

003211



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

86

FACULTAD DE CIENCIAS

**APLICACIÓN DE UML AL ANÁLISIS
ORIENTADO A OBJETOS**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
A C T U A R I O
P R E S E N T A :
GILDARDO YAHVEH/ROMERO RODRÍGUEZ

DIRECTOR DE TESIS: LIC. FRANCISCO LORENZO SOLSONA CRUZ



**FACULTAD DE CIENCIAS
U.N.A.M.**

2003

**TESIS CON
FALLA DE ORIGEN**

A



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACION DISCONTINUA



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

DRA. MARÍA DE LOURDES ESTEVA PERALTA
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:
" Aplicación de UML al Análisis Orientado a Objetos "

realizado por Gildardo Yahveh Romero Rodríguez con número de cuenta 09432449-5

quién cubrió los créditos de la carrera de Actuaría.

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis Lic. Francisco Lorenzo Solsona Cruz.

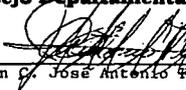
Propietario M. en C. Virginia Abrín Batule.

Propietario Mat. Adrián Girard Islas.

Suplente Act. Ricardo Humberto Sevilla Aguilar.

Suplente Act. David López Servín.

Consejo Departamental de Matemáticas

M. en C. José Antonio  FACULTAD DE CIENCIAS

CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

B

*Has sido y seguirás siendo mi gran motivación y mi fiel acompañante en cada paso de mi vida.
Indiscutiblemente no hubiera podido cerrar este ciclo de mi vida sin tu apoyo, sin tu compañía y sin tu
guía. Te doy gracias por todo lo que me has enseñado y doy gracias al padre por permitirnos continuar
juntos y poder disfrutar de este momento tan deseado por los dos.
Madre nuevamente muchas gracias y no olvides que te quiero mucho.*

*Cada uno de ustedes representa un pilar en mi vida, es la primera meta que conseguimos juntos.
Hermana: Tú sabes que has sido mi segunda mamá y que sin tu apoyo no estaría en este gran momento.
Hermanos: Gracias por el apoyo incondicional que me han brindado y les agradezco por levantarme y
hacerme fuerte en los momentos que llegué a doblegarme.*

*Eres un gran ser, es una bendición y un orgullo el tenerte como guía y amigo.
Carlos: doy gracias por toda la ayuda y la enseñanza que me has brindado.*

Amigos queridos amigos: Con ustedes me forme y aprendí lo indispensable que es contar con alguien para cualquier cosa. Fue un placer haberlos conocido y formar parte de ustedes durante nuestra formación académica. Los quiero mucho y ni crean que ya se libraron de mí.

La vida ha sido demasiado considerada conmigo; pero al permitirme contar con el apoyo de una gran mujer justo en este momento de mi vida de verdad me siento privilegiado.

Pequeña: Muchas gracias por permitirme entrar a tu vida y por ayudarme a mejorar como persona y recuerda que.... yo también Yo también.

Guillermo: Gracias por brindarme el apoyo necesario para la culminación de este trabajo.

Maestra Vicky: Gracias por confiar en los alumnos de la facultad de ciencias y mil gracias por impulsarnos a cerrar esta puerta tan importante para nosotros.

Francisco sin tu asesoría y tu amabilidad no hubiera culminado con este trabajo. Te agradezco infinitamente todas tus atenciones y el compartir conmigo tus conocimientos.

D

ÍNDICE

| | |
|---|------------|
| INTRODUCCIÓN | II |
| CAPÍTULO 1 FUNDAMENTOS DE MODELADO VISUAL CON UML | 3 |
| 1.1 Introducción a la Tecnología de Objetos y al Modelado Visual | 3 |
| 1.2 Introducción al Lenguaje de Modelado Unificado (Unified Model Language-UML) | 12 |
| 1.3 Diagramas de casos de uso | 33 |
| 1.4 Diagramas de Interacción | 35 |
| 1.5 Diagramas de Clases | 38 |
| CAPÍTULO 2 ANÁLISIS ORIENTADO A OBJETOS CON UML | 42 |
| 2.1 Introducción a la disciplina de Análisis y Diseño | 42 |
| 2.2 Análisis de Casos de Uso | 44 |
| 2.3 Análisis de dos Casos de Uso de un Sistema de Nómina | 51 |
| CAPÍTULO 3 | |
| ANÁLISIS CON UML DEL SISTEMA: | |
| REGISTRO DE REASEGURADORAS EXTRANJERAS | 70 |
| 3.1 Introducción | 70 |
| 3.2 Glosario | 71 |
| 3.3 Situación Actual | 73 |
| 3.4 Modelo de Casos de Uso | 82 |
| 3.5 Casos de Uso | 83 |
| 3.6 Análisis de Casos de Uso | 108 |
| CONCLUSIONES | 179 |
| BIBLIOGRAFÍA Y REFERENCIAS | 181 |

INTRODUCCIÓN

La Facultad de Ciencias cuenta con poca bibliografía sobre UML (Lenguaje de Modelado Unificado) y la que tiene, esta en inglés, lo que muchas veces impide a los estudiantes consultarla. Además como en todo, con leer un buen libro no es suficiente para comprender el tema en su totalidad. Generalmente se necesita una serie de ejemplos para entender de una mejor manera el tema en cuestión, pero para poder familiarizarse con el tema es necesario poner en práctica toda la teoría adquirida resolviendo problemas reales.

Es por esto que se busca que el presente trabajo sirva para introducir al lector en el amplio y relativamente nuevo tema del UML y a su vez que se utilice como un manual de referencia para aquellos interesados en realizar análisis orientado a objetos con UML.

El primer capítulo sirve de repaso sobre lo que es la tecnología de objetos (historia, conceptos básicos, etc.); sirve para conocer los tres elementos (componentes esenciales, reglas y mecanismos comunes) del UML y para comprender los diagramas más comunes en el análisis (utilizando UML) de un sistema.

En el capítulo dos se describe brevemente la disciplina de análisis y diseño. Además puede utilizarse como guía para elaborar el análisis orientado a objetos con UML.

Finalmente en el Capítulo tres se presenta la descripción de los procesos que se tienen que contemplar para realizar el Sistema de Registro de Reaseguradoras Extranjeras el cual se elaborará en la Comisión Nacional de Seguros y Fianzas. También se presenta la forma en que se plasman los requerimientos en casos de uso y el análisis con UML del sistema.

Esta última parte se utilizará como propuesta de análisis de dicho sistema con la intención de propiciar el cambio en la forma de desarrollar sistemas en dicha comisión.

CAPÍTULO 1

FUNDAMENTOS DE MODELADO VISUAL CON UML

1.1 Introducción a la Tecnología de Objetos y al Modelado Visual

La Tecnología de Objetos (OT por sus siglas en inglés) se define como un conjunto de principios (abstracción, encapsulación y polimorfismo) que junto con los lenguajes de programación orientados a objetos, bases de datos y otras herramientas que soportan estos principios, nos guían en la construcción de sistemas.

Los sistemas que se construyen utilizando la OT son mucho más adaptables a los cambios que puedan existir en los requerimientos de los usuarios, es decir, un pequeño cambio en los requerimientos no causa cambios masivos en el sistema, de tal manera que un cambio en el sistema puede ser localizado en una pequeña parte del mismo.

Estos sistemas tienen una arquitectura mucho mejor definida y permiten crear e implementar componentes reutilizables.

Esta tecnología tiene un poco más de 30 años. A continuación se presentan los acontecimientos más importantes que se han dado en su evolución:

En 1967, se diseñó Simula que fue el primer lenguaje en utilizar objetos y clases; posteriormente en 1972, Alan Key junto con otras personas crearon Smalltalk cuyas raíces fueron tomadas de Simula. En 1980 Smalltalk se convirtió en la primera versión comercial de un ambiente de programación orientado a objetos.

Bjarne Stroustrup, a finales de los años ochenta, liberó la versión del lenguaje C++, que no fue completamente un nuevo lenguaje sino la ampliación del lenguaje C.

Por último en 1991, James Gosling creó el lenguaje Oak siendo este, el lenguaje precursor de Java.

La OT se ha aplicado en diversos tipos de sistemas, por ejemplo:

- Aplicada a sistemas de tipo Cliente-Servidor y a desarrollos web, permite a las compañías encapsular las reglas del negocio en objetos, ayudándoles a distribuir procedimientos a través de Internet o de un sistema de redes¹.
- Aplicada en sistemas de tiempo real, permite que se desarrollen con una muy buena calidad, y con una muy buena adaptación a los cambios que el sistema pudiera tener. Como ejemplo de estos sistemas podemos mencionar: los sistemas inalámbricos, satelitales, sistemas de control de misiles, tráfico aéreo, sistemas de administración, asignación de ancho de banda, etc.

El diseño, en la OT, presenta diferencias respecto al diseño estructurado, es decir, en el mundo del diseño estructurado, siempre ha existido una difícil relación entre el modelo de datos (diagrama de entidad-relación) y el diagrama de flujo de datos, ya que en algunas partes coinciden, pero en otras se olvidan uno del otro por completo. En el diseño orientado a objetos se mezclan los datos y el flujo de datos desde el inicio del ciclo de vida del sistema. Es en este diseño donde se definen vistas estáticas y dinámicas del sistema; y donde se pueden encapsular datos, operaciones e incluso clases completas, con el objetivo de definir la estructura y el funcionamiento de las partes del sistema.

El diseño estructurado deposita su confianza en la estructura de datos, algoritmos sofisticados y en relaciones complejas entre los procedimientos y los datos. Y mientras el diseño estructurado promueve la reutilización de código a nivel de subrutinas individuales, el diseño orientado a objetos lo hace a nivel de clases.

El diseño orientado a objetos esta mucho más cerca que el diseño estructurado, de permitirnos desarrollar sistemas que sean extensibles, es decir, sistemas en los cuales se pueda dividir la solución que más se ajuste a un problema, en varias soluciones.

Haciendo esto, podemos asegurar que un cambio pequeño en los requerimientos del sistema no impactará con grandes modificaciones en el sistema².

¹ Las tecnologías como .Net de Microsoft o J2EE de Sun, se basan en la utilización de objetos.

Dentro de la OT se encuentran algunos conceptos básicos, por ejemplo:

Objeto.- es una entidad con una identidad única la cual encapsula su *estado y funcionamiento*. Un *objeto* tiene dos componentes clave: *Atributos y Métodos*.

Los *atributos* de un *objeto* y las relaciones que tenga el *objeto* con otro *objeto*, representan el *estado del objeto*, y sus *métodos* representan su *funcionamiento*.

El *estado* de un *objeto* es una de las posibles condiciones en la cual un objeto puede existir. Normalmente el *estado* cambia a través del tiempo. Por ejemplo el objeto Empleado tiene un *atributo* que se llama Status el cual representa si el empleado tiene contrato fijo o tiene contrato temporal; si a un empleado después de cierto tiempo le dan su contrato fijo, el *atributo* Status cambia. En este caso se dice que el *estado del objeto Empleado* cambió.

El *funcionamiento* determina como actúa el *objeto* y como reacciona a peticiones de otros *objetos*.

Cada *objeto* tiene una identidad única, aún cuando el *estado del objeto* sea idéntico al de otro *objeto*. Por ejemplo, en la vida real podemos tener dos personas que compartan las mismas características: nombre, cumpleaños, descripción del trabajo, etc., sin embargo no hay duda de que son dos individuos con identidad propia.

Clase.- es la descripción de un conjunto de objetos que comparten los mismos *atributos, operaciones, relaciones* y la misma semántica. Toda *clase* tiene tres secciones:

- *Nombre de la clase:* es como se distingue una *clase* de otra *clase*.
- *Estructura (atributos):* un *atributo* es una propiedad nombrada de la clase, la cual describe el rango de valores que un *objeto* específico de la *clase* puede tener. Una *clase* puede tener uno, varios o ningún *atributo*.
- *Comportamiento (operaciones):* una *operación* es la implementación de un servicio, la cual puede ser solicitada por cualquier *objeto* de la *clase*. Una *clase* puede tener una, varias o ninguna *operación*.

² La debilidad de las técnicas del análisis estructurado es el hecho de que hay una división entre el modelo de análisis y el modelo de diseño. Esta división causa que el sistema concebido y el desarrollado difieran a través del tiempo.

Una *clase* define un *objeto*, además define una plantilla de la estructura y el comportamiento de todos sus *objetos*. Los *objetos* creados a partir de una *clase* se dice que son *instancias de la clase*. La *clase* es la descripción estática y el *objeto* es la *instancia* en tiempo de ejecución de la *clase*.

Abstracción (primer principio básico de la orientación de objetos)- se puede definir como cualquier modelo que incluye los aspectos más importantes, esenciales o distinguidos de algo, olvidando los detalles menos importantes.

La *abstracción* permite el manejo de la complejidad, ya que se enfoca únicamente en las características esenciales de una entidad. Estas características son las que permiten distinguir a una entidad de otra. La *abstracción* define una frontera relativa a la perspectiva del observador. Concretamente no es la expresión de algo; solamente denota la idea esencial.

Ejemplos: Un estudiante es una persona que toma clases en la universidad; un profesor es una persona que da clases en la universidad; una materia es una clase que se imparte en la universidad.

Encapsulación (segundo principio básico de la orientación de objetos)- es la localización física de las características (por ejemplo *propiedades, funcionamiento*) dentro de una caja negra de abstracción, la cual oculta la implementación de las mismas, detrás de una interfaz pública.

A menudo, la *encapsulación* se refiere a la información oculta, por ejemplo, los clientes operan la interfaz sin saber como está implementada.

Ejemplo: El *objeto* Profesor Carlos tiene un *atributo* llamado MaxMaterias, el cual representa el máximo de materias que puede impartir el profesor (usualmente de tres a cuatro por semestre). Otro *objeto (objeto interfaz)* solicita al *objeto* Profesor Carlos, poner el máximo de materias con un valor de cuatro. El *atributo* MaxMaterias, es cambiado por la *operación* SetMaxMaterias () .

Con el ejemplo anterior podemos constatar el beneficio de la *encapsulación*, ya que el *objeto* solicitante no necesita saber como cambiar el valor de máximo de materias.

Además, el *objeto* solicitante no se ve afectado si el número o variables que se usan para definir el máximo de materias se incrementan.

La OT asegura que toda comunicación con el *objeto* sea a través de *operaciones* predefinidas.

A los datos internos del *objeto* sólo se puede tener acceso mediante las *operaciones* del *objeto*. Otros *objetos* no pueden entrar al *objeto* y cambiar los valores de sus *atributos*.

Modularidad (tercer principio básico de la orientación de objetos)- se define como la descomposición física y lógica de cosas complejas en pequeños conjuntos de piezas más manipulables.

Ejemplo: Supongamos que se va a desarrollar un Sistema de Registro de Materias. El sistema es muy grande y abstracto para poder comprenderlo a detalle. Por lo tanto el equipo de desarrollo decide partir el sistema en tres sistemas independientes: Sistema de Facturación, Sistema de Catálogo de Materias y Sistema de Control de Estudiantes.

Jerarquía (cuarto principio básico de la orientación de objetos)- es cualquier clasificación de abstracciones en una estructura de árbol.

La *Jerarquía* es una clasificación taxonómica y su uso nos facilita reconocer similitudes y diferencias. Por ejemplo, la botánica clasifica las plantas en familias y la química clasifica los elementos en una tabla periódica.

Polimorfismo- Nos permite manejar un mismo mensaje de diferentes maneras, dependiendo del objeto que lo reciba. Por ejemplo: se tiene un *objeto interfaz* que quiere conocer el valor presente de 3 instrumentos financieros (acciones, bonos y sociedades), sin embargo, el valor presente se calcula de diferente manera para cada instrumento. En un ambiente de desarrollo que no sea orientado a objetos, se tendría que escribir código de la siguiente forma:

```
If InstrumentoFinanciero = Accion then
    CalculaValorAccion()
Else
Begin
    If InstrumentoFinanciero = Bono then
        CalculaValorBono()
    Else
        CalculaValorSociedad();
End;
```

En la OT, cada instrumento financiero puede ser representado por una *clase*, y cada *clase* deseará saber cómo calcular su propio valor presente. El *objeto interfaz* simplemente necesita solicitarle al objeto específico “dame el valor presente” sin necesidad de mantener la información de cada una de las tres operaciones.

Generalización.- se puede definir como una relación de *especialización/generalización* en la cual los *objetos* del elemento *especializado* (el *hijo* o *subclase*), son sustituibles por *objetos* del elemento *generalizado* (el *padre* o *superclase*).

Herencia.- se puede definir como el mecanismo mediante el cual elementos más específicos, incorporan la estructura y el comportamiento de elementos más generales.

Los términos *generalización* y *herencia* se puede decir que significan lo mismo; si se quiere diferenciarlos, la *generalización* es el nombre de la relación entre las clases, y la *herencia* es el mecanismo que es representado por la relación de *generalización*. En la Figura 1.1-1 se ejemplifican los casos de *Generalización* y de *Herencia*.

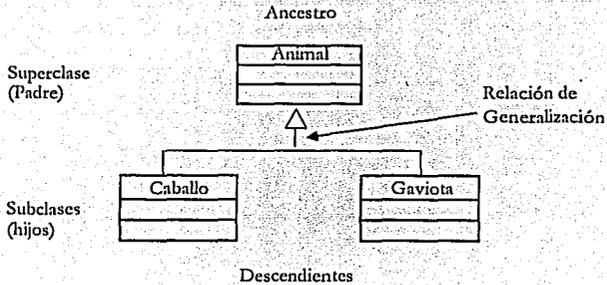


Figura 1.1-1

En la Figura 1-1 se dice que la clase *Caballo* (subclase) hereda a la clase *Animal* (superclase), ya que la clase *Caballo* hereda propiedades (extremidades, pelo) de la clase *Animal* y además cuenta con sus propias propiedades (linaje, con o sin herradura). Por la misma razón se dice que la clase *Gaviota* (subclase), hereda a la clase *Animal* (superclase).

Antes de hablar del modelado visual es necesario definir que es un *modelo*. Un *modelo* es una simplificación de la realidad, que nos proporciona los bosquejos de un sistema. Los *modelos* pueden incluir tanto planos detallados, como planos más generales, que proporcionan diferentes perspectivas del sistema que se quiere construir. Un buen *modelo*, incluye aquellos elementos que no son distinguidos en un determinado nivel de abstracción. Un sistema puede ser descrito de diferentes formas, utilizando diferentes tipos de *modelos*; y cada *modelo* es en consecuencia, una abstracción —semánticamente— cercana del sistema. Un *modelo* puede clasificarse como estructural cuando se enfoca en la estructura del sistema; o de funcionamiento, cuando se enfoca en la dinámica del mismo.

Al modelar se consiguen cuatro objetivos:

1. Los *modelos* ayudan a visualizar el sistema como se quiere que este sea. Ayudan al equipo de desarrollo a tener una misma visión del sistema a desarrollar. También brindan un mejor entendimiento del sistema.

2. Los *modelos* permiten especificar la estructura y el funcionamiento del sistema, es decir, permiten documentar la estructura y funcionamiento del sistema antes de la codificación.
3. Los desarrolladores utilizan los *modelos* como mapas, para construir el sistema.
4. Los *modelos* documentan las decisiones tomadas. Se puede decir que son una herramienta de larga duración ya que dan información que ayuda a tomar decisiones difíciles en la fase del diseño.

La experiencia del uso de modelos en todas las disciplinas de la ingeniería, ya tiene su historia. Esta experiencia sugiere cuatro principios de modelado:

1. *La elección de cuáles modelos se van a crear, tiene una profunda influencia en el cómo se va a atacar el problema y en cómo se va a conformar la solución.*

Los *modelos* que se escogen, en el desarrollo de *software*, pueden afectar enormemente la percepción de la realidad. Si se construye un sistema con la visión de tener datos relacionados y estructurados, es muy probable que se usen *modelos* de entidad-relación plasmando el funcionamiento en *store procedures* (procedimientos almacenados) y *triggers* (disparadores automáticos). Por otro lado, si se construye un sistema con una visión orientada a objetos, se terminará con un sistema cuya arquitectura se centra en varias *clases* y en *patrones de interacción*.

2. *Todo modelo puede ser expresado en distintos niveles de precisión.*

Si se está construyendo un edificio, se necesitan diferentes perspectivas, por ejemplo se necesita la opinión de los inversionistas sobre como va quedando el edificio. Otras veces, se necesita bajar al nivel de las estructuras, por ejemplo, cuando se tiene una difícil instalación de las tuberías ó de alguna estructura poco usual.

Sucede lo mismo con los *modelos* de sistemas. Cuando se está construyendo un sistema en el cual la interfaz es visual (Graphical User Interface systems), a veces un rápido y simple *modelo* ejecutable de la interfaz del usuario es lo que se

necesita; otras veces, es necesario bajar de nivel, por ejemplo, cuando se especifican las interfases del sistema o cuando se esta lidiando con los cuellos de botella de un sistema de redes. En cualquier caso, los mejores *modelos* son los que permiten escoger el nivel de detalle, dependiendo de quien y para que necesita observarlos.

3. *Los mejores modelos están conectados a la realidad.*

Todos los modelos simplifican la realidad. El truco es, estar seguros de que las simplificaciones realizadas no oculten detalles importantes del mundo real.

4. *Un solo modelo no es suficiente.*

Todo sistema complicado es mejor abordarlo a través de un conjunto pequeño de modelos que se puedan hacer y estudiar por separado, pero que estén relacionados.

TEMA CON
FALLA DE ORIGEN

1.2 Introducción al Lenguaje de Modelado Unificado (Unified Model Language-UML)

A mediados de los años setenta y finales de los ochenta, fue cuando los lenguajes de modelado orientados a objetos aparecieron como metodologías, incrementándose de menos de 10 a un poco más de 50 durante 1989-1994. Desafortunadamente muchos usuarios de estos métodos tuvieron problemas en encontrar completa satisfacción en cualquiera de estos lenguajes de modelado. A mediados de 1990, surgieron nuevos métodos, siendo los más notables los de Grady Booch, Ivar Jacobson (Ingeniería de Software Orientada a Objetos-OOSE), y el de James Rumbaugh (Técnica de Modelado de Objetos-OMT).

En octubre de 1994, se une Jim Rumbaugh a Grady Booch en la empresa Rational Software, para comenzar oficialmente el desarrollo del UML. La versión preliminar 0.8 del Método Unificado -así fue llamado-fue liberada en octubre de 1995. Siendo por esas fechas cuando Jacobson se integra a Rational, expandiendo el alcance del proyecto del UML al incorporar la OOSE, plasmando los resultados en la liberación de la versión 0.9. Durante 1996 fue obvio que varias organizaciones vieron al UML como una estrategia de negocios. Booch, Rumbaugh y Jacobson establecieron un consorcio con distintas corporaciones, con el objetivo de tener recursos para enfocarse a una firme y completa definición del UML.

Los socios que contribuyeron a definir la versión 1.0 del UML como un lenguaje de modelado bien definido, expresivo, poderoso y aplicable a un gran número de problemas de diferentes tipos de organizaciones fueron: Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Texas Instruments y Unisys.

El UML 1.0 fue propuesto como el lenguaje de modelado estándar por el Grupo de Administración de Objetos (OMG) en enero de 1997 sin tener éxito. En julio del mismo año, se revisa la versión 1.1, aceptándose como el lenguaje estándar de modelado³ hasta septiembre. Dicha versión fue adoptada por la OMG el 14 de noviembre.

³ Un lenguaje proporciona un vocabulario y las reglas para combinar las palabras de ese vocabulario y así poder comunicar algo. Un lenguaje de modelado es aquel que enfoca su vocabulario y sus reglas, a la representación conceptual y física de un sistema.

El mantenimiento del UML quedó a cargo del Grupo de Trabajo de Revisión (Revision Task Force- RTF) de la OMG. En junio de 1998 la RTF liberó la versión 1.2; un año más tarde se encargó de liberar la última versión que se conoce hasta el momento, la 1.3.

El vocabulario y las reglas del UML, indican cómo crear y leer buenos modelos, pero no cuáles modelos se deben crear, ni cuándo se deben crear. Ese es el papel del proceso de desarrollo de *software*. Un proceso bien definido ayuda a decidir que artefactos⁴ crear, que actividades y que trabajadores se deben utilizar para crear y manejar dichos artefactos. También ayuda a saber como usar esos artefactos para medir y controlar el proyecto en su totalidad.

El UML es un lenguaje que sirve para:

1. Visualizar

Típicamente, las compañías de *software* desarrollan su propio lenguaje para modelar sistemas. Esto trae como consecuencia, que para alguna persona que no tienen nada que ver con el sistema o para algún integrante nuevo del equipo de desarrollo, les sea más difícil de entender lo que está pasando con el sistema.

La comunicación con otras personas, mediante estos lenguajes propios, suele ser mala, al menos que todos los involucrados en el sistema hablen el mismo lenguaje. El UML brinda un conjunto de símbolos que representan expresiones bien definidas. De esta manera, un desarrollador puede construir un modelo en UML, y otro desarrollador puede interpretarlo con claridad la mayoría de las veces.

En un sistema, existen ciertas cuestiones que no se pueden entender a menos que se construyan *modelos* que trasciendan textualmente al lenguaje de programación. Por ejemplo, el significado de *jerarquía de clase* puede deducirse, pero no comprenderse de forma directa, con sólo mirar el código. El UML es un lenguaje gráfico que se encarga de este tipo de problemas. Construir *modelos* explícitos en el UML facilita la comunicación.

⁴ Un artefacto es una pieza de información que es usada o producida por un proceso de desarrollo de *software*.

2. Especificar

En este contexto *especificar* significa construir modelos que sean precisos, claros y completos. En particular, el UML se encarga de especificar todas las decisiones importantes que se deben tomar en el análisis, diseño e implementación, al desarrollar y utilizar un sistema.

3. Construir

El UML no es un lenguaje de programación visual. Sin embargo, sus modelos pueden comunicarse directamente con una variedad de lenguajes de programación, haciendo esto posible el hacer un mapeo de un modelo en UML a un lenguaje de programación como Java, C++ ó Visual Basic, incluso hacer un mapeo a tablas de una base de datos relacional o a el almacenamiento persistente de una base de datos orientada a objetos.

Este tipo de mapeos, permiten la ingeniería hacia adelante—también conocida como re-ingeniería por algunos autores— y la ingeniería en reversa.

La ingeniería hacia adelante se refiere a generar código en un lenguaje de programación, a partir de un modelo en UML. La ingeniería en reversa se refiere a la reconstrucción de un *modelo* a partir de la implementación; es decir, un cambio en una parte del código, al utilizar la ingeniería en reversa, se ve reflejado en el *modelo* que generó esa parte del código.

4. Documentar

El UML se encarga de documentar todos los detalles de la arquitectura de un sistema. También documenta los requerimientos del sistema y sus pruebas. Finalmente, el UML es un lenguaje que sirve para modelar las actividades de la planeación del proyecto y de la administración de versiones.

TESIS CON
FALLA DE ORIGEN

En primera instancia el UML se propuso para modelar sistemas de *software*, sin embargo, es lo suficientemente rico como para modelar sistemas que no sean de *software*, por ejemplo, la distribución de trabajo en el ordenamiento jurídico; la estructura y comportamiento de un enfermo en un sistema de atención médica o en el diseño de *hardware*. También se ha utilizado de forma efectiva en otras áreas tales como: Sistemas de información, Servicios financieros y actividades bancarias, Telecomunicaciones y transportes, Aeronáutica, Ciencias, Servicios distribuidos en Internet, entre otras.

Para entender el UML, se necesita crear un *modelo* conceptual del lenguaje y para poder hacer esto, se requiere conocer tres elementos principales: Los *Componentes Esenciales* del UML, las *Reglas* que establecen como se organizan los *Componentes Esenciales* y algunos *Mecanismos Comunes* que se aplican en el UML. Una vez comprendidas estas ideas, se estará listo para escribir y crear *modelos* básicos en UML.

Dentro de los *componentes esenciales* encontramos: *los Elementos de un Modelo*, *las Relaciones* y *los Diagramas*.

Los *elementos de un modelo* son las abstracciones principales del *modelo*; *las relaciones* se encargan de hacer la conexión entre los *elementos del modelo*; y los *diagramas* se encargan de agrupar colecciones interesantes de *elementos del modelo*.

Existen cuatro tipos de *elementos de un modelo* en UML: *Elementos Estructurales*, *Elementos de Comportamiento*, *Elementos de Agrupación* y *Elementos de Anotación*. Estos *elementos* son los *componentes esenciales* orientados a objetos del UML.

Los *elementos estructurales*, son los sustantivos de los *modelos*. Son las partes más estáticas de un *modelo* en UML, representan elementos conceptuales ó físicos. En total existen siete tipos de *elementos estructurales*:

1. *Clase*.- es la descripción de un conjunto de *objetos* que comparten los mismos *atributos*, *operaciones*, *relaciones* y la misma semántica. Una *clase* implementa una o más interfaces. Gráficamente, una *clase* se representa como un rectángulo que generalmente incluye su *nombre*, sus *atributos* y sus *operaciones*. Figura 1.2-1.

| |
|-----------|
| Ventana |
| tamaño |
| abrir () |
| cerrar () |

Figura 1.2-1

2. *Interfaz*- es una colección de *operaciones* que especifica un servicio de una *clase* ó de un *componente*. Por consiguiente, una *interfaz* describe el comportamiento externamente visible de esa *clase* o de ese *componente* y puede representar su comportamiento total o parcial. Una *interfaz* define un conjunto de especificación de *operaciones* y no un conjunto de implementación de *operaciones*. Gráficamente, una *interfaz* se representa como un círculo junto con su nombre. Es raro que una *interfaz* se encuentre sola, típicamente se asocia a la *clase* o *componente* que realiza la *interfaz*.



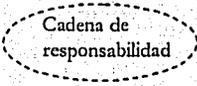
ISensor

Figura 1.2-2

3. *Colaboración*- es una sociedad de *clases*, *interfases* y de otros elementos que trabajan juntos para proporcionar un cierto comportamiento cooperativo que sea más grande que la suma de los comportamientos de cada una de sus partes; es la especificación de cómo un elemento—por ejemplo un *caso de uso* o una *operación*—es realizado por un conjunto de *clasificadores*³ y *asociaciones*⁴ jugando un papel determinado y utilizado de una forma en específico. Gráficamente, una *colaboración* se representa como una elipse con líneas entrecortadas y generalmente solo incluye su nombre. Figura 1.2-3.

³ Un clasificador es un mecanismo que describe características estructurales y de comportamiento; incluyen clases, interfaces, tipos de datos, señales, componentes, nodos, casos de uso y subsistemas.

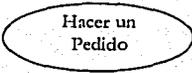
⁴ Una asociación es una relación estructural que describe un conjunto de ligas, en la cual una liga es una conexión entre objetos; es la relación semántica entre dos o más clasificadores que implica las conexiones entre sus instancias.



Cadena de
responsabilidad

Figura 1.2-3

4. *Caso de Uso*.- es la descripción de un conjunto de secuencias de acciones que un sistema realiza, y que produce un resultado observable de valor para un actor en particular. Un *caso de uso* se utiliza para estructurar los *elementos de comportamiento* de un *modelo* y un *caso de uso* es "realizado" por una *colaboración*. Gráficamente, un caso de uso se representa como una elipse de línea sólida y generalmente solo incluye su nombre. Figura 1.2-4.



Hacer un
Pedido

Figura 1.2-4

5. *Clase Activa*.- es lo mismo que una *clase* excepto que sus *objetos* representan elementos cuyo comportamiento es simultáneo con otros elementos. Gráficamente, una *clase activa* se representa como una *clase* pero con rectángulo remarcado y generalmente incluye su *nombre*, sus *atributos* y sus *operaciones*. Figura 1.2-5.

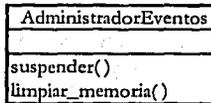


Figura 1.2-5

TESIS CON
FALLA DE ORIGEN

6. *Componente*.- es una parte física y reemplazable de un sistema que se conforma y proporciona la *realización*⁷ de un conjunto de *interfaces*. Gráficamente, un *componente* se representa con un rectángulo con lengüetas y generalmente solo incluye su nombre. Figura 1.2-6.

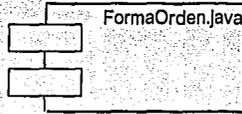


Figura 1.2-6

7. *Nodo*.- es un elemento que existe en tiempo de ejecución y representa un recurso computacional, en general tiene al menos alguna memoria y a menudo, capacidad de procesamiento. Un conjunto de *componentes* puede residir en un *nodo* y también puede emigrar de un *nodo* a otro. Gráficamente, un *nodo* se representa con un cubo y generalmente solo incluye su nombre. Figura 1.2-7.



Figura 1.2-7

Estos siete elementos son los *elementos estructurales* básicos que se pueden incluir en un modelo de UML. Hay variaciones para estos siete elementos, por ejemplo: *actores, señales y utilidades* (tipos de *clases*), *procesos y secuencias* (tipo de *clases activas*), aplicaciones, documentos, archivos, bibliotecas, páginas y tablas (tipo de *componentes*).

Los *elementos de comportamiento*, son los verbos de los *modelos*. Son las partes dinámicas de un *modelo* en UML, representan el comportamiento a través del tiempo y del espacio. En total existen dos tipos primarios de *elementos de comportamiento*:

⁷ Una realización es una relación semántica entre clasificadores, en dónde un clasificador especifica un contrato y otro clasificador garantiza llevarlo a cabo.

1. *Interacción*.- es un comportamiento que comprende un conjunto de *mensajes* que son intercambiados por un conjunto de *objetos* dentro de un contexto en particular para cumplir un propósito en específico. El comportamiento de una sociedad de objetos o de una operación individual, puede ser especificada con una *interacción*. Una *interacción* comprende cierto número de elementos, incluyendo *mensajes*, *secuencias de acción* (comportamiento invocado por un *mensaje*), y *ligas* (conexiones entre *objetos*). Gráficamente, un *mensaje* se representa con una línea dirigida y casi siempre incluye el *nombre* de su *operación*. Figura 1.2-8.

desplegar →

Figura 1.2-8

2. *Máquina de estados*.- es un comportamiento que especifica la secuencia de *estados* por los que atraviesa un *objeto* o una *interacción* durante su tiempo de vida en respuesta a eventos. El comportamiento de una *clase* o de una *colaboración de clases* puede especificarse con una *máquina de estados*. Una *máquina de estados* comprende cierto número de elementos, incluyendo *estados*, *transiciones* (flujo de un *estado* a otro), *eventos* (elementos que ejecutan *transiciones*). Gráficamente, un *estado* se representa con un rectángulo con las esquinas redondas, generalmente incluye su *nombre* y a veces *subestados*. Figura 1.2-9.

Esperando

Figura 1.2-9

Estos dos elementos son los *elementos de comportamiento* básicos que se pueden incluir en un modelo de UML. Semánticamente, estos elementos se relacionan con varios *elementos estructurales*, sobre todo con *clases*, *colaboraciones* y *objetos*.

Los *elementos de agrupación*, son las partes organizadoras de un modelo en UML. Son “cajas” en las cuales puede descomponerse un *modelo*. Existe un solo tipo de *elementos de agrupación*:

Paquete.- es un mecanismo que se emplea con el propósito de organizar elementos en grupos. Los *elementos estructurales*, *de comportamiento* e incluso *paquetes*, pueden colocarse en

un *paquete*. A diferencia de los componentes (que existen en tiempo de ejecución), un paquete es únicamente conceptual, es decir, existe sólo en la etapa de desarrollo. Gráficamente, un *paquete* se representa con un rectángulo con lengüeta, generalmente incluye su nombre y, algunas veces, su contenido. Figura 1.2-10.

