

Universidad Nacional de Ingeniería

FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS



USO DEL RATIONAL UNIFIED PROCESS PARA EL DESARROLLO DE SISTEMA DE SOFTWARE

TESIS

Para Optar el Título Profesional de:

INGENIERO DE SISTEMAS

KLEVER ABEL CAYTUIRO SANDOVAL

FANNY ROSARIO AVILA ROJAS

Lima - Perú

2004

RESUMEN

En este trabajo se presenta la manera como se ha creado un nuevo meta-proceso regido por la realidad encontrada en una entidad estatal, vale decir, el RENIEC (Registro Nacional de Identidad y Estado Civil), todo ello a partir del meta-proceso RUP (Rational Unified Process).

El RUP considera los estándares fijados por el SPEM (Software Process Engineering Metamodel) y se basa en el desarrollo iterativo e incremental, el modelamiento visual con UML, la orientación a objetos, el desarrollo dirigido por casos de uso, la arquitectura basada en componentes, la administración de requerimientos, el manejo de riesgos y la definición de roles y actividades entre otros.

El trabajo central se ha enfocado en la redefinición del RUP para que se adapte el meta-proceso resultante a la realidad del RENIEC.

Para demostrar su validez se ha incorporado el instanciamiento del meta-proceso obtenido en un proyecto piloto del RENIEC que se dirige a la atención en línea de los registros civiles. Los resultados han sido satisfactorios debido a que el sistema obtenido ha respondido a las exigencias de los usuarios finales y se ha desarrollado en el tiempo definido tanto por desarrolladores como por los usuarios del sistema. La documentación del sistema es realmente pequeña pues se basa en esquemas que generan en tiempo real la información que se solicite. Gracias a este trabajo se ha demostrado la necesidad de una nueva área funcional y la definición de nuevos roles para mejorar el trabajo en el departamento de informática del RENIEC.

DESCRIPTORES TEMÁTICOS

- Proceso Unificado.
- Desarrollo iterativo.
- Administración de requerimientos.
- Modelamiento visual.
- Tecnología orientada a objetos.
- Desarrollo dirigido por casos de uso.
- Arquitectura basada en componentes.
- Verificación continua de la calidad.
- Gestión de cambios y configuración.
- Incremento de la productividad en equipo.
- Ingeniería de Software.

ÍNDICE

INTRODUCCIÓN	5
CAPÍTULO I : OBJETIVOS Y ENTORNO DEL PROYECTO.....	7
1.1 OBJETIVO	7
1.2 ENTORNO DEL PROYECTO.....	7
1.2.1 PRESENTACIÓN DE LA ORGANIZACIÓN.....	7
1.2.2 ORGANIZACIÓN DEL PROYECTO.....	15
1.2.3 PROCESO DEL PROYECTO.....	15
CAPÍTULO II : MARCO TEÓRICO	18
2.1 MODELAMIENTO DE PROCESOS.....	18
2.1.1 METODO Y PROCESO.....	18
2.1.2 PROCESO Y PROCESO GENÉRICO	19
2.1.3 INTERÉS DE UN MODELO.....	19
2.1.4 ESTÁNDARES DE MODELAMIENTO	20
2.1.5 MODELAMIENTO PARA EL PROYECTO.....	26
2.2 PRINCIPIOS Y ELEMENTOS DEL PROCESO.....	26
2.2.1 PROCESO DEL DESARROLLO DE SOFTWARE	27
2.2.2 PRÁCTICA DE INGENIERÍA.....	39
2.2.3 PRINCIPIOS.....	40
2.2.4 ELEMENTOS DEL META-PROCESO.....	50
CAPÍTULO III : META-PROCESO PROPUESTO.....	63
3.1 DEFINICIÓN Y REALIZACIÓN DEL META-PROCESO RUPK.....	63

3.1.1 ENTORNO.....	63
3.1.2 DEFINICIÓN DEL META-PROCESO	63
3.1.3 EVALUACIÓN DEL PROCESO ACTUAL.....	64
3.2 META-PROCESO PROPUESTO	71
3.2.1 MODELO	71
CAPÍTULO IV : APLICACIÓN DEL META-PROCESO	83
4.1 INTRODUCCION.....	84
4.2 REQUERIMIENTOS	86
4.2.1 VISIÓN	86
4.2.2 ESPECIFICACIONES SUPLEMENTARIAS	87
4.2.3 GLOSARIO.....	91
4.3 MODELO DEL NEGOCIO	94
4.3.1 MODELO DE CASOS DE USO	94
4.3.2 IDENTIFICACIÓN DE PROCESOS DEL SISTEMA	100
4.3.3 DIAGRAMA DE CLASES DE DOMINIO	106
4.3.4 PROTOTIPEO DE LAS INTERFACES	107
4.3.5 DIAGRAMAS DE NAVEGACIÓN	107
4.4 MODELO DE CONCEPCIÓN	110
4.4.1 CLASES QUE REALIZAN LOS CASOS DE USO	110
4.4.2 DIAGRAMA DE PAQUETES	111
4.4.3 IDENTIFICACION DE OPERACIONES Y ATRIBUTOS.....	112
4.4.4 DIAGRAMA DE CLASES DEL SISTEMA	113
4.5 MODELO DE DESPLIEGUE	114
4.5.1 DIAGRAMA DE COMPONENTES.....	114

4.5.2 DIAGRAMA DE DESPLIEGUE	115
4.6 MODELO DE DATOS	116
4.6.1 RELACIÓN DE ENTIDADES	116
4.6.2 DIAGRAMA ENTIDAD RELACIÓN.....	120
4.7 UTILITARIOS.....	122
CONCLUSIONES Y RECOMENDACIONES.....	124
CONCLUSIONES.....	124
RECOMENDACIONES.....	126
GLOSARIO	128
BIBLIOGRAFIA.....	133
ANEXOS	135
ESPECIFICACIÓN DE CASOS DE USO	135
DIAGRAMAS DE SECUENCIA.....	164
DIAGRAMAS DE COLABORACIÓN	182
DEFINICIÓN DE INTERFACES DE USUARIO	200
COMPARACIÓN DEL VOCABULARIO UTILIZADO.....	232
TABLA COMPARATIVA DEL RUP Y EL RUPK	233

INTRODUCCIÓN

Frente al incremento en la complejidad del desarrollo de sistemas de información, se hace necesaria la identificación de una metodología apropiada que reduzca los niveles de complejidad, sea flexible a las tendencias tecnológicas y comulgue con los distintos puntos de vista de las personas involucradas en el aspecto del negocio al que se aboca el sistema.

Ante esta realidad, se opta por la solución concebida por Rational, el Proceso Unificado de Rational o RUP (Rational Unified Process), que incluye la modelación orientada a objetos y la utilización del Lenguaje de Modelamiento Unificado o UML (Unified Modeling Language).

El Proceso Unificado de Rational, es un proceso de Ingeniería de Software que cubre todo el ciclo de vida de un proyecto, y guía al equipo de desarrollo en la conducción de las actividades de administración y de ingeniería.

Dentro del RUP se ubican dos perspectivas: la perspectiva del administrador y la perspectiva del desarrollador. Para el administrador el sistema se desarrolla a través de cuatro fases o etapas, siendo estas: la etapa de incepción, la etapa de elaboración, la etapa de construcción y la etapa de transición; mientras que para el desarrollador el sistema pasa por un desarrollo iterativo de versiones que van haciendo al mismo cada vez más completo. El RUP plantea el trabajo de sistemas como un conjunto de subsistemas interconectados que bien pueden incluir las fases anteriormente mencionadas. El RUP se focaliza en la generación y mantenimiento de modelos en vez de crear interminables documentos.

Es posible crear productos de calidad, reduciendo la tasa de fallos, con correcta comprensión de los requerimientos del usuario final, una cómoda adaptación a los cambios frecuentes en el dominio del problema, teniendo una documentación generada en tiempo real a partir de los esquemas existentes. Por otro lado, el modelo de tres capas permite definir y distinguir las actividades tanto del lado del cliente, de la Base de Datos y de las reglas del negocio. El manejo de patrones de diseño favorece grandemente al desarrollo rápido de sistemas, por lo que una guía para el modelado de requisitos donde los diagramas de actividad y las clases que representan los conceptos del dominio, se obtienen de una forma directa a partir de un modelado del negocio centrado en el empleo de los casos de uso de UML.

El trabajo se sustenta sobre la demostración del real funcionamiento del RUP como método, y para abordar esta temática se ha escogido como aplicación ejemplo al Sistema de Registros Civiles brindado por el Registro Nacional de Identificación y Estado Civil (RENIEC) a los distintos ciudadanos que deseen acceder a dicho servicio. Como se puede notar, el aplicativo de ejemplo, trata de un sistema que está sometido al constante cambio y a la apertura de más servicios propios de su naturaleza abierta y de innovación tecnológica, sin perder la identificación institucional con el RENIEC.

CAPÍTULO I : OBJETIVOS Y ENTORNO DEL PROYECTO

1.1 OBJETIVO

El objetivo de esta investigación es definir un nuevo meta-proceso en una organización del sector público. Dicho meta-proceso será realizado a partir de un meta-proceso genérico, el RUP (Rational Unified Process). Se desea obtener un meta-proceso que esté de acuerdo a los estándares en el dominio, sea simplificado y adaptable a la organización objetivo, pudiendo evolucionar fácilmente y que además permita presentar un ejemplo ilustrativo sobre su uso en un proyecto.

1.2 ENTORNO DEL PROYECTO

1.2.1 PRESENTACIÓN DE LA ORGANIZACIÓN

La organización que va a utilizar el nuevo meta-proceso es el Registro Nacional de Identificación y Estado Civil (RENIEC), el cual es un organismo público autónomo que cuenta con personería jurídica de derecho público interno y goza de atribuciones en materia registral, técnica, administrativa, económica y financiera. Fue creado por Ley N° 26497 de fecha 12 de julio de 1995.

El RENIEC es el organismo técnico encargado de la identificación de los peruanos, otorga el documento nacional de identidad, registra hechos vitales: nacimientos, matrimonios, defunciones, divorcios y otros que

modifican el estado civil. Es autónomo por mandato de la Constitución y la ley, razón por la cual no pertenece a ningún sector del poder Ejecutivo, Legislativo ni del Poder Judicial y cuenta con la mejor tecnología informática actualizada, además de personal calificado para desempeñarse con éxito en la actividad de registro e identificación de personas como en la aplicación de normas, procesos y procedimientos que rigen esa actividad. Organiza y mantiene el registro único de identificación de las personas naturales; es su razón de ser, existe respondiendo a la necesidad de administrar y dirigir el sistema registral de los peruanos, que involucra el registro civil, registro de personas y registro de naturalización, lo que constituye el registro único y base de datos de identificación de todos los peruanos. En épocas electorales, la única participación del RENIEC es la de proporcionar el padrón electoral inicial al Jurado Nacional de Elecciones, para que éste lo apruebe y a su vez lo remita a la Oficina Nacional de Procesos Electorales como padrón electoral oficial a utilizar el día de las elecciones.

A continuación se procede a presentar la Misión y Visión del RENIEC:

Misión:

Organizar y mantener el Registro Único de Identificación de las personas naturales e inscribir los nacimientos, matrimonios, divorcios, defunciones y otros actos que modifican el Estado Civil.

Visión:

El Registro Nacional de Identificación y Estado Civil será el más avanzado de Latinoamérica en los aspectos de carácter humano y tecnológico.

La organización será adecuada a la realidad de nuestro país, no descuidando la atención en las zonas de menor desarrollo relativo.



Figura 1 Sede Central del Registro Nacional de Identidad y Estado Civil RENIEC

Como toda organización el RENIEC presenta una serie de gerencias, contándose entre ellas a la Gerencia de Informática, la cual tiene a su cargo el desarrollo y mantenimiento de todos los proyectos informáticos.

La Gerencia de Informática cuenta con 5 subgerencias:

- Sub Gerencia de Innovación Tecnológica.
- Sub Gerencia de Soporte Técnico.
- Sub Gerencia de Telecomunicaciones.
- Sub Gerencia de Planificación y Desarrollo de Sistemas Informáticos.
- Sub Gerencia de Base de Datos.

- **Sub Gerencia de Innovación Tecnológica.** Encargada de realizar investigación, normalización, evaluación y propuestas, transferencia de tecnología e innovación tecnológica aplicada a las necesidades de las diversas áreas del RENIEC.
- **Sub Gerencia de Soporte Técnico.** Responsable de asegurar que los recursos informáticos de la institución cubran los requerimientos para la implementación de sistemas, procesos, procedimientos operativos, informáticos, administrativos y estadísticos en un nivel de óptima performance. Planifica, administra, organiza, dirige y controla las actividades y los recursos inherentes, para la implantación, instalación, configuración, mantenimiento y optimización de los recursos informáticos.
 - Especialista Hardware y Software.
 - Especialista Soporte Usuarios.
 - Asistente Técnico Soporte.
- **Sub Gerencia de Telecomunicaciones.** Mantiene la operatividad de las telecomunicaciones de los usuarios de toda la institución, contando con recursos tecnológicos de última generación. Diseña, desarrolla e implementa la arquitectura de telecomunicaciones basados en recursos informáticos y tecnológicos de última generación. Atiende los requerimientos de los usuarios en cuanto a medición de rendimientos, eficiencias, para retroalimentar y mejorar la performance de las redes y las telecomunicaciones.
 - Especialista Redes.

