes y presiona el botón Guardar Aportación en la ventana Lista de prestamos pendientes .	04	El Sistema presenta un mensaje de confirmación
05	El Administrador presiona el botón Sí del mensaje de confirmación.	06	El sistema actualiza el pago de cuota de préstamo y presenta un mensaje de notificación.
07	El Administrador presiona el botón Aceptar del mensaje de notificación.	08	El caso de uso finaliza

Nombre:	Administrar Retiros	
Meta:	Registrar el retiro que realiza el socio del Fondo de Cesantía y Jubilación Administrativo de la UNL.	
Descripción:	El Socio presenta la solicitud de retiro del fondo de Cesantía o Jubilación para que sea aprobada por el Gerente y este a su vez ingresa al Sistema el retiro realizado por el socio.	
Precondiciones:	<ul style="list-style-type: none"> - El Gerente aprueba el retiro de fondos tanto para el de Cesantía como para el de Jubilación. - El Administrador debe ingresar al sistema - El Administrador debe estar autenticado con un nombre de usuario y contraseña. - El Administrador debe estar en la ventana Principal. - El Administrador debe presionar el Administrar Retiros del menú Financiero en la ventana Principal. - El Sistema activa la ventana Administrar Retiros. 	
Poscondiciones:	Registrar el retiro de fondos tanto para el Fondo de Cesantía como para el de Jubilación el sistema muestra la ventana Sistema del Fondo Administrativo de la UNL . <ul style="list-style-type: none"> - El estado del socio pasa de socio activo a socio pasivo después del retiro de jubilación. 	
Actores:	Administrador	
Tipo:	-Primario -Esencial	
Flujo Normal de eventos: Retiro de Cesantía.		
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA
01	El Administrador presiona el botón Realizar Retiro de la ventana Administrar Retiros .	02 El sistema presenta la ventana Realizar nuevo Retiro .
		03 El Sistema presenta la pestaña Retiros Fondo de Cesantía en la ventana Realizar nuevo Retiro .
04	El Administrador Ingresa el número de cuenta del socio y presiona enter en la ventana Realizar nuevo Retiro .	05 El sistema presenta los datos del socio en el panel Datos del Socio , los estados de cuenta en el panel Datos de Cuenta y el total del retiro permitido en la ventana Realizar nuevo Retiro .

06	El Administrador presiona el botón Imprimir en la ventana Realizar nuevo Retiro .	07	El Sistema presenta el Reporte de Comprobante de Retiro .
08	El Administrador presiona el icono de Imprimir de la ventana del Comprobante de Retiro .	09	El Sistema imprime el Comprobante de Retiro .
10	El Administrador cierra la ventana del Comprobante de Retiro presionando el icono de cierre de dicha ventana.	11	El Sistema presenta la pestaña Retiros Fondo de Cesantía en la ventana Realizar nuevo Retiro .
12	El Administrador presiona el botón Guardar en la ventana Realizar nuevo Retiro .	13	El sistema guarda el registro del retiro y presenta un mensaje de notificación.
14	El Administrador presiona el botón Aceptar del mensaje de notificación.	15	El caso de uso finaliza

Curso alternativo de eventos:

Curso alternativo A: Retiro de Jubilación

ACCIONES DEL ACTOR			
03	El Administrador selecciona la pestaña Retiros Fondo de Jubilación en la ventana Realizar nuevo Retiro .	04	El Sistema presenta la pestaña Retiros Fondo de Jubilación en la ventana Realizar nuevo Retiro .
05	El caso de uso continúa con el paso 04 del curso normal de eventos.		
07	El Administrador escoge la Jubilación requerida del panel Tipo de Jubilación en la ventana Realizar nuevo Retiro .	08	El sistema calcula el valor del retiro de acuerdo al tipo de jubilación en la ventana Realizar nuevo Retiro .
09	El caso uso continua con los pasos 06 del curso normal de eventos.		

Nombre:	Administrar Cuentas Contables
Meta:	Administrar las cuentas contables que se utilizan en la Contabilidad del Fondo de Cesantía y Jubilación Administrativo de la UNL
Descripción:	Administrar las cuentas contables que se utilizan en la Contabilidad dentro del Fondo Administrativo, las que forman parte durante todo el período contable.
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora presiona Administrar Plan De Cuentas del submenú Registrar Transacciones Diarias del menú Contabilidad en la ventana Principal. - El Sistema presenta la ventana Administrar Cuentas Contables.
Poscondiciones:	- El Sistema presenta la Administrar Cuentas Contables , después de haber creado, modificado o eliminado las cuentas contables.
Actores:	Administrador, Contadora.
Tipo:	-Primario -Esencial

Curso Normal de Eventos: Adicionar cuenta contable al plan de cuentas			
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Adicionar Cuenta Contable en la ventana Administrar Cuentas Contables .	02	El Sistema presenta la ventana Adicionar Cuenta Contable .
03	La Contadora ingresa la información en los campos activos en la ventana Adicionar Cuenta Contable .		
04	La Contadora escoge Activo del panel " cuenta del: " en la ventana Adicionar Cuenta Contable .		
05	La contadora escoge De Detalle del panel " Esta cuenta es: " en la ventana Adicionar Cuenta Contable .		
06	La Contadora presiona el botón Guardar en la ventana Adicionar Cuenta Contable .	07	El Sistema guarda la Cuenta Contable y presenta un mensaje de confirmación.

08	La Contadora presiona el botón Sí del mensaje de confirmación.	09	El caso de uso finaliza.
----	---	----	--------------------------

Curso alternativo A: Modificar Cuenta Contable

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La contadora selecciona la Cuenta Contable y presiona el botón Modificar Cuenta Contable en la ventana Administrar Cuentas Contables .	02	El Sistema presenta los datos de la Cuenta Contable en la ventana Actualizar Cuenta Contable .
03	El caso de uso continúa con el paso 3 del curso normal de eventos.		

Curso alternativo B: Eliminar Cuenta Contable

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La contadora selecciona la Cuenta Contable y presiona el botón Eliminar Cuenta Contable en la ventana Administrar Cuentas Contables .	02	El Sistema presenta un mensaje de confirmación para borrar la Cuenta Contable
03	La contadora presiona el botón Sí del mensaje de confirmación.	04	El Sistema borra la cuenta contable
05	El caso finaliza		

Nombre:	Ingresar Transacciones diarias		
Meta:	Ingresar las transacciones contables que lleva a cabo el fondo de Cesantía y Jubilación Administrativo de la UNL		
Descripción:	Ingresar las transacciones diarias contables dentro del fondo administrativo, transacciones que darán inicio al proceso contable dentro del fondo.		
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora selecciona la opción Asientos contables del submenú Transacciones Diarias del menú Contabilidad en la ventana Principal. - El Sistema presenta la ventana Administrar Asientos Contables. 		
Poscondiciones:	- El Sistema presenta la Administrar Asientos Contables , después de haber creado, modificado o eliminado asientos contables.		
Actores:	Administrador, Contadora		
Tipo:	-Primario -Esencial		
Flujo Normal de eventos: Asientos Contables			
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Crear Asiento Contable en la ventana Administrar Asientos Contables .	02	El Sistema presenta la ventana Ingresar Nuevo Asiento Contable .
03	La Contadora llena los campos habilitados en el panel Datos Principales en la ventana Ingresar Nuevo Asiento Contable .		
04	La Contadora debe ingresar el código de la cuenta contable en el panel Transacciones en la ventana Ingresar Nuevo Asiento Contable y presiona enter .	05	El Sistema presenta el nombre de la Cuenta Contable en la ventana Ingresar Nuevo Asiento Contable .
06	La Contadora llena los campos habilitados en el panel Transacciones y presiona el botón Agregar de dicho panel.	07	El Sistema agrega el ítem de la transacción al panel Asientos Contable en la ventana Ingresar Nuevo Asiento Contable .

08	La Contadora repite paso 4 – 7 del curso normal.	09	El Sistema suma los valores ingresados.
10	La Contadora presiona el botón Registrar Asiento en la ventana Ingresar Nuevo Asiento Contable .	11	El Sistema si no existe diferencia entre los valores guarda el Asiento Contable y presenta un mensaje de confirmación.
12	La Contadora presiona el botón Aceptar del mensaje de confirmación.	13	El Caso de Uso finaliza

Curso alternativo de eventos:

Curso alternativo A: Modificar Asiento Contable

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Modificar Asiento Contable en la ventana Administrar Asientos Contables .	02	El Sistema presenta la ventana Modificar Asiento Contable .
03	La Contadora ingresa el número del Asiento Contable y presiona el botón Modificar Asiento en la ventana Modificar Asiento Contable .	04	El Sistema presenta los datos del asiento contable en el panel Asientos Contables la ventana Modificar Asiento Contable .
05	La Contadora selecciona el ítem del Asiento Contable y presiona el botón Modificar Ítem en la ventana Modificar Asiento Contable .	06	El Sistema presenta los datos del ítem del Asiento Contable en la ventana Modificar Ítem Seleccionado del Libro Diario .
07	La Contadora modifica los campos de ítem del Asiento Contable y presiona el botón modificar en la ventana Modificar Ítem Seleccionado del Libro Diario .	08	El Sistema actualiza la información del asiento contable y presenta un mensaje de confirmación.
09	La Contadora presiona el botón Aceptar del mensaje de confirmación.	10	El Sistema presenta la ventana Modificar Asiento Contable .
11	La Contadora presiona el botón Salir en la ventana Modificar Asiento Contable .	12	El caso de uso finaliza

Curso alternativo B: Eliminar Asiento Contable

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Eliminar Asiento Contable en la ventana Administrar Asientos Contables .	02	El Sistema presenta la ventana Eliminar Asiento Contable .
03	La Contadora ingresa en número del Asiento Contable y presiona el botón Eliminar Asiento en la ventana Eliminar Asiento Contable .	04	El Sistema presenta un mensaje de confirmación para borrar el Asiento Contable.
05	La Contadora presiona el botón Sí del mensaje de confirmación.	06	El Sistema borra el asiento Contable.
		07	El caso de uso finaliza

Curso alternativo C: Generar Libro diario

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora selecciona el ítem Listar Libro Diario del submenú Libro Diario del submenú Transacciones Diarias en el menú Contabilidad .	02	El Sistema muestra la ventana Revisar Libro Diario .
03	La Contadora presiona el botón Imprimir de la ventana Revisar libro Diario .	04	El Sistema presenta el reporte de las transacciones realizadas en la ventana Reporte Libro Diario .
05	La Contadora presiona el icono de Imprimir de la ventana Reporte Libro Diario .	06	El Sistema imprime el Reporte Libro Diario .
07	La Contadora cierra la ventana del Reporte Libro Diario presionando el icono de cierre de dicha ventana.	08	El Sistema presenta la ventana Revisar Libro Diario .
		09	El caso de uso finaliza.

Curso alternativo D: Libro Mayor por cuenta

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora selecciona el ítem Elegir Cuenta a Mayorizar del submenú Transacciones Diarias en el Menú Contabilidad .	02	El Sistema muestra la ventana Generar Libro Mayor .
03	La Contadora ingresa el código de la Cuenta Contable y presiona el botón Procesar en la ventana Generar Libro Mayor .	04	El Sistema presenta las transacciones realizadas por la cuenta en el panel Detalle Mayorización en la ventana Generar Libro Mayor .
		05	El Sistema suma y presenta los valores del detalle de las transacciones en el panel Detalle Mayorización en la ventana Generar Libro Mayor .
06	La Contadora presiona el botón Imprimir en la ventana Generar Libro Mayor .	07	El Sistema presenta el Informe de la mayorización de la cuenta contable en la ventana Reporte Libro Mayor .
08	La Contadora presiona el icono de Imprimir de la ventana Reporte Libro Mayor .	09	El Sistema imprime el Reporte Libro Mayor .
10	La Contadora cierra la ventana del Reporte Libro Mayor presionando el icono de cierre de dicha ventana.		
11	La contadora presiona el botón Guardar en la ventana Generar Libro Mayor .	12	El Sistema guarda el libro mayor de la Cuenta y presenta un mensaje de notificación.
13	La Contadora presiona el botón NO del mensaje de confirmación.	14	El caso de uso finaliza.

Nombre:	Generar Balances Contables
Meta:	Generar los diferentes balances Contables que permitirán saber el estado económico del fondo Administrativo de la UNL.
Descripción:	Una vez realizada la mayorización se pueden generar los diferentes Balances Contables que permitirán saber la situación económica de Fondo de Cesantía y Jubilación administrativo de la UNL.
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora debe ubicar el menú contabilidad de la ventana Principal. - La contadora se debe ubicar en el submenú Generar Balances Contables del menú Contabilidad en la ventana Principal.
Poscondiciones:	- Después de Generar los Balances Contables el sistema muestra la ventana Sistema del Fondo Administrativo de la UNL .
Actores:	Administrador, Contadora
Tipo:	-Primario -Esencial

Flujo Normal de eventos: Generar Balance General

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora selecciona el ítem Balance General del submenú Generar Balances Contables del menú Contabilidad .	02	El Sistema presenta la ventana Generar Balance General .
03	La Contadora ingresa el nombre de la contabilidad y presiona el botón Generar en la ventana Generar Balance General .	04	El Sistema genera el Balance General y presenta el total en la ventana Generar Balance General .
05	La Contadora presiona el botón Imprimir de la ventana Generar Balance General .	06	El Sistema presenta la ventana del reporte del Balance General .
07	La Contadora presiona el icono de Imprimir de ventana del reporte del Balance General .	08	El Sistema imprime el Reporte del Balance General .
09	La Contadora cierra la ventana del reporte del Balance General presionando el icono de cierre de dicha ventana.	10	El caso de uso finaliza.

Curso alternativo de eventos:

Curso alternativo A: Generar Balance de Comprobación

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora selecciona el ítem Balance de Comprobación del submenú Generar Balances Contables del menú Contabilidad .	02	El Sistema muestra la ventana Generar Balance de Comprobación .
03	La Contadora ingresar el nombre del diario de contabilidad y presiona el botón Generar en la ventana Generar Balance de Comprobación .	04	El Sistema genera el Balance de Comprobación y presenta los datos del balance en el panel Resultado del Balance en la ventana Generar Balance de Comprobación .
05	La Contadora presiona el botón Imprimir de la ventana Generar Balance de Comprobación .	06	El Sistema presenta el Balance Comprobación la ventana del reporte del Balance de Comprobación .
07	La Contadora presiona el icono de Imprimir de la ventana del reporte del Balance de Comprobación .	08	El Sistema imprime el Balance de Comprobación .
09	La Contadora cierra la ventana del reporte del Balance de Comprobación presionando el icono de cierre de dicha ventana.	10	El caso de uso finaliza.

Nombre:	Generar Estado de Situación Final
Meta:	Generar el Estado de Situación Final del Fondo que permitirá saber el Estado económico del Fondo Administrativo de la UNL.
Descripción:	Una vez realizada la mayorización se pueden generar el Estado de Pérdidas y Ganancias y se puede realizar el cierre del Ejercicio Contable.
Precondiciones:	<ul style="list-style-type: none"> - La contadora debe estar autenticado. - La Contadora debe estar dentro del sistema. - La contadora debe seleccionar el submenú Estado de Situación Final del menú Contabilidad en la ventana Principal. - La Contadora debe seleccionar el ítem Pérdidas y Ganancias del submenú Estado de Situación Final. - La Contadora debe estar en la ventana Generar Estado de Pérdidas y Ganancias.
Poscondiciones:	- El Sistema muestra la ventana Sistema del Fondo Administrativo de la UNL .
Actores:	Administrador, Contadora
Tipo:	-Primario -Esencial
Flujo Normal de eventos: Estado de Pérdidas y Ganancias	

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora ingresa el nombre de la contabilidad y presiona el botón Generar de la ventana Generar Estado de Pérdidas y Ganancias .	02	El Sistema genera y presenta los datos de perdidas y ganancias en la ventana Generar Estado de Pérdidas y Ganancias .
03	La Contadora presiona el botón Imprimir de la ventana Generar Estado de Pérdidas y Ganancias .	04	El Sistema presenta la ventana del Reporte del Balance de Pérdidas y Ganancias .
05	La Contadora presiona el icono de Imprimir de la ventana del Reporte del Balance de Pérdidas y Ganancias .	06	El Sistema imprime el Balance de Pérdidas y Ganancias .
07	La Contadora cierra la ventana del Reporte del Balance de Pérdidas y Ganancias presionando el icono de cierre de dicha ventana.	08	El caso de uso finaliza.

Curso alternativo de eventos:

Curso alternativo A: Cierre del Período

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora selecciona el ítem Cierre del periodo Contable del submenú Estado de Situación Final .	02	El Sistema muestra la ventana Generar Cierre del Período Contable .
03	La Contadora selecciona los reportes que requiera del cierre de período en el panel "Obtener Reportes de:" de la ventana Generar Cierre del Período Contable .		
04	La Contadora presiona el botón Procesar de la ventana Generar Cierre del Período Contable .	05	El Sistema genera el cierre del período económico y presenta los reportes.
06	La Contadora presiona el icono de Imprimir de las ventanas de cada reporte Generado .	07	El Sistema imprime cada reporte generado.
08	La Contadora cierra cada ventana de los reportes presionando el icono de cierre de dichas ventanas.	09	El caso de uso finaliza.

Nombre:	Administrar Rol de pagos
Meta:	Administrar el rol de pagos de los socios que acceden al servicio de pago de sueldo mediante el fondo de Cesantía y Jubilación Administrativo de la UNL
Descripción:	Mantener los roles de pago de los socios que acceden al servicio de pago de sueldos mediante y el fondo y la emisión del cheque de sueldo.
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora debe seleccionar el submenú Administrar Rol de Pagos del menú Contabilidad en la ventana Principal. - El Sistema presenta la ventana Administrar Rol de Pagos.
Poscondiciones:	- El Sistema presenta la ventana Emitir cheques .
Actores:	Administrador, Contadora
Tipo:	-Primario -Esencial

Flujo Normal de eventos: Emitir Cheque

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Emitir Cheque en la ventana Administrar Rol de Pagos .	02	El Sistema muestra la ventana Emitir cheques .
		03	El sistema presenta la pestaña chequera en la ventana Emitir cheques .
04	La Contadora presiona el botón Buscar para seleccionar el socio en la pestaña Chequera .	05	El Sistema presenta la ventana Lista De Socios del Rol de Pago .
06	La Contadora selecciona el Socio y presiona el botón Seleccionar en la ventana Lista De Socios del Rol de Pago	07	El Sistema cierra la ventana Lista De Socios del Rol de Pago y llena con el Nombre del socio el campo Páguese a la Orden de: de la pestaña Chequera .
08	La Contadora llena los campos activos y selecciona el estado del cheque en la pestaña Chequera .		

09	La Contadora presiona el botón Aceptar de la pestaña Chequera .	10	El Sistema guarda el cheque y lo presenta en la tabla Relación de Cheques de la pestaña Chequera .
11	La contadora presiona el botón Imprimir en la pestaña chequera en la ventana Emitir Cheque .	12	El sistema presenta el cheque creado.
13	La Contadora presiona el icono de Imprimir de la ventana del cheque .	14	El Sistema imprime el cheque.
15	La Contadora cierra la ventana del cheque presionando el icono de cierre de dicha ventana.	16	El caso de uso finaliza.

Curso Alterno A: Auxiliar de Bancos

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
03	La Contadora selecciona la pestaña Auxiliar de Bancos en la ventana Emitir Cheques .	04	El Sistema presenta la pestaña Auxiliar de Bancos en la ventana Emitir Cheques .
05	La Contadora llena los campos del panel Buscar Cheque y presiona el botón Buscar en la pestaña Auxiliar de Bancos en la ventana Emitir Cheques .	06	El sistema muestra en el panel Lista de cheques los cheques emitidos.
07	El caso de uso finaliza.		

Nombre:	Administrar Inventario		
Meta:	Registrar todos los bienes del fondo de Cesantía y Jubilación Administrativo de la UNL		
Descripción:	Registrar los bienes que adquiere y posee el fondo de Cesantía		
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora debe seleccionar el menú Contabilidad en la ventana Principal. - La Contadora selecciona el ítem Administrar Inventario. - El Sistema presenta la ventana Administrar Inventario 		
Poscondiciones:	- El Sistema muestra la ventana Principal .		
Actores:	Administrador, Contadora		
Tipo:	-Primario -Esencial		
Flujo Normal de eventos: Crear Producto			
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Agregar Producto en la ventana Administrar Inventario .	02	El Sistema muestra la ventana Ingresar producto al Inventario .
03	La Contadora ingresa los datos en los campos activos de la ventana Ingresar producto al Inventario .	04	El Sistema calcula el costo total del producto y lo presenta en la ventana Ingresar producto al Inventario .
05	La Contadora selecciona el tipo de producto en el panel Producto de la ventana Ingresar producto al Inventario .		
06	La Contadora presiona el botón Aceptar de la ventana Ingresar producto al Inventario .	07	El Sistema guarda el producto y presenta un mensaje de confirmación
08	La contadora presiona el botón Aceptar del mensaje de confirmación.	09	El sistema presenta la ventana Administrar Inventario .
		10	El caso de uso finaliza.

Curso alternativo de eventos:

Curso alternativo A: Modificar Producto

‘ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora busca el producto y lo selecciona en la ventana Administrar Inventario.		
02	La Contadora presiona el botón Modificar Producto de la ventana Administrar Inventario.	03	El Sistema presenta el producto en la ventana Modificar producto del Inventario
04	El caso de uso continúa con el paso 03 del curso normal de eventos.		

Curso alternativo B: Eliminar Producto

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora busca el producto y lo selecciona en la ventana Administrar Inventario.		
02	La Contadora presiona el botón Eliminar Producto.	03	El Sistema presenta un mensaje de confirmación.
04	La contadora presiona SI en el mensaje de confirmación.	05	El sistema borra el producto del inventario y presenta un mensaje de notificación.
06	La contadora presiona Aceptar en el mensaje de notificación.		
07	El caso de uso finaliza		

Curso alternativo C: Crear el inventario

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Agregar Inventario en la ventana Administrar Inventario.	02	El Sistema presenta la ventana Ingresar un nuevo Inventario.
03	La contadora llena los campos disponibles del Inventario y presiona el botón Guardar en la ventana Ingresar un Nuevo Inventario.	04	El sistema guarda el inventario y presenta un mensaje de notificación
05	La contadora presiona Aceptar en el mensaje de notificación.	06	El sistema presenta la ventana Administrar Inventario.
07	El caso de uso finaliza		

Nombre:	Administrar Contabilidad		
Meta:	Crear Períodos de tiempo que permita llevar la contabilidad de la empresa		
Descripción:	Crear la fecha de inicio y la fecha de cierre del ejercicio contable.		
Precondiciones:	<ul style="list-style-type: none"> - La Contadora debe estar dentro del sistema. - La contadora debe estar autenticado. - La contadora debe seleccionar el menú Administración en la ventana Principal. - La Contadora selecciona el ítem Administrar Contabilidad del Sistema. - El Sistema presenta la ventana Administrar Contabilidad. 		
Poscondiciones:	- El Sistema muestra la ventana Principal .		
Actores:	Administrador, Contadora		
Tipo:	-Primario -Esencial		
Flujo Normal de eventos: Agregar Contabilidad			
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora presiona el botón Crear Contabilidad en la ventana Administrar Contabilidad .	02	El Sistema muestra la ventana Adicionar Período de Contabilidad .
03	La Contadora ingresa los datos en los campos activados de la ventana Adicionar Período de Contabilidad .		
04	La Contadora presiona el botón Guardar de la ventana Adicionar Período de Contabilidad .	05	El Sistema guarda el período contable y presenta un mensaje de información.
06	La contadora presiona el botón Aceptar del mensaje de información.	07	El sistema presenta la ventana Administrar Contabilidad .
		08	El caso de uso finaliza.

Curso alternativo de eventos:

Curso alternativo A: Modificar Contabilidad

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora busca la contabilidad y lo selecciona, presiona el botón Modificar Contabilidad en la ventana Administrar Contabilidad .	02	El Sistema presenta la contabilidad configurada en la ventana Modificar Contabilidad .
03	El caso de uso continúa con el paso 03 del curso normal de eventos.		

Curso alternativo B: Eliminar Contabilidad

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	La Contadora busca la contabilidad y lo selecciona, presiona el botón Eliminar Contabilidad en la ventana Administrar Contabilidad .	02	El Sistema presenta un mensaje de confirmación.
03	La contadora presiona SI en el mensaje de Confirmación.	04	El sistema borra la Contabilidad deseada y presenta un mensaje de notificación.
05	El Administrador presiona Aceptar en el mensaje de notificación.		
06	El caso de uso finaliza		

Nombre:	Administrar Intereses
Meta:	Ingresar al sistema las diferentes tasas de interés que se utilizan dentro de cada una de las transacciones que se realizan dentro del sistema.
Descripción:	El usuario ingresa los valores de los intereses que se utilizan en las diferentes transacciones
Precondiciones:	<ul style="list-style-type: none"> - El Administrador debe estar dentro del sistema. - El Administrador debe estar autenticado. - El Administrador debe seleccionar el menú Financiero en la ventana Principal. - El Administrador selecciona el ítem Administrar Interés - El Sistema presenta la ventana Administrar Intereses
Poscondiciones:	- El Sistema muestra la ventana Principal .
Actores:	Administrador
Tipo:	-Primario -Esencial

Flujo Normal de eventos: Ingresar Tipos de Interés

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón Tipos de Interés en la ventana Administrar Intereses .	02	El Sistema muestra la ventana Ingresar Tipos de Interés .
03	El Administrador ingresa los datos en los campos activos de la ventana Ingresar Tipos de Interés .		
04	El Administrador presiona el botón Guardar en la ventana Ingresar Tipos de Interés .	05	El Sistema guarda el nuevo tipo de interés y presenta un mensaje de confirmación.
06	El Administrador presiona el botón Si del mensaje de confirmación.	07	El caso de uso finaliza.

Curso alternativo de eventos:

Curso alternativo A: Eliminar Tipos de Interés

ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador selecciona el tipo de interés y presiona el botón Eliminar en la ventana Ingresar Tipos de Intereses .	02	El Sistema presenta un mensaje de Confirmación.
03	El Administrador presiona el botón Si del mensaje de Confirmación.	04	El Sistema elimina el tipo de interés.
		05	El caso de Uso finaliza

Curso alternativo B: Ingresar Tasas de Interés

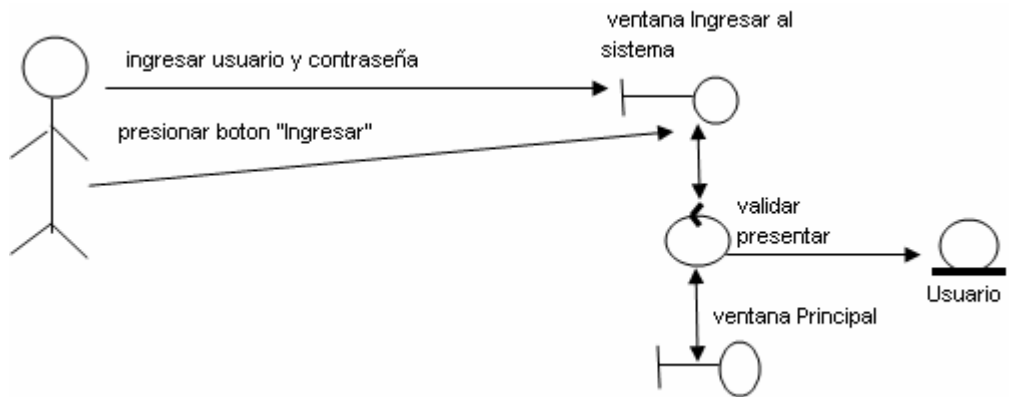
ACCIONES DEL ACTOR		RESPUESTA DEL SISTEMA	
01	El Administrador presiona el botón Tasas de Interés en la ventana Administrar Intereses .	02	El Sistema presenta la ventana Ingresar Tasas de Interés .
03	El Administrador ingresa el tipo de interés y presiona el botón Presentar en la ventana Ingresar Tasas de Interés .	04	El sistema presenta los datos del interés en la ventana Ingresar Tasas de Interés .
05	El Administrador ingresa los datos en los campos activos del Panel Agregar Tasas de Interés y presiona el botón Guardar en la ventana Ingresar Tasas de Interés .	06	El Sistema guarda los valores del interés y presenta un mensaje de confirmación
07	El Administrador presiona Aceptar en el mensaje de notificación.	08	El caso de uso finaliza

DIAGRAMAS DE ROBUSTEZ

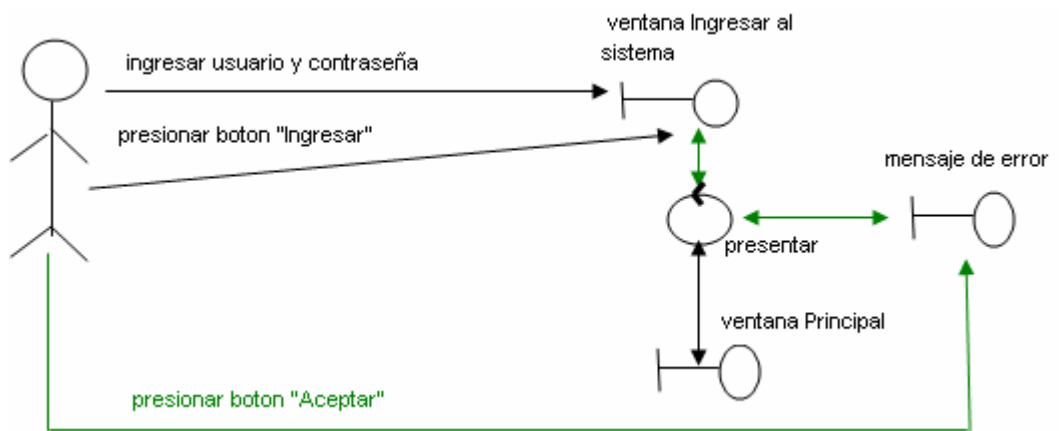
Análisis de Robustez del Sistema

A continuación se describen de manera gráfica todos los diagramas de robustez para cada use case encontrado.

