

**UNIVERSIDAD NACIONAL DE INGENIERIA**  
**Facultad de Ingeniería Industrial y de Sistemas**



**PROCESO UNIFICADO DE INGENIERIA DE  
SOFTWARE - RATIONAL - Y  
MODELAMIENTO VISUAL AL CASO DE  
ALQUILER Y VENTA DE CINTAS DE VIDEO**

**INFORME DE SUFICIENCIA**

**Para optar el Título Profesional de**

**Ingeniero de Sistemas**

**DORIS RUTH VALERIO SANABRIA**

**Lima – Perú**

**2002**



## DEDICATORIA

A mis adorados padres, autores ,guías y pilares de mi existencia.

A mi amadísimo esposo, apoyo incondicional y constante.

A mis hijos, fuente de inspiración para mi vida.



## **AGRADECIMIENTOS**

A DIOS, por darme la oportunidad de compartir esta obra con todos vosotros.

A mi Alma Mater, por los conocimientos, formación y experiencias impartidas para lograr excelentes profesionales.

A mis queridos maestros, por su dedicación y empeño otorgado con todo honor y ética.

A mi hermano, por el ánimo indismallable para lograr éste objetivo.

**INDICE**

	Descriptores temáticos	5
	Resumen Ejecutivo	6
	Introducción	8
I.	Antecedentes	10
	1. Historia de Rational	10
II.	Marco Teórico	12
	1. Metodología	12
	1.1. Sinopsis	12
	1.2. Que es el Proceso Unificado de Rational	12
	1.3. Utilización efectiva de las seis mejores prácticas	14
III.	Proceso de Toma de decisiones	18
	1. Descripción General del proceso	18
	Fases e iteraciones – La dimensión tiempo	19
	Los nueve flujos de tareas del proceso principal	35
	2. RUP – El Producto	44
	3. Herramientas Integradas que soportan eficientemente el RUP	48
IV.	Evaluación de Resultados	50
	1. Caso Práctico de Modelamiento Visual	51



V.	Conclusiones y Recomendaciones	61
	Bibliografía	62
	Anexos	63
	Diagrama de Casos de Uso	
	Diagrama de Clases – Análisis	
	Diagrama de Clases – Diseño	
	Diagrama de Modelo de Datos.	
	Diagramas de Secuencia.	
	Diagrama de Colaboración.	
	Diagrama de y Estado.	
	Diagrama de Componentes.	
	Diagrama de Despliege	



## DESCRIPTORES TEMÁTICOS

UML

RUP

RATIONAL

MODELAMIENTO VISUAL

ROSE

PROCESO UNIFICADO

ITERATIVO INCREMENTAL

CASO DE USO

DIAGRAMA DE SECUENCIA

DIAGRAMA DE COLABORACION



## RESUMEN EJECUTIVO

El Rational Unified Process (RUP) es un proceso de ingeniería de software. Provee un enfoque ordenado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su virtud principal es asegurar la producción de software de alta calidad, apropiado a las necesidades del usuario final, dentro de un cronograma y un presupuesto predecibles.

RUP es un producto desarrollado y mantenido por Rational Software. Los desarrolladores del RUP están trabajando muy próximos a los clientes, a los socios, a los grupos de productos Rational y sus empresas de consultoría, para garantizar que el RUP esté en continua actualización y crecimiento, incorporando las últimas experiencias, haciendo evolucionar y verificando las mejores prácticas.

El RUP potencia la productividad del equipo, al proveer a cada uno de sus miembros un fácil acceso a una base de conocimiento con lineamientos, templates y guías sobre qué herramientas usar para todas las actividades críticas del desarrollo. Al tener todos los miembros del equipo acceso a la misma base de conocimientos, no importa si se trabaja con requerimientos, diseño, prueba, administración del proyecto o administración de configuración, garantizamos que todos los miembros del equipo comparten un lenguaje común, un proceso común y una visión común de cómo desarrollar software. Las actividades especificadas por RUP crean y mantienen "modelos". Más que apuntar a la producción de una gran cantidad de documentos en papel, el RUP pone el acento en el desarrollo y el mantenimiento de modelos del sistema.

El RUP es una guía sobre como usar efectivamente el UML (Unified Modeling Language) es un lenguaje estándar que permite comunicar claramente requerimientos, arquitectura y diseños. El UML fue creado



originalmente por Rational Software y es ahora mantenido por la organización de estándares Object Management Group (OMG).

El RUP está soportado por herramientas, que automatizan gran parte del proceso. Son usadas para crear y mantener los diversos elementos – particularmente modelos – del proceso de ingeniería de software: modelización visual, programación, testing, etc. Son invaluableles en el apoyo al registro asociado con la administración de cambios, tanto como para la administración de configuración que acompaña a cada iteración.

El RUP es un proceso configurable. No existe un único proceso adecuado para todo el desarrollo de software. El RUP sirve para pequeños equipos de desarrollo tanto como para grandes organizaciones de desarrollo. El RUP está basado en una simple y clara arquitectura de proceso que suministra comunidad a través de una familia de procesos. Puede incluso ser variado para adecuarse a diversas situaciones. Contiene un Kit de desarrollo, que brinda soporte para configurar el proceso de modo de adecuarse a las necesidades de cada organización.

El RUP captura muchas de las mejores prácticas del desarrollo de software moderno, de tal modo que es adecuado a un amplio rango de proyectos y organizaciones. Desplegar estas mejores prácticas – usando el RUP como guía – ofrece a los equipos de desarrollo una cantidad de ventajas claves.





## INTRODUCCIÓN

En la época actual, el éxito de las organizaciones está directamente vinculado con la efectividad del uso de la Tecnología. En este marco, el desarrollo de software es claramente un proceso clave, ya sea ejecutado interna o externamente, con impacto directo en la calidad y disponibilidad de los Productos y Servicios de la Organización.

El Desarrollo y Evolución de Software en las organizaciones se ve permanentemente impactado, por un lado, por la creciente cantidad de requerimientos y, por el otro, por los cambios y variedad de aplicaciones disponibles, ambientes operativos, plataformas, infraestructura de redes, ambientes de desarrollo y lenguajes de programación.

Cada vez más se requiere que el Desarrollo y Evolución de Software logre productos de mayor calidad y en menor tiempo, respondiendo a los requerimientos crecientes, en ambientes tecnológicos cambiantes, con procesos bien definidos, predecibles y en condiciones de ser permanentemente mejorados.

Asimismo, la posibilidad creciente de contratación de soluciones, requiere fortalecer los procesos de adquisición de software y servicios informáticos en todos sus aspectos, desde la evaluación y selección hasta el control de cumplimiento de contratos.

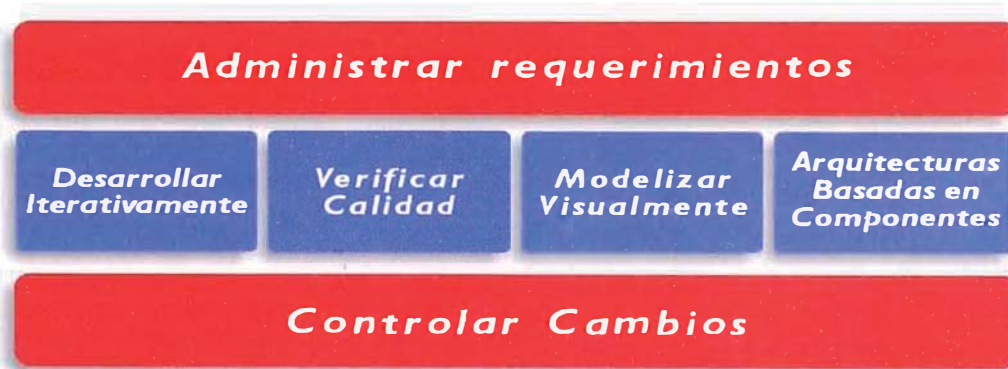
Muchas organizaciones han decidido invertir en mejorar sus procesos de Desarrollo, Evolución y Adquisición de Software, obteniendo excelente retorno al lograr mayor calidad en sus productos y disminuir sus tiempos y costos de implementación.



El Proceso Unificado de Rational (RUP) describe como aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas.

RUP provee a cada miembro del equipo de las guías de proceso, plantillas y mentores de herramientas necesarios para que el team completo tome ventaja de, entre otras, las siguientes mejores prácticas:

1.





## I. ANTECEDENTES

### HISTORIA DE RATIONAL

Rational se fundó respectivamente en 1981 por la Paul Levy y Mike Devlin que son hoy el presidente de la compañía y jefe el funcionario ejecutivo. Rational estaba inspirado por la observación que el software estaba tocando un papel importante como conductor en la economía mundial. Desde entonces, la importancia de software como un artefacto de crecimiento económico mundial y como un diferenciador competitivo para las compañías, servicios, y los productos han crecido a una proporción acelerada.

Rational se fundó con una misión muy clara que ha permanecido inalterado desde ese tiempo: para asegurar el éxito de clientes que dependen de su habilidad desarrollar o desplegar las aplicaciones del software. Nuestra meta es ayudar a clientes a crear los sistemas del software que ellos necesitan lograr sus metas comerciales. Nosotros ayudamos a clientes a resolver los problemas básicos inherente diseñando, desarrollando, probando, y las aplicaciones del software , habilitando la creación de aplicaciones del software más rápidamente y con menos riesgo y calidad más alta y fiabilidad que es posible con otros acercamientos.



Rational envió sus primeros productos en 1984 y ha crecido dramáticamente desde entonces, en la profundidad tecnológica, fuerza financiera, y las personas. Rational tiene más de 26,000 organizaciones individuales como clientes, usando más de 432,000 licencias de sus productos, a partir del 1999 de marzo, y el número está creciendo rápidamente.

Rational ha desarrollado un único, equipo-unificándose solución que combina productos mercado-principales, servicios profesionales, y la dirección intelectual en la ciencia de desarrollo del software. Rational tiene una presencia del campo extensa mundial con profesionales del software experimentados que pueden ayudar a clientes quienes reciben el valor máximo de nuestros productos y principios. Si la habilidad de desarrollar o desplegar las aplicaciones del software es importante a usted, Rational es el mejor compañero capaz ayudarle a conseguir el trabajo hecho.



## II. MARCO TEORICO

### 1. Metodología

Describimos a continuación el Rational Unified Process (RUP) ó Proceso Unificado de Ingeniería de Software.

#### 1.1. Sinopsis

A continuación presentamos una descripción general de RUP. RUP es un proceso de ingeniería de software. Es una base de conocimiento accesible a través de un navegador de Internet. RUP incrementa la productividad de un equipo de desarrollo y permite la difusión de las mejores prácticas a través de lineamientos, templates y guías sobre qué herramientas usar a lo largo del ciclo de vida de las actividades. RUP utiliza el lenguaje de modelización unificado (UML) estándar de la industria

#### 1.2. Qué es el Proceso Unificado de Rational ?

El RUP es un proceso de ingeniería de software. Provee un enfoque ordenado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su virtud principal es asegurar la producción de software de alta calidad, apropiado a las necesidades del usuario final, dentro de un cronograma y un presupuesto predecibles.

RUP es un producto desarrollado y mantenido por RATIONAL SOFTWARE. Los desarrolladores de RUP están trabajando muy próximos a los clientes, a los socios, a los grupos de productos RATIONAL y sus empresas de consultoría, para garantizar que el RUP esté en continua actualización y



crecimiento, incorporando las últimas experiencias, haciendo evolucionar y verificando las mejores prácticas.

El RUP potencia la productividad del equipo, al proveer a cada uno de sus miembros un fácil acceso a una base de conocimiento con lineamientos, templates y guías sobre qué herramientas usar para todas las actividades críticas del desarrollo. Al tener todos los miembros del equipo acceso a la misma base de conocimientos, no importa si se trabaja con requerimientos, diseño, prueba, administración del proyecto o administración de configuración, garantizamos que todos los miembros de equipo comparten un lenguaje común, un proceso común y una visión común de cómo desarrollar software-

Las actividades especificadas por RUP crean y mantienen "modelos". Más que apuntar a la producción de una gran cantidad de documentos en papel, el RUP pone el acento en el desarrollo y el mantenimiento de modelos del sistema.

El RUP es una guía sobre como usar efectivamente el UML. El UML es un lenguaje estándar que permite comunicar claramente requerimientos, arquitectura y diseños. El UML fue creado originalmente por Rational Software, y es ahora mantenido por la organización de estándares Object Management Group (OMG).

El RUP está soportado por herramientas, que automatizan gran parte del proceso. Son usadas para crear y mantener los diversos elementos – particularmente modelos – del proceso de ingeniería de software: modelización visual, programación, testing , etc. Son invaluableles en el apoyo a la registración asociada con la administración de cambios, tanto como para la administración de configuración que acompaña a cada iteración.



El RUP es un proceso configurable. No existe un único proceso adecuado para todo el desarrollo de software. El RUP sirve para pequeños equipos de desarrollo tanto como para grandes organizaciones de desarrollo. El RUP está basado en una simple y clara arquitectura de proceso que suministra comunidad a través de una familia de procesos. Puede incluso ser variado para adecuarse a diversas situaciones. Contiene un Kit de Desarrollo, que brinda soporte para configurar el proceso de modo de adecuarse a las necesidades de cada organización. El RUP captura muchas de las mejores prácticas del desarrollo de software moderno, de tal modo que es adecuado a un amplio rango de proyectos y organizaciones. Desplegar estas mejores prácticas – usando el RUP como guía - ofrece a los equipos de desarrollo una cantidad de ventajas clave. En la próxima sección, describimos las mejores prácticas del RUP:

### 1.3. Utilización efectiva de las 6 mejores prácticas

El RUP describe cómo instalar efectivamente enfoques probados de desarrollo de software. Son llamadas "mejores prácticas" no tanto porque se pueda cuantificar precisamente su valor, sino más bien porque son usadas comúnmente en la industria por organizaciones exitosas. El RUP suministra a cada equipo lineamientos, templates y guías sobre herramientas necesarias para que el equipo completo tome las ventajas, entre otras, de las siguientes mejores prácticas:

- Desarrollar software iterativamente
- Administrar requerimientos
- Usar arquitecturas basadas en componentes
- Modelizar software visualmente



- Verificar la calidad de software
- Controlar los cambios al software

### 1.3.1. Desarrollar software iterativamente

Dada la actual sofisticación de los sistemas, no es posible definir primero secuencialmente el problema completo, diseñar las soluciones completas, construir el software y al final realizar la prueba del producto. Se requiere un enfoque iterativo que permita una comprensión creciente del problema a través de refinamientos sucesivos, y de hacer crecer en forma incremental una solución efectiva en múltiples iteraciones. El RUP soporta una aproximación iterativa al desarrollo que indica los mayores riesgos en cada etapa del ciclo de vida, reduciendo significativamente el perfil de riesgo del proyecto. Este enfoque iterativo ayuda a atacar el riesgo a través de progresos demostrables, liberación de versiones que permiten el compromiso y la retroalimentación del usuario final. Puesto que cada iteración acaba con una liberación de versión, el equipo de desarrollo permanece focalizado en producir resultados y los controles frecuentes de estado ayudan a garantizar que el proceso sigue en tiempo. Una enfoque iterativo facilita también los reajustes tácticos de requerimientos y características del cronograma.

### 1.3.2. Administrar requerimientos

El RUP describe como relevar, organizar y documentar la funcionalidad requeridas; documentar alternativas y decisiones, y comunicar los requerimientos del negocio. Las nociones de casos de uso y escenarios utilizados en el proceso han demostrado ser una manera excelente de capturar los requerimientos funcionales y para garantizar que éstos conduzcan al diseño, implementación y testeado del software, en forma similar a lo que hará finalmente el sistema completo para satisfacer las





necesidades del usuario. Proveen caminos coherentes y rastreables a la vez a través del desarrollo y del sistema entregado.

### 1.3.3. Usar arquitecturas basadas en componentes

El proceso se centra en el temprano desarrollo de una arquitectura robusta ejecutable, antes de comprometer recursos para un desarrollo a escala completa. Describe como diseñar una arquitectura elástica que es flexible, se adecua a los cambios, es intuitivamente comprensible, y promueve el más efectivo reuso del software. El RUP soporta el desarrollo de software basado en componentes. Los componentes son módulos no triviales, subsistemas que cumplen una función clara. El RUP provee una aproximación sistemática para definir una arquitectura que use componentes nuevos y existentes. Son ensamblados en una arquitectura bien definida, incluso ad hoc, o en una infraestructura de componentes tales como Internet, CORBA, y COM, para las cuales está emergiendo una industria de componentes reusables.