- Especialista Telecomunicaciones.
- Especialista Seguridad Informática.
- **Sub Gerencia de Planificación y Desarrollo de Sistemas Informáticos.** Encargada de planificar, organizar, dirigir y controlar el desarrollo de los proyectos informáticos relacionados con sistemas de información y estadísticos, mediante la sistematización y/o automatización de procesos y actividades funcionales.
 - Jefe de Proyecto Informático.
 - Analista de Sistemas.
 - Analista Funcional.
 - Analista Programador.
 - Técnico Programador.
 - Secretaria
- **Sub Gerencia de Base de Datos.** Responsable de administrar la información perteneciente al RENIEC. Se encarga de garantizar la integridad, confiabilidad y seguridad en el acceso y modificación de la base de datos del RENIEC, así como de establecer mecanismos de registro histórico de modificaciones, autenticación de usuarios, elaboración de reportes y consultas relacionadas con el análisis de la información para implementar el Datawarehouse, la información estadística de gestión, integración de la información y usos de herramientas OLAP, administra y

supervisa componentes de acceso a datos y los servicios en línea en operación.

- Analista de Base de Datos.
- Administrador de Base de Datos.
- Analista de Estadísticos.



Figura 2 Sistema Automatizado de Registros Civiles del RENIEC

Para evaluar el nuevo meta-proceso, se tomará como proyecto piloto a la Automatización de Registros Civiles del RENIEC; el cual se encargará de registrar, digitalizar y emitir certificados de actas de nacimientos, matrimonios y defunciones registradas en las oficinas de Registros Civiles y que permite obtener una copia certificada del acta desde cualquier punto del país de manera inmediata.

El trabajo realizado por esta área se basa en:

- Tecnología orientada a objetos.
- Los casos de uso y la ingeniería de eficiencia.

- El desarrollo iterativo.
- Arquitectura basada en componentes.
- Los lenguajes de modelamiento visuales (UML en general, SDL-Lenguaje Visual que genera código para tiempo real).
- Útiles de modelamiento, simulación y generación de código asociado.

La participación en este proyecto permitirá el desarrollo de una competencia particular en el meta-proceso.

A la fecha no se ha hecho el desarrollo de un meta-proceso moderno referido al Proceso Unificado en la institución. Por ello el desarrollo del meta-proceso permitirá proponer a la institución una opción de trabajo en función a los componentes de un meta-proceso. Los servicios actuales cubrirán las 4 fases del RUP (concepción, elaboración, construcción y transición) y las disciplinas de desarrollo, vale decir aquellas partes del meta-proceso propios del modelamiento. Además se ofrece la información con la notación UML y sus distintas fórmulas adaptadas a cada rol en el proyecto.

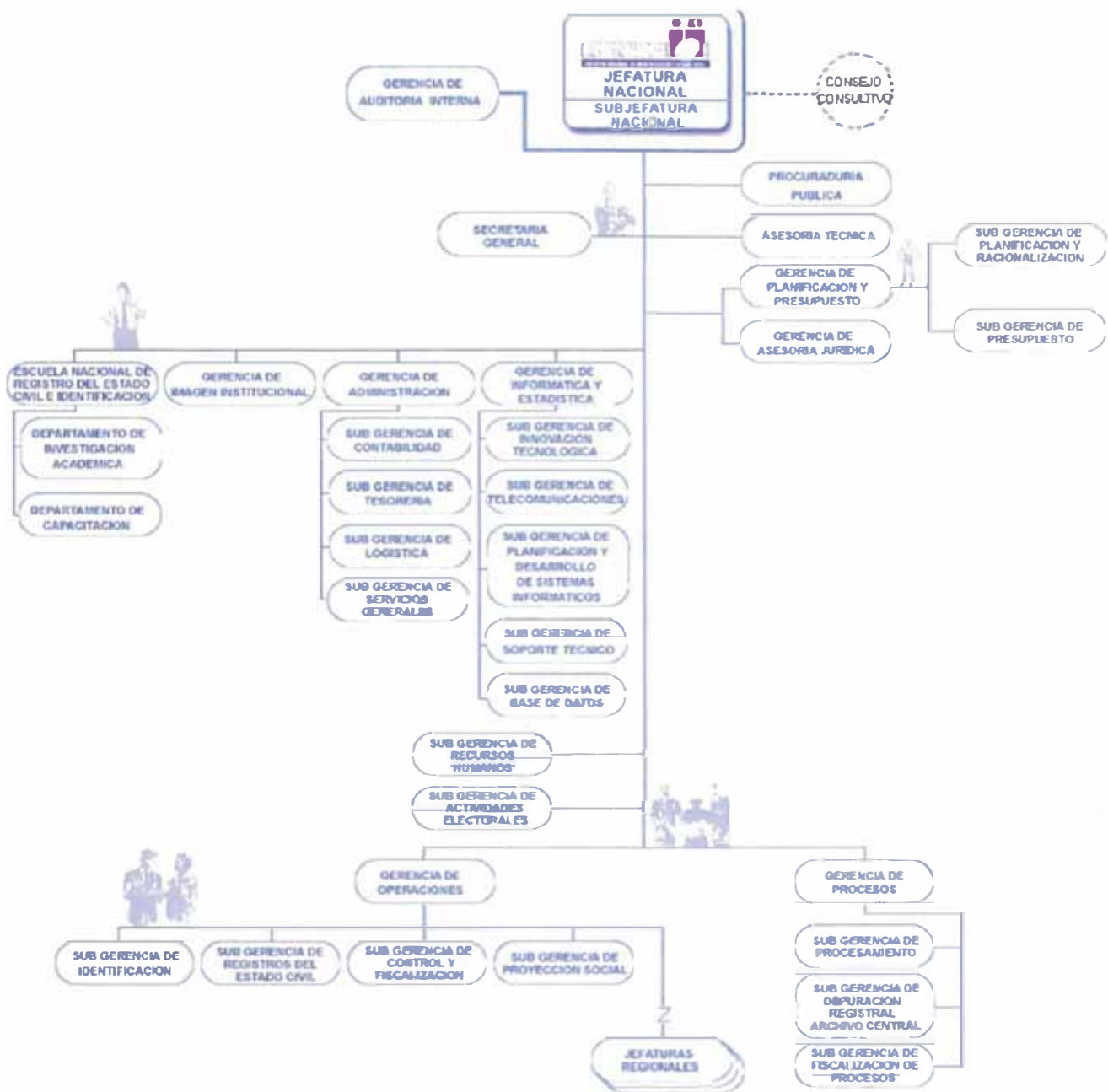


Figura 3 Organigrama del RENIEC.

1.2.2 ORGANIZACIÓN DEL PROYECTO

La Sub Gerencia de Desarrollo de Sistemas Informáticos es responsable de este proyecto. El equipo del proyecto esta constituido por 2 personas con la colaboración periódica de 2 trabajadores pertenecientes a la Gerencia de Operaciones dentro del área de Registros Civiles del RENIEC, encargadas de administrar los Registros Civiles y de interactuar con las distintas municipalidades del país.

El conocimiento de la organización objetivo está asegurado: Desarrollo de Sistemas Informáticos es una Sub Gerencia perteneciente al RENIEC que conoce todos sus procesos internos.

1.2.3 PROCESO DEL PROYECTO

A medida que se aplica el simplificado meta-proceso que retoma los principios del RUP, se pueden distinguir los siguientes lineamientos:

- Gestión de riesgos.
- Definición de productos, roles y actividades.
- Gestión del proyecto con puntos de avance para la toma de decisiones.
- El ciclo iterativo, como una evaluación y una actualización constante de los productos de trabajo a partir de las revisiones y reuniones.

Para implementar un nuevo proceso, es necesario primero evaluar el proceso actual, definir sus evoluciones, planificar su realización, realizar el nuevo proceso, pero además formar la organización para su utilización, probar el proceso sobre uno o muchos procesos pilotos, hacer una

evaluación de esta utilización, para después rehacer uno o muchos de estos ciclos.

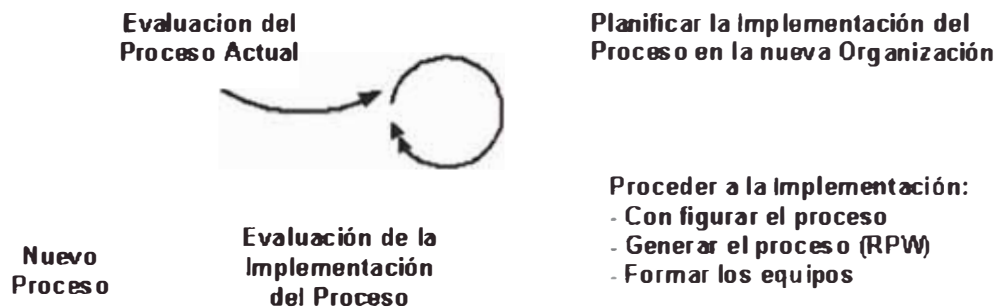


Figura 4 Implementación de un Proceso

El plan de desarrollo inicial esta compuesto por 5 iteraciones, cada una de un mes con los siguientes objetivos:

Iteración 1:

- Evaluación del proceso actual.
- Objetivos del nuevo proceso.
- Lista de riesgos.

Iteración 2:

- Reconocimiento del Rational Rose, Rational XDE, PoseidonUML, ArgosUML, NetBeans y Sun One Studio para el uso de sus herramientas de modelamiento.
- Un proceso implementado con la disciplina de “expresión de exigencias”.

Iteración 3:

- Un proceso implementado con las disciplinas de análisis, concepción y gestión del proyecto.

- Un ejemplo presentado como proyecto para el análisis, concepción y gestión del proyecto.

Iteración 4:

- Un proceso implementado con las disciplinas de Implementación y pruebas.
- Un ejemplo presentado como proyecto para la Implementación y pruebas.

Iteración 5:

- Un proceso implementado con las disciplinas de desarrollo y entorno.
- Un ejemplo presentado como proyecto para el desarrollo y entorno.

CAPÍTULO II : MARCO TEÓRICO

2.1 MODELAMIENTO DE PROCESOS

2.1.1 METODO Y PROCESO

Antes de hablar del modelamiento, debemos notar que el término proceso es relativamente nuevo. Hace algunos años se hablaba de los métodos: SADT, SA/RT, Hood y OMT, los mismos que se presentaban como métodos que ocasionalmente eran restringidos tanto para el análisis como para la concepción de sistemas.

Si nos remontamos a los orígenes del UM (Método Unificado), podemos observar que los primeros trabajos realizados en 1995 estaban totalmente enfocados al trabajo de modelamiento y dejaban de lado el concepto de método. Esta decisión fue el origen del éxito del UML y de su rápida difusión.

Es necesario hacer la distinción entre lenguaje de modelamiento y método: No es posible tener un método único utilizable para todos los proyectos en los distintos campos existentes. Esto se da también en el modelamiento, se pueden facilitar los cambios, pero cada uno debe conservar su cultura y entorno. Un proceso intelectual como aquel de desarrollo, es un bien cultural de una organización.

Por ello, no existe un método universal, ni tampoco un método único referido a la tecnología de objetos y el lenguaje UML.

Los procesos modernos aparecieron después de la estandarización del UML que terminó por completo con los problemas del lenguaje entre los analistas de sistemas.

La noción de proceso es más basta que aquella de método, de hecho se aproxima de manera más cercana a una metodología, vale decir, de un todo que cubre un conjunto de actividades dentro de un proyecto. Los procesos modernos engloban por ejemplo la gestión del proyecto, y no se encasillan solamente en el desarrollo.

2.1.2 PROCESO Y PROCESO GENÉRICO

Los procesos modernos no son aplicables directamente: Ellos definen principios y una arquitectura, pero deben ser adaptados a la organización y al proyecto específico. Son procesos genéricos. Este es el caso del RUP, que por otro lado es presentado como un "framework". El proceso que nosotros hemos desarrollado, RUPK, ha sido creado a partir del proceso genérico RUP. Hemos preferido utilizar el término proceso genérico antes que proceso unificado.

2.1.3 INTERÉS DE UN MODELO

No está en discusión el definir si se debe utilizar un meta-proceso. Tampoco si el meta-proceso debe ser pesado o ligero: es el trabajo necesario para adaptarlo el que definirá su uso, a partir de la evaluación de la organización actual.

Existen dos inconvenientes para el modelamiento:

- El modelamiento siempre es difícil, y la dificultad es mayor al modelar un proceso pues se trata de encarar ciertas actividades humanas.
- El modelamiento toma tiempo y el ciclo de validación es muy largo: es necesario probar el proceso en el proyecto.

El primer beneficio de trabajar con modelos es el de poder hacer preguntas precisas. Los otros beneficios están relacionados a las facilidades brindadas en la comunicación y en la puesta en marcha del proceso, y su generación automática a partir del modelo. Estos beneficios tienen mucho que ver con la existencia de un lenguaje estándar para describir los procesos y de utilitarios para automatizar su creación.

Las entradas y salidas del proceso son los productos de trabajo de un desarrollo. El modelamiento de procesos prepara la transformación de modelo de productos. La mayoría de transformaciones son realizadas actualmente de manera manual. El proceso suministra guías para más adelante. A mediano plazo ciertas transformaciones podrían ser automatizadas.