CASO DE USO: AUTENTICAR USUARIO
Curso Normal de Eventos: Ingresar al Sistema

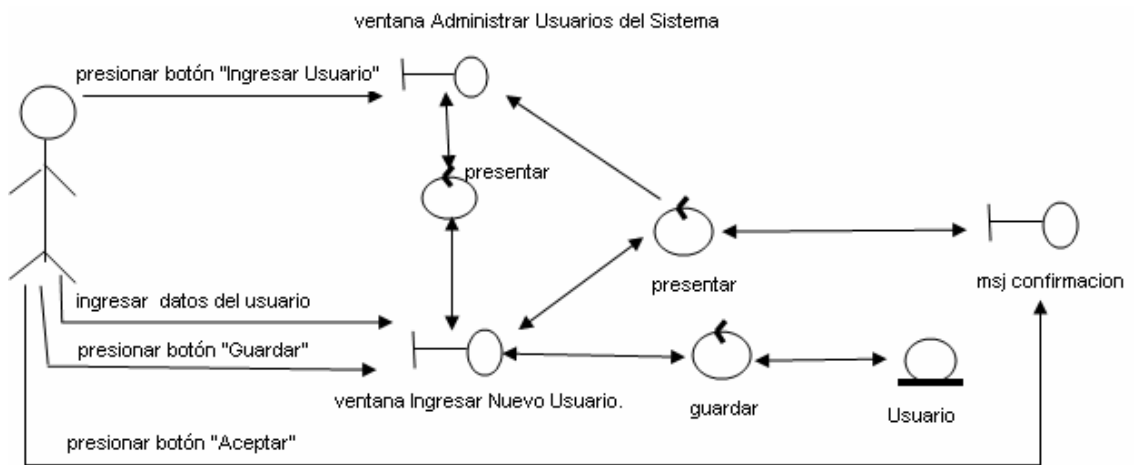


Curso Alterno A: nombres mal ingresados

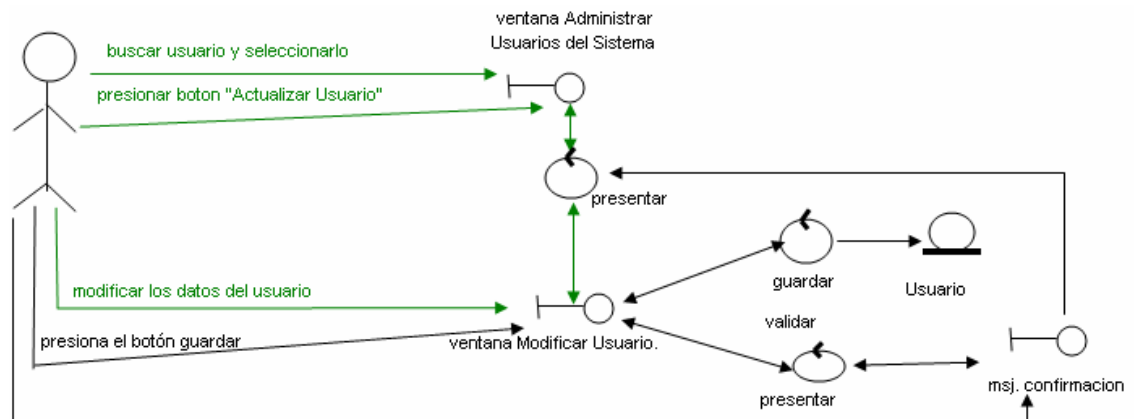


CASO DE USO: ADMINISTRAR USUARIOS DEL SISTEMA

Curso Normal de Eventos: Agregar Usuario

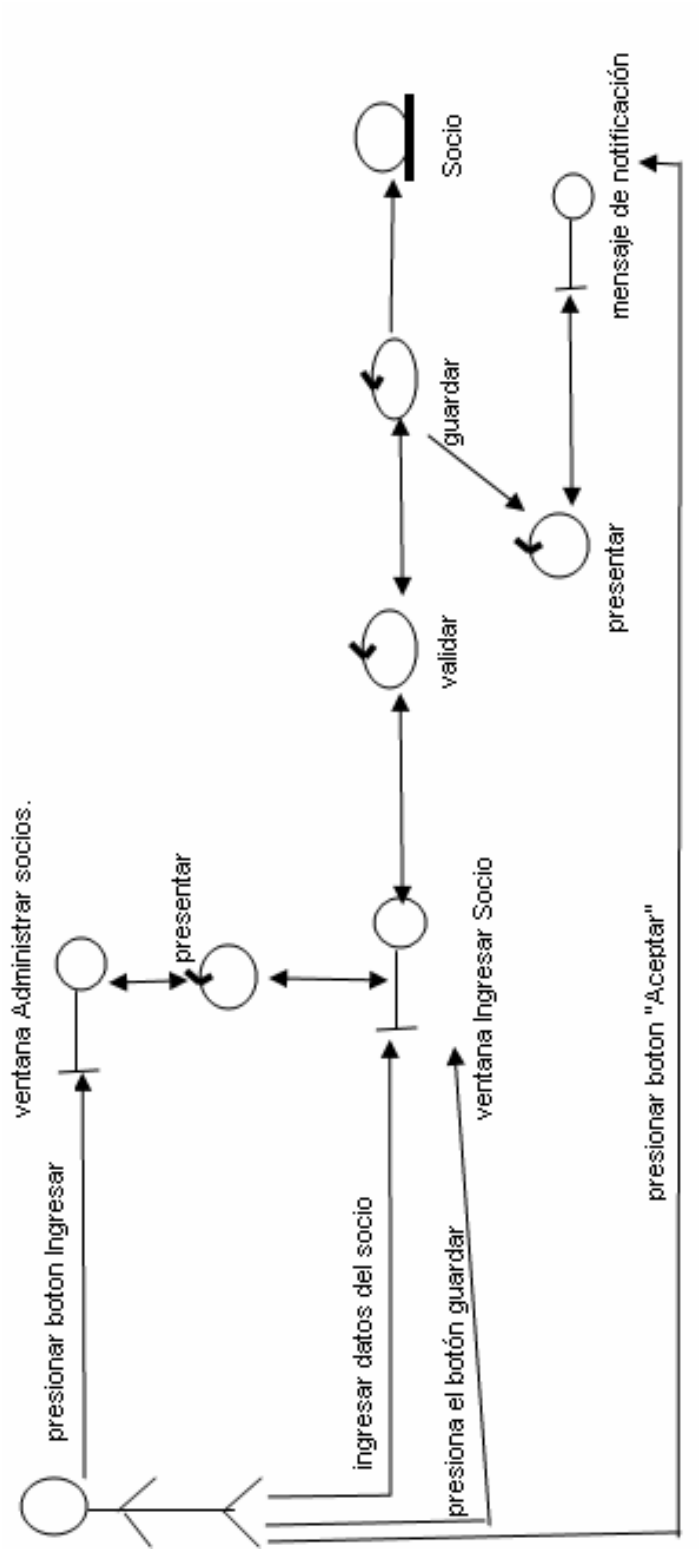


Curso Alternativo A: Actualizar Usuario

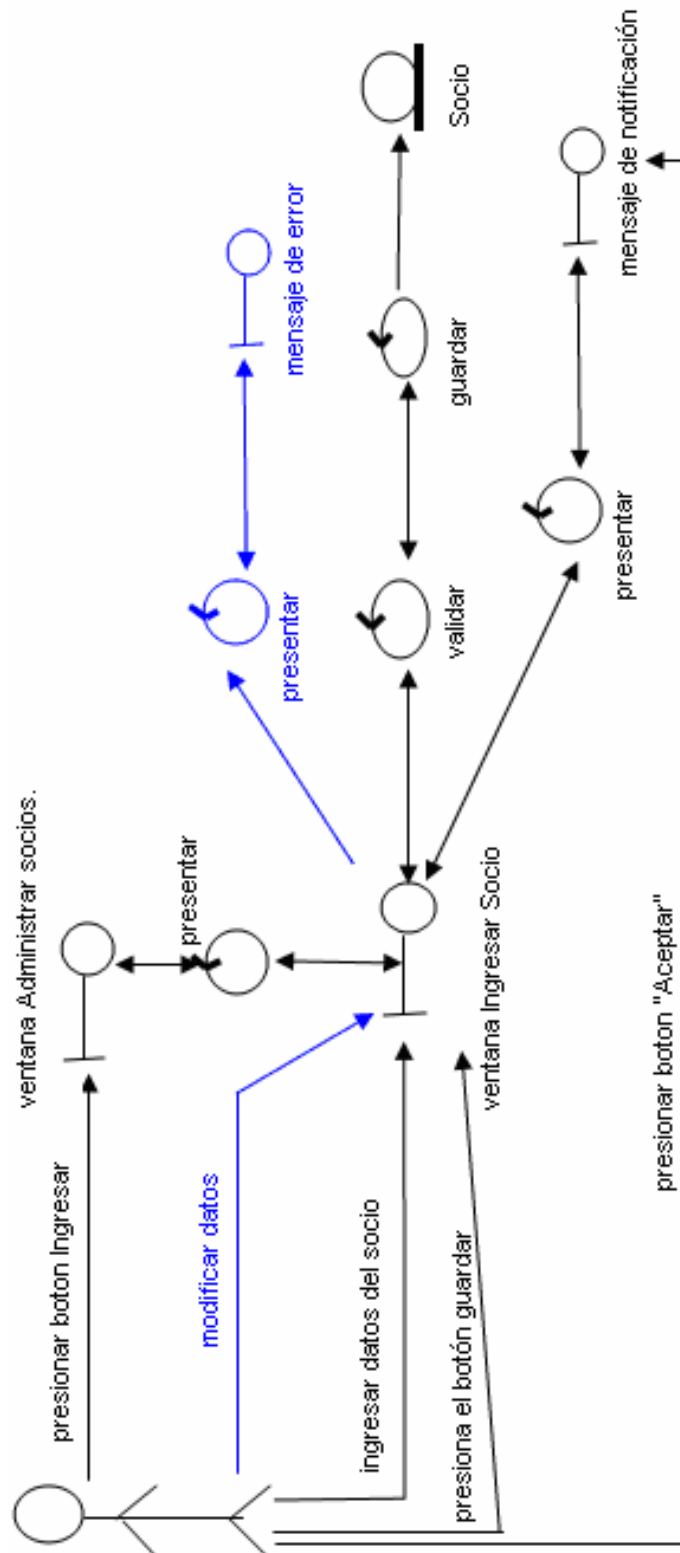


CASO DE USO: ADMINISTRAR SOCIOS

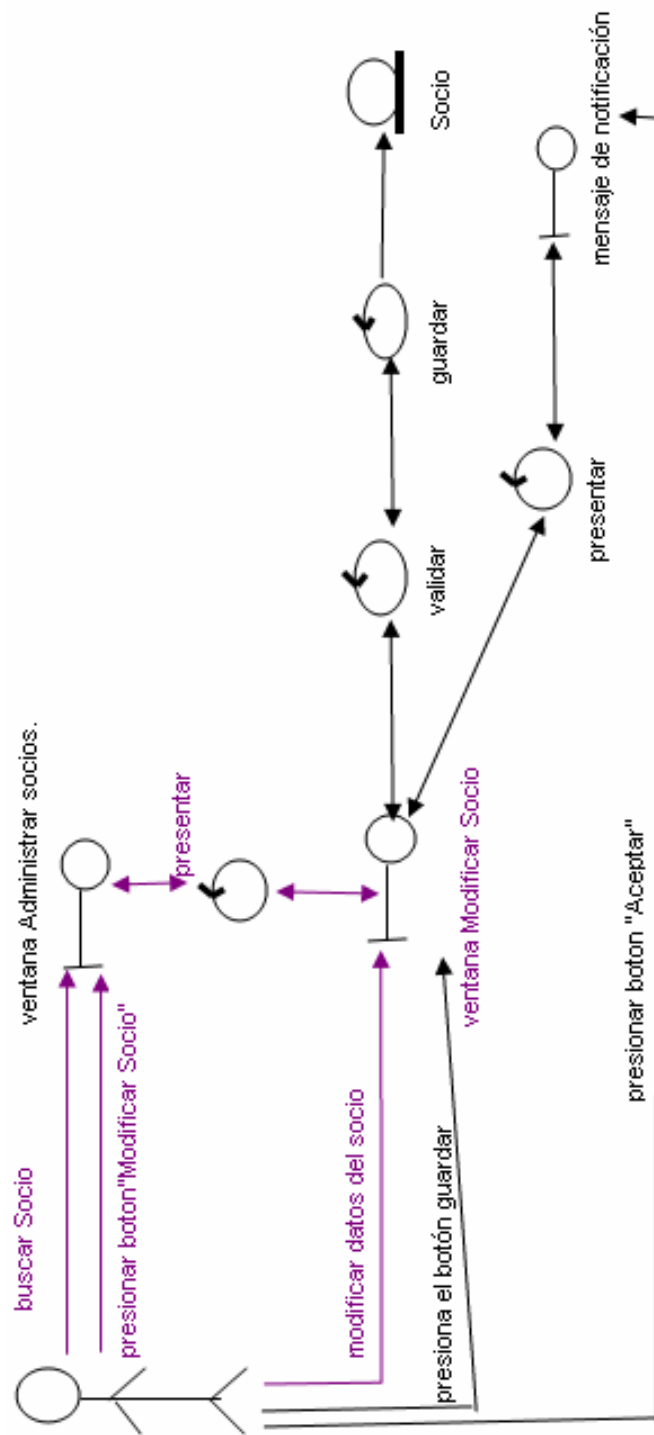
Curso Normal de eventos: Ingresar Socio



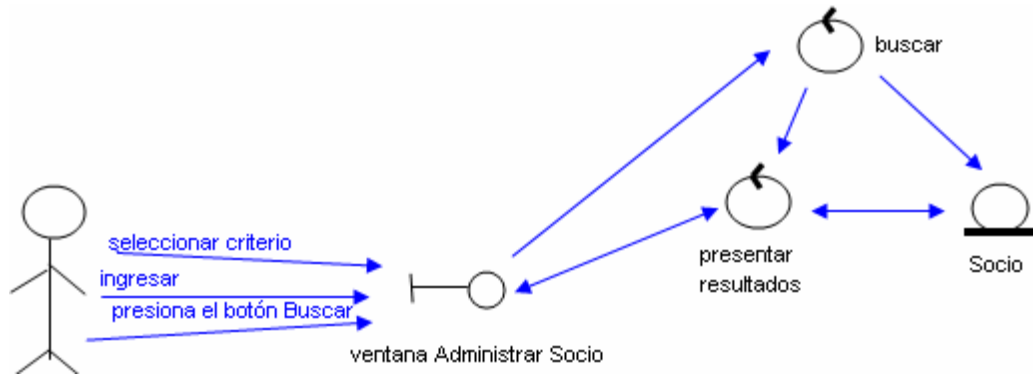
Curso alterno A: Datos mal ingresados



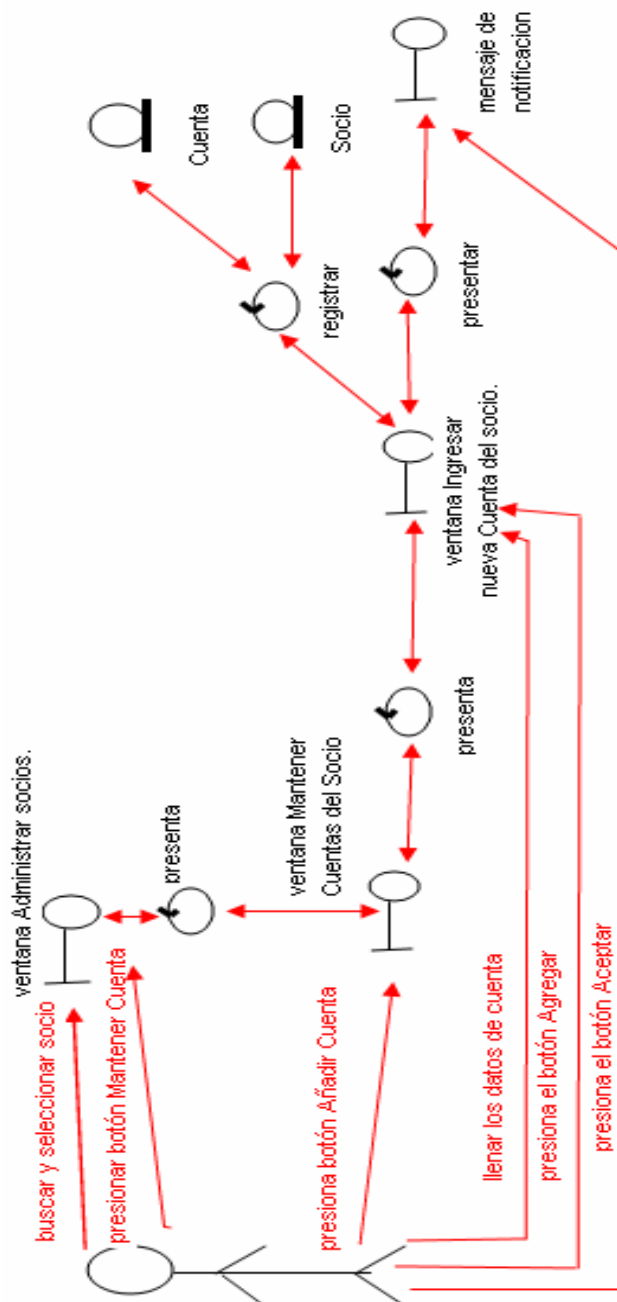
Curso alternativo B: Modificar datos del socio



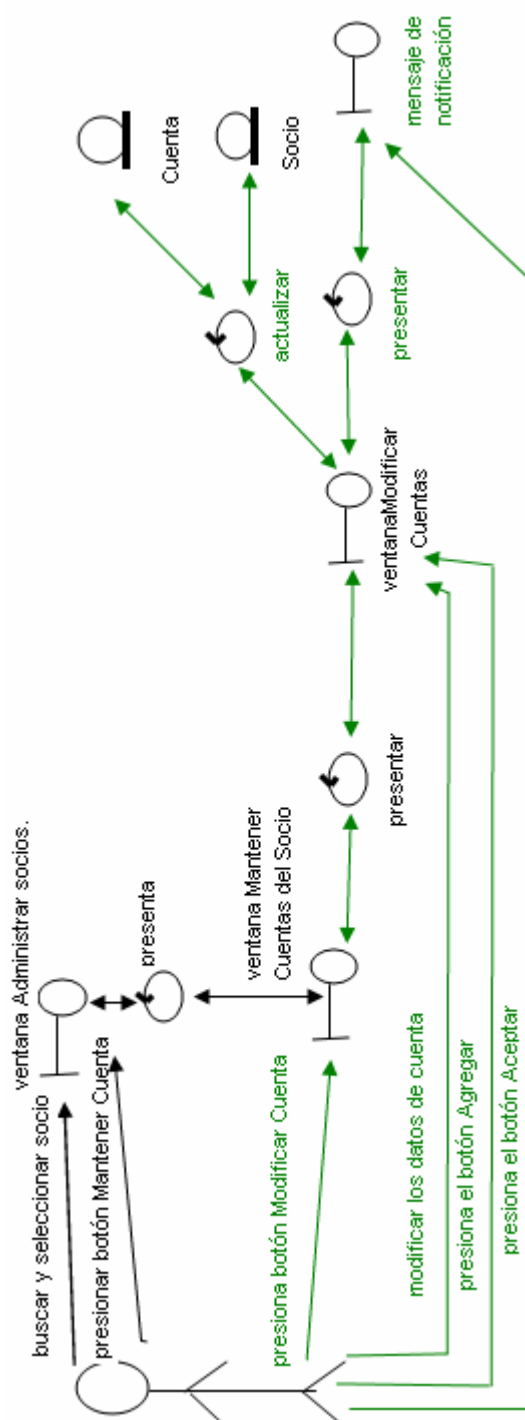
Curso Alterno C: Buscar Socio



Curso Alterno D: Asignar Cuentas al Socio

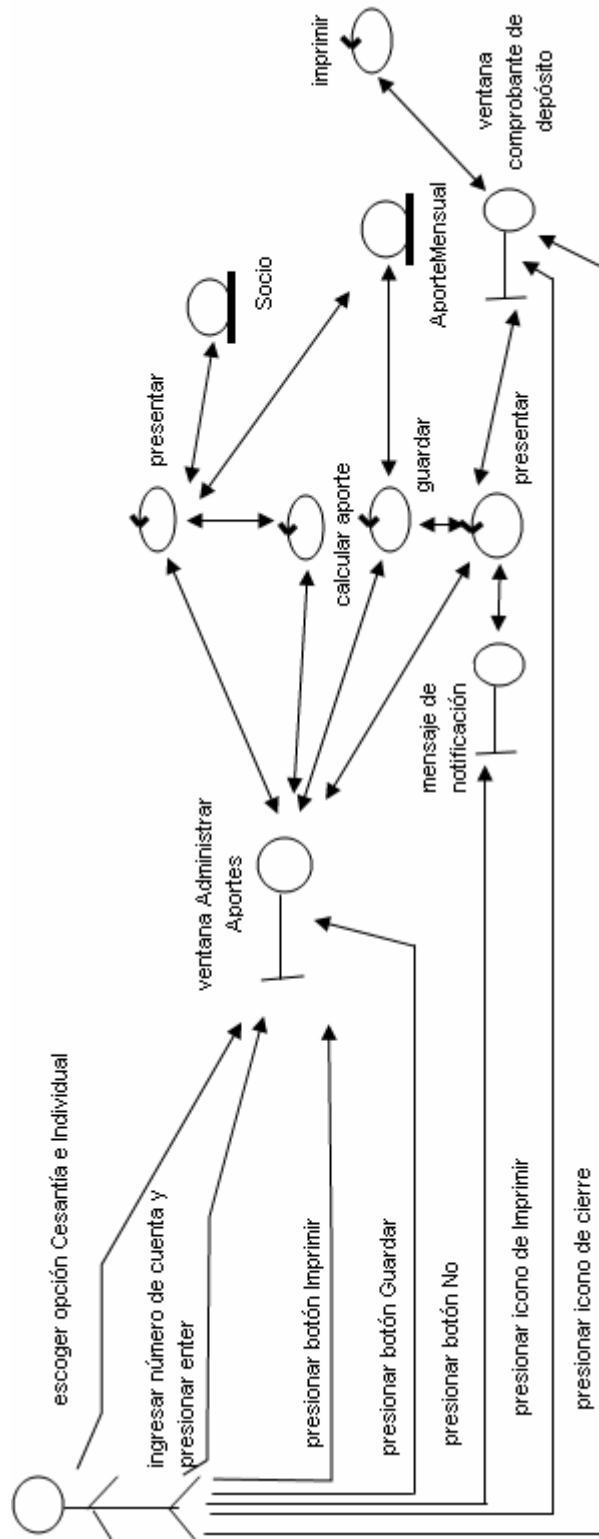


Curso Alterno E: Modificar Cuentas al Socio

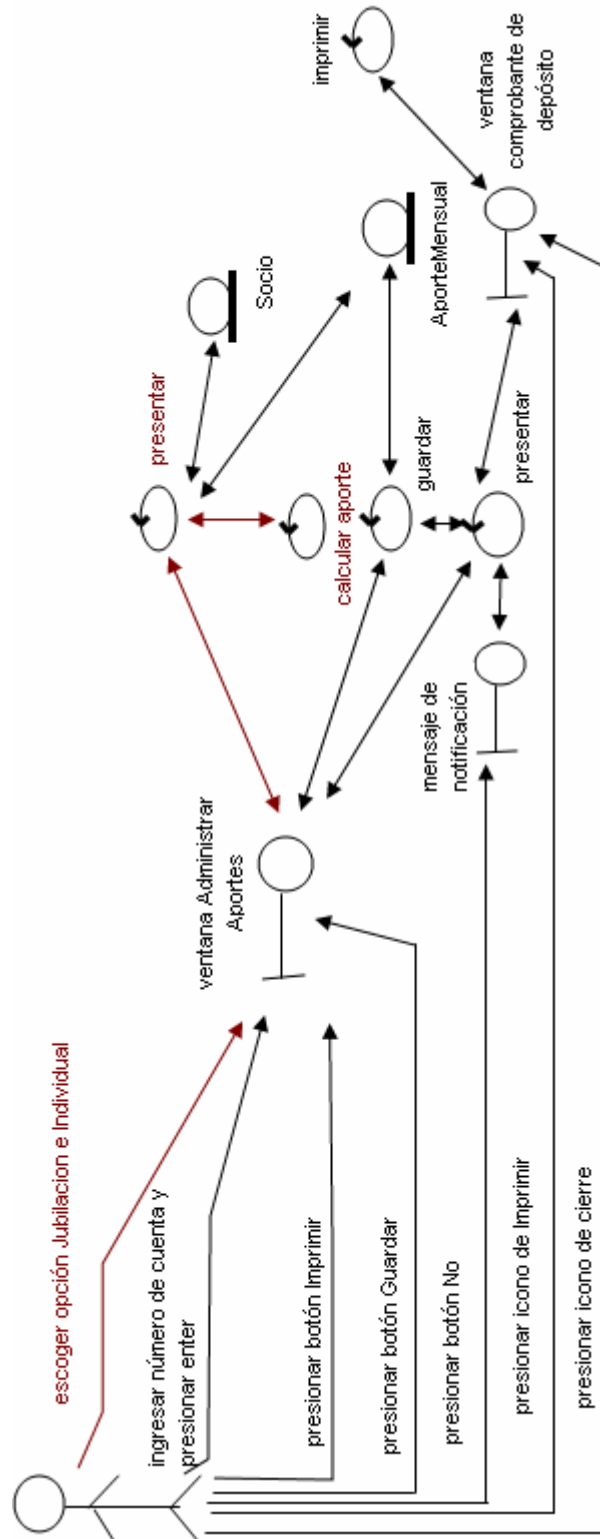


CASO DE USO: ADMINISTRAR APORTES

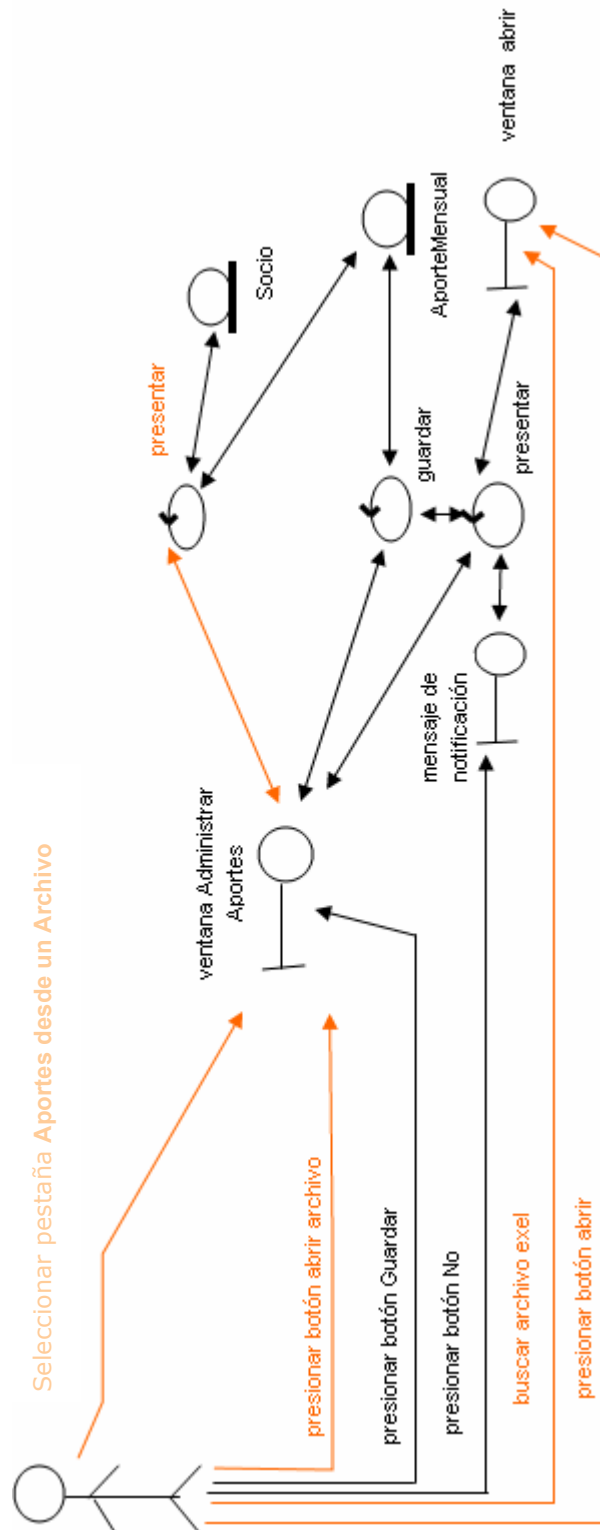
Curso Normal de Eventos: Aportes Cesantía



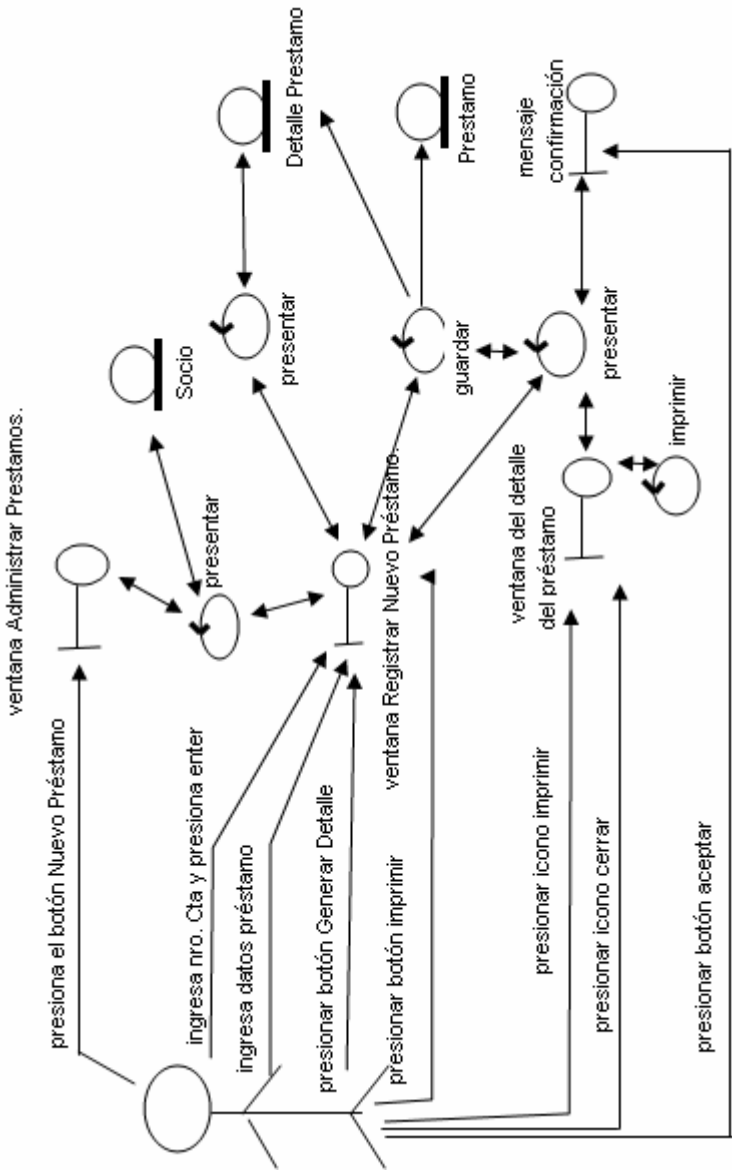
Curso Alterno A: Aportes Jubilación



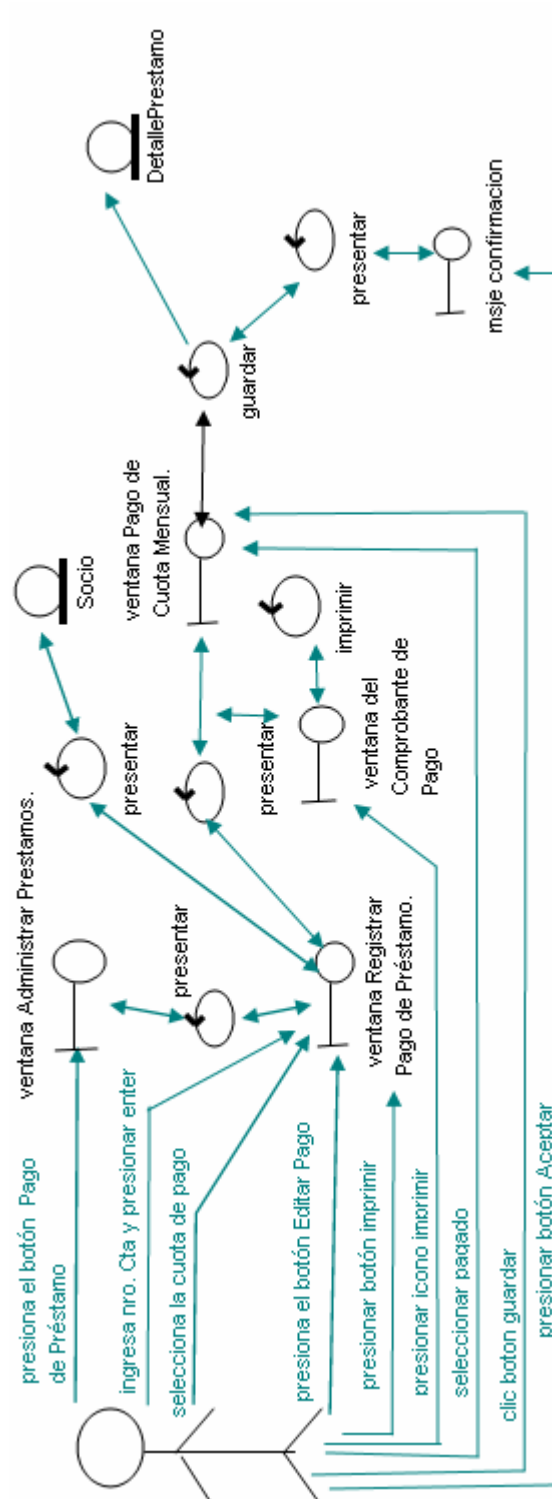
Curso Alterno B: Aportes Grupales



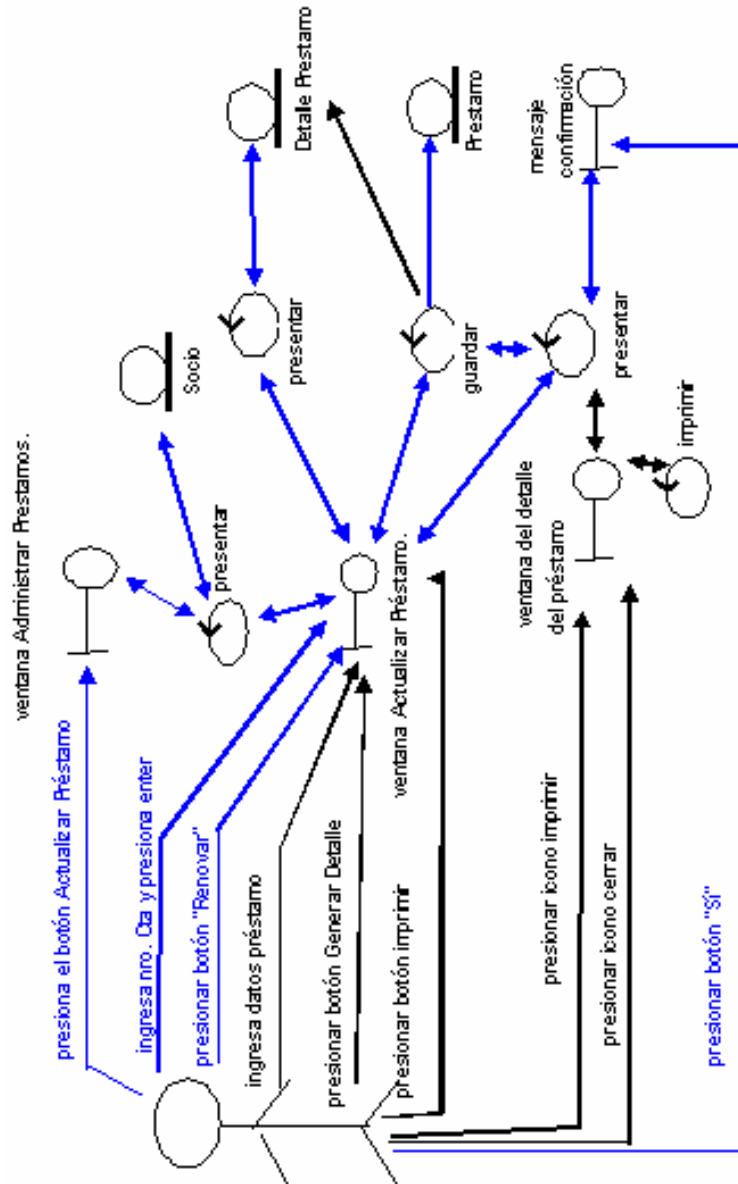
CASO DE USO: ADMINISTRAR PRÉSTAMO
Curso Normal de Eventos: Registrar Préstamo



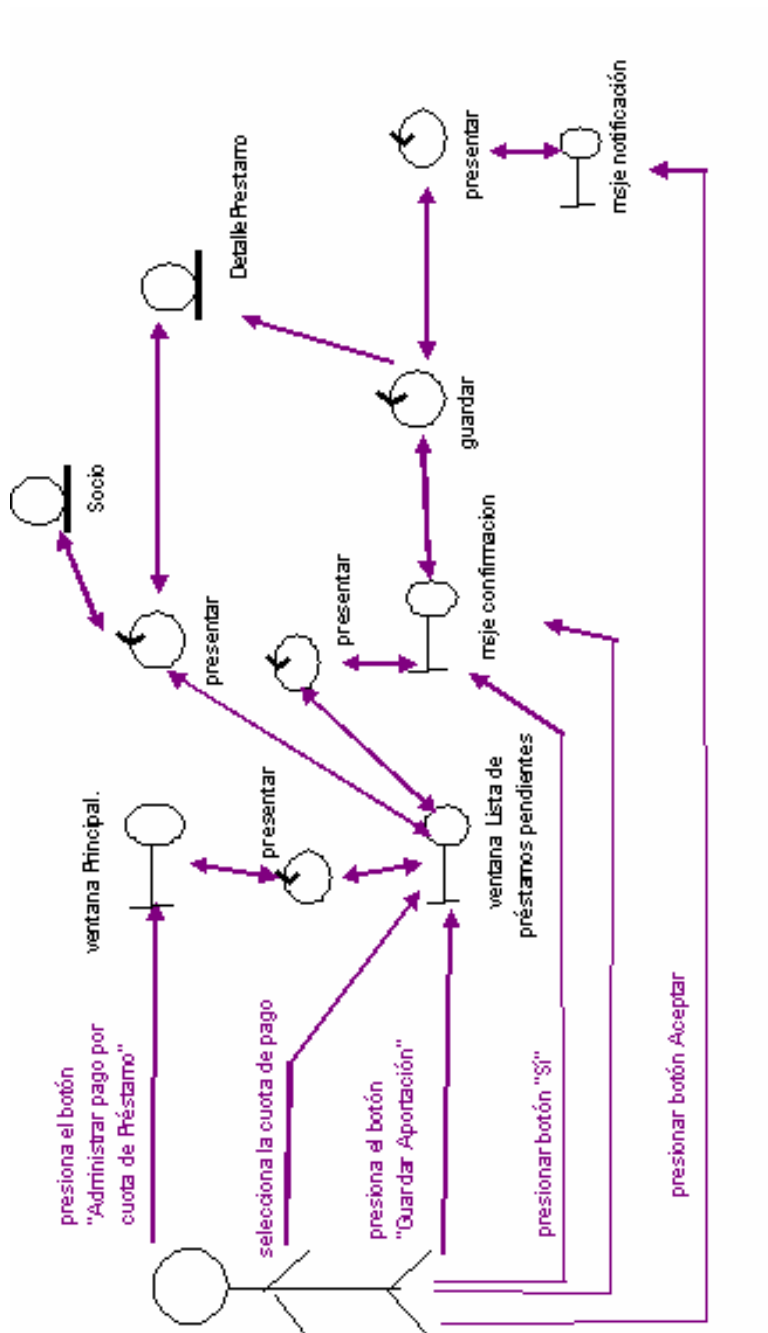
Curso alternativo A: Registrar Pago de Préstamo