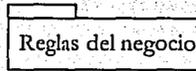


Figura 1.2-10

Los *paquetes* son los *elementos de agrupación* básicos, con los que se puede organizar un *modelo* en UML. Existen variaciones de los *paquetes*, como *marcos de trabajo*, *modelos*, *subsistemas*.

Los *elementos de anotación* son los elementos explicatorios de un modelo en UML. Son los comentarios que se pueden usar para describir, informar o remarcar algún elemento en un *modelo*. Existe un tipo de *elementos de anotación*:

Nota- es un símbolo que sirve para interpretar restricciones y para adjuntar comentarios a un elemento o a una colección de elementos. Gráficamente, una *nota* se representa con un rectángulo con un doblado en una de sus esquinas, junto con un comentario textual o gráfico. Figura 1.2-11.

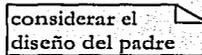


Figura 1.2-11

Las *notas* son los *elementos de anotación* básicos, que se pueden incluir en un *modelo* en UML. Existen variaciones de las *notas*, como los *requerimientos* (con los cuales se especifica algún comportamiento deseado, desde una perspectiva externa del *modelo*).

Dentro de un *modelo* en UML, existen cuatro tipos de *relaciones*: *Dependencia*, *Asociación*, *Generalización* y *Realización*. Estas *relaciones* son los *componentes esenciales* relacionales del UML.

1. Una *dependencia* es una relación de uso entre dos elementos, en la cual un cambio en la especificación de un elemento (elemento independiente) puede afectar a otro elemento (elemento dependiente). Gráficamente, una *dependencia* se representa con una línea entrecortada, posiblemente dirigida, y en ocasiones incluye una etiqueta. Figura 1.2-12.



Figura 1.2-12

2. Una *asociación* es una relación estructural que describe un conjunto de ligas, en la cual una liga es una conexión entre objetos. Gráficamente, una *asociación* se representa con una línea sólida, posiblemente dirigida; en ocasiones incluye una etiqueta, y regularmente contiene otros *adornos*⁸, como son *multiplicidad*⁹ y nombres de los *roles*¹⁰. Figura 1.2-13.

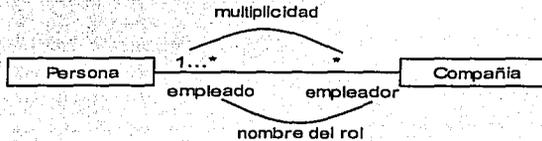


Figura 1.2-13

Una *agregación* es un tipo especial de *asociación*, la cual representa una relación estructural entre un todo y las partes de ese todo. Gráficamente, una *agregación* se representa como una *asociación* pero con un diamante “abierto” del lado del todo. Figura 1.2-14.

⁸ Un *adorno* se define como la especificación de un elemento, agregada a su notación gráfica básica.

⁹ La *multiplicidad* es la especificación del valor o del rango de valores que un conjunto puede asumir.

¹⁰ Un *rol* es el comportamiento de una entidad que participa en un contexto en particular.

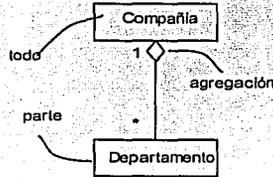


Figura 1.2-14

3. Una *generalización* es una relación de *especialización/generalización* en la cual los *objetos* del elemento *especializado* (el *hijo* o *subclase*), son sustituibles por *objetos* del elemento *generalizado* (el *padre* o *superclase*). De este modo, el *hijo* comparte la estructura y el comportamiento del *padre*. Gráficamente, una relación de *generalización* se representa con una línea sólida con una punta de flecha vacía, apuntando al *padre*. Figura 1.2-15.

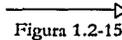


Figura 1.2-15

4. Una *realización* es una relación semántica entre *clasificadores*, en dónde un *clasificador* especifica un contrato y otro *clasificador* garantiza llevarlo a cabo. Se pueden encontrar relaciones de *realizaciones* en dos lugares: entre interfaces y las clases o componentes que las realizan, y entre *casos de uso* y las *colaboraciones* que los realizan. Gráficamente, una relación de *realización* se representa con una línea entrecortada con una punta de flecha vacía. Figura 1.2-16.



Figura 1.2-16

Estas cuatro *relaciones* son los elementos relacionales básicos que se pueden incluir en un *modelo* de UML. Existen variaciones (*estereotipos*) de estas cuatro *relaciones*, por ejemplo: *refinación*, *trazas*, *inclusión*, *extensión*, *completo* e *incompleto*.

Los *diagramas* completan los *componentes esenciales* del UML. Son la presentación gráfica de un conjunto de elementos, en la mayoría de los casos representados como gráficas conectadas de vértices (*cosas* o *elementos*) y arcos (*relaciones*). Se pueden dibujar *diagramas* para visualizar un

sistema desde diferentes perspectivas, de tal modo que un *diagrama* es una proyección dentro de un sistema. Existen nueve tipos de *Diagramas* en el UML:

1. Un *diagrama de clases* muestra un conjunto de *clases*, *interfases* y *colaboraciones* con sus *relaciones*. Estos *diagramas* son los que comúnmente se encuentran al modelar un sistema orientado a objetos; se encargan de la vista estática de diseño de un sistema.
2. Un *diagrama de objetos* muestra un conjunto de objetos y sus relaciones. Representa fotos estáticas de las *instancias* de los elementos que se encuentran en los *diagramas de clases*. Estos diagramas se encargan de la *vista estática del diseño* o *vista estática de procesos* de un sistema—como los *diagramas de clases*—, sólo que desde una perspectiva de un caso real o de un prototipo.
3. Un *diagrama de casos de uso* muestra un conjunto de *casos de uso*, *actores* y de sus *relaciones*. Se encargan de la *vista estática de casos de uso* de un sistema. Estos *diagramas* son muy importantes en la organización y modelado de los comportamientos de un sistema.

Un *diagrama de interacción* muestra un conjunto de *objetos* y sus *relaciones*, incluyendo los *mensajes*¹¹ que pueden ser enviados entre ellos. Estos *diagramas* se encargan de la *vista dinámica* de un sistema. Existen dos tipos de *diagramas de interacción*:

4. Los *diagramas de secuencia*, son los que resaltan el tiempo de ordenamiento de los *mensajes*,
5. los *diagramas de colaboración*, son los que resaltan la organización estructural de los *objetos* que envían y reciben *mensajes*.

Los dos tipos de *diagramas* son isomorfos, es decir, se puede tomar cualquiera de los dos tipos y transformarlo al otro tipo.

¹¹ Un mensaje es la especificación de una comunicación entre objetos que transmiten información con la espera de que ocurra una actividad.

6. Un *diagrama de estado* muestra una *máquina de estado*, junto con sus *estados*, transiciones, *eventos* y *actividades*. Estos *diagramas* se encargan de la *vista dinámica* de un sistema. Son de importancia en el modelado del comportamiento de una *interfaz*, *clase* o *colaboración*; resaltan el comportamiento de *eventos* ordenados de un *objeto*.
7. Un *diagrama de actividad* es un tipo especial de un *diagrama de estado*, que muestra el flujo de una actividad a otra dentro del sistema. Estos *diagramas* se encargan de la *vista dinámica* de un sistema. Son de gran utilidad al modelar la función de un sistema y al querer resaltar el flujo de control entre objetos.
8. Un *diagrama de componentes* muestra la *organización* y las *dependencias* entre un conjunto de *componentes*. Estos *diagramas* se encargan de la *vista estática de implementación* de un sistema.
9. Un *diagrama de instalación* muestra la configuración de los *nodos* y de los *componentes* que viven en ellos. Los *diagramas de instalación* se encargan de la *vista estática de instalación* de una *arquitectura*.

La lista anterior, no es una lista cerrada de *diagramas*, sin embargo estos nueve *diagramas* son los más comunes en la práctica del UML.

El UML como cualquier otro lenguaje, tiene ciertas *reglas* cuya función es especificar como puede construirse un *modelo bien hecho*¹². Existen *reglas* para:

- *Nombres* indica de que modo se puede llamar a los *elementos*, *relaciones* y a los *diagramas*
- *Alcance* indica el contexto que da un significado específico a los *nombres*
- *Visibilidad* indica cómo los *nombres* pueden ser utilizados y vistos por los demás
- *Integridad* indica cómo los *elementos* se pueden relacionar propia y consistentemente con otros *elementos*
- *Ejecución* indica de que modo se puede simular un *modelo* dinámico

¹² Se entiende como modelo bien hecho, aquel modelo que es semánticamente congruente consigo mismo y que concuerda con todos los modelos con los que se encuentra relacionado.

Los modelos construidos durante el desarrollo de sistemas, tienden a evolucionar; además pueden ser vistos desde diferentes perspectivas, por varios grupos de individuos, en diferente tiempo. Por estas razones es común que el equipo de desarrollo, no sólo construya *modelos bien hechos* sino también modelos que sean:

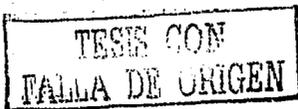
- *Abreviados* ciertos *elementos* son ocultos, con el fin de simplificar la vista del *modelo*
- *Incompletos* ciertos *elementos* pueden ser omitidos
- *Inconsistentes* la integridad de un *modelo* no esta garantizada

Las *reglas* del UML propician más no obligan, a tomar en cuenta cualquier tipo de pregunta que surja en el *análisis, diseño e implementación* de un sistema y al considerar estas preguntas durante la construcción del sistema, se impulsa a los tres *modelos* arriba mencionados, a que lleguen a ser *modelos bien hechos*.

Una construcción se hace más simple y más armoniosa al conformar patrones de características comunes. Por ejemplo una casa, puede construirse con un estilo Victoriano o con un estilo francés, utilizando ciertos patrones arquitectónicos que definen dichos estilos. Lo mismo sucede en el UML, es decir, se vuelve más simple con la presencia de cuatro *mecanismos comunes* que se aplican consistentemente a lo largo de todo el lenguaje.

Las *especificaciones* son el primer *mecanismo* del UML, se encuentran detrás de cada una de las notaciones gráficas, y proporcionan comentarios textuales y semánticos de los *componentes esenciales*. Por ejemplo, detrás de una *clase* hay una *especificación* que provee el conjunto completo de *atributos, operaciones y comportamientos* que la *clase* contiene; visualmente una *clase* puede mostrar solo una pequeña parte de su *especificación*.

El segundo mecanismo son los *adornos*, los cuales pueden ser *adornos* gráficos o textuales y sirven para representar algunos otros detalles de la *especificación* de algún *componente esencial*. En la Figura 1.2-17 se muestra una *clase* con *adornos* que indican que es una *clase* abstracta con dos operaciones públicas, una protegida y una privada.



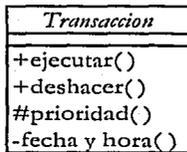


Figura 1.2-17

Las *divisiones comunes* son el tercer mecanismo del UML. En el modelado de sistemas orientados a objetos, el mundo se divide al menos en dos formas:

Primero, existe la división en *clase* y *objeto*. Una *clase* es una *abstracción*; un *objeto* es la manifestación concreta de esa *abstracción*. Casi todos los *componentes esenciales* del UML, tienen este mismo tipo de (*clase/objeto*) dicotomía. Por ejemplo, se tienen *casos de uso* y sus *instancias*, *componentes* y sus *instancias*, *nodos* y sus *instancias*, y así sucesivamente. Gráficamente, en el UML, un *objeto* se representa de la misma forma que una *clase* solo que el nombre del objeto se subraya. En la Figura 1.2-18 se presenta una *clase* llamada Cliente junto con tres *objetos*: Juan (el cual es marcado explícitamente como un *objeto* de tipo Cliente), :Cliente (un objeto anónimo de tipo Cliente) y el *objeto* Elisa (el cual es marcado en esta *especificación*, como un *objeto* de tipo Cliente, sin embargo aquí no se muestra explícitamente).

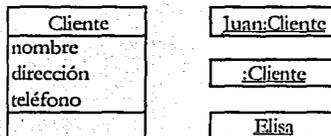


Figura 1.2-18

Segundo, existe una separación entre *interfaz* e *implementación*. Una *interfaz* declara un contrato, y la *implementación* representa la *realización* concreta. Casi todos los *componentes esenciales* del UML, tienen este tipo de (*interfaz/implementación*) dicotomía. Por ejemplo, se tienen *casos de uso* y las *colaboraciones* que los realizan, *operaciones* y los *métodos* que las implementan. En la Figura 1.2-19 se representa un *componente* llamado

TESIS CON
FALLA DE ORIGEN

ayudaortografia.dll y dos *interfases*, una llamada IDesconocida y la otra IOrtografia. En este caso se dice que el *componente* implementa las *interfases*.

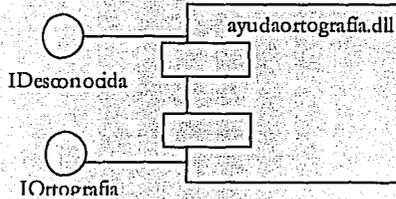


Figura 1.2-19

Los *mecanismos de extensibilidad* conforman el cuarto *mecanismo* del UML. Estos mecanismos incluyen: *estereotipos*, *valores etiquetados* y *restricciones*. Un *estereotipo*, extiende el vocabulario del UML, permitiendo la definición de nuevos tipos de *componentes esenciales*, en base a los *componentes* existentes. Estos nuevos *componentes esenciales* pueden contener significados adicionales, pero aún así aplican en todos los casos donde el *componente* original es usado. El nombre de un *estereotipo* se encierra entre el símbolo << >>. Los *valores etiquetados*, extienden las propiedades de los *componentes esenciales* del UML, permitiendo la definición de nueva información en la *especificación* de esos *componentes*, por último las *restricciones*, extienden el significado de los *componentes esenciales*, permitiendo la definición de nuevas *reglas* o la modificación de las existentes. Conjuntamente, estos tres *mecanismos de extensibilidad* permiten moldear y ampliar el UML para las necesidades de un proyecto en particular. En la Figura 1.2-20 se muestra la utilización de *estereotipos*, *valores etiquetados* y de *restricciones*.

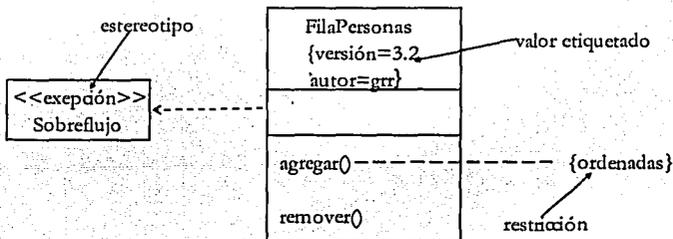


Figura 1.2-20

Con estos *mecanismos* se termina con la descripción de los tres elementos principales que se deben conocer para entender el UML.

Recordando el hecho de que los *modelos* construidos durante el desarrollo de sistemas, tienden a evolucionar y además pueden ser vistos desde diferentes perspectivas, por varios grupos de individuos, en diferente tiempo; se puede llegar a decir que un sistema puede ser visto desde diferentes perspectivas, por varios grupos de individuos, en diferente tiempo. La *arquitectura* de un sistema quizá, es el artefacto más importante que se puede utilizar para administrar estas perspectivas durante el ciclo de vida del sistema. La *arquitectura* se puede definir como el conjunto de decisiones significativas sobre la organización de un sistema de *software*; las decisiones sobre el estilo arquitectónico que guía a dicha organización, es decir los *elementos* estáticos y dinámicos junto con sus *interfaces*, sus *colaboraciones* y su composición.

La *arquitectura de software*, no solo tiene que ver con la estructura y el comportamiento de un sistema, sino también con la utilización, funcionalidad, rendimiento, adaptación al cambio, reutilización, comprensibilidad, limitaciones económicas, tecnológicas y la interrelación que hay entre estas, y las cuestiones estéticas.

Para describir de una mejor manera la arquitectura de los sistemas de *software*, se pueden utilizar cinco *vistas* interdependientes y complementarias. Cada *vista* es una proyección dentro de la organización y estructura de un sistema, enfocada en un aspecto en particular.

En la Figura 1.2-21 se muestran las cinco *vistas*.



Figura 1.2-21

- La *vista de casos de usos* incluye los *casos de uso* los cuales describen el comportamiento de un sistema, desde el punto de vista de los usuarios finales, analistas y probadores o control de calidad. Esta *vista* realmente no especifica la organización

de un sistema de *software*; lo que especifica, son los intereses que forman la *arquitectura* del sistema. En el UML, los aspectos estáticos de esta vista, son capturados en los *diagramas de casos de uso*; y los dinámicos, en los *diagramas de interacción, de estado y de actividades*.

- La *vista de diseño* de un sistema, abarca las *clases, interfases, y colaboraciones* que forman el vocabulario del problema y su solución. Esta *vista* principalmente mantiene los requerimientos funcionales del sistema, es decir, los servicios que el sistema debe proveer a sus usuarios finales. En el UML, los aspectos estáticos de esta vista, son capturados en los *diagramas de clase y de objetos*; y los dinámicos, en los *diagramas de interacción, de estado y de actividades*.
- La *vista de proceso* de un sistema, se encarga de las *secuencias y procesos* que forman los *mecanismos* de concurrencia y de sincronización del sistema. Esta *vista* se encarga principalmente del rendimiento y de la escalabilidad del sistema. En el UML, los aspectos estáticos y dinámicos son capturados en el mismo tipo de *diagramas* que en la *vista de diseño*, pero con un enfoque especial en las *clases activas*.
- La *vista de implementación* de un sistema, incluye los *componentes* y archivos que se utilizan para ensamblar y liberar la parte física del sistema. Esta vista se encarga de la administración de la configuración de las liberaciones del sistema, integradas por componentes poco independientes y por archivos que pueden ensamblarse de diferentes maneras para producir un sistema que se pueda ejecutar. En el UML, los aspectos estáticos de esta vista, son capturados en los *diagramas de componentes*; y los dinámicos, en los *diagramas de interacción, de estado y de actividades*.
- La *vista de instalación* se enfoca en los *nodos* que forman la topología del *hardware* del sistema en la cual se va a ejecutar. Esta *vista* principalmente se encarga de la distribución, entrega y la instalación de las partes que componen la parte física del sistema. En el UML, los aspectos estáticos de esta vista, son capturados en los *diagramas de instalación*; y los dinámicos, en los *diagramas de interacción, de estado y de actividades*.

No todos los sistemas requieren de todas las *vistas*, el número de *vistas* va a depender del sistema que se esté construyendo. Por ejemplo un procesador de texto sencillo, no requiere de una *vista de implementación*.

El UML, proporciona un estándar para los *artefactos* de desarrollo que deben ser controlados e intercambiados, más no un estándar para el proceso de desarrollo. El UML no está atado a ningún ciclo de vida del desarrollo de *software* en particular. Sin embargo, para obtener un mejor beneficio del UML, se debe considerar un proceso que sea *conducido por casos de uso*, es decir, que los *casos de uso* sean utilizados como los *artefactos* elementales para establecer el comportamiento deseado del sistema, para verificar y validar la *arquitectura* del sistema, para pruebas y para comunicar a los diferentes equipos involucrados en el proyecto. También debe considerarse un proceso que se *centre en su arquitectura*, es decir, que la *arquitectura* del sistema sea utilizada como *artefacto* elemental para visualizar, construir y administrar el sistema durante el desarrollo. Por último se debe considerar un proceso que sea *iterativo e incremental*, es decir, un proceso que permita la liberación de varias versiones del sistema y que en cada nueva versión se incorporen mejoras que se centren en atacar los riesgos más significativos con la finalidad de alcanzar el éxito del proyecto.

Un proceso que cumpla con estas tres características puede partirse en *fases*. Una *fase* es un intervalo de tiempo entre dos puntos importantes del proceso, en el momento que se reúne un conjunto de objetivos bien definidos, los *artefactos* son completados y las decisiones son tomadas, se puede pasar a la siguiente *fase*. En la Figura 1.2-22 se grafican las cuatro *fases* del ciclo de vida de desarrollo de *software*: *concepción o comienzo*, *elaboración*, *construcción* y *transición*, contra los flujos de trabajo, mostrando la variación de importancia a través del tiempo.

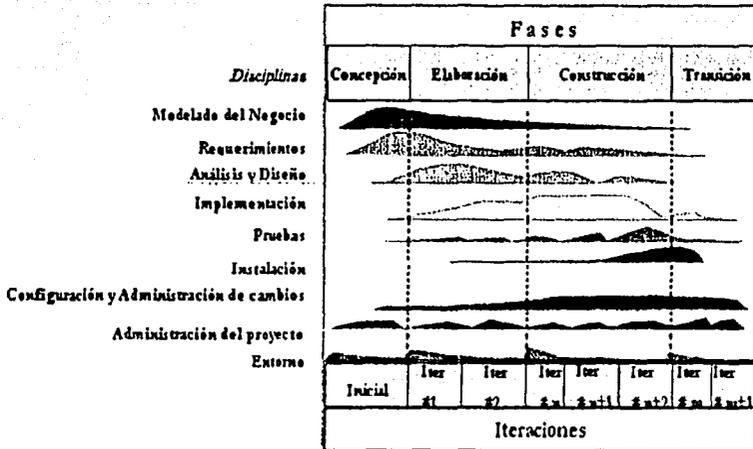


Figura 1.2-22

En la *concepción o comienzo* surge la semilla de ideas para el desarrollo, a tal punto de ser lo suficientemente bien fundadas para garantizar la entrada a la fase de *elaboración*.

En la *elaboración* la visión del producto y su arquitectura son definidas. En esta fase los requerimientos del sistema son articulados y se les dan prioridades. Los requerimientos del sistema pueden cubrir desde aspectos muy generales hasta criterios de evaluación muy específicos. Cada especificación de un comportamiento funcional o no funcional, suministra una base para las pruebas.

En la *construcción*, el *software* es traído desde una *arquitectura* inicial, a una *arquitectura* lista para ser traspasada a la comunidad de los usuarios. En esta parte, también los requerimientos del sistema y sobre todo sus criterios de evaluación, son constantemente reevaluados frente a las necesidades de negocio del proyecto, y los recursos son asignados de una forma apropiada para enfrentar los riesgos del proyecto.

TESIS CON
FALLA DE ORIGEN

En la *Transición* el *software* es dejado en las manos de la comunidad de los usuarios. Raramente el proceso de desarrollo de *software* termina aquí, incluso en esta fase, debido a que continuamente, el sistema es perfeccionado, las fallas son eliminadas y se agregan características que no se incluyeron en versiones anteriores.

Un elemento que distingue a este proceso y que atraviesa las cuatro *fases* es la *iteración*. Una *iteración* es un conjunto distinguido de actividades, con un plan y un criterio de evaluación que resulta en una versión interna y/o externa. Esto significa que el ciclo de vida de desarrollo de *software* puede ser caracterizado, involucrando una serie continua de versiones ejecutables de la arquitectura del sistema. Es por esto que se pone énfasis en que la arquitectura sea un *artefacto* importante que conduzca al UML a modelar las diferentes *vistas* de la *arquitectura* de un *sistema*.

TESIS CON
FALLA DE ORIGEN

1.3 DIAGRAMAS DE CASOS DE USO

Un *modelo de casos de uso* describe los requerimientos funcionales de un *sistema*, en términos de *casos de uso*. En él se modelan las funciones previstas del *sistema* y de su ambiente. Sirve como un contrato entre los usuarios y los desarrolladores. Este *modelo* permite a los usuarios validar que el *sistema* llegue a ser como ellos lo esperan y permite a los desarrolladores asegurarse que lo que están construyendo sea exactamente el producto deseado. Debido a que es un instrumento de planeación muy poderoso, éste generalmente se utiliza en todas las *fases* del ciclo de vida de desarrollo del software como a continuación se describe: los diseñadores lo utilizan como base de su trabajo y para obtener una visión general del *sistema*; los probadores o control de calidad lo usan para planear sus actividades de pruebas lo antes posible; los encargados de desarrollar las siguientes versiones del *sistema* lo utilizan para entender como trabajan las versiones existentes; los encargados de documentar el *sistema* lo utilizan como base para escribir las guías de usuario; los arquitectos lo revisan para identificar arquitectónicamente la funcionalidad significativa. Todo lo anterior es de importancia, sin embargo el papel más importante del *modelo* es el de comunicar el comportamiento del *sistema* a los usuarios.

Un *modelo de casos de uso* consiste de *casos de uso* y de *actores*. Un *actor* puede definirse como cualquier cosa que intercambie información con el *sistema*; un *actor* es externo al *sistema*, puede ser un recipiente de información; puede representar un humano o una máquina. Por otra parte, un *caso de uso* modela el dialogo entre los *actores* y el *sistema*; un *caso de uso* es iniciado por un *actor* con la finalidad de invocar cierto funcionamiento que se encuentra dentro del *sistema*. La relación entre un *caso de uso* y un *actor* se representa con una flecha dirigida (del *actor* al *caso de uso*) la cual recibe el nombre de *comunicación/ asociación*. En la Figura 1.3-1 se presenta un ejemplo de un *diagrama de casos de uso*.

TESIS CON
FALLA DE ORIGEN

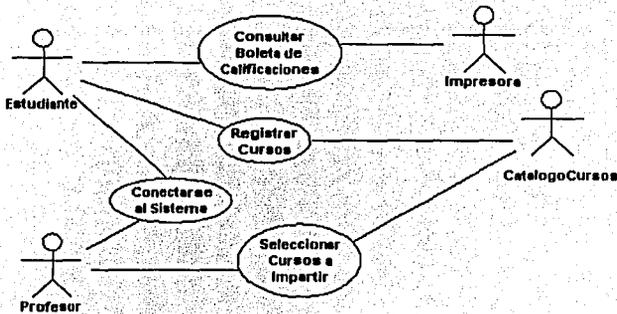


Figura 1.3-1

Cada *caso de uso* del modelo se describe a detalle, mostrando paso a paso la *interacción* del sistema con los actores y la función del sistema en el *caso de uso*. Las propiedades de cada *caso de uso* pueden documentarse en la *especificación del caso de uso*. Dicha especificación puede incluir: una breve descripción del *caso de uso*, los flujos de eventos¹³ del *caso de uso*, requerimientos especiales¹⁴, precondiciones¹⁵, post-condiciones¹⁶ y algún punto de extensión que se considere necesario. Como complemento para modelar los requerimientos del sistema, se puede elaborar un *glosario* con el fin de definir los términos comunes de todos los *modelos* y facilitar la comunicación entre los expertos del dominio y los desarrolladores. Se recomienda elaborar durante las fases de *concepción* y de *elaboración* ya que es importante tener una terminología en común desde un principio.

¹³ Los flujos de eventos son descripciones textuales de lo que el sistema realiza con relación al caso de uso. El flujo básico explica la secuencia más común de los eventos: la historia normal de las actividades y la terminación exitosa. Los flujos alternos muestran las excepciones que pueden ocurrir dentro del flujo básico. Puede haber varios flujos de eventos, por ejemplo, un flujo básico y diversos alternos.

¹⁴ Como las restricciones de diseño y aquellos requerimientos no funcionales que no se muestran en el *modelo de casos de uso* como son: los requerimientos que relacionan o afectan la utilidad del sistema (requerimientos de entrenamiento que especifican como el sistema puede ser utilizado por los actores con facilidad), cualquier requerimiento concerniente a la confiabilidad del sistema (medidas cuantitativas tales como el tiempo medio entre las fallas o defectos por cada mil líneas de código), las características de rendimiento del sistema (tiempos específicos de respuesta) y cualquier requerimiento relacionado con el mantenimiento del sistema.

¹⁵ Son condiciones que debe existir antes de que el *caso de uso* comience.

¹⁶ Son condiciones que debe existir después de que el *caso de uso* termine.

1.4 DIAGRAMAS DE INTERACCIÓN

Anteriormente se mencionó que los *diagramas de secuencias* y los de *colaboración*, son tipos de *diagramas de interacción*.

Un *diagrama de secuencia* describe un *patrón de interacción* entre *objetos*, organizado en orden cronológico. En este *diagrama* se muestra la participación de los *objetos* en la *interacción*, mediante sus *líneas de vida* y los *mensajes* que envían a otros *objetos*. Uno o más *diagramas de secuencia* pueden ilustrar las interacciones de los *objetos*, típicamente se tiene un *diagrama principal* para el flujo de eventos y uno para cada uno de los sub-flujos del *caso de uso*. Los *diagramas de secuencia* son de suma importancia para los diseñadores, ya que les ayuda a tener claro los roles o papeles de los *objetos* en un flujo y a determinar las *interfaces* y las responsabilidades¹⁷ de las *clases*.

Un *diagrama de secuencia* contiene *objetos*, *instancias de actores*, *mensajes*, *focos de control* y *líneas de vida*. La *línea de vida* representa la existencia del *objeto* en un tiempo determinado. El *objeto* se dibuja en la parte superior de la *línea de vida* incluyendo su *nombre* y su *clase* subrayados y separados por dos puntos: *nombre del objeto: nombre de la clase*. La *clase* de los *objetos* puede no especificarse. Normalmente se crea un *diagrama de secuencia* con *objetos* especificando más tarde sus *clases*. También puede omitirse el *nombre* de los *objetos*, sin embargo, si se quiere hacer distinción entre *objetos* de la misma clase, se recomienda nombrarlos.

La *instancia de un actor* es quien inicia la *interacción* y generalmente es representada por la primera *línea de vida* de la izquierda. Si se tienen varias *instancias de actores* en el mismo *diagrama*, se debe hacer lo posible por dejar una *instancia* en la primera *línea de vida* del lado izquierdo y la otra en la última *línea de vida* del lado derecho. La *interacción* entre actores no se muestra en el *diagrama*, ya que los *actores* por definición son externos al *sistema*.

Un *mensaje* se muestra con una flecha horizontal que va de la *línea de vida* de un *objeto* a la *línea de vida* de otro *objeto*. La flecha se etiqueta con el nombre del *mensaje* y con sus parámetros. La etiqueta se puede numerar para mostrar la secuencia de los *mensajes* en la interacción. Por último, el *foco de control* es un rectángulo alto y delgado que representa el periodo de tiempo en el cual un *objeto* ejecuta una acción directamente o a través de un procedimiento

¹⁷ Una responsabilidad es un contrato o una obligación de una clase.

dependiente de otro *objeto*. La parte superior del rectángulo se alinea con el inicio de la acción y la parte inferior con la culminación. En la Figura 1.4-1 se muestra un *diagrama de secuencia*.

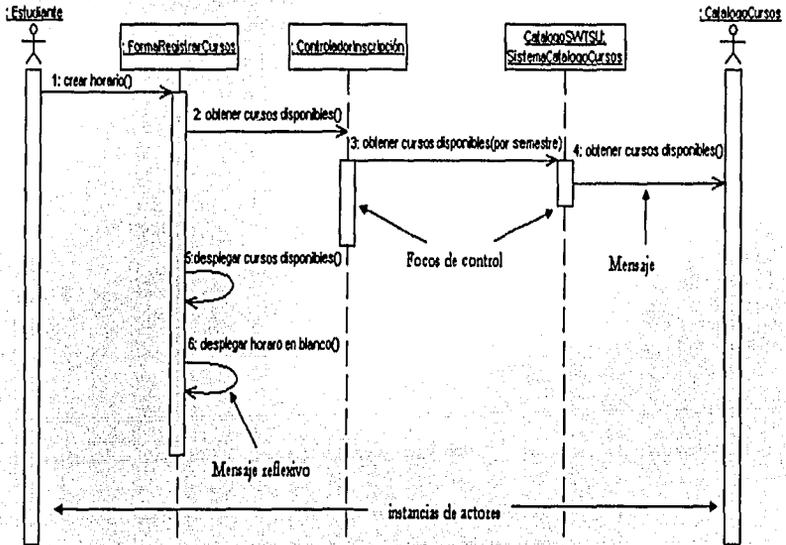


Figura 1.4-1

Un *diagrama de colaboración* muestra la forma en que interactúan los *objetos* para realizar el comportamiento de un *caso de uso* en particular o de una parte de un *caso de uso*. Estos *diagramas*, son la fuente primaria de información que se utiliza para determinar las *interfases* y las responsabilidades de las *clases*. Dada su estructura, los *diagramas de colaboración* son más apropiados para las actividades de análisis; específicamente para representar las *interacciones* más simples de un número pequeño de *objetos*. Conforme el número de *objetos* y *mensajes* crece, el *diagrama* es más difícil de entender. Además es difícil de mostrar en estos *diagramas*, información adicional descriptiva o información no estructurada que puede ser más sencillo agregar como una *nota* en un *diagrama de secuencias*.

Un *diagrama de colaboración* contiene *objetos*, *instancias de actores*, *ligas* y *mensajes*. Los *objetos* y *mensajes* se representan de la misma manera que en un *diagrama de secuencias*.

Una *liga* es la relación entre *objetos* por la cual se pueden enviar los *mensajes*. Puede ser la *instancia* de una *asociación* o puede ser anónima, es decir su *asociación* no se especifica. Se representa con una línea sólida entre dos *objetos*. Figura 1.4-2.

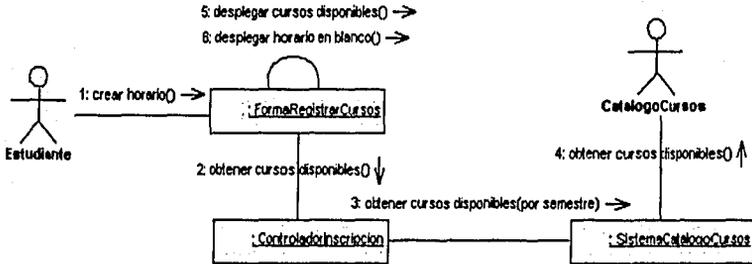


Figura 1.4-2

Dado que ambos *diagramas* (de *secuencia* y de *colaboración*) derivan la misma información, se dice que son *semánticamente equivalentes*. Como consecuencia, se puede tomar cualquiera de los dos *diagramas* y convertirlo al otro sin perder información. A pesar de esto, se pueden mencionar las siguientes observaciones: los *diagramas de colaboración* son mejores para visualizar patrones de *colaboración*, para visualizar todos los efectos de un *objeto* dado y son más fáciles de utilizar en sesiones de lluvia de ideas; los *diagramas de secuencia* son mejores para visualizar el flujo completo, para especificaciones de tiempo real y para *escenarios* complejos.

TEMA CON
FALLA DE ORIGEN

1.5 DIAGRAMAS DE CLASES

Al modelar la vista estática de un sistema, los *diagramas de clases* son típicamente utilizados para modelar el vocabulario del *sistema* al determinar las abstracciones que son parte de, o que están fuera del *sistema*; es decir, las fronteras del *sistema*. Los *diagramas de clases* especifican estas abstracciones y sus responsabilidades. También se utilizan para modelar un diseño de base de datos; es decir, los *diagramas de clases* son un super conjunto de diagramas entidad-relación (E-R), con la diferencia de que los diagramas de entidad-relación sólo se enfocan en los datos y los de *clases* también modelan el comportamiento del sistema. En la Figura 1.5-1 se muestra un *diagrama de clases*.

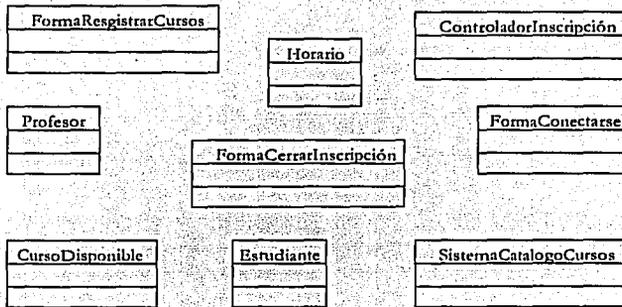


Figura 1.5-1

No es raro encontrar *sistemas* en desarrollo que contengan cientos o incluso miles de *clases* diferentes. Administrar un número tan grande de *clases* puede ser problemático. Es en esta parte donde se hace uso de los *paquetes*. En la Figura 1.5-2 se muestra la agrupación de *clases* en un *paquete*.