2.1.4 ESTÁNDARES DE MODELAMIENTO

2.1.4.1 MODELAMIENTO CON UML

La comunidad informática a adoptado rápidamente el UML como estándar de modelamiento para el análisis y diseño de sistemas. Es tentativo utilizar el UML para el modelamiento de procesos. El lenguaje es rico y permite un entendimiento sencillo de las necesidades del entorno con los estereotipos.

Existe una necesidad en algunos para formar una estandarización adaptando el UML al modelamiento de procesos.

2.1.4.2 SPEM

El OMG (Object Management Group), después de la creación del UML hizo un RFP (Request For Proposal) sobre el “Software Process Engineering Management” en Noviembre de 1999. Como resultado de este trabajo nació el SPEM (Software Process Engineering Metamodel). Nosotros hacemos referencia a la versión ad 2001-03-08 difundida el 2 de Abril del 2001.

Los trabajos de la OMG han sido realizados en colaboración con sociedades especializadas en procesos de ingeniería de sistemas tales como IBM, Fujitsu, Unisys, Alcatel y Rational. Es por este motivo que el RUP se tiene gran concordancia con el SPEM.

El resultado de los trabajos es un meta-modelo para la descripción de procesos. Se presenta la utilización del UML con una orientación a objetos para describir los procesos de ingeniería de sistemas.

El objetivo del SPEM es el de definir un lenguaje común para definir los procesos, además de facilitar la comunicación entre los diferentes utilitarios de fabricación de procesos.

El SPEM permite unificar el vocabulario utilizado para describir los procesos. Entre dos procesos, frecuentemente el mismo término es utilizado y comprendido de maneras distintas. Por ejemplo: actividad, fase, iteración.

Una herramienta basada en SPEM sería una herramienta para el uso y personalización de procesos.

2.1.4.3 EL META-MODELO SPEM

La idea central del SPEM es que un proceso es la colaboración entre dos entidades activas y abstractas llamadas **Roles**, que realizan operaciones llamadas **Actividades** sobre entidades concretas y tangibles llamadas **Productos de Trabajo**.

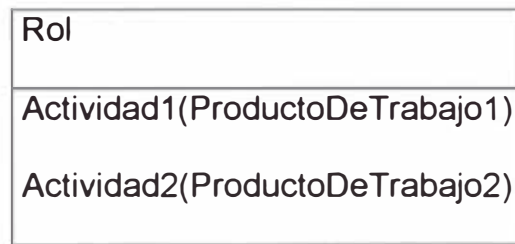


Figura 5 Descripción de un rol a través del diagrama de clases del UML

La figura anterior muestra este concepto fundamental con la notación UML de clases. A partir de este modelo se puede relacionar a la actividad con el producto para obtener un simple modelo (incompleto), que es la base del meta-modelo.

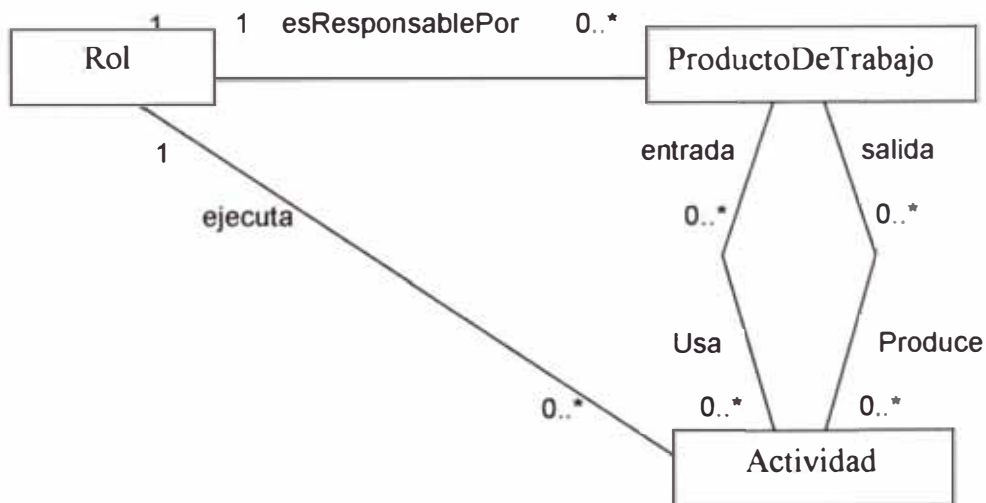


Figura 6 Diagrama UML que muestra las interacciones existentes entre roles, actividades y productos.

Muchos roles pueden colaborar para el intercambio de productos y el desencadenamiento de la ejecución de ciertas actividades. El objetivo global de la ejecución de un proceso es el de proveer un conjunto de productos de trabajo dentro de un estado bien definido.

El SPEM define los siguientes conceptos básicos:

- Las tareas son modeladas como Actividades.
- Las técnicas son modeladas como Guías.
- Los roles son modelados como Roles de Proceso.
- Los productos son modelados como Productos de Trabajo.
- Las fases son modeladas como Fases.

El SPEM utiliza la orientación a objetos para modelar una gran familia de procesos de software relacionados, utilizando la notación UML. Se establece además la arquitectura de cuatro capas del modelamiento definida por la OMG:

- M0. Un proceso en ejecución, vale decir, el proceso de producción en el mundo real, también conocido como la instancia de un modelo.
- M1. La definición de procesos correspondientes. Esta capa incluye procesos genéricos mejor conocidos como personalización específica de estos procesos de acuerdo a como son utilizados en un proyecto.
- M2. Meta-modelo del Proceso (SPEM - UML). Esta capa sirve como plantilla para el nivel M1.

- M3. MetaObject Facility (MOF), también conocido como Meta-meta-modelo.

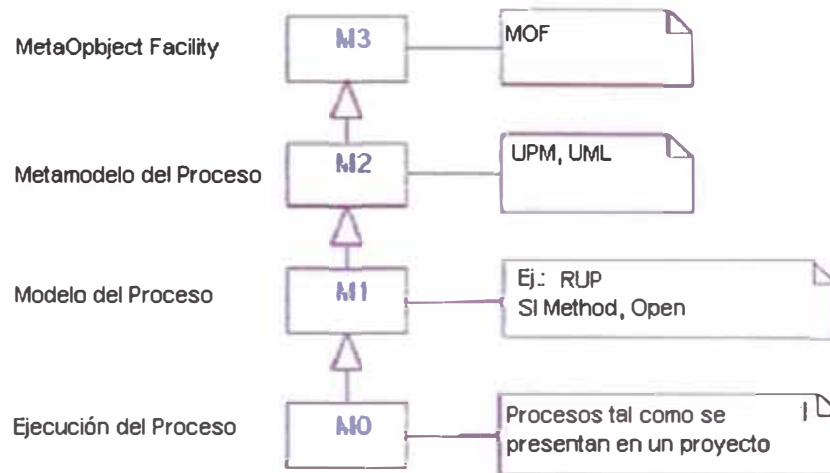


Figura 7 Capas del modelamiento SPEM

La especificación SPEM está estructurada como una especialización UML y también provee un MOF basado en un meta-modelo para facilitar el intercambio con herramientas no basadas en UML.

El SPEM ha sido construido del paquete SPEM_Foundation, el cual es un subconjunto del UML 1.4 y el paquete SPEM_Extensions, los mismos que adicionan constructores y semánticas requeridas por la ingeniería de software.



Figura 8 Los paquetes SPEM Foundation y SPEM Extensions.

La figura 5 presenta la estructura interna del paquete SPEM_Extensions, en términos de sus subpaquetes, y presenta las dependencias entre estos paquetes y los paquetes SPEM_Fundation.

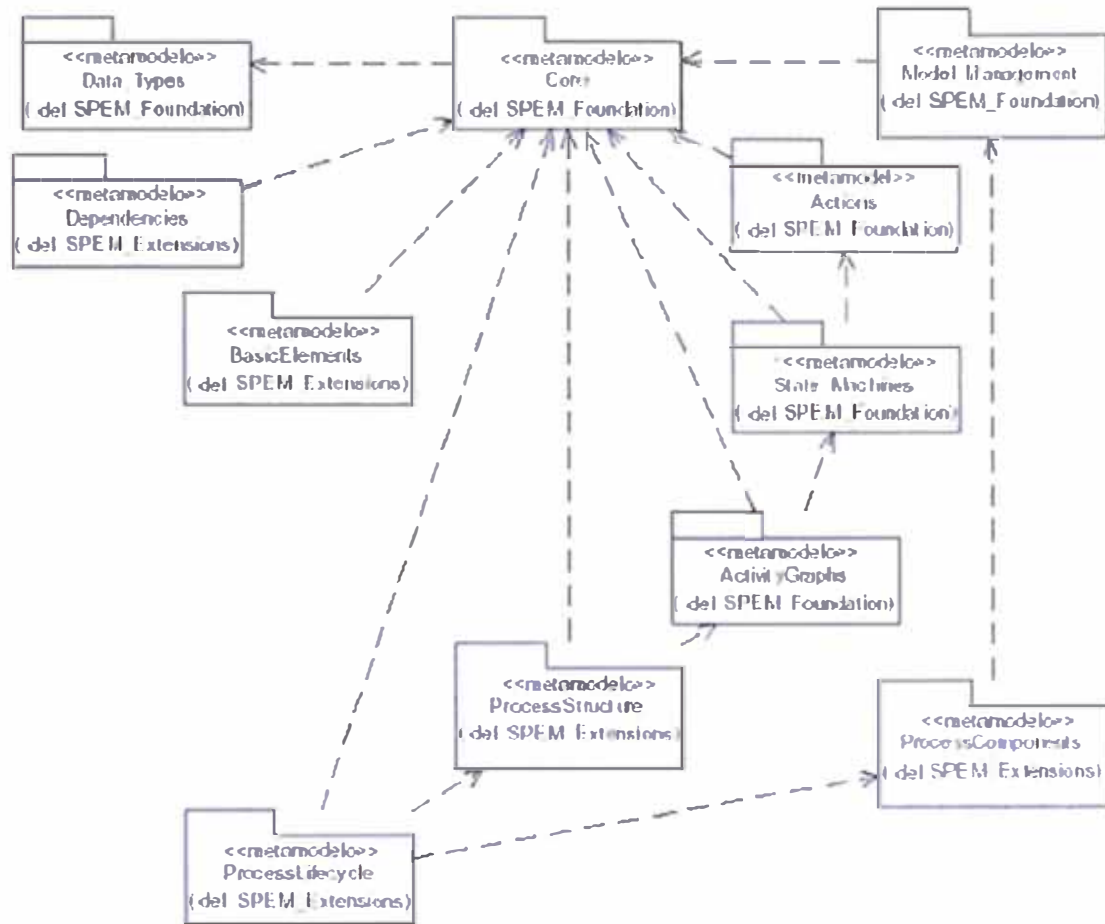


Figura 9 Estructura del paquete SPEM

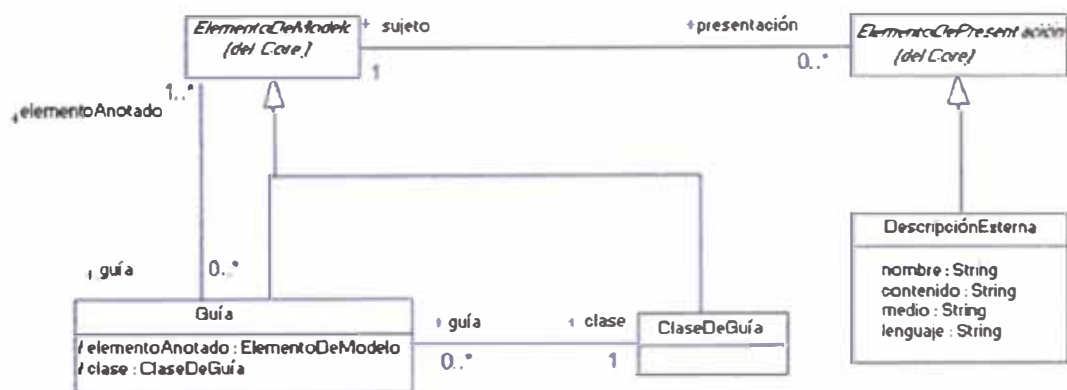


Figura 10 Paquete de Elementos básicos del SPEM

Nosotros no iremos más lejos en la descripción del meta-modelo. La mayoría de conceptos son retomados en la presentación del RUPK.

2.1.5 MODELAMIENTO PARA EL PROYECTO

Nuestro objetivo es el de realizar un meta-proceso a partir del RUP genérico, conforme a los estándares. Una parte del trabajo necesario para producir el RUPK puede realizarse a nivel del modelo. Nuestro trabajo de modelamiento ha consistido en definir nuestro meta-proceso suprimiendo, reutilizando, o especializando algunas partes del modelo dado por el RUP.

Debemos notar que el RUPK es por si mismo un modelo de procesos pudiendo ser “instanciado” e inclusive “heredado” para la definición de otros meta-procesos en otros proyectos.