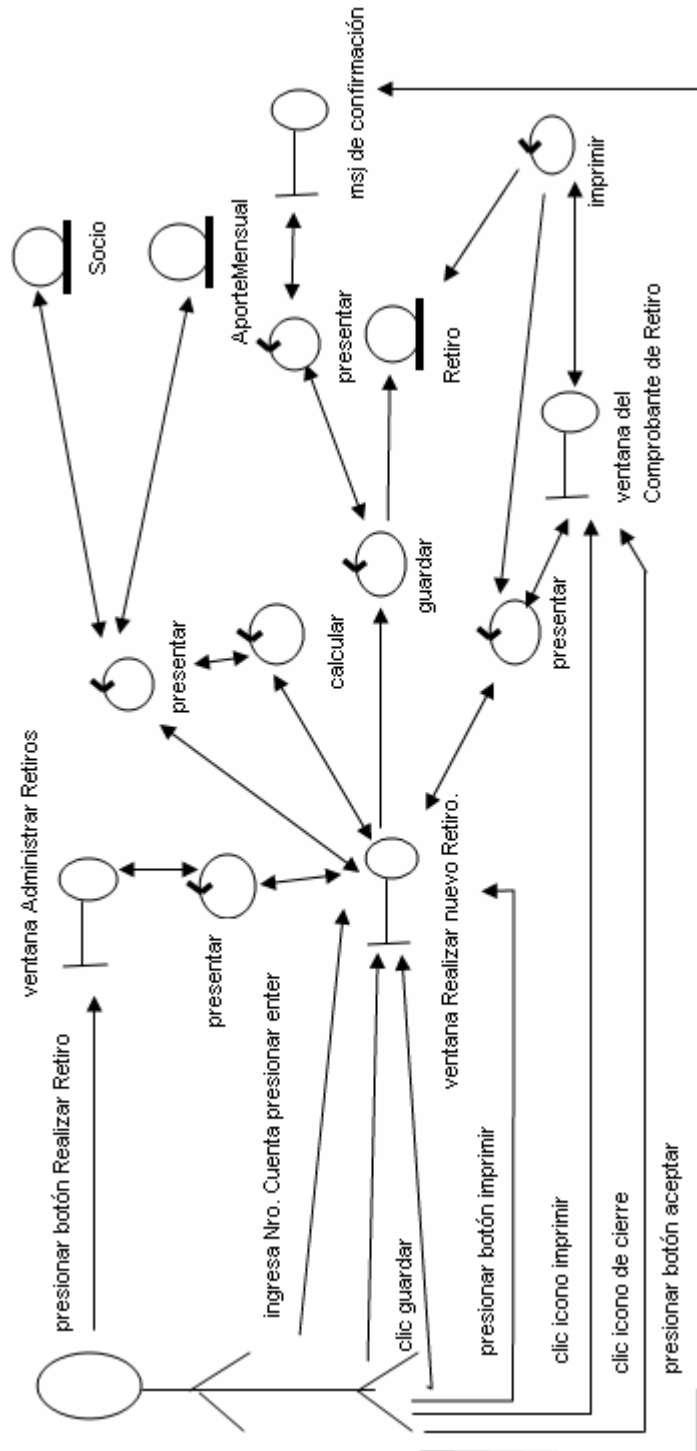
Curso alternativo B: Actualizar Préstamos



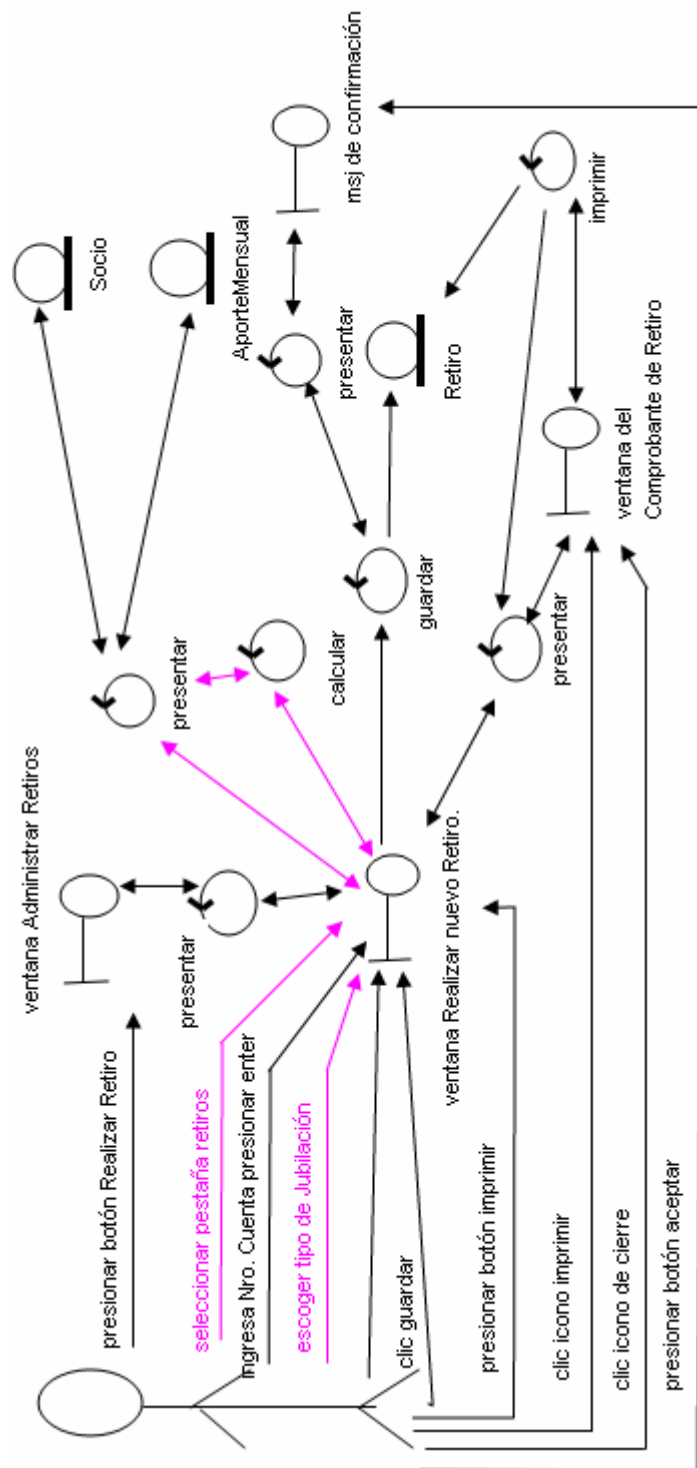
Curso alternativo C: Pago de cuotas por Préstamos



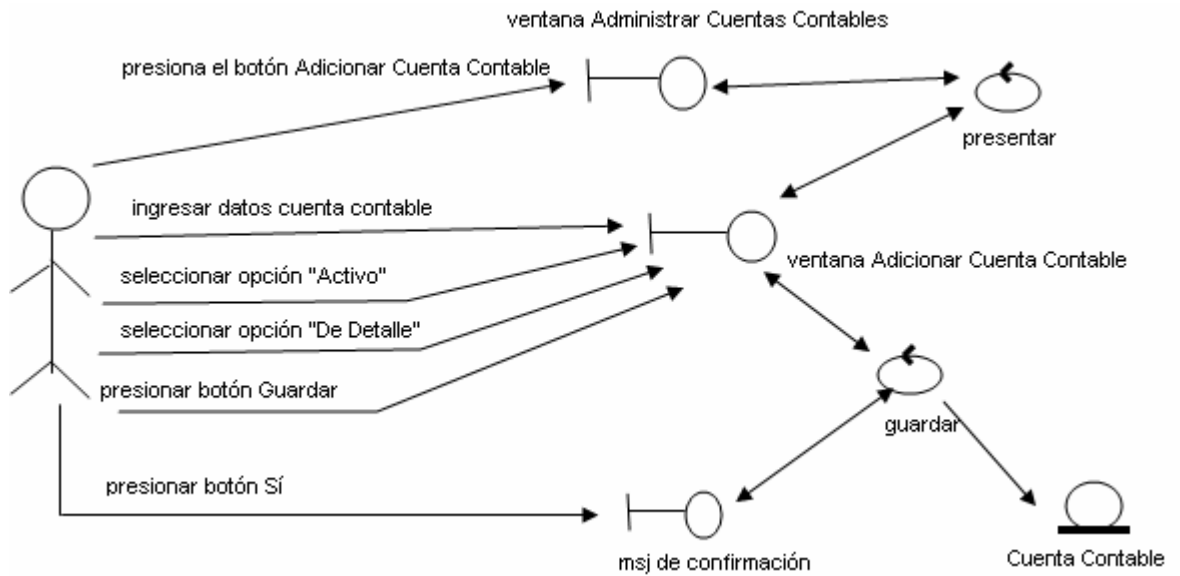
CASO DE USO: ADMINISTRAR RETIRO
Curso Normal de Eventos: Retiro de Cesantía



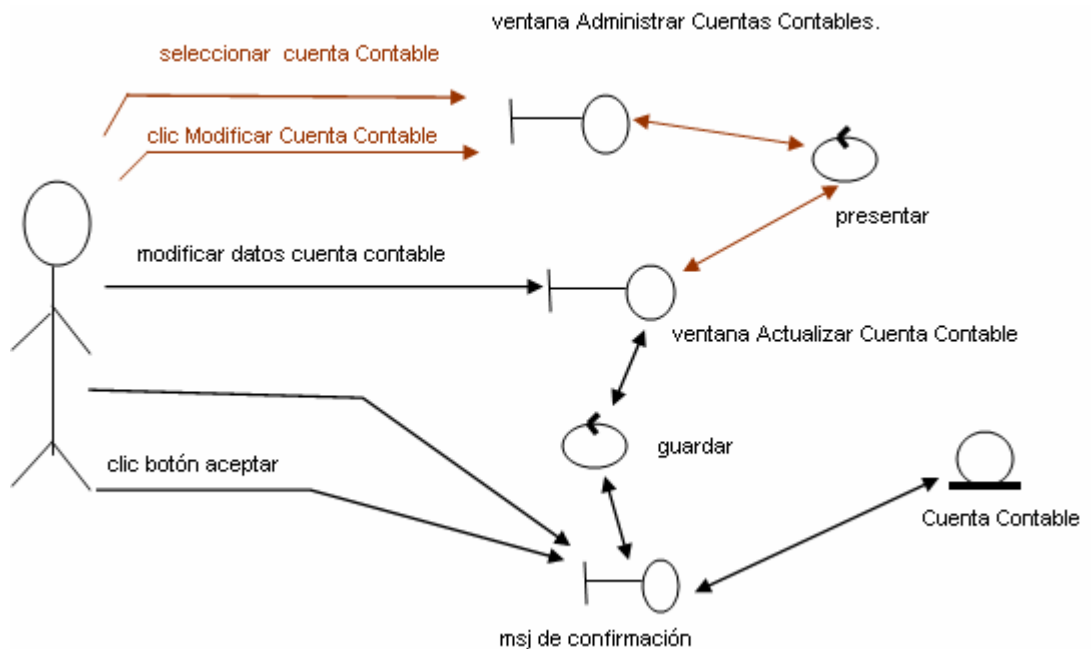
Curso Alterno A: Retiro de Jubilación



CASO DE USO: ADMINISTRAR CUENTAS CONTABLES
Curso Normal de eventos: Adicionar Cuenta Contable al Plan de Cuentas



Curso Alternativo A: Modificar Cuenta Contable

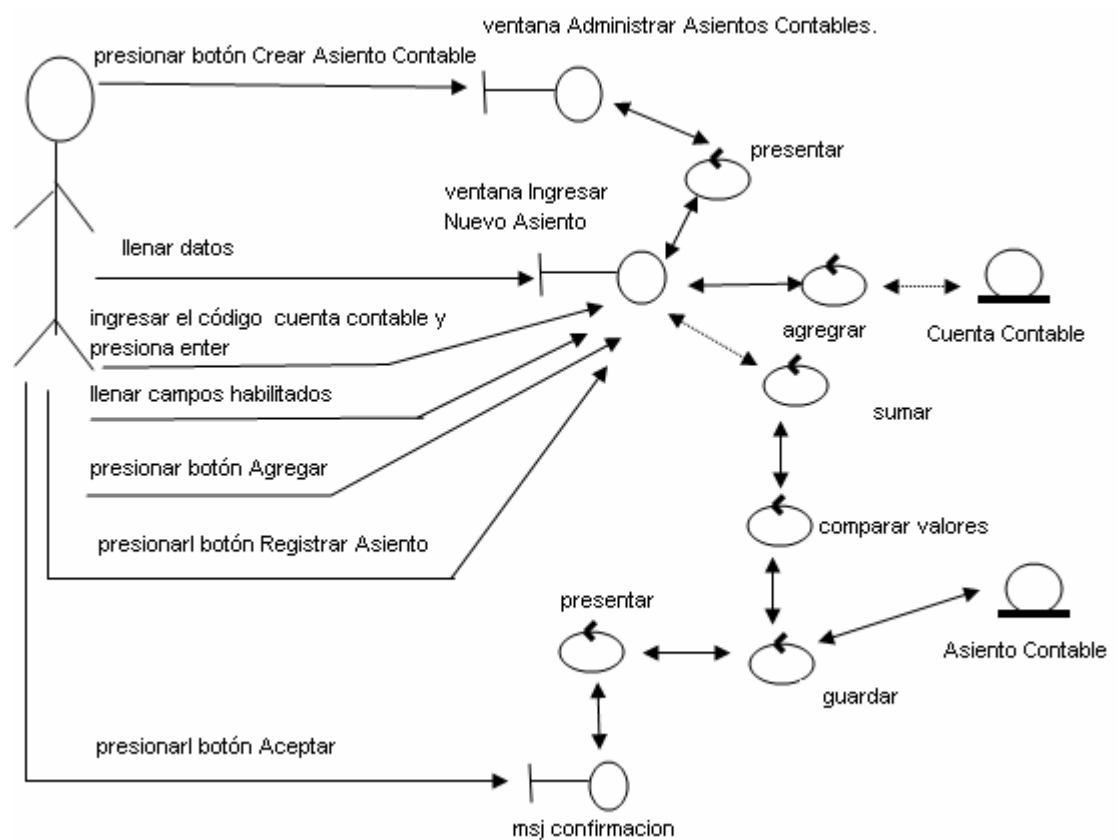


Curso Alterno B: Eliminar Cuenta Contable

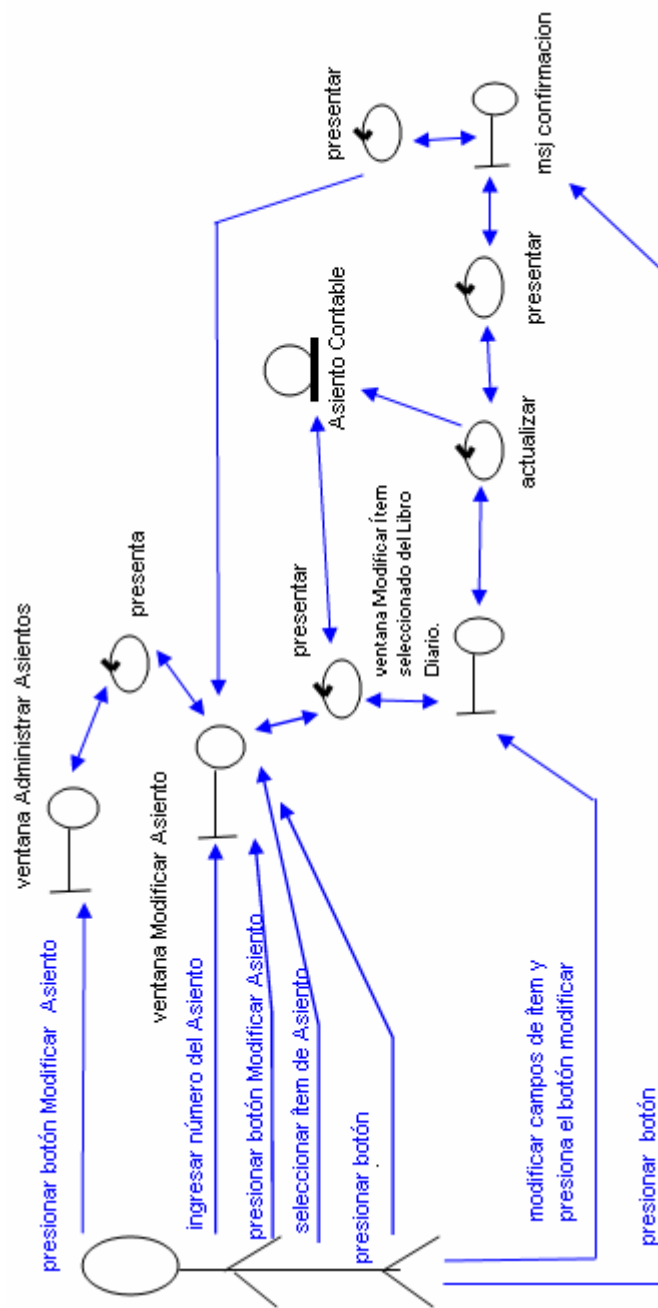


CASO DE USO: INGRESAR TRANSACCIONES DIARIAS

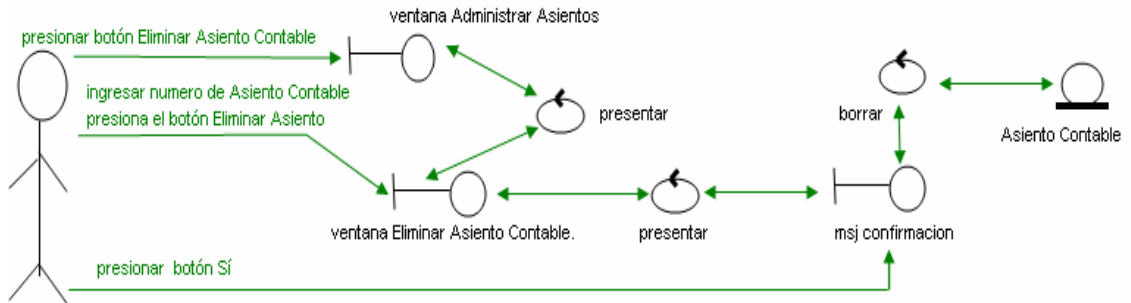
Curso Normal de eventos: Asientos Contables



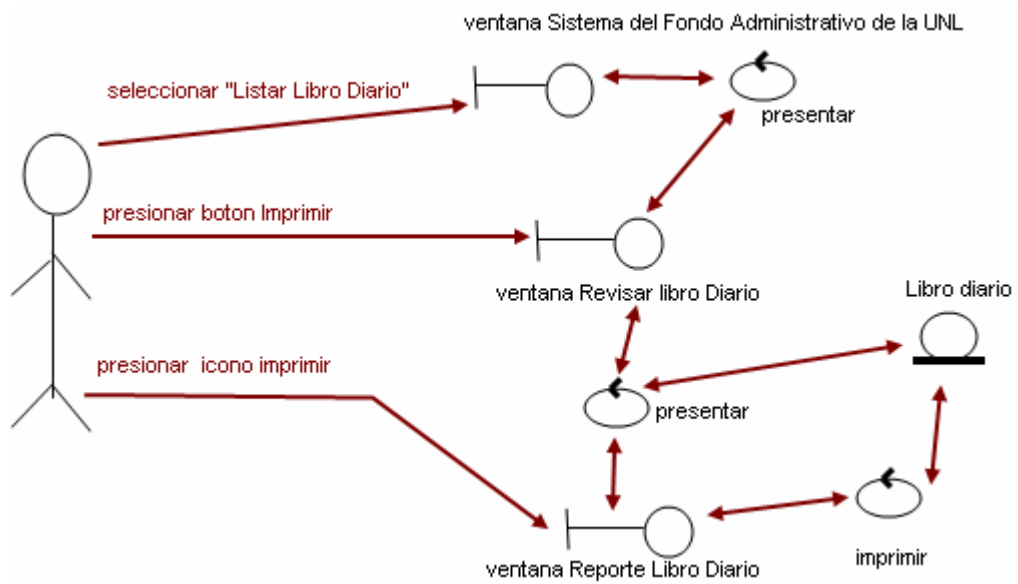
Curso Alterno A: Modificar Asiento Contable



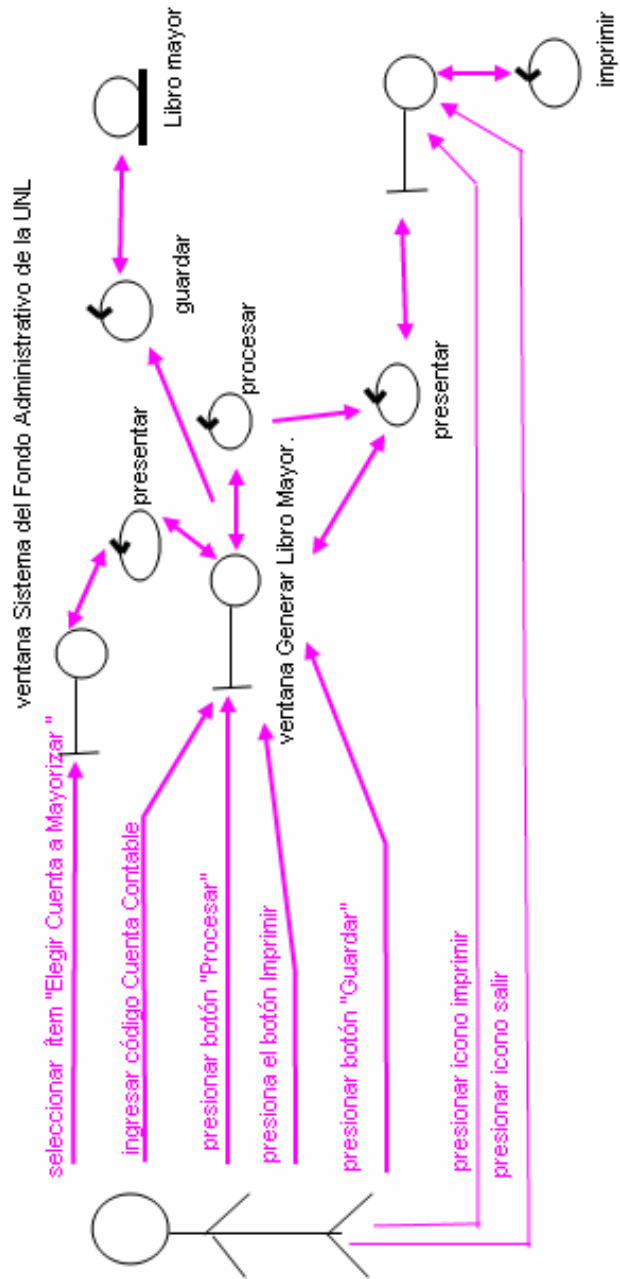
Curso Alterno B: Eliminar Asiento Contable



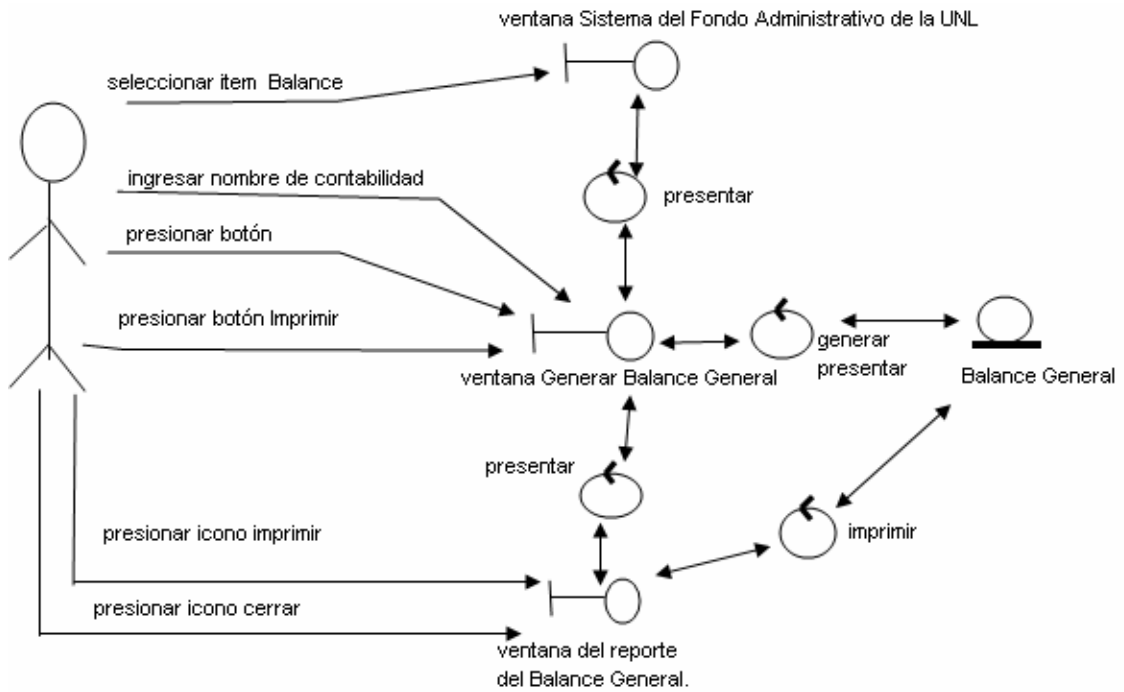
Curso Alterno C: Generar Libro Diario



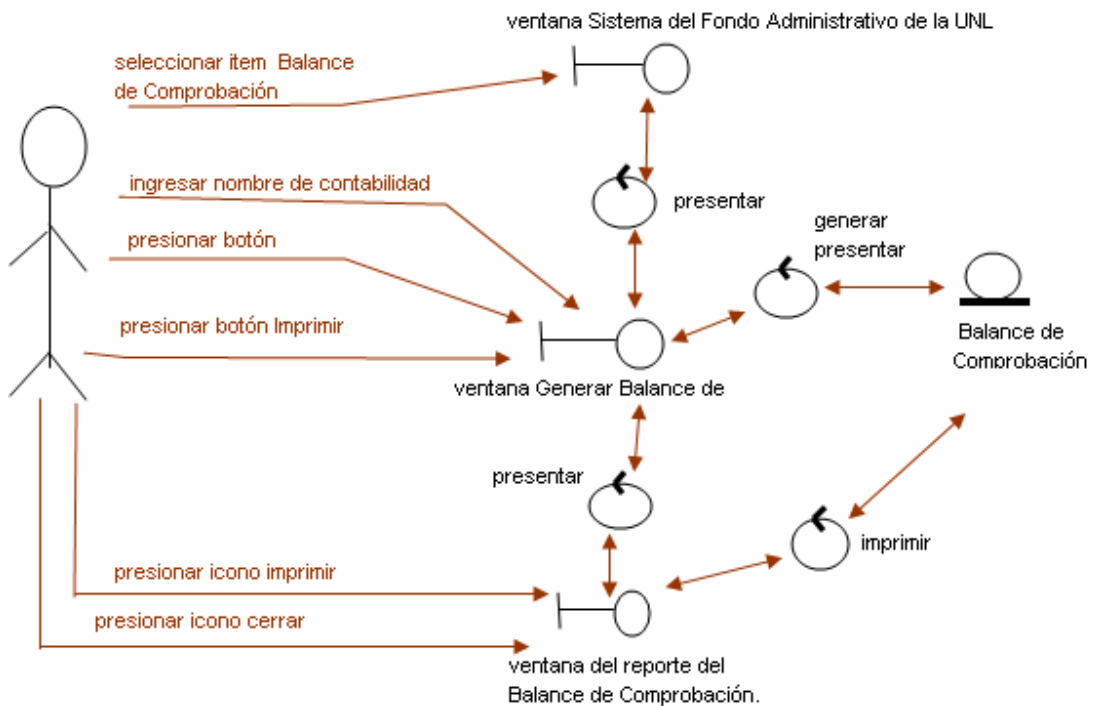
Curso Alterno D: Libro Mayor por Cuenta



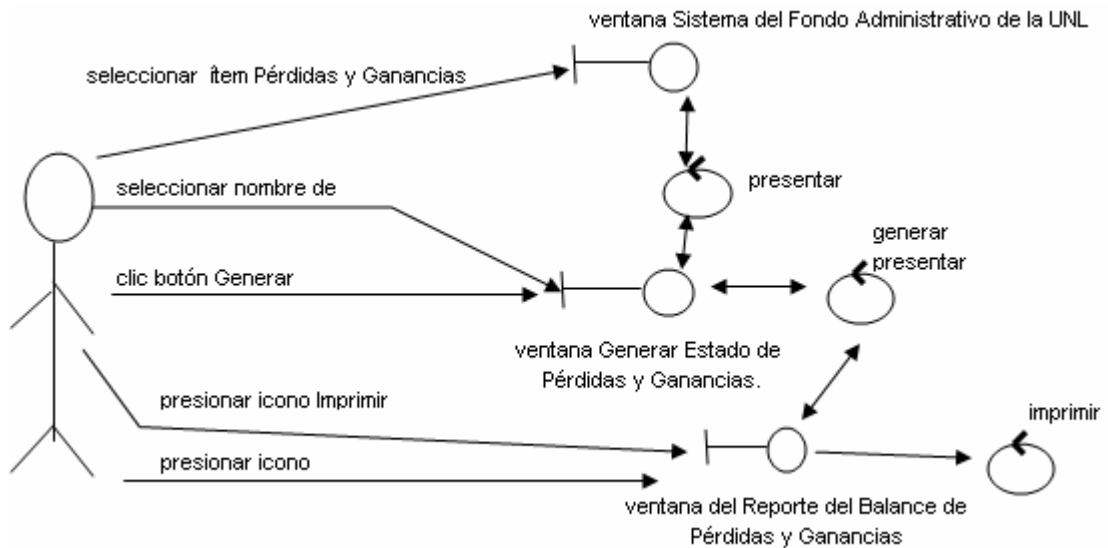
CASO DE USO: GENERAR BALANCES CONTABLES
Curso Normal de Eventos: Generar Balance General



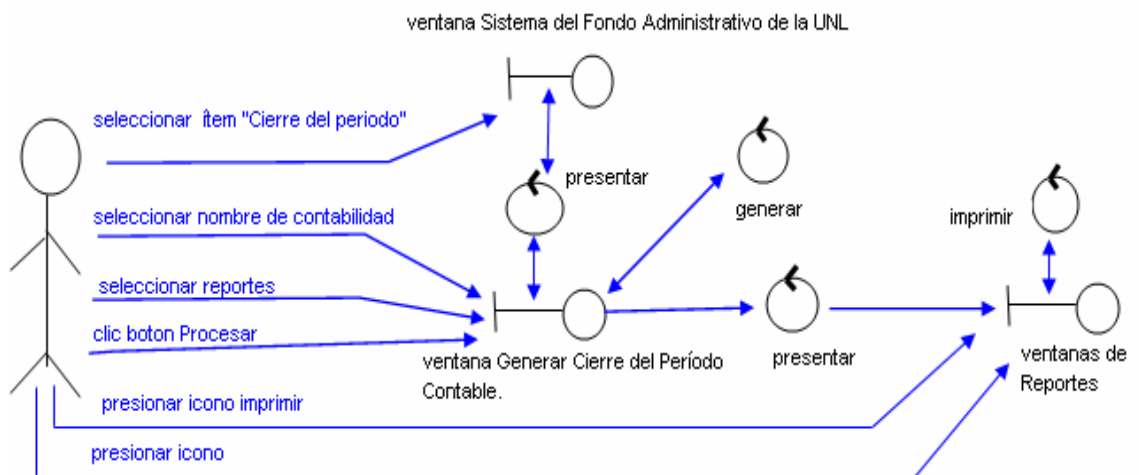
Curso Alterno A: Generar Balances de Comprobación



CASO DE USO: GENERAR ESTADO DE SITUACIÓN FINAL
Curso Normal de Eventos: Estado de Pérdidas y Ganancias