TESIS CON
FALLA DE ORIGEN

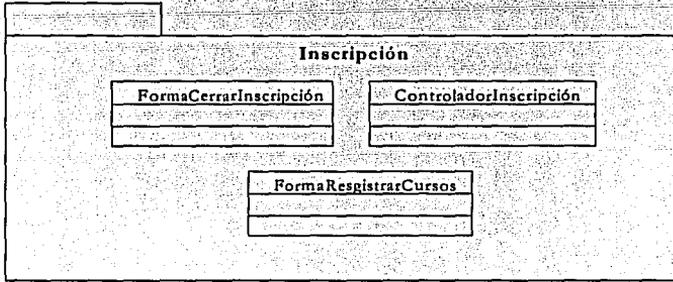


Figura 1.5-2

Un *diagrama de clases* contiene *asociaciones* (anteriormente se definió una *asociación* entre *objetos* y hay que recordar que un *objeto* es una *instancia* de una *clase*). Generalmente el tipo de *asociaciones* son simples, es decir, la *asociación* existe exactamente entre dos *clases*. En ocasiones una *clase* puede tener una *asociación* a ella misma. Esto no siempre significa que una *instancia* de una *clase* tenga una *asociación* a ella misma, más bien que una *instancia* de una *clase* tiene una *asociación* a otra *instancia* de la misma *clase*. A esta *asociación* se le conoce como *reflexiva*. Figura 1.5-3.

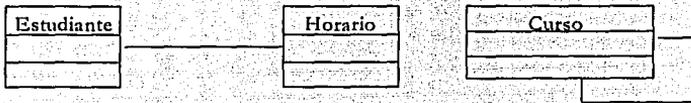


Figura 1.5-3

TESIS CON
FALLA DE ORIGEN

Para cada *asociación* existen dos decisiones de *multiplicidad* que se tienen que tomar. La *multiplicidad* indica cuantas *instancias* de una *clase* están relacionadas con una *instancia* de otra *clase*. En la Figura 1.5-4 se muestra los siete tipos de indicadores para la *multiplicidad*.

| | |
|---------------------------------------|--------|
| No especificada | _____ |
| Exactamente una | _____ |
| | 1 |
| Ninguna o más (muchas, ilimitadas) | _____ |
| | 0..* |
| Una o más (muchas, ilimitadas) | _____ |
| | 1..* |
| Ninguna o una | _____ |
| | 0..1 |
| Un rango en específico | _____ |
| | 2..4 |
| Múltiples rangos | _____ |
| | 2,4..6 |

Figura 1.5-4

TESIS CON
FALLA DE ORIGEN

Un *diagrama de clases* también puede incluir *agregaciones*. Las agregaciones se utilizan para modelar relaciones con las que se quiere decir que algo es una parte de. Por ejemplo, una librería contiene libros, los departamentos de una empresa están compuestos por empleados, una computadora esta compuesta por un número de dispositivos. Para modelar una *agregación*, el agregado o el todo (departamento) tiene una *asociación-agregación* a sus partes que lo constituyen (empleado).

En la Figura 1.5-5 se muestra un *diagrama de clases* con *asociaciones*, *multiplicidad* y con una *agregación*.

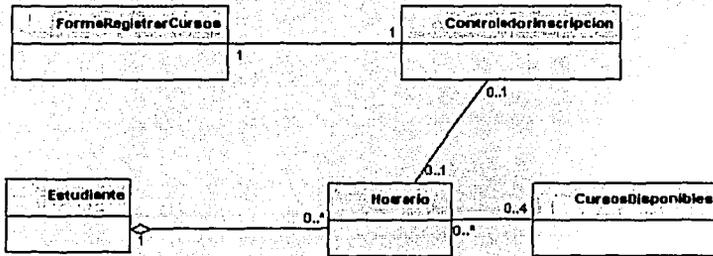


Figura 1.5-5

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 2

ANÁLISIS ORIENTADO A OBJETOS CON UML

2.1 INTRODUCCIÓN A LA DISCIPLINA DE ANÁLISIS Y DISEÑO

Antes de describir la disciplina de análisis y diseño, comenzaremos con una breve descripción de la disciplina de requerimientos. Dicha disciplina nos permite leer e interpretar los artefactos de requerimientos que sirven como punto de partida para las actividades de análisis y diseño.

Los propósitos de la disciplina de requerimientos son:

- Establecer y mantener un acuerdo entre los usuarios y las personas involucradas en el proyecto.
- Proporcionar a los desarrolladores un mejor entendimiento de los requerimientos del *sistema*.
- Definir los límites o fronteras del *sistema*.
- Proveer una estimación de costo y tiempo de desarrollo del *sistema*.

La disciplina de requerimientos es sumamente importante y debe de realizarse con cuidado ya que de todo lo que se plasme o recopile en los artefactos de requerimientos (*modelos de casos de uso*, especificaciones adicionales de los *casos de uso*, glosario), se basan las siguientes disciplinas para continuar con el desarrollo del *sistema*.

La disciplina de análisis y diseño funciona de la misma forma en que se modela, es decir, se comienza con un *modelo de objetos* que se asemeja al mundo real (análisis) y después se encuentran soluciones más abstractas (pero más fundamentales) a un problema más generalizado (diseño). El análisis se enfoca en entender el problema y comenzar a desarrollar un *modelo visual* acerca de lo que estamos tratando de construir, independientemente de la *implementación* y de las cuestiones de tecnología, mientras que el diseño se enfoca en entender la solución al problema. El análisis se enfoca en trasladar los requerimientos funcionales en conceptos de *software* y el diseño los no funcionales.

En el análisis se idealiza el diseño y en el diseño se intenta desarrollar un modelo el cual nos permita tener una transición transparente a la fase de codificación.

TESIS CON
FALLA DE ORIGEN

El análisis se compone de dos actividades: análisis arquitectónico y análisis de *casos de uso*. En el análisis arquitectónico se comienza a definir las partes o piezas del sistema y sus relaciones, organizando estas piezas o partes en capas muy bien definidas con dependencias explícitas, centrándose en las capas de mayor nivel o más generales del sistema. Los propósitos del análisis arquitectónico son: proveer los insumos al proceso de planeación, definir los patrones arquitectónicos, los mecanismos de análisis, las convenciones de modelado para el *sistema* e identificar las abstracciones clave del *sistema*.

Los mecanismos de análisis ayudan a definir el comportamiento de una *clase* o *componente*, o la cooperación necesaria entre *clases* y/o *componentes*. Entre estos mecanismos tenemos: persistencia, comunicación, distribución, manejo de transacciones, seguridad, redundancia, interfaz heredada y base de datos heredada.

Las convenciones de modelado proporcionan un estilo guía para el modelo. Es aquí donde se definen los diagramas y elementos del modelado que se utilizarán así como las reglas de uso de los elementos y de los diagramas, convenciones de nombres, etc.

Con la identificación de abstracciones clave, se busca obtener un primer acercamiento a los elementos clave que colaboran en el *sistema* para llevar a cabo la solución del problema. La identificación de elementos implica: definir *clases de análisis* preliminares (las podemos obtener del conocimiento del dominio, de los requerimientos, en el glosario, en el modelo de dominio o modelo de negocios si lo hubiera), definir relaciones entre las *clases de análisis*, modelar *clases de análisis* y relaciones en *diagramas de clases* y asignar las *clases de análisis* a los mecanismos de análisis necesarios.

La actividad de análisis arquitectónico puede omitirse en caso de que el riesgo arquitectónico sea bajo o en caso de que no se tenga experiencia en este análisis.

En el análisis de *casos de uso* es donde se identifican las *clases* iniciales del *sistema*. Los objetivos en este análisis son: identificar las *clases* que intervienen en el flujo de eventos de los *casos de uso*; distribuir el comportamiento de los *casos de uso* entre las *clases*; identificar las responsabilidades, *atributos* y *asociaciones* de dichas *clases*. El análisis de casos de uso es una de las actividades más importantes (si no es que la más importante) dentro de las actividades en la disciplina de análisis y diseño.

Además del modelo de *casos de uso*, esta actividad puede apoyarse en la identificación de abstracciones clave, en los mecanismos de análisis y en la especificación de capas del sistema.

TESIS CON
FALLA DE ORIGEN

2.2 ANÁLISIS DE CASOS DE USO

El análisis de *casos de uso* consta de los siguientes pasos:

1. Complementar la descripción del flujo de eventos de los casos de uso.

El flujo de eventos de los *casos de uso* se enfoca en las interacciones entre los usuarios y el *sistema*, funcionando como una caja negra, en la cual, los detalles internos son omitidos. Como consecuencia, es necesario convertir esa caja negra en una caja transparente, en la cual se empieza a describir la manera en que el sistema trabajará internamente. Por ejemplo: supongamos que se tiene un paso en el *caso de uso* Registrar Cursos de un Sistema de Inscripción a Cursos, que diga: *El sistema despliega una lista de los cursos disponibles*, el texto posiblemente se podría expandir de la siguiente forma: *El sistema recupera una lista de los cursos actualmente disponibles del catálogo de cursos de la base de datos heredada.*

2. Para cada *caso de uso* se necesita:

- Buscar *clases (clases de análisis)* a partir de la descripción del comportamiento del *caso de uso*.

El propósito de este paso es identificar un conjunto candidato de elementos de modelado (*clases de análisis*) que sea capaz de ejecutar el comportamiento descrito en los *casos de uso*.

Existen tres *estereotipos de clases de análisis*:

- a) *Clase frontera*.- se utiliza para modelar la interacción entre el *sistema* y sus alrededores (*actores*). Esta interacción transforma y traslada eventos y notifica los cambios en la presentación del *sistema*. Una *clase frontera* es la *interfaz* o intermediario entre el *sistema* y elementos externos al mismo. Existe una *clase frontera* por cada pareja *actor/caso de uso*. Figura 2.2-1.

TESIS CON
FALLA DE ORIGEN

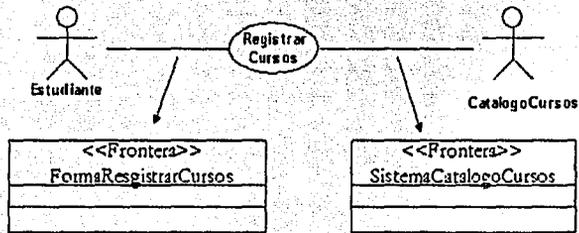


Figura 2.2-1.

Al identificar este tipo de *clases* es importante concentrarse en las responsabilidades y no irse a los detalles de la *interfaz* con el usuario o de cómo implementar protocolos de comunicación con dispositivos o con otros *sistemas*. Cabe mencionar que los *objetos frontera* (instancias de las *clases frontera*) normalmente tienen un tiempo de vida tan largo como una *instancia* de un *caso de uso*.

- b) *Clase entidad*.-se utiliza para modelar información que debe ser almacenada en el *sistema*. Una *clase entidad* por lo general no es específica de un *caso de uso*, en algunos casos, no es ni específica del *sistema*. Los *objetos entidad* (instancias de las *clases entidad*) son utilizadas para guardar y actualizar información sobre algún fenómeno, como un evento, una persona, o algún objeto de la vida real. La complejidad de un *objeto entidad* está relacionada con el *objeto* que representa. Los *objetos entidad* son independientes del entorno (*actores*). En la Figura 2.2-2 se muestran ejemplos de *clases entidad*.

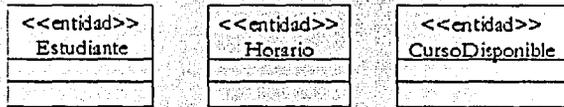


Figura 2.2-2.

- c) *Clase control*, controlan el comportamiento de uno o varios *casos de uso*. Las *clases control* contribuyen a entender al *sistema* ya que representan la dinámica del sistema, manejando las principales tareas y flujos de control. Un *objeto control* (*instancias de las clases control*) generalmente muere cuando termina la ejecución de un *caso de uso*. Los *casos de uso* más complejos requieren una o más *clases control* para coordinar el comportamiento de los *objetos* en el *sistema*. Una recomendación para la identificación inicial de las *clases control* es definir una *clase control* por cada *caso de uso*. Figura 2.2-3.

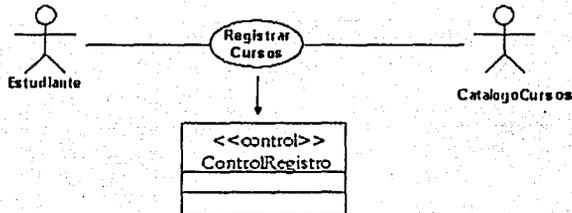


Figura 2.2-3.

3. Para cada *caso de uso* se necesita:

- Distribuir el comportamiento a las clases.

Los propósitos de este paso son: expresar el comportamiento de los *casos de uso* en términos de la *colaboración* de las *clases de análisis* y determinar las responsabilidades de las *clases de análisis*. Esto implica que una vez identificadas las *clases de análisis* para cada flujo en los *casos de uso* se debe:

TESIS CON
FALLA DE ORIGEN

- a) Asignar las responsabilidades del *caso de uso* a las *clases de análisis*.- una responsabilidad es un contrato o una obligación de la *clase* y son realizadas por sus *operaciones* y *atributos*. A una *clase frontera* se asigna el comportamiento que involucre comunicación con un *actor*; a una *clase entidad* se asigna el comportamiento que involucre *encapsulamiento* y *abstracción* de datos y a una *clase control* se asigna el comportamiento específico para un *caso de uso* o una parte importante del flujo de eventos. En este punto resulta importante hacernos la siguiente pregunta: ¿Quién tiene la responsabilidad sobre un dato?; si una *clase* tiene el dato, se asigna la responsabilidad de ese dato a la *clase*. Si múltiples *clases* tienen el dato: se pone la responsabilidad en una *clase* y se añaden relaciones a las otras *clases*; se crea una nueva *clase*, se asigna la responsabilidad en esta nueva *clase* y se agregan relaciones a las otras *clases*; se pone la responsabilidad en una *clase control* y se añaden relaciones a las *clases* que requieran esa responsabilidad.

Debemos cuidar que todas las relaciones que se agreguen sean consistentes con las *clases* que conecten. Sin embargo, hay que tener cuidado y solo debemos crear *clases* nuevas, cuando estemos seguros de que no existe ninguna *clase* existente que puede ejecutar la responsabilidad.

- b) Modelar las interacciones entre las *clases de análisis* con *diagramas de interacción*.-el modelado de interacciones permite ver la *colaboración* entre los *objetos* en un *caso de uso* en particular.

4. Para cada *clase de análisis* encontrada se debe:

- Describir sus responsabilidades

Las responsabilidades se constituyen en una o más *operaciones* en las *clases* de diseño; estas pueden ser caracterizadas por la acción que el *objeto* ejecuta o por el conocimiento que el *objeto* mantiene y proporciona a otros *objetos*. En este nivel (de análisis), las responsabilidades son derivadas de los *mensajes* en los *diagramas de interacción*. Para cada *mensaje*, se debe examinar la *clase* del *objeto* que recibe el *mensaje*. Si la responsabilidad no existe todavía, se deberá crear una nueva responsabilidad que proporcione el comportamiento solicitado. Otras responsabilidades pueden

derivarse de requerimientos no funcionales. Puede ser necesario complementar la descripción de alguna responsabilidad o crear una nueva responsabilidad para un requerimiento de ese tipo. Las responsabilidades de las *clases de análisis* pueden ser documentadas de dos formas:

- a) Como *operaciones* (poniéndoles un nombre) de análisis. Estas *operaciones* probablemente serán cambiadas o evolucionarán en el diseño.
- b) Textualmente como parte de la descripción de las *clases de análisis*.

En la Figura 2.2-4 se muestra como un mensaje en un *diagrama de interacción* deriva una responsabilidad a un *diagrama de clases*.

Diagrama de Interacción

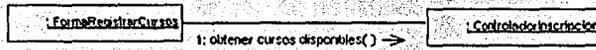


Diagrama de Clases

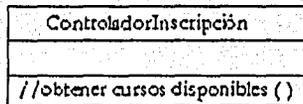


Figura 2.2-4.

5. Para cada *clase de análisis* encontrada se debe:

- Describir los *atributos y asociaciones*

Los *atributos* son utilizados para almacenar información. Durante el análisis, los tipos de los *atributos* pueden indicarse, aunque no es necesario que correspondan con el lenguaje de programación en uso. Un atributo puede ser visto desde diferentes perspectivas. Por ejemplo, pensemos en una *clase Cliente*, con un *atributo* nombre. Desde una perspectiva conceptual el *atributo* nombre indica que los clientes tienen nombres; desde un *modelo* de diseño, este *atributo* indica que un *objeto* Cliente puede decir su nombre y tiene algún modo de establecer su nombre; en el *modelo* de implementación se dice que Cliente tiene un miembro o variable de *instancia* para su nombre. Dependiendo del detalle del diagrama y del modelo, la notación de un *atributo* puede mostrar: nombre, tipo, valor

predeterminado y visibilidad. Entre los tipos de *atributos* más comunes tenemos: numérico, cadena o texto, fecha y hora.

Por otra parte una *asociación* representa relaciones entre *objetos*, estas indican alguna conexión significativa entre las instancias. Desde la perspectiva de análisis, las *asociaciones* representan relaciones conceptuales entre *clases*. En el análisis, una asociación se considera abstracta y no es una afirmación sobre las conexiones entre las entidades del *software*.

- Describir los mecanismos de análisis

Este paso se lleva a cabo solo cuando se haya realizado el análisis arquitectónico. Consiste en relacionar los mecanismos de análisis identificados y definidos en el análisis arquitectónico con las *clases de análisis*. No todas las clases tienen mecanismos de análisis asociados. También puede suceder que algunas clases requieran de varios mecanismos.

Las *clases* que requieran de un mecanismo de escritura y/o lectura (por ejemplo almacenar y consultar la información de un estudiante) son relacionadas con el mecanismo de *persistencia*; las *clases* cuyo mantenimiento se lleva a cabo fuera del *sistema* que estamos construyendo (cualquier otro *sistema* donde se consulte información desde nuestro *sistema*), se relacionan con el mecanismo de *interfaz heredada*; las *clases* que necesiten tener un acceso controlado (un estudiante solo puede modificar su horario) se relacionan con el mecanismo de *seguridad* y las *clases* que son identificadas como distribuidas (por ejemplo controladores), se relacionan con el mecanismo de *distribución*. En la Figura 2.2-5 se muestran algunos ejemplos de relaciones.

| Clases de Análisis | Mecanismos de Análisis |
|---------------------|---------------------------------|
| Estudiante | Persistencia, Seguridad |
| Horario | Persistencia, Seguridad |
| CursoDisponible | Persistencia, Interfaz heredada |
| Curso | Persistencia, Interfaz heredada |
| Control de Registro | Distribución |

Figura 2.2-5.

TESIS CON
FALLA DE ORIGEN

6. Unificar *clases de análisis*.

El propósito de este paso es asegurar que cada *clase de análisis* represente un concepto simple y bien definido que no mezcle responsabilidades.

Antes de que el trabajo de la arquitectura pueda ser realizado, las *clases de análisis* necesitan ser filtradas para asegurar que un número mínimo de conceptos nuevos sean creados. El nombre de la *clase de análisis* debe capturar la esencia del rol jugado por la clase en el sistema. Estos nombres deben ser únicos.

Clases que definan un comportamiento similar o un mismo fenómeno deben ser unidas. Se deben unir también *clases entidad* que definan los mismos atributos, aún cuando el comportamiento definido sea distinto, agregando el comportamiento de la *clase* que se esta unificando. Al actualizar cualquier *clase*, debe actualizarse cualquier referencia a esa *clase* en los *casos de uso*.

TESIS CON
FALLA DE ORIGEN

2.3 ANÁLISIS DE DOS CASOS DE USO DE UN SISTEMA DE NÓMINA

Antes de empezar con el análisis de los *casos de uso*, presentaremos el *diagrama de casos de uso* de un sistema de nómina. Figura 2.3-1.

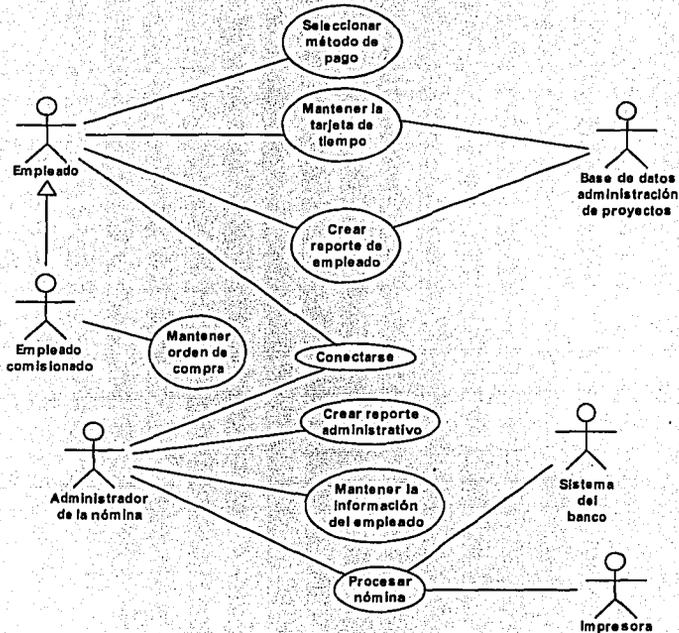


Figura 2.3-1.

Para efectos prácticos, haremos el análisis solo para dos *casos de uso*: Procesar nómina y Mantener tarjeta de tiempo.

La descripción del flujo de eventos de dichos *casos de uso*, ha sido completada; con lo que queda cubierto el primer paso del análisis de *casos de uso*.

1) Caso de Uso: Procesar Nómina

1.1. Breve Descripción

Este caso de uso describe la forma en que se procesa la nómina los viernes y los últimos días de trabajo del mes.

1.2. Flujo de Eventos

1.2.1. Flujo Básico

Este caso de uso inicia cuando es tiempo de procesar la nómina. La nómina es procesada por el administrador de la nómina todos los viernes y los últimos días de trabajo del mes.

1. El sistema recupera todos los empleados a los que se les debe pagar en el periodo de pago específico.
2. El sistema calcula el pago utilizando la tarjeta de tiempo del empleado (es donde se registran las horas trabajadas y proyectos facturados del empleado), sus órdenes de compra, información del empleado (salario, beneficios, etc.) y todas las deducciones legales.
3. Si la forma de pago es por correo o en caja, el sistema imprime un cheque.
4. Si la forma de pago es por depósito, el sistema crea una transacción bancaria la cual la envía al sistema del banco para procesarla.
5. El caso de uso termina cuando todos los empleados recuperados por el sistema reciben su pago en el periodo de pago específico y el administrador de la nómina notifica que el proceso de nómina se ha completado.

1.2.2. Flujos Alternativos

1.2.2.1. Sistema Bancario no disponible

Si el sistema bancario no está disponible, el sistema intentará después de un tiempo determinado enviar la transacción bancaria. El sistema

TESIS CON
FALLA DE ORIGEN

continuará intentando re-transmitir la transacción hasta que el sistema bancario este disponible.

1.3. Requerimientos Especiales

Ninguno.

1.4. Pre - Condiciones

Ninguna.

1.5. Post - Condiciones

Los pagos para cada empleado han sido procesados.

1.6. Puntos de Extensión

Ninguno.

TESIS CON
FALLA DE ORIGEN

Análisis del Caso de Uso: Procesar Nómina

Enfocándonos en el *diagrama de casos de uso* del sistema de nómina (Figura 2.3-2), en el caso de

uso: procesar nómina obtenemos:

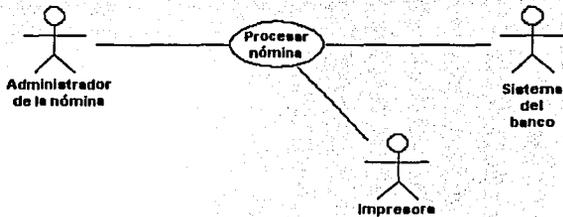


Figura 2.3-2

1) Comenzamos con la identificación de clases.

a) De tipo frontera:

Recordemos que una clase frontera es una *interfaz* o intermediario entre el sistema y elementos externos al mismo y que existe una *clase frontera* por cada pareja actor/caso de uso. Basándonos en lo anterior, para este caso de uso tenemos las siguientes tres clases de tipo frontera: Figura 2.3-3

TESIS CON
FALLA DE ORIGEN

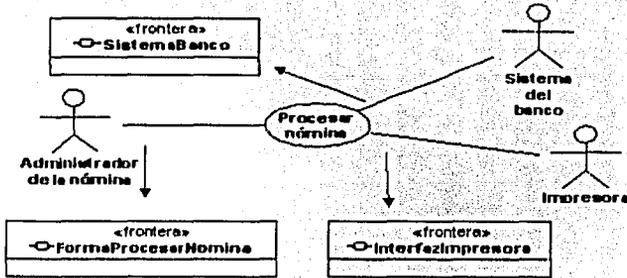


Figura 2.3-3

b) De tipo entidad:

Las fuentes potenciales para encontrar este tipo de *clases* son: el glosario, modelo del negocio (si lo hubiera) y los *casos de uso*. Para identificar *clases de entidad* en los *casos de uso*, se sugiere subrayar los sustantivos en el flujo de eventos del *caso de uso*. Una vez identificados se debe quitar: candidatos redundantes, candidatos *actores* ambiguos, términos de implementación, *atributos* (dejarlos para después) y las *operaciones*. Al filtrar los sustantivos debemos estar concientes de que distintos términos pueden referir a un mismo *objeto*, un término puede referirse a más de un objeto, el lenguaje natural es muy ambiguo y de que un sustantivo puede estar disfrazado como verbo o un verbo puede estar disfrazado como sustantivo.

Los resultados al identificar las *clases*, dependen de la habilidad de escritura de la persona que esta elaborando esta tarea.

Para este *caso de uso* se tienen las siguientes *clases*:



Figura 2.3-4

c) De tipo control:

Una *clase control* es lo único que debe enfocarse es en dirigir a las clases de otro tipo, indicándoles que deben hacer algo. Recordando que un buen inicio para la identificación *clases control* es definir una *clase control* por cada *caso de uso*, tenemos:

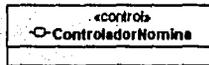


Figura 2.3-5

TESIS CON
FALLA DE ORIGEN

2) Distribuir el comportamiento.

Para cada *clase de análisis* encontrada, tenemos que asignarles responsabilidades de acuerdo al tipo de *clase de análisis* que se trate (como se indicó en el capítulo dos sección 2.2):

| Clase de análisis | Tipo de clase de análisis | Responsabilidades |
|--------------------|---------------------------|--|
| FormaProcesoNómina | Frontera | Indicar el inicio del proceso de nómina |
| SistemaBanco | Frontera | Enviar transacción bancaria al sistema del banco |
| InterfazImpresora | Frontera | Imprimir cheque de pago |
| Empleado | Entidad | Obtener tarjeta de tiempo correspondiente |
| TarjetaTiempo | Entidad | Obtener la información de la tarjeta de tiempo |
| OrdenCompra | Entidad | Obtener la información de la orden de compra |
| Cheque | Entidad | Crear cheque con la cantidad a pagar |
| ControladorNómina | Control | Procesar la nómina |

En seguida, el modelado del comportamiento del *caso de uso* lo haremos utilizando *diagramas de colaboración*. Un solo *diagrama de colaboración* no es suficiente ya que debemos modelar la mayor parte del flujo de eventos para asegurarnos que todos o casi todos los requerimientos estén identificados en las *operaciones* de las *clases* participantes. No obstante, podemos comenzar a modelar el flujo básico, el cual es el más común e importante de los eventos. Para fines prácticos presentamos únicamente el *diagrama de colaboración* del flujo básico.

TESIS CON
FALLA DE ORIGEN

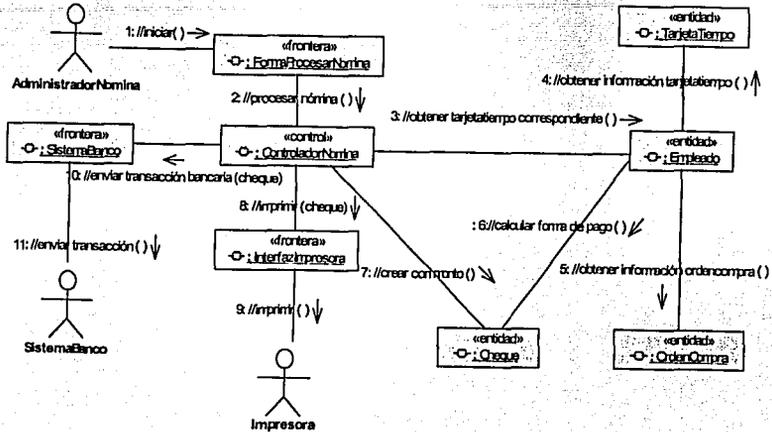


Figura 2.3-6

3) Describir responsabilidades.

Una vez modelado el comportamiento del caso de uso, crearemos el *diagrama de clases participantes* el cual se enfoca en las *clases* identificadas en el caso de uso y en las responsabilidades que se asignaron a cada *clase*. En nuestro caso el *diagrama de clases participantes* es el siguiente:

TESIS CON FALLA DE ORIGEN

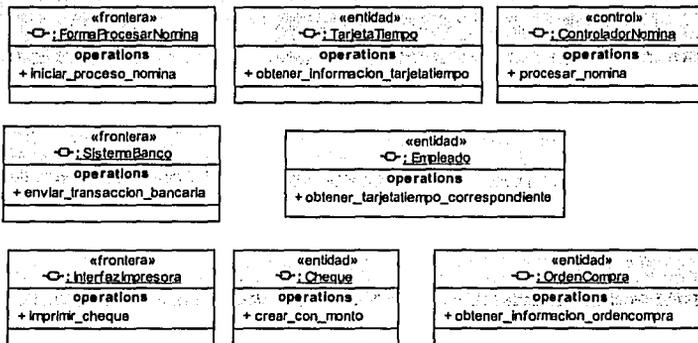


Figura 2.3-7

4) Describir *atributos y asociaciones*.

Agregaremos *atributos* a cada *clase*, en base a la información que nos interese almacenar en cada una de las *clases*. Los *atributos* pueden ser: las propiedades o características de las *clases* identificadas. La información cuyo valor es lo que importa y que por un lado únicamente pertenece a un *objeto* o bien es información que no representa comportamiento.

En cuanto a las *asociaciones*, una primera aproximación es tener una *asociación* por cada liga que se tenga en el *diagrama de colaboración* correspondiente a este *caso de uso*. Cada *asociación* tiene dos papeles, cada papel es una dirección en la asociación y cada uno de ellos tiene *multiplicidad*, la cual es una indicación de la cantidad de objetos que participan en la relación de *asociación*.

Siguiendo lo anterior, el *diagrama de clases participantes* se modifica al siguiente diagrama:

TESIS CON FALLA DE ORIGEN

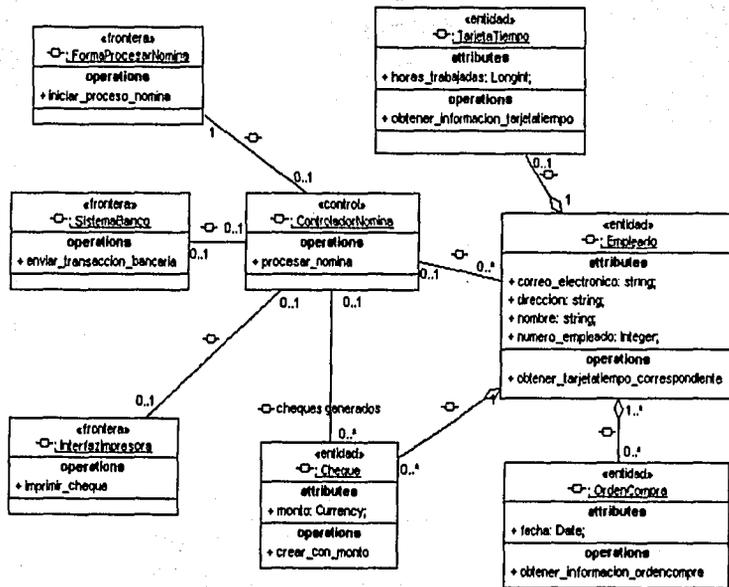


Figura 2.3-8

5) Describir mecanismos de análisis.

En este caso encontramos las siguientes relaciones:

| Clases de Análisis | Mecanismos de Análisis |
|---------------------|-------------------------|
| SistemaBanco | Interfaz heredada |
| Empleado | Persistencia, Seguridad |
| Cheque | Persistencia |
| ControladorNomina | Distribución |
| InterfazImpresora | Ninguno |
| OrdenCompra | Persistencia |
| TarjetaTiempo | Persistencia |
| FormaProcesarNomina | Ninguno |

TESIS CON FALLA DE ORIGEN

6) Unificar *clases*.

Este caso lo omitiremos ya que únicamente tenemos identificadas las *clases de análisis* de un *caso de uso*.

TESIS CON
FALLA DE ORIGEN

2) Caso de uso: Mantener la tarjeta de tiempo.

2.1. Breve Descripción

Este caso de uso permite al empleado actualizar y confirmar la información de su tarjeta de tiempo. Tanto los empleados asalariados como los que trabajan por horas, deben confirmar semanalmente su tarjeta de tiempo registrando el número de horas trabajadas durante la semana y en que proyectos se facturaron las horas trabajadas. Un empleado solamente puede hacer cambios en su tarjeta de tiempo en un periodo de pago específico, siempre y cuando la información de la tarjeta de tiempo no haya sido confirmada por el empleado.

2.2. Flujo de Eventos

2.2.1. *Flujo Básico*

Este caso de uso inicia cuando el empleado desea ingresar las horas trabajadas en su tarjeta de tiempo.

1. El sistema recupera y despliega la tarjeta de tiempo del empleado. Si la tarjeta de tiempo no existe para el empleado para el pago específico, el sistema crea una. Las fechas (de inicio y fin) de la tarjeta de tiempo son determinadas por el sistema y no pueden ser modificadas por el empleado.
2. El sistema recupera y despliega la lista de los números de factura disponibles en la base de datos heredada (administración de proyectos).
3. El empleado selecciona los números de factura apropiados y registra las horas trabajadas para cualquier fecha deseada (dentro del rango de fechas de la tarjeta de tiempo).
4. Una vez que el empleado haya registrado la información, el sistema guarda los cambios de la tarjeta de tiempo.

TESIS CON
FALLA DE ORIGEN

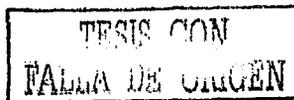
2.2.1.1. Confirmar tarjeta de tiempo

1. En cualquier momento, el empleado puede solicitar que el sistema confirme la tarjeta de tiempo.
2. En ese momento el sistema asigna la fecha actual a la tarjeta de tiempo como fecha de confirmación y cambia el estado de la tarjeta de tiempo a confirmada.
3. El sistema valida la tarjeta de tiempo revisando el número de horas trabajadas contra cada número de factura. El total de las horas contra todos los números de factura no debe exceder ningún límite establecido para el empleado (por ejemplo el empleado no puede trabajar horas extras).
4. El sistema retiene el número de horas trabajadas para cada número de factura en la tarjeta de tiempo.
5. El sistema guarda la tarjeta de tiempo.
6. El sistema hace a la tarjeta de tiempo de solo lectura, es decir, a partir de ese momento, el empleado no puede realizar ningún cambio a la tarjeta de tiempo.