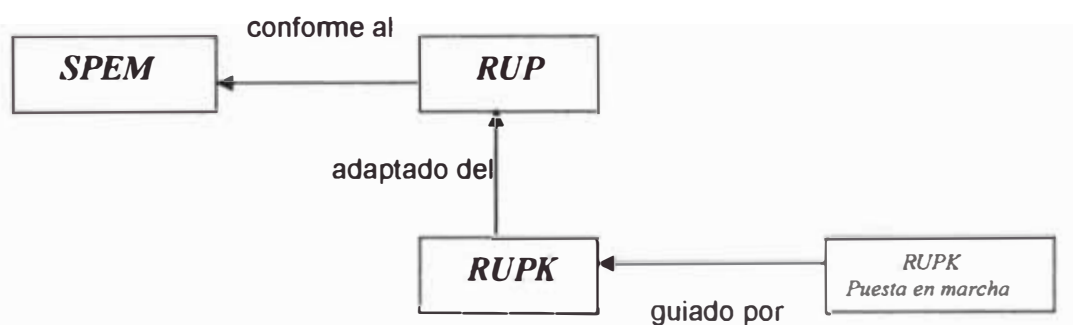


Figura 11 Niveles del modelo

2.2 PRINCIPIOS Y ELEMENTOS DEL PROCESO

Nuestro meta-proceso llamado RUPK es adaptado del RUP. Se apoya sobre las mismas prácticas de ingeniería, retoma la mayoría de sus principios y está compuesto de los mismos tipos de elementos.

2.2.1 PROCESO DEL DESARROLLO DE SOFTWARE

El proceso de desarrollo de software define **quién** está haciendo **qué**, **cuándo** y **cómo** alcanzar un determinado objetivo. En la ingeniería del software el objetivo es construir un producto de software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible. El efecto global es el fomento de una visión y una cultura comunes.

Un proceso de desarrollo de software debería también ser capaz de evolucionar durante un largo período de tiempo. Durante esta evolución debería limitar su alcance en un momento del tiempo dado, a las habilidades que permitan las tecnologías, herramientas, personas y patrones de organización.

- **Tecnologías.** El proceso debe construirse sobre las tecnologías (lenguajes de programación, sistemas operativos, computadores, estructuras de red, entornos de desarrollo, etc.) disponibles en el momento en que se va a emplear el proceso. Por ejemplo, hace veinte años el modelado visual no era realmente de uso general. En aquellos tiempos un creador de un proceso, prácticamente tenía que asumir que se usarían diagramas hechos a mano. Esa suposición limitaba mucho el grado en el cual el creador del proceso podía establecer el modelado dentro del proceso.

- **Herramientas.** Los procesos y las herramientas deben desarrollarse en paralelo. Las herramientas son esenciales en el proceso. Dicho de una forma, Un proceso ampliamente utilizado puede soportar la inversión necesaria para crear las herramientas que lo soporten.
- **Patrones de organización.** Aunque los desarrolladores de software no pueden ser expertos tan independientes, están muy lejos de ser trabajadores autómatas. El creador del proceso debe adaptar el proceso a las realidades del momento.

Los ingenieros del proceso deben equilibrar estos cuatro conjuntos de circunstancias. Además, el equilibrio debe estar presente no sólo ahora, sino también en el futuro. El creador del proceso debe diseñar el proceso de forma que pueda evolucionar, de igual forma que el desarrollador intenta desarrollar un sistema que además de funcionar debe evolucionar con éxito en los años sucesivos. Un proceso debe madurar durante varios años antes de alcanzar el nivel de estabilidad que le permitirá resistir los rigores del desarrollo de productos comerciales, manteniendo a la vez un nivel razonable de riesgo en su utilización. El desarrollo de un producto nuevo es muy arriesgado en sí mismo como para añadirle el riesgo de un proceso que esté poco validado por la experiencia de su uso. En estas circunstancias, un proceso puede ser estable. Sin este equilibrio de tecnologías, herramientas, personas y organización, el uso del proceso sería bastante arriesgado.

2.2.1.1 ANTECEDENTES

El Proceso Unificado está equilibrado por ser un producto final de tres décadas de desarrollo y uso práctico. Su desarrollo como

producto sigue un camino, desde el Proceso Objectory (primera publicación en 1987) pasando por el Proceso Objectory de Rational (publicado en 1997) Hasta el Proceso Unificado de Rational (publicado en 1998). Su desarrollo ha recibido influencias de muchas fuentes.

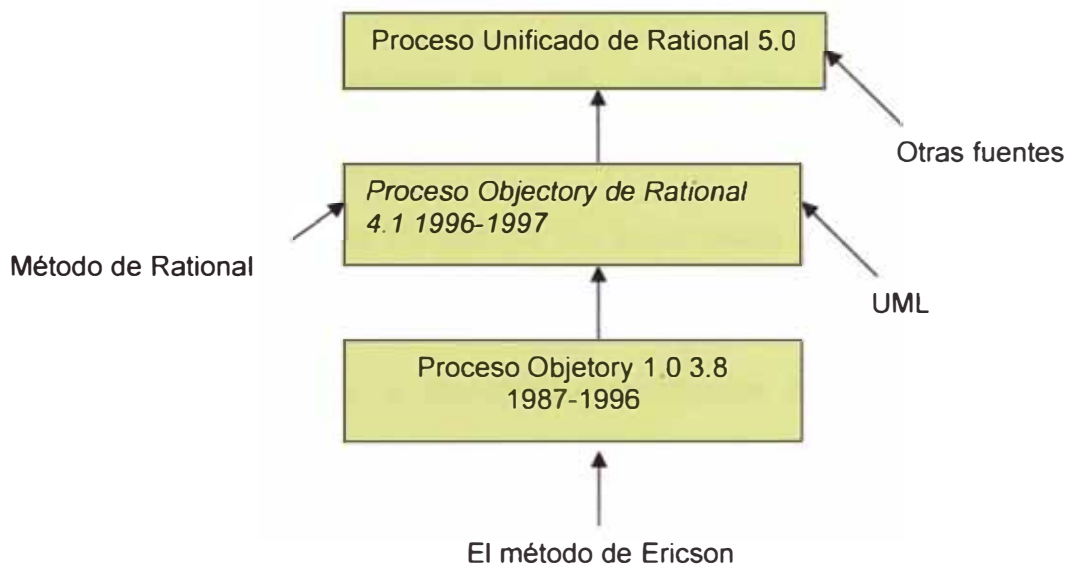


Figura 12 Secuencia de soluciones del Proceso Unificado

2.2.1.1.1 El método de Ericson

El Proceso Unificado posee raíces profundas, dado que gran parte de su innovación se basa en el conocimiento. Como primer paso hacia el alumbramiento del desarrollo del Proceso Unificado, nos remontaremos a 1967 para esbozar los logros de Ericson. Ericson modelaba el sistema entero como un conjunto de bloques interconectados (en UML, se los conoce como “subsistemas” y se implementan mediante “componentes”). Después, ensamblaba los bloques de más bajo nivel en subsistemas de más alto nivel, para hacer el sistema más manejable. Identificaban los bloques estudiando los casos de negocio (ahora conocidos como “casos de uso”),

previamente especificados. Para cada caso de uso, identificaban los bloques que debían cooperar para realizarlo. Con el conocimiento de las responsabilidades de cada bloque, preparaban su especificación. Sus actividades de diseño producían un conjunto de diagramas de bloques estáticos con sus interfases, agrupados en subsistemas. Estos diagramas de bloques se corresponden directamente con una versión simplificada de los diagramas de clases o paquetes de UML, pues sólo se mostraban las asociaciones que se utilizaban para comunicaciones.

El primer producto del trabajo de las actividades de diseño era una descripción de arquitectura de software. Se basaba en una comprensión de los requisitos más críticos, y describía brevemente cada bloque y su agrupamiento en subsistemas. Un conjunto de diagramas de bloques describía a los bloques y a sus interconexiones. Sobre las interconexiones se comunicaban señales, es decir, un tipo de mensaje. Todos los mensajes quedaban descritos, uno por uno, en una biblioteca de mensajes. La descripción de la arquitectura de software y la biblioteca de mensajes eran los documentos fundamentales que guiaban el trabajo de desarrollo, pero también se utilizaban para presentar el sistema a los clientes. En aquellos momentos (1968) los clientes no estaban acostumbrados a que se les presentasen los productos de software por medios similares a los datos de proyectos de ingeniería.

Para cada caso de uso, los ingenieros preparaban bien un diagrama de secuencia o bien un diagrama de colaboración. Estos diagramas mostraban cómo los bloques se comunicaban dinámicamente para llevar a

cabo el caso de uso. Preparaban una especificación en forma de grafo de estados (que incluía sólo estados y transiciones) y un grafo de transición de estados (una versión simplificada del diagrama de actividades de UML). Este método, el diseñar a partir de bloques con interfases bien definidas, fue la clave del éxito. De ese modo, se podía crear una nueva configuración del sistema, intercambiando un bloque por otro que proporcionase las mismas interfases.

Por tanto, los bloques no eran sólo subsistemas y componentes de código fuente; se compilaban en bloques ejecutables, se instalaban en la máquina destino uno por uno, y se comprobaba que funcionaban con el resto de los bloques ejecutables. Más aún, debía ser posible instalar cada bloque ejecutable, nuevo o modificado, sobre la marcha en un sistema en ejecución, mientras éste gestionaba llamadas de sistemas de telefonía en operación durante el 100 por ciento del tiempo. No se puede parar sistemas de ese tipo sólo para realizar cambios.

En esencia, el método que utilizaban era el que hoy conocemos como desarrollo basado en componentes. Ivar Jacobson fue el creador de este método de desarrollo. Él dirigió su evolución hacia un proceso de desarrollo de software durante muchos años en el periodo anterior a Objectory.

2.2.1.1.2 El Lenguaje de Descripción y Especificación

Un avance significativo durante este periodo fue la publicación en 1976 por parte del CCITT, el organismo internacional para la estandarización en el área de telecomunicaciones, del lenguaje de Especificación y Descripción (Specification and Description Language SDL) para el

comportamiento funcional de los sistemas de telecomunicación. Este estándar, influenciado significativamente por el método de Ericson, especificaba un sistema como un conjunto de bloques interconectados que se comunicaban unos con otros, únicamente a través de mensajes (llamados “señales” en el estándar). Cada bloque contenía un conjunto de “procesos”, que era el término SDL para designar las clases activas. Un proceso poseía instancias de manera muy parecida a cómo lo hacen las clases en términos de orientación a objetos. Las instancias de los procesos interactuaban mediante mensajes. SDL proponía diagramas que eran especializaciones de lo que hoy UML llama diagramas de clases, diagramas de actividad, diagramas de colaboración y diagramas de secuencia.

Por tanto, SDL era un estándar de modelado de objetos especializado. Se actualiza periódicamente, y todavía lo utilizan más de 10 000 desarrolladores y cuenta con el soporte de varios fabricantes de herramientas. Fue desarrollado inicialmente hace veinte años, y estaba muy por delante de su tiempo. Sin embargo, se desarrolló en un momento en el cual el modelado de objetos no había madurado. Probablemente, SDL será sustituido por UML, que se estandarizó en 1997.

2.2.1.1.3 Objectory

En 1987, Ivar Jacobson dejó Ericson y fundó Objectory AB en Estocolmo. Durante los siguientes 8 años, él y sus colaboradores desarrollaron un proceso denominado Objectory (abreviatura de “Object Factory”, fábrica de objetos). Extendieron su uso en otras industrias además de las telecomunicaciones, y en otros países aparte de Suiza.

Aunque el concepto de caso de uso había estado presente en el trabajo de Ericson, ahora se le había dado un nombre (que se presentó en la conferencia OOPSLA de 1987), se había desarrollado una técnica de representación, y la idea se había ampliado para abarcar más variedad de aplicaciones. Es decir, los casos de uso que dirigían el desarrollo se hicieron más claros. Dicho de otro modo, la arquitectura que conduce a los desarrolladores e informa a los usuarios comenzó a destacar.

Los flujos de trabajo sucesivos se representaron en una serie de modelos: requisitos-casos de uso, análisis, diseño, implementación y prueba. Un modelo es una perspectiva del sistema. Las relaciones entre los modelos de esta serie eran importantes para los desarrolladores como forma de hacer el seguimiento de una característica de un extremo a otro de la serie de modelos. De hecho, la trazabilidad se convirtió en un prerrequisito del desarrollo dirigido por casos de uso. Los desarrolladores podían seguir la traza de un caso de uso a través de la secuencia de modelos hasta el código fuente o bien, cuando surgían problemas, volver hacia atrás.

El desarrollo del proceso Objectory continuó en una serie de versiones, desde el Objectory 1.0 en 1988 a la primera versión iterativa, Objectory 3.8, en 1995.

Es importante hacer notar que el propio producto Objectory llegó a ser visto como un sistema. Esta forma de ver un producto (como un producto en forma de sistema) proporcionaba una manera mejor de desarrollar una nueva versión de Objectory a partir de una anterior. Este modo de desarrollar Objectory hizo más fácil ajustarlo para cubrir las necesidades

específicas de diferentes organizaciones de desarrollo. El hecho de que el propio proceso de desarrollo de software Objectory era un producto de ingeniería era una característica única.

La experiencia en el desarrollo de Objectory también aportó ideas sobre cómo diseñar los procesos generales sobre los cuales opera un negocio. Eran aplicables los mismos principios y estos se recogieron en un libro en 1995.

2.2.1.1.4 El método de Rational

Rational Software Corporation compró Objectory AB a finales de 1995 y la tarea de unificar los principios básicos subyacentes en los procesos de desarrollo existentes adquirió una urgencia especial. Rational había desarrollado algunas prácticas de desarrollo de software, la mayoría de ellas complementarias a las contenidas en el Objectory.