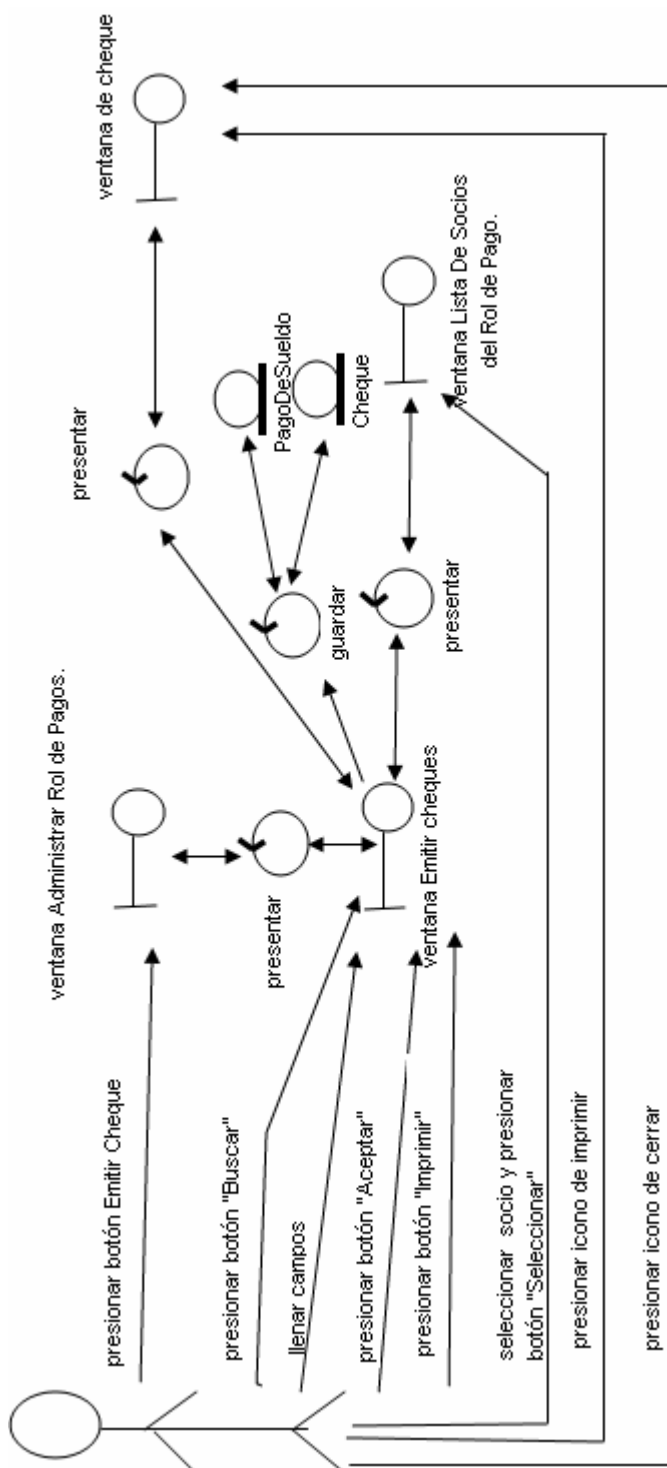


Curso Alterno A: Cierre de Periodo

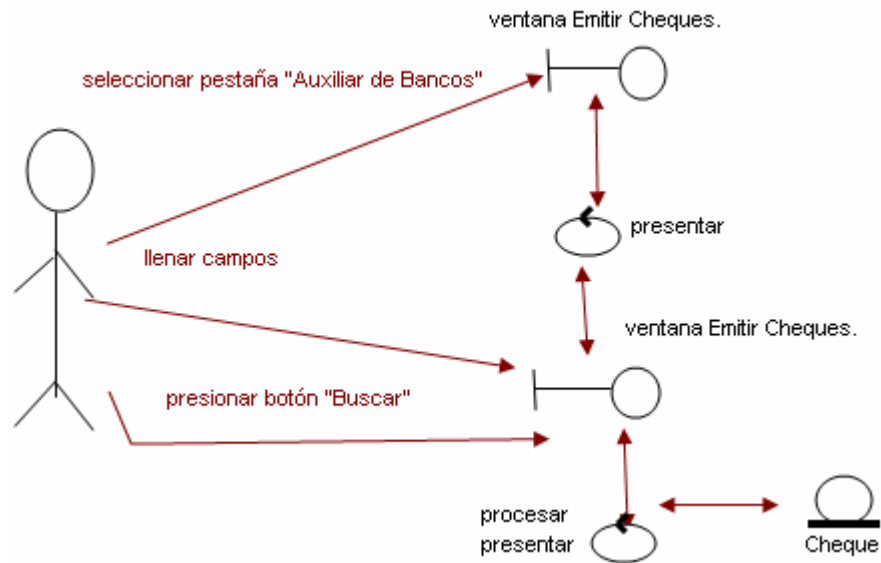


CASO DE USO: ADMINISTRAR ROL DE PAGO

Curso Normal de Eventos: Emitir Cheque

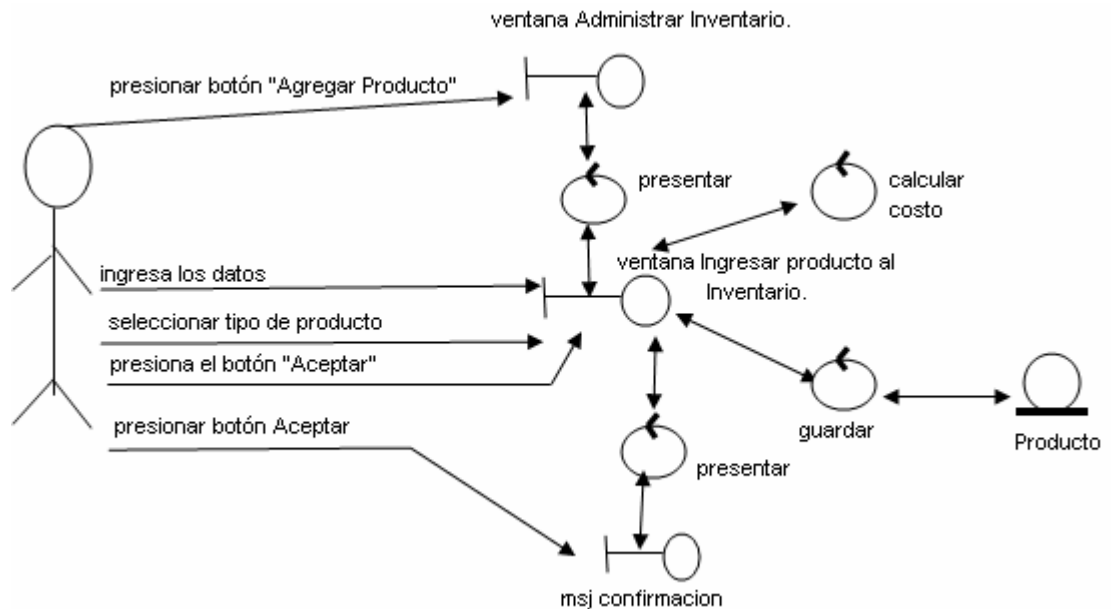


Curso Alterno A: Auxiliar de Bancos

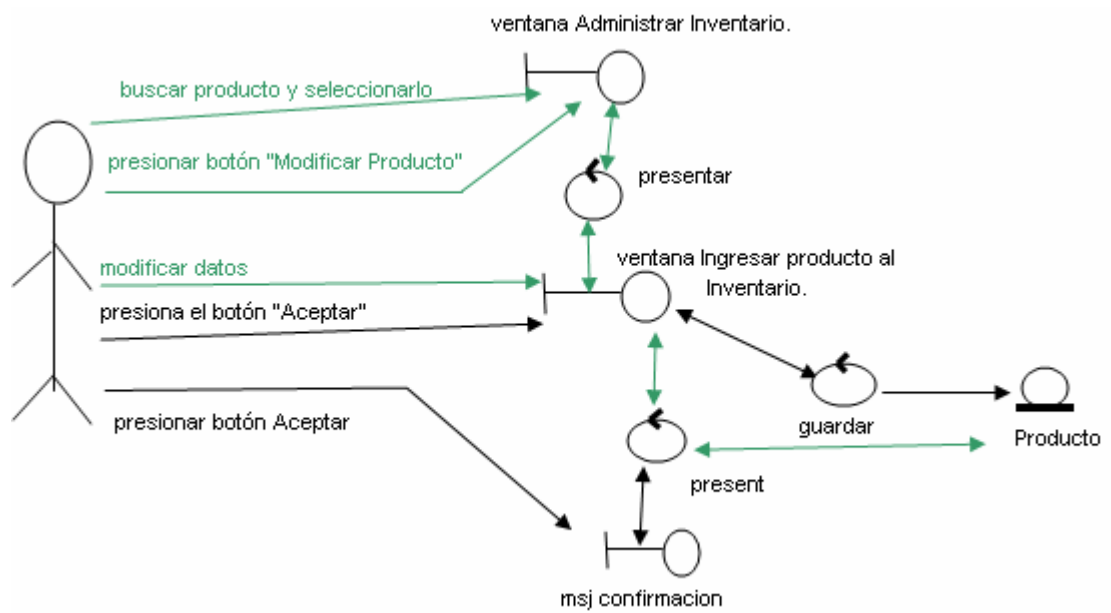


CASO DE USO: ADMINISTRAR INVENTARIO

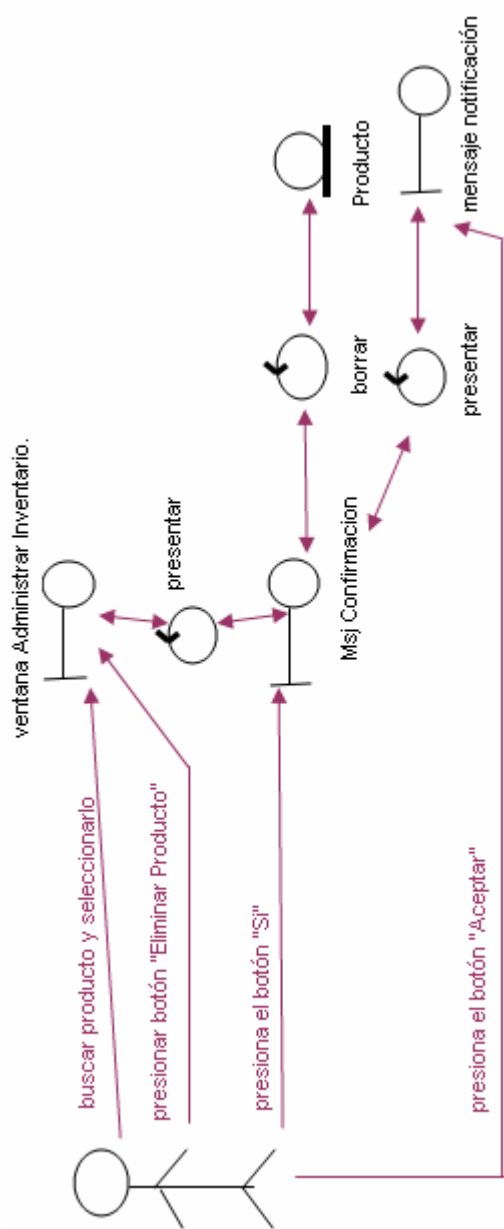
Curso Normal de Eventos: Crear Producto



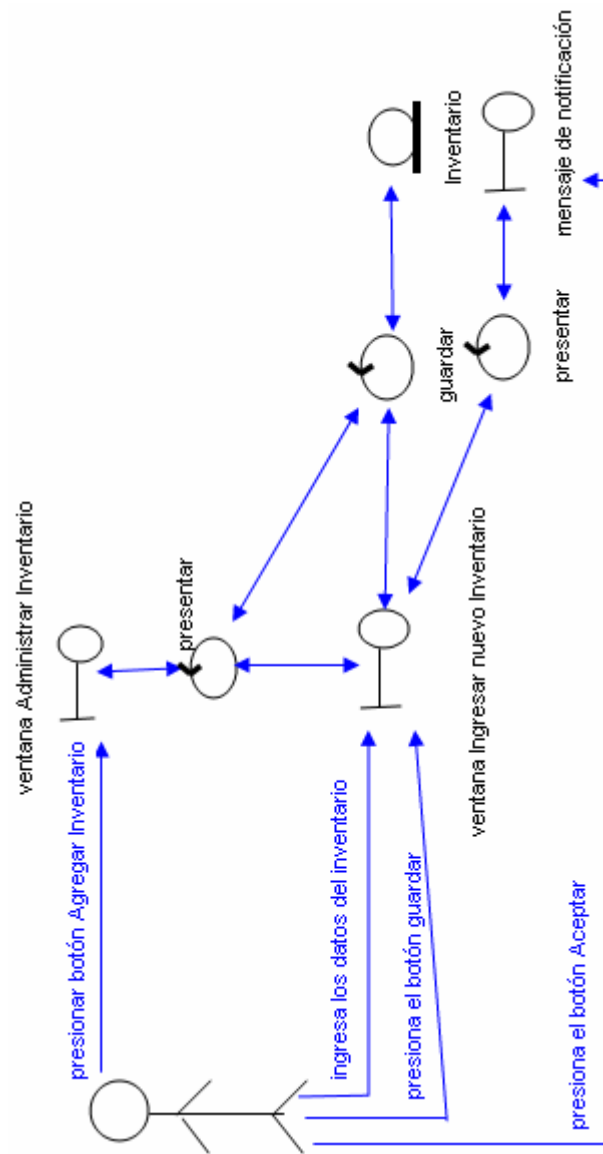
Curso Alterno A: Modificar Producto



Curso Alterno B: Eliminar Producto

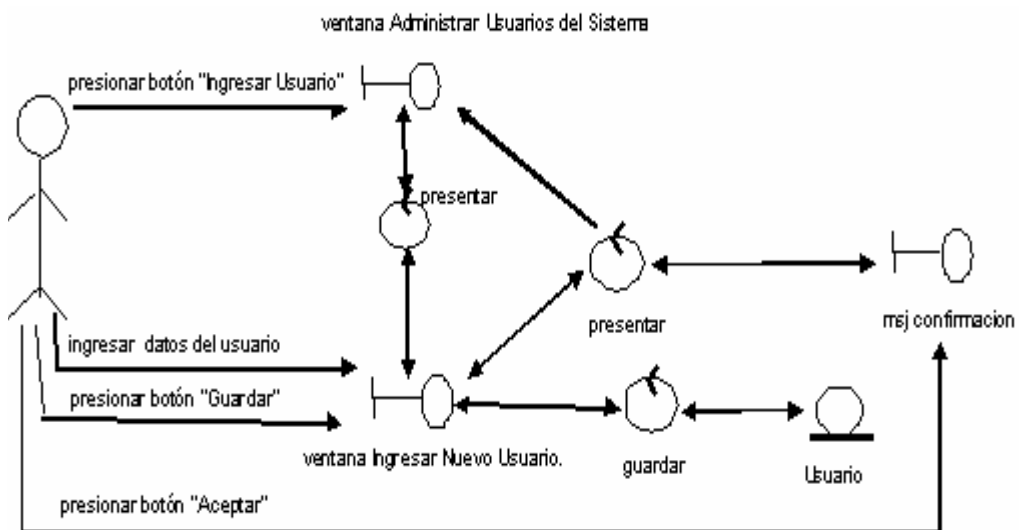


Curso Alterno C: Crear el Inventario

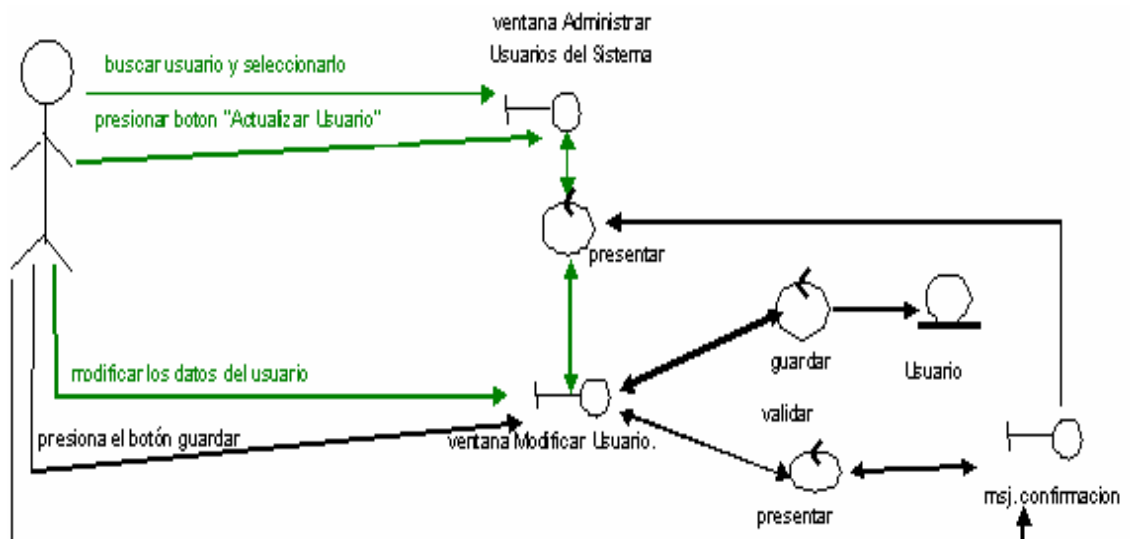


CASO DE USO: ADMINISTRAR USUARIOS DEL SISTEMA

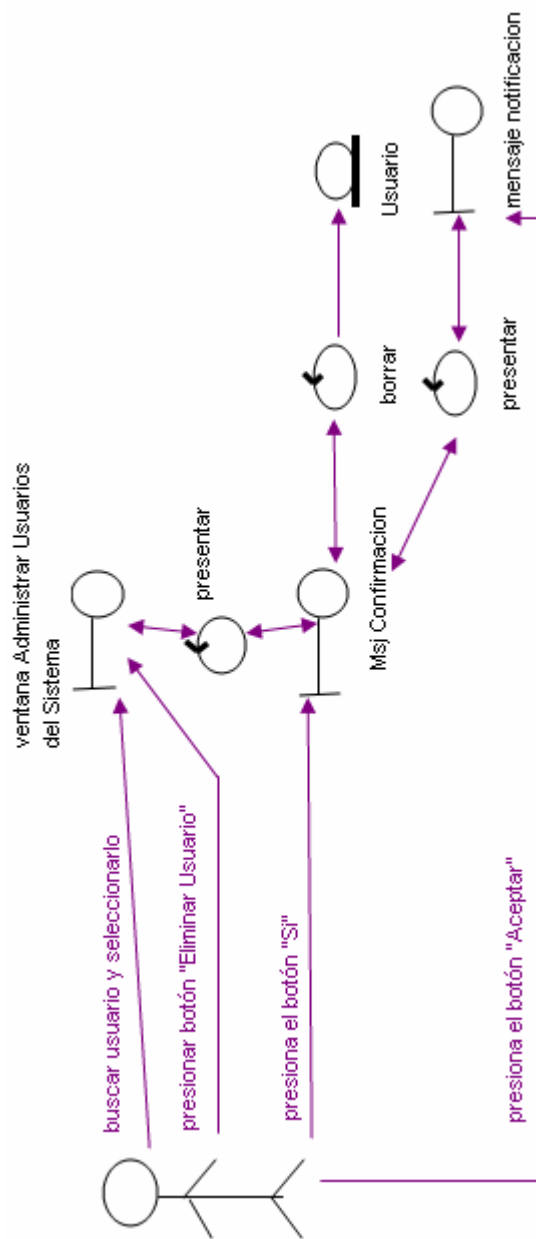
Curso Normal de Eventos: Agregar Usuario



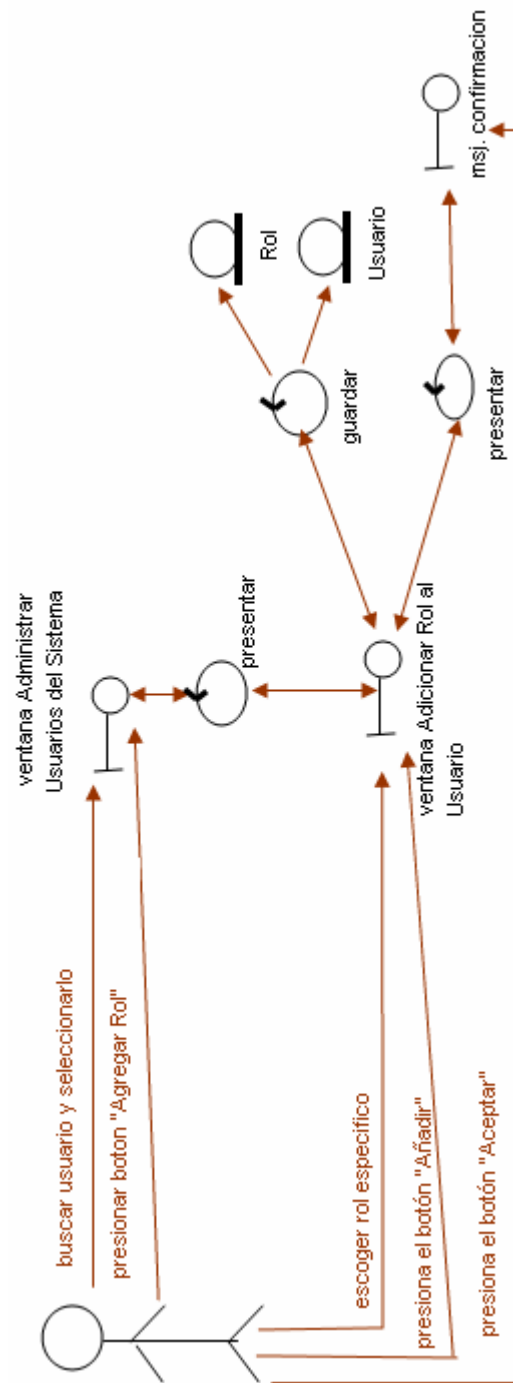
Curso Alternativo A: Actualizar Usuario



Curso Alterno B: Eliminar Usuario

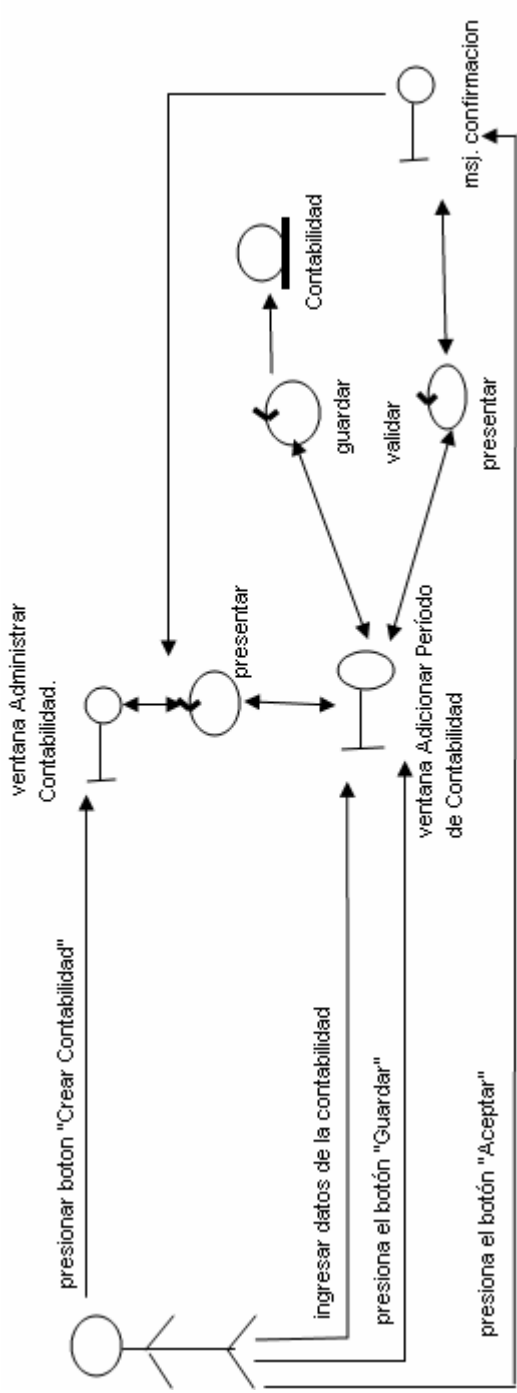


Curso Alterno C: Asignar Rol

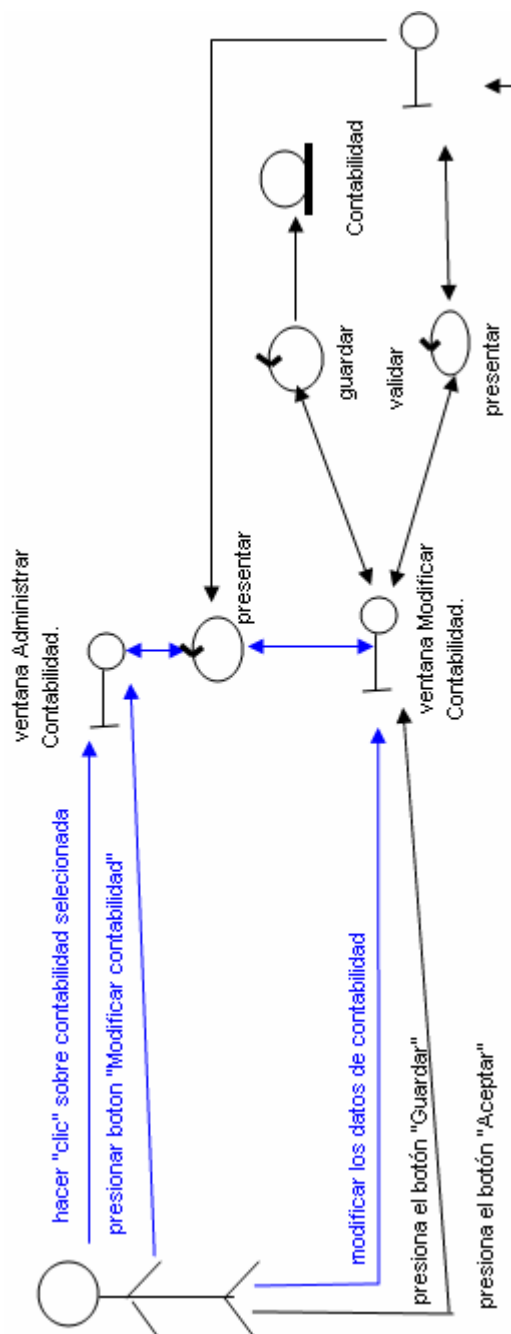


CASO DE USO: ADMINISTRAR CONTABILIDAD

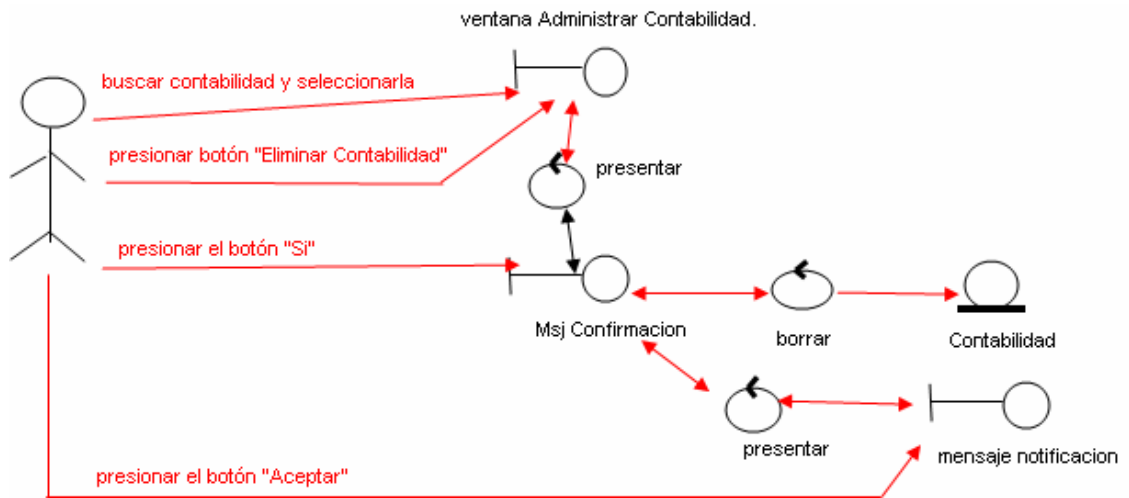
Curso Normal de Eventos: Agregar Contabilidad



Curso Alterno A: Modificar Contabilidad

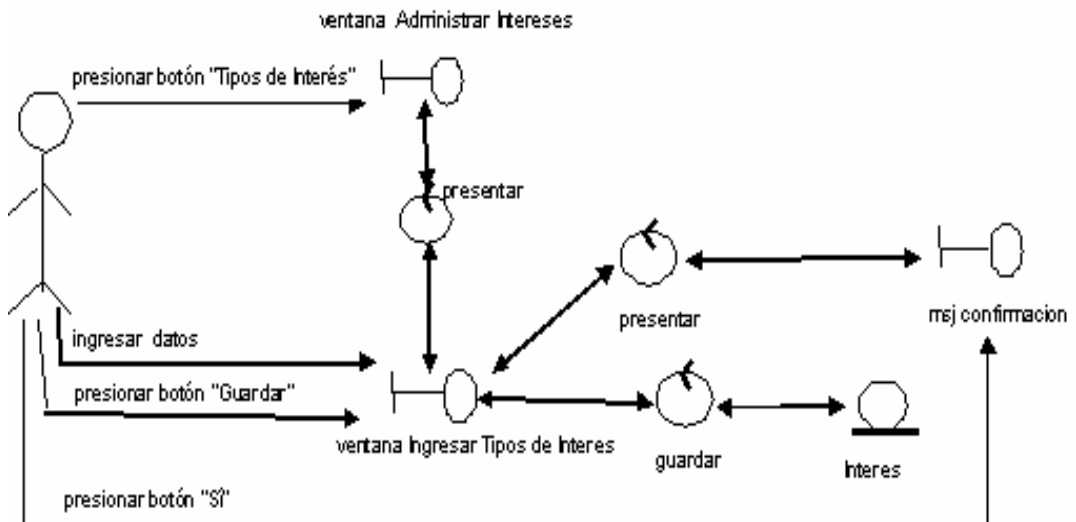


Curso Alterno B: Eliminar Contabilidad

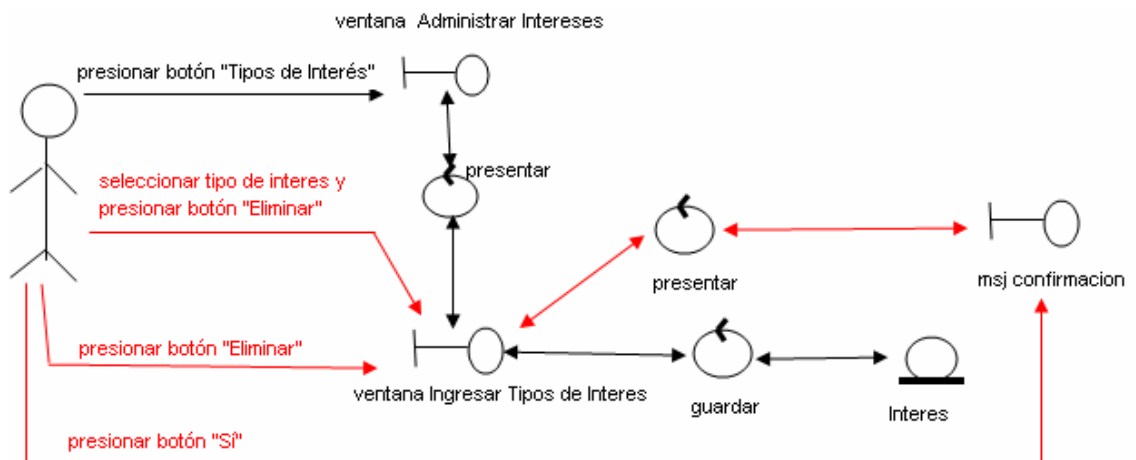


CASO DE USO: ADMINISTRAR INTERESES

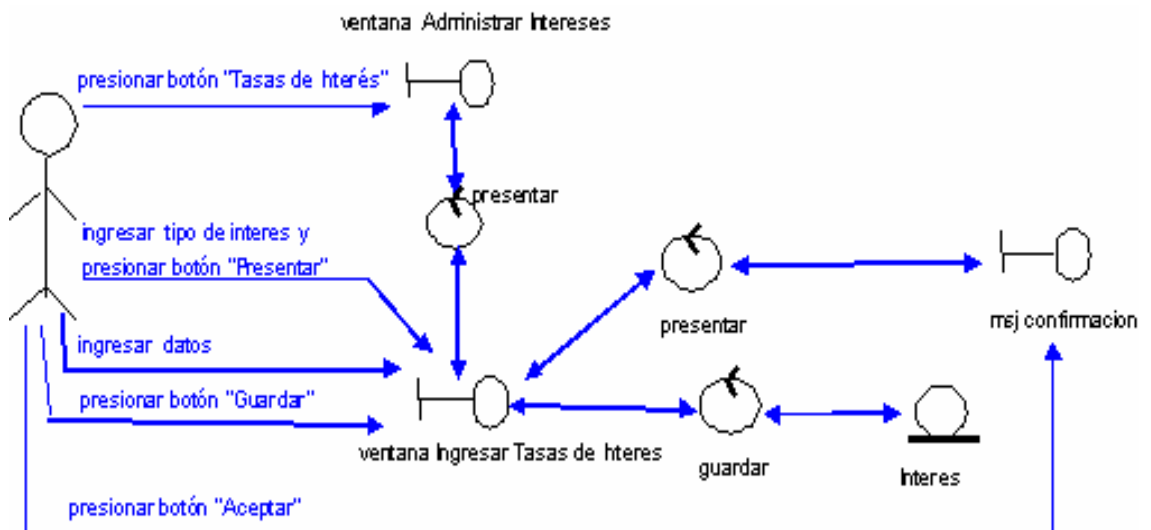
Curso Normal de Eventos: Ingresar tipos de Interés



Curso Alternativo A: Eliminar Tipos de Interés



Curso Alternativo B: Ingresar Tasas de Interés

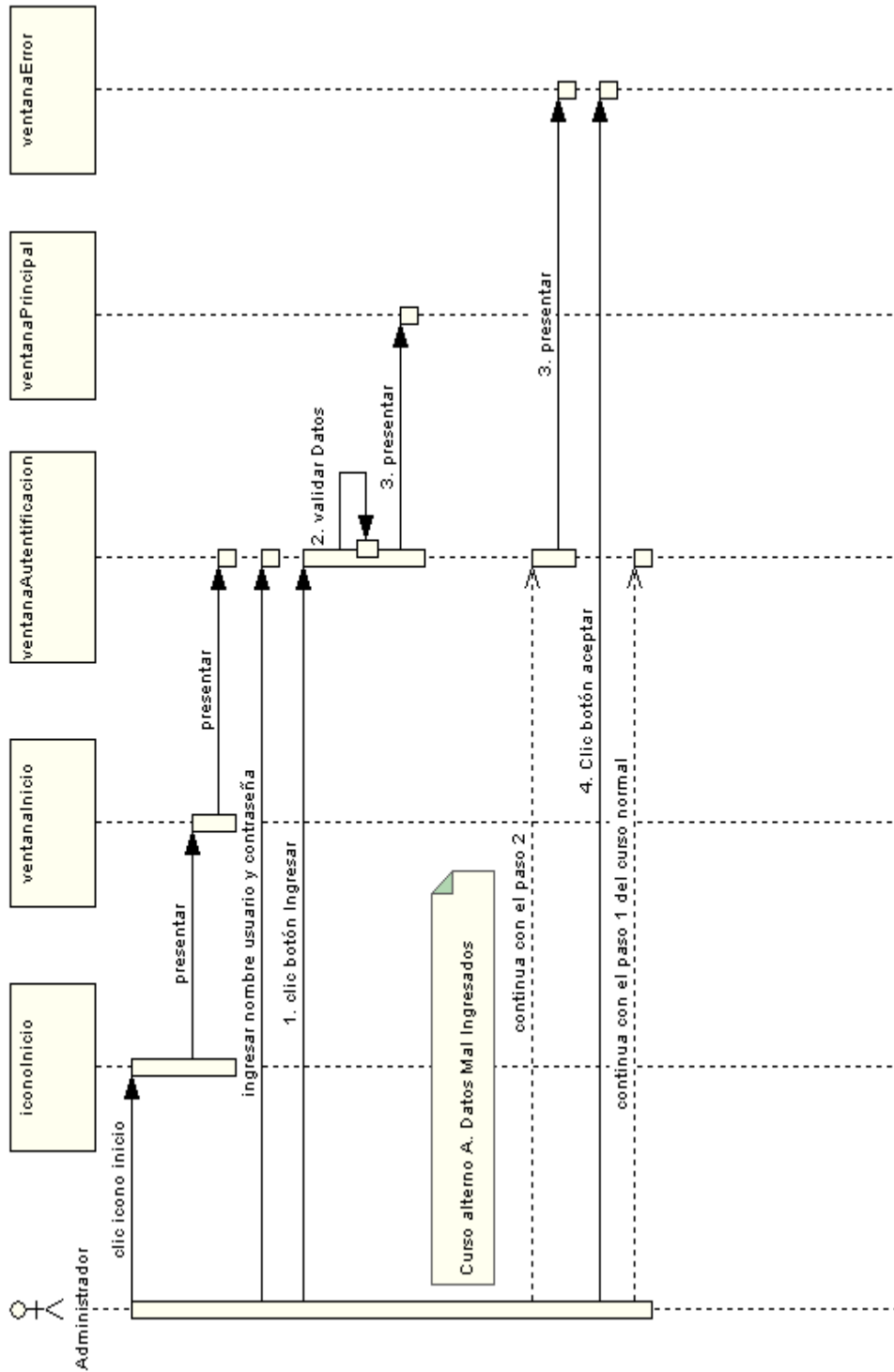


DIAGRAMAS DE SECUENCIA

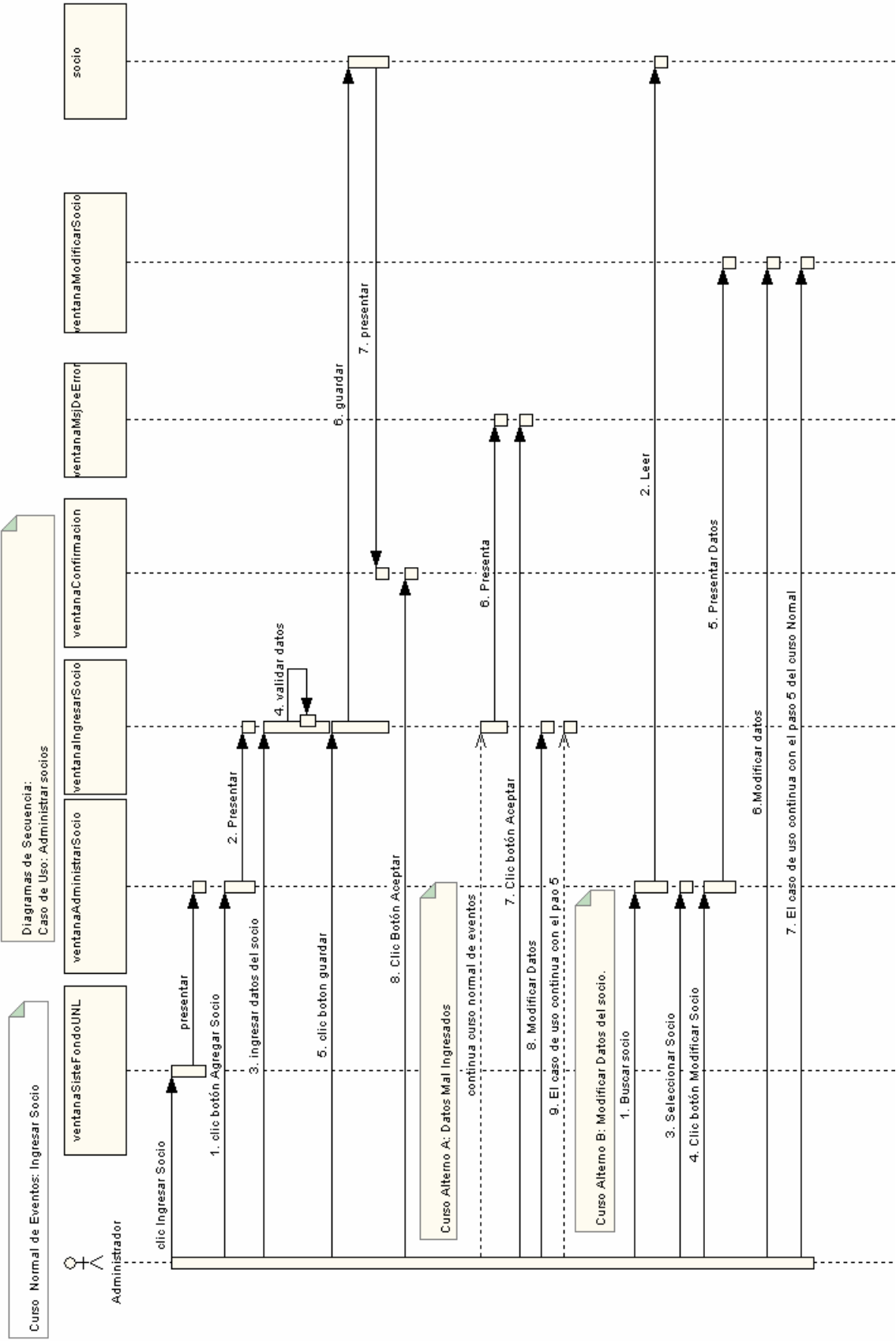
Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario.