### 1.3.4. Modelizar software visualmente

El proceso muestra como modelizar visualmente software para capturar la estructura y el comportamiento de las arquitecturas y componentes. Esto permite esconder los detalles y escribir código usando "Building blocks gráficos". Las abstracciones visuales ayudan a comunicar diferentes aspectos del software, ver como los elementos del software se adaptan unos a otros, asegurar que los bloques constructivos son consistentes con el código, mantiene la consistencia entre un diseño y su implementación, y promueve comunicación inequívoca. El UML estándar, creado por Rational Software, es el cimiento para una modelización visual exitosa.



### 1.3.5. Verificar la calidad de software

Pobre performance de las aplicaciones y pobre confiabilidad son factores comunes que inhiben dramáticamente la aceptabilidad de las aplicaciones de software actuales. Por consiguiente, la calidad debe ser revisada respecto de los requerimientos de confiabilidad, funcionalidad, performance de las aplicaciones y performance del sistema. El RUP asiste en la planificación, diseño, implementación, ejecución y evaluación de estos tipos de testeo. El aseguramiento de la calidad es construido dentro del proceso, en todas las actividades, implicando a todos los participantes, usando medidas y criterios objetivos, y no tratándolo como una actividad aislada o separada realizada por un grupo diferente.

### 1.3.6. Controlar los cambios al software

La habilidad para administrar los cambios --dando certeza de que cada cambio es aceptable, y ser capaz de rastrear los cambios – es esencial en un ambiente en el cual el cambio es inevitable. El proceso describe como controlar, rastrear y monitorear los cambios para permitir un desarrollo iterativo exitoso. También guía sobre como establecer espacios de trabajo seguros para cada desarrollador suministrando aislamiento de los cambios hechos en otros espacios de trabajo y controlando los cambios de todos los elementos de software (por ejemplo, modelos, código, documentos, etc.). Y hace trabajar a un equipo en conjunto como una única unidad describiendo como automatizar la integración y construir la administración.



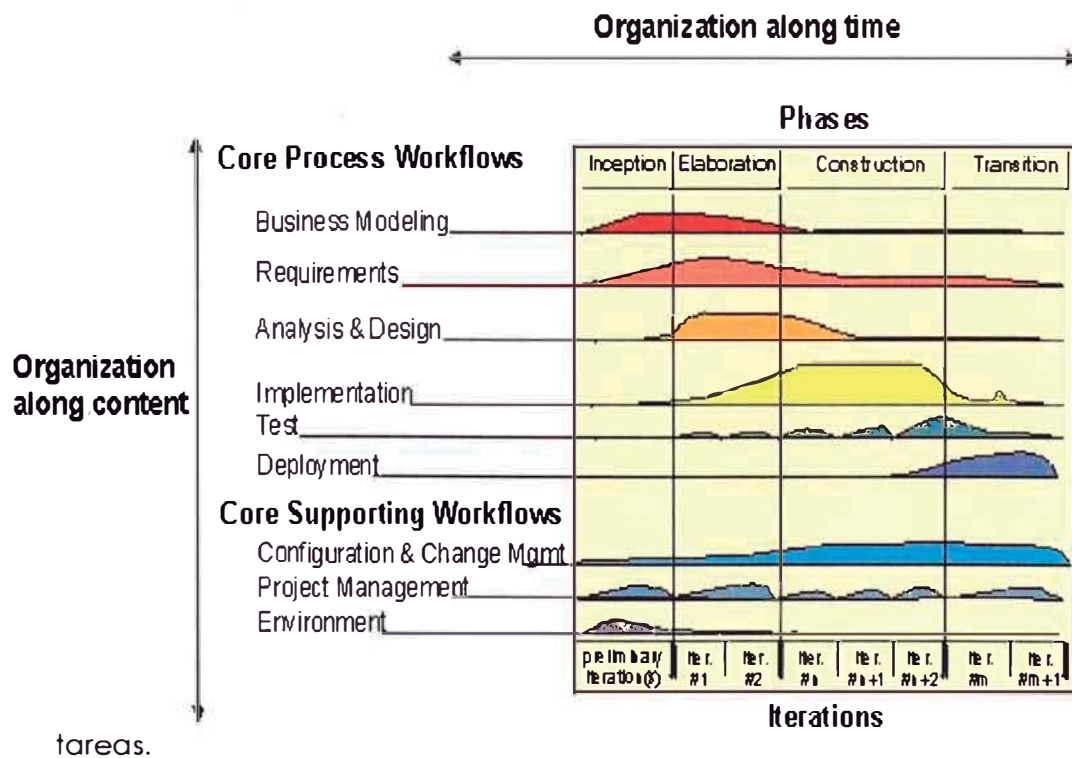
### III. PROCESO DE TOMA DE DECISIONES

#### 1. Descripción general del Proceso

##### Dos dimensiones

El proceso puede ser descrito en dos dimensiones, o a lo largo de dos ejes:

- El eje horizontal representa el tiempo y muestra el aspecto dinámico del proceso tal como se realiza, y está expresado en términos de ciclos, fases, iteraciones y puntos de control.
- El eje vertical representa el aspecto estático del proceso: como está descrito en términos de actividades, elementos, desarrolladores y flujos de tareas.





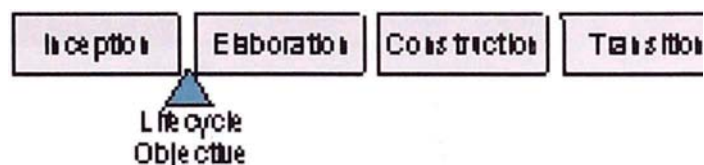
**El gráfico del modelo iterativo muestra como el proceso está estructurado en dos dimensiones.**

### **Fases e iteraciones – La dimensión tiempo**

Esto es la organización dinámica del proceso a lo largo del tiempo.

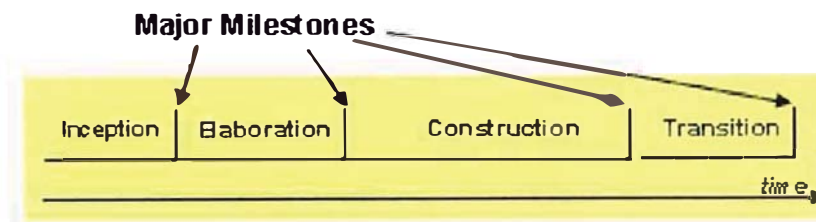
El ciclo de vida del software está partido en ciclos, y cada ciclo trabaja sobre una nueva generación del producto.

El RUP divide cada ciclo de desarrollo en cuatro fases consecutivas.



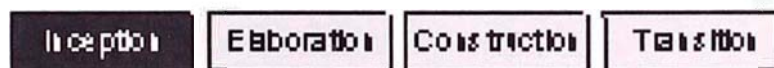
- Fase de conceptualización
- Fase de elaboración
- Fase de construcción
- Fase de transición

Cada fase concluye con un punto de control bien definido – un punto en el tiempo en el cual ciertas decisiones críticas deben ser tomadas, y por lo tanto deben haber sido alcanzadas metas clave.



### Las fases y principales puntos de control en el proceso

Cada fase tiene un objetivo específico.



### Fase de conceptualización

Durante esta fase se establece el caso de negocio para el sistema y se limita el alcance del proyecto. Para cumplir esto, se debe identificar todas las entidades externas con las cuales el sistema interactuará (actores) y se define la naturaleza de esta interacción en un nivel alto. Esto incluye identificar todos los casos de uso y describir algunos significativos. El caso de negocio incluye criterios de éxito, riesgo, análisis y estimación de los recursos necesarios, y un plan de fase que muestre las fechas de los principales puntos de control.

El resultado de la fase de conceptualización es:

- Un documento panorámico: una visión general de los requerimientos esenciales del proyecto, características clave, y principales exigencias.
- Un modelo de caso de uso inicial (completo al 10 / 20%)



- Un glosario inicial del proyecto (puede ser opcionalmente expresado como un modelo de dominio)
- Un caso de negocio inicial, que incluya contexto del negocio, criterios de éxito (proyección de ganancias, reconocimiento del mercado, etc.) y presupuesto financiero.
- Una determinación inicial de riesgo
- Un plan del proyecto, que muestre fases e iteraciones
- Un modelo del negocio, si es necesario
- Uno o varios prototipos.

#### **Punto de control: Objetivos del ciclo de vida**

Al final de esta fase de conceptualización está el principal punto de control del proyecto : objetivos del ciclo de vida. Los criterios de evaluación para esta fase de conceptualización son:

- Acuerdo entre los participantes sobre la definición de alcance y costo/cronograma estimado.
- Comprensión de los requerimientos evidenciados por la fidelidad a los casos de uso primarios.
- Credibilidad del costo/cronograma estimado, prioridades, riesgos, y proceso de desarrollo.
- Profundidad y amplitud de cada prototipo de arquitectura tal como fue desarrollado.
- Erogaciones reales versus erogaciones planeadas.



## Fase de elaboración



El propósito de la fase de elaboración es analizar el dominio del problema, establecer una base de arquitectura sólida, desarrollar el plan del proyecto, y eliminar los mayores elementos de riesgo del proyecto. Para alcanzar estos objetivos, se debe tener una visión del sistema "de un kilómetro de ancho y un centímetro de profundidad". Las decisiones sobre arquitectura deben ser hechas con comprensión del sistema completo, su alcance, su funcionalidad principal y sus requerimientos no funcionales tales como requerimientos de performance.

La fase de elaboración es la más crítica de las cuatro. Mientras que el proceso debe siempre considerar los cambios, las actividades de la fase de elaboración garantizan que la arquitectura, los requerimientos y los planes están suficientemente estables, y el riesgo suficientemente mitigado, como para poder determinar previsiblemente el costo y el cronograma para completar el desarrollo.

En la fase de elaboración, se construye un prototipo de arquitectura ejecutable en una o más iteraciones, dependiendo del alcance, la magnitud y el riesgo del proyecto. Este esfuerzo debería indicar por lo menos casos de uso identificados en la fase de conceptualización, los que exponen típicamente los principales riesgos técnicos del proyecto. Mientras un prototipo evolutivo de un componente producido con calidad es siempre la meta, esto no excluye el desarrollo de uno más prototipos exploratorios, descartables para mitigar riesgos específicos tales

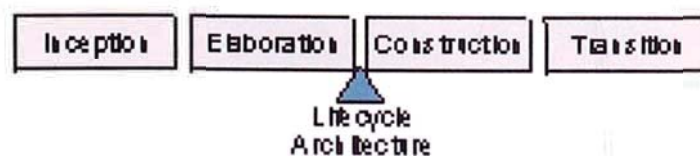


como negociaciones de diseño/requerimientos, estudio de factibilidad de componentes, o demostraciones a los inversores, clientes y usuarios finales.

El resultado de la fase de elaboración es:

- Un modelo de caso de uso (completo por lo menos en un 80%) – habiendo sido ya identificados todos los casos de uso y actores, y habiendo ya desarrollado la descripción de la mayoría de los casos de uso
- Requerimientos suplementarios que capturen los requerimientos no funcionales y cualquier requerimiento que no esté asociado a un caso de uso específico.
- Una descripción de la arquitectura de software
- Un prototipo de arquitectura ejecutable
- Una lista de riesgos revisada y el caso de negocio revisado
- Un plan de desarrollo para todo el proyecto, incluyendo el plan global, que muestre iteraciones y criterios de evaluación para cada iteración
- Especificar el proceso de desarrollo que se usará
- Un manual de usuario preliminar

#### **Punto de control: arquitectura del ciclo de vida**



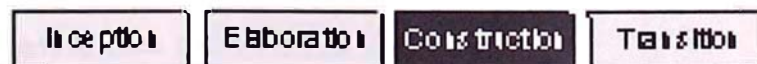




En el final de la fase de elaboración está el segundo punto de control importante del proyecto, El punto del control de la arquitectura del ciclo de vida. En este punto, se examina detalladamente los objetivos y alcances del sistema, la elección de la arquitectura y la resolución de los principales riesgos.

Los principales criterios de evaluación para la fase de elaboración incluyen las respuestas a estas preguntas:

- Es estable la visión del producto?
- Es estable la arquitectura?
- Muestra la demostración ejecutable que los elementos principales de riesgo han sido indicados y resueltos?
- Está suficientemente detallado y afinado el plan para la fase de construcción? Está apoyado en bases de estimación creíbles?
- Todos los participantes están de acuerdo en que se puede obtener la visión actual si se ejecuta el plan para desarrollar el sistema completo, en el contexto de la arquitectura actual?



### **Fase de construcción**

Durante la fase de construcción, todos los componentes restantes y características de la aplicación son desarrollados e integrados al producto, y todas sus funcionalidades son enteramente testeadas. Esta fase de construcción es, en un sentido, un proceso de manufactura en el



cual se pone el acento en la administración de recursos y el control de las operaciones para optimizar costos, tiempos y calidad. En este sentido, la atención se traslada del desarrollo de la propiedad intelectual durante la conceptualización y elaboración, al desarrollo de productos instalables durante la construcción y la transición.

En muchos proyectos es conveniente poder realizar actividades en paralelo. Estas actividades paralelas pueden acelerar significativamente la disponibilidad de versiones instalables; pueden también incrementar la complejidad de la administración de recursos y la sincronización del flujo de tareas. Una arquitectura robusta y un plan comprensible están altamente relacionados. En otras palabras, una de las cualidades críticas de la arquitectura es su facilidad de construcción. Esta es una de las razones por las cuales el desarrollo balanceado de la arquitectura y el plan son acentuados durante la fase de elaboración.

El resultado de la fase de construcción es un producto listo para ser puesto en manos del usuario final. Consiste, como mínimo, en:

- El producto de software integrado en las plataformas adecuadas
- Los manuales del usuario
- Una descripción de la versión vigente.

#### **Punto de Control: Capacidad Operativa Inicial**





Al final de la fase de construcción está el tercer principal punto de control del proyecto (Punto de Control: Capacidad Operativa Inicial). En este punto, se decide si el software, los lugares y los usuarios están listos para estar operativos, sin exponer el proyecto a altos riesgos. Esta versión es llamada a menudo versión "beta".

Los criterios de evaluación para la fase de construcción incluyen la respuesta a estas preguntas:

- Es este release del producto suficientemente maduro y estable para ser instalado en la comunidad usuaria?
- Están todos los recursos listos para la transición hacia la comunidad usuaria?



### Fase de transición

El propósito de la fase de transición es transferir el sistema a la comunidad usuaria. Una vez que el sistema fue entregado al usuario final, habitualmente surgen cuestiones que requieren desarrollo de nuevas versiones, corrección de ciertos problemas, o conclusión de facilidades que habían sido pospuestas.

Se ingresa en la fase de transición cuando un release está suficientemente maduro para ser instalado en el dominio del usuario final. Esto requiere típicamente que algún subconjunto utilizable del sistema haya sido completado en un aceptable nivel de calidad y que la



documentación del usuario esté disponible de modo que la transición al usuario dé resultados positivos a todas las partes. Esto incluye:

- "Beta testing" para validar el nuevo sistema contra las expectativas del usuario
- Operación paralela con un sistema heredado que está siendo reemplazado
- Conversión de las bases de datos operacionales
- Entrenamiento de usuarios y del equipo de mantenimiento

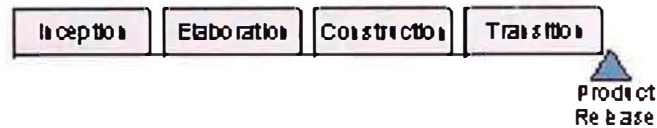
La fase de transición se centra en las actividades requeridas para poner el software en manos de los usuarios. Típicamente, esta fase incluye varias iteraciones, incluyendo versiones beta, versiones de disponibilidad general, tanto como reparación de errores y versiones de mejoramiento. Se consume considerable esfuerzo en desarrollar la documentación orientada al usuario, entrenamiento de usuarios, apoyo a los usuarios durante su utilización inicial del sistema, y reaccionar ante la retroalimentación del usuario. En este punto del ciclo de vida, sin embargo, la retroalimentación del usuario debe ser limitada a cuestiones de sintonía, configuración, instalación y utilizabilidad.

Los objetivos primarios de la fase de transición incluyen:

- Obtener la autonomía del usuario
- Obtener el acuerdo de los participantes de que la instalación ha sido completa y que es consistente con los criterios de evaluación de la visión.
- Perfeccionar el producto final.



### Punto de control: Release del Sistema



Al final de la fase de transición está el cuarto punto importante de control del proyecto, el Punto de Control de Release del Sistema. En este punto se decide si los objetivos han sido alcanzados, y si se podría comenzar otro ciclo de desarrollo. En algunos casos, este punto de control puede coincidir con el final de la fase de conceptualización para el próximo ciclo.

Los criterios de evaluación primarios para la fase de transición implican la respuesta a la siguiente pregunta:

- Está satisfecho el usuario?