Por ejemplo, como recordaron James E. Archer Junior y Michael T. Devlin en 1986, En 1981, Rational se dispuso a crear un entorno interactivo que mejoraría la productividad en el desarrollo de grandes sistemas de software.. A continuación dijeron que en este esfuerzo, eran importantes el diseño orientado a objetos, la abstracción, la ocultación de la información, la reutilización y el prototipeo.

Muchos libros, artículos y documentos internos detallan los desarrollos de Rational desde 1981, pero quizás las dos contribuciones más importantes al proceso fueron los énfasis en la arquitectura y el desarrollo iterativo. Por ejemplo, en 1990 Mike Devlin escribió un artículo introductorio sobre un proceso de desarrollo iterativo dirigido por la arquitectura. Philippe

Krutchén, a cargo de la división de Prácticas de Arquitectura en Rational, firmó artículos sobre la iteración y la arquitectura.

Mencionaremos uno de ellos, un artículo sobre una representación de la arquitectura con cuatro vistas: la vista lógica, la vista de procesos, la vista física y la vista de desarrollo, más una vista adicional que ilustraba las primeras cuatro vistas mediante casos de uso o escenarios. La idea de tener un conjunto de vistas, en lugar de tratar de meter todo en un único tipo de diagrama, nació de la experiencia de Krutchén en varios proyectos grandes. Las vistas múltiples permitieron encontrar, tanto a los usuarios como a los desarrolladores, lo que necesitaban para sus diferentes objetivos con la vista adecuada.

Hay gente que percibe el desarrollo iterativo como algo caótico o anárquico. El método de cuatro fases (comienzo, elaboración, construcción y transición) se diseñó para estructurar mejor y controlar el proceso durante las iteraciones. La planificación detallada de las fases y la ordenación de las iteraciones dentro de las fases fue un esfuerzo conjunto entre Walker Royce y Rich Reitman, junto con la participación continua de Grady Booch y Philippe Krutchén.

Booch estaba presente desde los principios de Rational, y en 1996 en uno de sus libros mencionó dos principios fundamentales sobre la arquitectura y la iteración:

- “Un estilo de desarrollo dirigido por la arquitectura es normalmente la mejor aproximación para la creación de la mayoría de los proyectos complejos basados en el software”.

- “Para que un proyecto orientado a objetos tenga éxito, debe aplicarse un proceso iterativo e incremental”.

2.2.1.1.5 El Proceso Objectory de Rational: 1995-1997

En el momento de la fusión, Objectory 3.8 había demostrado que se puede crear y modelar un proceso de desarrollo de software como si fuese un producto. Había diseñado la arquitectura original de un proceso de desarrollo de software. Había identificado un conjunto de modelos que documentaban el resultado del proyecto. Estaba correctamente desarrollado en áreas como el modelado de casos de uso, análisis y diseño, aunque en otras áreas (gestión de requisitos aparte de los casos de uso, implementación y pruebas) no estaba tan bien desarrollado. Además, decía poco sobre gestión del proyecto, gestión de la configuración, distribución, y sobre la preparación del entorno de desarrollo (obtención del proceso y las herramientas).

Por ello se le añadieron la experiencia y prácticas de Rational para formar el Proceso Objectory de Rational 4.1. Se añadieron, en concreto, las fases y la aproximación iterativa controlada. Se hizo explícita la arquitectura en forma de una descripción de la arquitectura (el núcleo de la organización de desarrollo de software).

Se desarrolló una definición precisa de la arquitectura, considerada como la parte más significativa de la organización del sistema. Representaba la arquitectura como vistas arquitectónicas de los modelos. Se amplió el desarrollo iterativo, pasando de ser un concepto relativamente general a ser

un método dirigido por los riesgos que consideraba la arquitectura en primer lugar.

En estos momentos UML estaba en fase de desarrollo y se incorporó como el lenguaje de modelado del Proceso Objectory de Rational (Rational Objectory Process, ROP). Sus autores del ROP colaboraron como creadores originales del UML. El equipo de desarrollo del proceso, liderado por Philippe Krutchen, corrigió algunas de las debilidades del ROP reforzando, por ejemplo, la gestión del proyecto, basada en aportaciones de Royce.

2.2.1.1.6 El Lenguaje Unificado de Modelamiento

Era evidente desde hace algún tiempo la necesidad de un lenguaje de modelado visual y consistente, en el cual expresar los resultados de las numerosas metodologías de orientación a objetos existentes a principios de los noventa. Durante este periodo, Grady Booch, por ejemplo, era autor del método Booch, y James Rumbaugh era el desarrollador principal de OMT (Object Modeling Technique) creado en el Centro de Investigación y Desarrollo de General Electric. Cuando este último se incorporó a Rational en Octubre de 1994, los dos comenzaron el trabajo de unificar sus métodos, en coordinación con muchos de los clientes de Rational. Publicaron la versión 0.8 del Método Unificado en Octubre de 1995, casi en el mismo momento de la incorporación de Ivar Jacobson a Rational.

Los tres, trabajando juntos, publicaron la versión 0.9 del Lenguaje Unificado de Modelado. El esfuerzo se amplió para incluir a otros metodologistas, así como a diversas empresas que incluían a IBM, HP y Microsoft, cada una de las cuales contribuyó al estándar en evolución. En

noviembre de 1997, después de pasar por el proceso de estandarización, el Object Management Group publicó como estándar la versión 1.1 del Lenguaje Unificado de Modelado.

2.2.1.1.7 El Proceso Unificado de Rational

Durante este periodo, Rational compró o se fusionó a otras empresas fabricantes de herramientas. Cada una de ellas aportó a la mezcla su experiencia en áreas del proceso que ampliaron más aún el proceso Objectory de Rational:

- Requisite Inc. aportó su experiencia en gestión de requisitos.
- SQA Inc. había desarrollado un proceso de prueba para acompañar a su producto de pruebas, y lo añadió a la dilatada experiencia de Rational en este campo.
- Pure-Atria añadió su experiencia en gestión de configuración a la de Rational.
- Performance Awareness añadió las pruebas de rendimiento y las de carga.
- Vigortech añadió su experiencia en ingeniería de datos.

El proceso también se amplió con un nuevo flujo de trabajo para el modelado del negocio, que se utiliza para obtener requisitos a partir de los procesos de negocio que el software va a cubrir. También se extendió con diseño de interfases de usuario dirigido por los casos de uso (basado en el trabajo de Objectory AB).

A mediados de 1998 el Proceso Objectory de Rational se había convertido en un proceso capaz de soportar el ciclo de vida del desarrollo en

su totalidad. Para ello, integraba una amplia variedad de aportaciones, no sólo de los "tres amigos", sino de las muchas fuentes sobre las cuales se basaron Rational y UML. En junio, Rational publicó una nueva versión del producto, el Proceso Unificado de Rational 5.0. En ese momento, por primera vez, se pusieron a disposición del público en general muchos elementos de ese proceso propietario.

El cambio de nombre refleja el hecho de que la unificación a tenido lugar en muchas dimensiones: unificación de técnicas de desarrollo, a través del Lenguaje Unificado de Modelado, y unificación del trabajo de muchos metodologistas (no sólo en Rational sino también en las oficinas de los cientos de clientes que llevaban utilizando el proceso muchos años).

2.2.2 PRÁCTICA DE INGENIERÍA

El RUP descansa sobre 6 pilares:



- Desarrollo iterativo.
- Gestión de exigencias.
- Modelamiento visual.
- Arquitectura basada en componentes.
- Verificación continua de la calidad.
- Gestión de cambios y configuración.

2.2.3 PRINCIPIOS

2.2.3.1 ADAPTACIÓN AL DESARROLLO Y AL MANTENIMIENTO

Un proceso es un conjunto de etapas parcialmente ordenadas donde la ejecución se dirige hacia la atención de un objetivo: dentro del campo de la Ingeniería de Sistemas este objetivo es la realización de un producto o su mantenimiento.

Explicado en términos de modelamiento, un proceso de Ingeniería de Sistemas es un proceso donde el objetivo es la mejora de la organización que desarrolla los sistemas, el Rational Unified Process (RUP) es un proceso genérico para el desarrollo informático orientado a objetos.

El proceso tiene por finalidad asegurar la producción de un sistema de calidad que responda a los requerimientos de los usuarios finales, con respecto a costos y tiempos; para ello se toman los siguientes principios:

- Dar una aproximación ordenada de la asignación de tareas y responsabilidades al interior de la organización encargada del desarrollo.
- Incluir todo lo referente al mantenimiento. Cuando un sistema es desarrollado paso a paso, el desarrollo es el proceso de creación de un sistema a partir de las exigencias, pero una vez que el sistema ha tomado forma (después de haber pasado el ciclo de desarrollo inicial), todo desarrollo posterior es un proceso de conformidad del sistema a nuevas exigencias o exigencias que han sido modificadas. Todo lo anteriormente expuesto se aplica a lo largo de todo el ciclo de vida del sistema.

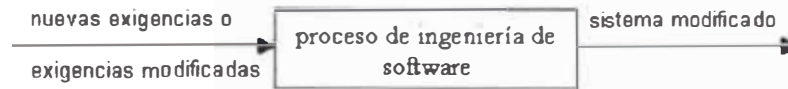


Figura 13 Objetivo de un proceso

2.2.3.2 PROCESO DE DOS DIMENSIONES

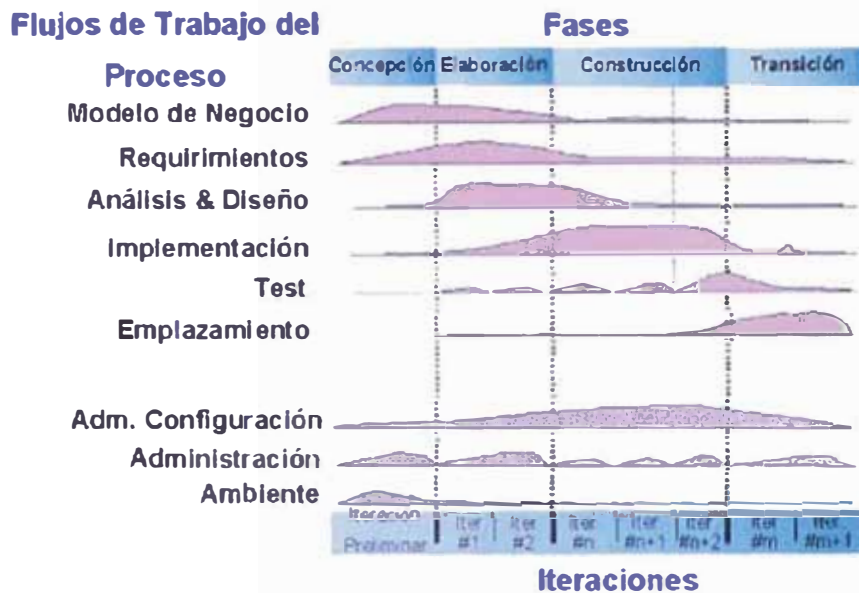


Figura 14 Presentación esquemática del proceso de acuerdo a dos ejes

El proceso trabaja en función a dos ejes:

- El eje horizontal representa la secuencia del trabajo en el tiempo: el ciclo de vida del proceso, y su explicación en términos de fases, iteraciones y cadenas.
- El eje vertical representa la organización del trabajo en términos de componentes del proceso (disciplinas, flujos de trabajo, grupos de actividades, productos, actividades, roles, etc.).

Esta distinción es fundamental pues nos permite ubicar realmente las iteraciones.

2.2.3.3 CICLO ITERATIVO E INCREMENTAL

El principal problema del ciclo "en V" (ciclo de vida del software que define la ejecución de las actividades y procesos de manera secuencial) es que encara la gestión de los riesgos demasiado tarde dentro del desarrollo, de tal manera que sus ocurrencias son muy costosas debido a que se trata de reparar errores ocurridos en fases precedentes. Esto conduce a retrasos, sobrecostos e inclusive en casos, la anulación del proyecto.

Una alternativa es la de contar con un ciclo iterativo e incremental. Inspirado en el modelo en espiral de Barry Boehm, se basa en la identificación de riesgos en el proyecto de manera temprana dentro del ciclo de vida, cuando aún es posible atenderlos, atenuarlos, y redimensionarlos. Otra característica de este ciclo es la elaboración de productos tangibles (iterativamente hasta su finalización) a lo largo de cada fase: documentos, prototipos, modelos, código fuente, ejecutables, etc..