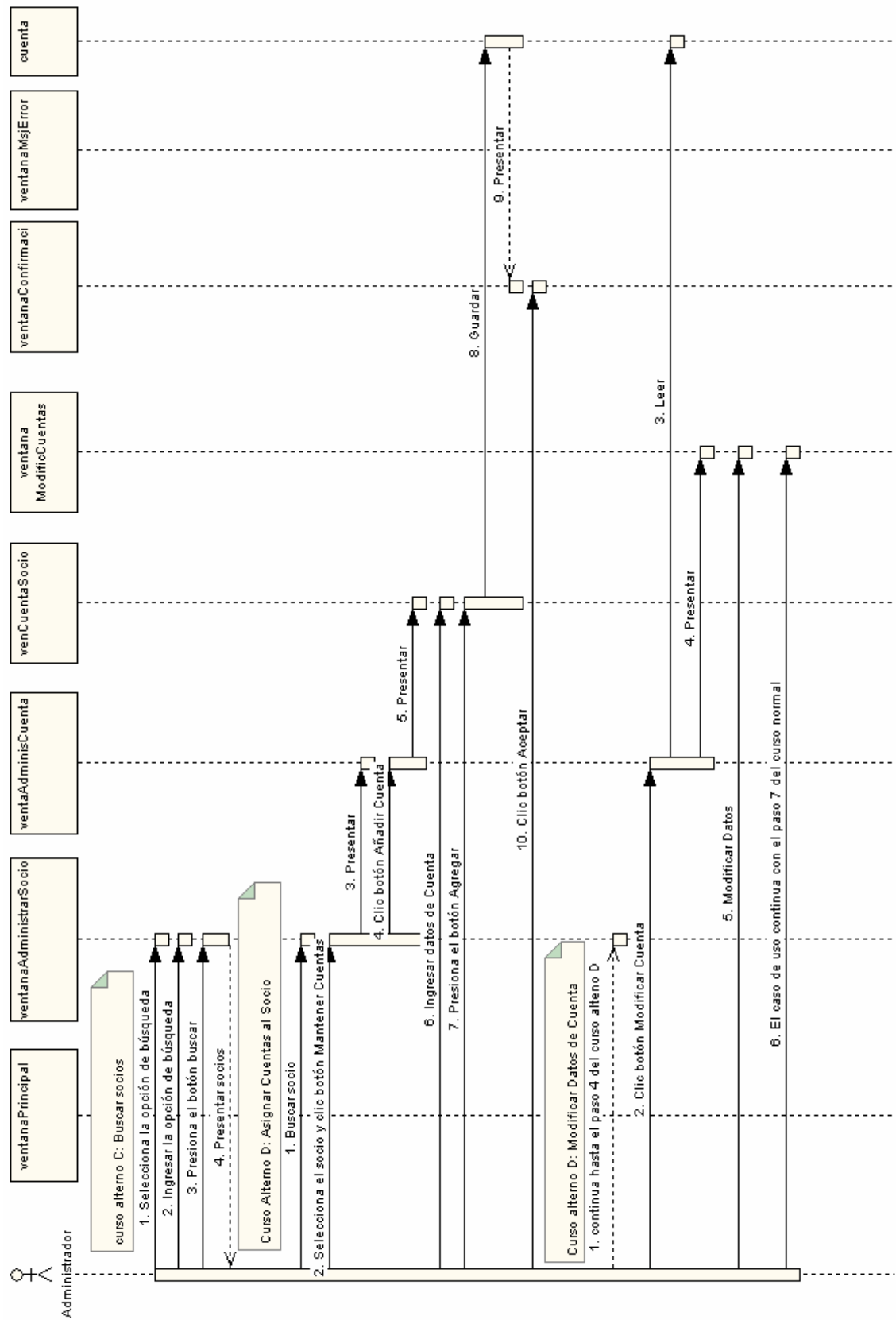
A continuación se describe de manera gráfica los diagramas de secuencia del sistema definidos para cada use case del sistema.

CASO DE USO: AUTENTIFICAR USUARIOS

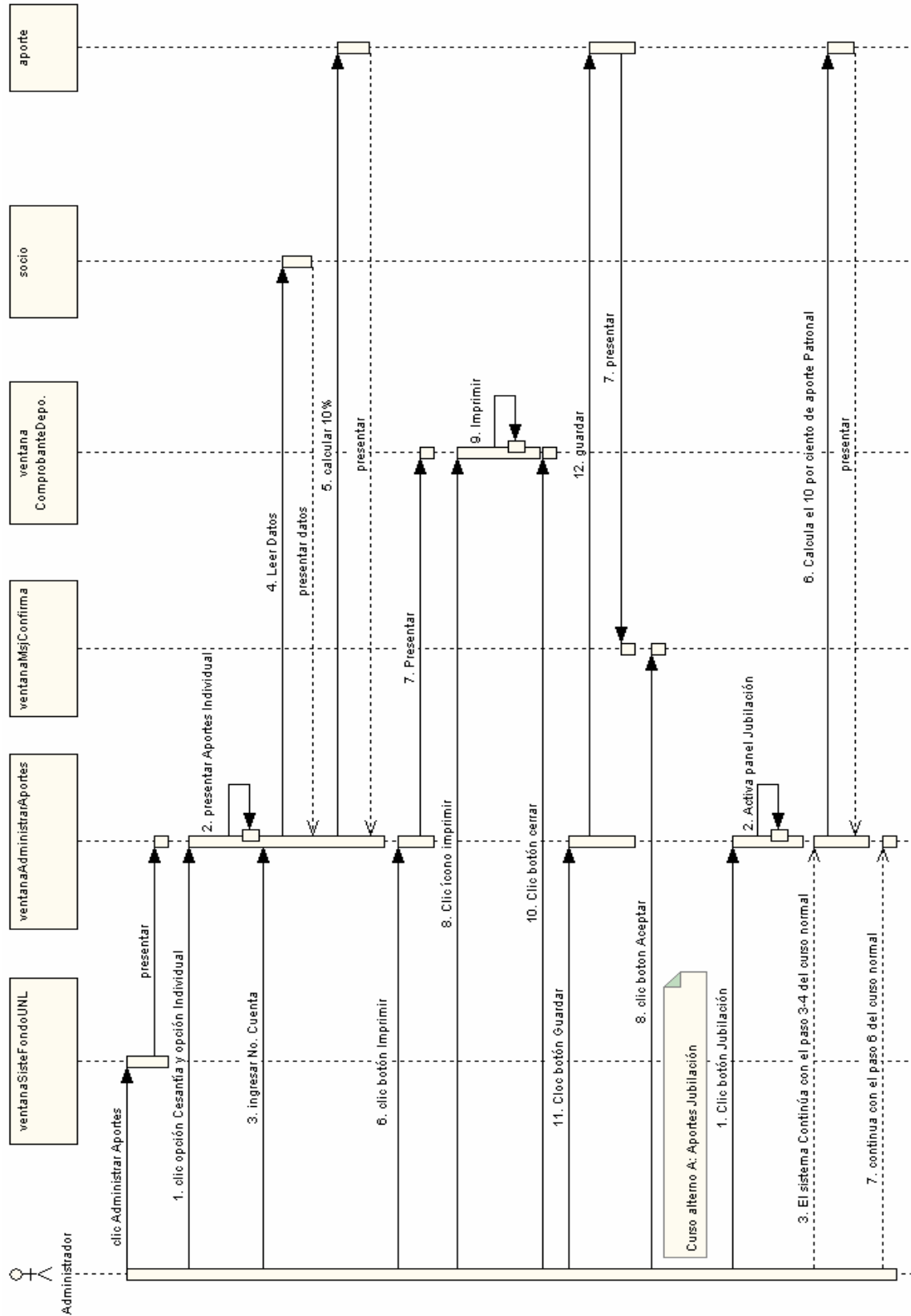


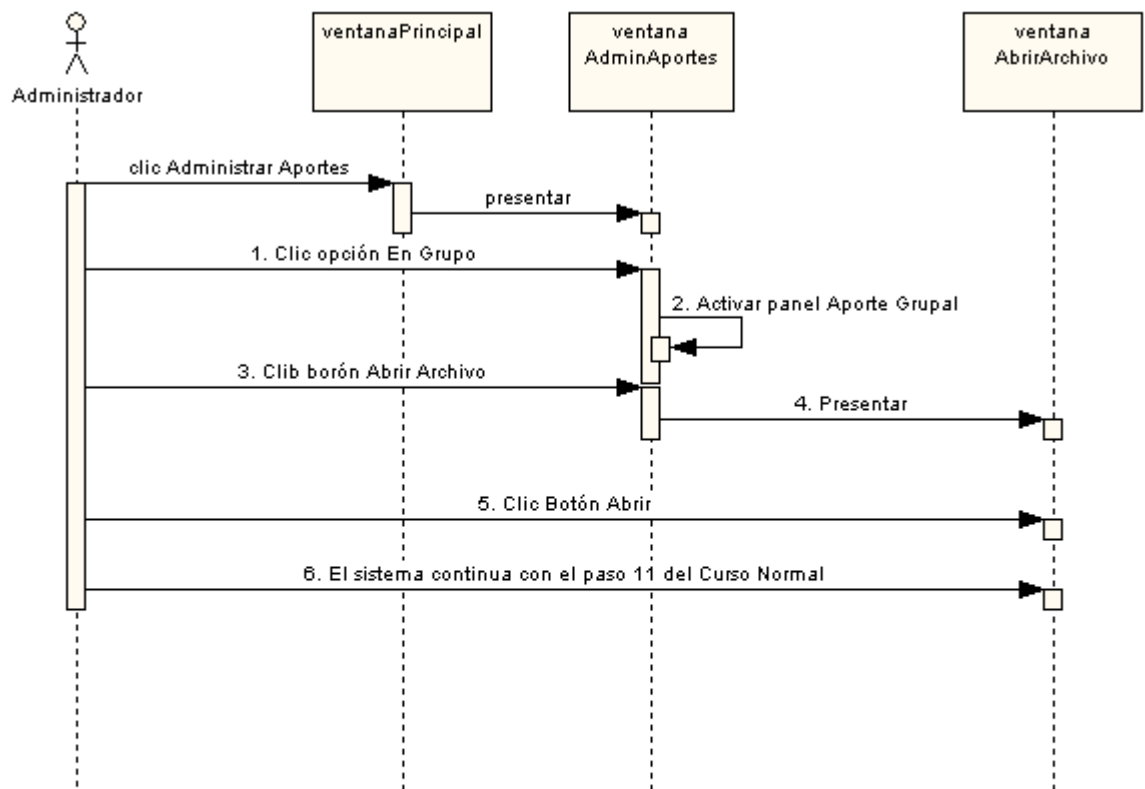
CASO DE USO: ADMINISTRAR SOCIOS





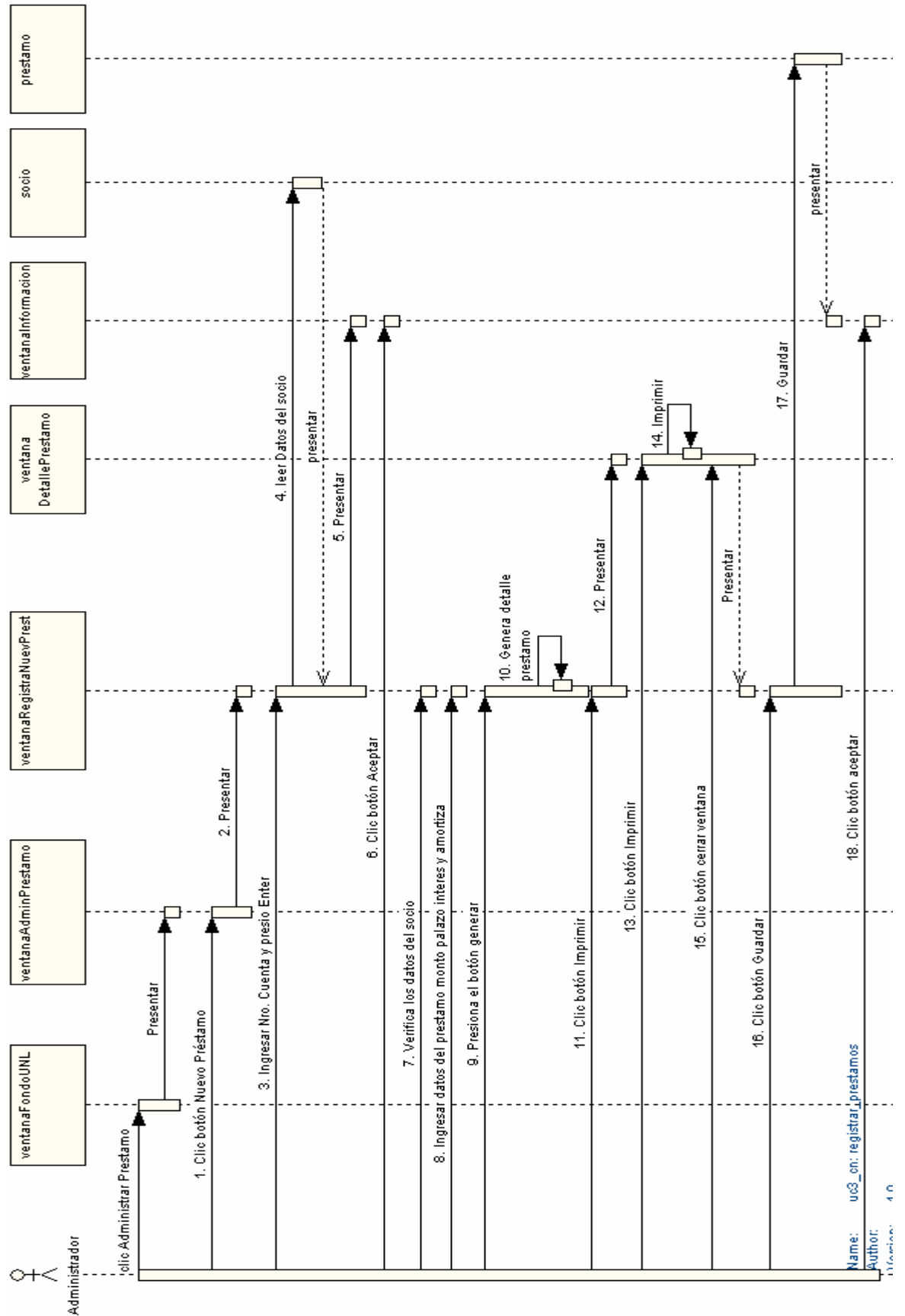
CASO DE USO: ADMINISTRAR APORTES



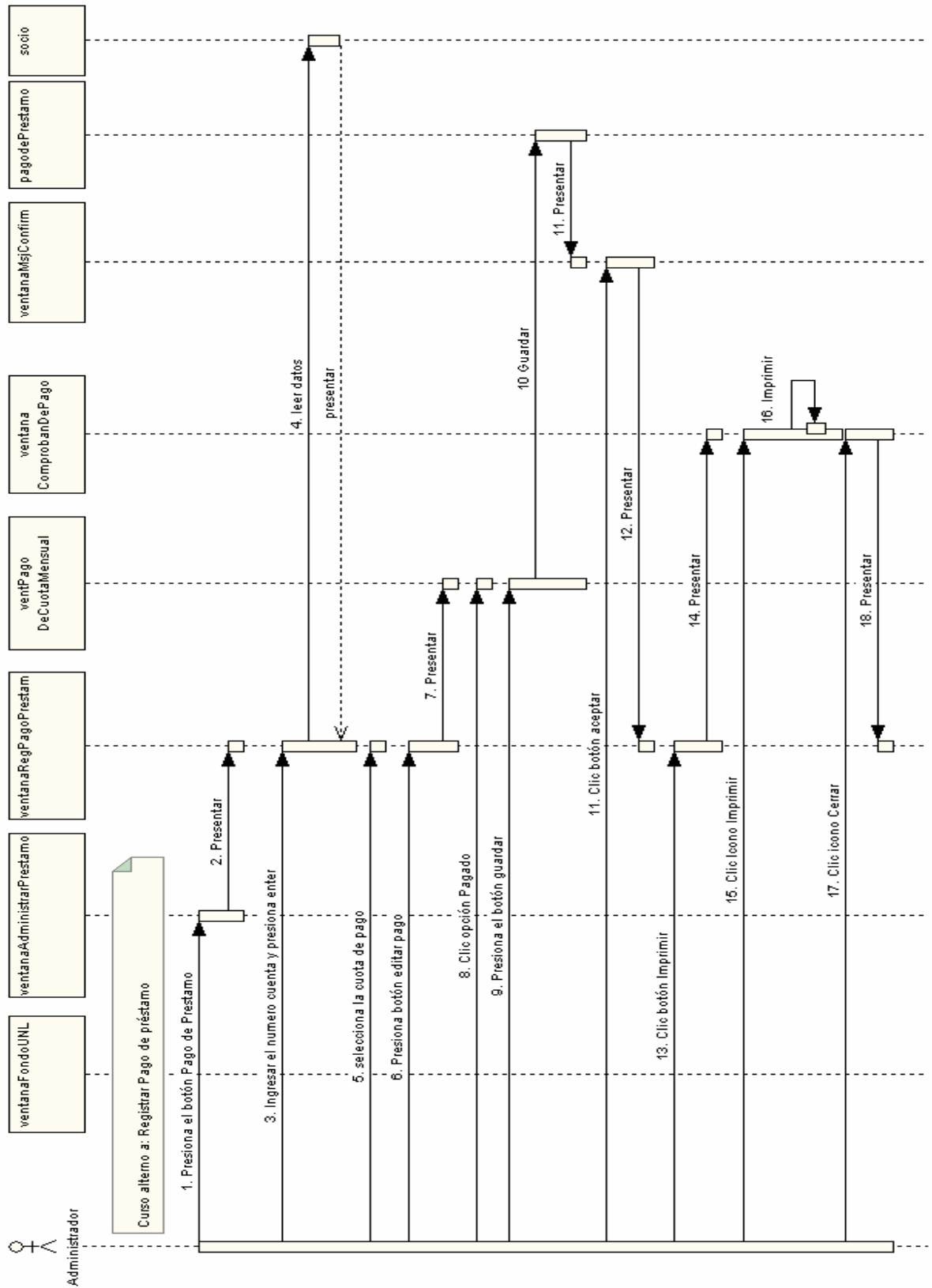


Name: ca_2: Aporte Grupal
 Author:
 Version: 1.0
 Created: 17/12/2006 11:52:50
 Updated: 17/12/2006 12:07:29

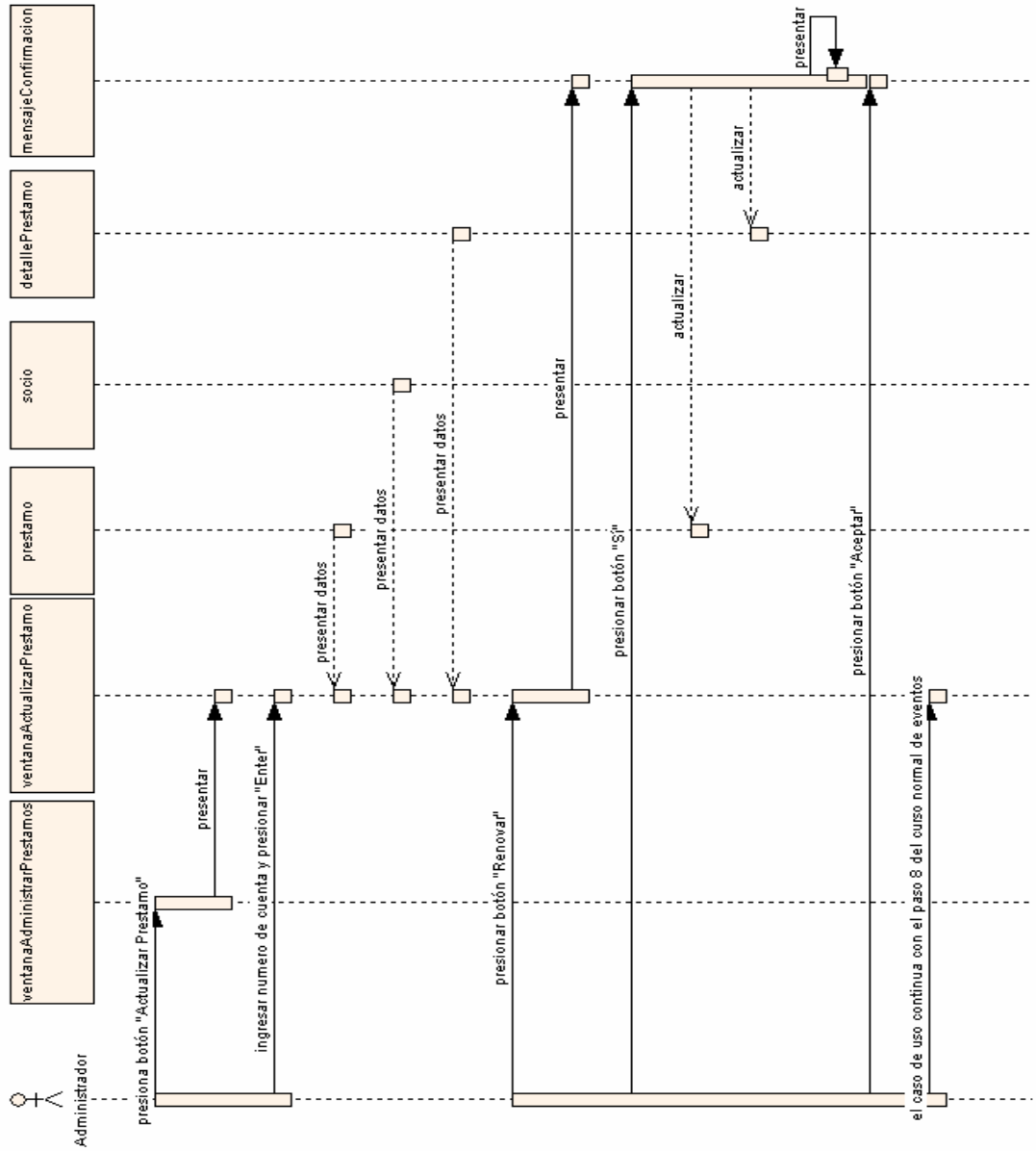
CASO DE USO: ADMINISTRAR PRÉSTAMOS



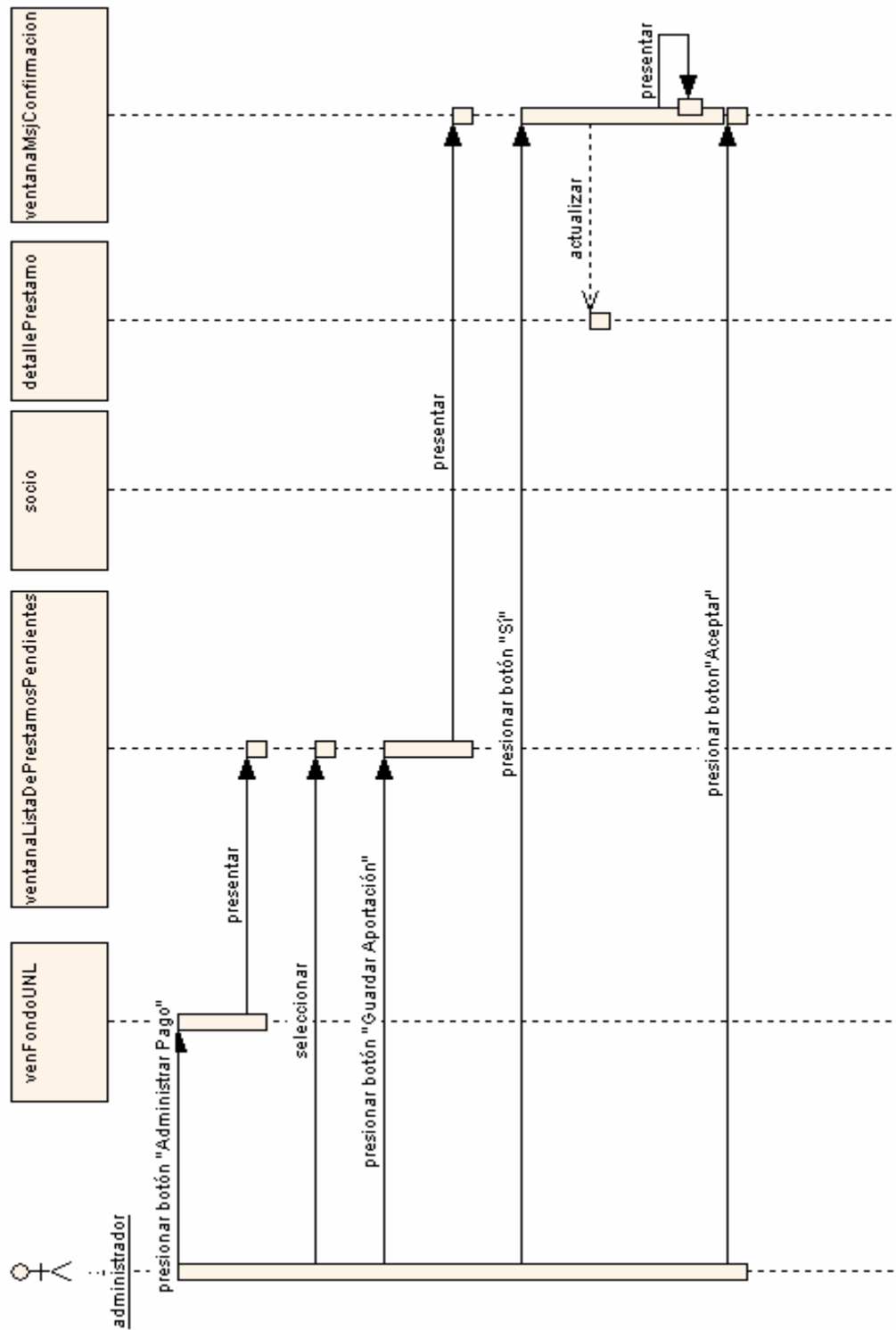
Curso alternativo A: Registrar Pago de Préstamo



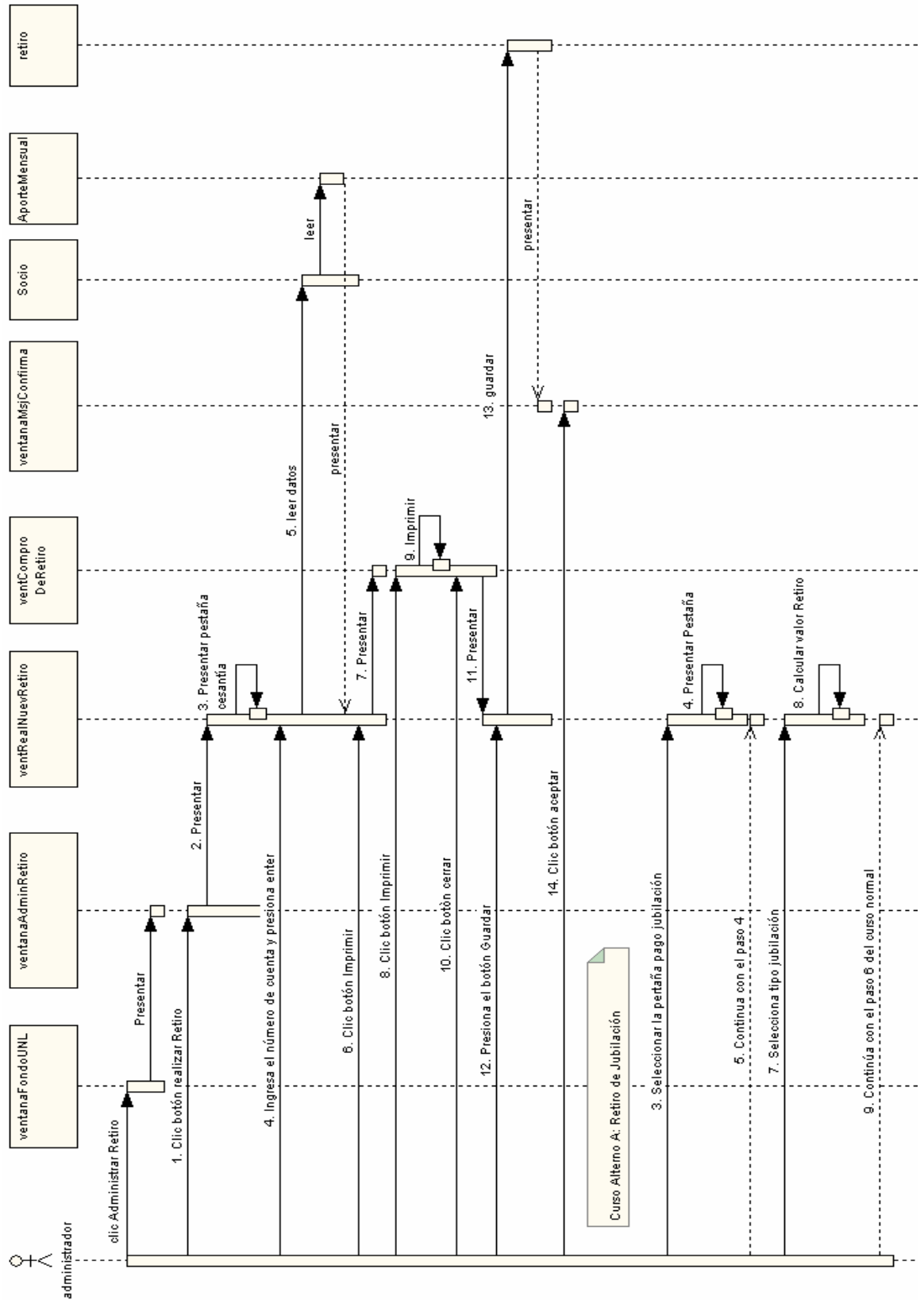
Curso alternativo B: Actualizar Préstamos



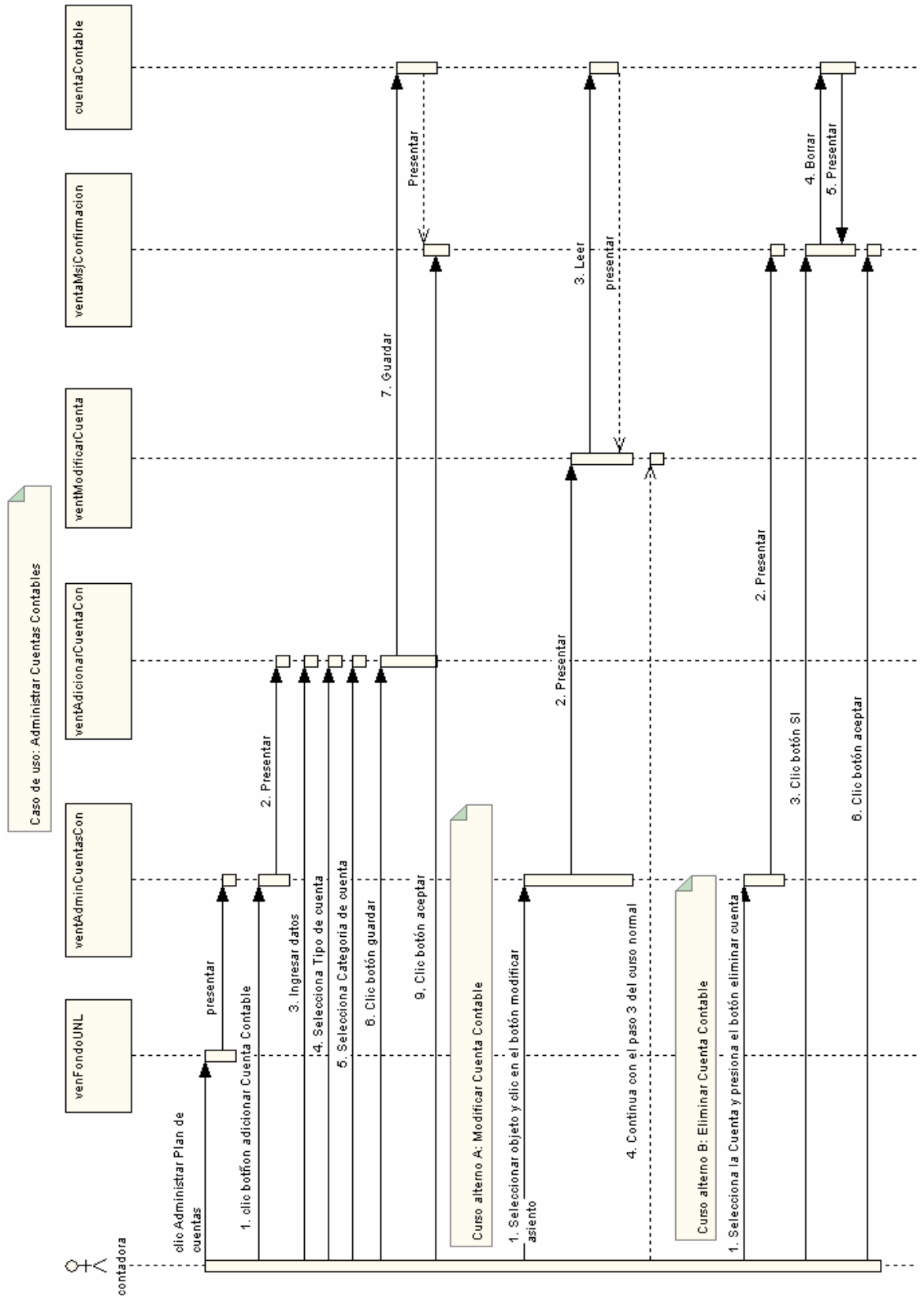
Curso alternativo C: Pago de cuotas por Préstamos