2.2.2. Flujos Alternativos

2.2.2.1. Número de horas inválido

Si en el flujo básico, se registra un número de horas inválido para un solo día (>24) o el número excede el máximo permitido para un empleado, el sistema desplegará un mensaje de error y solicitará ingresar un número de horas válido. El empleado deberá registrar el número válido o cancelar la operación (en este caso termina el caso de uso).



2.2.2.2. Tarjeta de tiempo ya confirmada

Si en el flujo básico, la tarjeta de tiempo actual del empleado ya se había confirmado, el sistema desplegará una copia de solo lectura de la tarjeta de tiempo e informará al empleado que la tarjeta de tiempo ya ha sido confirmada y no se le puede realizar ningún cambio. El empleado acepta el mensaje y el caso de uso termina.

2.2.2.3. Base de datos (administración de proyectos) no disponible

Si en el flujo básico, no se puede realizar la conexión a la base de datos heredada, el sistema desplegará un mensaje de error afirmando que la lista de los números de factura no esta disponible. El empleado acepta el mensaje de error y decide si continúa (sin los números de factura elegibles) o cancela la operación (en este caso cualquier cambio a la tarjeta de tiempo se descarta y el caso de uso termina).

Nota: Sin los números de factura elegibles, el empleado puede cambiar las horas de los números de factura listados en la tarjeta de tiempo, pero no puede agregar horas a algún número de factura que no este listado en la tarjeta de tiempo.

2.3. Requerimientos Especiales

Ninguno.

2.4. Pre - Condiciones

El empleado debe conectarse al sistema antes de que el caso de uso se inicie.

2.5. Post - Condiciones

Si el caso de uso termina con éxito, la tarjeta de tiempo del empleado se guardará en el sistema, de lo contrario, el estado del sistema no cambia.

2.6. Puntos de Extensión

Ninguno.

TESIS CON
FALLA DE ORIGEN

Análisis del Caso de Uso: Mantener la tarjeta de tiempo.

Enfocándonos en el *diagrama de casos de uso del sistema de nómina* (Figura 2.3-9), en el *caso de uso*: mantener la tarjeta de tiempo; obtenemos:



Figura 2.3-9

1) Comenzamos con la identificación de clases.

a) De tipo frontera:



Figura 2.3-10

b) De tipo entidad:

Para este *caso de uso* se tienen las siguientes:



Figura 2.3-11

- c) De tipo control:



Figura 2.3-12

- 2) Distribuir el comportamiento.

- a) Asignación de responsabilidades:

En este punto es posible encontrar asignaciones que ya habíamos hecho en el análisis del *caso de uso: procesar nómina*; sin embargo, no es necesario ahondar en el tema, ya que más adelante se simplificarán al unificar las *clases de análisis*.

| Clase de análisis | Tipo de clase de análisis | Responsabilidades |
|----------------------------|---------------------------|---|
| FormaTarjetaTiempo | Frontera | Mantenimiento a la tarjeta de tiempo desplegar la tarjeta de tiempo Registrar horas a los números de factura Guardar cambios a la tarjeta de tiempo |
| ControladorTarjetaTiempo | Control | Obtener tarjeta de tiempo correspondiente Obtener números de factura Actualizar los cambios de la tarjeta de tiempo Guardar cambios a la tarjeta de tiempo |
| Empleado | Entidad | Obtener tarjeta de tiempo correspondiente |
| TarjetaTiempo | Entidad | Obtener la información de la tarjeta de tiempo Actualizar los cambios de la tarjeta de tiempo Guardar cambios a la tarjeta de tiempo |
| BD_AdministraciónProyectos | Frontera | Obtener números de factura |

TESIS CON
FALLA DE ORIGEN

b) Modelado del comportamiento:

Para este caso de uso utilizaremos un *diagrama de secuencia* para modelar el comportamiento. También es válido utilizar un *diagrama de colaboración*.

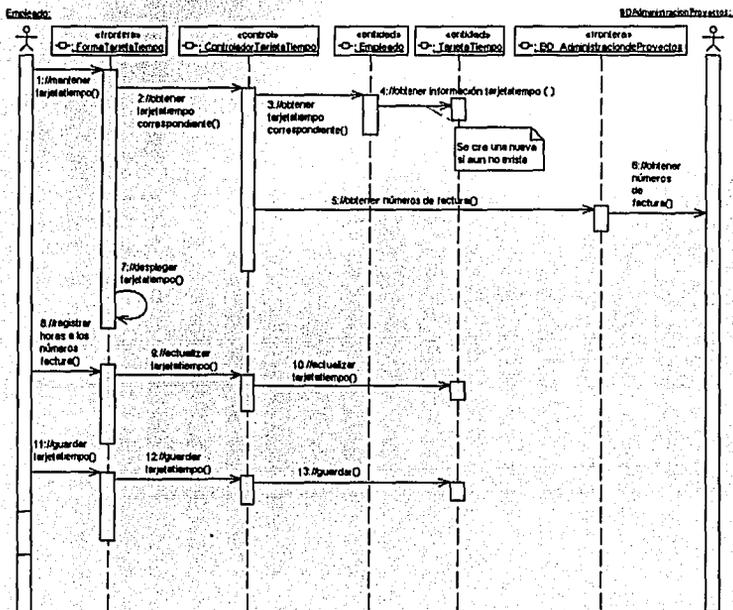


Figura 2.3-13

TESIS CON FALLA DE ORIGEN

3) Diagrama de clases participantes.

Para este caso de uso presentaremos un solo diagrama de clases participantes el cual contiene atributos, operaciones y asociaciones.

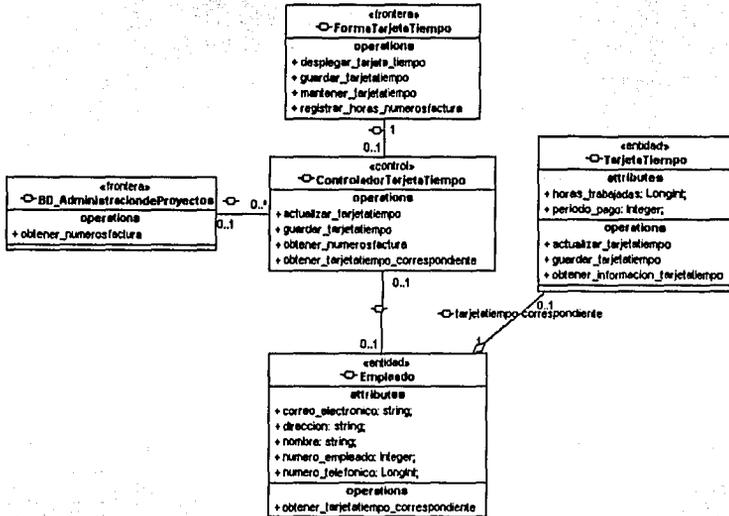


Figura 2.3-14

Como podemos observar la clase TarjetaTiempo en este diagrama aparece con más responsabilidades y atributos que en el diagrama de clases participantes del caso de uso: *procesar nómina*. Lo expuesto anteriormente es correcto ya que cada caso de uso nos puede ir aportando nuevas responsabilidades y atributos a las clases de análisis. Cabe mencionar, que no por tener más atributos o responsabilidades en un diagrama que en otro, estamos hablando de clases diferentes.

TESIS CON
FALLA DE ORIGEN

4) Describir mecanismos de análisis.

En este caso encontramos las siguientes relaciones:

| Clases de Análisis | Mecanismos de Análisis |
|------------------------------|-------------------------|
| BD_AdministraciondeProyectos | Interfaz heredada |
| Empleado | Persistencia, Seguridad |
| ControladorTarjetaTiempo | Distribución |
| FormaProcesarTarjetaTiempo | Ninguno |
| TarjetaTiempo | Persistencia |

5) Unificar *clases*.

En este paso lo que tenemos que hacer es fijarnos en las *clases de análisis* que hemos identificado en los dos *casos de uso* y verificar que dos *clases* no definan un mismo concepto. Por ejemplo, en el *diagrama de clases participantes* del *caso de uso procesar nómina* se identificó la *clase :Trabajador* y en el *diagrama de clases participantes* del *caso de uso mantener tarjeta tiempo* se identificó la *clase :Empleado*; a primera vista podríamos pensar que son dos *clases* distintas, sin embargo al verificar el concepto que están representando nos damos cuenta que se pueden unificar y quedarnos con cualquiera de las dos.

Otra tarea que hay que realizar en esta parte, es la unificación de responsabilidades de una misma *clase* que aparece en diferentes *casos de uso*, por ejemplo, a la *clase :TarjetaTiempo* (a cual identificamos para los dos *casos de uso* que hemos analizado) le asignamos la responsabilidad *obtener_información_tarjetatiempo* (identificada en el *caso de uso: procesar nómina*) y las responsabilidades *obtener_información_tarjetatiempo*, *actualizar_tarjetatiempo* y *guardar_tarjetatiempo* (identificada en el *caso de uso: mantener tarjeta de tiempo*); en este caso tenemos que verificar si la responsabilidad *obtener_información_tarjetatiempo* representan la misma responsabilidad para ambos *casos de uso*, de ser así, se deja una sola vez la responsabilidad; en el caso contrario, necesitamos buscar un mejor nombre para alguna de las dos responsabilidades.

Con esto terminamos la ejemplificación de los pasos que hay que realizar en el análisis de un *caso de uso*.

CAPÍTULO 3

ANÁLISIS CON UML DEL SISTEMA: REGISTRO DE REASEGURADORAS EXTRANJERAS

3.1 Introducción

Hoy en día, debido al vertiginoso ritmo de vida con el que vivimos y a la tecnología con la que contamos, le es necesario a cualquier organización o institución, hacer todo lo posible por procesar los insumos que recibe, con la mejor calidad y con la mayor eficiencia.

En específico mencionaremos a la Comisión Nacional de Seguros y Fianzas (CNSF). Dicha comisión, se encarga de supervisar que la operación de los sectores asegurador y afianzador se apege al marco normativo; preservando la solvencia y estabilidad financiera de las instituciones con el fin de garantizar los intereses del público usuario.

La CNSF cuenta con la DIRECCION GENERAL DE SUPERVISION DE REASEGURO (DGSR) la cual se encarga —entre otras funciones— de emitir opinión sobre la inscripción en el Registro General de Reaseguradoras Extranjeras (RGRE), para tomar reaseguro y reafianzamiento del país, así como el establecimiento de oficinas de representación de reaseguradoras extranjeras a cargo de la Secretaría de Hacienda y Crédito Público (SHCP).

Una de las funciones que el Supervisor de Reaseguro lleva a cabo, es dar seguimiento a la información relacionada con la inscripción, renovación, cancelación, cambios y con las oficinas de representación de las reaseguradoras extranjeras que solicitan inscripción en el RGRE.

En términos generales los procesos que el supervisor de reaseguro desempeña para el seguimiento de información, consisten en capturar en Excel la información de los oficios que se elaboran en diferentes etapas y posteriormente poder consultarla y darla a conocer al sector asegurador. Esta forma de mantener la información presenta inconvenientes relacionados con seguridad y con funcionalidad, por ejemplo, toda la información se tiene en varios archivos de Excel, si por algún motivo el disco duro de la computadora donde se tienen estos archivos sufre algún daño o simplemente si por error se borran los archivos, no tendríamos forma de recuperarla; es un tanto laborioso el estar filtrando la información en Excel según el tipo de consulta que se quiere y después pegarla en otro archivo para imprimirla. Estas son algunas de las causas por las que surge la necesidad de desarrollar un sistema para el RGRE.

3.2 GLOSARIO

1. *Introducción*

Este documento se utiliza para definir la terminología específica en el dominio del sistema, explicando aquellos términos que pueden ser desconocidos para los lectores de los casos de uso o de los documentos del proyecto.

2. *Definiciones*

Agencia Calificadora

Entidad de conocido prestigio internacional que se encarga de evaluar que las reaseguradoras acrediten su solvencia y estabilidad para efectuar operaciones de reaseguro y reafianzamiento en México.

Intermediario

Persona física o moral que, en su calidad de agente, relaciona una institución de seguros con el público usuario de los servicios que ésta brinda; la relación se perfecciona a través de una solicitud que el asegurado potencial firma y que es entregada a la institución para el trámite correspondiente; el mismo término se usa cuando se trata de relacionar una institución de seguros con un reasegurador, para proporcionar a la misma los servicios que requiere.

Oficinas de Representación

Entidad que se establece en el territorio nacional y que sólo actúa a nombre y por cuenta de sus representadas (reaseguradoras) para aceptar o ceder riesgos en reaseguro o responsabilidades en reafianzamiento.

Pool

Unión de varias instituciones de seguros para operar, según normas comunes establecidas, un cierto ramo de seguros con el fin de que durante un ciclo operativo, estén expuestas a suertes similares y, al cierre del mismo, compartan resultados (siniestros, utilidades,

pérdidas) según participación de cada institución al pool. Varias instituciones de seguros pueden integrar un pool de reaseguro formando una compañía de reaseguro.

Pool Atómico

Mecanismo que surge de convenios internacionales y que tiene como propósito la dispersión de riesgos nucleares, con particularidades que lo distinguen de otros procesos de reaseguro relativos a riesgos tradicionales.

Proponente

Es el que propone a la reaseguradora para inscribirla en el RGRE. El proponente puede ser la misma reaseguradora, una compañía de seguros o un intermediario.

Reafianzamiento

El reafianzamiento es la fianza mediante la cual una institución se obliga a pagar a otra, en proporción correspondiente, las cantidades que ésta debe cubrir al beneficiario por fianza.

Reaseguradora Extranjera

Entidad del exterior que desea celebrar contratos de reaseguro y reafianzamiento con alguna entidad mexicana de seguros o de fianzas.

Reaseguro

Es la transferencia de riesgos por parte de una compañía aseguradora, llamada cedente, a otra denominada reasegurador, mediante el pago de una prima.

Subíndice

Es una clasificación adicional que se le da a las calificaciones otorgadas.

3.3 SITUACIÓN ACTUAL

Se requiere de un sistema que ayude a dar mantenimiento a las inscripciones, renovaciones, (considerando la parte de pool atómico) cancelaciones y cambios de reaseguradoras así como a las autorizaciones, clausuras, renovaciones y cambios de las oficinas de representación. El sistema también debe permitir llevar a cabo el monitoreo de las calificaciones de las reaseguradoras e imprimir reportes de consultas específicas.

Actualmente el mantenimiento de información se lleva a cabo en hojas de Excel. A continuación se describe como se llevan a cabo de forma manual los procesos que el sistema debe considerar.

REASEGURADORAS

Inscripciones

El proceso de inscripción inicia con un oficio que emite la SHCP a la CNSF, del cual se capturan los siguientes datos:

Nombre Reaseguradora, País, Estado, Domicilio, Teléfono, Fax, Email, Proponente, Número Oficio Solicitud, Fecha Oficio Solicitud.

La CNSF busca en Internet los siguientes datos de la reaseguradora: *Agencia Calificadora, Calificación, Subindice y Fecha Calificación.* Cuando las reaseguradoras se inscriben en un pool atómico con regla 18, no se capturan dichos campos. Si la información se encuentra en la página de la Agencia Calificadora A.M. Best, también se captura el *número de control A.M. Best.*

Una reaseguradora puede ser calificada por más de una *Agencia Calificadora.* Para cada *Agencia* se captura *Calificación, subindice y Fecha de calificación.*

Cuando las reaseguradoras se inscriben en un pool atómico, se capturan los siguientes datos:

TESIS CON
FALLA DE ORIGEN

Nombre y ciudad del pool, fecha de alta, además se identifica si tiene *cláusula solidaridad* y/o *cláusula participación (mínimo debe tener una)* y dependiendo de la regla con la que se inscribió se captura lo siguiente:

- a) Regla 18.- Se captura la reaseguradora siempre como *integrante* del pool,
- b) Regla 3.-Se captura si se inscribió como *líder* y/o *administrador*.

Después la CNSF elabora un oficio dirigido a la SHCP en el cual se expresa su opinión (favorable o desfavorable) de la inscripción. De este oficio se captura el *resultado de opinión, número y fecha del oficio*.

La CNSF revisa que la calificación con la que se quiere inscribir dicha reaseguradora, este dentro de las mínimas permitidas. De no ser así, será causa de que el resultado de opinión sea desfavorable.

Enseguida la SHCP revisa el oficio de la CNSF y puede coincidir o no con la opinión de la CNSF. Si la SHCP decide autorizar la inscripción, emite un oficio de autorización a la CNSF el cual contiene el *Número de registro* con el que se identificará a la reaseguradora, caso contrario, se suspende el proceso de inscripción.

En el momento que la CNSF recibe el oficio de autorización, se captura: *Número de registro, Número Oficio SHCP, Fecha Oficio SHCP y Fecha de inscripción*. Además la CNSF emite un oficio circular con el cual se da a conocer al sector la nueva reaseguradora. De este oficio circular se captura el *Número y Fecha de Oficio*.

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de inscripción.

Posteriormente se captura la *Fecha de publicación* con que se da de alta en el archivo que se publica en la página Web la información de la reaseguradora.

Renovaciones

Este proceso se lleva a cabo cada año y se inicia de la siguiente manera:

La SHCP emite un listado a la CNSF con las reaseguradoras que solicitaron renovación. De dicho listado se captura: *proponente de renovación*, el *año de renovación* que se está solicitando la renovación, la *Agencia Calificadora*, *Calificación*, *Subíndice*, *Fecha calificación*, *Fecha escrito renovación* que traen para renovarse.

En algunos casos el listado puede traer hasta 4 proponentes con una o más Agencias Calificadoras, Calificaciones, Subíndices y Fechas de Calificación.

Si al solicitar la renovación una reaseguradora:

- a) Sigue con el mismo pool que se dio de alta, entonces no se registra nada.
- b) Cambio alguna de las características del pool, entonces se da de baja (*Fecha Baja*) el pool con que se dio de alta y se registra la alta del nuevo pool.
- c) Se registro en otro pool diferente al que ya pertenecía, entonces se deja el pool con el que se dio de alta y se registran los datos correspondientes al nuevo pool.

La CNSF emite un oficio circular con el cual se da a conocer la renovación. De este oficio circular se captura el *Número* y *Fecha de Oficio circular*.

Todas las reaseguradoras que no soliciten renovación, y todas aquellas que soliciten ya no ser renovadas, se marcan como cancelaciones.

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de renovación.

TESIS CON
FALLA DE ORIGEN

Cancelaciones

En el momento que la CNSF detecte alguna anomalía, en cualquiera de los requisitos que deben cumplir las reaseguradoras, se encarga de emitir un oficio de opinión a la SHCP en el cual explica los motivos por los que considera cancelar el registro de la reaseguradora. La CNSF captura el *número y fecha del oficio de opinión* y el *sustento del oficio*.

La SHCP se encarga de avisar a la reaseguradora y en caso de considerarlo pertinente, la SHCP emite un oficio de prórroga el cual envía a la CNSF. De este oficio se registra el *número y fecha del oficio de prórroga*.

Si en este plazo la reaseguradora no corrige sus anomalías, la SHCP da su baja definitiva y le informa a la CNSF mediante el oficio de baja del cual se captura el *número y fecha del oficio de baja y la fecha de baja*.

La CNSF emite un oficio circular con el cual se da a conocer al sector la baja de la reaseguradora en cuestión. De este oficio circular se captura el *Número y Fecha de Oficio circular*.

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de cancelaciones.

Posteriormente se captura la *Fecha* con que se da de baja en el archivo que se publica en la página Web la información de la reaseguradora.

Cambios

El proceso de cambios inicia con un oficio que emite la SHCP a la CNSF en el cual se mencionan los cambios que ha tenido la reaseguradora en cuestión, como pueden ser, cambios en: *razón social, domicilio, teléfono, fax, email, estado, país*.

TESIS CON
FALLA DE ORIGEN

La CNSF captura del oficio: *Número y Fecha de Oficio, Tipo(s) de cambio(s), Descripción(es) del(os) cambio(s) y la fecha de cambio.*

Los cambios en el caso de pool atómico se manejan a nivel general, por ejemplo, la reaseguradora "X" pertenecía al pool "Y" con características "xy" y a partir de cierta fecha pertenece al pool "Z" con características "zy".

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de cambios.

Posteriormente se captura la *Fecha* con que se actualiza en el archivo que se publica en la página Web la información de la reaseguradora

Monitoreo

Consiste en monitorear cada mes las calificaciones que dan las distintas Agencias Calificadoras a las reaseguradoras. En este proceso los datos a capturar son: *fecha de monitoreo, Agencia Calificadora, calificación, subíndice y las observaciones* que se pudieran tener.

La CNSF arma un archivo el cual contiene: *nombre reaseguradora, ciudad pool, estado y país de la reaseguradora, país reaseguradora, número de registro; la calificación de la Agencia Calificadora A.M. Best, la calificación de la Agencia Calificadora Standard and Poor's y la calificación de alguna otra Agencia Calificadora a un mes especificado* de todas las reaseguradoras vigentes. Este archivo se publica mensualmente en la página Web de la comisión y refleja los últimos cambios que hayan tenido las reaseguradoras.

OFICINAS DE REPRESENTACIÓN

Autorizaciones

El proceso de autorización inicia con un oficio que emite la SHCP a la CNSF, del cual se capturan los siguientes datos: *Nombre Oficina Representación, Número de Registro de la Reaseguradora, Representante Legal, Puesto del Representante Legal, Director General, Domicilio, Teléfono, Fax, Mail, Oficio Solicitud, Fecha Oficio Solicitud.*

La Dirección de Reaseguro manda un memo a la Dirección de Jurídico del cual se captura el *número y fecha del memo*. Posteriormente Jurídico les responde con un nuevo memo del cual se captura el *Número y fecha del memo respuesta.*

Una vez que la CNSF revisa la solicitud y la documentación enviada, verifica si satisface los requisitos y procede a elaborar un oficio de opinión el cual se envía a la SHCP.

La CNSF guarda el *resultado de opinión, número y fecha del oficio.*

Enseguida la SHCP revisa el oficio de la CNSF y puede o no coincidir con la opinión de la CNSF.

Si después la SHCP otorga la autorización, notifica a la CNSF mediante un oficio del cual se capturan los siguientes datos: *fecha de inicio de operación, Número Oficio SHCP, Fecha Oficio SHCP*. Además la CNSF emite un oficio circular con el cual se da a conocer al sector la nueva oficina de representación. De este oficio circular se captura el *Número y Fecha de Oficio*.

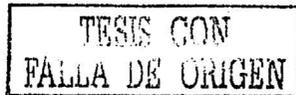
La CNSF arma un archivo el cual contiene: *nombre y país de la oficina de representación, número de registro, representante legal, dirección y teléfono* de todas las oficinas vigentes. Este archivo se publica mensualmente en la página Web de la comisión y refleja los últimos cambios que hayan tenido las oficinas de representación. La CNSF captura la *Fecha* en que se agregó una oficina de representación a dicho archivo.

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de autorización.

Una reaseguradora puede tener una o más oficinas de representación.

Clausuras

Existen dos formas de iniciar el proceso para las clausuras:



- 1) Si la SHCP emite un oficio de clausura a la CNSF, se capturan el *número y fecha del oficio de clausura*.
- 2) Si la CNSF detecta alguna anomalía en cualquiera de los requisitos que deben cumplir las oficinas de representación, emite un oficio de opinión a la SHCP en el cual explica los motivos por los que considera que se debe clausurar la oficina de representación. De este oficio se captura el *número y fecha del oficio de opinión*. Posteriormente la SHCP contesta con un oficio de clausura del cual se captura el *número y fecha del oficio de clausura y la fecha de clausura*.

Después la CNSF emite un oficio circular con el cual se da a conocer al sector la clausura de la oficina de representación en cuestión. De este oficio circular se captura el *Número y Fecha de Oficio circular*.

Posteriormente se captura la *Fecha* con que se da de baja en el archivo que se publica en la página Web la información de la oficina de representación

Existe un campo de *Observaciones* en el cual se capturan los detalles que se consideren pertinentes de cualquier etapa del proceso de clausura.

Cambios

El proceso de cambios inicia con un oficio que emite la SHCP a la CNSF en el cual se mencionan los cambios que ha tenido la oficina de representación en cuestión, como pueden ser, cambios en: *representante legal, puesto del representante, domicilio, teléfono, fax y el email*.

La CNSF captura del oficio: *Número y Fecha de Oficio, Tipo(s) de Cambio(s), Descripción(es) del(os) cambio(s) y la fecha de cambio*.

Posteriormente se captura la *Fecha* con que se actualiza en el archivo que se publica en la página Web la información de la oficina de representación

Los Administradores de Reaseguro son los encargados de darle mantenimiento a toda la información.

Consulta de Información

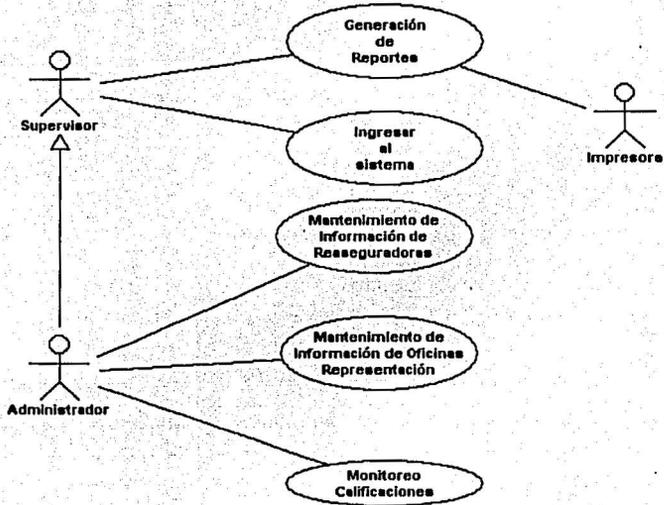
Este proceso consiste en filtrar la información de una o más hojas de Excel (según el tipo de consulta) e imprimirla si se desea. Los Supervisores y Administradores de Reaseguro pueden realizar las siguientes consultas:

TESIS CON
FALLA DE ORIGEN

1. Calificaciones por número de registro y periodo
2. Inscripciones de reaseguradoras por periodo
3. Renovaciones de reaseguradoras por año de renovación
4. Cancelaciones de reaseguradoras por periodo
5. Cambios de reaseguradoras por número de registro y periodo
6. Pooles por reaseguradora y por periodo
7. Autorizaciones de oficinas de representación por periodo
8. Clausuras de oficinas de representación por periodo
9. Cambios de oficinas de representación por periodo

TESIS CON
FALLA DE ORIGEN

3.4 MODELO DE CASOS DE USO



TESIS CON
FALLA DE ORIGEN

3.5 CASOS DE USO

1) Caso de Uso : Ingresar al sistema

1.1. Breve Descripción

Este caso de uso describe la forma en que cualquier actor ingresa al Sistema de Registro General de Reaseguradoras Extranjeras (SRGRE).

1.2. Flujo de Eventos

1.2.1. *Flujo Básico*

Este caso de uso inicia cuando el actor desea ingresar al SRGRE.

1. El sistema solicita que el actor ingrese su nombre y password.
2. El actor registra su nombre y password.
3. El sistema valida el nombre y password capturados e ingresa al actor al sistema.

1.2.2. *Flujos Alternativos*

1.2.2.1. Nombre/Password Inválidos

Si en el *flujo básico* el actor ingresa un nombre/password inválido, el sistema despliega un mensaje de error. El actor puede escoger entre regresar al principio del *flujo básico* o cancelar la entrada al sistema, en este punto el caso de uso termina.

1.3. Requerimientos Especiales

Ninguno.

1.4. Pre - Condiciones

Ninguna.

1.5. Post – Condiciones

Si el caso de uso fue exitoso, el actor es conectado al sistema. Si no, el sistema permanece sin cambios.

1.6. Puntos de Extensión

Ninguno.

2) Caso de Uso: Mantenimiento de Información de Reaseguradoras

2.1. Breve Descripción

Este caso de uso permite al Administrador, dar mantenimiento a la información de las Reaseguradoras. Esto contempla aumentar, renovar, modificar y eliminar reaseguradoras.

2.2. Flujo de Eventos

2.2.1. Flujo Básico

Este caso de uso se inicia cuando el Administrador quiera agregar, modificar, renovar o eliminar la información de alguna reaseguradora.

1. El sistema solicita al Administrador que indique el proceso que quiere realizar (inscripción, renovación, cancelación, o algún cambio de la reaseguradora).
2. Una vez que el Administrador selecciona el proceso, uno de los siguientes sub-flujos se ejecuta:

Si el Administrador selecciona "Inscripciones" el sub-flujo **Inscribir Reaseguradora** se ejecutará.

Si el Administrador selecciona "Renovaciones" el sub-flujo **Renovaciones** se ejecutará.

Si el Administrador selecciona "Cancelaciones" el sub-flujo **Cancelar Reaseguradora** se ejecutará.

Si el Administrador selecciona "Cambios" el sub-flujo **Cambios en Reaseguradora** se ejecutará.

2.2.1.1. Inscribir Reaseguradora

1. El sistema solicita al Administrador el nombre de la reaseguradora.
2. El sistema recupera la información asociada a la reaseguradora y permite al Administrador agregar o modificar los siguientes datos:
 - País, Estado, Domicilio, Teléfono, Fax, Email
 - Proponente(s) (al menos 1 máximo 4)
 - Número Oficio Solicitud, Fecha Oficio Solicitud.
 - Agencia(s) Calificadora(s) (al menos 1 máximo 4) (aplica para inscripciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Calificación(es) (al menos 1 máximo 4) (aplica para inscripciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Subíndice(s) (al menos 1 máximo 4) (aplica para inscripciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Fecha(s) de Calificación(s) (al menos 1 máximo 4) (aplica para inscripciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Clave de inscripción (aplica para inscripción con riesgo atómico)
 - Nombre del pool (aplica para inscripción con riesgo atómico)
 - País del pool (aplica para inscripción con riesgo atómico)
 - Participación en pool (aplica para inscripción con riesgo atómico)
 - Fecha de Alta en pool (aplica para inscripción con riesgo atómico)
 - Cláusula solidaridad y/o Cláusula participación (aplica para inscripción con riesgo atómico y mínimo se debe registrar una)
 - Resultado de opinión, Número y Fecha del oficio de opinión.
 - Número de registro
 - Número y fecha del Oficio de la SHCP
 - Fecha de Inscripción
 - Número y fecha del Oficio Circular
 - Observaciones
 - Fecha de Publicación en la Web

3. Una vez que el Administrador ingresó y/o actualizó la información necesaria, el sistema guarda la información asociándola al identificador de la reaseguradora (ID).

2.2.1.2. Renovaciones

Este sub-flujo se lleva acabo cada año.

El sistema solicita al Administrador ingresar: Año de renovación

1. El sistema despliega el número de registro de todas las reaseguradoras vigentes al año de renovación ingresado.
2. El Administrador revisará uno por uno los número de registro de las reaseguradoras y marcará aquellos correspondientes a las reaseguradoras que se van a renovar.
3. El sistema desplegará y permitirá al Administrador ingresar—para cada uno de los números de registro marcados— los siguientes datos:
 - Proponente(s) de Renovación(es) (al menos 1 máximo 4)
 - Año de renovación
 - Fecha(s) Escrito de Renovación(es) (al menos 1 máximo 4)
 - Agencia(s) Calificadora(s) (al menos 1 máximo 4) (para renovaciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Calificación(es) (al menos 1 máximo 4) (para renovaciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Subíndice(s) (al menos 1 máximo 4) (para renovaciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Fecha(s) de Calificación(s) (al menos 1 máximo 4) (para renovaciones con riesgos no atómicos o riesgo atómico con regla 3)
 - Clave de inscripción (para inscripción con riesgo atómico)
 - Nombre del pool (para inscripción con riesgo atómico)

- País del pool (para inscripción con riesgo atómico)
 - Participación en pool (para inscripción con riesgo atómico)
 - Fecha de Alta en pool (para inscripción con riesgo atómico)
 - Fecha de Baja en pool (para inscripción con riesgo atómico)
 - Cláusula solidaridad y/o Cláusula participación (para inscripción con riesgo atómico y mínimo se debe registrar una)
 - Número y fecha del Oficio Circular
 - Observaciones
4. El sistema registrará las cancelaciones de las reaseguradoras cuyos números de registro no fueron marcados para ser renovados
 5. Una vez que el Administrador ingreso y/o actualizó la información necesaria, el sistema guarda la información asociándola al ID de la reaseguradora.

2.2.1.3.Cancelar Reaseguradora

1. El sistema solicita al Administrador ingresar el número de registro de la reaseguradora que se quiere cancelar.
2. El sistema despliega y permite al Administrador ingresar los siguientes datos:
 - Número y fecha del Oficio Opinión
 - Sustento del Oficio de Opinión
 - Número y fecha del Oficio de Prórroga
 - Número y Fecha del Oficio de Baja
 - Fecha de cancelación
 - Número y Fecha del Oficio Circular
 - Fecha de baja en la Web
 - Observaciones

3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al ID de la reaseguradora.

2.2.1.4. Cambios en Reaseguradora

1. El sistema solicita al Administrador ingresar el número de registro de la reaseguradora que se quiere cambiar.
2. El sistema despliega y permite al Administrador ingresar los siguientes datos:
 - Número y Fecha del Oficio de Cambios
 - Tipo(s) de cambio(s)
 - Descripción(es) del cambio(s) (es decir los nuevos valores de los campos que cambiaron)
 - Fecha de Cambio
 - Fecha Actualización en la Web
 - Observaciones
3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al ID de la reaseguradora

2.2.2. Flujos Alternos

2.2.2.1. Existencia de Reaseguradora

Quando el Administrador ingresa el nombre de la reaseguradora, en el sub-flujo **Inscribir Reaseguradora**, el sistema verifica si ya existe dicha reaseguradora. Si no existe, el sistema genera y asigna un número único de identificación (ID) a la reaseguradora.

2.2.2.2. Número de Registro no encontrado

Si, en cualquiera de los dos sub-flujos, **Cancelar Reaseguradora** o **Cambios Reaseguradora**, el número de registro ingresado no existe, el sistema muestra un mensaje de error. El Administrador puede ingresar un número de registro diferente o cancelar la operación con lo que se terminaría el caso de uso.

2.2.2.3. Características del pool sin ningún cambio

Si, en el sub-flujo **Renovaciones** el Administrador identifica que las características del pool de renovación son iguales a las del "pool" del año anterior, el Administrador no registra ningún campo correspondiente a riesgo atómico.

2.2.2.4. Características del pool con cambios

Si, en el sub-flujo **Renovaciones** el Administrador identifica que alguna característica del pool de renovación difiere con alguna característica del pool del año anterior:

1. El Administrador solicita "cancelar" pool del año anterior
2. El sistema solicita al Administrador ingresar una Fecha de Baja para el pool existente
3. El Administrador ingresa los nuevos datos que aplican para riesgo atómico del sub-flujo de **Renovaciones**.
4. El sistema guarda la información asociándola al ID de la reaseguradora.

2.2.2.5. Registrar nuevo pool de Renovación

Si, en el sub-flujo *Renovaciones*, el Administrador desea “registrar” un nuevo pool de renovación:

1. El Administrador solicita “registrar” nuevo pool
2. El sistema solicita los datos que aplican para riesgo atómico del sub-flujo de *Renovaciones*.
3. El sistema guarda la información asociándola al ID de la reaseguradora.

2.2.2.6. Eliminar Cancelación

Si, en el sub-flujo *Renovaciones* o *Cancelar Reaseguradora*, el Administrador desea “eliminar” la cancelación de una reaseguradora:

1. El Administrador solicita “eliminar cancelación”
2. El sistema solicita el número de registro de la reaseguradora
3. El sistema elimina la cancelación de la reaseguradora correspondiente al número de registro ingresado.