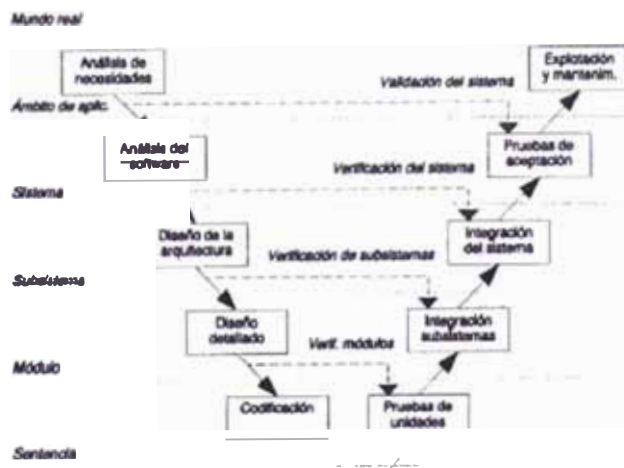


Figura 15 Ciclo de Vida del Software en V

Ello permite arreglar ciertos problemas puntuales del desarrollo informático:

- Eliminación del efecto túnel. Los elementos son producidos tempranamente y verificados al final de cada iteración o fase y no al final del proyecto.
- Mejor comunicación entre los desarrolladores y los usuarios finales del sistema a través de la disciplina de la expresión de las exigencias.
- El equipo de desarrollo sólo se concentra en un momento determinado a los riesgos más críticos.
- Las pruebas continuas de los productos permiten evaluar objetivamente el avance del proyecto (lo cual disminuye además la carga de las pruebas al final del proyecto debido a que las pruebas son repartidas a lo largo de todo el proyecto).
- Las incoherencias entre las exigencias, concepción y elaboración son detectadas tempranamente.
- El equipo de trabajo mejora continuamente el proceso a través de su experiencia y conocimientos adquiridos de proyectos pasados.
- La flexibilidad del ciclo de desarrollo permite dirigir de manera más sencilla y oportuna las demandas de modificaciones o la ocurrencia de riesgos.
- El esfuerzo del desarrollo es constante a lo largo de todo el trabajo; los elementos producidos son integrados a la medida.
- Facilidad del reuso por la descomposición de componentes.

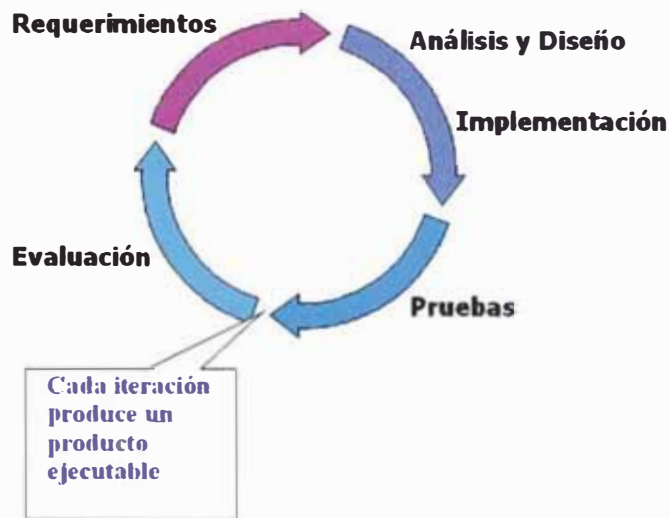


Figura 16 Desarrollo Iterativo

2.2.3.4 FASES

Las iteraciones se realizan dentro de un cuadro de fases. Todas las fases son distintas en términos de duración y esfuerzo.

El ciclo de vida se compone de cuatro fases: concepción, elaboración, construcción y transición. Desde una perspectiva de gestión del proyecto, cada una de estas 4 fases secuenciales concluye con un hito importante.

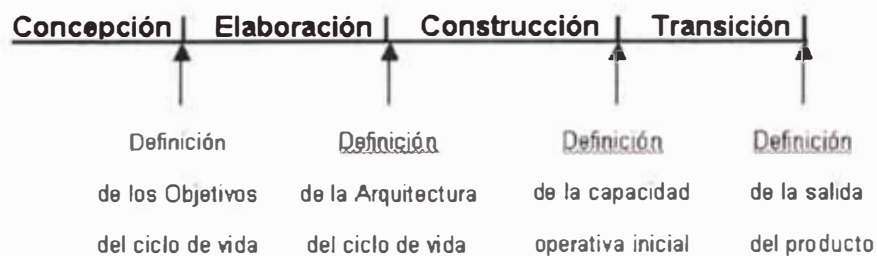


Figura 17 El encadenamiento de las fases y sus hitos

2.2.3.4.1 FASE DE CONCEPCIÓN

La concepción es la fase a través de la cual se decide si es oportuno realizar o no el proyecto. Los objetivos principales de la fase de concepción son los siguientes:

- Definir una visión compartida del proyecto, que contenga los criterios de aceptación.
- Determinar los casos de uso críticos del sistema, los escenarios donde se localizan las principales interrogantes de la concepción.
- Mostrar por lo menos una arquitectura que sea aplicable a estos distintos escenarios.
- Estimar de manera global los costos y tiempos del proyecto (se entraría en más detalle dentro de la fase de elaboración).
- Estimar los riesgos potenciales (todo aquello que pueda generar imprevistos).
- Preparar el entorno de desarrollo del proyecto.

2.2.3.4.2 FASE DE ELABORACIÓN

En esta fase se tiene por finalidad obtener una arquitectura estable que pueda ser utilizada como una base sólida para la implementación dentro de la fase de construcción.

La arquitectura depende de:

- Las exigencias que tienen mayor impacto sobre la arquitectura del sistema.
- La evaluación de riesgos.

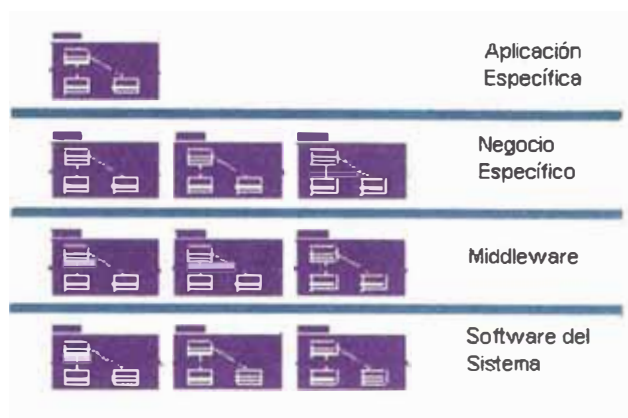


Figura 18 Desarrollo de la Arquitectura.

La estabilidad de la arquitectura queda demostrada a través de uno o varios prototipos de arquitectura. Los objetivos principales de la fase de elaboración son los siguientes:

- Llegar a la certeza que la arquitectura (a partir de los escenarios significativos), las exigencias y los planes son lo suficientemente estables, que los riesgos significativos desde el punto de vista de la arquitectura sean lo suficientemente aminorados para poder elaborar previsiones fiables dentro del marco de costos y la duración del desarrollo restante.
- Producir un prototipo evolutivo con componentes de calidad, así como uno o varios prototipos “de exploración” lanzados para atenuar los riesgos (cambio en los conceptos, cambios en las exigencias, reuso de componentes, flexibilidad del producto) o para hacer una demostración a los clientes.
- Demostrar que la arquitectura soportará las exigencias del sistema (en términos de costos y tiempos razonables).

- Definir y ubicar el entorno de desarrollo (redacción del plan del ciclo de vida del proyecto, planes tipo, guías, y pre-configuración de utilitarios).

2.2.3.4.3 FASE DE CONSTRUCCIÓN

Dentro de la fase de construcción se realiza el sistema a partir de la arquitectura establecida. La fase de construcción es de cierto modo la etapa de producción, donde la atención es puesta en la gestión de los recursos y el control de las operaciones para optimizar costos, tiempos y calidad.

Los principales objetivos de la fase de construcción son los siguientes:

- Minimizar los costos de desarrollo por medio de la optimización de recursos. Es esencial disponer de una arquitectura robusta si se desea lograr un alto grado de paralelismo de los recursos.
- Alcanzar la calidad adecuada rápidamente.
- Producir versiones utilizables (alfa, beta o cualquier otra para efectos de prueba) lo más rápidamente posible.
- Completar el análisis, desarrollo y pruebas de todas las funcionalidades.
- Desarrollar iterativamente y de manera incremental un producto completo listo para la transición hacia la comunidad de usuarios (ello implica la descripción de los casos de uso restantes, enriquecer la concepción, completar la implementación, y probar el sistema).
- Decidir si el sistema, los usuarios y distintos puntos están listos para el despliegue de la aplicación.

2.2.3.4.4 FASE DE TRANSICIÓN

A través de la fase de transición se asegura que el sistema esté disponible para los usuarios finales. La fase de transición puede escalarse sobre varias iteraciones, incluyendo la prueba del producto antes de su salida y ajustes menores basados en las recomendaciones de los usuarios (para mejorar de cierta forma la configuración, instalación y forma de uso).

Al final de la fase de transición los objetivos deben haber sido atendidos y el proyecto debe estar en el punto de cierre.

Los objetivos principales de la fase de transición son los siguientes:

- El acuerdo entre las personas que intervienen en el proyecto sobre los criterios de aceptación del producto que debe cumplir la versión a ser entregada.
- La prueba beta para validar el nuevo sistema en función de los requerimientos de los usuarios.
- La instalación de bases de datos operacionales.
- La formación de usuarios y responsables del mantenimiento.
- La corrección de errores, mejora de la performance y uso del producto.

También es importante garantizar la autonomía del usuario en el sistema.

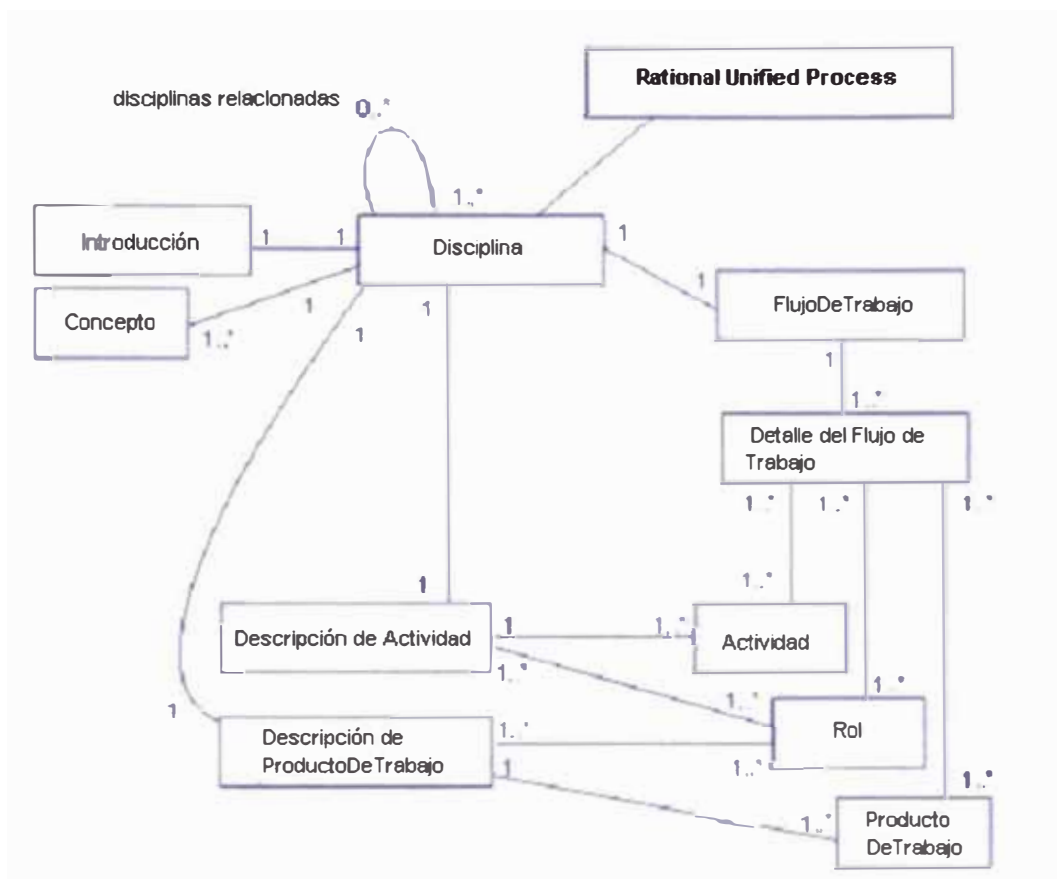


Figura 19 Vista de alto nivel del RUP

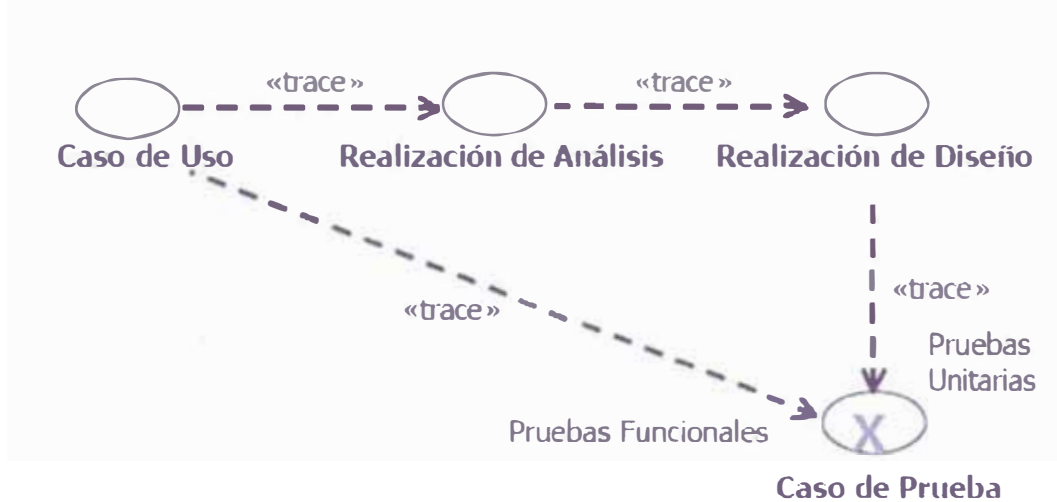


Figura 20 El RUP se encuentra dirigido por casos de uso.