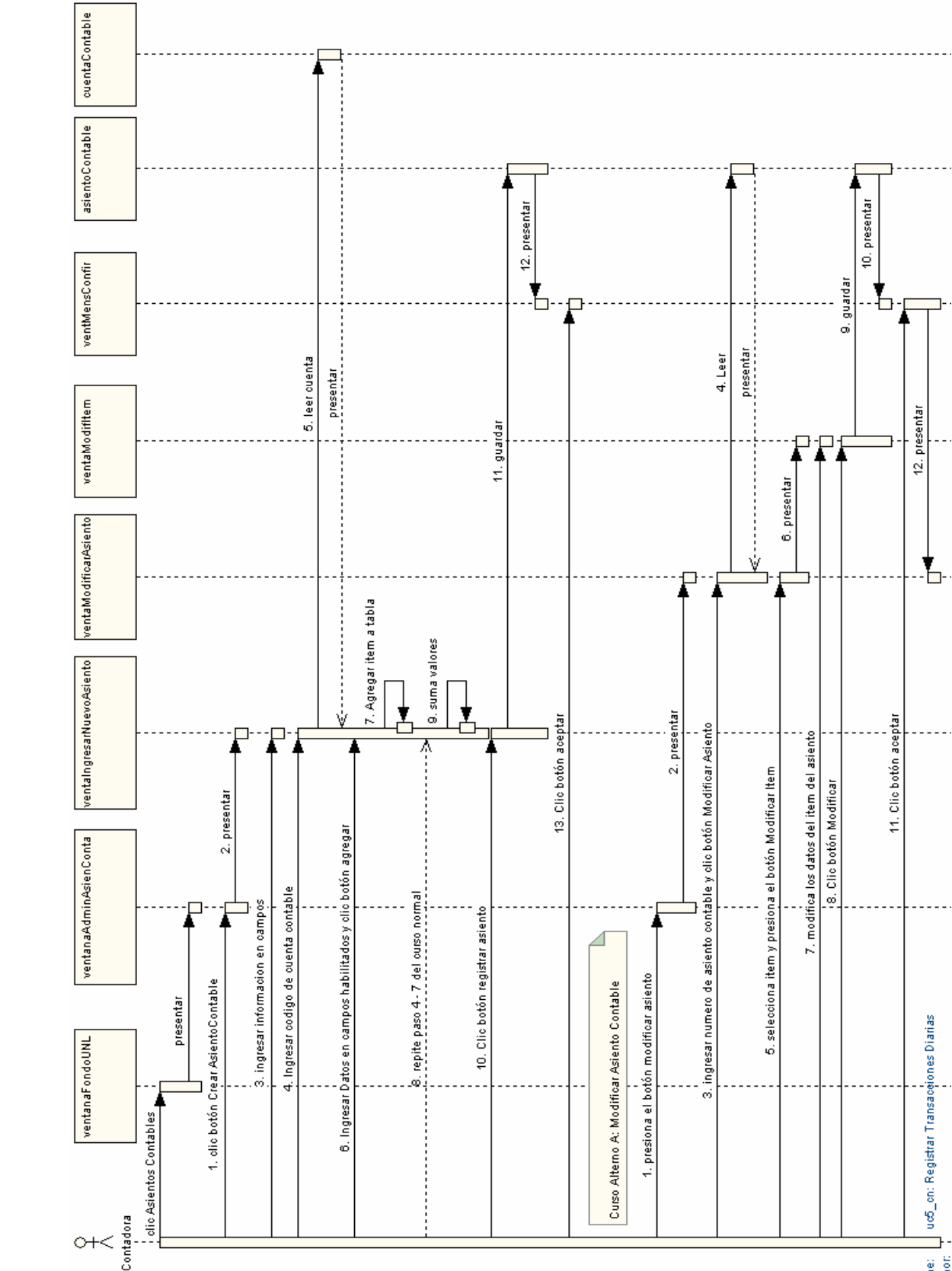
CASO DE USO: ADMINISTRAR RETIROS



CASO DE USO: ADMINISTRAR CUENTAS CONTABLES



CASO DE USO: INGRESAR TRANSACCIONES DIARIAS



Name:
Author:

uc5_c_ai: Registrar Transacciones Diarias1

Version: 1.0

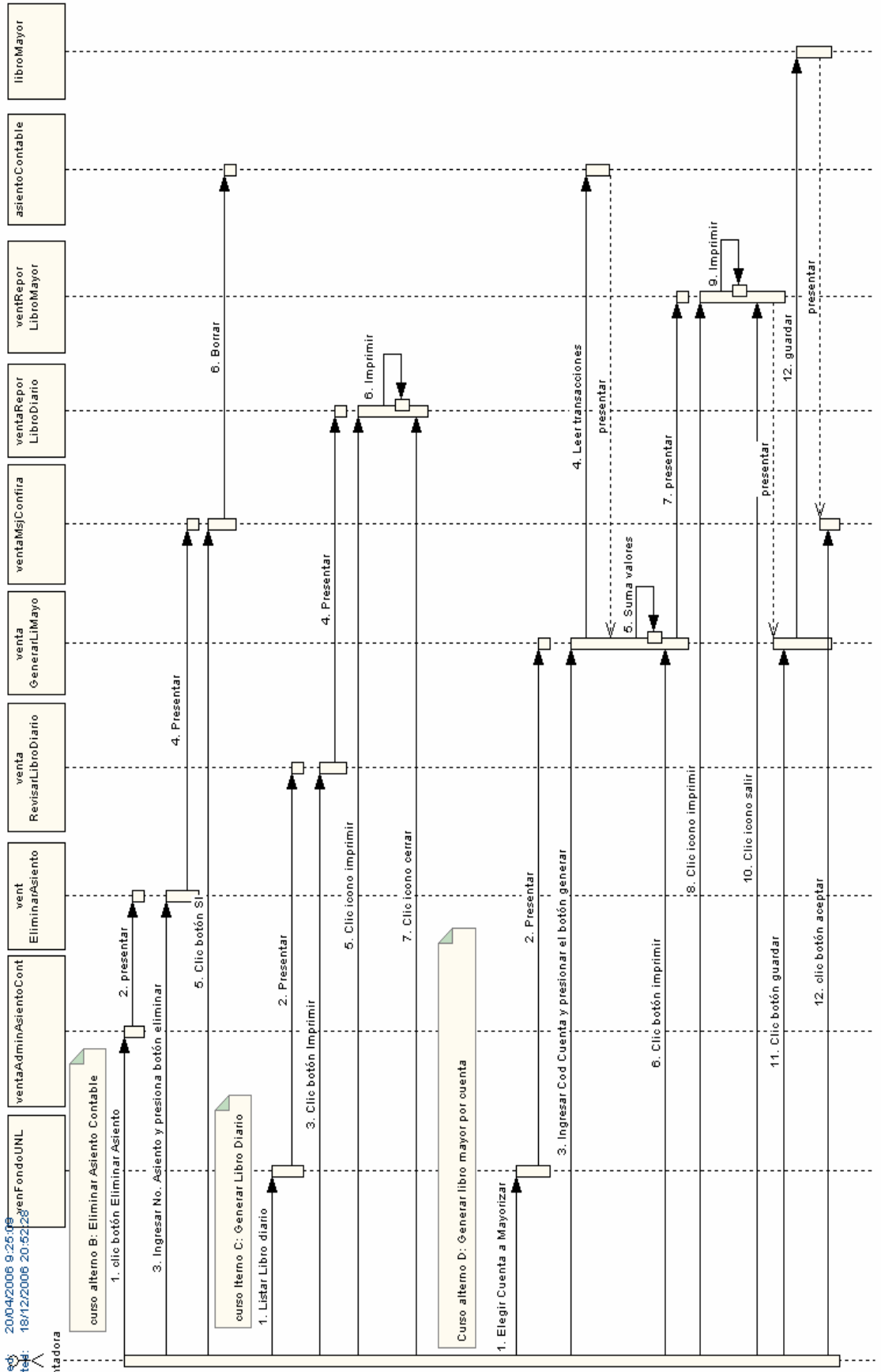
Author:

Created: 20/04/2006 9:25:09

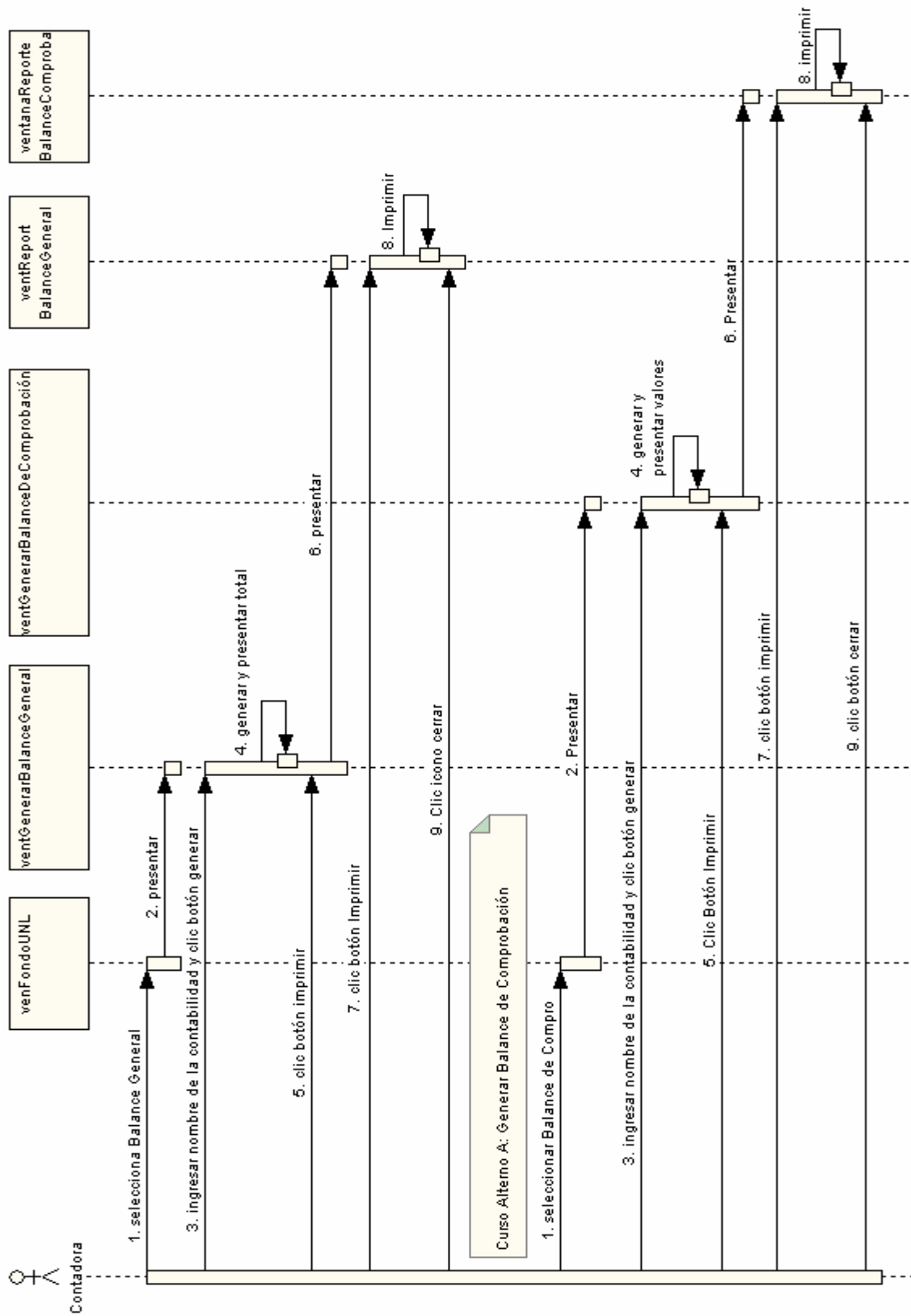
Updated: 18/12/2006 20:52:28

enFondoUNL

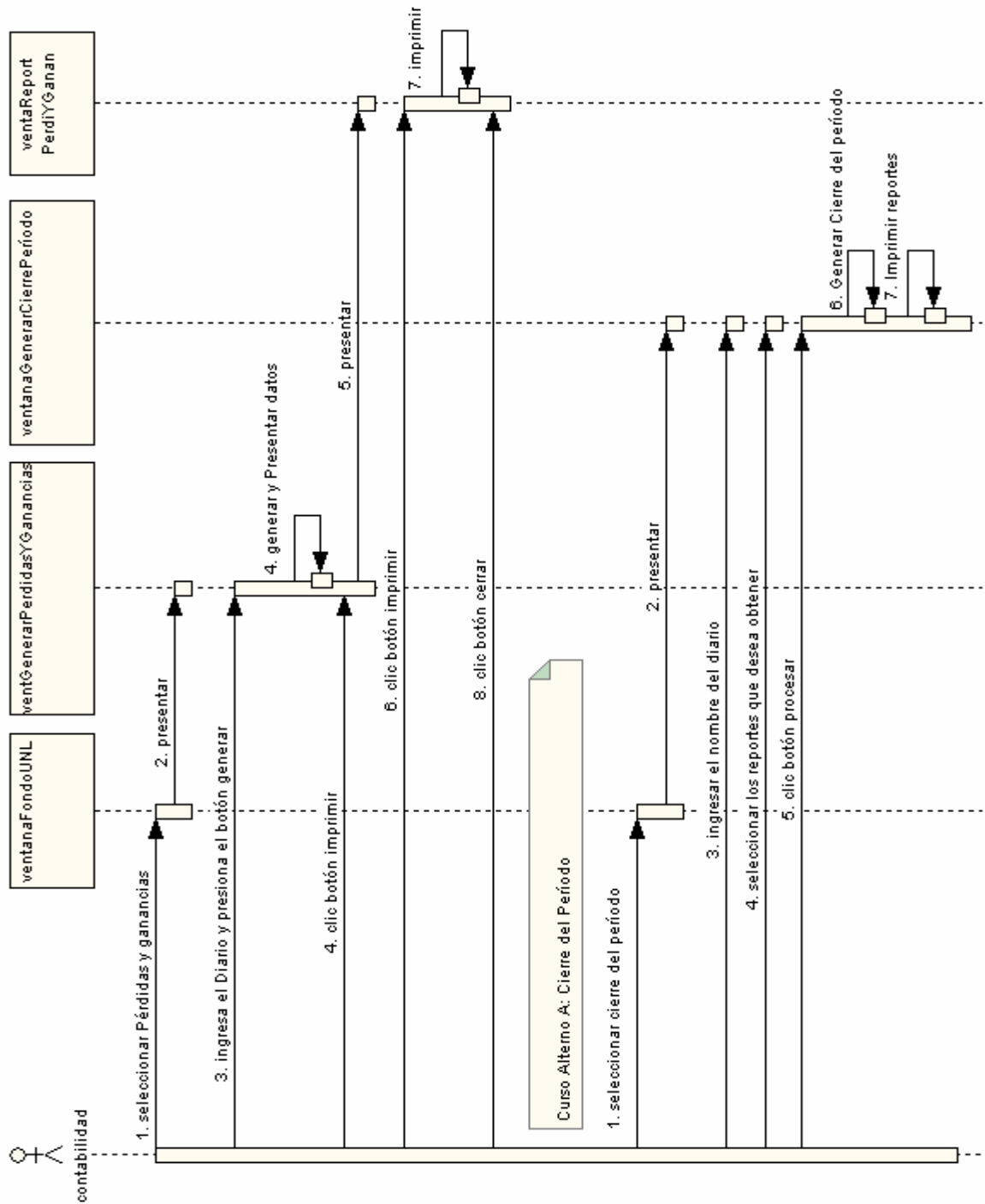
Contadora



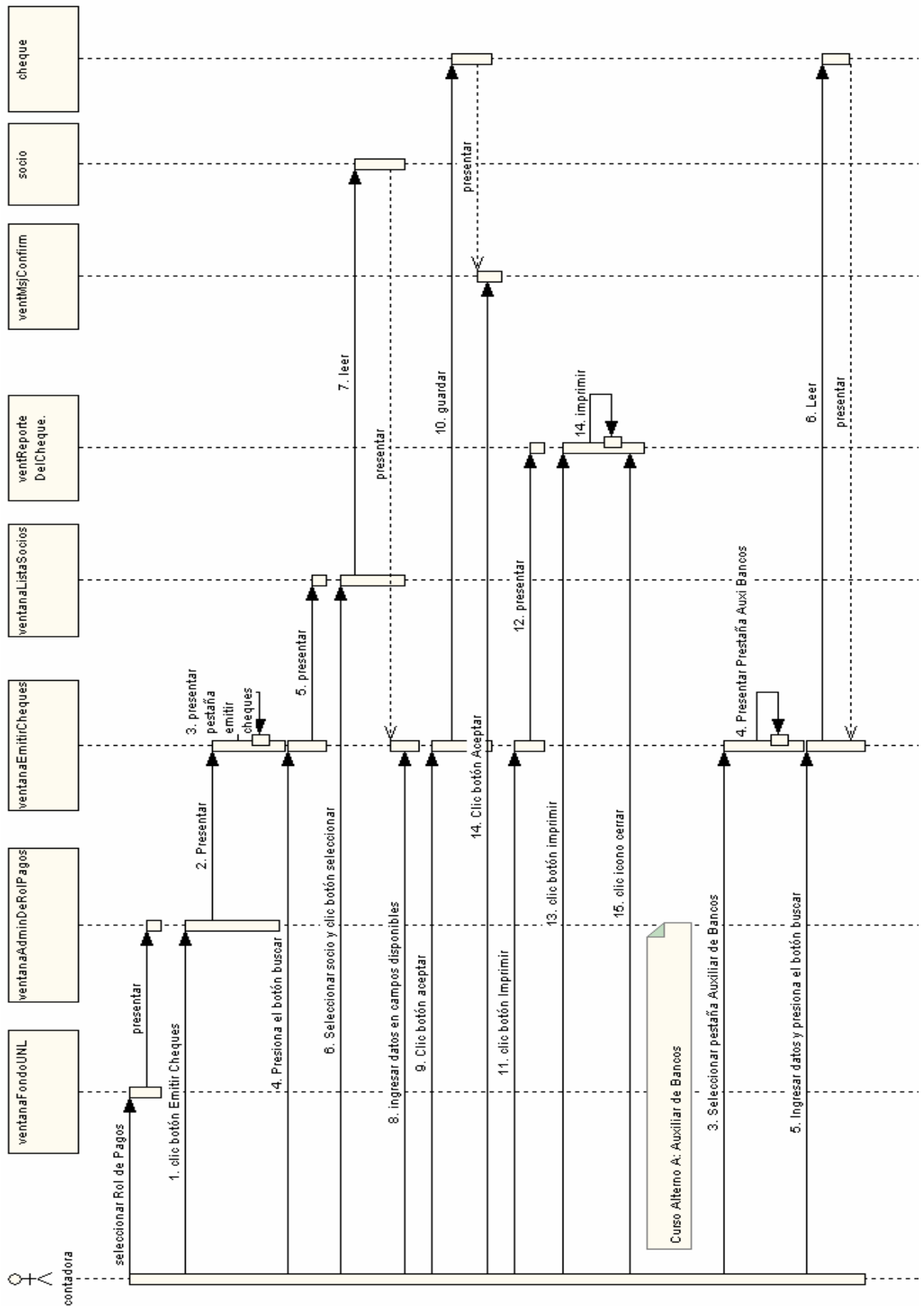
CASO DE USO: GENERAR BALANCES CONTABLES



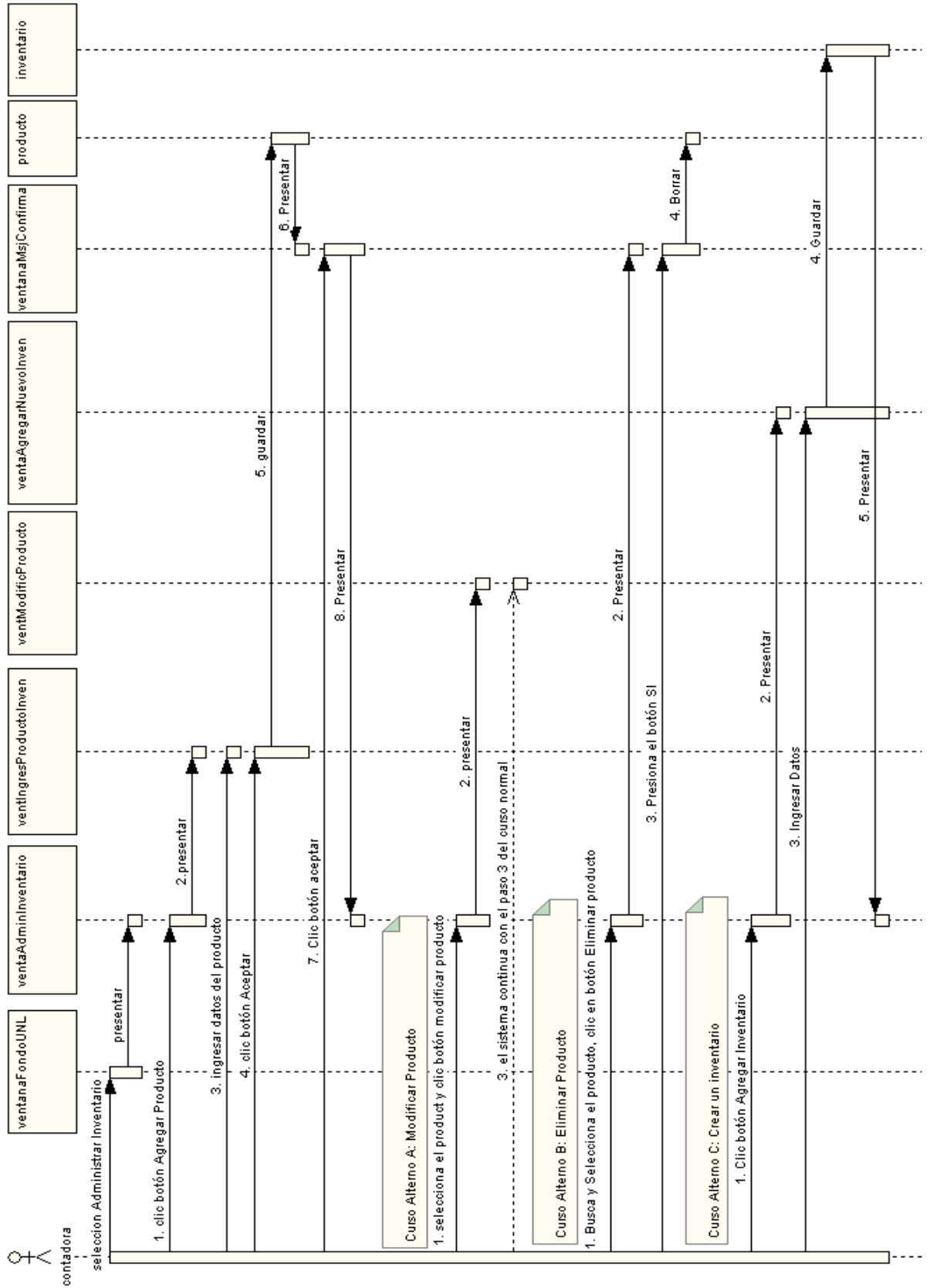
CASO DE USO: GENERAR ESTADO DE SITUACIÓN FINAL