### Iteraciones

Cada fase del RUP puede ser descompuesta en iteraciones. Una iteración es un ciclo completo de desarrollo, que da como resultado una versión (interna o externa) de un producto ejecutable, un subconjunto del producto final en desarrollo, que crece de modo incremental de iteración en iteración para llegar a ser el sistema final.

### Beneficios del enfoque iterativo

Comparado con el tradicional proceso en cascada, el proceso iterativo tiene las siguientes ventajas:

Los riesgos se mitigan más temprano

El cambio es más manejable

Alto nivel de reuso

El equipo de proyecto puede aprender a lo largo del camino

Mejor calidad global

### Estructura estática del proceso

Un proceso describe quién está haciendo qué, cómo y cuándo. El RUP se representa utilizando cuatro elementos principales de modelización:

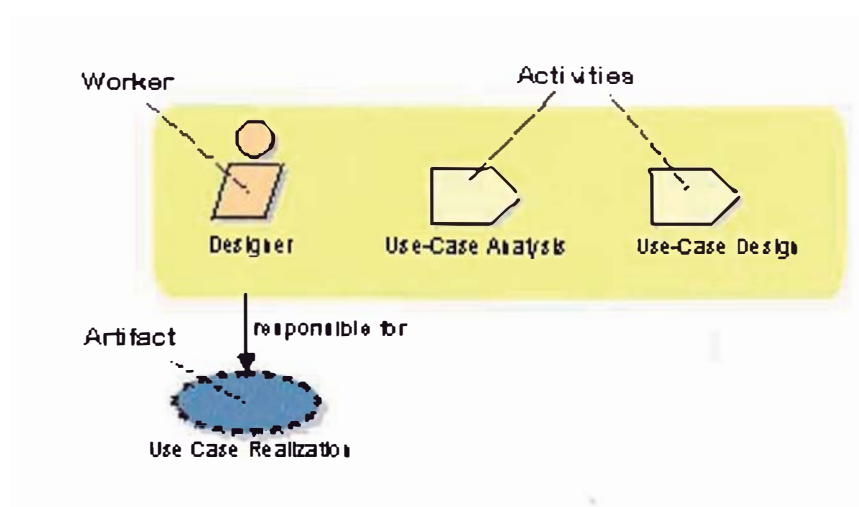
Desarrolladores, el "quién"

Actividades, el "cómo"

Elementos, el "qué"

Flujos de tarea, el "cuándo"

### Actividades, elementos y desarrolladores

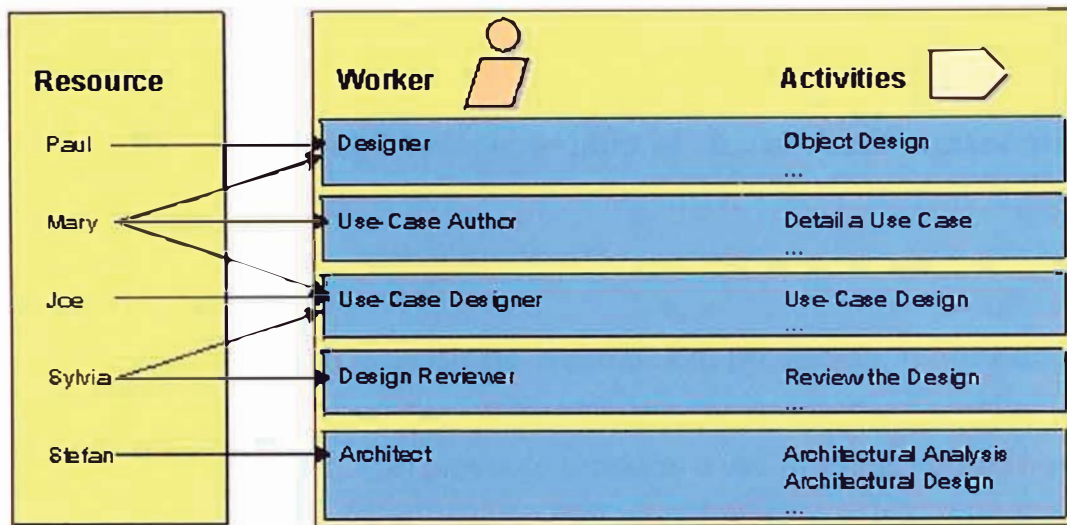




## **Desarrolladores, actividades y elementos**

### **Desarrollador**

*Desarrollador* define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos que trabajan en conjunto como un equipo. Se puede considerar un desarrollador como un "sombrero" que un individuo puede usar en el proyecto. Un individuo puede usar varios sombreros diferentes. Esta es una distinción importante, pues es natural pensar en un desarrollador como el individuo o el equipo mismo, pero en el RUP el desarrollador es más bien el rol que define como los individuos deben realizar su trabajo. Las responsabilidades que asignamos a un desarrollador incluyen dos cosas: realizar un cierto conjunto de actividades y ser propietario de un conjunto de elementos.



## Gente y desarrolladores

### Actividad

Una *actividad* de un desarrollador específico es una unidad de trabajo que se le puede pedir que realice en ese rol. La actividad tiene un propósito claro, expresado usualmente en términos de crear o actualizar algún elemento, tal como modelo, clase o plan. Cada actividad es asignada a un desarrollador específico. La granularidad de una actividad va generalmente de unas pocas horas a unos pocos días, incluye usualmente a un desarrollador, y afecta uno o un pequeño número de elementos. Una actividad puede ser usada como un elemento para planificación y avance; si es demasiado pequeña, será despreciada, y si es demasiado grande, el avance deberá ser expresado en términos de partes de actividad.

Ejemplo de actividades:

- Planificar una iteración, para el Desarrollador: Project Manager
- Encontrar casos de uso y actores, para el Desarrollador: Analista de Sistemas





- Revisar el diseño, para el Desarrollador: Revisor de Diseño
- Ejecutar pruebas de performance, para el Desarrollador: Testeador de performance

### **Elemento**

Un *elemento* es una porción de información, producida, modificada o usada por un proceso. Los elementos son los productos tangibles del proyecto, las cosas que el proyecto produce o usa mientras trabaja hacia el producto final. Los elementos son usados como input por desarrolladores para realizar un actividad, y son el resultado o output de tales actividades. En términos de diseño orientado a objetos, así como las actividades son operaciones sobre un objeto activo (el desarrollador), los elementos son los parámetros de tales actividades.

Los elementos pueden tomar varios perfiles o formas:

- Un modelo, tal como el Modelo de Caso de Uso o el Modelo de diseño
- Un elemento dentro de un modelo, tal como una clase, un caso de uso o un subsistema
- Un documento tal como un Caso de Negocio o un Documento de Arquitectura de Software
- Código fuente
- Ejecutables

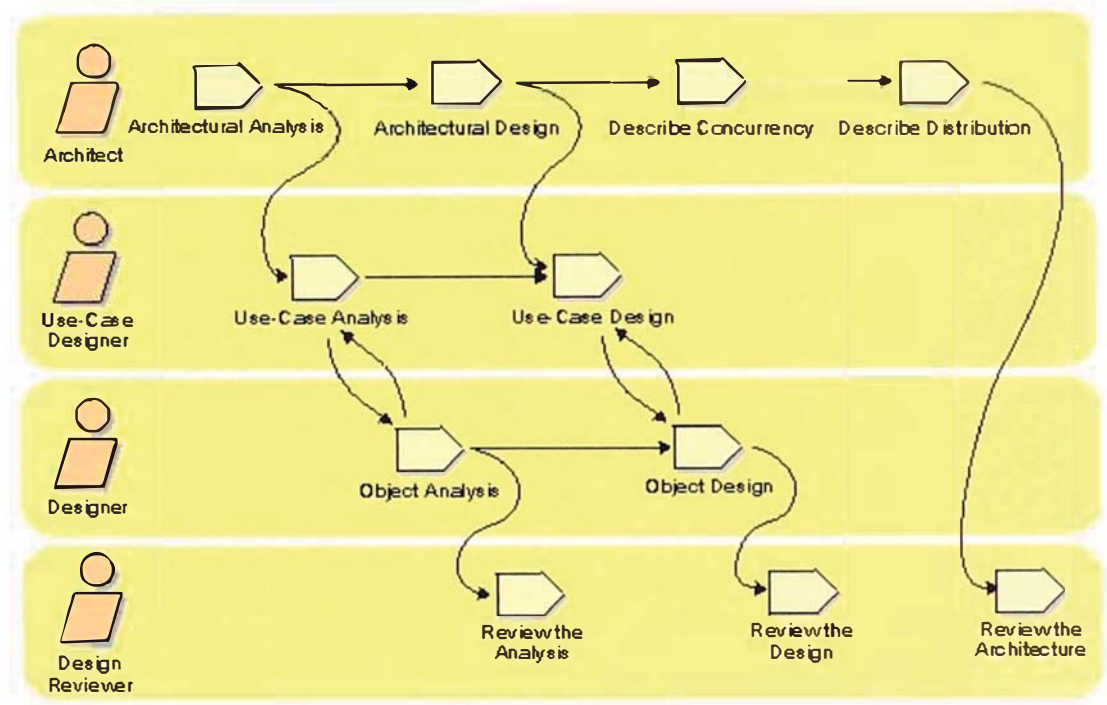


### **Flujos de tareas**

Una mera enumeración de todos los desarrolladores, actividades y elementos no constituye realmente un proceso. Es necesaria una manera de describir comprensiblemente secuencias de actividades que produzcan algún resultado que se pueda evaluar, y muestre las interacciones entre los desarrolladores.

Un *flujo de tareas* es una secuencia de actividades que produce un resultado de valor observable.

En términos de UML, un flujo de tareas puede ser expresado en un diagrama secuencial, un diagrama de colaboración, o un diagrama de actividades. Usamos una forma de diagramas de actividad en este documento.

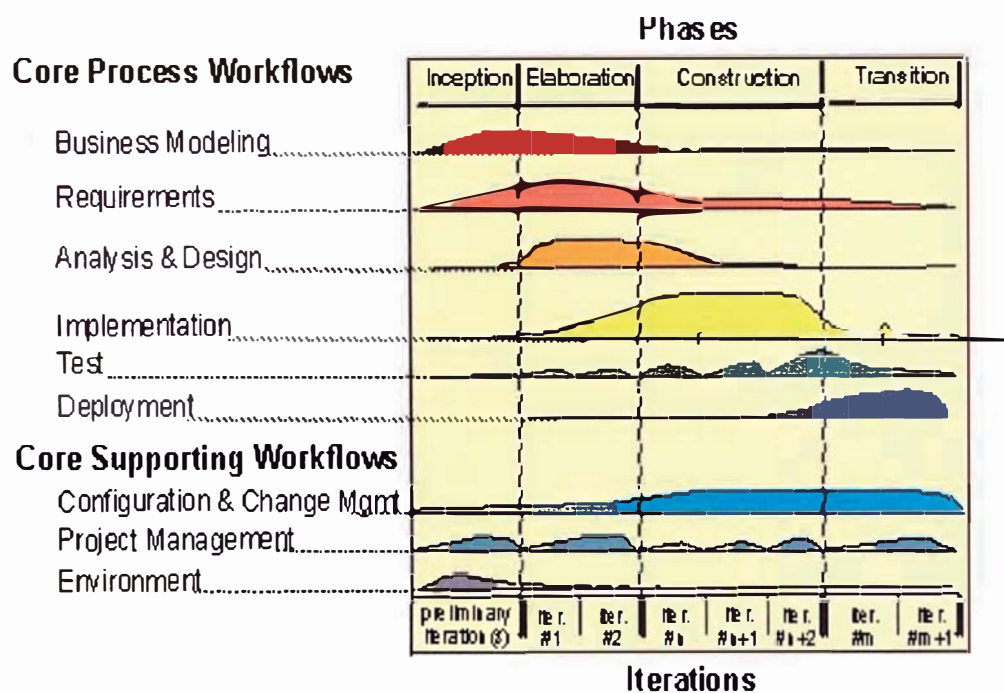


### Ejemplo de flujo de tareas

Nótese que no siempre es posible o práctico representar todas las dependencias entre actividades. A menudo dos actividades están más estrechamente entrelazadas de lo que se representa, especialmente cuando incluyen al mismo desarrollador o al mismo individuo. Las personas no son máquinas, y el flujo de tareas no puede ser interpretado literalmente como un programa para la gente, que debe ser seguido exacta y mecánicamente. En la próxima sección trataremos sobre el tipo más esencial de flujo de tareas en el proceso, llamado Flujo de tareas principal.

### Flujo de tareas principal

Hay nueve flujos de tareas principales en el RUP, que representan un particionamiento de todos los desarrolladores y actividades en grupos lógicos.



### Los nueve flujos de tareas del proceso principal

Los flujos de tarea del proceso principal están divididos en seis flujos de tareas principales "de ingeniería":

- Flujo de tareas de modelización del negocio
- Flujo de tareas de requerimientos
- Flujo de tareas de análisis y diseño
- Flujo de tareas de implementación
- Flujo de tareas de testeo
- Flujo de tareas de instalación

Y tres flujos de tareas principales "de apoyo"



- Flujo de tareas de administración del proyecto
- Flujo de tareas de configuración y administración de cambios
- Flujo de tareas de ambiente

Aunque los nombres de los seis flujos de tareas principales "de ingeniería" puedan evocar las fases secuenciales de un proceso tradicional en cascada, debemos tener presente que las fases de un proceso iterativo son diferentes y que estos flujos de tareas se repasan una y otra vez en el curso del ciclo de vida. El real flujo de tareas completo de un proyecto vincula a estos nueve flujos de tareas núcleo, y los repite con diverso énfasis e intensidad en cada iteración.

### **Modelización del Negocio**

Uno de los problemas principales en muchos proyectos es que la comunidad de ingeniería del software y la comunidad de ingeniería del negocio no se comunican adecuadamente entre sí. Esto conduce a que el output de la ingeniería de negocio no se utiliza apropiadamente como input del esfuerzo de desarrollo de software, y viceversa. El RUP rectifica esto, proveyendo un lenguaje y un proceso comunes para ambas comunidades, así como les muestra el modo de crear y mantener directa rastreabilidad entre los modelos de negocio y de software.

En Modelización del Negocio documentamos los procesos de negocios utilizando los llamados casos de uso del negocio. Esto asegura una comprensión común entre todos los protagonistas sobre lo que el proceso de negocio necesita para ser soportado en la organización. Los casos de uso del negocio son analizados para comprender como el negocio debe



soportar los procesos de negocios. Esto está documentado en un modelo de objetos.

Muchos proyectos pueden optar por no hacer modelización del negocio.

### **Requerimientos**

La meta del flujo de tareas de Requerimientos es describir lo que el sistema debería hacer y permite a los desarrolladores y clientes ponerse de acuerdo sobre esta descripción. Para conseguirlo relevamos, organizamos y documentamos la funcionalidad requerida y las restricciones y documentamos negociaciones y decisiones.

Se crea un documento de Visión, y las necesidades de los protagonistas son investigadas. Se identifica a los actores, representando a los usuarios, y a todo otro sistema que pueda interactuar con el sistema en desarrollo. Los casos de uso están identificados, representando el comportamiento del sistema. Puesto que los casos de uso están desarrollados según las necesidades de los actores, el sistema tiene más posibilidades de ser relevante para los usuarios.

Cada caso de uso es descrito en detalle. La descripción del caso de uso muestra como el sistema interactúa paso a paso con los actores y lo que el sistema hace. Los requerimientos no-funcionales son descritos en Especificaciones Suplementarias.

Los casos de uso funcionan como unificadores a través del ciclo de desarrollo del SW sistema. El mismo modelo de caso de uso es usado durante la captura de requerimientos, análisis y diseño, y testeo.



## **Análisis y Diseño**

La meta del flujo de tareas de Análisis y Diseño es mostrar como el sistema será realizado en la fase de implementación. Se quiere construir un sistema que:

- Realice – en un ambiente de implementación específico – las tareas y funciones especificadas en las descripciones de casos de uso.
- Cumpla con todos los requerimientos
- Esté estructurado para ser robusto y flexible

Análisis y Diseño da como resultado un modelo de diseño y opcionalmente un modelo de análisis. El modelo de diseño sirve como abstracción del código fuente; esto es, el modelo de diseño actúa como un "plano" de cómo el código fuente está estructurado y escrito.

El modelo de diseño consiste en clases de diseño estructuradas en paquetes de diseño y subsistemas de diseño con interfases bien definidas, que representan aquello en lo que se convertirán los componentes en la implementación. También contiene descripciones de cómo objetos de estas clases de diseño colaboran para realizar los casos de uso.