2.2.4 ELEMENTOS DEL META-PROCESO

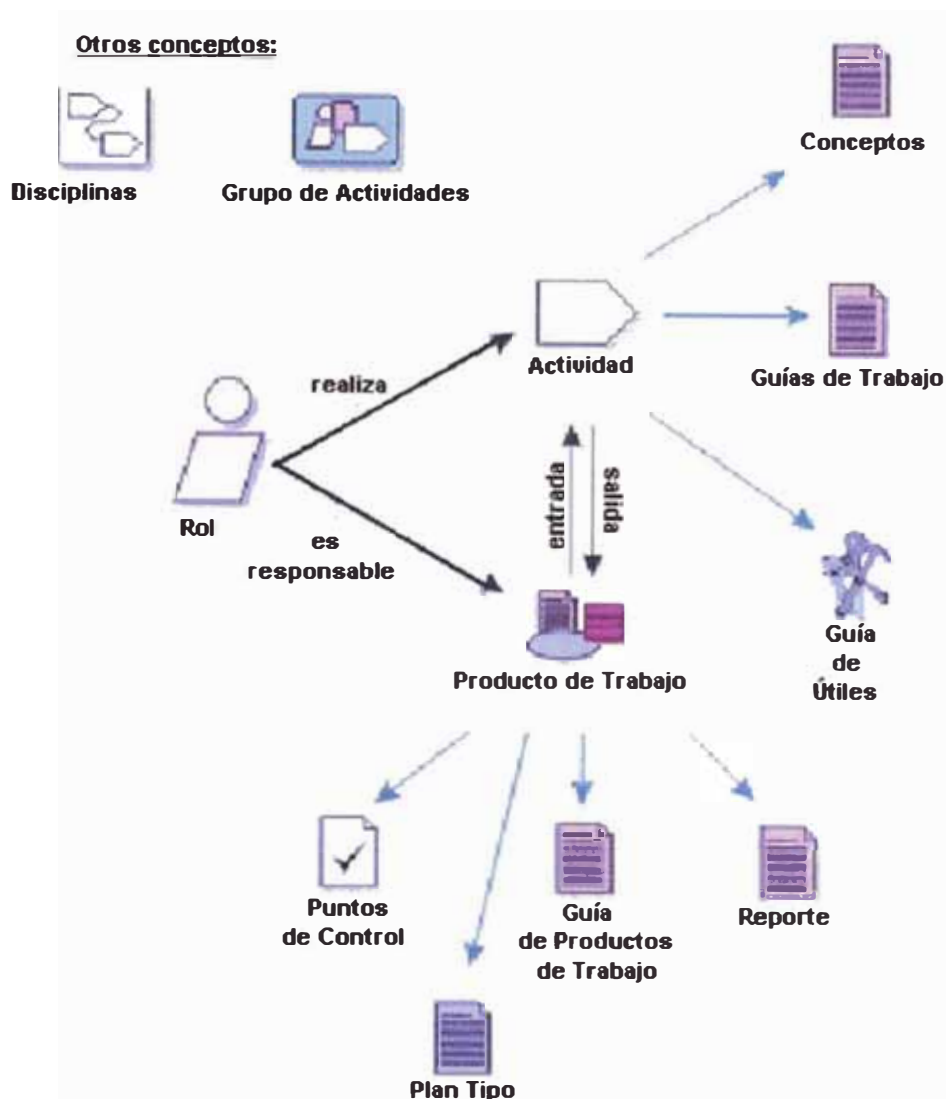


Figura 21 Elementos del RUP retomados para el RUPK

2.2.4.1 ROL

El concepto de rol es el elemento central del proceso. Un rol define el comportamiento y las responsabilidades del individuo o un conjunto de individuos que trabajan en equipo, dentro del contexto de la organización que desarrolla los sistemas informáticos.

Los roles no son obligatoriamente individuales; ellos describen cómo deben comportarse los individuos dentro del área de desarrollo y sus

responsabilidades. Los miembros de una organización de desarrollo pueden tener asignados muchos roles. La asignación de los roles recae sobre el jefe del proyecto a lo largo de la organización del proyecto y la planificación, permitiendo a diferentes personas representar muchos y variados roles, pudiendo un mismo rol, ser representado por varias personas a la vez.

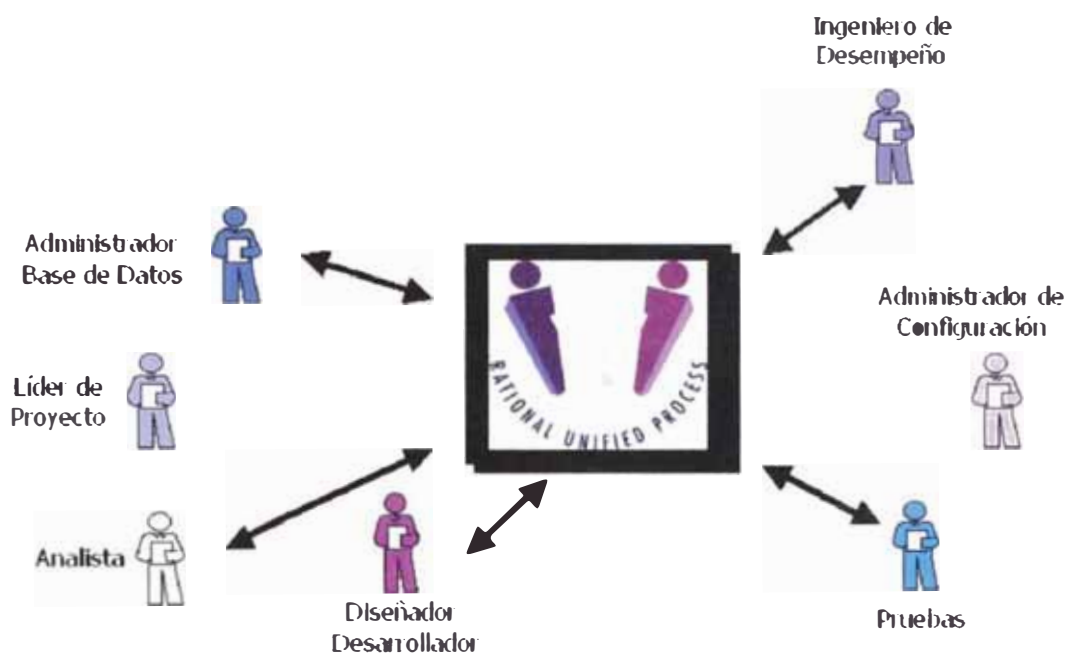


Figura 22 Los Roles en el RUP comparten la misma Base de conocimiento, proceso, Lenguaje de modelamiento y vista de desarrollo de software

2.2.4.2 ACTIVIDAD

Los roles realizan actividades. Una actividad es una unidad de trabajo realizada por un rol dentro del contexto del proyecto.

Una actividad tiene un objetivo claro, en general expresado en términos de creación o puesta en marcha del producto de trabajo, tal como un modelo, una clase, un plan, etc.. Cada actividad es ubicada bajo la

responsabilidad única de un rol. La granularidad de una actividad varía desde algunas horas hasta días. Una actividad es una unidad elemental del planeamiento.

Las actividades pueden ser repetidas en muchas ocasiones (por el mismo rol pero no necesariamente por el mismo individuo) sobre un mismo producto de trabajo, en particular una actividad es frecuentemente ejecutada para cada iteración.

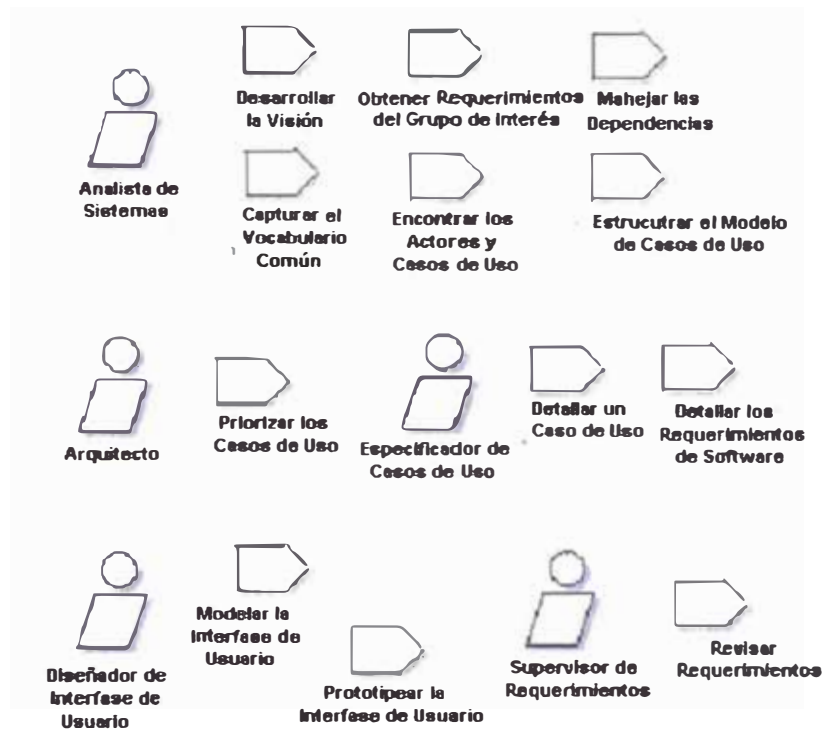


Figura 23 Roles y Actividades definidos para una Disciplina (SPEM)

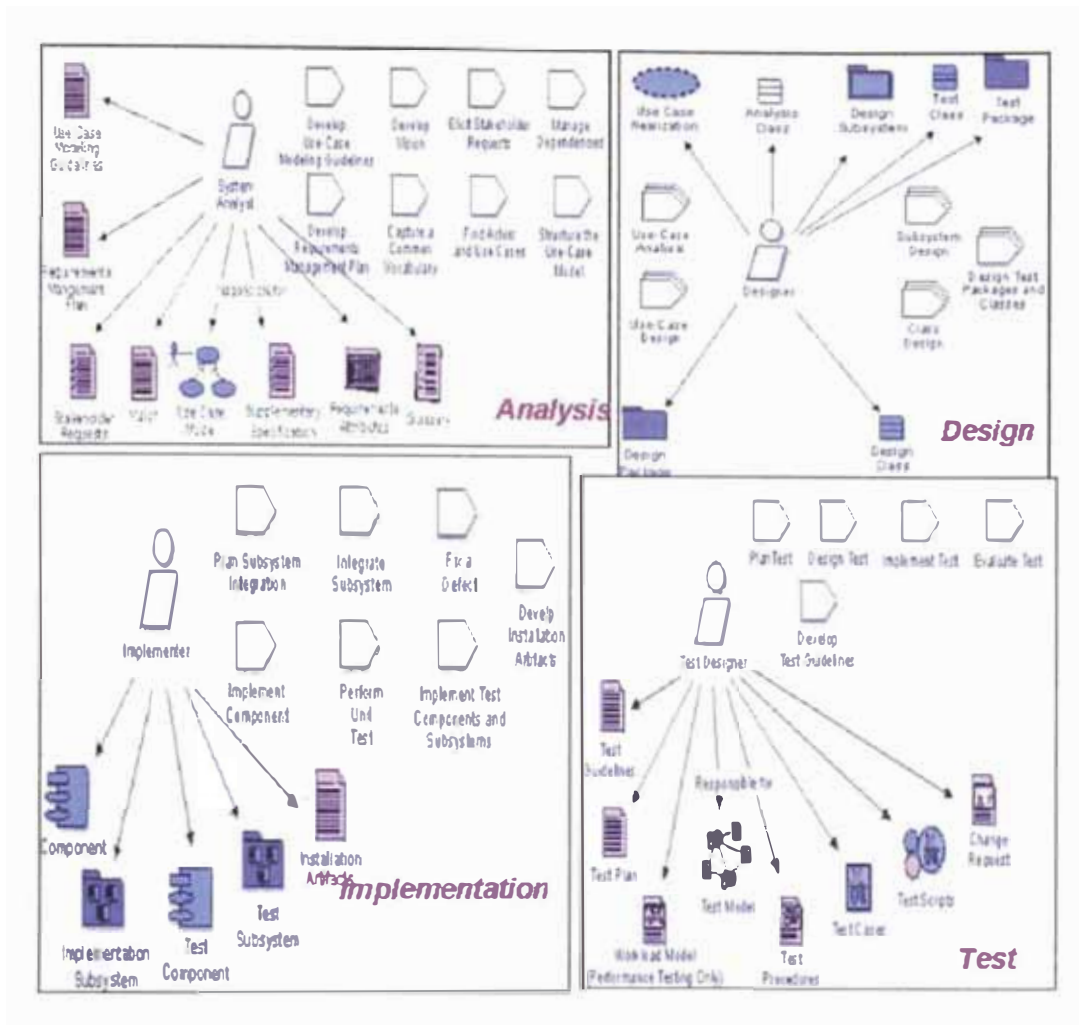


Figura 24 Roles (Workers) y Actividades definidos por el RUP

2.2.4.3 ETAPAS

Las actividades son descompuestas en muchas etapas. Existen tres categorías principales de etapas:

- Las etapas de reflexión: Cuando la persona que ocupa el rol comprende la naturaleza de su tarea, recolecta y examina los productos de trabajo de entrada, y formula la salida.
- Las etapas de realización. Cuando la persona que ocupa el rol crea ciertos productos de trabajo.
- Las etapas de revisión. Cuando los productos creados y modificados son inspeccionados de acuerdo a ciertos criterios.