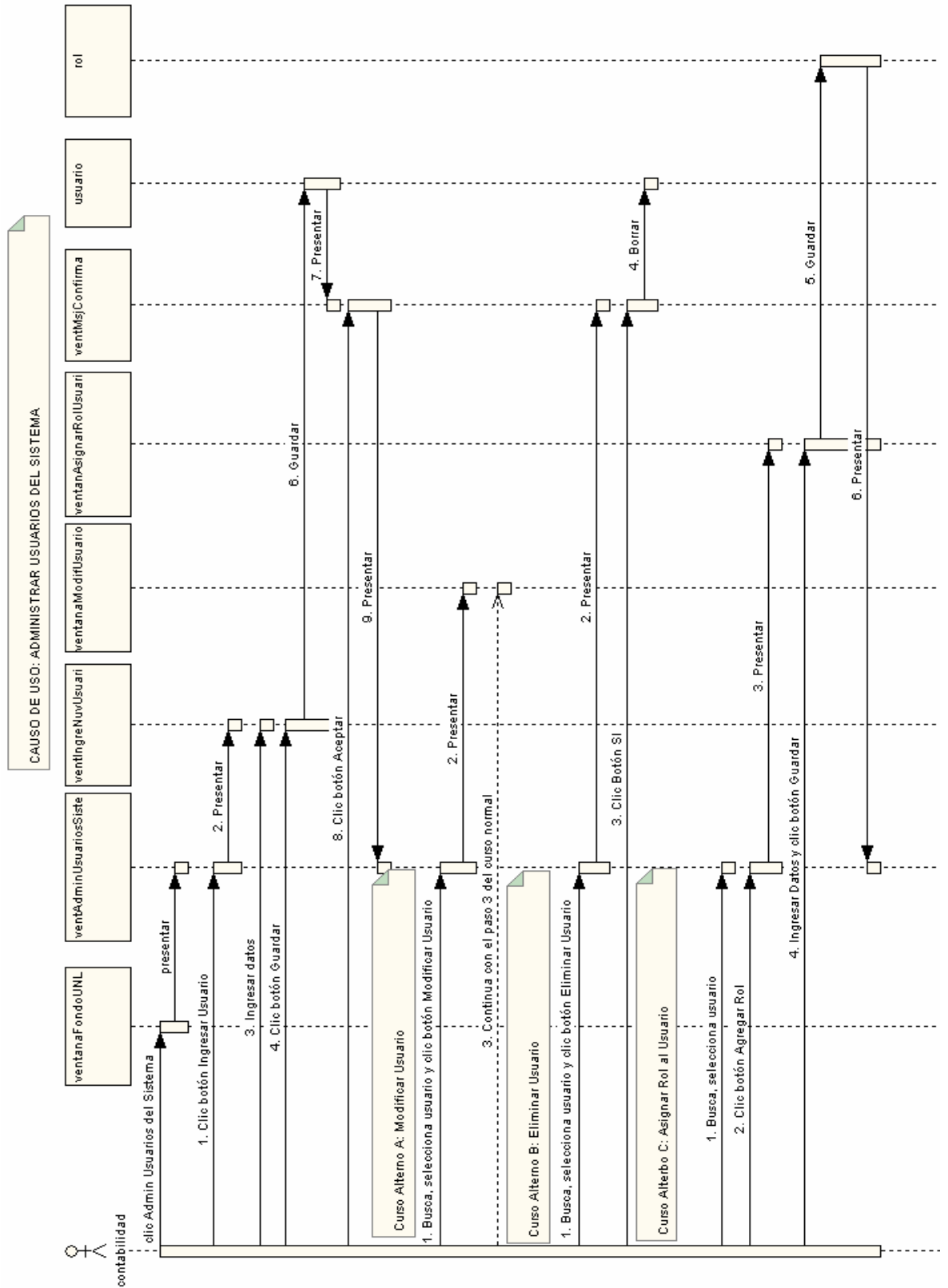
CASO DE USO: ADMINISTRAR ROL DE PAGOS



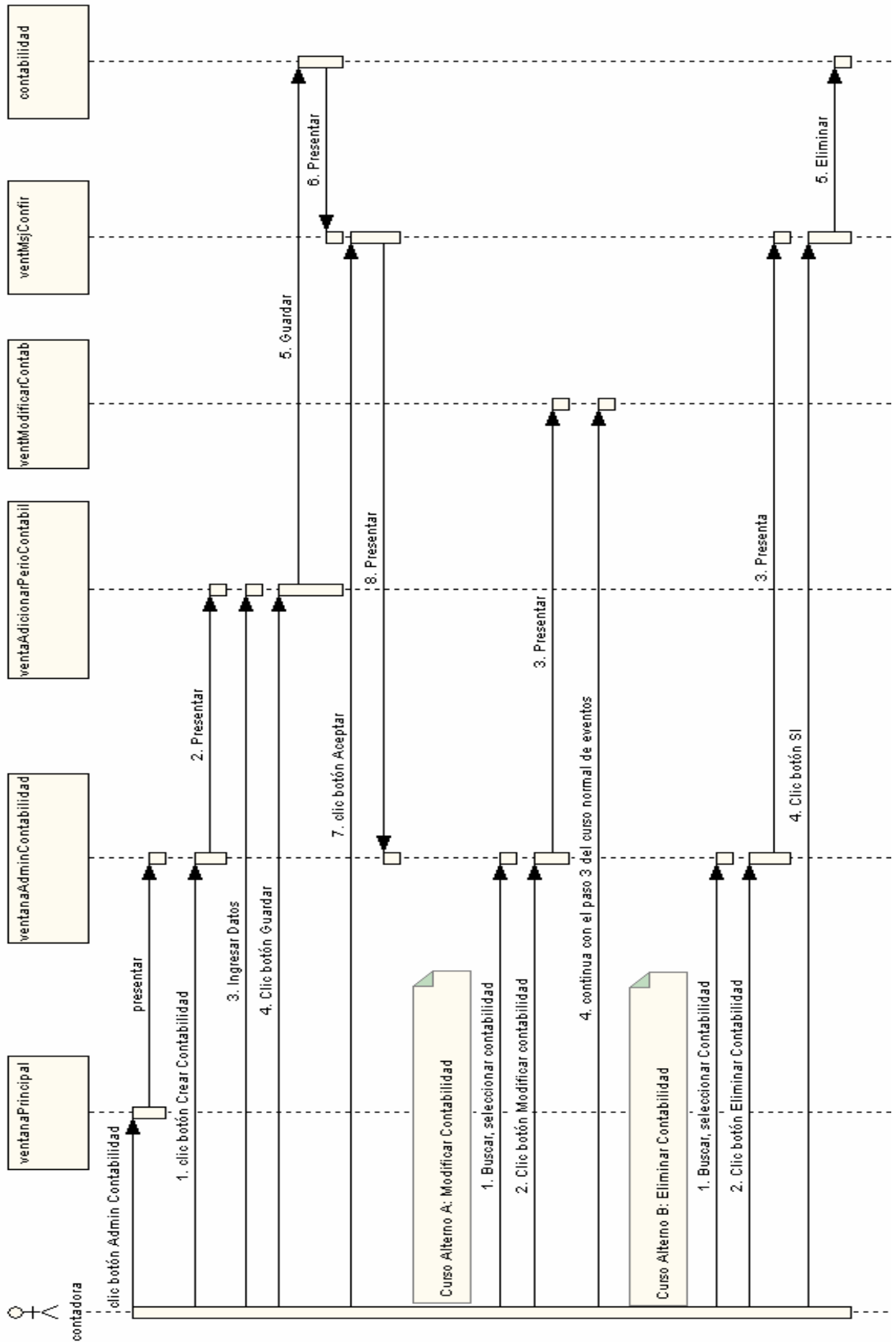
CASO DE USO: ADMINISTRAR INVENTARIO



CASO DE USO: ADMINISTRAR USUARIOS DEL SISTEMA

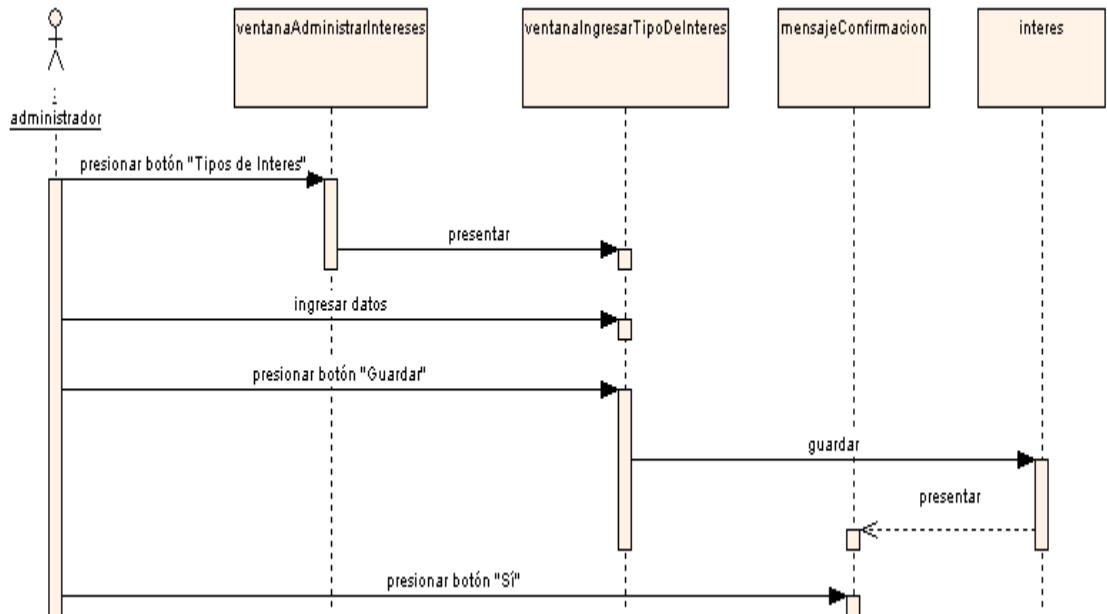


CASO DE USO: ADMINISTRAR CONTABILIDAD

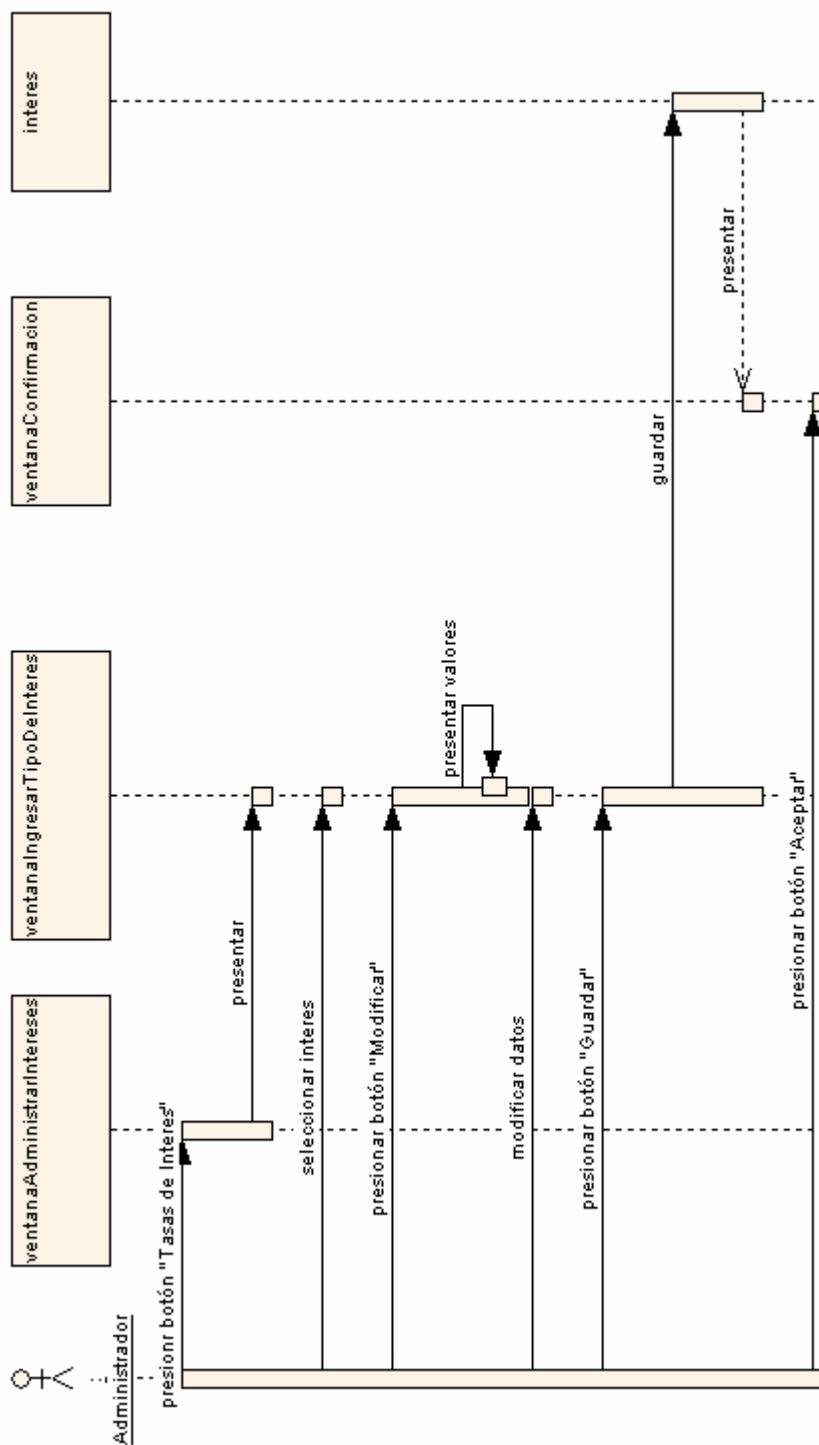


CASO DE USO: ADMINISTRAR INTERESES

Curso Normal de Eventos: Ingresar tipos de Interés



Curso Alterno A: Eliminar Tipos de Interés



Curso Alterno B: Ingresar Tasas de Interés

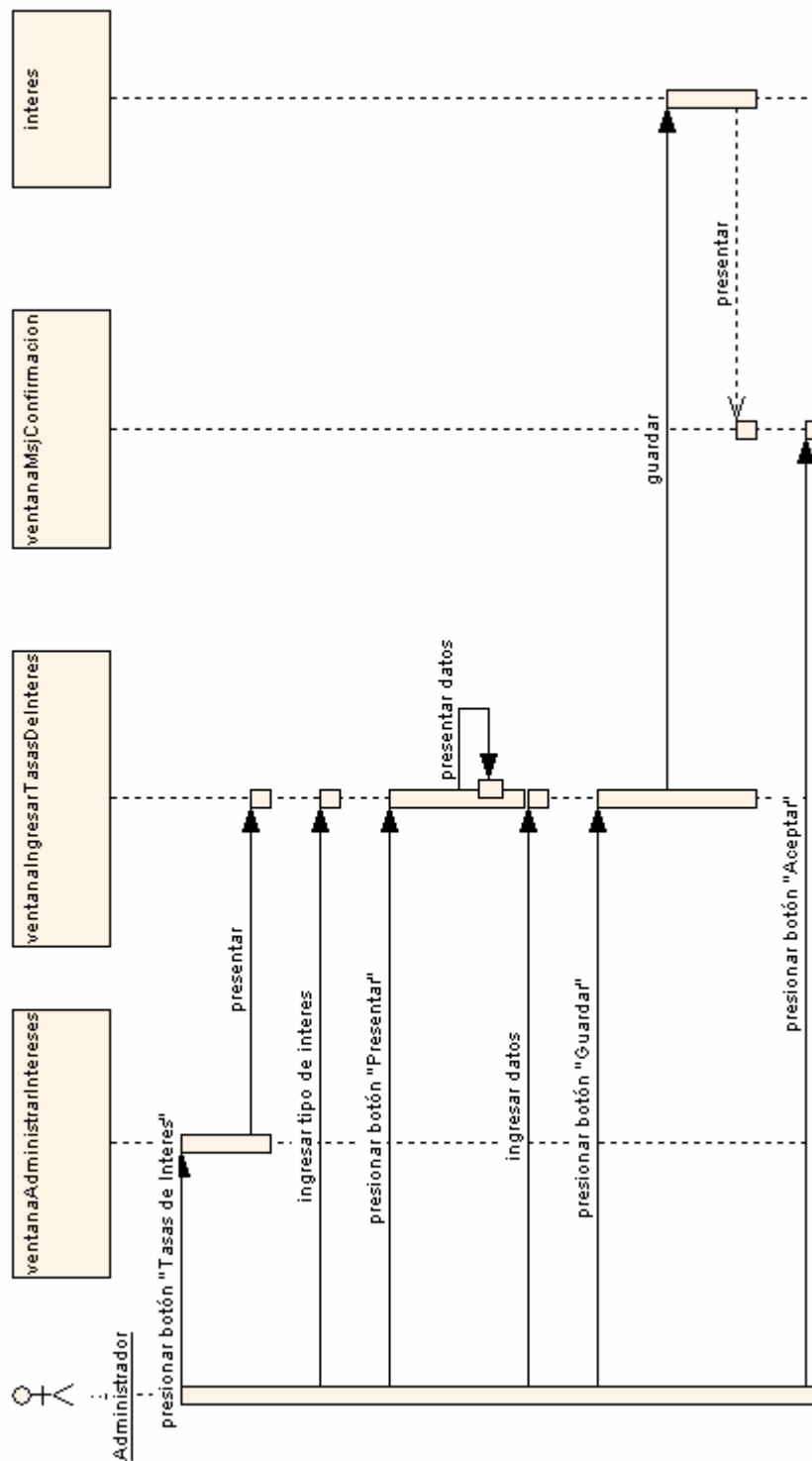
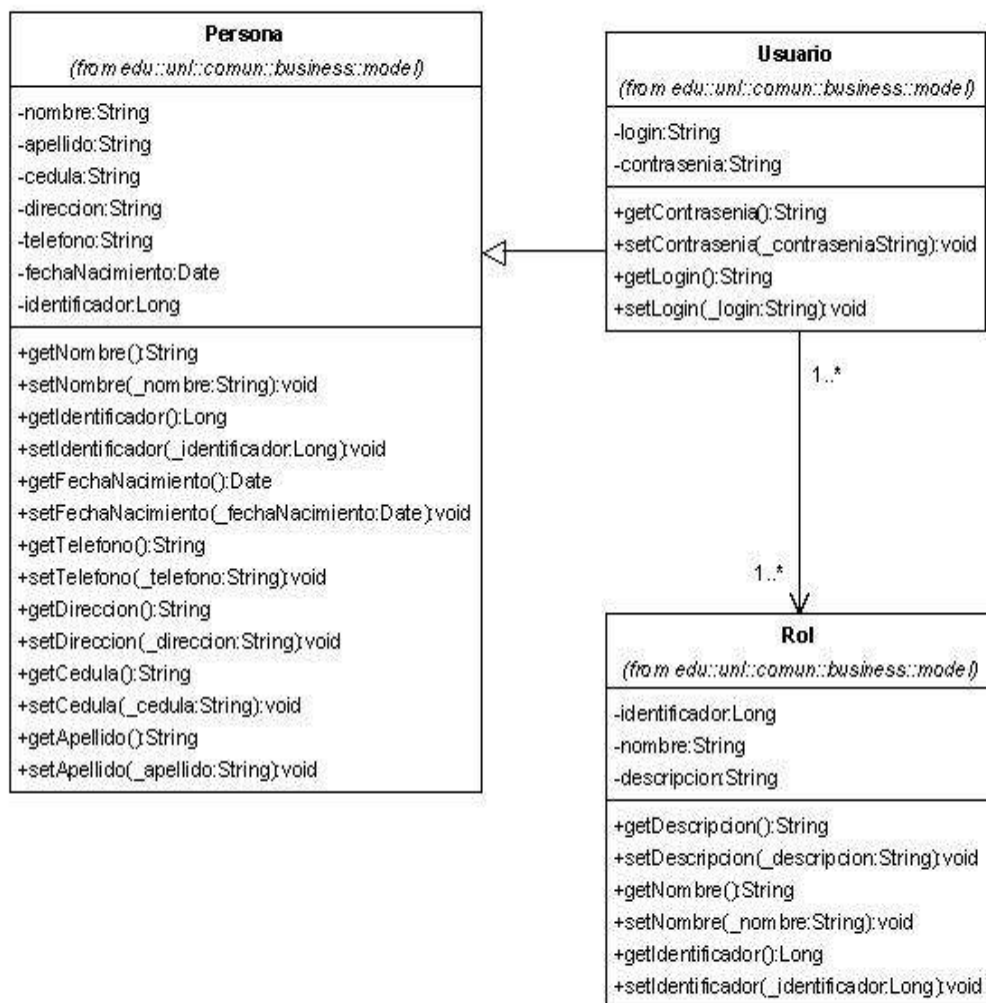


DIAGRAMA DE CLASES

A continuación se describen los diagramas de clases implementados en el sistema, que definen el comportamiento de cada objeto y además describen la forma en que se comunican cada uno de los objetos implementados en el sistema

diagrama comun



Created with Poseidon for UML Community Edition. Not for Commercial Use.

DISEÑO

DIAGRAMA DE PAQUETES

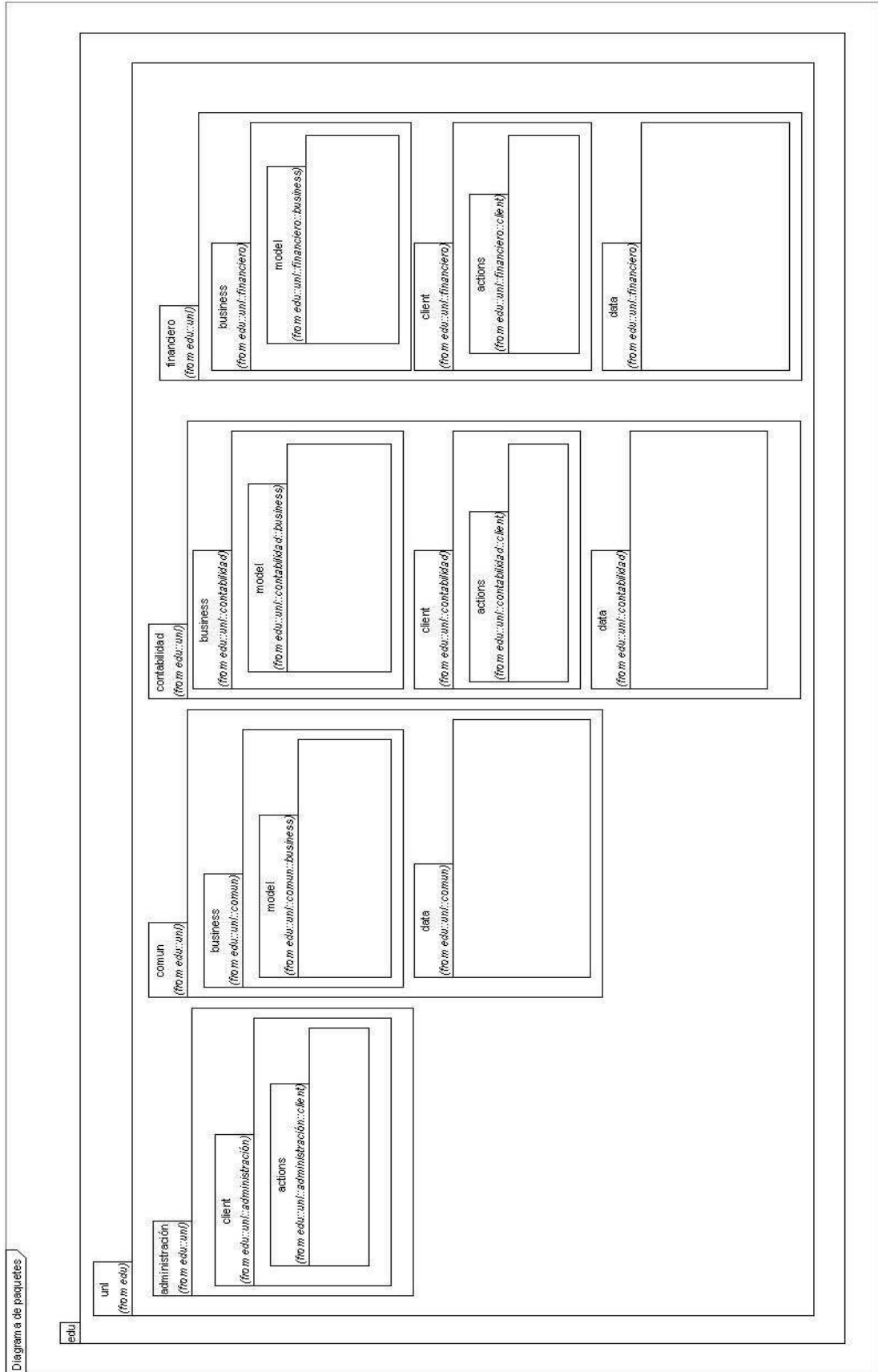
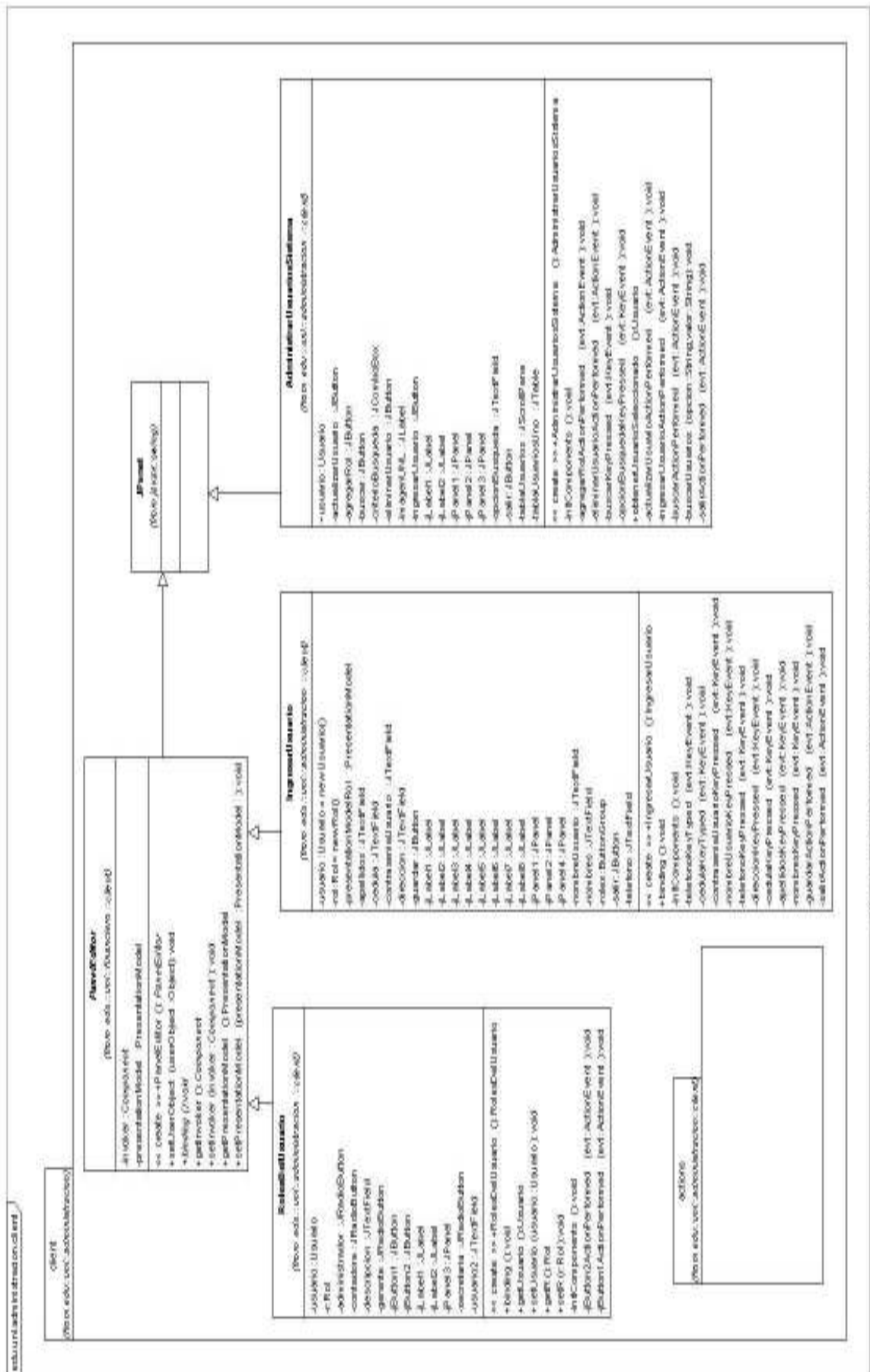
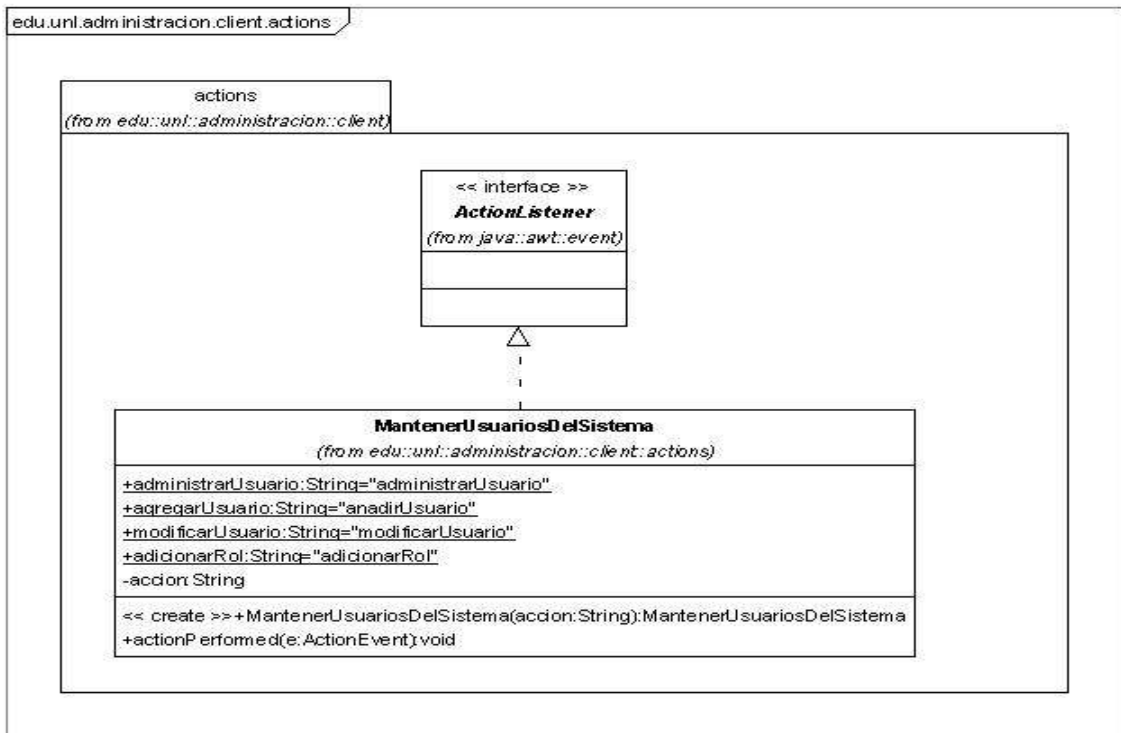
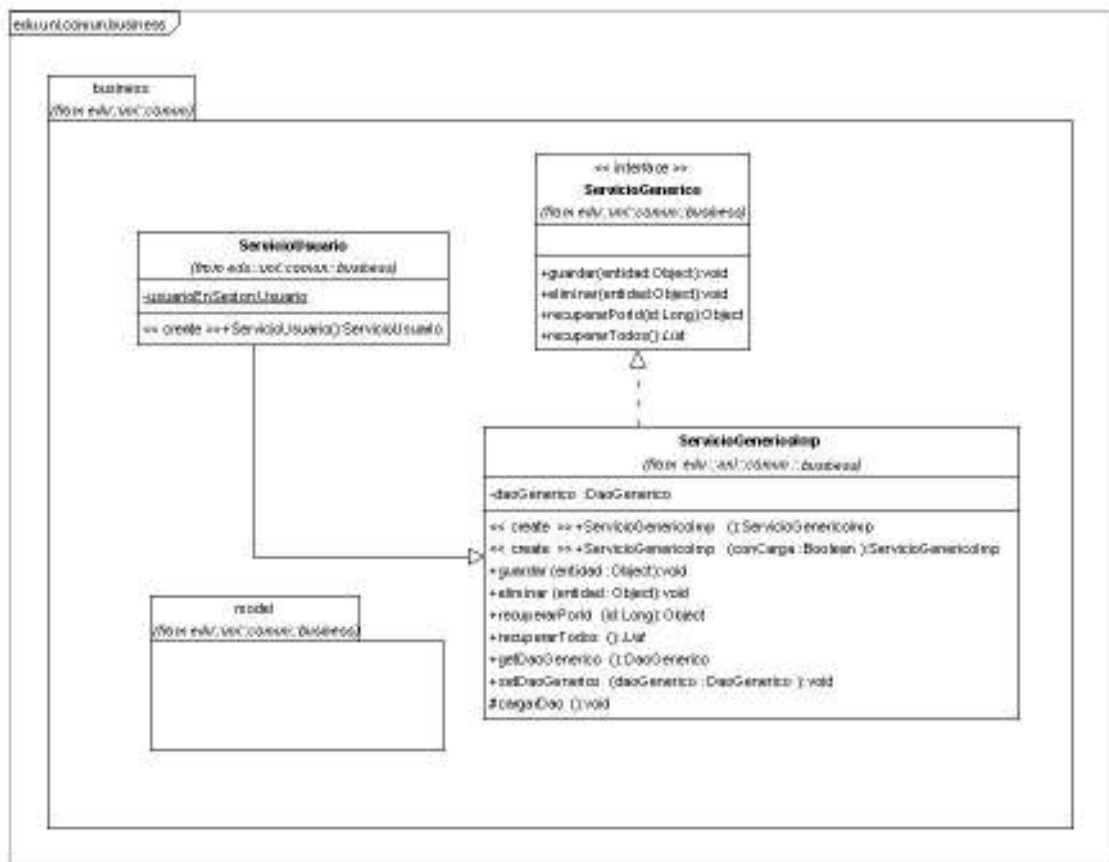


DIAGRAMA DE CLASES POR PAQUETE

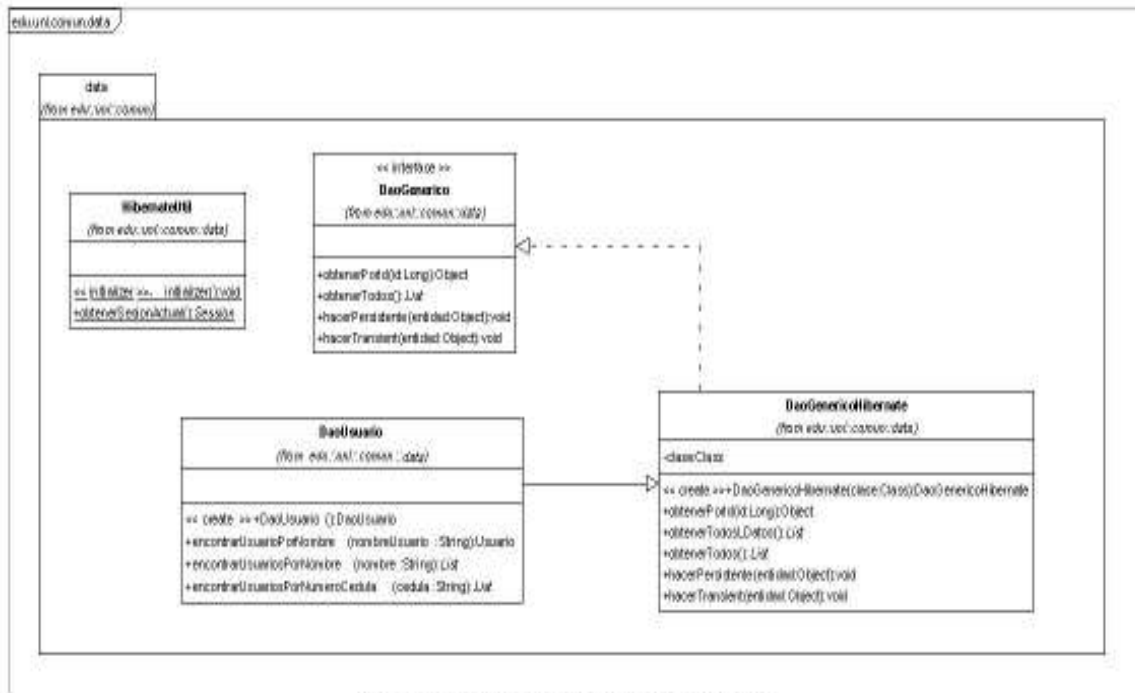
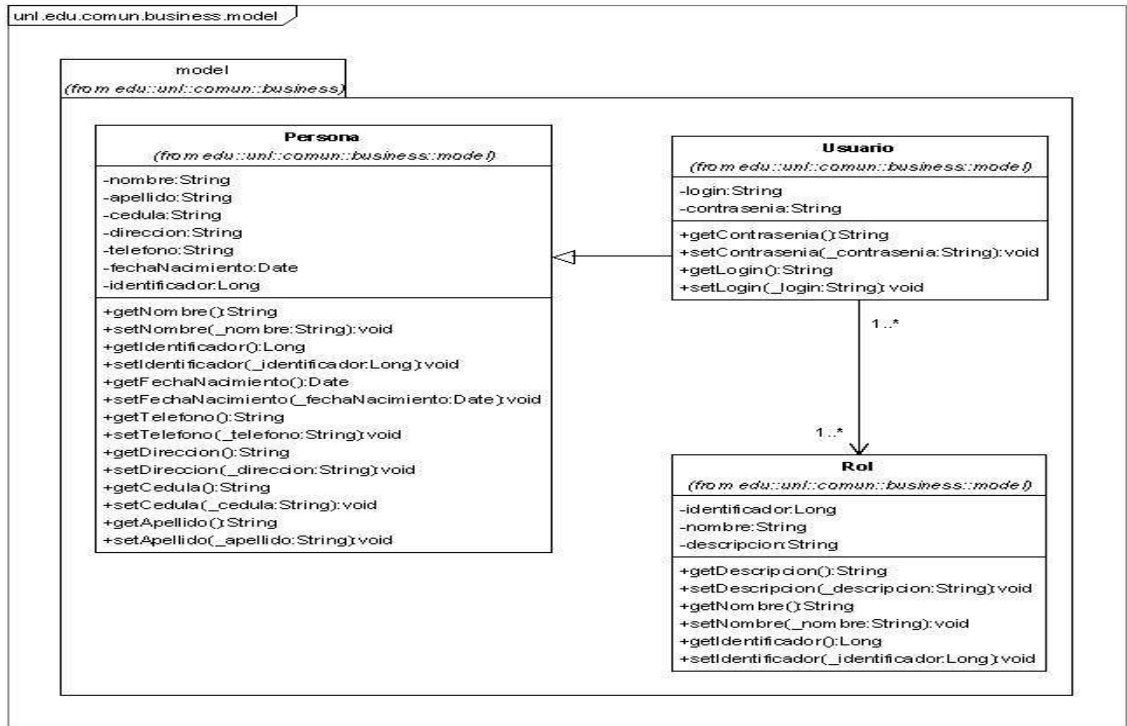


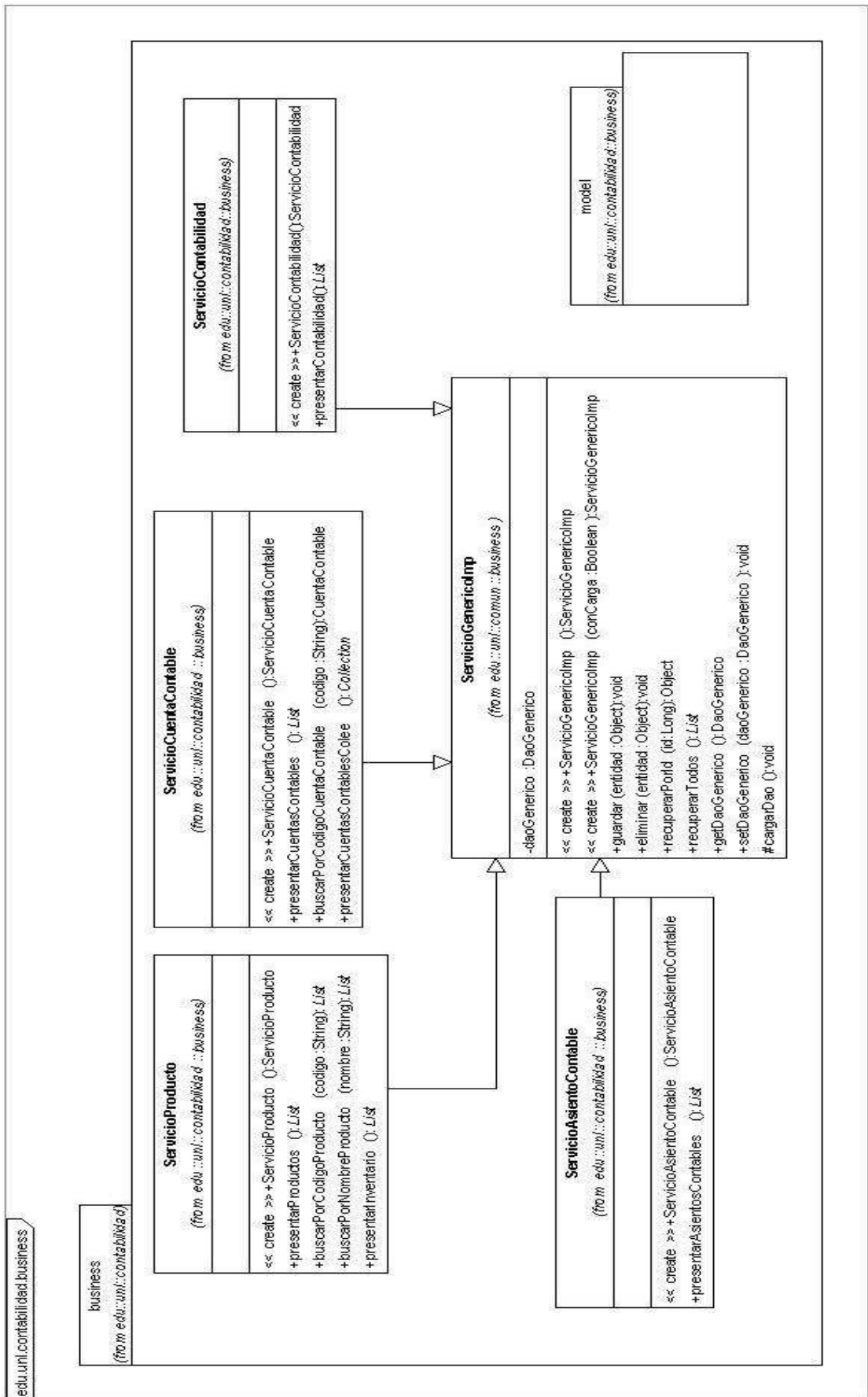


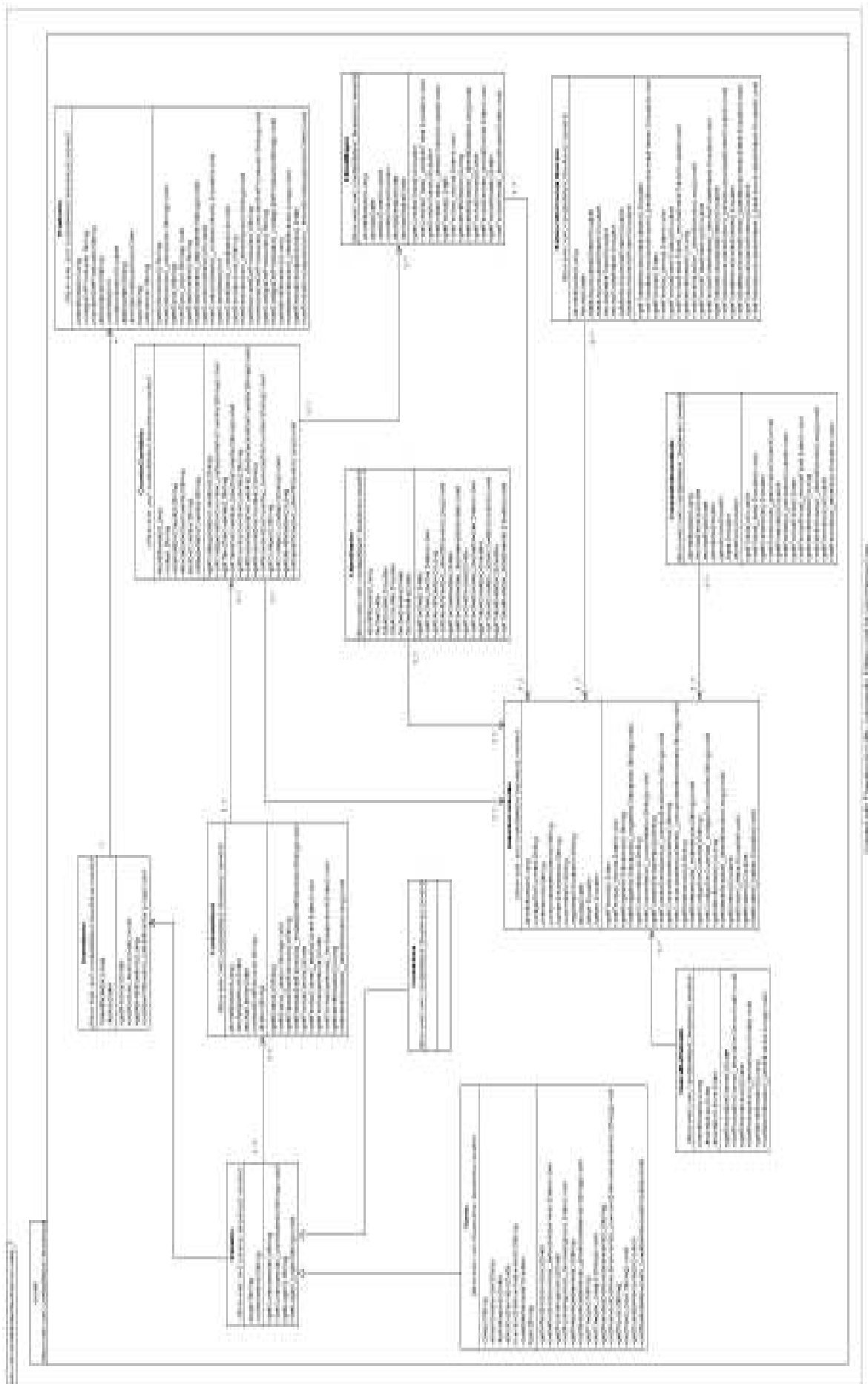
Created with Poseidon for UML Community Edition. Not for Commercial Use.