Las actividades de diseño están centradas en la noción de arquitectura. La producción y validación de esta arquitectura es el foco principal de las primeras iteraciones de diseño. La arquitectura está representada por una cantidad de vistas arquitectónicas. Estas vistas capturan las principales decisiones de diseño estructurales. En esencia, las vistas arquitectónicas son abstracciones o simplificaciones del diseño total, en las cuales características importantes se hacen más visibles dejando detalles de lado. La arquitectura es un vehículo importante no sólo para desarrollar un



buen modelo de diseño, sino también para aumentar la calidad de cada modelo construido durante el desarrollo del sistema.

### **Implementación**

Los propósitos de la implementación son:

- Definir la organización del código, en términos de subsistemas de implementación organizados en capas
- Implementar clases y objetos en términos de componentes (archivos fuente, binarios, ejecutables, y otros)
- Testear los componentes desarrollados como unidades
- Integrar los resultados producidos por implementadores individuales (o equipos) en un sistema ejecutable.

El sistema se realiza a través de la implementación de componentes. El RUP describe como reusar componentes existentes, o implementar componentes nuevos con responsabilidades bien definidas, haciendo que el sistema sea más fácil de mantener, y aumentando las posibilidades de reuso.

Los componentes están estructurados en Subsistemas de Implementación. Los Subsistemas toman la forma de directorios, con información adicional estructural o de administración. Por ejemplo, un subsistema puede ser creado como un directorio o una carpeta en un sistema de archivos, o un subsistema en Rational/Apex para C++ o Ada, o paquetes que usen Java.

### **Testeo**

Los propósitos del testeo son:

Verificar la interacción entre objetos





Verificar la apropiada integración de todos los componentes del software

Verificar que todos los requerimientos hayan sido correctamente implementados

Identificar y garantizar que los defectos están corregidos antes de la instalación del software.

El RUP propone un enfoque iterativo, lo que significa que se prueba a lo largo del proyecto. Esto permite encontrar los defectos lo más temprano posible, lo que reduce radicalmente el costo de repararlos. Los tests son realizados a lo largo de tres dimensiones de calidad : confiabilidad, funcionalidad y performance del sistema. Para cada una de estas dimensiones de calidad, el proceso describe como se prueba a lo largo del ciclo de vida : la planificación, el diseño, la implementación, la ejecución y la evaluación.

Se describen estrategias con respecto a cuando y como automatizar el testeo. La automatización del testeo es especialmente importante al usar un enfoque iterativo, para permitir el testeo de regresión al final de cada iteración y para cada nueva versión del producto.

### **Instalación**

El propósito del flujo de tareas de instalación es producir exitosamente releases del producto, y entregar el software a sus usuarios finales. Cubre un amplio rango de actividades, entre ellos:

- Producir releases externas del software
- Empaquetar software
- Distribuir el software
- Instalar el software



- Suministrar ayuda y asistencia a los usuarios

En muchos casos, incluye actividades tales como:

- Planificación y conducción de beta testing
- Migración de software o datos existentes
- Aceptación formal

Aunque las actividades de instalación están mayormente centradas alrededor de la fase de transición, muchas de las actividades deben ser incluidas en fases tempranas, para prepararse para la instalación al final de la fase de construcción.

Los flujos de tareas de Instalación y Ambiente del RUP incluyen menos detalles que otros flujos de tareas.

### **Administración del proyecto**

La Administración de Proyectos de Software es el arte de equilibrar objetivos opuestos, administrar el riesgo y superar exigencias para entregar, exitosamente, un producto que satisfaga a la vez las necesidades de clientes y los usuarios. El hecho de que tan pocos proyectos sean indiscutiblemente exitosos dice bastante sobre la dificultad de esta tarea.

Este flujo de tareas enfoca principalmente el aspecto específico de un proceso iterativo de desarrollo. Nuestra meta, en esta sección, es facilitar la tarea proveyendo:

- Una estructura para administrar proyectos de software intensivos
- Lineamientos para planificar, proveer de personal, ejecutar y monitorear proyectos



- Una estructura para administrar el riesgo.

### **Administración de Configuración y Administración de Cambios**

En este flujo de tareas se describe como controlar los numerosos elementos producidos por las muchas personas que trabajan en un proyecto común. El control ayuda a evitar confusión costosa, y garantiza que los elementos resultantes no están en conflicto debido a alguno de los siguientes problemas:

- Actualización simultánea – Cuando dos o más desarrolladores trabajan separadamente sobre el mismo elemento, el último que hace un cambio destruye el trabajo del anterior.
- Notificación limitada – Cuando se resuelve un problema en elementos compartidos por varios desarrolladores, y algunos de ellos no son notificados del cambio.
- Múltiples versiones – Los programas más grandes son desarrollados en versiones evolutivas. Una versión podría estar en uso del cliente, mientras que otro está en testeo, y el tercero aún está en desarrollo. Si se encuentran problemas en cualquiera de las versiones, las correcciones deben ser propagadas a todas. Pueden surgir confusiones que lleven a costosas reparaciones y retrabajos si los cambios no son cuidadosamente monitoreados y controlados.

Este flujo de tareas describe cómo se puede administrar desarrollo paralelo, desarrollo efectuado en múltiples sitios, y cómo automatizar el proceso de conformación de "builds". Esto es especialmente importante en un proceso iterativo, en el cual se puede querer ser capaz de generar "builds" diariamente, a veces sería imposible sin una poderosa automatización. También describe cómo se puede mantener un registro



de auditoría sobre por qué, cuando y por quién cada elemento fue cambiado.

Este flujo de tareas también cubre los cambios en los requisitos de administración, por ejemplo, como reportar los defectos, administrarlos a través de su ciclo de vida, y cómo usar los datos defectuosos para rastrear progresos y tendencias.

### **Ambiente**

El propósito del flujo de tareas de ambiente es proveer a la organización de desarrollo de software el ambiente – procesos y herramientas – que son necesarios para el soporte al equipo de desarrollo.

Este flujo de tareas enfoca las actividades para configurar el proceso en el contexto de un proyecto. También enfoca las actividades para desarrollar los lineamientos necesarios para soportar un proyecto. Suministra un procedimiento paso a paso, que describe como se implementa un proceso en una organización.

El flujo de tareas de Ambiente también contiene un Kit de desarrollo que da los lineamientos, templates y herramientas necesarios para adecuar el proceso al cliente. El Kit de desarrollo es descrito con mucho detalle en la sección "Kit de Desarrollo para el Proceso de adecuación al cliente", más adelante en este documento.



## 2. RUP – EL PRODUCTO

El producto RUP consiste en:

- Una base de conocimiento “web-enabled searchable”, que provee a todos los miembros del equipo lineamientos, templates y guías sobre el uso de herramientas para todas las actividades críticas de desarrollo. La base de conocimientos puede ser dividida en:
  - Lineamientos extensivos para todos los miembros del equipo, y todas las porciones del ciclo de vida del software. La guía está prevista para ambas situaciones: tanto para el proceso de pensamiento de alto nivel como para las más tediosas actividades cotidianas. La guía está publicada en formato HTML para un acceso fácil independiente de la plataforma desde cada escritorio.
  - Guías de uso que proveen consejos sobre herramientas que cubren todo el ciclo de vida. Las guías para el uso de herramientas están publicadas en formato HTML para un acceso fácil independiente de la plataforma desde cada escritorio.
  - Ejemplos y templates Rational Rose, que guían sobre cómo estructurar la información en Rational Rose cuando se sigue el RUP (Rational Rose es la herramienta de Rational para modelización visual)
  - Templates SoDA – más de 10 templates SoDA que ayudan a automatizar la documentación del software (SoDA es la herramienta Rational para automatización de la documentación)
  - Templates Microsoft Word – más de 30 templates que asisten para la documentación en todos los flujos de tareas y en todas las partes del ciclo de vida.



- Planificadores de Microsoft Project – Muchos administradores encuentran difícil crear planes en Microsoft Project que reflejen un enfoque de desarrollo iterativo. Nuestros templates ejemplifican la creación de planes de proyecto para desarrollos iterativos, de acuerdo con el RUP:
- Kit de desarrollo – describe como adecuar al cliente y extender el RUP a las necesidades específicas del proyecto, a la vez que provee herramientas y templates para ayudar en este esfuerzo. Este Kit de desarrollo es descrito más adelante, en esta sección.
- Acceso al centro de recursos que contiene la más reciente documentación, actualizaciones, sugerencias y técnicas, tanto como referencias a productos y servicios adicionales.

### **Navegando en la base de conocimientos**

Conocer el RUP permite acceder al contenido con cualquiera de los populares navegadores web, tales como Microsoft Internet Explorer y Netscape Navigator.

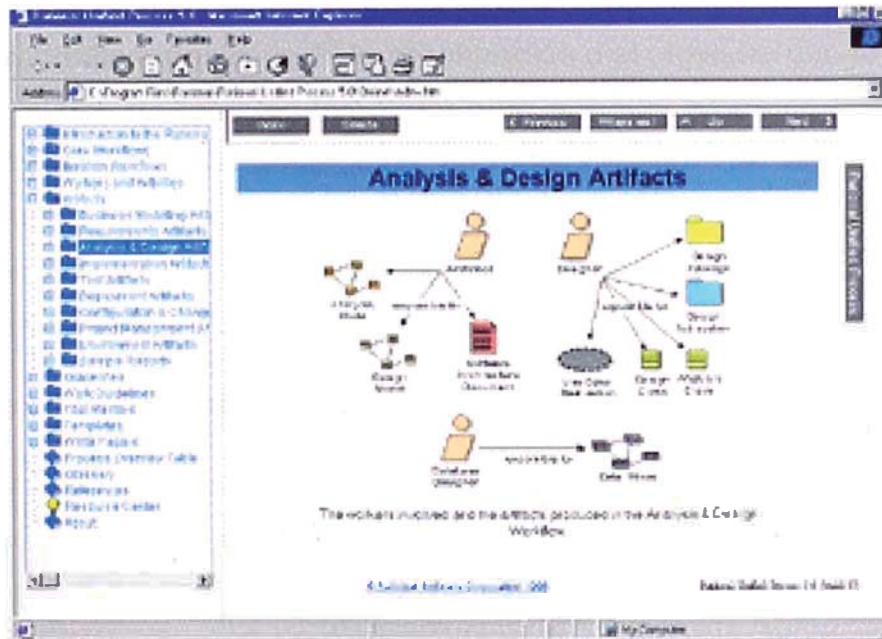
La base de conocimientos contiene vínculos de hipertexto y se presentan panoramas de elementos de proceso a través de imágenes interactivas, facilitando el hallazgo de información relevante de manera intuitiva. La poderosa máquina de búsqueda, el índice, y el árbol de búsqueda, facilitan el uso del proceso.

La información está presentada de varias maneras diferentes, permitiéndole mirar la información relevante para su rol, para una actividad específica o para un flujo de tareas. Se proveen visitas guiadas para el aprendizaje fácil del proceso para los roles claves del proyecto.

#### Vista del diagrama ampliado



Imágenes interactivas y botones de navegación facilitan el hallazgo de



las informaciones específicas que se están buscando.

### Kit de desarrollo para el proceso de adecuación al cliente

El RUP es suficientemente general y completo para ser usado "tal cual" por algunas organizaciones de desarrollo de software. Sin embargo, en muchas circunstancias, este proceso de ingeniería de software necesitará ser modificado, ajustado y recortado para adecuarse a las características específicas, exigencias e historia de la organización que lo adopta. En particular, un proceso no debe ser seguido ciegamente, generando así trabajo inútil y produciendo elementos que tiene poco valor agregado. Debe ser hecho tan pequeño como sea posible y seguir siendo capaz de



cumplir su misión para producir rápida y predeciblemente software de alta calidad.

El proceso contiene un Kit de desarrollo, que contiene lineamientos para que se pueda adecuar el proceso al cliente, de modo que satisfaga las necesidades específicas de la organización o el proyecto que lo adopta. Se han incluido también templates para adecuar procesos, tanto como herramientas para generación o manipulación de dispositivos de búsqueda, índices, mapas, visualizadores de árboles, etc. El Kit de desarrollo permite a la organización que lo adecua mantener el aspecto y el espíritu del RUP.

### **Integración con herramientas**

Un proceso de ingeniería de software requiere herramientas para soportar todas las actividades en el ciclo de vida del sistema, especialmente para soportar el desarrollo, mantenimiento y registración de variados elementos del modelo en particular. Un proceso de desarrollo iterativo plantea requerimientos especiales sobre el conjunto de herramientas que se usarán, tales como una mejor integración entre herramientas e ingeniería de ida y vuelta entre modelos y código. También se necesitan herramientas que guarden huellas de los cambios, para soportar el requerimiento de rastreabilidad, para automatizar la documentación, tanto como herramientas para automatizar el testeado que facilite el testeado regresivo. El RUP puede ser usado con una variedad de herramientas, ya sea de Rational o de otros proveedores. Sin embargo, Rational provee muchas herramientas integradas que soportan eficientemente el RUP.

Más adelante se encontrará una lista de algunas de las herramientas de Rational que soportan el RUP.





El RUP contiene guías para el uso de herramientas para casi todos estos productos. Una *guía para el uso de herramienta* es una guía paso a paso que describe en detalle cómo operar una herramienta (por ejemplo, qué menús disparar, que información ingresar en las ventanas de diálogo, y como navegar una herramienta) para realizar una actividad dentro del proceso. La guía para el uso de herramientas permite vincular el proceso, independiente de herramientas, con la real manipulación de las herramientas en el trabajo diario.

### 3. Herramientas Integradas que soportan eficientemente el RUP:

- ✓ **Rational Requisite Pro** – Mantiene a todo el equipo de desarrollo actualizado a través del proceso de desarrollo de aplicaciones haciendo que los requerimientos se puedan escribir, comunicar y cambiar fácilmente.
- ✓ **Rational Clear Quest** - Un producto Windows y basado en Web de administración de solicitudes de cambio que permite a los equipos de proyecto rastrear y administrar todas las actividades de cambio que ocurren durante el desarrollo del ciclo de vida.
- ✓ **Rational Rose** La herramienta líder en el mundo de modelización visual para el proceso de modelización del negocio, análisis de requerimientos y diseño de arquitectura de componentes.
- ✓ **Rational SODA** Automatiza la producción de documentación para todo el proceso de desarrollo de software, reduciendo dramáticamente el tiempo y el costo de documentar el software.
- ✓ **Rational Purify** – Una herramienta de control de errores en tiempo de corrida para software de aplicación y elementos para



desarrolladores que programan en C/C++, ayuda a detectar errores de memoria.

✓ **Rational Visual Quantify** – Una herramienta avanzada de monitoreo de desempeño (performance) para desarrolladores de aplicaciones y elementos programados en C++, Visual Basic y Java que ayuda a eliminar cuellos de botella.

✓ **Rational Visual PureCoverage** – Señala automáticamente áreas de código no ejecutadas en testeo, de modo que los desarrolladores puedan testear a fondo, eficiente y efectivamente sus aplicaciones.

✓ **Rational Team Test** – Crea, mantiene y ejecuta tests funcionales automatizados, permitiendo testear a fondo el código y determinar si el software cubre los requerimientos y funciona como se espera.

✓ **Rational Performance Studio** – Una herramienta fácil de usar, precisa y cuantificable, que mide y predice la performance de sistemas cliente/servidor y Web.

✓ **Rational Clear Case** – Herramienta de administración de configuración de software, líder en el mercado, que da a los administradores de proyecto la posibilidad de rastrear la evolución de cada proyecto de desarrollo de software.



#### IV. EVALUACIÓN DE RESULTADOS

Atendiendo los requerimientos de información relacionados a los servicios de desarrollo y mantenimiento de software se debe obligatoriamente emplear la Metodología de desarrollo de software orientado a objetos que servirá para respaldar el estudio de la metodología con un caso práctico realizado utilizando la metodología Rational.

A continuación presentamos los requerimientos del Caso práctico con lo que se debera realizar lo siguiente:

- Elaborar el Modelo Conceptual del Sistema.
- Identificar los diferentes escenarios para los casos de uso identificado.
- Para los escenarios identificados elaborar Diagramas de Secuencia y Diagramas de Colaboración.
- Organizar la Información en Subsistemas diseñando los Diagramas de Clase,.
- Establecer un diagrama de estado para un objeto en particular.

Esta información estará incluida en los anexos.



**[TNT VIDEOS]**

Especificación de Requerimientos de  
Software

© 2002



## ESPECIFICACIÓN DE REQUERIMIENTOS

Se desea realizar un sistema informático orientado a las actividades de alquiler y venta de videos de una tienda de cintas de videos . Este sistema se detalla en los requerimientos funcionales y no funcionales que se describen a continuación.

### 1.1 REQUERIMIENTOS FUNCIONALES

Se detallan los requerimientos funcionales que deben ser cubiertos por el sistema propuesto.