2.2.2.7. Renovar reaseguradora cancelada

Si, en el sub-flujo *Renovaciones* o *Cancelar Reaseguradora*, el Administrador desea renovar una reaseguradora que por error no renovó:

1. El Administrador solicita “renovar reaseguradora cancelada”
2. El sistema solicita el número de registro de la reaseguradora
3. El sistema elimina la cancelación de la reaseguradora correspondiente al número de registro ingresado.
4. El sistema despliega y permite capturar los datos correspondientes del sub-flujo *Renovaciones* de la reaseguradora correspondiente al número de registro ingresado.

2.3. Requerimientos Especiales

Ninguno

2.4. Pre – Condiciones

El Administrador se debe conectar al sistema antes de que inicie este caso de uso.

2.5. Post – Condiciones

Si el caso de uso es exitoso, la información de la reaseguradora es actualizada en el sistema. De otro modo el sistema no sufre ningún cambio.

2.6. Puntos de Extensión

Ninguno

3) Caso de Uso: Mantenimiento de Información de Oficinas de Representación

3.1. Breve Descripción

Este caso de uso permite al Administrador, dar mantenimiento a la información de las Oficinas de Representación. Esto contempla aumentar, renovar, modificar y eliminar oficinas de representación.

3.2. Flujo de Eventos

3.2.1. Flujo Básico

Este caso de uso se inicia cuando el Administrador quiera agregar, modificar, renovar o eliminar la información de alguna oficina de representación.

1. El sistema solicita al Administrador que indique el proceso que quiere realizar (autorización, clausuras, o algún cambio de la oficina de representación).
2. Una vez que el Administrador seleccione el proceso, uno de los siguientes sub-flujos se ejecuta:

Si el Administrador selecciona "Autorizaciones" el subflujo **Autorizar Oficina de Representación** se ejecutará.

Si el Administrador selecciona "Clausuras" el subflujo **Clausurar Oficina de Representación** se ejecutará.

Si el Administrador selecciona "Cambios" el sub-flujo **Cambios en Oficinas de Representación** se ejecutará.

3.2.1.1. Autorizar Oficina de Representación

1. El sistema solicita al Administrador el nombre de la Oficina de Representación.
2. El sistema recupera la información asociada a la oficina de representación y permite al Administrador agregar o modificar los siguientes datos:
 - Nombre de la Oficina de Representación
 - Número de Registro de la Reaseguradora
 - Represente Legal
 - Puesto del Representante
 - Director General
 - Domicilio, Teléfono, Fax, Email
 - Número Oficio Solicitud, Fecha Oficio Solicitud
 - Número y Fecha de memo
 - Número y Fecha de memo respuesta
 - Resultado de opinión, Número y Fecha del oficio de opinión
 - Número y Fecha del Oficio de la SHCP
 - Fecha de Inicio de Operación
 - Número y Fecha del Oficio Circular
 - Observaciones
 - Fecha de Publicación en la Web
3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al identificador de la oficina de representación (ID).

3.2.1.2. Clausurar Oficina de Representación

1. El sistema solicita al Administrador ingresar el Nombre de la Oficina de Representación que se quiere clausurar.

2. El sistema despliega y permite al Administrador ingresar los siguientes datos:
 - Número y Fecha de Oficio de Opinión (para el caso de que la CNSF detecte anomalías en los requisitos de las oficinas de representación)
 - Sustento del oficio de opinión (para el caso de que la CNSF detecte anomalías en los requisitos de las oficinas de representación)
 - Número y Fecha del Oficio de Clausura
 - Fecha de Clausura
 - Número y Fecha del Oficio Circular
 - Fecha con que se da de baja en la Web
 - Observaciones
3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al ID de la oficina de representación

3.2.1.3. Cambios en Oficina de Representación

1. El sistema solicita al Administrador ingresar el Nombre de la Oficina de Representación que se quiere cambiar.
2. El sistema despliega y permite al Administrador ingresar los siguientes datos:
 - Número y Fecha del Oficio de Cambios
 - Tipo(s) de cambio(s)
 - Descripción(es) del cambio(s) (es decir los nuevos valores de los campos que cambiaron)
 - Fecha de Cambio
 - Fecha de Actualización en la Web
 - Observaciones

3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al ID de la reaseguradora.

3.2.2. Flujos Alternos

3.2.2.1. Existencia de Oficina de Representación

Quando el Administrador ingresa el nombre de la oficina de representación, en el sub-flujo **Autorizar Oficina de Representación**, el sistema verifica si ya existe dicha oficina de representación. Si no existe, el sistema genera y asigna un número único de identificación (ID) a la oficina de representación.

3.2.2.2. Oficina No Encontrada

Si, en cualquiera de los dos sub-flujos, **Clausurar Oficina de Representación** o **Cambios en Oficina de Representación**, el nombre de la oficina de representación ingresado no existe, el sistema muestra un mensaje de error. El Administrador puede ingresar un nombre de oficina de representación diferente o cancelar la operación con lo que se terminaría el caso de uso.

3.2.2.3. Eliminar Clausura

Si, en el sub-flujo **Clausurar Oficina de Representación**, el Administrador desea "eliminar" la clausura de una oficina de representación:

1. El Administrador solicitará "eliminar clausura"
2. El sistema solicitará el nombre de la oficina de representación que se quiera clausurar
3. El sistema eliminará la clausura de la oficina de representación.

3.3. Requerimientos Especiales

Ninguno

3.4. Pre – Condiciones

El Administrador se debe conectar al sistema antes de que inicie este caso de uso.

3.5. Post – Condiciones

Si el caso de uso es exitoso, la información de la oficina de representación es actualizada en el sistema. De otro modo el sistema no sufre ningún cambio.

3.6. Puntos de Extensión

Ninguno

4) Caso de Uso: Monitoreo de Calificaciones

4.1. Breve Descripción

Este caso de uso permite al Administrador, dar seguimiento a las calificaciones de las Reaseguradoras.

4.2. Flujo de Eventos

4.2.1. Flujo Básico

Este caso de uso se inicia cuando el Administrador necesite agregar, la información mensual de las características de las calificaciones de cada reaseguradora vigente.

1. El sistema solicita al Administrador ingresar el número de Registro de la Reaseguradora que se quiere monitorear.
2. El sistema despliega y permite al Administrador ingresar los siguientes datos:
 - Fecha de Monitoreo
 - Agencia(s) Calificadora(s) (al menos 1 máximo 4)
 - Calificación(es) (al menos 1 máximo 4)
 - Subíndice(s) (al menos 1 máximo 4)
 - Observaciones
3. Una vez que el Administrador ingreso y/o actualizo la información necesaria, el sistema guarda la información asociándola al ID de la reaseguradora.

4.2.2. Flujos Alternos

4.2.2.1. Existencia de Reaseguradora

Si en el flujo básico el número de registro ingresado no existe, el sistema muestra un mensaje de error. El Administrador puede ingresar un número de registro diferente o cancelar la operación con lo que se terminaría el caso de uso.

4.3. Requerimientos Especiales

Ninguno

4.4. Pre – Condiciones

El Administrador se debe conectar al sistema antes de que inicie este caso de uso.

4.5. Post – Condiciones

Si el caso de uso es exitoso, la información del monitoreo es actualizada en el sistema. De otro modo el sistema no sufre ningún cambio.

4.6. Puntos de Extensión

Ninguno

5) Caso de Uso: Generación de Reportes

5.1. Breve Descripción

Este caso de uso permite al Supervisor, consultar la información de reaseguradoras, oficinas de representación y del monitoreo de calificaciones. El Supervisor tendrá la opción de imprimir las consultas que realice.

5.2. Flujo de Eventos

5.2.1. Flujo Básico

Este caso de uso se inicia cuando el Supervisor necesite consultar la información de reaseguradoras, oficinas de representación o del monitoreo de calificaciones.

1. El Supervisor indica al sistema el tipo de consulta que quiere realizar:
 - 1.1. Consulta de Reaseguradoras
 - 1.2. Consulta de Oficinas de Representación
2. Si el Supervisor elige Consulta de Reaseguradoras; el Supervisor podrá elegir entre las siguientes consultas:
 - Si el Supervisor elige Consulta de Calificaciones de Reaseguradoras, el sub-flujo: **Calificaciones por número de registro y periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Inscripciones de Reaseguradoras, el sub-flujo: **Inscripciones de Reaseguradoras por periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Renovaciones de Reaseguradoras, el sub-flujo: **Renovaciones de Reaseguradoras por año de renovación** se ejecutará.

- Si el Supervisor elige Consulta de Cancelaciones de Reaseguradoras, el sub-flujo: **Cancelaciones de Reaseguradora por periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Cambios de Reaseguradoras, el sub-flujo: **Cambios de Reaseguradora por número de registro y periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Pooles por Reaseguradora, el sub-flujo: **Pooles por Reaseguradora por periodo** se ejecutará.
3. Si el Supervisor elige Consulta de Oficinas de Representación, el Supervisor podrá elegir entre las siguientes consultas:
- Si el Supervisor elige Consulta de Autorizaciones de Oficinas de Representación, el sub-flujo: **Autorizaciones de Oficinas de Representación por periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Clausuras de Oficinas de Representación, el sub-flujo: **Clausuras de Oficinas de Representación por periodo** se ejecutará.
 - Si el Supervisor elige Consulta de Cambios de Oficinas de Representación, el sub-flujo: **Cambios de Oficinas de Representación por nombre y periodo** se ejecutará.

5.2.1.1. Calificaciones por numero de registro y periodo

1. El sistema solicita al Supervisor ingresar el número de registro de la Reaseguradora.
2. El sistema solicita al Supervisor ingresar el mes inicial de la consulta.
3. El sistema solicita al Supervisor ingresar el mes final de la consulta.
4. El sistema despliega para cada mes de consulta, la siguiente información:
 - Nombre de la Reaseguradora
 - Número de Registro
 - Agencia(s) Calificadora(s) (al menos 1 máximo 4)
 - Calificación(es) (al menos 1 máximo 4)

- Subíndice(s) (al menos 1 máximo 4)
- Observaciones

5. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.2. Incripciones de Reaseguradoras por periodo

1. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
2. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
3. El sistema despliega para cada una de las reaseguradoras que se inscribieron dentro del periodo solicitado, la siguiente información:
 - Nombre de la Reaseguradora
 - Número de registro
 - Fecha de Inscripción
 - País, Estado, Ciudad, Domicilio
 - Proponente(s) (al menos 1 máximo 4)
 - Observaciones
4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.3. Renovaciones de Reaseguradoras por año de renovación

1. El sistema solicita al Supervisor ingresar el año de renovación.
2. El sistema despliega para cada una de las reaseguradoras que se renovaron en el año de renovación solicitado, la siguiente información:
 - Nombre de la Reaseguradora
 - Número de registro
 - Año de renovación
 - País, Estado, Ciudad, Domicilio
 - Proponente(s) (al menos 1 máximo 4)

-Observaciones

3. El Supervisor puede elegir la opción de imprimir la consulta realizada

5.2.1.4. Cancelaciones de Reaseguradora por periodo

1. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
2. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
3. El sistema despliega para cada una de las reaseguradoras que se cancelaron dentro del periodo solicitado, la siguiente información:
 - Nombre de la Reaseguradora
 - Número de registro
 - Fecha de Cancelación
 - País, Estado, Ciudad, Domicilio
 - Observaciones
4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.5. Cambios de Reaseguradora por nombre y periodo

1. El sistema solicita al Supervisor ingresar el número de registro de la Reaseguradora.
2. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
3. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
4. El sistema despliega para cada uno de los cambios efectuados dentro del periodo solicitado, la siguiente información:
 - Nombre de la Reaseguradora
 - Número de registro
 - Tipo de cambio

- Fecha del cambio
- Descripción anterior del dato que cambio
- Descripción posterior del dato que cambio
- Observaciones

4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.6. Pools por Reaseguradora por periodo

1. El sistema solicita al Supervisor ingresar el número de registro de la Reaseguradora.
2. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
3. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
4. Para cada Pool encontrado de la reaseguradora, en el periodo solicitado, el sistema desplegará la siguiente información :
 - Nombre Reaseguradora
 - Número de Registro
 - Nombre del Pool
 - País del pool
 - Participación en pool
 - Fecha de Alta en pool
 - Fecha de Baja en Pool
 - Cláusula solidaridad y/o Cláusula participación
 - Observaciones
5. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.7. Autorizaciones de Oficinas de Representación por periodo

1. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
2. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
3. El sistema despliega para cada una de las oficinas de representación que se autorizaron dentro del periodo solicitado, la siguiente información:
 - Nombre de la Oficina de Representación
 - Representante Legal
 - Nombre de la Reaseguradora
 - Fecha de Inicio de Operación
 - Observaciones
4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.8. Clausuras de Oficinas de Representación por periodo

1. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
2. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
3. El sistema despliega para cada una de las oficinas de representación que se clausuraron dentro del periodo solicitado, la siguiente información:
 - Nombre de la Oficina de Representación
 - Representante Legal
 - Nombre de la Reaseguradora
 - Fecha de Clausura
 - Observaciones

4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.1.9. Cambios de Oficinas de Representación por nombre y periodo

1. El sistema solicita al Supervisor ingresar el nombre de la Oficina de Representación
2. El sistema solicita al Supervisor ingresar la fecha inicial de la consulta.
3. El sistema solicita al Supervisor ingresar la fecha final de la consulta.
4. El sistema despliega para cada uno de los cambios, de la oficinas de representación efectuados dentro del periodo solicitado, la siguiente información:
 - Nombre de la Oficina de Representación
 - Representante Legal
 - Nombre de la Reaseguradora
 - Tipo de cambio
 - Fecha de Cambio
 - Descripción anterior del dato que cambio
 - Descripción posterior del dato que cambio
 - Observaciones
4. El Supervisor puede elegir la opción de imprimir la consulta realizada.

5.2.2 Flujos Alternos

5.2.2.1 Imprimir reportes

Si el supervisor elige imprimir la consulta realizada en cualquier sub-flujo, el sistema imprime la consulta en Excel.

5.3. Requerimientos Especiales

Ninguno

5.4. Pre-condiciones

El Supervisor se debe conectar al sistema antes de que inicie este caso de uso.

5.5. Post-condiciones

Ninguna

5.6. Puntos de extensión

Ninguno

TESIS CON
FALLA DE ORIGEN

3.6 ANÁLISIS DE CASOS DE USO

Identificación de clases, asignación y descripción de responsabilidades

A continuación se muestran las clases de tipo frontera, tipo control y de tipo entidad que se identificaron en cada caso de uso así como la asignación de responsabilidades a cada una de ellas.

| | |
|----------------------------------|-------------------------------|
| Caso de Uso | 1) Ingresar al sistema |
| Clase de Análisis | InterfazIngresar |
| Tipo de Clase de Análisis | Frontera |
| Responsabilidades | |
| -Registrar usuario y password | -Validar usuario y password |

| | |
|--|--|
| Caso de Uso | 2) Mantenimiento de Información de Reaseguradoras |
| Clase de Análisis | InterfazMantenerReaseguradora |
| Tipo de Clase de Análisis | Frontera |
| Responsabilidades | |
| -Cancelar pool del año anterior | -Desplegar los cambios de una reaseguradora |
| -Desplegar la cancelación de una reaseguradora | -Desplegar la inscripción de una reaseguradora |
| -Desplegar los números de registro correspondientes a las reaseguradoras que se pueden renovar | -Desplegar los datos principales, agencias calificadoras y pooles de una reaseguradora |
| -Desplegar la renovación de una reaseguradora | -Eliminar la cancelación de una reaseguradora |
| -Guardar los cambios de una reaseguradora | -Guardar la cancelación de una reaseguradora |
| -Guardar la inscripción de una reaseguradora | -Guardar los datos principales, agencias |

Análisis con UML del sistema: Registro General de Reaseguradoras Extranjeras

| | |
|--|--|
| | calificadoras y pooles de una reaseguradora |
| -Guardar la renovación de una reaseguradora | -Ingresar el año de renovación de las renovaciones |
| -Ingresar el nombre de una reaseguradora | -Ingresar el número de registro de una reaseguradora |
| -Registrar un nuevo pool de renovación | -Renovar una reaseguradora cancelada por error |
| -Seleccionar el tipo de mantenimiento (inscripciones, renovaciones, cancelaciones o cambios) | -Seleccionar los números de registro de las reaseguradoras que se van a renovar |
| | |
| Clase de Análisis | ControlReaseguradora |
| Tipo de Clase de Análisis | Control |
| Responsabilidades | |
| -Cancelar pool del año anterior | -Eliminar cancelación de una reaseguradora |
| -Guardar cambios de una reaseguradora | -Guardar cancelación de una reaseguradora |
| -Guardar inscripción de una reaseguradora | -Guardar datos principales, agencias calificadoras y pooles de una reaseguradora |
| -Guardar renovación de una reaseguradora | -Obtener cambios de una reaseguradora |
| -Obtener cancelación de una reaseguradora | -Obtener números de registro correspondientes a las reaseguradoras que se pueden renovar |
| -Obtener datos principales, agencias calificadoras y pooles de una reaseguradora | -Obtener renovación de una reaseguradora |
| -Registrar nuevo pool de renovación | |
| | |
| Clase de Análisis | Reaseguradora |
| Tipo de Clase de Análisis | Entidad |

**TESIS CON
FALLA DE ORIGEN**

| | |
|---|--|
| Responsabilidades | |
| -Guardar inscripción de una reaseguradora | -Guardar información de los datos principales, guardar agencias calificadoras y pools de una reaseguradora |
| -Obtener la información sobre los datos principales, agencias calificadoras, pools e inscripciones de una reaseguradora | |
| | |
| Clase de Análisis | Inscripción |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Guardar información de inscripción | -Obtener cancelación reaseguradora |
| -Obtener información de inscripción | -Obtener números de registro correspondientes a las reaseguradoras que se pueden renovar |
| | |
| Clase de Análisis | Agencia Calificadora |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Guardar información de la agencia calificadora | -Obtener información de la agencia calificadora |
| | |
| Clase de Análisis | Pool |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Cancelar pool del año anterior | -Guardar información del pool |
| -Obtener información del pool | -Registrar información de nuevo pool |
| | |
| Clase de Análisis | Renovación |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Cancelar pool del año anterior | -Guardar información de la renovación |

**TESIS CON
FALLA DE ORIGEN**

| | |
|---|--|
| -Obtener información de la renovación | -Registrar información de nuevo pool |
| Clase de Análisis | Cancelación |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Eliminar información de la cancelación | -Guardar información de la cancelación |
| -Obtener información de la cancelación | |
| Clase de Análisis | ReasegCambios |
| Tipo de Clase de Análisis | Entidad |
| Responsabilidades | |
| -Guardar información del cambio | -Obtener información del cambio |

| | |
|---|--|
| Caso de Uso | 3) Mantenimiento de Información de Oficinas de Representación |
| Clase de Análisis | InterfazMantenerOficina |
| Tipo de Clase de Análisis | Frontera |
| Responsabilidades | |
| -Desplegar la autorización de una oficina de representación | -Desplegar la clausura de una oficina de representación |
| -Desplegar los datos principales de una oficina de representación | -Eliminar la clausura de una oficina de representación |
| -Guardar la autorización de una oficina de representación | -Guardar los cambios de una oficina de representación |
| -Guardar la clausura de una oficina de representación | -Guardar los datos principales de una oficina de representación |
| -Ingresar el nombre de la oficina de representación | -Seleccionar el tipo de mantenimiento (autorizaciones, clausuras, cambios) |
| Clase de Análisis | ControlOficina |
| Tipo de Clase de Análisis | Control |

**TESIS CON
FALLA DE ORIGEN**

| | |
|--|--|
| Responsabilidades | |
| -Eliminar clausura de una oficina de representación | -Guardar autorización de una oficina de representación |
| -Guardar cambios de una oficina de representación | -Guardar clausura de una oficina de representación |
| -Guardar datos principales de una oficina de representación | -Obtener cambios de una oficina de representación |
| -Obtener clausura de una oficina de representación | -Obtener datos principales de una oficina de representación |
| Clase de Análisis | |
| Oficina | |
| Tipo de Clase de Análisis | |
| Entidad | |
| Responsabilidades | |
| -Guardar autorización de una oficina de representación | -Guardar información de los datos principales de una oficina de representación |
| -Obtener información de los datos principales de una oficina de representación | |
| Clase de Análisis | |
| Autorización | |
| Tipo de Clase de Análisis | |
| Entidad | |
| Responsabilidades | |
| -Guardar información de la autorización | -Obtener información de la autorización |
| Clase de Análisis | |
| Clausura | |
| Tipo de Clase de Análisis | |
| Entidad | |
| Responsabilidades | |
| -Guardar información de la clausura | -Obtener información de la clausura |
| Clase de Análisis | |
| OficinaCambios | |
| Tipo de Clase de Análisis | |
| Entidad | |

| Responsabilidades | |
|---------------------------------|---------------------------------|
| -Guardar información del cambio | -Obtener información del cambio |

| | |
|---|---|
| Caso de Uso | 4) Monitoreo de Calificaciones |
| Clase de Análisis | InterfazMonitoreo |
| Tipo de Clase de Análisis | Frontera |
| Responsabilidades | |
| -Desplegar el monitoreo de la calificación de una reaseguradora | -Guardar el monitoreo de la calificación de una reaseguradora |
| -Ingresar el número de registro de una reaseguradora | |
| Clase de Análisis | |
| ControlMonitoreo | |
| Tipo de Clase de Análisis | |
| Control | |
| Responsabilidades | |
| -Guardar monitoreo de la calificación de una reaseguradora | -Obtener monitoreo de la calificación de una reaseguradora |
| Clase de Análisis | |
| Monitoreo | |
| Tipo de Clase de Análisis | |
| Entidad | |
| Responsabilidades | |
| -Guardar información del monitoreo | -Obtener información del monitoreo |

| | |
|--|---|
| Caso de Uso | 5) Generación de Reportes |
| Clase de Análisis | InterfazReportes |
| Tipo de Clase de Análisis | Frontera |
| Responsabilidades | |
| -Ingresar el año de renovación | -Ingresar las fechas de consulta (fecha inicial, fecha final) |
| -Ingresar el número de registro de una reaseguradora | -Seleccionar tipo de consulta |
| Clase de Análisis | |
| ControlReportes | |

**TESIS CON
FALLA DE ORIGEN**

Análisis con UML del sistema: Registro General de Reaseguradoras Extranjeras

| Tipo de Clase de Análisis | Control |
|---|--|
| Responsabilidades | |
| -Obtener las oficinas de representación autorizadas en un periodo en específico | -Obtener calificaciones por número de registro y periodo |
| -Obtener los cambios de una oficina de representación en un periodo en específico | -Obtener por número de registro los cambios de una reaseguradora en un periodo en específico |
| -Obtener las reaseguradoras canceladas en un periodo en específico | -Obtener la oficinas de representación clausuradas en un periodo en específico |
| -Obtener las reaseguradoras que se inscribieron en un periodo en específico | -Obtener los pooles a los que pertenece o ha pertenecido una reaseguradora en un periodo en específico |
| -Obtener las reaseguradoras que se renovaron en un año en específico | |
| | |
| Clase de Análisis | InterfazImpresora |
| Tipo de Clase de Análisis | Interfaz |
| Responsabilidades | -Imprimir las consultas realizadas |

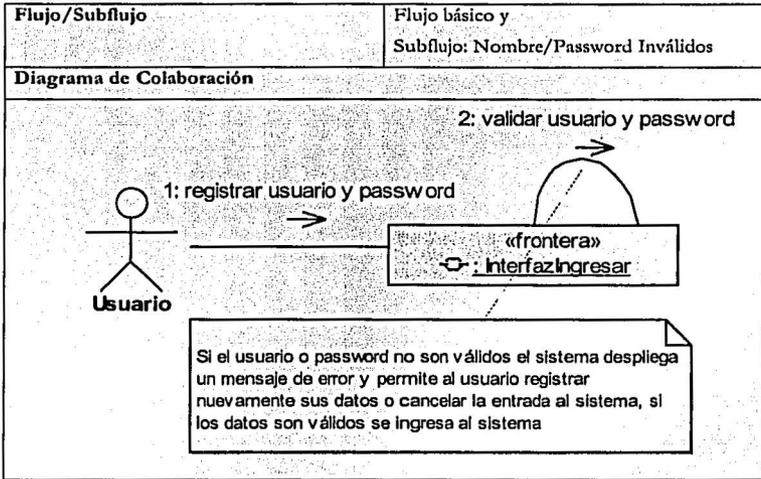
Diagramas de Interacción

Enseguida se presentan los diagramas de colaboración para cada caso de uso.

FRANCISCO DE CORDOVA

Caso de Uso:

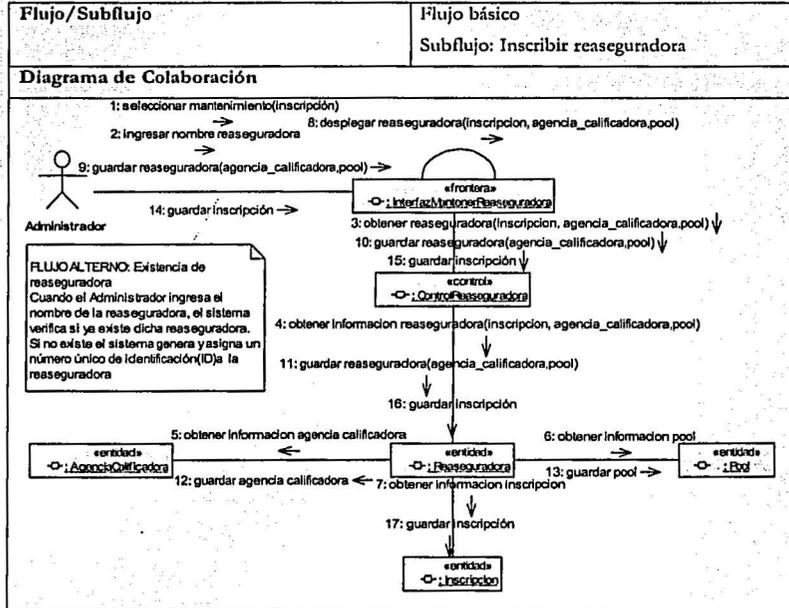
1) Ingresar al Sistema



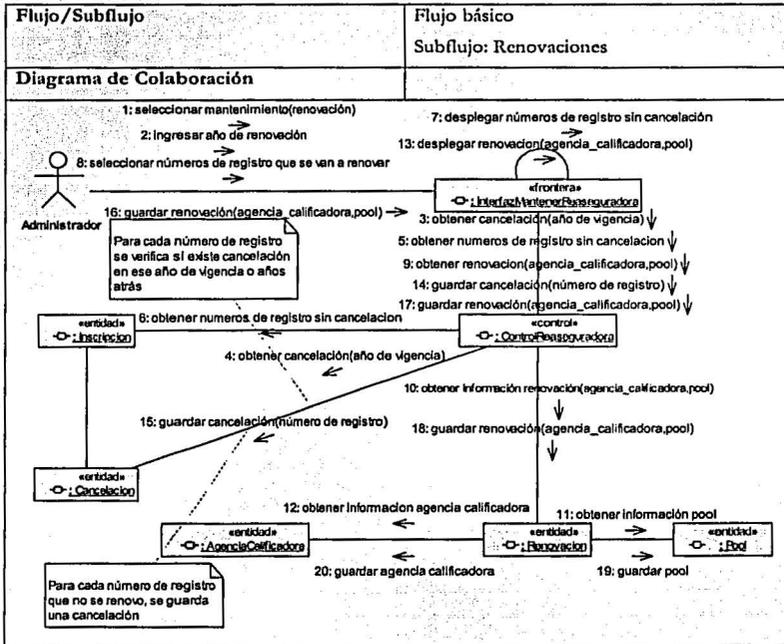
TESIS CON
FALLA DE ORIGEN

Caso de Uso:

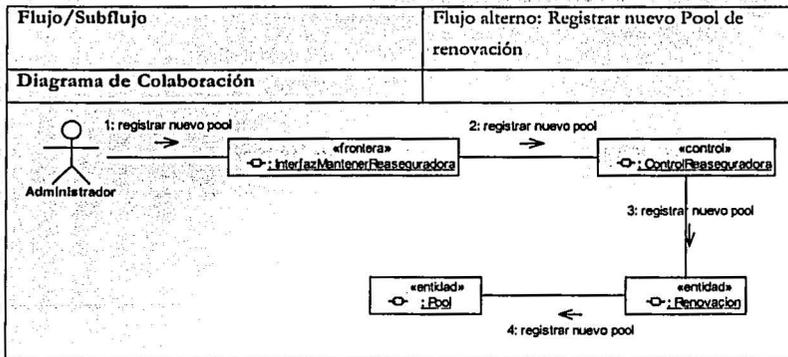
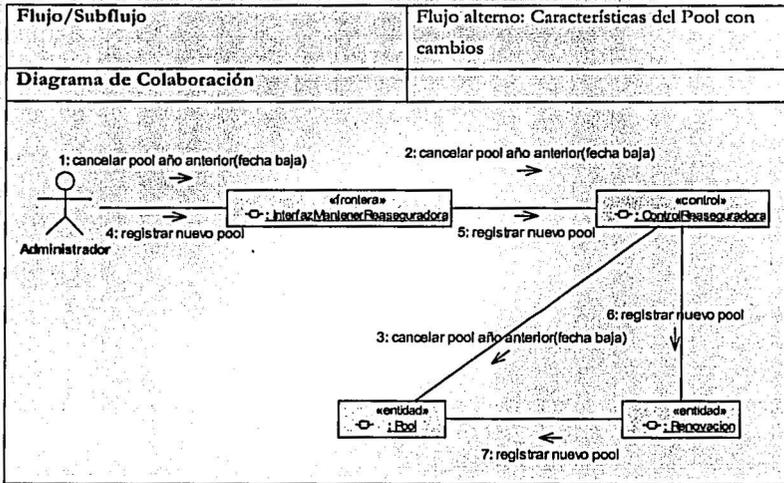
2) Mantenimiento de Información de Reaseguradoras



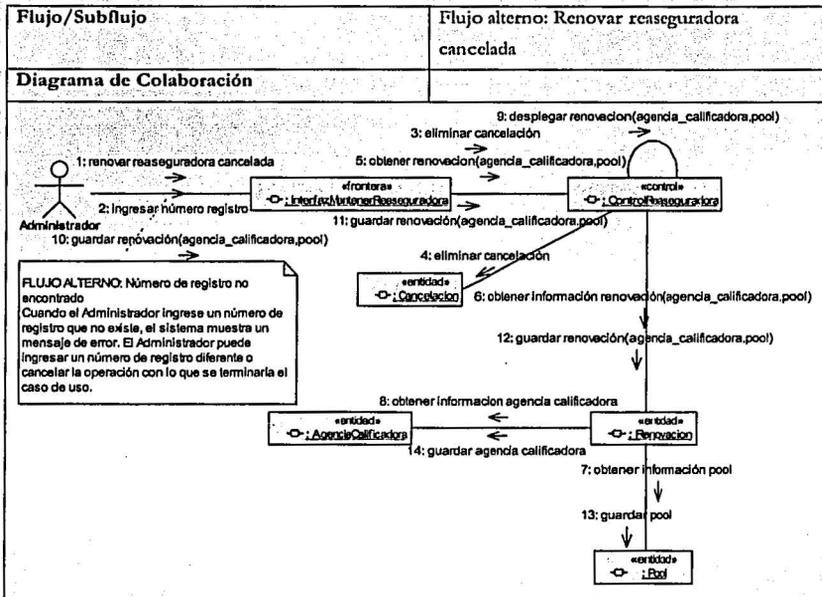
TESIS CON
 FALLA DE ORIGEN



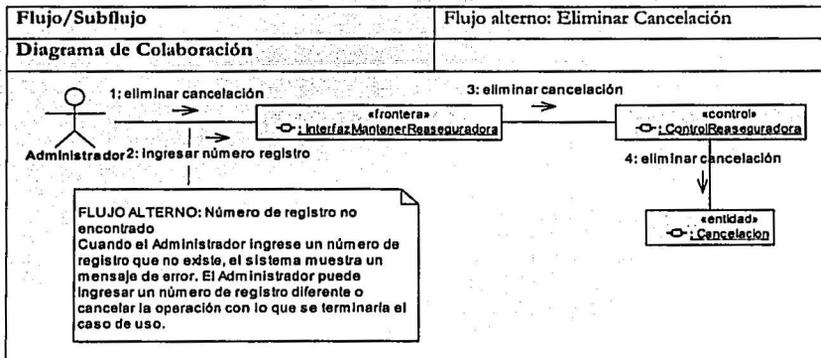
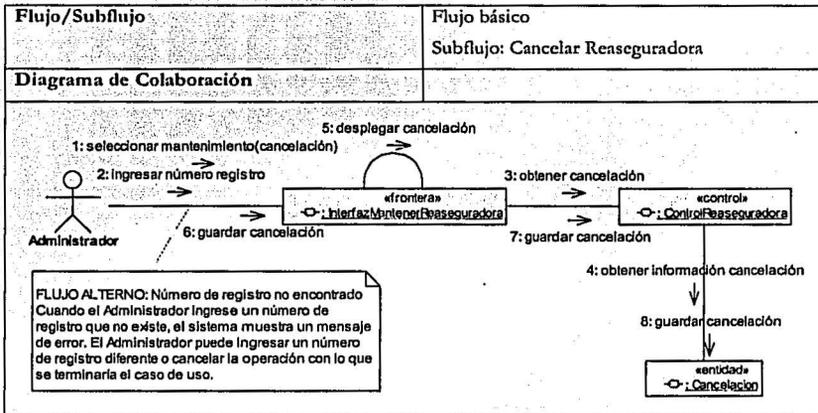
TESIS CON FALLA DE ORIGEN