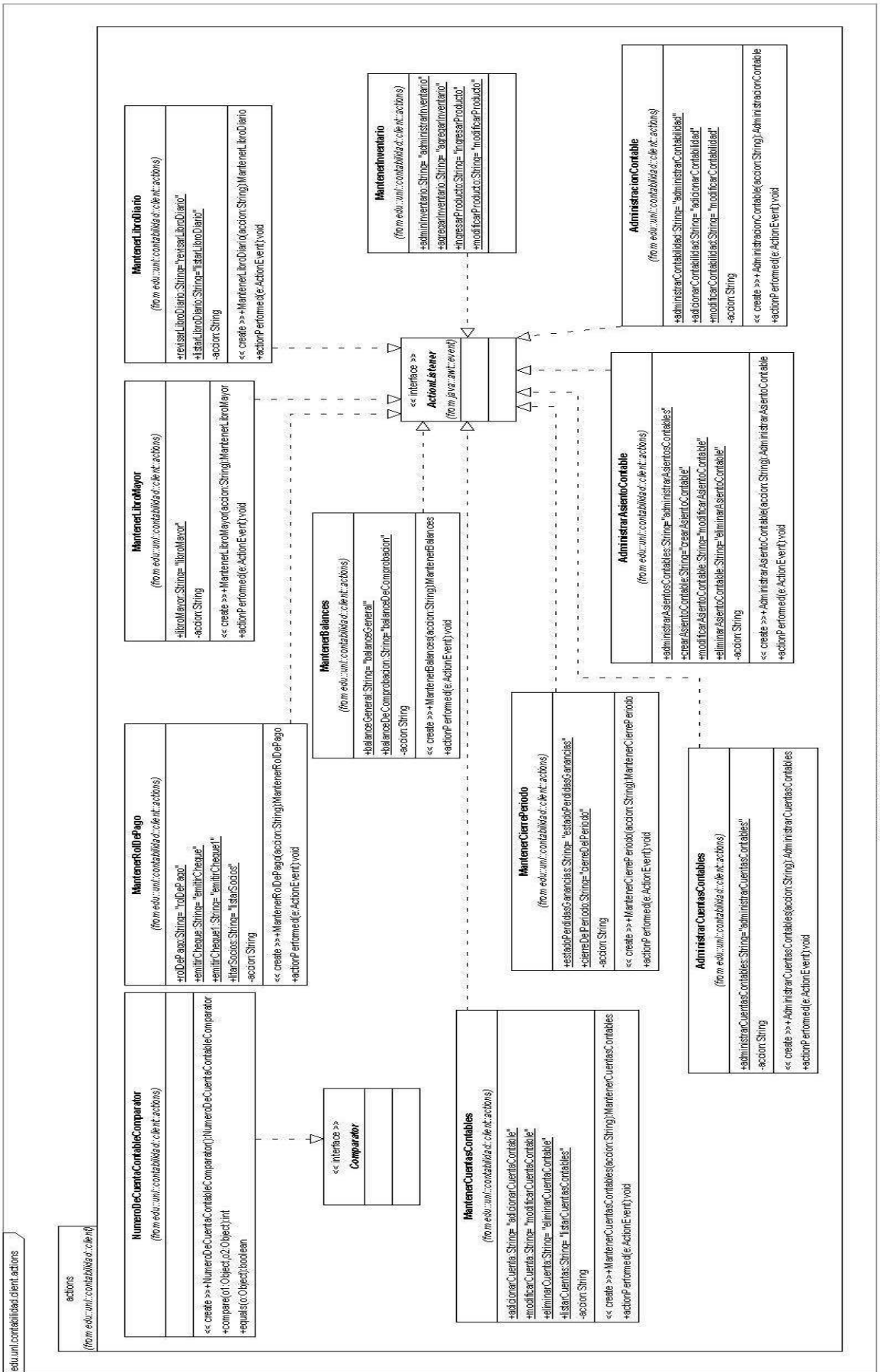


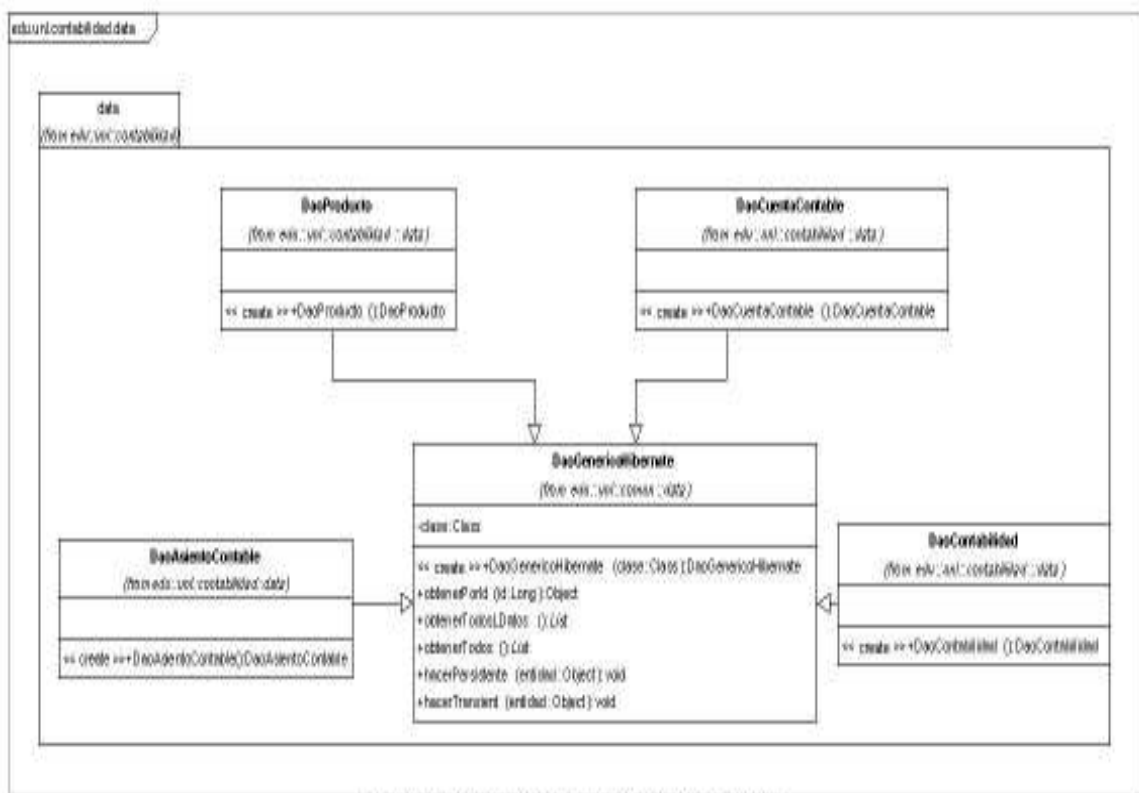
Created with Poseidon for UML Community Edition. Not for Commercial Use.





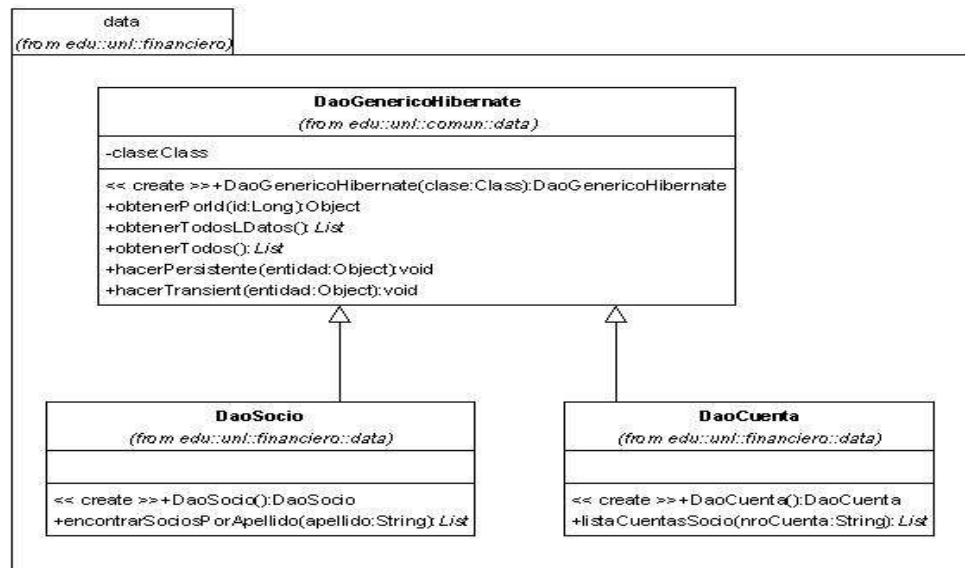






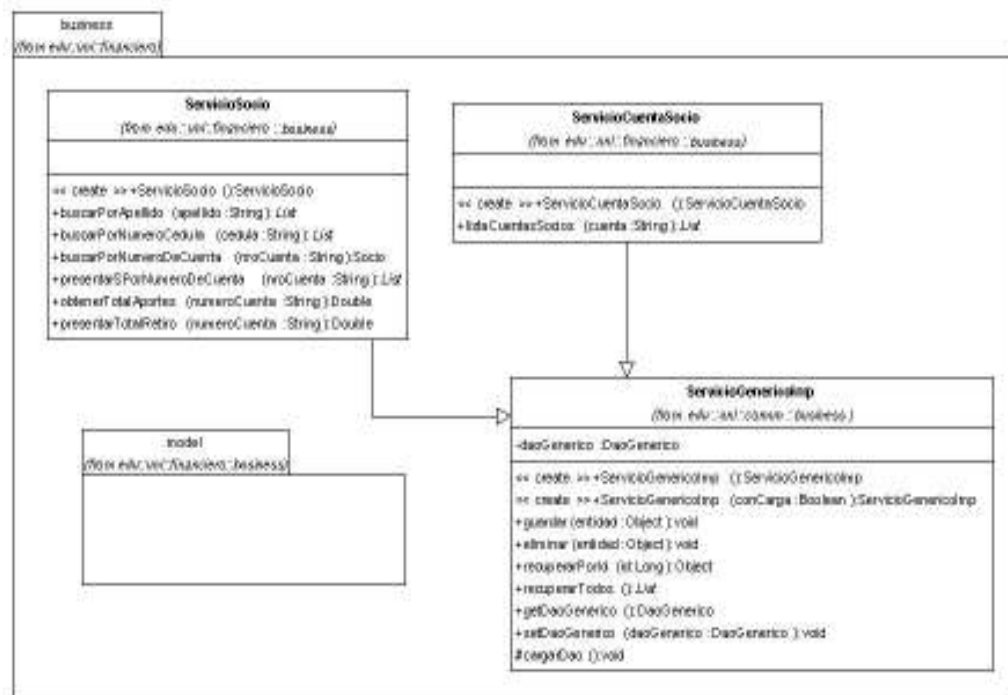
Created with Poseidon for UML Community Edition. Not for Commercial Use.

edu.unl.financiero.data

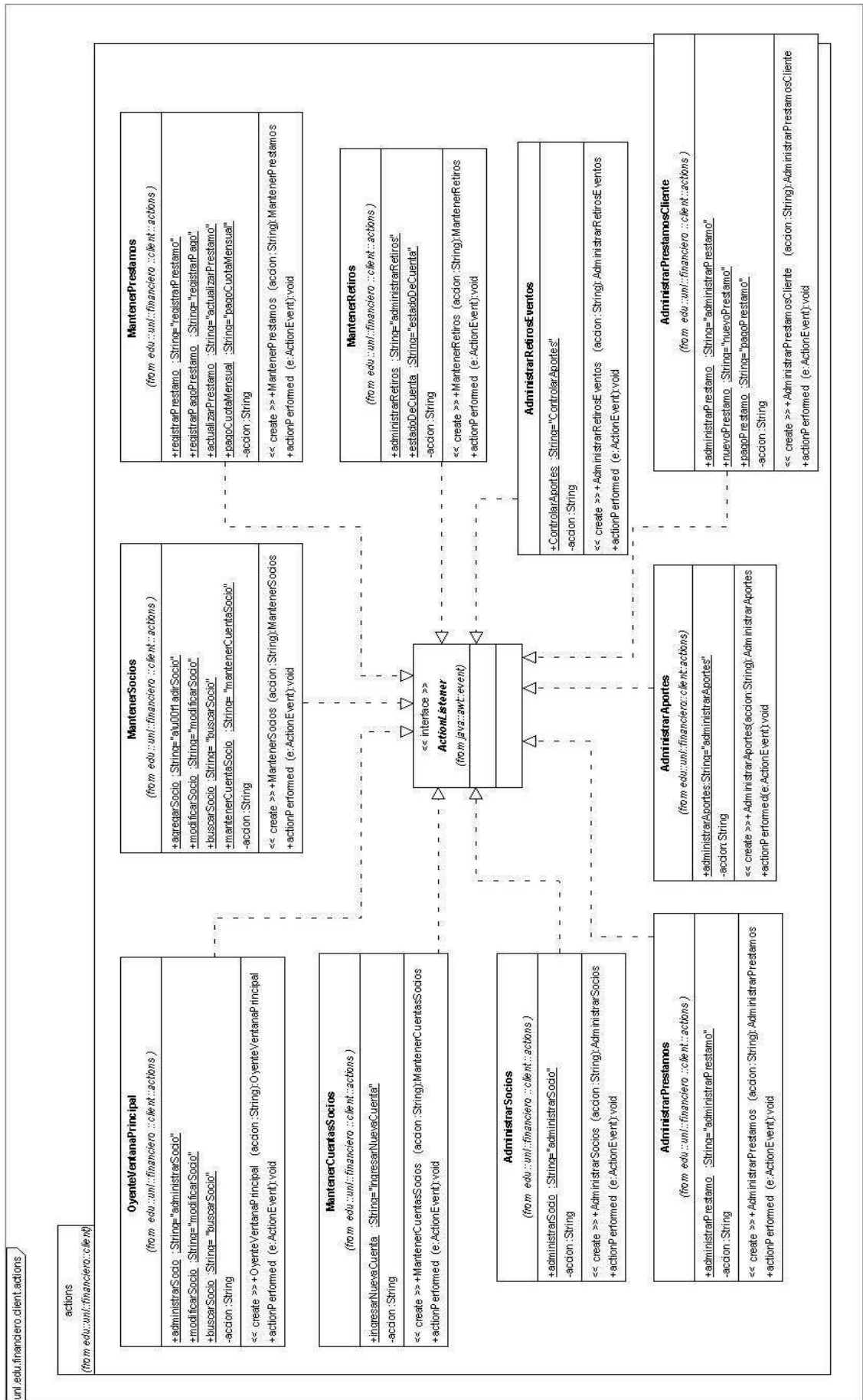


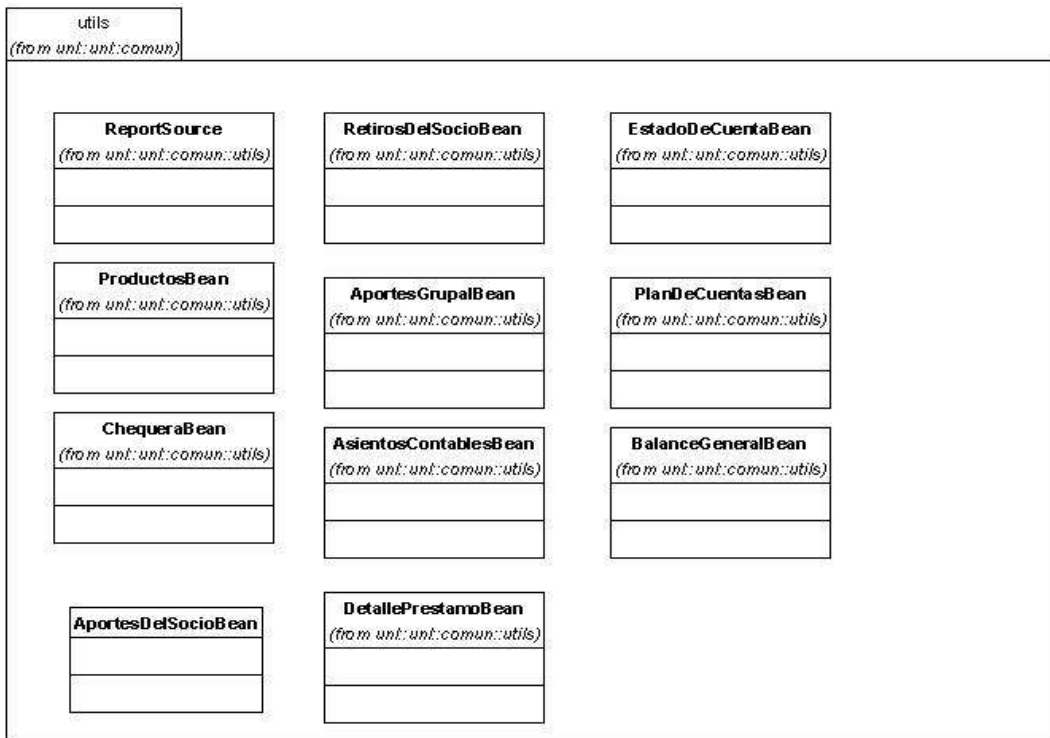
Created with Poseidon for UML Community Edition. Not for Commercial Use.

edu.unl.financiero.business



Created with Poseidon for UML Community Edition. Not for Commercial Use.





Arquitectura del Sistema

El Sistema fue diseñado por capas como se muestra a continuación:

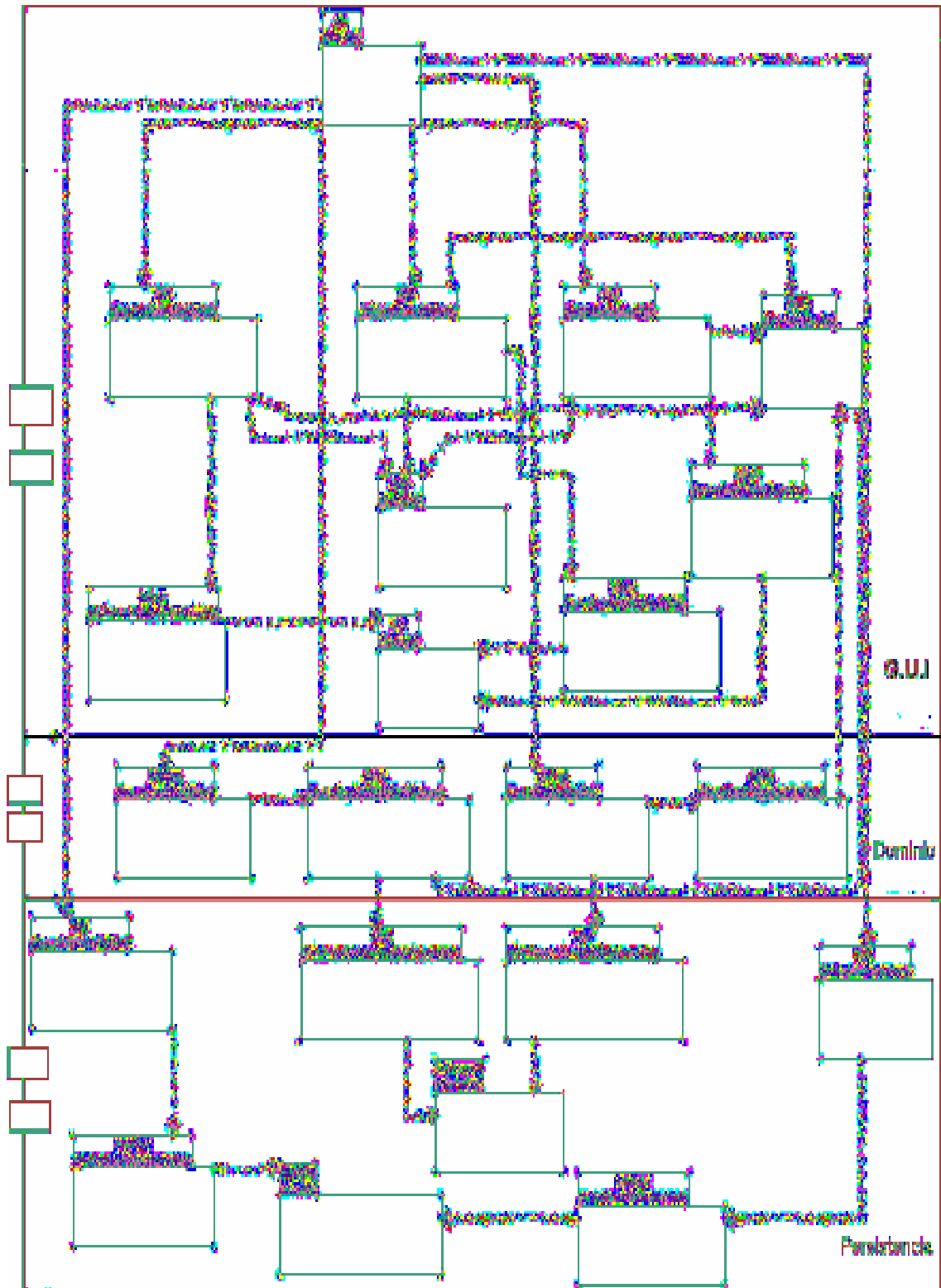
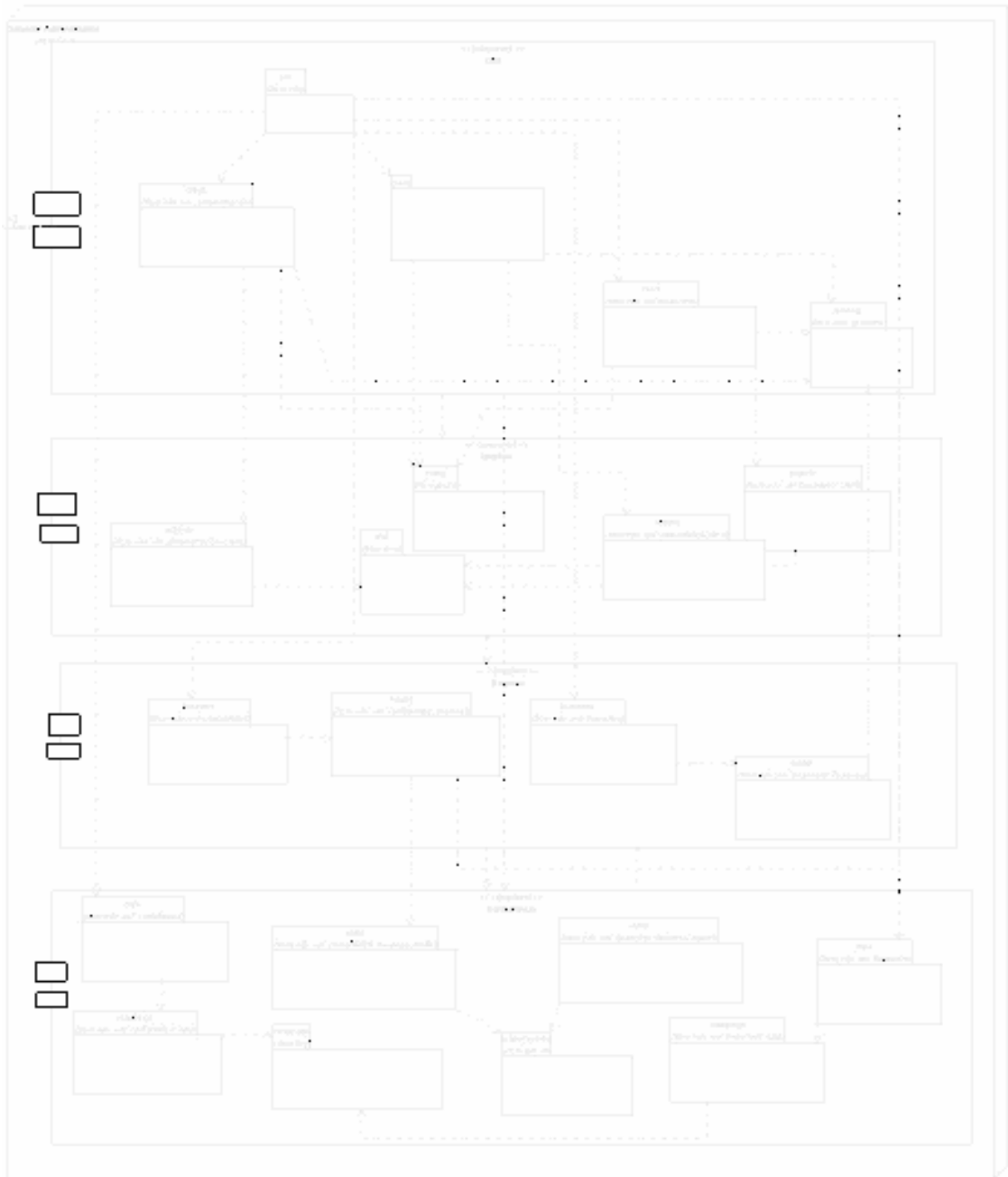


Fig. Diagrama de Capas

Diagrama de Componentes



CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Con la culminación del presente trabajo de Tesis, hemos llegado a concluir lo siguiente:

- ✎ La utilización del modelo de desarrollo ICONIX nos facilitó la identificación y definición de los procesos e información que se maneja en el Sistema Financiero Contable del Fondo de Cesantía y Jubilación Administrativo de la Universidad Nacional de Loja, gracias a la determinación directa de los requerimientos; y, a la aplicación de las etapas de desarrollo: dinámica y estática.
- ✎ La utilización de herramientas gráficas durante el proceso de desarrollo del Sistema Financiero Contable nos permitió acelerar y minimizar el tiempo de desarrollo de nuestra aplicación.
- ✎ La utilización de frameworks java como: jgoodies con binding, las librerías iReport, hibernate, jExcel, librerías gratuitas LookAndFields (Visualizaciones graficas), etc.; aceleran y mejoran el desarrollo de aplicaciones realizadas en JAVA; asimismo, mejora la visualización de los componentes Swing y JFC, a un entorno gráfico más agradable al usuario.
- ✎ Por medio del sistema Financiero Contable se pudo automatizar el área Financiera y Contable del Fondo de Cesantía y Jubilación de la Universidad Nacional de Loja, agilizando y mejorando los procesos internos.
- ✎ Los objetivos planteados de analizar, diseñar e implementar un sistema Financiero-Contable; con la implementación del sistema se cumplieron en un 100%.
- ✎ El desarrollo del Sistema Financiero Contable para Fondo de Cesantía y Jubilación del Trabajador Administrativo de la Universidad Nacional de Loja, nos permitió seguir capacitándonos e investigando, así como llevar a la

práctica todos los conocimientos adquiridos en el transcurso de nuestros estudios universitarios.

- ✎ La implementación de la base de datos MySQL fue bastante sencilla gracias a la aplicación en el sistema de motor de persistencia de datos **Hibernate**, que facilita el manejo de transacciones (guardar, borrar, modificar, consultar) desde la Interfaz de Usuario (realizada en JFC/swing de java) con MYSQL.
- ✎ La utilización de motores de persistencia de Datos manejan de manera sencilla el mapeo de datos y facilitan la conexión, y actualización de los datos que se encuentran en una Base de Datos, haciendo transparente los sistemas orientados a objetos con el Modelo Entidad Relación.
- ✎ La administración de usuarios del sistema mediante de uso de nombres de usuario y contraseñas permitió dar seguridades en el acceso al sistema Financiero Contable para mantener la integridad de los datos y además poder obtener el reporte de transacciones que realiza el usuario en el sistema con la finalidad de realizar la Auditoria del Sistema.

5.2. RECOMENDACIONES

Finalizando el presente trabajo nos permitimos plantear las siguientes recomendaciones:

- ✎ En la etapa de análisis se debe procurar obtener la mayor cantidad de información posible de la empresa o Institución para la cuál se desarrolla el sistema, para en lo posterior no tener problemas de no saber la manera en que funciona tal o cuál proceso.
- ✎ Se motive el uso de tecnologías de software libre en el desarrollo de proyectos de Tesis dentro de la Carrera de Ingeniería en Sistemas, ya que estas tecnologías engloban un amplio margen de trabajo y se desarrollan software de alta calidad, eficiencia y seguridad, además se cuenta con mucha información disponible en libros, Internet, etc.
- ✎ Utilizar modelos de desarrollo de Software como el Modelo ICONIX, iconix usa diagramas básicos UML que pueden ser aplicados por cualquier analista de sistemas en diferentes proyectos de desarrollo.
- ✎ Para lograr el uso óptimo del sistema, se recomienda utilizar soporte de Hardware de última generación, como un procesador Pentium IV o superior, 512 MB de memoria, un monitor de 15 pulgadas y con una buena resolución.
- ✎ Utilizar las diferentes librerías JAVA que permiten interactuar de mejor manera con otras aplicaciones y acceder de manera directa con los componentes swing. Estas librerías pueden descargarse en código fuente y además cuentan con un gran respaldo para solucionar problemas en los foros que se encuentran en la WEB.
- ✎ Utilizar IDE's comunitarios (software libre) para acelerar la etapa de implementación, estos IDE's cuentan con la ayuda necesaria mientras se está escribiendo el código y herramientas gráficas que facilitan la edición gráfica de formularios o Interfaces gráficas. Además cuentan con soporte mundial a través de muchos sitios web.

- ✎ La Universidad Nacional de Loja por medio de la Carrera de Ingeniería en Sistemas apoye y difunda la investigación y desarrollo de software basado en tecnologías "OPEN SOURCE" para lograr la eficiencia académica y el fortalecimiento académico de la Carrera.

- ✎ Para el manejo del Sistema Financiero-Contable en su parte contable debe ser utilizado por un Contador Público, para que exista coherencia en las transacciones contables diarias que realiza el Fondo de Cesantía.

TABLA DE ILUSTRACIONES

<i>Figura 1: Modelo de desarrollo Iconix</i>	<i>13</i>
<i>Figura 2: Símbolo UML de la Representación de una clase</i>	<i>21</i>
<i>Figura 3: Símbolo UML de la Clase Lavadora</i>	<i>21</i>
<i>Figura 4: Procedimiento del Modelado del Dominio (Parte1).....</i>	<i>22</i>
<i>Figura 5: Procedimiento para construir el modelo de dominio (Parte 2)</i>	<i>22</i>
<i>Figura 6: Representación de un Caso de Uso.....</i>	<i>26</i>
<i>Figura 7: Diagrama de Casos de Uso</i>	<i>27</i>
<i>Figura 8: Relaciones entre objetos.....</i>	<i>30</i>
<i>Figura 9: Diagrama de Secuencia</i>	<i>32</i>
<i>Figura 10: Diagrama de Secuencia Detallado</i>	<i>33</i>
<i>Figura 11: Diagrama de Clases.....</i>	<i>34</i>

BIBLIOGRAFÍA

1. E. Castillo, A. Cobo, P. Gómez, C. Solares: JAVA TM un lenguaje de programación multiplataforma para Internet, España, PARAINFO.
2. GUTIÉRRES, Abraham. Métodos de Investigación. Novena Edición. Quito-Ecuador, 2000. Pág.38.
3. LEY GENERAL DE INSTITUCIONES DEL SISTEMA FINANCIERO
4. HIBERNATE. Documentación de Referencia de Hibernate. 2005.
5. LARMAN, Craig(1999), UML y Patrones, Introducción al Análisis y diseño Orientado a Objeto
6. SANCHEZ, Jorge. Java2: incluye Swing, Threads, programación en red, JavaBeans, JDBC y JSP / Servlets. [CD-ROM]. 2005. Pág. 12.
7. SUN. Swing y JFC (Java Foundation Classes). 2006. [CD-ROM].
8. VÉLEZ SERRANO, José F. Técnicas Avanzadas de Diseño de Software [CD-ROM]. 2006.

SITIOS WEB

1. BIENVENIDO A NETBEANS Y A WWW_NETBEANS_ORG, PORTAL DEL IDE JAVA OPEN SOURCE. [en línea]. 2006. [www.netbeans.org].
2. ECKEL BRUCE. Piensa en Java. [en línea]. Tercera Edición. 2006. [http://www.BruceEckel.com]
3. CARLOS SILVA & ANDRES CERON. Introducción a la API de Excel. [en línea]. 2006. [http://profesores.elo.utfsm.cl/~agv/elo330/2s05/projects/CeronSilva/index.htm].
4. MANUAL DE JAVA. 2006. [en línea]. [http://www.manual-java.com/]
5. RJ. LORIMER. JGoodies: Understanding Binding. 2005 [en línea]. [http://www.javalobby.org/].
6. SANCHEZ, Jorge. NetBeans 5.0. Guia rápida. [CD-ROM]. 2005.

ANEXOS

ANEXO 1

ENTREVISTA AL GERENTE

1. Que requisitos debe tener un trabajador administrativo para ser socio del fondo
2. Como se hace para registrar un nuevo socio
3. Como se llama la cuenta que tiene cada socio dentro del fondo de jubilación
4. Con que monto pueden ingresar los nuevos socios
5. Cuanto aporta cada socio mensualmente y cual es porcentaje de interés por las aportaciones
6. Cuanto es el monto máximo de préstamo y el interés mensual por el préstamo
7. Que tiempo debe aportar el socio antes de acceder a un préstamo.
8. que plazos tiene el socio para pagar el préstamo
9. cada que tiempo cada socio puede retirar los fondos.
10. El socio se puede separarse libremente del fondo
11. Como se calcula el monto para la devolución del dinero en caso de retiro del socio
12. Como se realiza el pago de la jubilación a cada socio, cuantos años debe tener de aportación.
13. Como se realiza el pago de la cesantía a cada socio
14. Como se registra un préstamo a cada uno de los socios

15. como ingresa los pagos que realiza el socio en la devolución del dinero.

16. De que forma el fondo realiza el descuento a cada socio para las aportaciones

17. Que reportes se utilizan dentro del fondo

18. Cuantas personas trabajan en el fondo y que labores desempeñan cada uno de ellos

<u>CARGO</u>	<u>LABORES</u>
1.
2.
3.

ANEXO 2

ENTREVISTA PARA CONTABILIDAD

1. ¿Qué contabilidad se utiliza dentro del Fondo de Cesantía y Jubilación Administrativo de la UNL?
2. ¿Qué libros de Contabilidad Utiliza dentro del Fondo?
3. ¿Qué tipo de reportes utiliza para presentar informes?
4. ¿Cuántas contabilidades lleva en el Fondo Administrativo de la UNL?
5. ¿Cuántas personas hacen uso de la contabilidad dentro del fondo de la UNL?

ANEXO 3

ANEXO 4
ANTEPROYECTO