#### ALQUILER DE CINTAS DE VIDEO

- ❑ El alquiler de videos se realiza a través de PC
- ❑ El alquiler de videos se realiza sólo para clientes BlockBuster.
- ❑ Se solicita al cliente su tarjeta BlockBuster, de no portarla se le solicita el DNI.
- ❑ En la caja se realiza la lectura del código de barra de la tarjeta Blockbuster, del cual se obtienen los datos personales del cliente, y la lista de videos alquilados.
- ❑ En la caja se realiza la lectura del código de barra de cada cinta, del cual se obtiene su título y precio del alquiler.
- ❑ Se puede alquilar hasta 5 cintas de video para los clientes normales y tantas películas quieran alquilar los clientes VIP.



- ❑ Toda vez que una cinta de video es registrado en la transacción, se debe determinar el precio total de la cinta, lo cual dependerá de la cantidad solicitada y el precio unitario.
- ❑ En todo momento se debe determinar el monto total del alquiler.
- ❑ Todo alquiler se efectúa al contado..
- ❑ El alquiler se paga en efectivo
- ❑ Una vez procesada la transacción se actualizará el inventario de cintas de video, y la cantidad de videos alquilados por el cliente.
- ❑ El alquiler de una o varias cintas se realizan hasta por un máximo de dos días.
- ❑ En cada transacción, deberá registrarse el identificador del cajero que realiza la transacción de alquiler.

### **DEVOLUCIÓN DE CINTAS DE VIDEOS**

- ❑ La devolución se realiza a través de pc.
- ❑ En la caja se realiza la lectura del código de barra de cada cinta, del cual se obtiene su título, el precio de alquiler y el código del cliente al que se le ha rentado la cinta.
- ❑ A partir del día en curso, se determina si el cliente debe pagar una mora por cada cinta alquilada.
- ❑ Toda vez que una cinta es registrada en la transacción, se debe determinar el monto total de la mora.
- ❑ La mora se paga en efectivo.
- ❑ Una vez procesada la transacción, se actualizará el inventario de cintas de video y los datos de transacción del cliente (Se elimina la lista de videos alquilados del cliente y el día del alquiler).
- ❑ Se deberá emitir un comprobante de devolución al cliente.
- ❑ En cada transacción, deberá registrarse el identificador del cajero que realiza la transacción de devolución.



### **VENTA DE CINTAS DE VIDEO.**

- ❑ La venta de cintas de video se realiza a través de la PC.
- ❑ En la caja se realiza la apertura del código de barra de cada video del cual se obtiene su título y precio de venta.
- ❑ Se pueden vender tantas cintas de videos como lo desee el cliente.
- ❑ Toda vez que una cinta de video es registrado en la transacción, se debe determinar el precio total de la cinta, lo cual dependerá de la cantidad solicitada y el precio unitario.
- ❑ En todo momento se debe determinar el monto total de la venta .
- ❑ Toda venta se efectúa al contado.
- ❑ La venta se paga en efectivo.
- ❑ Una vez procesada la transacción, se actualizará el inventario de cintas de video.
- ❑ Se deberá emitir un comprobante de venta al cliente.
- ❑ En cada transacción, deberá registrarse el identificador del cajero que realiza la transacción de alquiler.

### **COMPRAS DE CINTAS DE VIDEO**

- ❑ Las cintas de video a ser compradas se determinarán a partir de la existencia del mismo, del nivel de existencia del mismo y de los estrenos de las películas que se presenten.
- ❑ Las cintas de video pueden ser compradas a diferentes proveedores, para ello se debe proporcionar información para la toma de decisión, en función de los precios de oferta de los proveedores por cada película.



- ❑ Se mantiene una cartera de proveedores de los que se conoce RUC, nombre, dirección, teléfono, fax, dirección electrónica. Y para cada proveedor se conoce cuales son las películas que suministran y el precio de oferta por cada cinta de video de las películas.
- ❑ La compra se realiza en tres etapas:

**Preparación de la Orden de compra.** Aquí se indica para cada orden de compra a que proveedor se le va a comprar, el título de la película, la fecha, el número de la orden de compra y la cantidad de cintas de video de dicho título a comprar.

**Recepción de los videos.** Aquí se procede a recibir las cintas de video de un título pedido. Los que vienen acompañados de una guía de remisión. Se realiza el chequeo físico de las cintas de video y se registra en la Orden de compra de los precios de cada cinta de video solicitada y se determina el valor total de la venta. El inventario de la mercadería debe quedar totalmente actualizado, reflejándose en el movimiento la guía de remisión. El pedido puede recibirse parcial o totalmente.

- ❑ Una orden de compra se define como recibida cuando todas las cintas de video requeridas de un título han sido recibidos.
- ❑ Anulación de la Orden de compra. Las órdenes de compra s pueden ser anuladas en cualquier momento siempre y cuando la misma no se haya recibido.
- ❑ La Compra Puede Realizarse Al Crédito o al Contado.

## **INVENTARIO DE PRODUCTOS**

- ❑ Para cada video se debe registrar: código, título, actores, género, categoría, existencia mínima permisible, existencia máxima





permisible, nivel del próximo pedido y un registro de cada cinta de las que se conoce su número de referencia.

- ❑ En el registro constan todos los movimientos de la película, todos los movimientos de ingresos (compras e ingresos por devolución ) y egresos (ventas y salidas por préstamo) así como los ajustes de existencia.
- ❑ Los precios de alquiler son los siguientes:

Estreno 12 soles

Niño: 9 soles

Favorita : 4 soles.

- ❑ Una cinta puede quedar suspendido para su alquiler debido a su falta de rotación o a su deterioro.
- ❑ La tienda ofrece videos con los siguientes géneros: drama, terror, ciencia ficción y acción.
- ❑ La distribución de los videos en la tienda se da por estrenos, favoritas y videos para niños.
- ❑ Las transacciones de compra venta y alquiler de videos deben actualizar el inventario.

### **REGISTRAR NUEVOS CLIENTES NORMALES**

- ❑ Para registrar nuevos clientes, solo les solicita su DNI y su recibo de agua, de luz o de teléfono.
- ❑ Una vez recibido estos datos se registra un nuevo cliente dentro del sistema.



## **REGISTRAR NUEVOS CLIENTES VIP**

- ❑ Un cliente normal puede convertirse en un cliente VIP.
- ❑ Si un cliente normal mantiene mas de 3 alquileres de cintas por semana pasa a convertirse en un cliente VIP.
- ❑ Así como se convierte en un cliente VIP, puede dejar de serlo si es que deja de alquilar mas de 3 veces por semana.

## **PAGOS DE DEUDAS A PROVEEDORES**

El pago de deudas a proveedores se realiza cancelando parte o el monto total de la deuda que se tiene con el proveedor.

## **FUENTES DE REQUERIMIENTOS FUNCIONALES**

Los requerimientos fueron obtenidos por las necesidades de los propios clientes y/o sugerencias de los usuarios y consultores comerciales, estos se capturaron por medio de encuestas, entrevistas y considerando los procedimientos de la empresa.

## **1.2 REQUERIMIENTOS DE INFORMES**

### **INFORME DE CLIENTES MOROSOS**

En este informe deberá contar con el código del cliente, la lista de cintas de video no devuelvan y el valor total de mora.

### **INFORME DE CASSETTES ALQUILADOS**

En este informe deberá contar con la lista de cintas de video alquiladas con sus respectivos códigos y títulos.



### **INFORME DE PELÍCULAS POR GENERO**

En este informe deberá contar con la lista de cintas de video alquiladas con sus respectivos códigos y títulos y género seleccionado.

### **INFORME DE PELÍCULAS POR DIRECTOR**

En este informe deberá contar con la lista de cintas de video alquiladas con sus respectivos códigos y títulos y director seleccionado.

### **INFORME DE ALQUILER DE CINTAS DE VIDEO POR CAJEROS EN EL MES**

El informe deberá presentar en forma de un listado por cajero la siguiente información: mes, y por cada cajero su código, nombre, y monto total de los alquileres por mes.

### **INFORME DE VENTAS DE VIDEOS POR CAJEROS EN EL MES**

El informe deberá presentar en forma de un listado por cajero la siguiente información: mes, y por cada cajero su código, nombre, y monto total de las ventas por mes.

## **1.3 REQUERIMIENTOS NO FUNCIONALES**

### **1.3.1 Requerimientos de seguridad y privacidad**

- El sistema deberá proporcionar mecanismos de seguridad y protección de la información.
- Un estricto control de los usuarios a las compras y ventas.



- El acceso a las PC en los puntos de la venta se deberá realizar previo ingreso de la identificación correspondiente del cajero, el cual contará con una clave de acceso.
- Los cajeros solo podrán efectuar las ventas, alquileres y devoluciones en los puntos.
- No se recomienda que el sistema de software que se instale en los puntos de ventas, alquileres y devoluciones, contenga acceso al resto de las funcionalidades del sistema. Si es posible que allí sólo se ubiquen las funciones propias para las ventas, alquileres y devoluciones.

### **1.3.2 Requerimientos de recursos de computo**

#### **1.3.2.1 Requerimientos de Hardware**

- Se debe operar sobre una red de área local (LAN) en la que se ubica un servidor donde residen todos los datos de inventario, compras, alquileres, devoluciones y ventas de cintas de video.
- Las estaciones de trabajo son PC que operan bajo el entorno Windows 98.
- La información sobre inventario, compras, alquileres, devoluciones y ventas de cintas de video se comparten a todas las PC.
- El registro de las ventas, alquileres y devoluciones se realizan por medio de los lectores de códigos de barra que se encuentran conectadas a las PC y estas a su vez al servidor.
- Se dispone de un segundo servidor para manejo y ejecución de las transacciones, el que a su vez se conecta por la LAN con el servidor de datos principales.

#### **1.3.2.2 Requerimientos de Software**



**Programas que deben de ser instalados:**

Visual basic 6.0

Sql Server 7.0

Requerimientos de Comunicaciones

Protocolo TCP/IP

Programa servidor, ejecutable

Event Log

Admin Service

License Logging Service



## CONCLUSIONES Y RECOMENDACIONES

- ◇ RUP es un proceso de ingeniería de software. Su principal virtud es asegurar la producción de software de alta calidad, apropiado a las necesidades del usuario final, dentro de un cronograma y un presupuesto predecibles.
- ◇ RUP potencia la productividad del equipo, al proveer a cada uno de sus miembros un fácil acceso a una base de conocimiento con lineamientos, templates y guías sobre qué herramientas usar para todas las actividades críticas del desarrollo.
- ◇ El RUP es una guía sobre como usar efectivamente el UML. El UML es un lenguaje estándar aceptado por todos las empresas importantes en el rubro de la tecnología, que permite comunicar claramente requerimientos, arquitectura y diseños.
- ◇ El RUP captura muchas de las mejores prácticas del desarrollo de software moderno, de tal modo que es adecuado a un amplio rango de proyectos y organizaciones. Desplegar estas mejores prácticas (usando el RUP como guía) ofrece a los equipos de desarrollo una cantidad de ventajas clave. A continuación las mejores prácticas del RUP:
  - ✓ Desarrollar software iterativamente
  - ✓ Administrar requerimientos
  - ✓ Usar arquitecturas basadas en componentes
  - ✓ Modelizar software visualmente
  - ✓ Verificar la calidad de software
  - ✓ Controlar los cambios al software.



## BIBLIOGRAFÍA

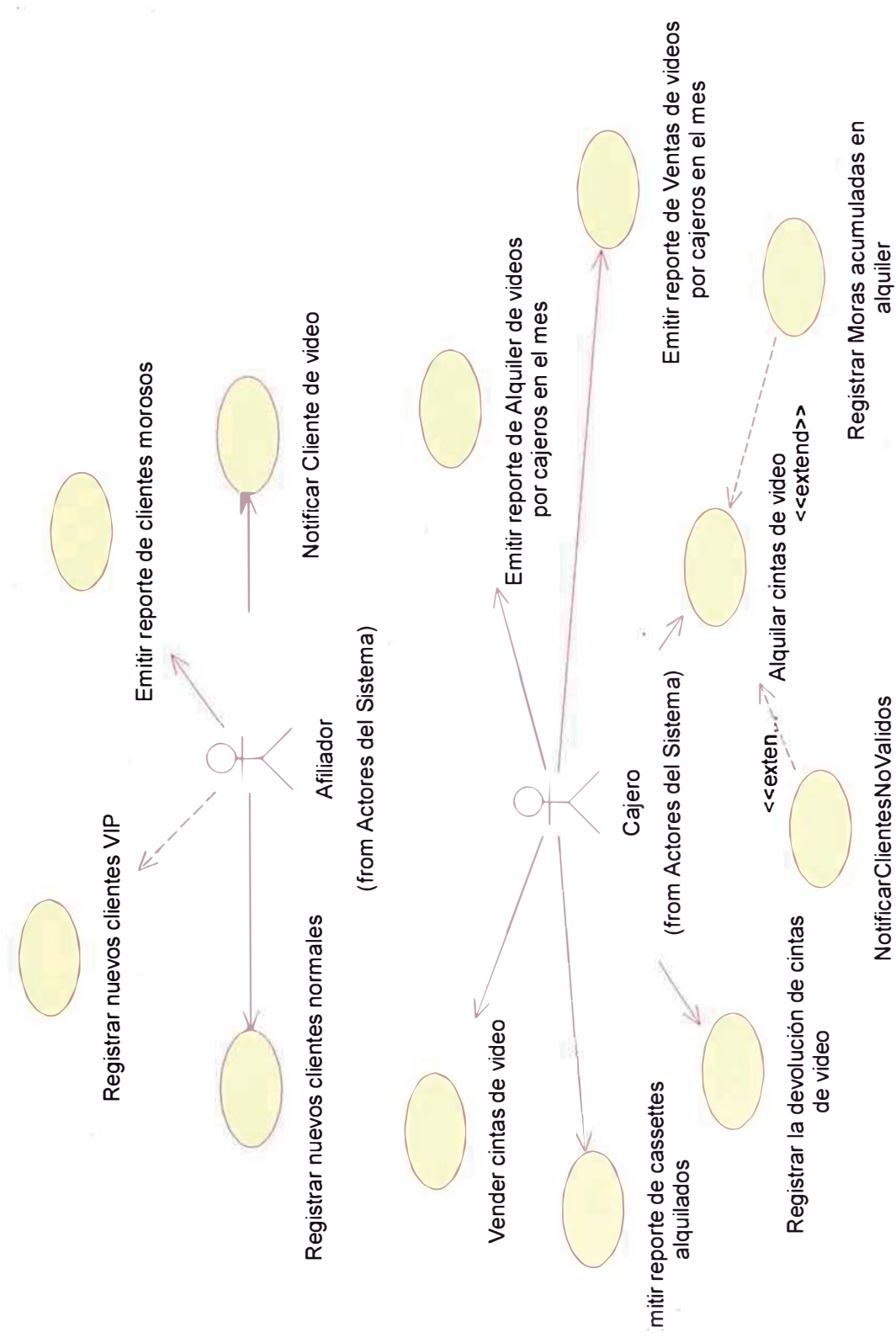
- [www.rational.com](http://www.rational.com)
- Martin Fowler, "UML Distilled" ("UML Gota a Gota")
- Terry Quatrani, "Visual Modeling ...", un caso de estudio
- Terry Quatrani, "Visual Modeling with Rational Rose and UML"
- Philippe Kruchten, "The Rational Unified Process -Introduction"
- James Rumbaugh, Ivar Jacobson, Grady Booch, "El lenguaje Unificado del Modelado"
- Ivar Jacobson, Grady Booch, James Rumbaugh, "El Proceso Unificado de Desarrollo de Software"
- International Council in SE (INCOSE) Tools Database