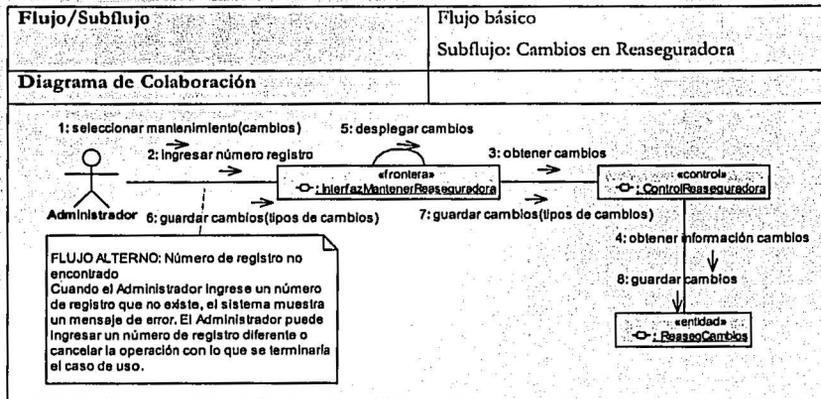


TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

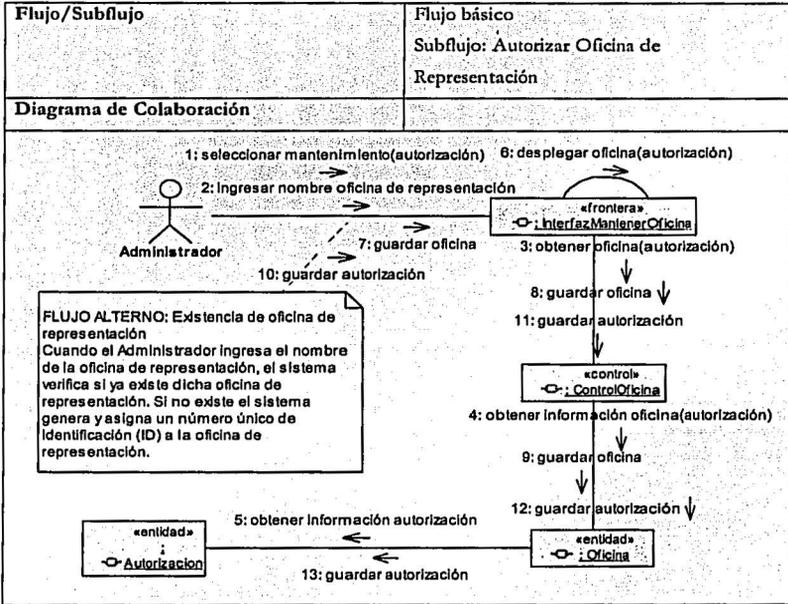




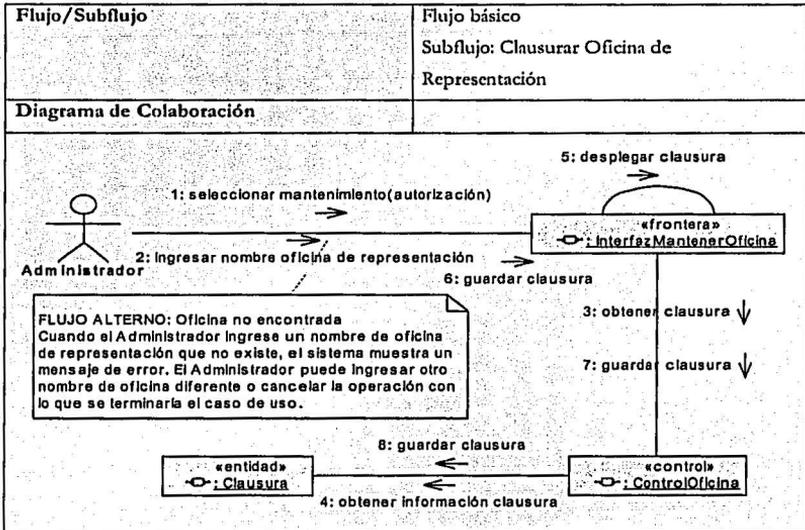
TESIS CON
 FALLA DE ORIGEN

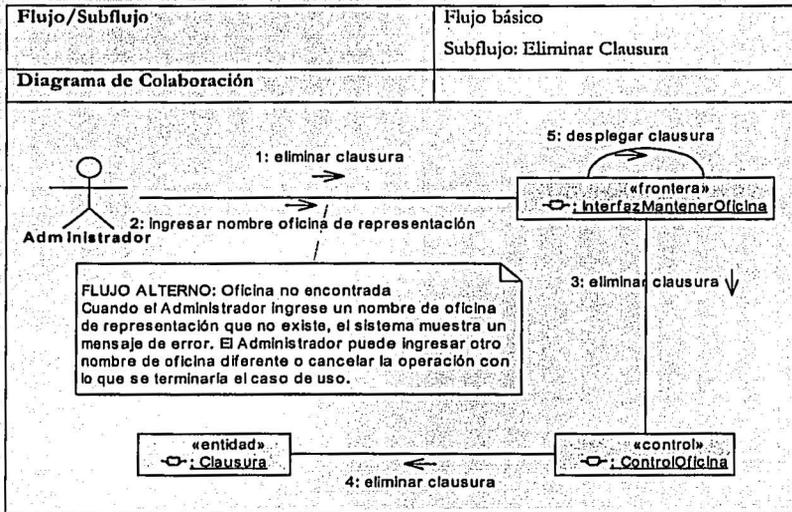
Caso de Uso:

3) Mantenimiento de Información de Oficinas de Representación

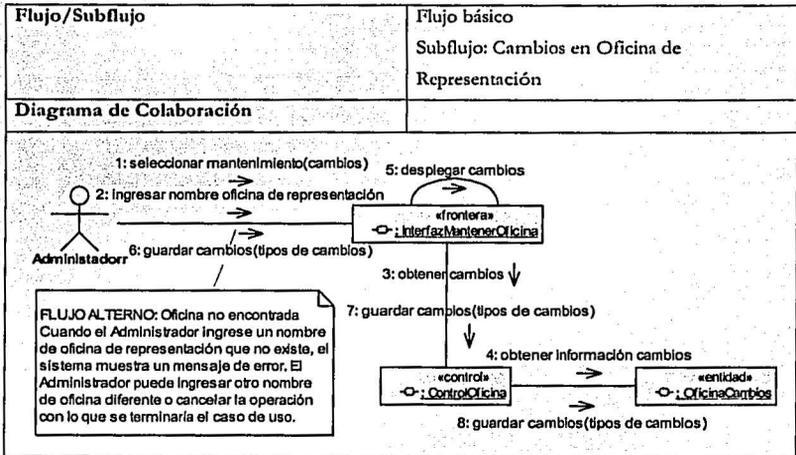


TESIS CON
 FALLA DE ORIGEN



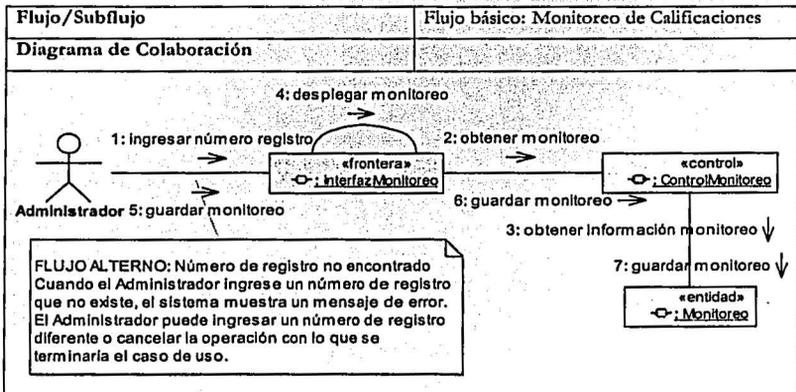


TESIS CON
FALLA DE ORIGEN



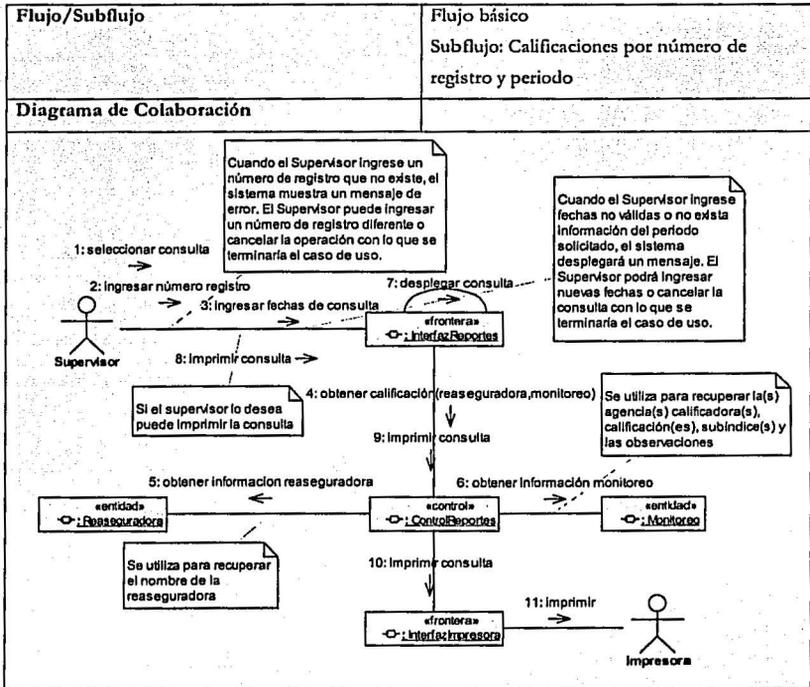
Caso de Uso:

4) Monitoreo de Calificaciones

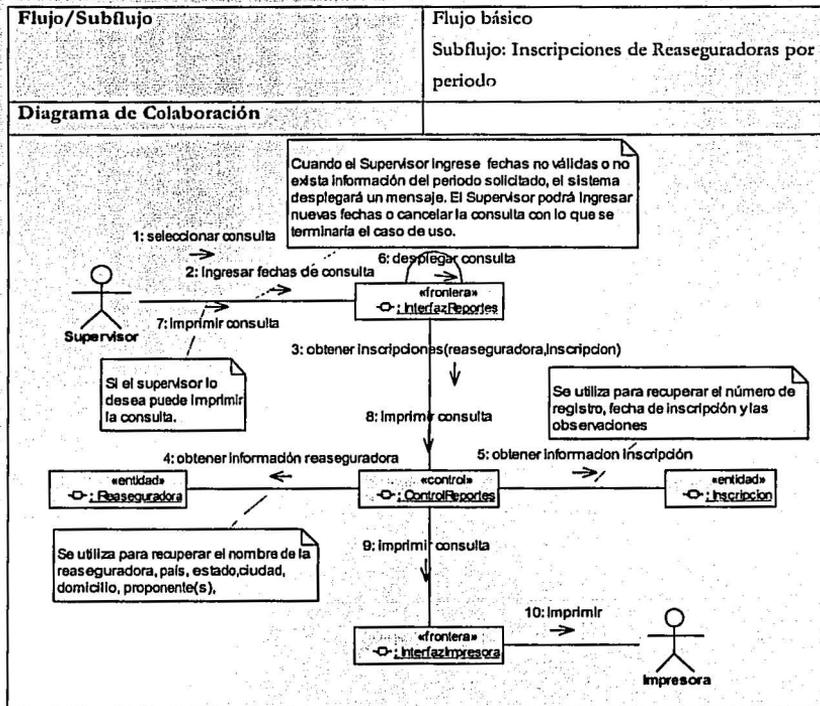


Caso de Uso:

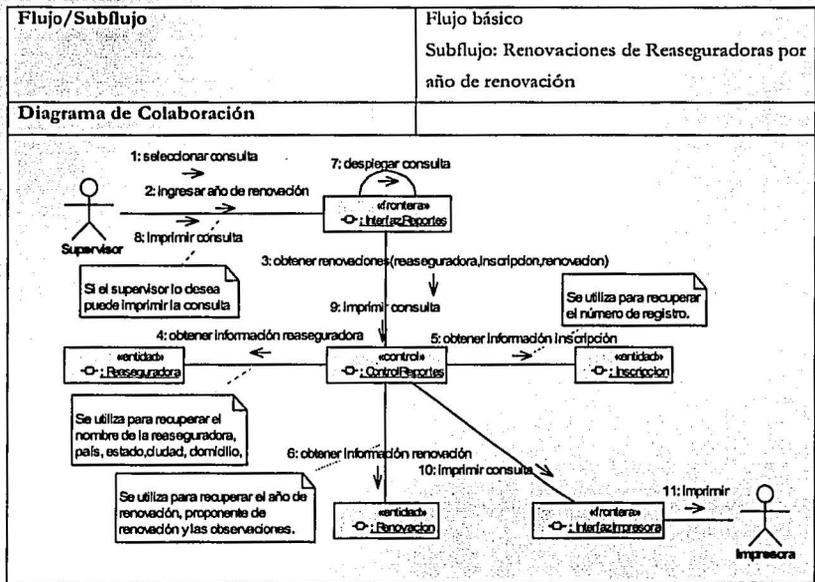
5) Generación de Reportes



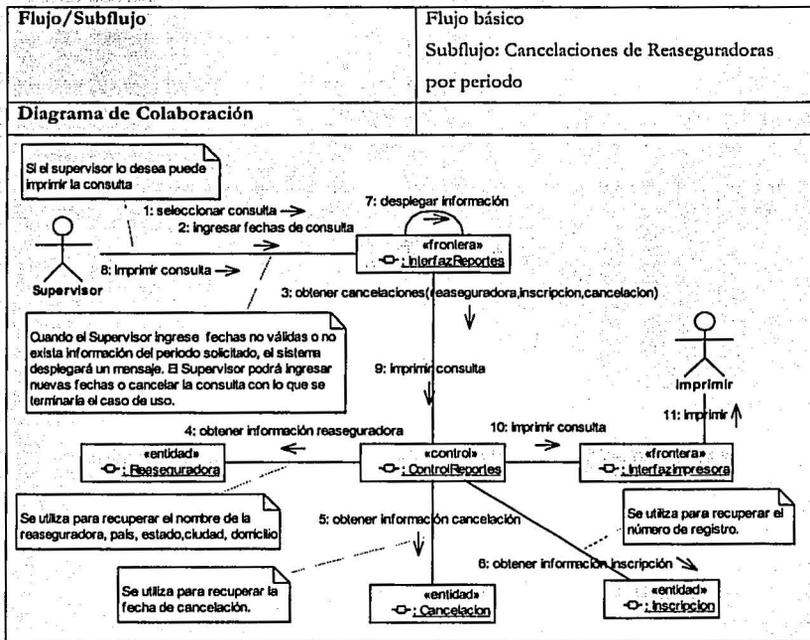
TESIS CON
FALLA DE ORIGEN



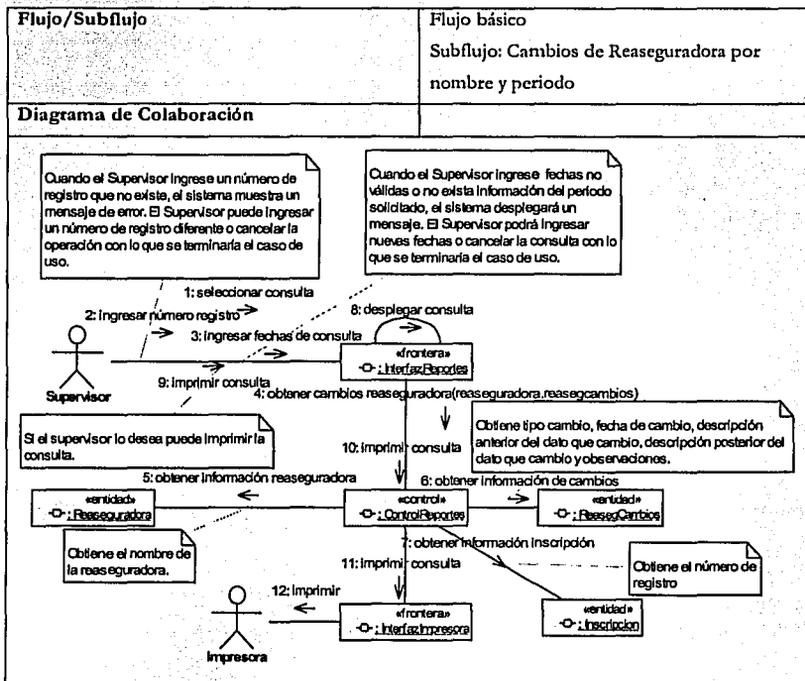
TESIS CON FALLA DE ORIGEN



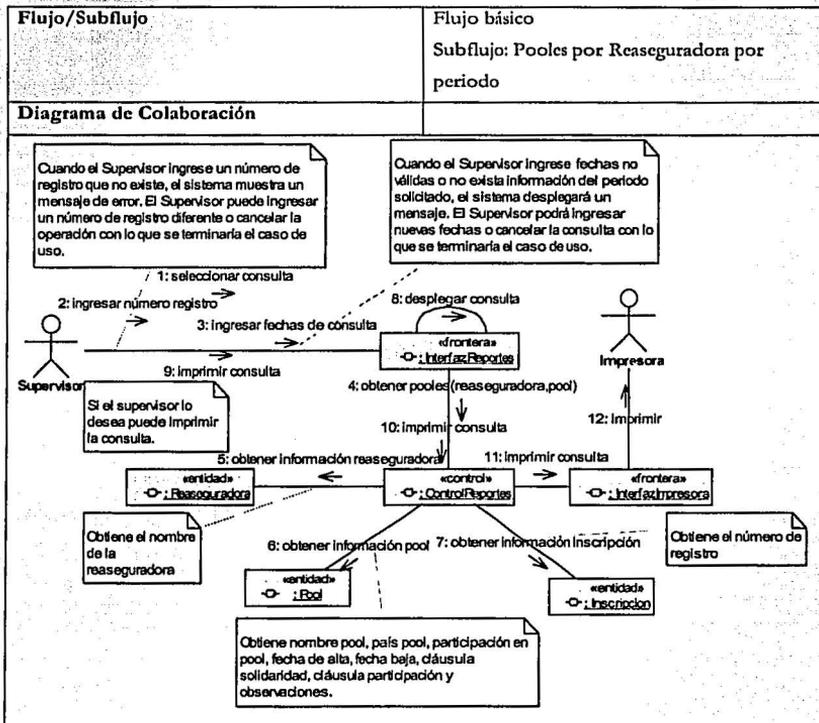
TESIS CON FALLA DE ORIGEN

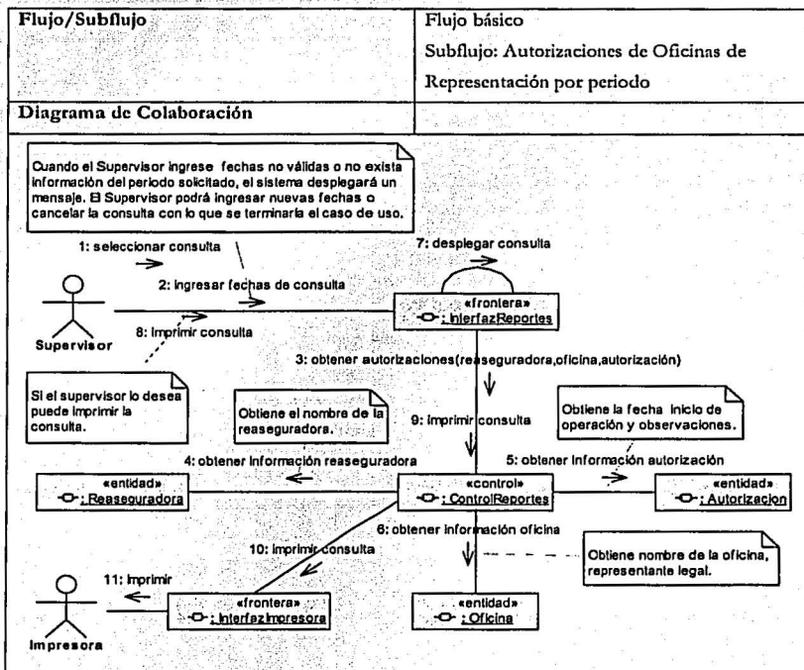


TESIS CON
FALLA DE ORIGEN

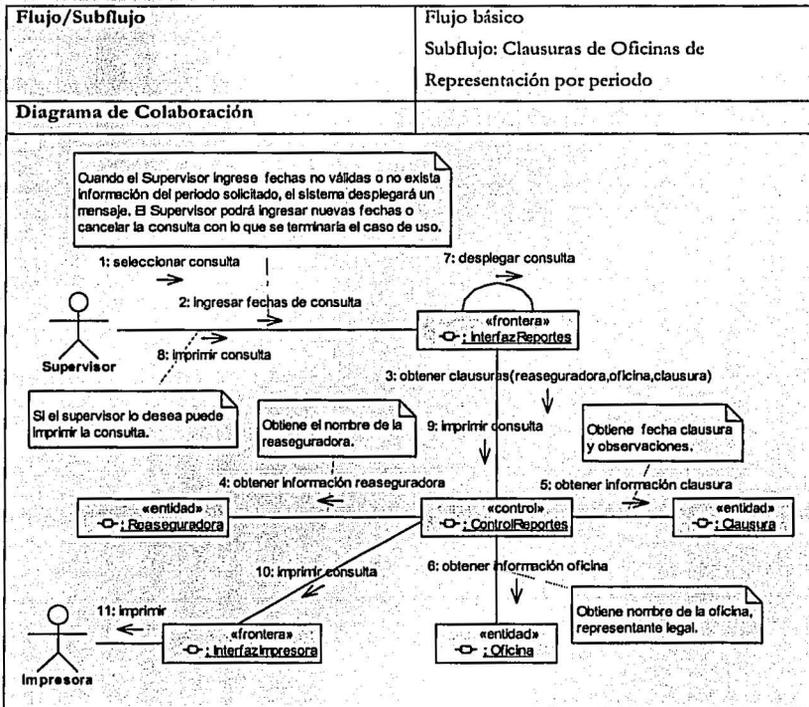


TESIS CON
 FALLA DE ORIGEN

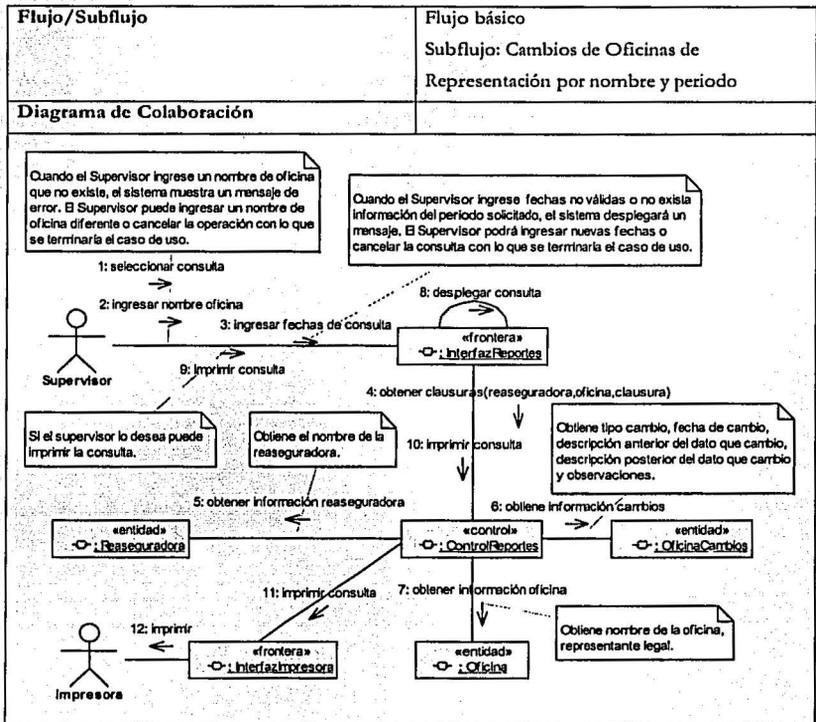




TESIS CON
 FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN



TESIS CON
 FALLA DE ORIGEN

Descripción de Responsabilidades

A continuación se presentan las clases identificadas en cada caso de uso ya con atributos y con operaciones (responsabilidades), además se presenta una breve descripción de cada una de sus responsabilidades.

| | |
|---|---|
| Caso de Uso | 1) Ingresar al sistema |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">«frontera»</p> <p style="text-align: center;">◄ InterfazIngresar</p> <hr/> <p style="text-align: center;">operations</p> <p>+ registrar_usuario_y_password</p> <p>+ validar_usuario_y_password</p> </div> |
| Responsabilidad: registrar_usuario_y_password | |
| Descripción: Permitir al usuario capturar su clave de acceso para ingresar al sistema. | |
| Responsabilidad: validar_usuario_y_password | |
| Descripción: Verificar que la clave del usuario este autorizada para ingresar al sistema y en caso de no estar autorizada, prohibir el acceso al mismo. | |

| | |
|-------------------|---|
| Caso de Uso | 2) Mantenimiento de Información de Reaseguradoras |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;">«frontera»</p> <p style="text-align: center;">◄ InterfazMantenerReaseguradora</p> <hr/> <p style="text-align: center;">operations</p> <p>+ cancelar_pool_anio_anterior(fecha_baja)</p> <p>+ desplegar_cambios</p> <p>+ desplegar_cancelacion</p> <p>+ desplegar_numeros_de_registro_sin_cancelacion</p> <p>+ desplegar_reaseguradora(inscripcion,agencia_calificadora,pool)</p> <p>+ desplegar_renovacion(agencia_calificadora,pool)</p> <p>+ eliminar_cancelacion</p> <p>+ guardar_cambios(tipos_de_cambios)</p> <p>+ guardar_cancelacion</p> <p>+ guardar_inscripcion</p> <p>+ guardar_reaseguradora(agencia_calificadora,pool)</p> <p>+ guardar_renovacion(agencia_calificadora,pool)</p> <p>+ ingresar_anio_renovacion</p> <p>+ ingresar_nombre_reaseguradora</p> <p>+ ingresar_numero_registro</p> <p>+ registrar_nuevo_pool</p> <p>+ renovar_reaseguradora_cancelada</p> <p>+ seleccionar_mantenimiento</p> <p>+ seleccionar_numeros_de_registro_a_renovar</p> </div> |

| |
|--|
| Responsabilidad: cancelar_pool_anio_anterior |
| Descripción: Permitir al administrador cancelar pool del año anterior cuando encuentre características diferentes con respecto al pool de renovación. |
| Responsabilidad: desplegar_cambios |
| Descripción: Mostrar la información relacionada con los cambios de una reaseguradora. |
| Responsabilidad: desplegar_cancelacion |
| Descripción: Mostrar la información relacionada con la cancelación de una reaseguradora. |
| Responsabilidad: desplegar_numeros_de_registro_sin_cancelacion |
| Descripción: Mostrar los número de registro que no hayan sido cancelados. |
| Responsabilidad: desplegar_reaseguradora |
| Descripción: Mostrar los datos de una reaseguradora así como la información referente a la(s) agencia(s) calificador(a)s de la misma, mostrar la información referente a pool(es) si es que existe y por último mostrar la información referente a la inscripción de la reaseguradora en caso de ser aceptada en el registro general de reaseguradoras extranjeras. |
| Responsabilidad: desplegar_renovacion |
| Descripción: Mostrar los datos referentes a la renovación de una reaseguradora así como la información asociada a la(s) agencia(s) calificador(a)s de la misma y en caso de existir, mostrar la información de el(los) pool(es) de renovación. |
| Responsabilidad: eliminar_cancelacion |
| Descripción: Permitir al administrador eliminar una cancelación que se haya registrado por error o que se haya registrado por no marcar para renovación el número de registro de una reaseguradora. |
| Responsabilidad: guardar_cambios |
| Descripción: Permitir al administrador guardar el o los cambios de una reaseguradora o guardar las modificaciones realizadas a la información de cambios. |
| Responsabilidad: guardar_cancelacion |
| Descripción: Permitir al administrador guardar la cancelación de una reaseguradora o guardar las modificaciones realizadas a la información de la cancelación. |
| Responsabilidad: guardar_inscripcion |
| Descripción: Permitir al administrador guardar la inscripción de una reaseguradora o guardar las modificaciones realizadas a la información de la inscripción. Cabe mencionar |

| |
|--|
| <p>que si no se ingresa el número de registro y la fecha de inscripción la información no puede guardarse.</p> |
| <p>Responsabilidad: guardar_reaseguradora</p> <p>Descripción: Permitir al administrador guardar los datos, la o las agencias calificadoras y el o los pooles de una reaseguradora o guardar las modificaciones realizadas a esta información.</p> |
| <p>Responsabilidad: guardar_renovacion</p> <p>Descripción: Permitir al administrador guardar la renovación de una reaseguradora, la o las agencias calificadoras con que se esta renovando y el o los pooles de renovación o guardar las modificaciones realizadas a esta información.</p> |
| <p>Responsabilidad: ingresar_anio_renovacion</p> <p>Descripción: Permitir al administrador ingresar el año de renovación para recuperar los números de registro de las reaseguradoras que no han sido canceladas.</p> |
| <p>Responsabilidad: ingresar_nombre_reaseguradora</p> <p>Descripción: Permitir al administrador ingresar el nombre de la reaseguradora para darla de alta en el sistema o en su defecto para recuperar toda la información (excepto la información relacionada con cancelaciones y cambios) asociada a la reaseguradora.</p> |
| <p>Responsabilidad: ingresar_numero_registro</p> <p>Descripción: Permitir al administrador ingresar el número de registro de una reaseguradora para recuperar la información de la cancelación o de los cambios que haya tenido la reaseguradora.</p> |
| <p>Responsabilidad: registrar_nuevo_pool</p> <p>Descripción: Permitir al administrador registrar los datos correspondientes a un nuevo pool de renovación.</p> |
| <p>Responsabilidad: renovar_reaseguradora_cancelada</p> <p>Descripción: Permite al administrador capturar la información referente a la renovación de alguna reaseguradora que no fue renovada por error.</p> |
| <p>Responsabilidad: seleccionar_mantenimiento</p> <p>Descripción: Permite al administrador seleccionar el proceso (inscripción, renovación, cancelación, cambios) que quiera llevar a cabo en el mantenimiento de reaseguradoras.</p> |
| <p>Responsabilidad: seleccionar_numeros_de_registro_a_renovar</p> <p>Descripción: Permite al administrador indicar los números de registro de las</p> |

| | |
|---|---|
| reaseguradoras que solicitaron renovación. | |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px;"> <p align="center">«control»</p> <p align="center">➔ ControlReaseguradora</p> <p align="center">operations</p> <ul style="list-style-type: none"> + cancelar_pool_anio_anterior(fecha_baja) + eliminar_cancelacion(numero_registro) + guardar_cambios(tipos_de_cambios) + guardar_cancelacion + guardar_inscripcion + guardar_reaseguradora(agenzia_calificadora,pool) + guardar_renovacion(agenzia_calificadora,pool) + obtener_cambios + obtener_cancelacion + obtener_numeros_de_registro_sin_cancelacion + obtener_reaseguradora(inscripcion, agencia_calificadora,pool) + obtener_renovacion(agenzia_calificadora,pool) + registrar_nuevo_pool </div> |
| Responsabilidad: cancelar_pool_anio_anterior | |
| Descripción: Indicar a la clase Pool que cancele el pool del año anterior. | |
| Responsabilidad: eliminar_cancelacion | |
| Descripción: Indicar a la clase Cancelacion que elimine la cancelación de una reaseguradora. | |
| Responsabilidad: guardar_cambios | |
| Descripción: Indicar a la clase ReasegCambios que guarde los cambios de una reaseguradora. | |
| Responsabilidad: guardar_cancelacion | |
| Descripción: Indicar a la clase Cancelacion que guarde la cancelación de una reaseguradora. | |
| Responsabilidad: guardar_inscripcion | |
| Descripción: Indicar a la clase Reaseguradora que se debe guardar la inscripción de una reaseguradora. | |
| Responsabilidad: guardar_reaseguradora | |
| Descripción: Indicar a la clase Reaseguradora que guarde los datos de una reaseguradora. Indicar a la clase AgenciaCalificadora que guarde la(s) agencia(s) calificadora(s) de una reaseguradora. Indicar a la clase Pool que guarde el o los pool(es) de una reaseguradora. | |
| Responsabilidad: guardar_renovacion | |
| Descripción: Indicar a la clase Renovacion que guarde la renovación de una reaseguradora. Indicar a la clase AgenciaCalificadora que guarde la(s) agencia(s) | |

TESIS CON
FALLA DE ORIGEN

calificadora(s) con que se esta renovando una reaseguradora. Indicar a la clase Pool que guarde el o los pool(es) de renovación de una reaseguradora.

Responsabilidad: obtener_cambios

Descripción: Solicitar a la clase ReasegCambios la información relacionada con los cambios de una reaseguradora.

Responsabilidad: obtener_cancelacion

Descripción: Solicitar a la clase Cancelacion la información relacionada con la cancelación de una reaseguradora.

Responsabilidad: obtener_numeros_de_registro_sin_cancelacion

Descripción: Solicitar a la clase Inscripcion los números de registro de las reaseguradoras que no tienen fecha de cancelación.

Responsabilidad: obtener_reaseguradora

Descripción: Solicitar a la clase Reaseguradora los datos de una reaseguradora. Solicitar a la clase Inscripcion la información relacionada a la inscripción de una reaseguradora. Solicitar a la clase AgenciaCalificadora la información de la(s) agencia(s) calificadora(s) de una reaseguradora. Solicitar a la clase Pool la información referente al o a los pool(es) de una reaseguradora.

Responsabilidad: obtener_renovacion

Descripción: Solicitar a la clase Renovacion la información relacionada con la renovación de una reaseguradora. Solicitar a la clase AgenciaCalificadora la información de la(s) agencia(s) calificadora(s) con que se renovó una reaseguradora. Solicitar a la clase Pool la información referente al o a los pool(es) de renovación de una reaseguradora.

Responsabilidad: registrar_nuevo_pool

Descripción: Indicar a la clase Renovacion que se debe guardar la información del pool de renovación.