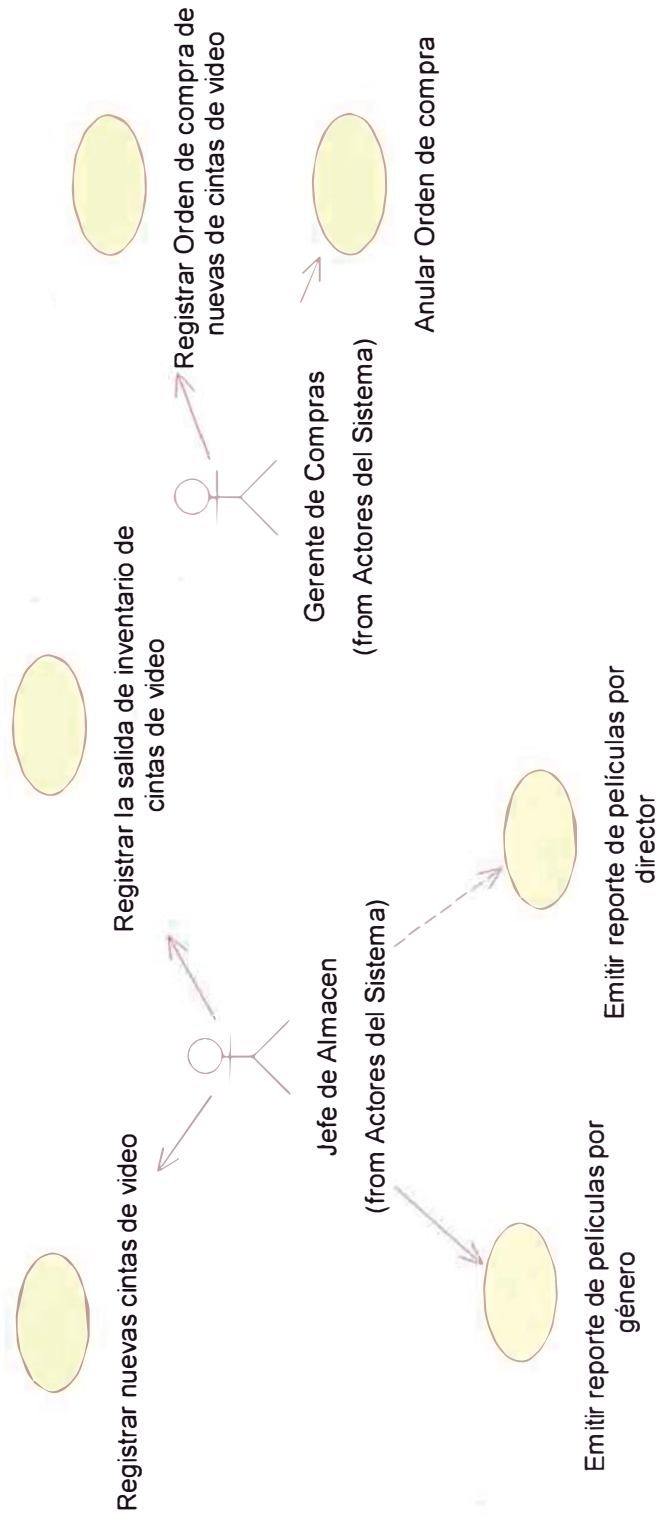


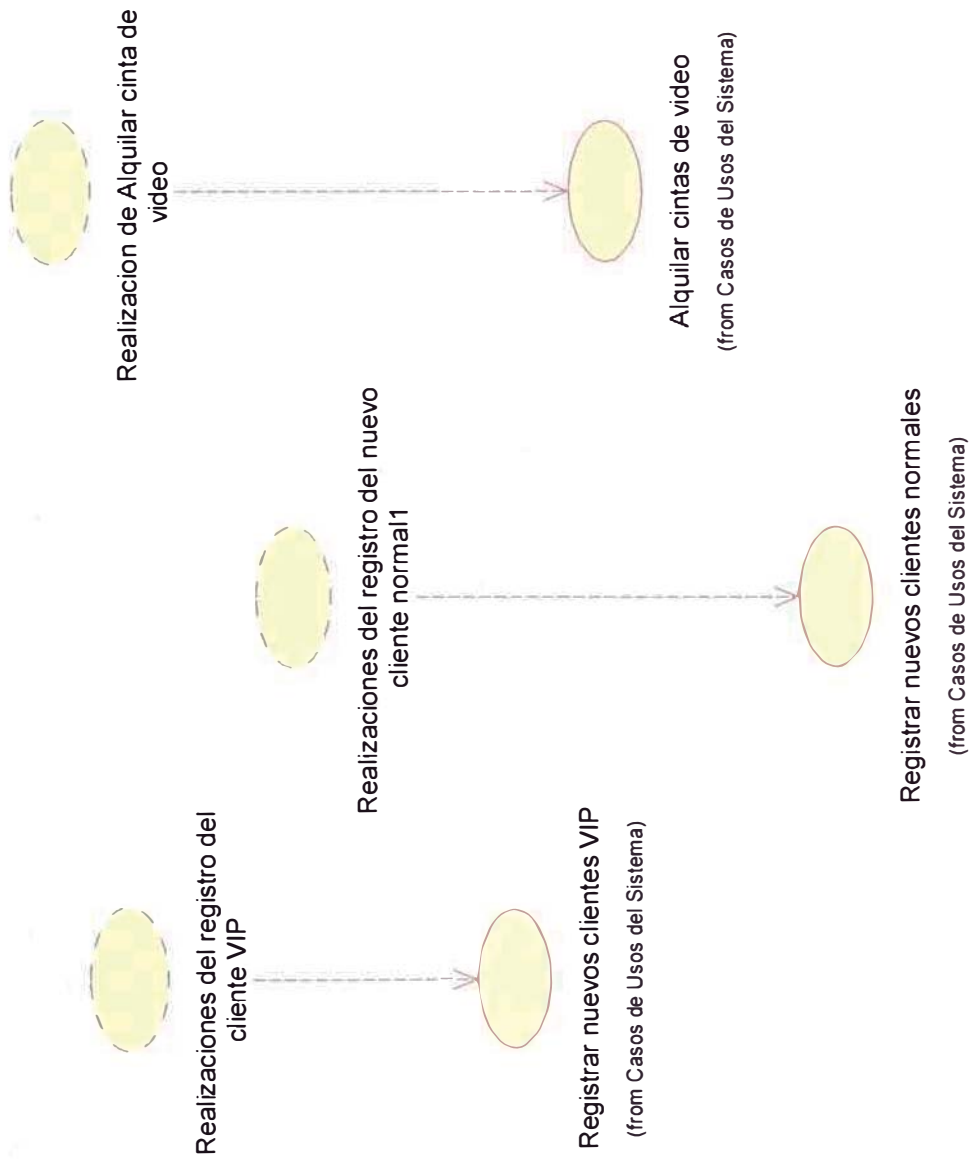
## ANEXOS

- Diagrama de Casos de Uso
- Diagrama de Clases – Análisis
- Diagrama de Clases – Diseño
- Diagrama de Modelo de Datos.
- Diagramas de Secuencia.
- Diagrama de Colaboración.
- Diagrama de y Estado.
- Diagrama de Componentes.
- Diagrama de Despliegue.

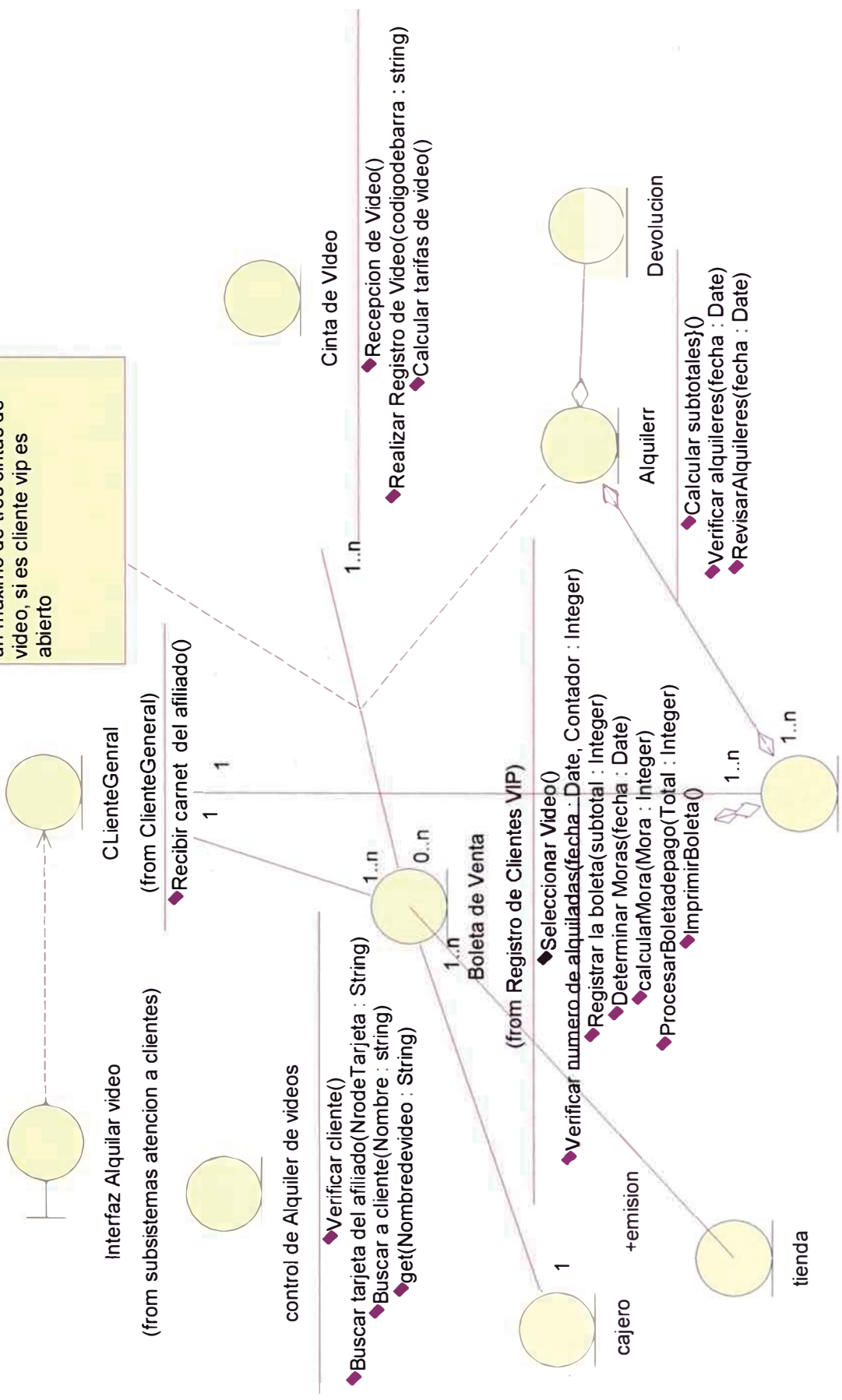




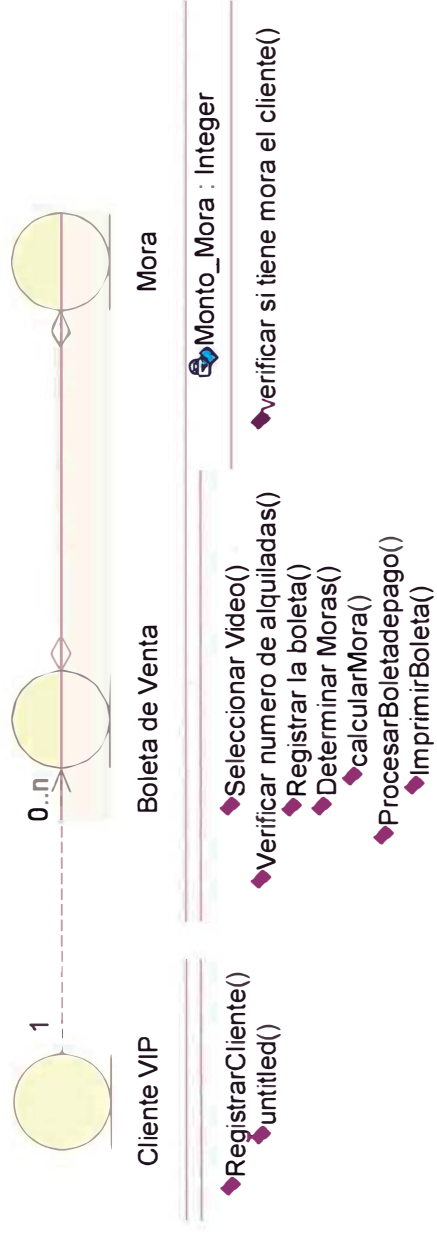


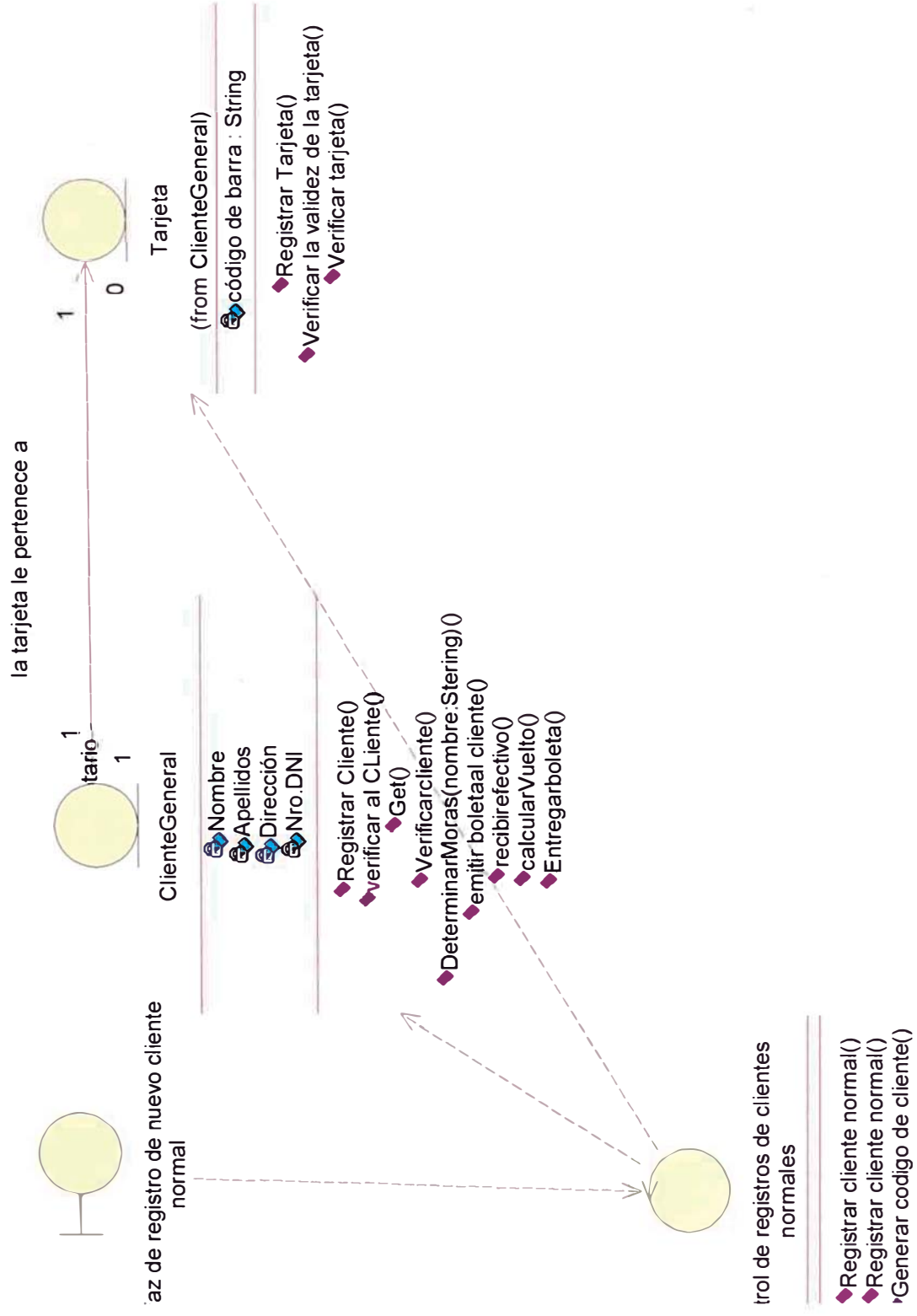


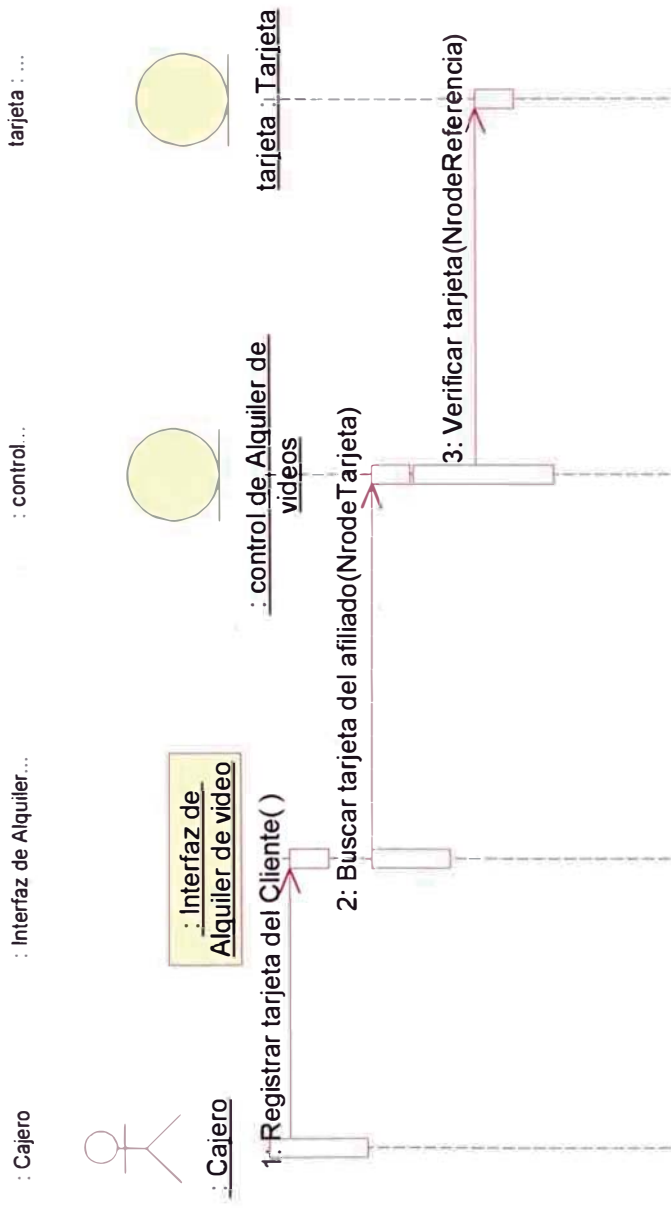
Si es cliente normal se exige un maximo de tres cintas de video, si es cliente vip es abierto

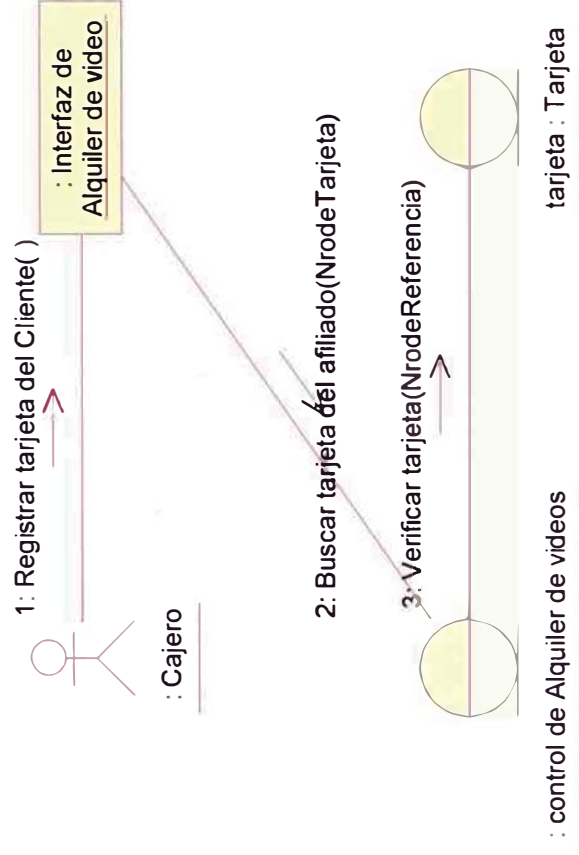


(from Registro de Clientes VIP)  
 ◆ verificar si tiene mora el cliente()

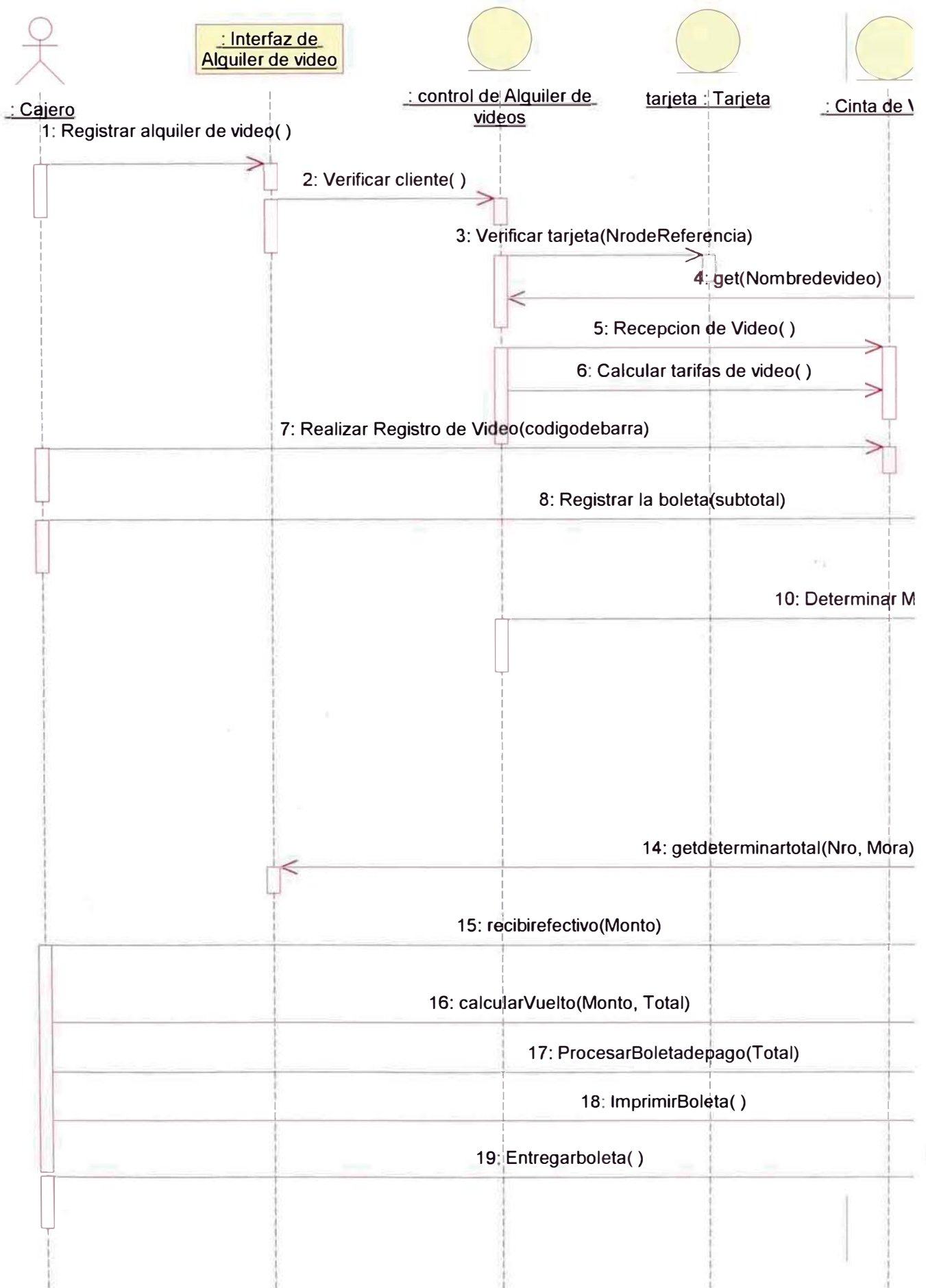












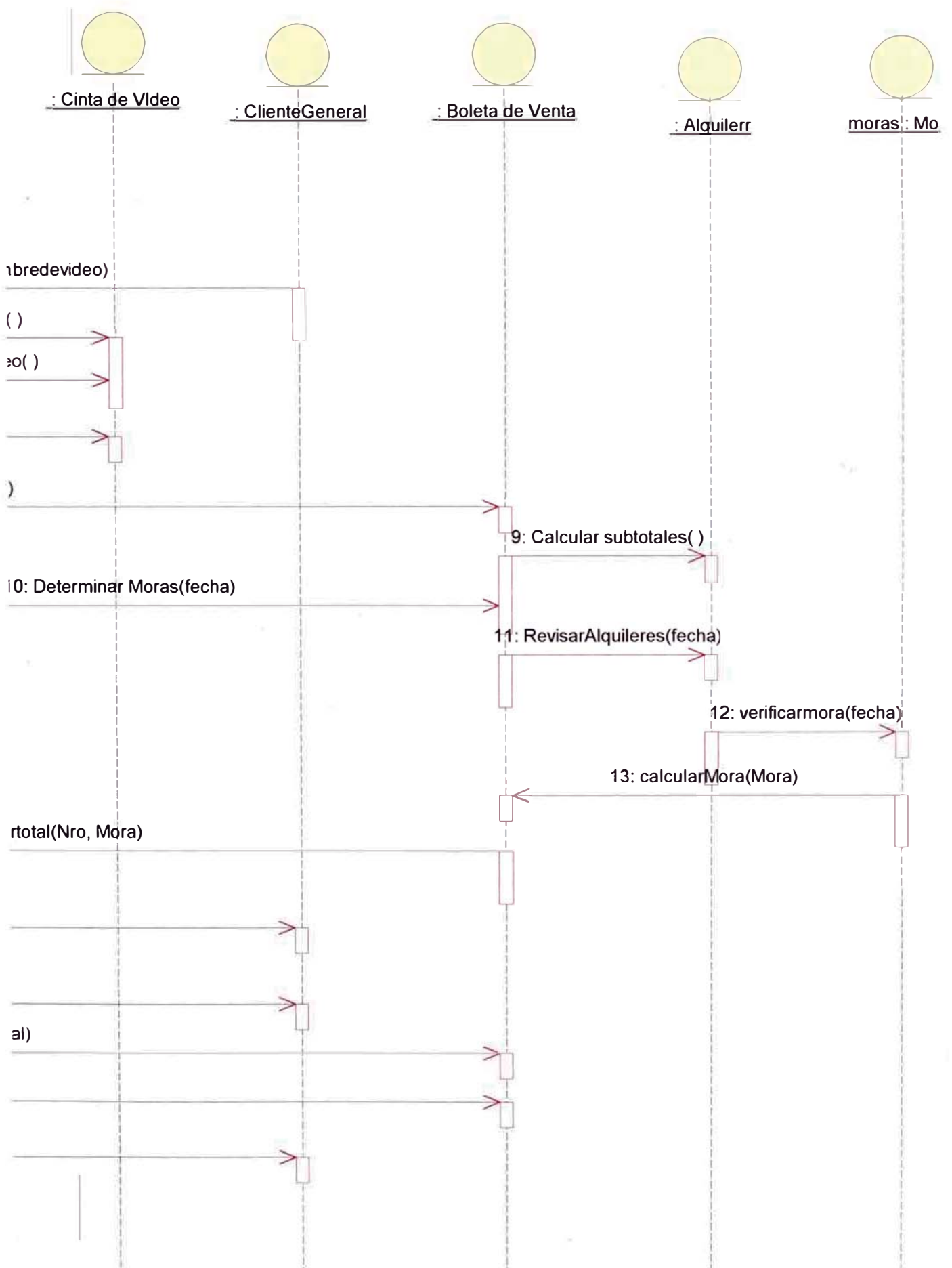
: Cinta ...

: Cliente...

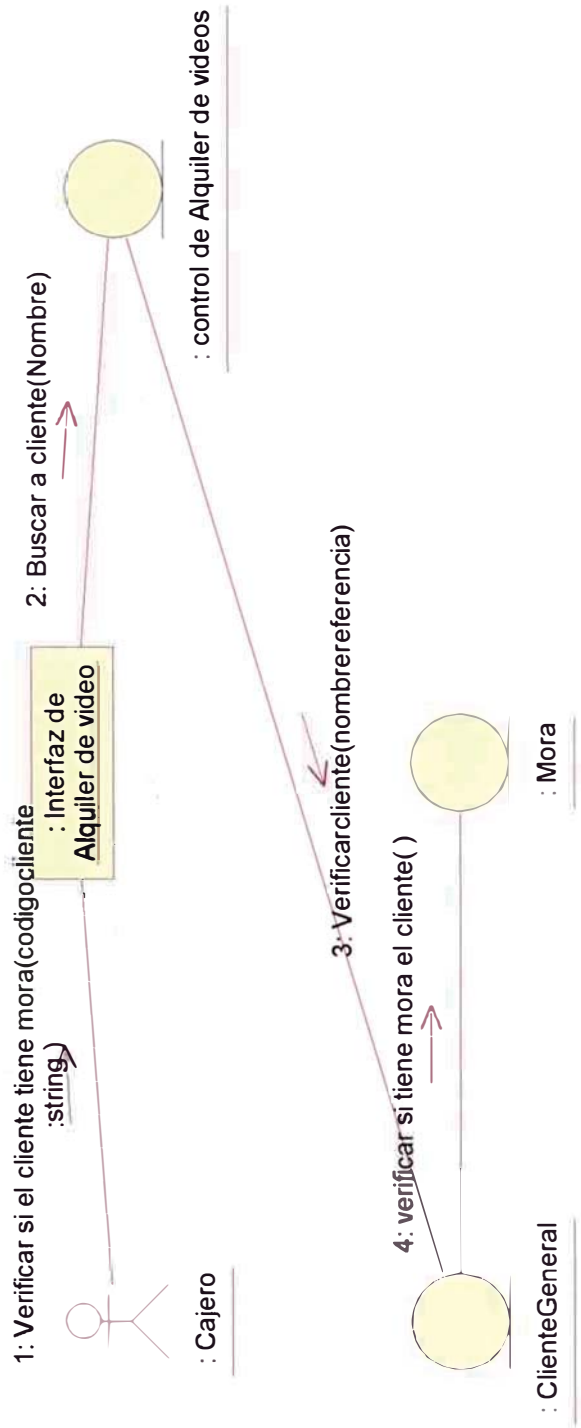
: Boleta ...