TESIS CON
FALLA DE ORIGEN

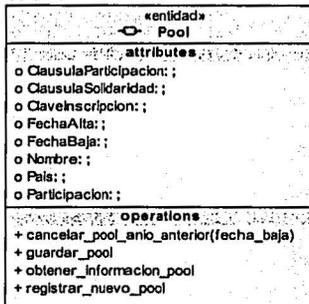
| | | | | | | |
|--|--|----------------------------|-----------|---|------------|--|
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td style="text-align: center;"> «entidad» Reaseguradora </td> </tr> <tr> <td style="text-align: center;"> atributos </td> </tr> <tr> <td> o Domicilio ; o Email ; o Estado ; o Fax ; o FechaOficioOpinion ; o FechaOficioSolicitud ; o DReaseguradora ; o Nombre ; o NumeroOficioOpinion ; o NumeroOficioSolicitud ; o Pais ; o Proponente ; o ResultadoOpinion ; o Telefono ; </td> </tr> <tr> <td style="text-align: center;"> operations </td> </tr> <tr> <td> + guardar_inscripcion + guardar_reaseguradora(agencia_calificadora,pool) + obtener_informacion_reaseguradora(inscripcion, agencia_calificadora,pool) </td> </tr> </table> | «entidad» Reaseguradora | atributos | o Domicilio ; o Email ; o Estado ; o Fax ; o FechaOficioOpinion ; o FechaOficioSolicitud ; o DReaseguradora ; o Nombre ; o NumeroOficioOpinion ; o NumeroOficioSolicitud ; o Pais ; o Proponente ; o ResultadoOpinion ; o Telefono ; | operations | + guardar_inscripcion + guardar_reaseguradora(agencia_calificadora,pool) + obtener_informacion_reaseguradora(inscripcion, agencia_calificadora,pool) |
| «entidad» Reaseguradora | | | | | | |
| atributos | | | | | | |
| o Domicilio ; o Email ; o Estado ; o Fax ; o FechaOficioOpinion ; o FechaOficioSolicitud ; o DReaseguradora ; o Nombre ; o NumeroOficioOpinion ; o NumeroOficioSolicitud ; o Pais ; o Proponente ; o ResultadoOpinion ; o Telefono ; | | | | | | |
| operations | | | | | | |
| + guardar_inscripcion + guardar_reaseguradora(agencia_calificadora,pool) + obtener_informacion_reaseguradora(inscripcion, agencia_calificadora,pool) | | | | | | |
| <p>Responsabilidad: guardar_inscripcion</p> <p>Descripción: Indicar a la clase Inscripción que debe guardar la inscripción de una reaseguradora.</p> | | | | | | |
| <p>Responsabilidad: guardar_reaseguradora</p> <p>Descripción: Guardar los datos de una reaseguradora. Indicar a la clase AgenciaCalificadora que guarde la(s) agencia(s) calificadora(s) de una reaseguradora. Indicar a la clase Pool que guarde el o los pool(es) de una reaseguradora.</p> | | | | | | |
| <p>Responsabilidad: obtener_informacion_reaseguradora</p> <p>Descripción: Proporcionar a la clase ControlReaseguradora los datos de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrarlos. Solicitar a la clase Inscripcion la información relacionada con la inscripción de una reaseguradora. Solicitar a la clase AgenciaCalificadora la información de la(s) agencia(s) calificadora(s) de una reaseguradora. Solicitar a la clase Pool la información relacionada con el o los pools de una reaseguradora.</p> | | | | | | |

**TESIS CON
 FALLA DE ORIGEN**

| | | | | | | | |
|---|--|-----------|------------------------|------------|---|------------|---|
| <p>Clase de Análisis</p> | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">«entidad»</td> </tr> <tr> <td style="text-align: center;">-o Inscricion</td> </tr> <tr> <td style="text-align: center;">attributes</td> </tr> <tr> <td> <ul style="list-style-type: none"> o FechaInscricion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o NumeroRegistro ; o Observaciones ; </td> </tr> <tr> <td style="text-align: center;">operations</td> </tr> <tr> <td> <ul style="list-style-type: none"> + guardar_inscricion + obtener_informacion_inscricion + obtener_numeros_de_registro_sin_cancelacion </td> </tr> </table> | «entidad» | -o Inscricion | attributes | <ul style="list-style-type: none"> o FechaInscricion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o NumeroRegistro ; o Observaciones ; | operations | <ul style="list-style-type: none"> + guardar_inscricion + obtener_informacion_inscricion + obtener_numeros_de_registro_sin_cancelacion |
| «entidad» | | | | | | | |
| -o Inscricion | | | | | | | |
| attributes | | | | | | | |
| <ul style="list-style-type: none"> o FechaInscricion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o NumeroRegistro ; o Observaciones ; | | | | | | | |
| operations | | | | | | | |
| <ul style="list-style-type: none"> + guardar_inscricion + obtener_informacion_inscricion + obtener_numeros_de_registro_sin_cancelacion | | | | | | | |
| <p>Responsabilidad: guardar_inscricion</p> | | | | | | | |
| <p>Descripción: Guardar la información relacionada con la inscripción de una reaseguradora.</p> | | | | | | | |
| <p>Responsabilidad: obtener_informacion_inscricion</p> | | | | | | | |
| <p>Descripción: Proporcionar a la clase ControlReaseguradora la información relacionada con la inscripción de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrar dicha información.</p> | | | | | | | |
| <p>Responsabilidad: obtener_numeros_de_registro_sin_cancelacion</p> | | | | | | | |
| <p>Descripción: Proporcionar a la clase ControlReaseguradora los números de registro de las reaseguradoras que no tienen fecha de cancelación para que la clase InterfazMantenerReaseguradora pueda mostrarlos.</p> | | | | | | | |
| <p>Clase de Análisis</p> | <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">«entidad»</td> </tr> <tr> <td style="text-align: center;">-o AgenciaCalificadora</td> </tr> <tr> <td style="text-align: center;">attributes</td> </tr> <tr> <td> <ul style="list-style-type: none"> o Calificacion ; o ControlAMBest ; o FechaCalificacion ; o Nombre ; o SubIndice ; </td> </tr> <tr> <td style="text-align: center;">operations</td> </tr> <tr> <td> <ul style="list-style-type: none"> + guardar_agencia_calificadora + obtener_informacion_agencia_calificadora </td> </tr> </table> | «entidad» | -o AgenciaCalificadora | attributes | <ul style="list-style-type: none"> o Calificacion ; o ControlAMBest ; o FechaCalificacion ; o Nombre ; o SubIndice ; | operations | <ul style="list-style-type: none"> + guardar_agencia_calificadora + obtener_informacion_agencia_calificadora |
| «entidad» | | | | | | | |
| -o AgenciaCalificadora | | | | | | | |
| attributes | | | | | | | |
| <ul style="list-style-type: none"> o Calificacion ; o ControlAMBest ; o FechaCalificacion ; o Nombre ; o SubIndice ; | | | | | | | |
| operations | | | | | | | |
| <ul style="list-style-type: none"> + guardar_agencia_calificadora + obtener_informacion_agencia_calificadora | | | | | | | |
| <p>Responsabilidad: guardar_agencia_calificadora</p> | | | | | | | |
| <p>Descripción: Guardar la información relacionada con la(s) agencia(s) calificadora(s) de una reaseguradora.</p> | | | | | | | |
| <p>Responsabilidad: obtener_informacion_agencia_calificadora</p> | | | | | | | |

Descripción: Proporcionar a la clase ControlReaseguradora la información relacionada a la(s) agencia(s) calificadora(s) de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrar dicha información.

Clase de Análisis



Responsabilidad: cancelar_pool_anio_anterior

Descripción: Guardar la fecha de baja para el(los) Pool(es) del año anterior.

Responsabilidad: guardar_pool

Descripción: Guardar la información relacionada con el o los pool(es) de una reaseguradora.

Responsabilidad: obtener_informacion_pool

Descripción: Proporcionar a la clase ControlReaseguradora la información relacionada con el o los pooles de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrar dicha información.

Responsabilidad: registrar_nuevo_pool

Descripción: Guardar la información relacionada con el o los Pooles de una reaseguradora.

TESIS CON
FALLA DE ORIGEN

| | | | | | | | |
|--|--|-----------|---------------------|------------|---|------------|--|
| <p>Clase de Análisis</p> | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">«entidad»</td> </tr> <tr> <td style="text-align: center;">↳ Renovacion</td> </tr> <tr> <td style="text-align: center;">attributes</td> </tr> <tr> <td> <ul style="list-style-type: none"> o AñoRenovacion; ; o FechaEscrito; ; o FechaOficioCircular; ; o NumeroOficioCircular; ; o Observaciones; ; o ProponenteRenovacion; ; </td> </tr> <tr> <td style="text-align: center;">operations</td> </tr> <tr> <td> <ul style="list-style-type: none"> + guardar_renovacion(agenzia_calificadora, pool) + obtener_informacion_renovacion(agenzia_calificadora, pool) + registrar_nuevo_pool </td> </tr> </table> | «entidad» | ↳ Renovacion | attributes | <ul style="list-style-type: none"> o AñoRenovacion; ; o FechaEscrito; ; o FechaOficioCircular; ; o NumeroOficioCircular; ; o Observaciones; ; o ProponenteRenovacion; ; | operations | <ul style="list-style-type: none"> + guardar_renovacion(agenzia_calificadora, pool) + obtener_informacion_renovacion(agenzia_calificadora, pool) + registrar_nuevo_pool |
| «entidad» | | | | | | | |
| ↳ Renovacion | | | | | | | |
| attributes | | | | | | | |
| <ul style="list-style-type: none"> o AñoRenovacion; ; o FechaEscrito; ; o FechaOficioCircular; ; o NumeroOficioCircular; ; o Observaciones; ; o ProponenteRenovacion; ; | | | | | | | |
| operations | | | | | | | |
| <ul style="list-style-type: none"> + guardar_renovacion(agenzia_calificadora, pool) + obtener_informacion_renovacion(agenzia_calificadora, pool) + registrar_nuevo_pool | | | | | | | |
| <p>Responsabilidad: guardar_renovacion</p> <p>Descripción: Guardar la información relacionada con la información de la renovación de una reaseguradora. Indicar a la clase AgenciaCalificadora que guarde la(s) agencia(s) calificadora(s) con que se esta renovando una reaseguradora. Indicar a la clase Pool que guarde el o los pool(es) de renovación de una reaseguradora.</p> | | | | | | | |
| <p>Responsabilidad: obtener_informacion_renovacion</p> <p>Descripción: Proporcionar a la clase ControlReaseguradora la información relacionada con la renovación de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrarla. Solicitar a la clase AgenciaCalificadora la información relacionada con la(s) agencia(s) calificadora(s) de una reaseguradora. Solicitar a la clase Pool la información relacionada con el o los pool(es) de una reaseguradora.</p> | | | | | | | |
| <p>Responsabilidad: registrar_nuevo_pool</p> <p>Descripción: Indicar a la clase Pool que debe guardar el o los pool(es) de renovación de una reaseguradora.</p> | | | | | | | |

**TESIS CON
FALLA DE ORIGEN**

| | |
|---|--|
| <p>Clase de Análisis</p> | <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">«entidad»</p> <p style="text-align: center;">◉ Cancelacion</p> <hr/> <p style="text-align: center;">«atributes»</p> <ul style="list-style-type: none"> o FechaBajaWeb; ; o FechaCancelacion; ; o FechaOficioBaja; ; o FechaOficioCircular; ; o FechaOficioOpinion; ; o FechaOficioProrroga; ; o NumeroOficioBaja; ; o NumeroOficioCircular; ; o NumeroOficioOpinion; ; o NumeroOficioProrroga; ; o Observaciones; ; o SustentoOficioOpinion; ; <hr/> <p style="text-align: center;">«operations»</p> <ul style="list-style-type: none"> + eliminar_cancelacion + guardar_cancelacion + obtener_informacion_cancelacion </div> |
| <p>Responsabilidad: eliminar_cancelacion</p> | |
| <p>Descripción: Borrar la información relacionada con la cancelación de una reaseguradora.</p> | |
| <p>Responsabilidad: guardar_cancelacion</p> | |
| <p>Descripción: Guardar la información relacionada con la cancelación de una reaseguradora.</p> | |
| <p>Responsabilidad: obtener_informacion_cancelacion</p> | |
| <p>Descripción: Proporcionar a la clase ControlReaseguradora la información relacionada con la cancelación de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrar dicha información.</p> | |
| <p>Clase de Análisis</p> | <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">«entidad»</p> <p style="text-align: center;">◉ ReasegCambios</p> <hr/> <p style="text-align: center;">«atributes»</p> <ul style="list-style-type: none"> o Cambio; ; o FechaActualizacionWeb; ; o FechaCambio; ; o FechaOficioCambios; ; o NumeroOficioCambios; ; o Observaciones; ; o TipoCambio; ; <hr/> <p style="text-align: center;">«operations»</p> <ul style="list-style-type: none"> + guardar_cambios(tipos_de_cambios) + obtener_informacion_cambios </div> |
| <p>Responsabilidad: guardar_cambios</p> | |
| <p>Descripción: Guardar la información relacionada con los cambios que hayan tenido los datos de una reaseguradora.</p> | |

TESIS CON
 FALLA DE ORIGEN

Responsabilidad: obtener_informacion_cambios

Descripción: Proporcionar a la clase ControllReaseguradora la información relacionada con los cambios que hayan tenido los datos de una reaseguradora para que la clase InterfazMantenerReaseguradora pueda mostrar dicha información.

| | |
|--|---|
| Caso de Uso | 3) Mantenimiento de Información de Oficinas de Representación |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">«frontera»</p> <p style="text-align: center;">◉ InterfazMantenerOficina</p> <hr/> <p style="text-align: center;">operations</p> <ul style="list-style-type: none"> + desplegar_clausura + desplegar_oficina(autorizacion) + eliminar_clausura + guardar_autorizacion + guardar_cambios(tipos_de_cambios) + guardar_clausura + guardar_oficina + ingresar_nombre_oficina + seleccionar_mantenimiento </div> |
| Responsabilidad: desplegar_clausura | |
| Descripción: Mostrar la información relacionada con la clausura de una oficina de representación. | |
| Responsabilidad: desplegar_oficina | |
| Descripción: Mostrar los datos de una oficina de representación así como la información relacionada con la autorización de una oficina de representación. | |
| Responsabilidad: eliminar_clausura | |
| Descripción: Permitir al administrador eliminar una clausura que se haya registrado por error. | |
| Responsabilidad: guardar_autorizacion | |
| Descripción: Permitir al administrador guardar nueva información relacionada a la autorización de una oficina de representación o guardar las modificaciones realizadas a dicha información. Cabe mencionar que si no se ingresa la fecha de inicio de operación la información no puede guardarse. | |
| Responsabilidad: guardar_cambios | |

**TESIS CON
FALLA DE ORIGEN**

| | | | | |
|---|--|--|-------------------|---|
| <p>Descripción: Permitir al administrador guardar nueva información referente a el(los) cambio(s) de una oficina de representación o guardar las modificaciones hechas a dicha información.</p> | | | | |
| <p>Responsabilidad: guardar_clausura</p> | | | | |
| <p>Descripción: Permitir al administrador guardar la clausura de una oficina de representación o guardar las modificaciones realizadas a la información de la clausura.</p> | | | | |
| <p>Responsabilidad: guardar_oficina</p> | | | | |
| <p>Descripción: Permitir al administrador guardar los datos de una oficina de representación o guardar las modificaciones realizadas a esta información.</p> | | | | |
| <p>Responsabilidad: ingresar_nombre_oficina</p> | | | | |
| <p>Descripción: Permitir al administrador ingresar el nombre de la oficina de representación para darla de alta en el sistema o en su defecto para recuperar toda la información (excepto la información relacionada con clausuras y cambios) asociada a la oficina de representación.</p> | | | | |
| <p>Responsabilidad: seleccionar_mantenimiento</p> | | | | |
| <p>Descripción: Permite al administrador seleccionar el proceso (autorización, clausura, cambios) que quiere llevar a cabo en el mantenimiento de oficinas de representación.</p> | | | | |
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td align="center"> <p>«control»  ControlOficina</p> </td> </tr> <tr> <td align="center"> <p>operations</p> </td> </tr> <tr> <td> <ul style="list-style-type: none"> + eliminar_clausura + guardar_autorizacion + guardar_cambios(tipos_de_cambios) + guardar_clausura + guardar_oficina + obtener_cambios + obtener_clausura + obtener_oficina(autorizacion) </td> </tr> </table> | <p>«control»  ControlOficina</p> | <p>operations</p> | <ul style="list-style-type: none"> + eliminar_clausura + guardar_autorizacion + guardar_cambios(tipos_de_cambios) + guardar_clausura + guardar_oficina + obtener_cambios + obtener_clausura + obtener_oficina(autorizacion) |
| <p>«control»  ControlOficina</p> | | | | |
| <p>operations</p> | | | | |
| <ul style="list-style-type: none"> + eliminar_clausura + guardar_autorizacion + guardar_cambios(tipos_de_cambios) + guardar_clausura + guardar_oficina + obtener_cambios + obtener_clausura + obtener_oficina(autorizacion) | | | | |
| <p>Responsabilidad: eliminar_clausura</p> | | | | |
| <p>Descripción: Indicar a la clase Clausura que elimine la clausura de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: guardar_autorizacion</p> | | | | |
| <p>Descripción: Indicar a la clase Oficina que se debe guardar la autorización de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: guardar_cambios</p> | | | | |
| <p>Descripción: Indicar a la clase OficinaCambios que guarde los cambios de una oficina de representación.</p> | | | | |

| | |
|---|---|
| <p>Responsabilidad: guardar_clausura</p> <p>Descripción: Indicar a la clase Clausura que guarde la clausura de una oficina de representación.</p> | |
| <p>Responsabilidad: guardar_oficina</p> <p>Descripción: Indicar a la clase Oficina que guarde los datos de una oficina de representación.</p> | |
| <p>Responsabilidad: obtener_cambios</p> <p>Descripción: Solicitar a la clase OficinaCambios la información relacionada con los cambios de una oficina de representación.</p> | |
| <p>Responsabilidad: obtener_clausura</p> <p>Descripción: Solicitar a la clase Clausura la información relacionada con la clausura de una oficina de representación.</p> | |
| <p>Responsabilidad: obtener_oficina</p> <p>Descripción: Solicitar a la clase Oficina los datos de una oficina de representación. Solicitar a la clase Autorizacion la información relacionada a la autorización de una oficina de representación.</p> | |
| <p>Clase de Análisis</p> | <p align="center">«entidad»  Oficina</p> <p align="center">atributos</p> <ul style="list-style-type: none"> o DirectorGeneral ; o Domicilio ; o Email ; o Fax ; o FechaMemo ; o FechaMemoRespuesta ; o FechaOficioOpinion ; o FechaOficioSolicitud ; o IDOficinaRepresentacion ; o Nombre ; o NumeroMemo ; o NumeroMemoRespuesta ; o NumeroOficioOpinion ; o NumeroOficioSolicitud ; o NumeroRegistroReaseguradora ; o PuestoRepresentante ; o RepresentanteLegal ; o ResultadoOpinion ; o Telefono ; <p align="center">operations</p> <ul style="list-style-type: none"> + guardar_autorizacion + guardar_oficina + obtener_informacion_oficina(autorizacion) |
| | |

**TESIS CON
FALLA DE ORIGEN**

| | | | | |
|--|--|--|--|--|
| <p>Responsabilidad: guardar_autorizacion</p> <p>Descripción: Indicar a la clase Autorización que debe guardar la autorización de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: guardar_oficina</p> <p>Descripción: Guardar los datos de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: obtener_informacion_oficina</p> <p>Descripción: Proporcionar a la clase ControlOficina los datos de una oficina de representación para que la clase InterfazMantenerOficina pueda mostrarlos. Solicitar a la clase Autorizacion la información relacionada con la autorización de una reaseguradora.</p> | | | | |
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td> <p>«entidad»</p> <p>◊ Autorizacion</p> </td> </tr> <tr> <td> <p>attributes</p> <ul style="list-style-type: none"> o FechaInicioOperacion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o Observaciones ; </td> </tr> <tr> <td> <p>operations</p> <ul style="list-style-type: none"> + guardar_autorizacion + obtener_informacion_autorizacion </td> </tr> </table> | <p>«entidad»</p> <p>◊ Autorizacion</p> | <p>attributes</p> <ul style="list-style-type: none"> o FechaInicioOperacion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o Observaciones ; | <p>operations</p> <ul style="list-style-type: none"> + guardar_autorizacion + obtener_informacion_autorizacion |
| | <p>«entidad»</p> <p>◊ Autorizacion</p> | | | |
| <p>attributes</p> <ul style="list-style-type: none"> o FechaInicioOperacion ; o FechaOficioCircular ; o FechaOficioSHCP ; o FechaPublicacionWeb ; o NumeroOficioCircular ; o NumeroOficioSHCP ; o Observaciones ; | | | | |
| <p>operations</p> <ul style="list-style-type: none"> + guardar_autorizacion + obtener_informacion_autorizacion | | | | |
| <p>Responsabilidad: guardar_autorizacion</p> <p>Descripción: Guardar la información relacionada con la autorización de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: obtener_informacion_autorizacion</p> <p>Descripción: Proporcionar a la clase ControlOficina la información relacionada con la autorización de una oficina de representación para que la clase InterfazMantenerOficina pueda mostrar dicha información.</p> | | | | |

**TESIS CON
FALLA DE ORIGEN**

| | | | | |
|---|--|-------------------------------|---|--|
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td> «entidad» - Clausura </td> </tr> <tr> <td> atributes <ul style="list-style-type: none"> o FechaBajaWeb ; o FechaClausura ; o FechaOficioCircular ; o FechaOficioClausura ; o FechaOficioOpinion ; o NumeroOficioCircular ; o NumeroOficioClausura ; o NumeroOficioOpinion ; o Observaciones ; o SustentoOficioOpinion ; </td> </tr> <tr> <td> operations <ul style="list-style-type: none"> + guardar_clausura + obtener_informacion_clausura </td> </tr> </table> | «entidad» - Clausura | atributes <ul style="list-style-type: none"> o FechaBajaWeb ; o FechaClausura ; o FechaOficioCircular ; o FechaOficioClausura ; o FechaOficioOpinion ; o NumeroOficioCircular ; o NumeroOficioClausura ; o NumeroOficioOpinion ; o Observaciones ; o SustentoOficioOpinion ; | operations <ul style="list-style-type: none"> + guardar_clausura + obtener_informacion_clausura |
| «entidad» - Clausura | | | | |
| atributes <ul style="list-style-type: none"> o FechaBajaWeb ; o FechaClausura ; o FechaOficioCircular ; o FechaOficioClausura ; o FechaOficioOpinion ; o NumeroOficioCircular ; o NumeroOficioClausura ; o NumeroOficioOpinion ; o Observaciones ; o SustentoOficioOpinion ; | | | | |
| operations <ul style="list-style-type: none"> + guardar_clausura + obtener_informacion_clausura | | | | |
| <p>Responsabilidad: guardar_clausura</p> <p>Descripción: Guardar la información relacionada con la clausura de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: obtener_informacion_clausura</p> <p>Descripción: Proporcionar a la clase ControlOficina la información relacionada con la clausura de una oficina de representación para que la clase InterfazMantenerOficina pueda mostrar dicha información.</p> | | | | |
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td> «entidad» - OficinaCambios </td> </tr> <tr> <td> atributes <ul style="list-style-type: none"> o Cambio ; o FechaActualizacionWeb ; o FechaCambio ; o FechaOficioCambios ; o NumeroOficioCambios ; o Observaciones ; o TipoCambio ; </td> </tr> <tr> <td> operations <ul style="list-style-type: none"> + guardar_cambios(tipos_de_cambios) + obtener_informacion_cambios </td> </tr> </table> | «entidad» - OficinaCambios | atributes <ul style="list-style-type: none"> o Cambio ; o FechaActualizacionWeb ; o FechaCambio ; o FechaOficioCambios ; o NumeroOficioCambios ; o Observaciones ; o TipoCambio ; | operations <ul style="list-style-type: none"> + guardar_cambios(tipos_de_cambios) + obtener_informacion_cambios |
| «entidad» - OficinaCambios | | | | |
| atributes <ul style="list-style-type: none"> o Cambio ; o FechaActualizacionWeb ; o FechaCambio ; o FechaOficioCambios ; o NumeroOficioCambios ; o Observaciones ; o TipoCambio ; | | | | |
| operations <ul style="list-style-type: none"> + guardar_cambios(tipos_de_cambios) + obtener_informacion_cambios | | | | |
| <p>Responsabilidad: guardar_cambios</p> <p>Descripción: Guardar la información relacionada con los cambios que hayan tenido los datos de una oficina de representación.</p> | | | | |
| <p>Responsabilidad: obtener_informacion_cambios</p> <p>Descripción: Proporcionar a la clase ControlOficina la información relacionada con los cambios que hayan tenido los datos de una oficina de representación para que la clase</p> | | | | |

Interfaz Mantener Oficina pueda mostrar dicha información.

| Caso de Uso | 4) Monitoreo de Calificaciones |
|---|--|
| <p>Clase de Análisis</p> | <div style="border: 1px solid black; padding: 5px;"> <p>«frontera»</p> <p>◊ InterfazMonitoreo</p> <hr/> <p>operations</p> <p>+ desplegar_monitoreo</p> <p>+ guardar_monitoreo</p> <p>+ Ingresar_numero_registro</p> </div> |
| <p>Responsabilidad: desplegar_monitoreo</p> <p>Descripción: Mostrar la información relacionada con el monitoreo de calificaciones de una reaseguradora.</p> | |
| <p>Responsabilidad: guardar_monitoreo</p> <p>Descripción: Permitir al administrador guardar el monitoreo de calificaciones de una reaseguradora o guardar las modificaciones realizadas a la información del monitoreo.</p> | |
| <p>Responsabilidad: ingresar_numero_registro</p> <p>Descripción: Permitir al administrador ingresar el número de registro de una reaseguradora para recuperar la información asociada del monitoreo de calificaciones de dicha reaseguradora.</p> | |
| <p>Clase de Análisis</p> | <div style="border: 1px solid black; padding: 5px;"> <p>«control»</p> <p>◊ ControlMonitoreo</p> <hr/> <p>operations</p> <p>+ guardar_monitoreo</p> <p>+ obtener_monitoreo</p> </div> |
| <p>Responsabilidad: guardar_monitoreo</p> <p>Descripción: Indicar a la clase Monitoreo que guarde el monitoreo de calificaciones de la reaseguradora.</p> | |
| <p>Responsabilidad: obtener_monitoreo</p> <p>Descripción: Solicitar a la clase Monitoreo la información relacionada con el monitoreo de calificaciones de una reaseguradora.</p> | |

TESIS CON
 FALLA DE ORIGEN

| | | | | |
|--|---|--|---|--|
| <p>Clase de Análisis</p> | <table border="1"> <tr> <td data-bbox="576 112 812 153"> <p>«entidad» Monitoreo</p> </td> </tr> <tr> <td data-bbox="576 153 812 260"> <p>atributes</p> <ul style="list-style-type: none"> o AgenciaCalificadora ; o Calificacion ; o FechaCalificacion ; o Observaciones ; o Subindice ; </td> </tr> <tr> <td data-bbox="576 260 812 333"> <p>operations</p> <ul style="list-style-type: none"> + guardar_monitoreo + obtener_informacion_monitoreo </td> </tr> </table> | <p>«entidad» Monitoreo</p> | <p>atributes</p> <ul style="list-style-type: none"> o AgenciaCalificadora ; o Calificacion ; o FechaCalificacion ; o Observaciones ; o Subindice ; | <p>operations</p> <ul style="list-style-type: none"> + guardar_monitoreo + obtener_informacion_monitoreo |
| <p>«entidad» Monitoreo</p> | | | | |
| <p>atributes</p> <ul style="list-style-type: none"> o AgenciaCalificadora ; o Calificacion ; o FechaCalificacion ; o Observaciones ; o Subindice ; | | | | |
| <p>operations</p> <ul style="list-style-type: none"> + guardar_monitoreo + obtener_informacion_monitoreo | | | | |
| <p>Responsabilidad: guardar_monitoreo</p> <p>Descripción: Guardar la información relacionada con el monitoreo de calificaciones de una reaseguradora.</p> | | | | |
| <p>Responsabilidad: obtener_informacion_monitoreo</p> <p>Descripción: Proporcionar a la clase ControlMonitoreo la información relacionada con el monitoreo de calificaciones de una reaseguradora para que la clase InterfazMonitoreo pueda mostrar dicha información.</p> | | | | |

| | |
|---|--|
| Caso de Uso | 5) Generación de Reportes |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">«frontera»</p> <p style="text-align: center;">↳ InterfazReportes</p> <p style="text-align: center;">operations</p> <ul style="list-style-type: none"> + desplegar_consulta + ingresar_año_renovacion + ingresar_fecha_consulta + ingresar_numero_registro + seleccionar_consulta </div> |
| <p>Responsabilidad: desplegar_consulta</p> | |
| <p>Descripción: Mostrar la información de la consulta solicitada.</p> | |
| <p>Responsabilidad: ingresar_año_renovacion</p> | |
| <p>Descripción: Permitir al usuario ingresar el año de renovación para recuperar la información de la consulta "Renovaciones de Reaseguradoras por año de renovación".</p> | |
| <p>Responsabilidad: ingresar_fecha_consulta</p> | |
| <p>Descripción: Permitir al usuario ingresar la fecha inicial y la fecha final del periodo que se quiere consultar.</p> | |
| <p>Responsabilidad: ingresar_numero_registro</p> | |
| <p>Descripción: Permitir al usuario ingresar el número de registro de la reaseguradora que se quiere consultar.</p> | |
| <p>Responsabilidad: seleccionar_consulta</p> | |
| <p>Descripción: Permite al usuario seleccionar el tipo de consulta que quiere realizar ya sea para reaseguradoras o para oficinas de representación.</p> | |
| Clase de Análisis | <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">«control»</p> <p style="text-align: center;">↳ ControlReportes</p> <p style="text-align: center;">operations</p> <ul style="list-style-type: none"> + imprimir_consulta + obtener_autorizaciones(reaseguradora,oficina,autorizacion) + obtener_calificacion(reaseguradora,monitoreo) + obtener_cambiosoficina(oficina,reaseguradora,oficinacambios) + obtener_cambiosreaseg(reaseguradora,reasegcambios) + obtener_cancelaciones(reaseguradora,inscripcion,cancelacion) + obtener_clausuras(reaseguradora,oficina,clausura) + obtener_inscripciones(reaseguradora,inscripcion) + obtener_pooles(reaseguradora,pool) + obtener_renovaciones(reaseguradora,inscripcion,renovacion) </div> |
| <p>Responsabilidad: imprimir_consulta</p> | |
| <p>Descripción: Indicar a la clase InterfazImpresora que la consulta realizada se tiene que</p> | |

TESIS CON
 FALLA DE ORIGEN

| |
|--|
| <p>imprimir.</p> |
| <p>Responsabilidad: obtener_autorizaciones</p> <p>Descripción: Solicitar a la clase Oficina el nombre y el representante legal de una oficina de representación; solicitar a la clase Reaseguradora el nombre de la reaseguradora a la que esta asociada la oficina de representación; solicitar a la clase Autorizacion la fecha de inicio de operación y observaciones de las oficinas que se autorizaron el periodo solicitado.</p> |
| <p>Responsabilidad: obtener_calificacion</p> <p>Descripción: Solicitar a la clase Reaseguradora el nombre de la reaseguradora y solicitar a la clase Monitoreo la información correspondiente a el monitoreo de calificaciones de la reaseguradora del periodo solicitado.</p> |
| <p>Responsabilidad: obtener_cambiosoficina</p> <p>Descripción: Solicitar a la clase Oficina el nombre y el representante legal de una oficina de representación; solicitar a la clase Reaseguradora el nombre de la reaseguradora a la que esta asociada la oficina de representación; solicitar a la clase OficinaCambios el o los tipo(s) de cambio(s), fecha(s) de cambio(s), descripción(es) anterior(es) del o de los datos que cambiaron, descripción(es) posterior(es) del o de los datos que cambiaron y las observaciones de las oficina de representación.</p> |
| <p>Responsabilidad: obtener_cambiosreaseg</p> <p>Descripción: Solicitar a la clase Reaseguradora el nombre, país, estado, ciudad, domicilio de una reaseguradora; solicitar a la clase Inscripcion el número de registro de la reaseguradora; solicitar a la clase ReasegCambios el o los tipo(s) de cambio(s), fecha(s) de cambio(s), descripción(es) anterior(es) del o de los datos que cambiaron, descripción(es) posterior(es) del o de los datos que cambiaron y las observaciones de la reaseguradora.</p> |
| <p>Responsabilidad: obtener_cancelaciones</p> <p>Descripción: Solicitar a la clase Reaseguradora el nombre, país, estado, ciudad, domicilio de una reaseguradora; solicitar a la clase Inscripcion el número de registro de la reaseguradora; solicitar a la clase Cancelacion la fecha de cancelación y observaciones de las reaseguradoras que se cancelaron en el periodo solicitado.</p> |
| <p>Responsabilidad: obtener_clausuras</p> <p>Descripción: Solicitar a la clase Oficina el nombre y el representante legal de una oficina de representación; solicitar a la clase Reaseguradora el nombre de la reaseguradora a la que esta asociada la oficina de representación; solicitar a la clase Clausura la fecha de clausura y</p> |

| | | | | | |
|--|---|------------|---------------------|------------|------------|
| observaciones de las oficinas que se clausuraron el periodo solicitado. | | | | | |
| Responsabilidad: obtener_inscripciones | | | | | |
| Descripción: Solicitar a la clase Reaseguradora el nombre, país, estado, ciudad, domicilio, proponente(s) y solicitar a la clase Incripcion la fecha de inscripción, número de registro y observaciones de las reaseguradoras que se inscribieron en el periodo solicitado. | | | | | |
| Responsabilidad: obtener_pooles | | | | | |
| Descripción: Solicitar a la clase Reaseguradora el nombre de una reaseguradora; solicitar a la clase Incripcion el número de registro de la reaseguradora; solicitar a la clase Pool el nombre del pool, país del pool, participación en pool, fecha de alta del pool, fecha de baja del pool, cláusula solidaridad, cláusula participación y las observaciones de el o los pooles de la reaseguradora. | | | | | |
| Responsabilidad: obtener_renovaciones | | | | | |
| Descripción: Solicitar a la clase Reaseguradora el nombre, país, estado, ciudad, domicilio; solicitar a la clase Renovacion el año de renovación, proponente(s) de renovación y las observaciones de las reaseguradoras que se renovaron en el año solicitado. | | | | | |
| Clase de Análisis | <table border="1"><tr><td>«frontera»</td></tr><tr><td>◊ InterfazImpresora</td></tr><tr><td>operations</td></tr><tr><td>+ Imprimir</td></tr></table> | «frontera» | ◊ InterfazImpresora | operations | + Imprimir |
| «frontera» | | | | | |
| ◊ InterfazImpresora | | | | | |
| operations | | | | | |
| + Imprimir | | | | | |
| Responsabilidad: imprimir | | | | | |
| Descripción: Solicitar a la impresora que imprima la consulta realizada. | | | | | |

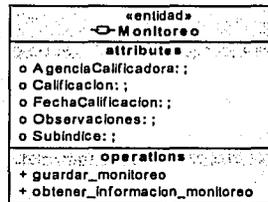
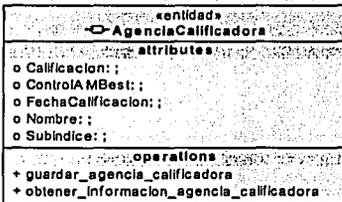
TESIS CON
FALLA DE ORIGEN

Descripción de mecanismos de análisis

Esta descripción no se llevo a cabo ya que no se realizó el análisis arquitectónico del sistema.

Unificación de clases de análisis

Para este caso sólo se encontraron dos clases que pueden ser unificadas: la clase AgenciaCalificadora y la clase Monitoreo.

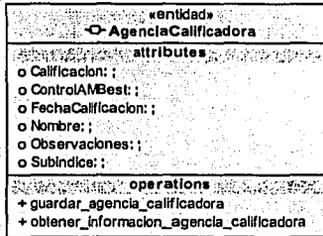


Las dos clases tienen los mismos atributos sólo que la clase Monitoreo tiene de más el atributo Observaciones por lo tanto se agrega dicho atributo a la clase AgenciaCalificadora.

Por otro lado al observar la descripción de las responsabilidades guardar_agencia_calificadora y guardar_monitoreo hacen lo mismo es decir, se encargan de guardar la información de cada clase, con lo que concluimos que podemos dejar cualquiera de las dos responsabilidades.

Lo mismo ocurre con las responsabilidades obtener_informacion_agencia_calificadora y obtener_informacion_monitoreo quedando nuestra clase unificada de la siguiente manera:

TESIS CON
FALLA DE ORIGEN



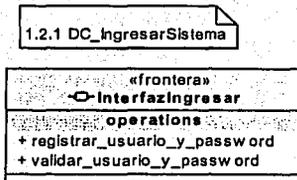
TESIS CON
FALLA DE ORIGEN

Diagramas de clases participantes

Para finalizar con el análisis de casos de uso presentamos los diagramas de clases participantes de los flujos y subflujos de cada caso de uso en los cuales podemos observar las asociaciones que existen entre las clases participantes.