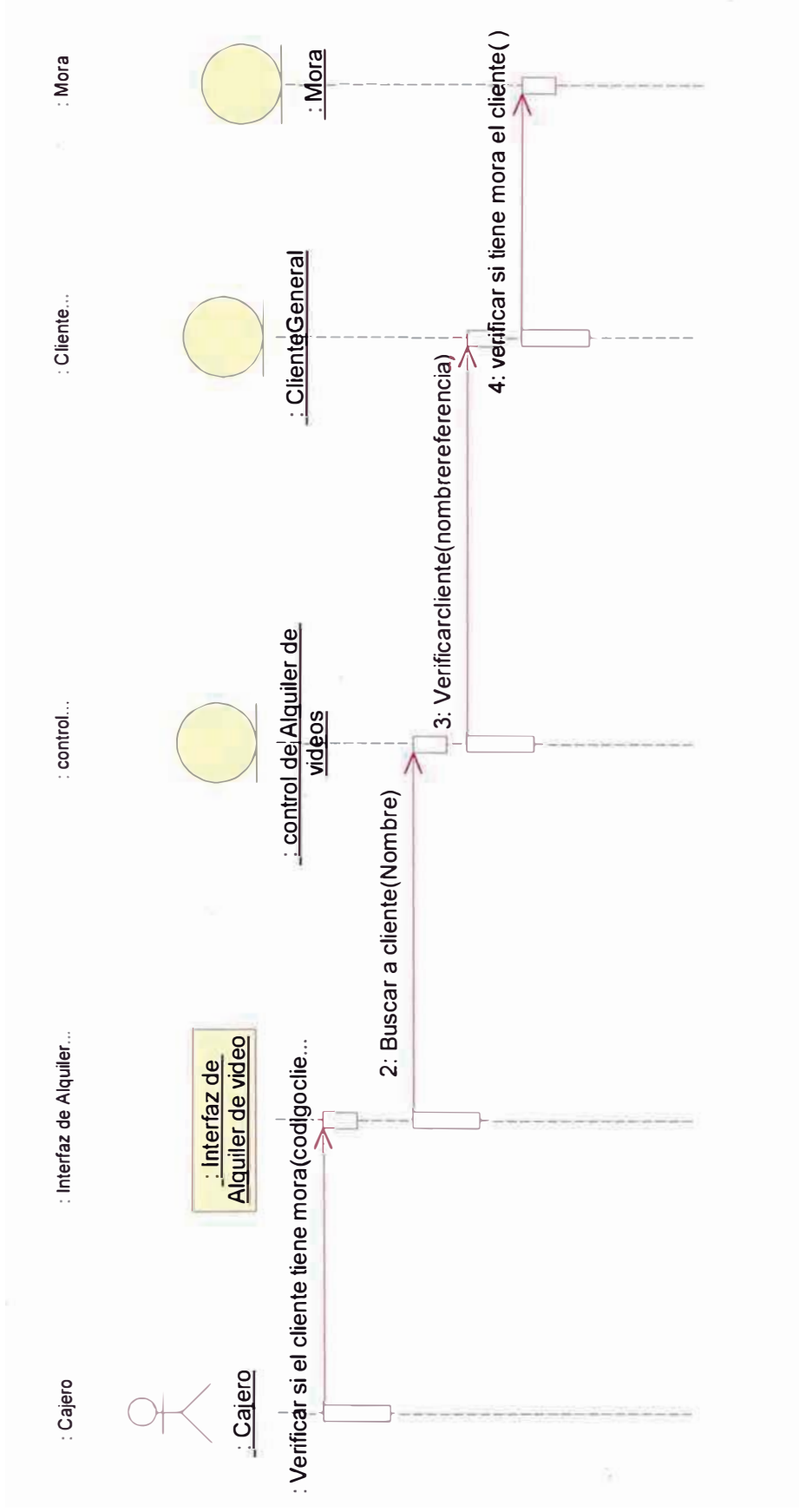
: Alquilerr

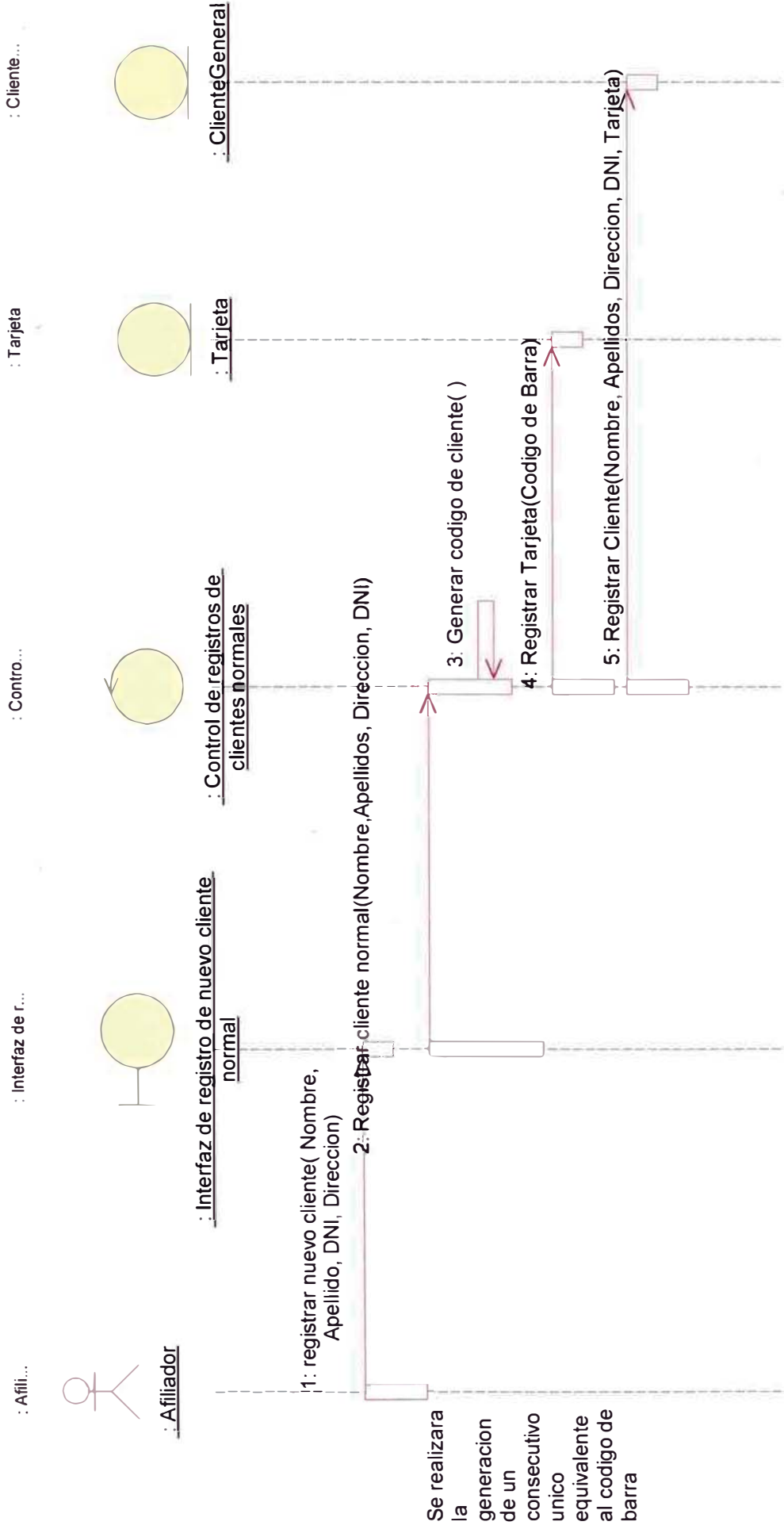
moras : ...





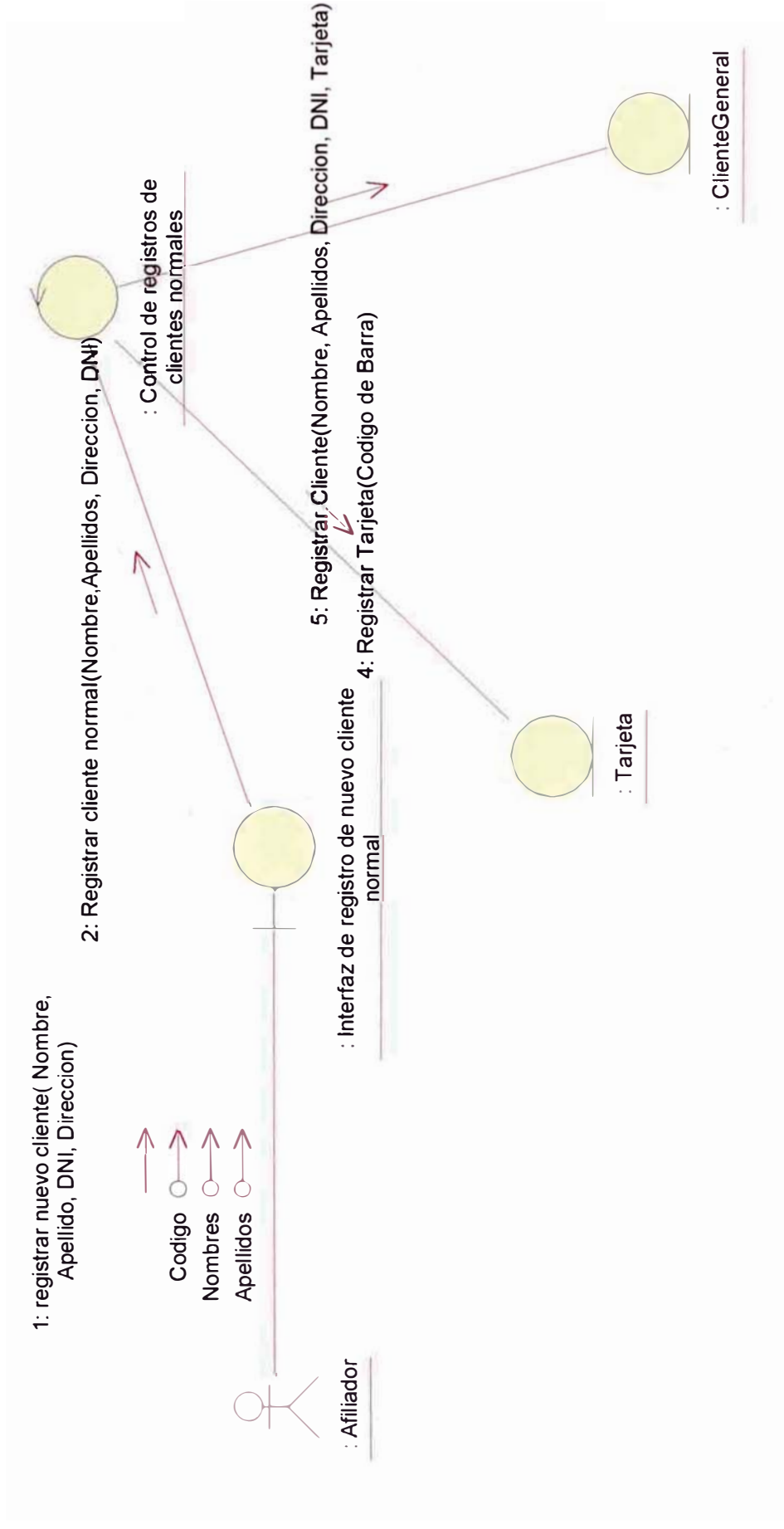


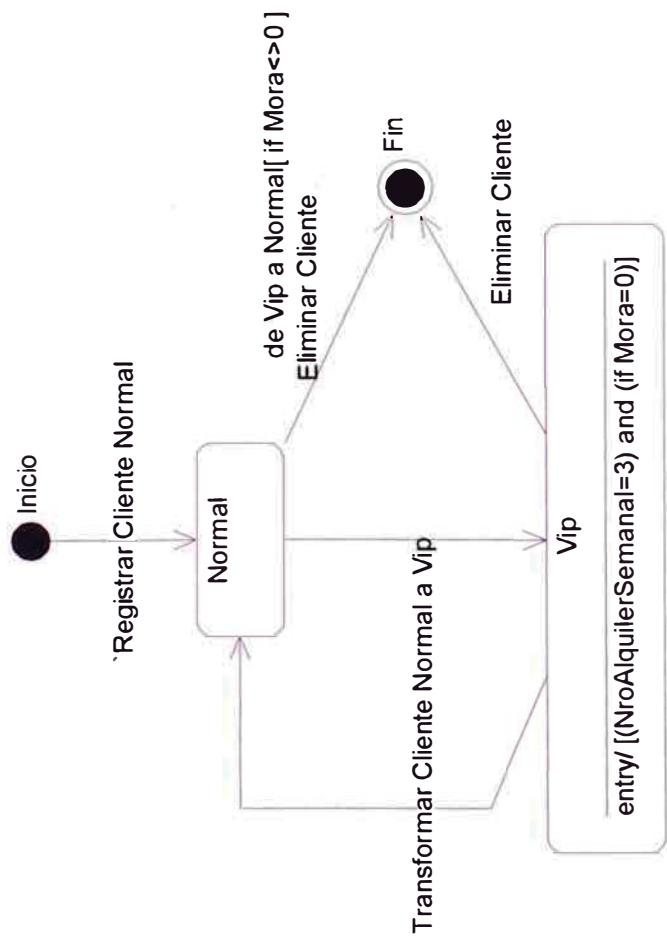




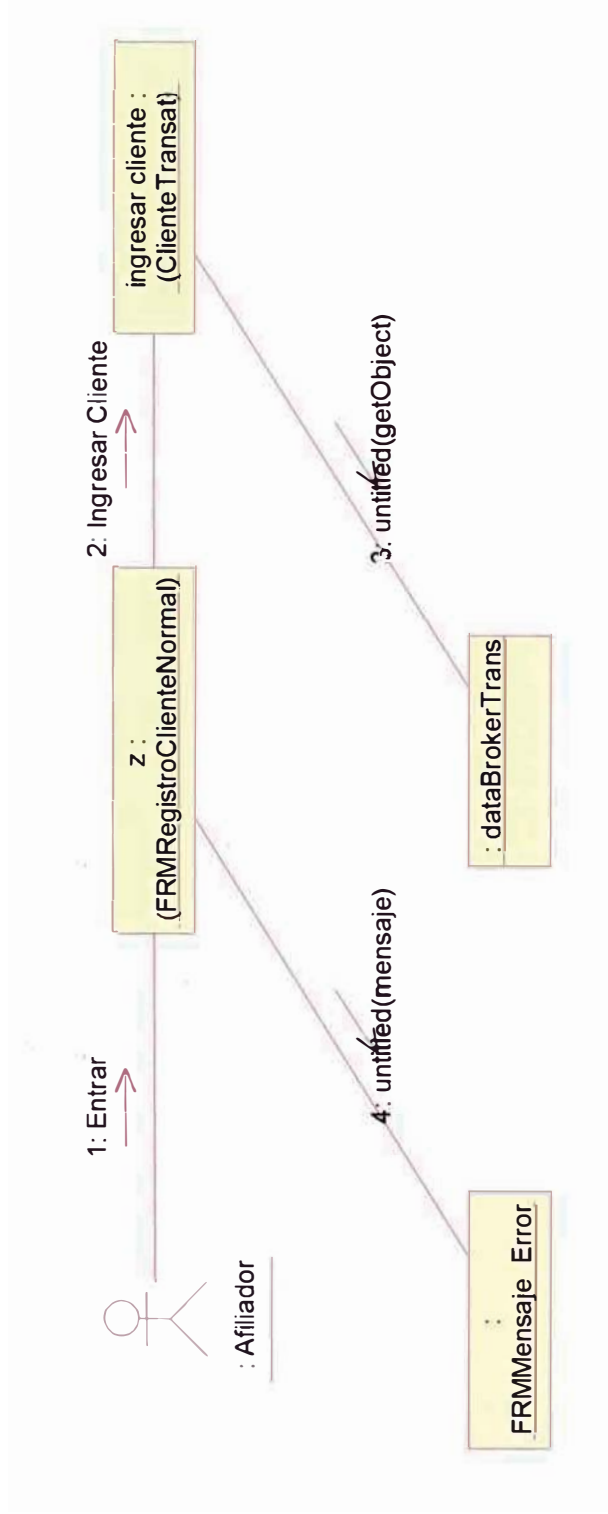
Se realizara la generacion de un consecutivo unico al codigo de barra

transformar clientevip, no es necesario crear pero no todos pueden cambiar es nesario q...

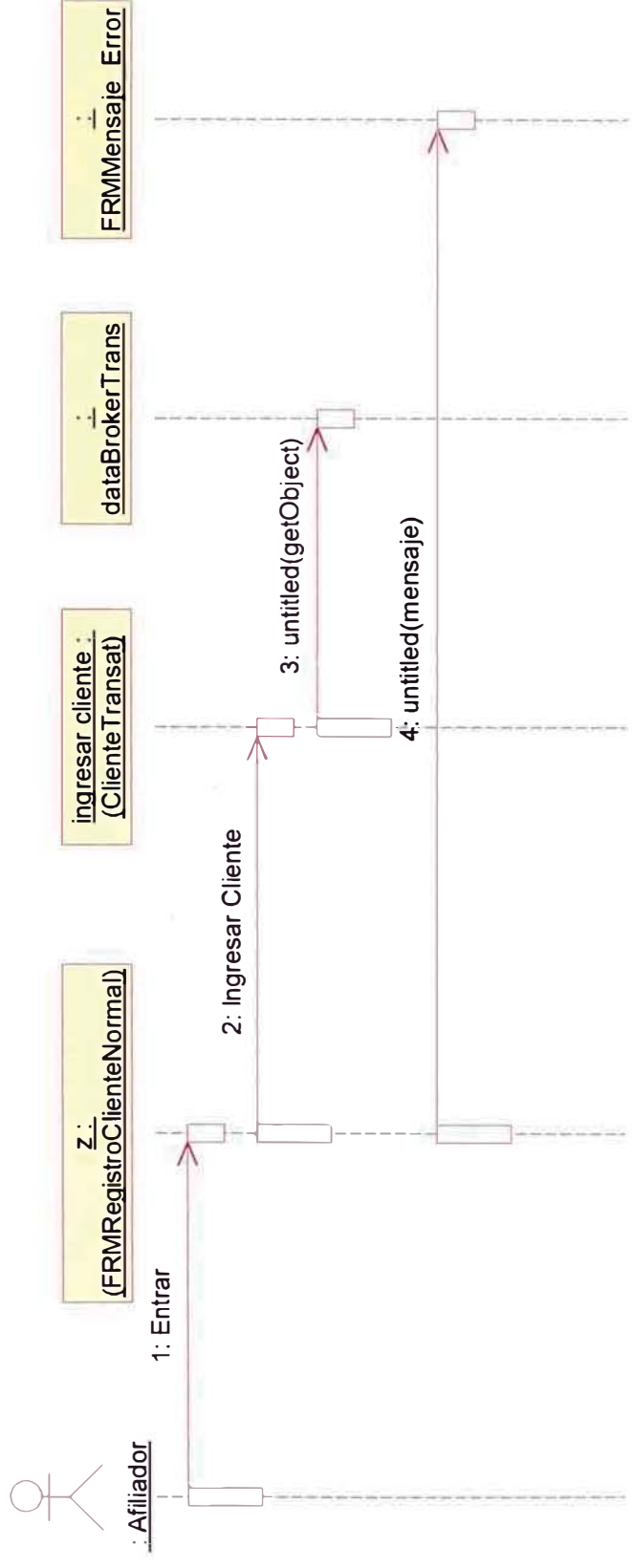








: Afiliador : z : (FRMRegistroClienteNormal) ingresar cliente : (Client... : dataBrokerTrans : FRMMensaje\_Error



: Afili... : FRMClienteVip : ClienteGeneral : Video : Verifica cancelación : (Mo... : Boleta de Venta