Caso de Uso:

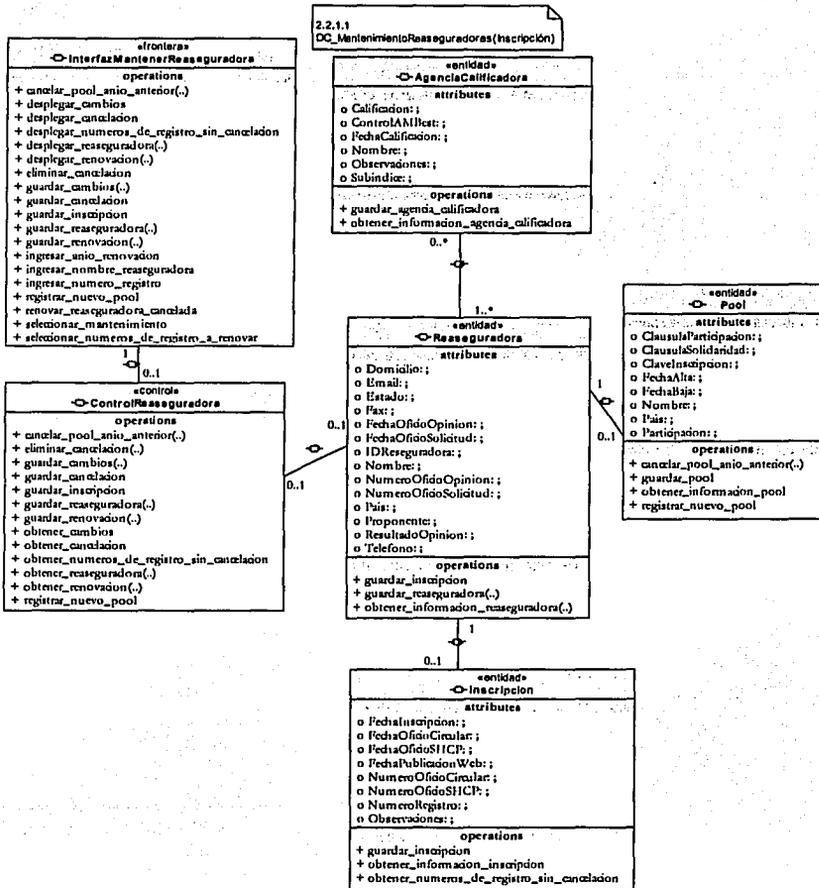
1) Ingresar al Sistema



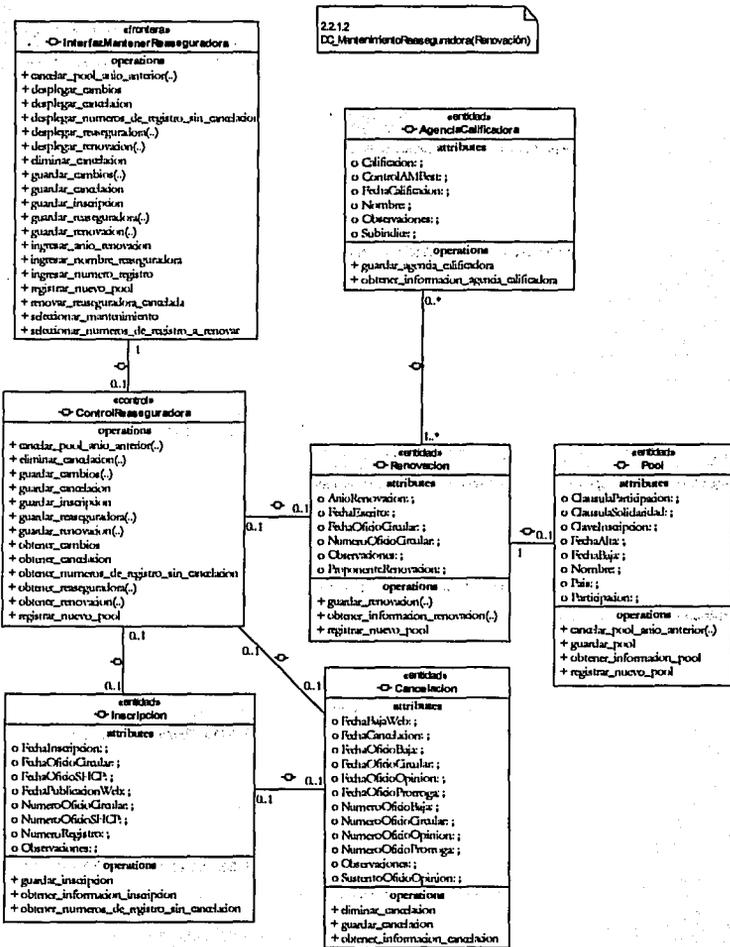
TESIS CON
FALLA DE ORIGEN

Caso de Uso:

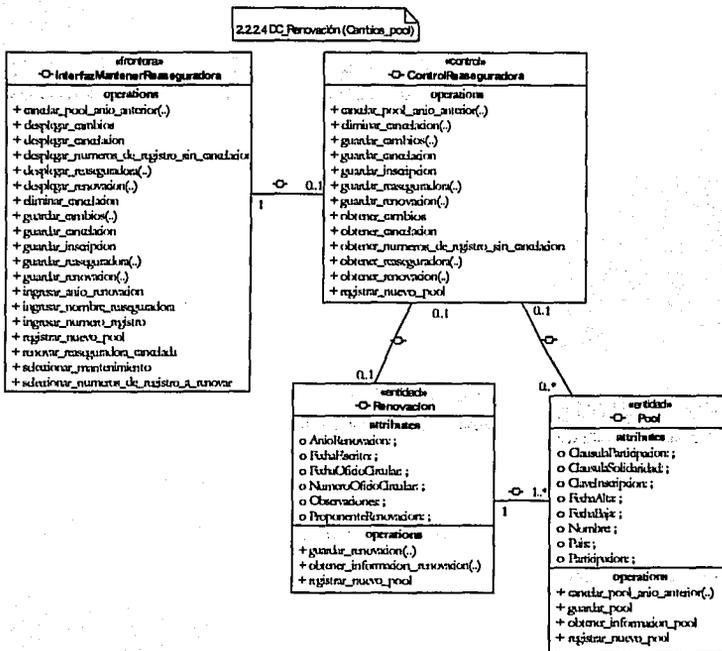
2) Mantenimiento de Información de Reaseguradoras



TESIS CON FALLA DE ORIGEN

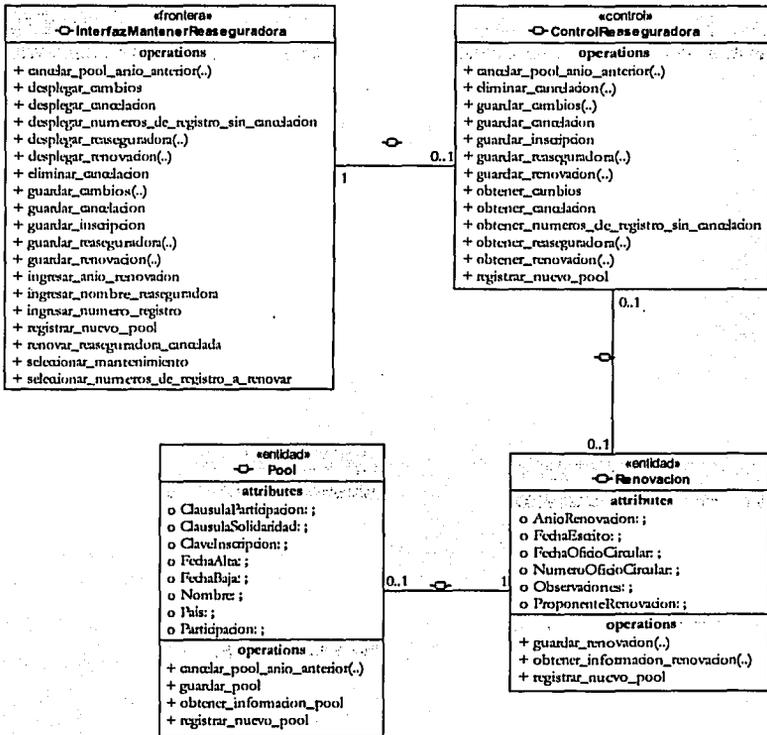


TESIS CON FALLA DE ORIGEN

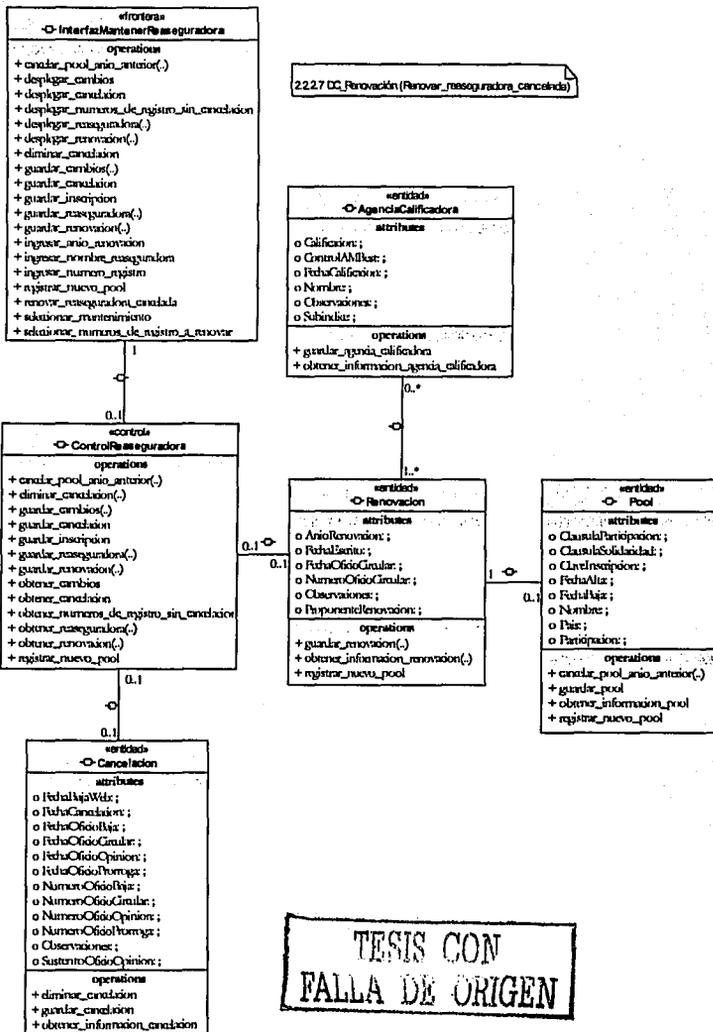


TESIS CON FALLA DE ORIGEN

2.2.2.5 DC_Renovación (Nuevo_pool)



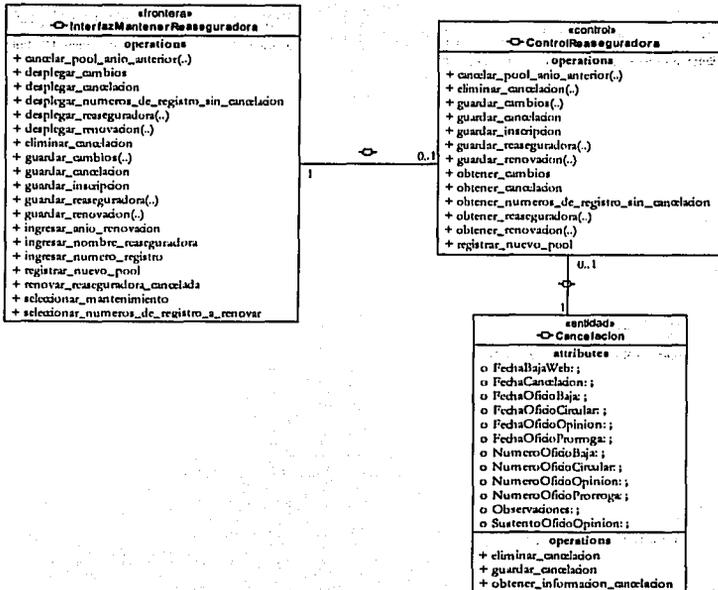
TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

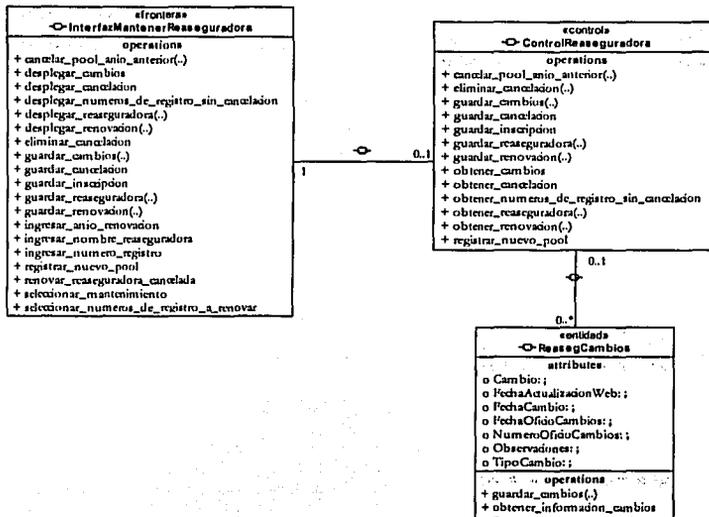
2.2.1.3

DC_MantenimientoReaseguradora(Cancelación)



TESIS CON
 FALLA DE ORIGEN

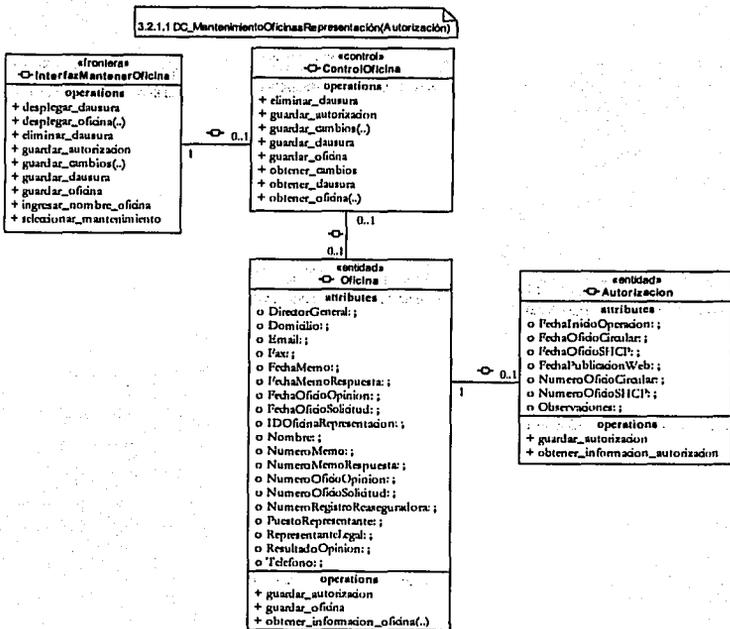
2.2.1.4 DC_MantenimientoReaseguradora(Cambios)



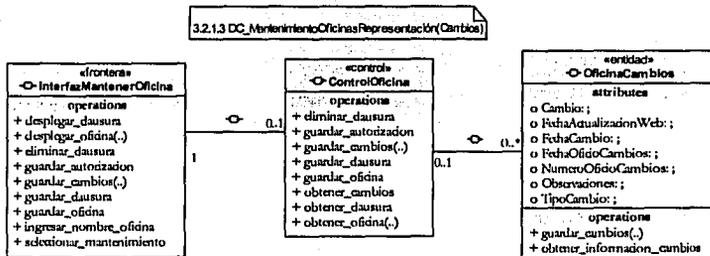
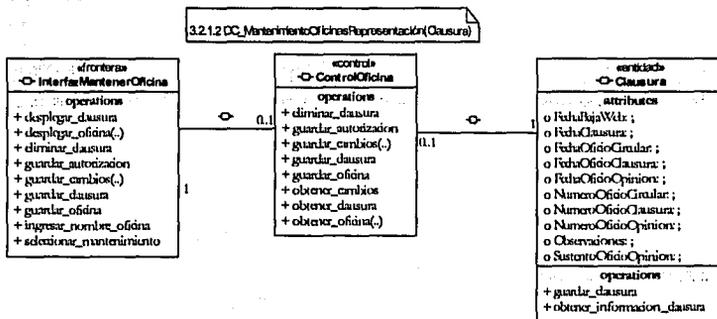
TESIS CON FALLA DE ORIGEN

Caso de Uso:

3) Mantenimiento de Información de Oficinas de Representación



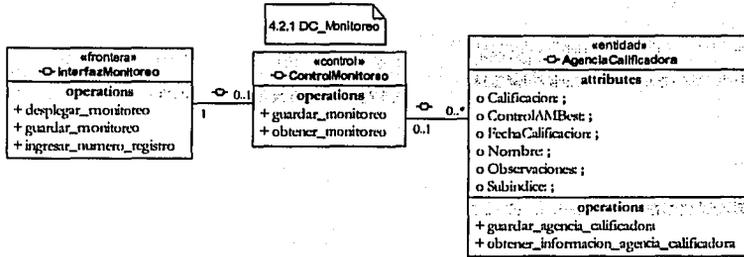
TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

Caso de Uso:

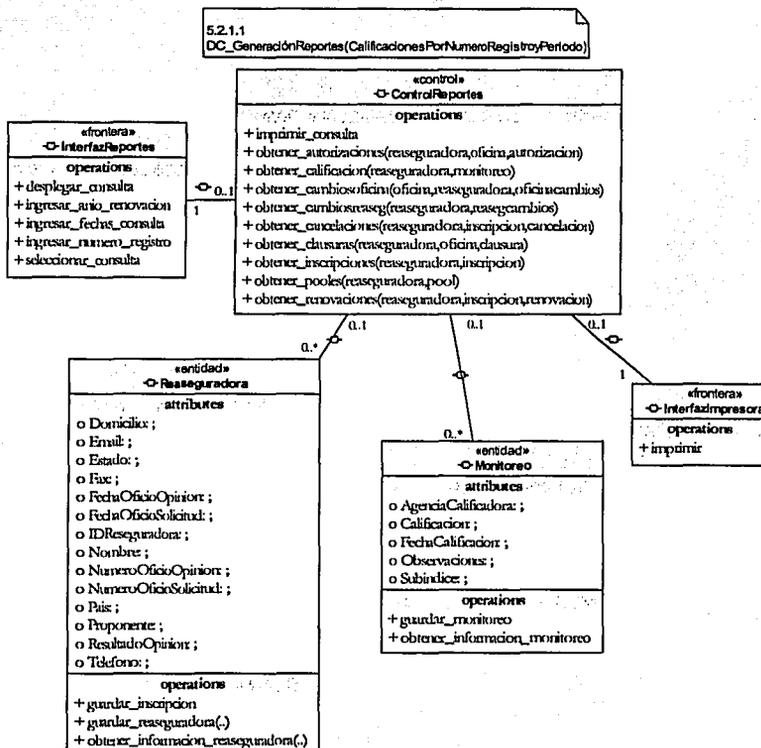
4) Monitoreo de Calificaciones



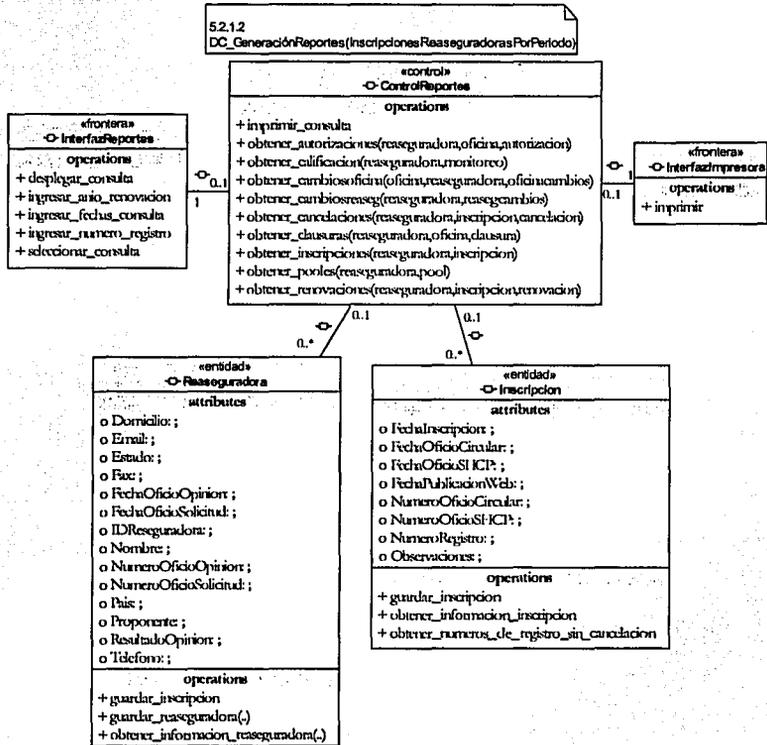
TESIS CON
FALLA DE ORIGEN

Caso de Uso:

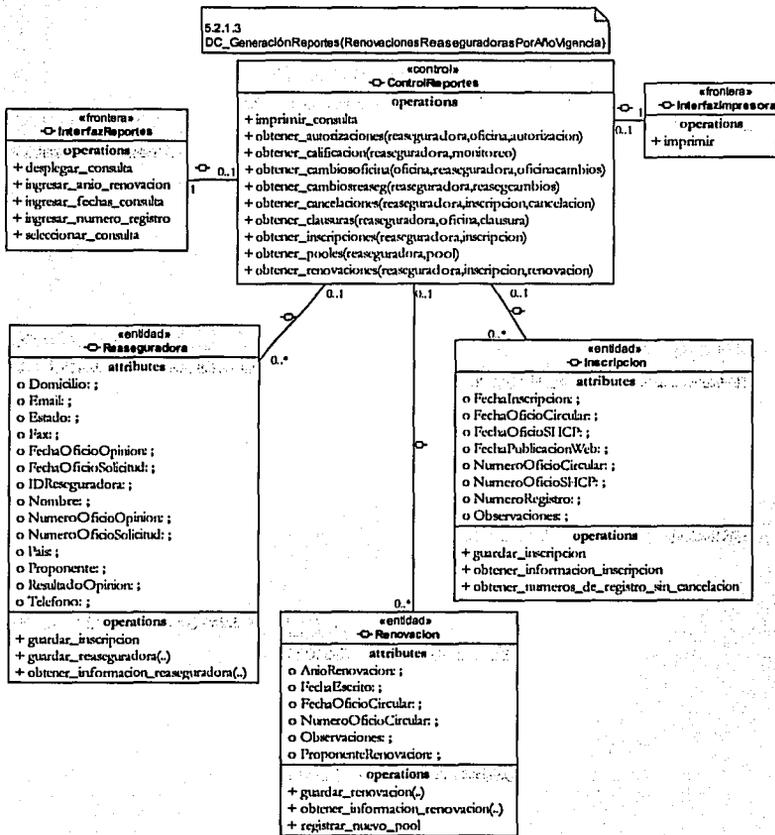
5) Generación de Reportes



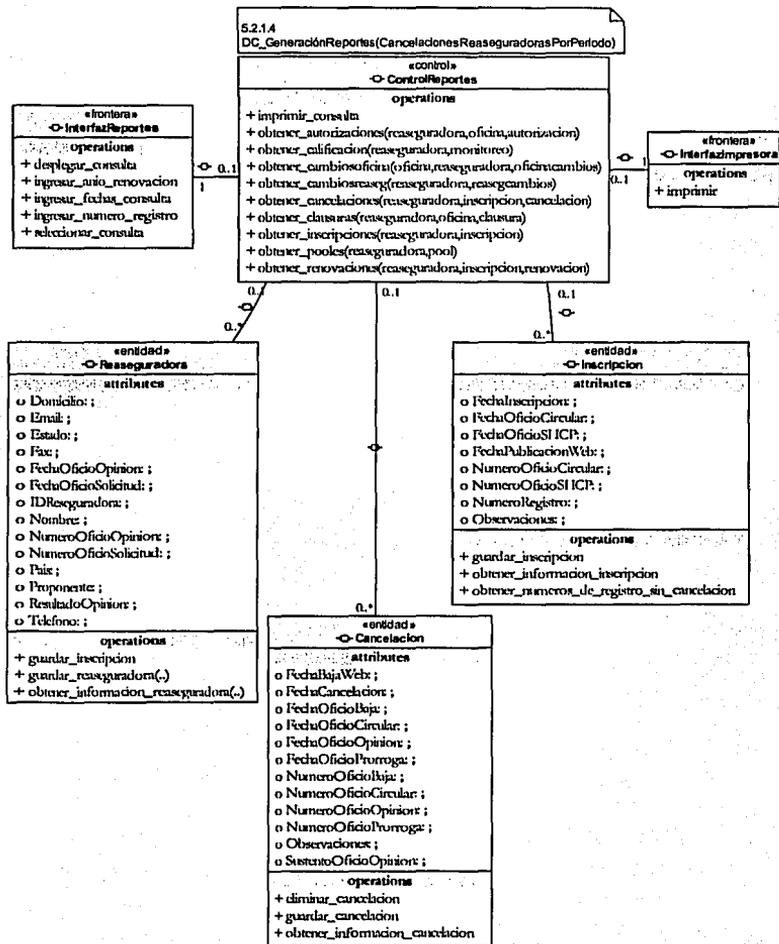
TESIS CON
FALLA DE ORIGEN



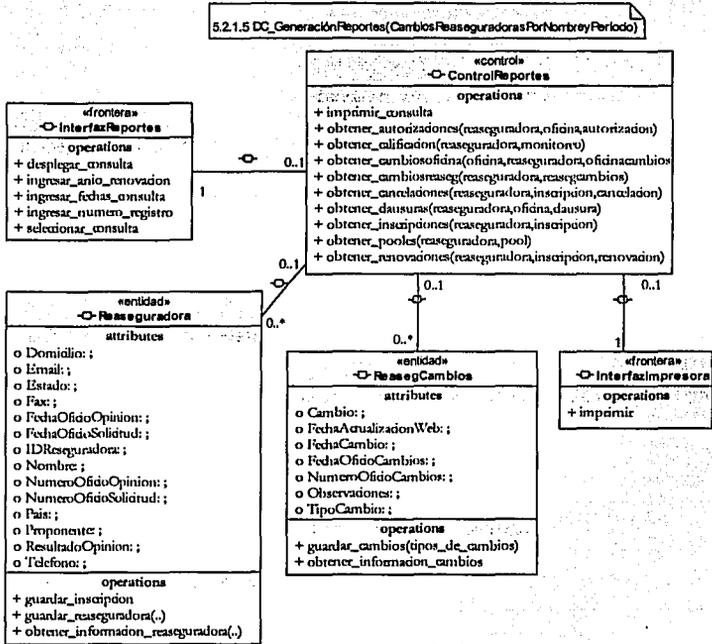
TESIS CON
FALLA DE ORIGEN



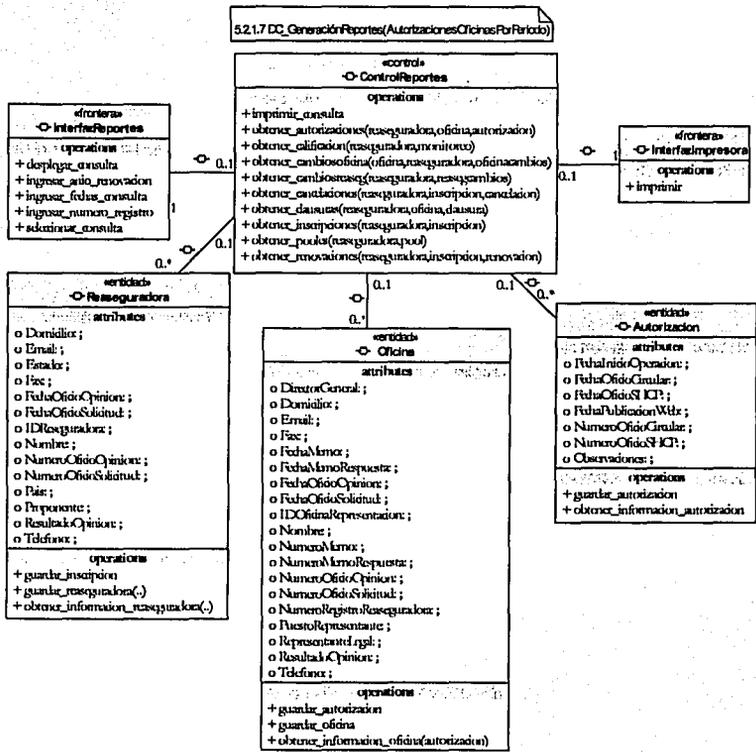
TESIS CON
FALLA DE CUBIEN



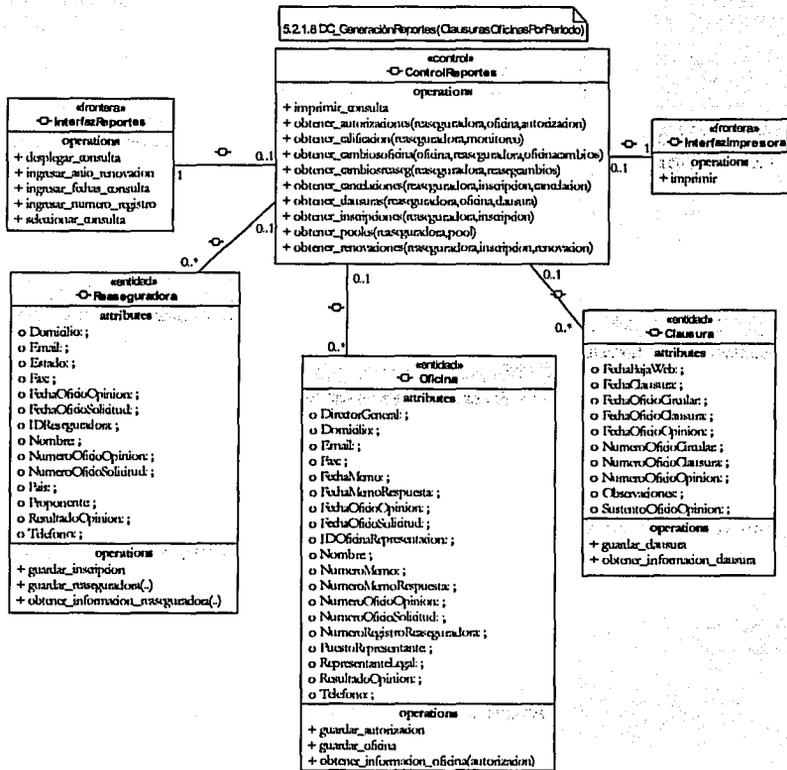
TESIS CON FALLA DE ORIGEN



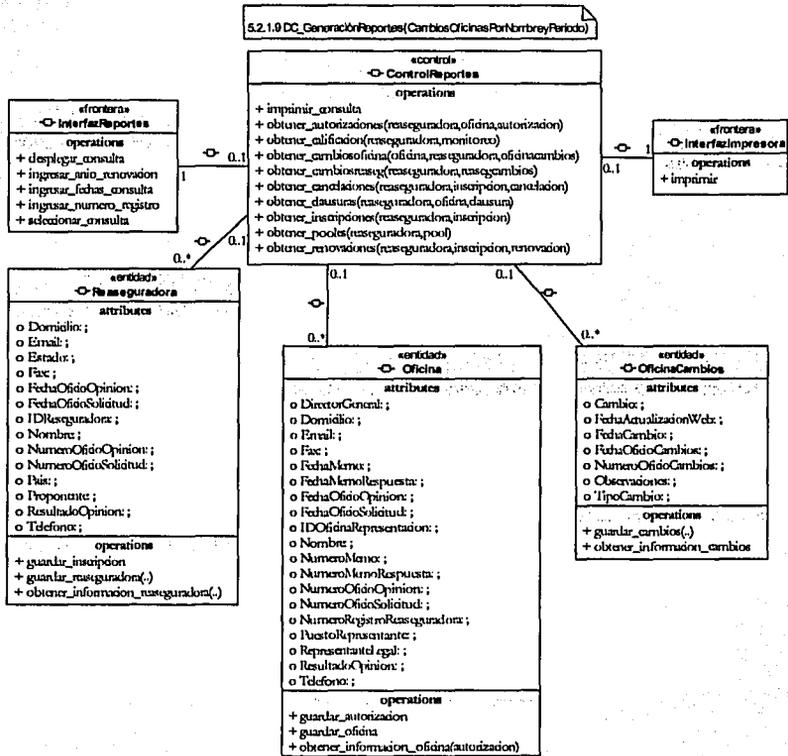
TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

CONCLUSIONES

Al llevar a cabo el análisis de casos de uso del sistema Registro General de Reaseguradoras Extranjeras se pudo constatar que la identificación de los tres tipos de clases que sugiere dicho análisis fue de gran utilidad para comenzar a estructurar el sistema, al permitir la asignación del comportamiento o del funcionamiento que el sistema involucra de una forma mucho más sencilla, clara y ordenada. Cabe mencionar que el grupo de clases propuesto en un principio, fue sufriendo modificaciones conforme se fueron asignando responsabilidades a cada clase, por ejemplo, algunas clases fueron divididas; otras sufrieron cambios en la información que iban a almacenar o algunas otras, simplemente no eran necesarias. Otro aspecto que influyó en la modificación del grupo de clases fue el verificar qué información tenía que registrarse desde un principio y qué información se iba a registrar dependiendo de las condiciones que se cumplieran en algún flujo de algún caso de uso.

Se pudo apreciar que con la elaboración de los diagramas de colaboración se pudo plasmar de una forma muy ordenada y precisa la relación que existe entre las clases participantes en cada caso de uso lo que ayudo a reasignar responsabilidades e incluso a incorporar responsabilidades que no habían sido consideradas lo que nos condujo a obtener una excelente aproximación de la estructura del sistema.

La descripción de responsabilidades jugó un papel muy importante en el análisis de casos de uso ya que por muy sencilla o trivial que fuese la responsabilidad, al describirla, se obtuvo una mejor percepción del comportamiento de cada una de las clases y de la forma en que colaboran entre si.

Debido a que en el sistema no se presentaron asociaciones complejas o que necesitarán de una explicación más a fondo, simplemente se indicaron en los diagramas de clases participantes. En cuanto a los atributos se decidió no entrar a detalle en especificaciones tales como el tipo y la visibilidad del atributo ya que al hacerlo, resultaba mayor el tiempo invertido que la contribución a la descripción del comportamiento del sistema y se sugiere profundizar esta parte en el diseño.

El software empleado en el desarrollo del análisis fue Model Maker. En un principio se presentaron algunas dificultades con el manejo de dicho software pero en la medida en que se fue utilizando se fueron resolviendo las necesidades que se presentaban y en poco tiempo se fueron descubriendo las ventajas o bondades con las que cuenta y terminó por convencer al equipo de desarrollo de la CNSF de que es una buena herramienta con la cual se puede obtener un análisis sumamente controlado y bien documentado y con la que se facilita de forma notoria el comienzo del diseño del sistema de una manera clara y transparente sucediendo lo mismo con la transición del diseño a la implementación.

Resultó un poco difícil el acostumbrarse a dejar de pensar en tablas y registros y comenzar a entender el entorno de los objetos o clases, el asimilar que ahora el funcionamiento y el comportamiento del sistema se asocia a cada una de las clases y ya no de forma directa en funciones o procedimientos y que además en un principio se llegó a pensar que sería muy engorroso trabajar y lidiar con tantos conceptos y elementos del UML. Estas fueron las pocas dificultades que se presentaron durante el desarrollo del análisis de casos de uso.

Por otra parte, el equipo de desarrollo de la CNSF utilizó este trabajo como guía en el aprendizaje de los conceptos básicos del UML y se basó en él, para llevar a cabo el análisis de casos de uso del sistema Validador Único de Información.

Simultáneamente se apoyaron de un libro sobre UML. Los comentarios recibidos fueron que en cuanto a la descripción de lo que es el UML, las dos fuentes son claras y precisas; pero en cuanto a la información relacionada con el análisis de casos de uso, coincidieron en que les fue más claro y sencillo de seguir y de aplicar la serie de pasos descritos en el capítulo dos de este trabajo que la información descrita en el libro, con lo que se cumplió uno de los objetivos: que este trabajo sirva como referencia para el análisis de un sistema usando UML.

BIBLIOGRAFÍA Y REFERENCIAS

Grady Booch, Jlm Rumbaugh, Ivar Jacobson
The Unified Modeling Language User Guide
Addison-Wesley, 1999

Carlos Alberto Fernández y Fernández
Programación Orientada a Objetos II Notas de Clase
Instituto de Electrónica y Computación Universidad Tecnológica de la Mixteca

Fundamentals of Visual Modeling with UML Student Manual Version 2002.05.00
Rational University

Object-Oriented Analysis with UML Student Manual Version 2002.05.00 Volume 1
Rational University

Object-Oriented Analysis with UML Student Manual Version 2002.05.00 Volume 2
Rational University

Antonio Míonzoni Consorti
Reaseguro Financiero y A.R.T.
Facultad de Ciencias, U.N.A.M.

Direcciones de Internet:

www.cnsf.gob.mx

www.offixfiscal.com.mx