Todas estas etapas no son forzosamente necesarias para una actividad, y pueden ser formuladas en forma de flujos.

2.2.4.4 GUÍAS DE TRABAJO.

Las etapas pueden ser asociadas a las guías de trabajo, que presentan las técnicas y consejos prácticos que son útiles para realizar la actividad.

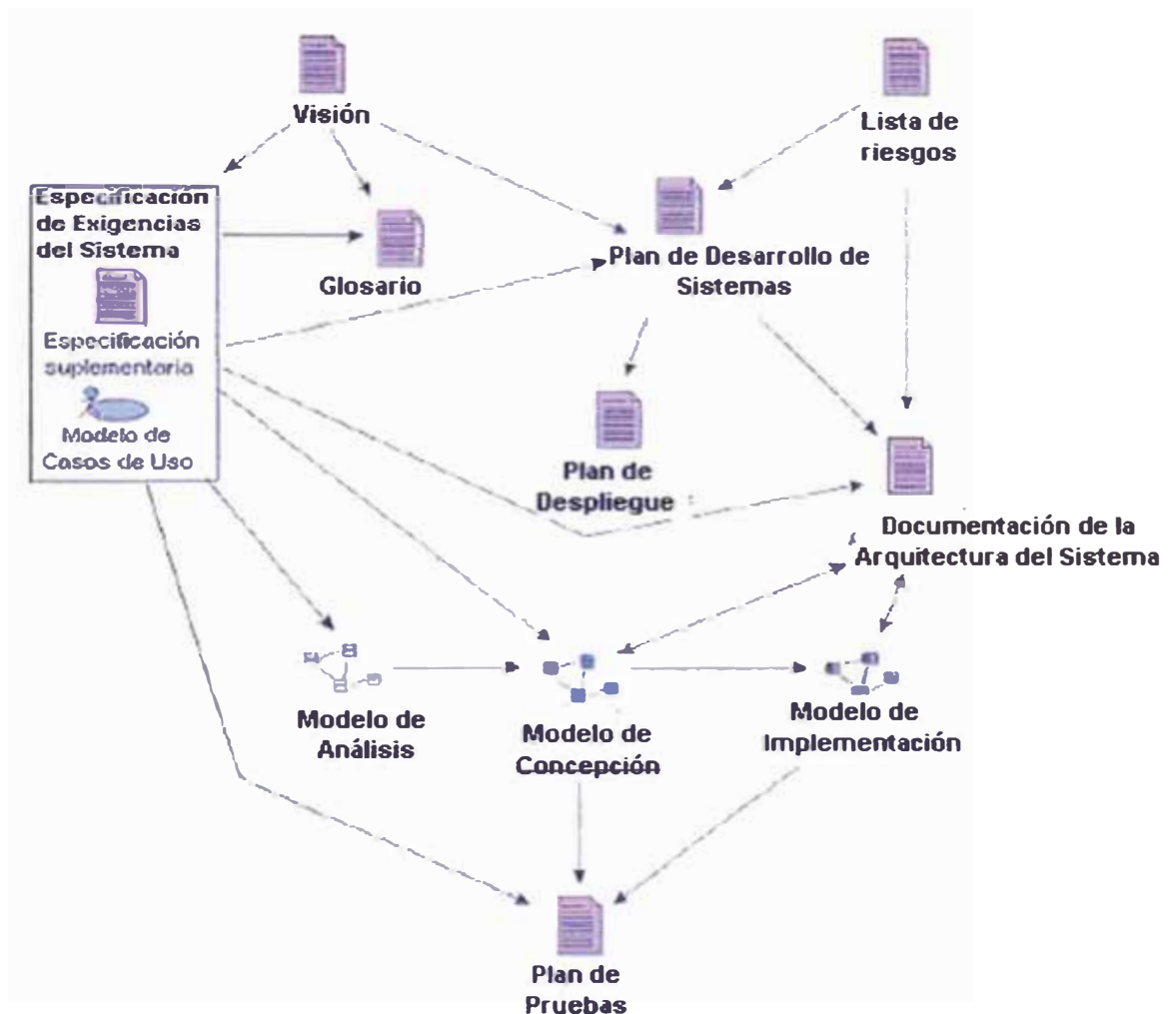


Figura 25 Los principales productos de trabajo del RUP
y el flujo de información entre ellos

2.2.4.5 PRODUCTOS DE TRABAJO.

Las actividades poseen productos de trabajo de entrada y productos de trabajo de salida. Un producto de trabajo es un elemento tangible producido o utilizado durante el proceso: los roles utilizan los productos de trabajo para realizar actividades, y los producen a lo largo de estas mismas actividades. Los productos de trabajo están bajo la responsabilidad de un solo rol, siguiendo la idea que cada elemento del proceso debe estar bajo la responsabilidad de una persona específica. Así como una sola persona es responsable del producto de trabajo, muchos otros pueden utilizarlo e incluso modificarlo, si se les da el permiso correspondiente.

El diagrama anterior muestra cómo la información circula a lo largo del proyecto tanto para el SPEM como para el RUPK, a través de los productos de trabajo; las flechas muestran cómo los cambios de un producto pueden propagarse en otros productos de trabajo. Para ser más claros, muchos productos de trabajo han sido omitidos (por ejemplo los productos que constituyen el modelo de concepción: las clases y paquetes no han sido representados).

Para simplificar la organización de productos de trabajo, se les organiza en conjuntos de productos de trabajo que tiendan a ser utilizados en el mismo momento o con el mismo objetivo.

Los productos de trabajo pueden tomar muchas formas:

- Un modelo, tal como el Modelo de Casos de Uso que contiene otros productos de trabajo.

- Un elemento del modelo, es decir, un elemento que se hace parte integrante del modelo, tal como un caso de uso o un paquete de casos de uso.
- Un documento, tal como la Visión o el Documento de la Arquitectura del Sistema.
- El código fuente y los ejecutables (tales como los Componentes).

Los artículos (productos entregables a clientes) sólo son un conjunto de productos de trabajo.

Los productos de trabajo no son obligatoriamente documentos. Los procesos tradicionales estimulan de manera excesiva la producción de documentos, y en particular los documentos en papel. El RUPK desaprueba la producción sistemática de documentos en papel. Lo más eficaz y pragmático para dirigir los productos de trabajo de un proyecto es mantener los productos de trabajo en el interior de los utilitarios utilizados para crearlos. Cuando sea necesario, se pueden generar los documentos de manera instantánea a partir de estos utilitarios, para el momento en el que sean necesarios. Se pueden considerar de la misma manera a las librerías de artículos para los clientes como parte integrante de los utilitarios, en lugar de dar lugar a un llenado de papel. Este método garantiza que los documentos estarán siempre actualizados y basados en la última versión que genere el proyecto, no generándose un esfuerzo de producción suplementaria. Sin embargo, existirán ciertos productos de trabajo que se encontrarán bajo la forma de documentos en papel, en el caso de entradas

externas al proyecto, o en el simple caso que constituye la descripción de un elemento.

2.2.4.6 PLAN TIPO

Los planes tipo son los modelos o los prototipos. Asociados con la descripción de un producto de trabajo, son utilizados para crearlos. Los planes tipo son referidos a los utilitarios usados por los productos.

Por ejemplo:

- Planes tipo Microsoft Word utilizados para los productos de trabajo de tipo documento, y algunos reportes.
- Planes tipo diagramas UML a través del utilitario de modelamiento UML ArgosUML del cual se extrae la información de los modelos de caso de uso generados en dicho formato.

Los planes tipo son parte de los elementos que se adaptan a una organización.

2.2.4.7 REPORTE

Los modelos y elementos de los modelos pueden ser asociados a los reportes. Un reporte extrae información de un modelo y sus elementos a partir de un utilitario. Por ejemplo, un reporte que presente un producto de trabajo o un conjunto de productos de trabajo para una revisión. A diferencia de los productos de trabajo regulares, los reportes no forman parte de las versiones. pueden ser regenerados en cualquier momento, regresando al producto de trabajo que los ha generado.

2.2.4.8 GUÍAS DEL PRODUCTO Y PUNTO DE CONTROL

Con frecuencia se asocian los productos de trabajo a las guías y puntos de control que presentan información sobre la manera de crearlos o editarlos, evaluarlos y utilizarlos. Una buena parte de la sustancia del proceso está contenida en las guías de los productos de trabajo; las descripciones de las actividades capturan la esencia de aquello que se está realizando, mientras que las guías capturan la esencia del arte de ejecutar el trabajo. Los puntos de control dan una referencia rápida para evaluar la calidad de un producto.

Las guías y los puntos de control son útiles bajo ciertos contextos: para ayudar a decidir sobre qué hacer, para ayudar a hacerlo y para ayudar a verificar el resultado.

2.2.4.9 FLUJO DE TRABAJO

Una sencilla enumeración de todos los roles, actividades y productos de trabajo no constituye un proceso, es necesario describir las secuencias de actividades usuales para producir los resultados de calidad, y mostrar las interacciones entre los roles. Un flujo de trabajo es una secuencia de actividades que produce un resultado observable.

El flujo de trabajo de una disciplina es descrito en un Diagrama de Actividades UML. De hecho, este diagrama presenta los grupos de actividades, presentados más adelante.

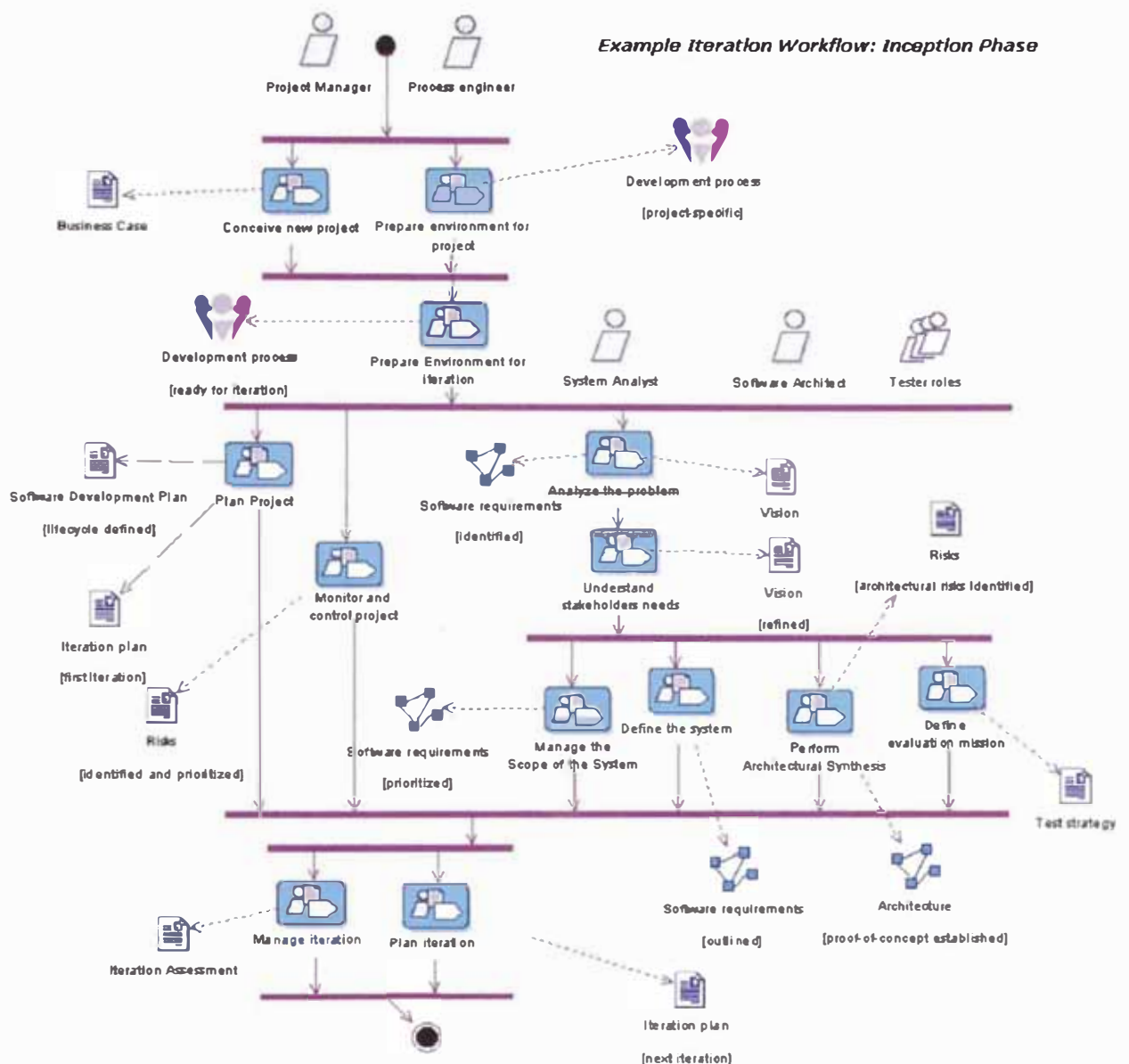


Figura 26 Flujo de Trabajo clásico (RUP)

2.2.4.10 DISCIPLINA

Una disciplina es una composición de procesos, organizada de acuerdo a una perspectiva característica de la Ingeniería de Sistemas. Los nombres de las disciplinas representan los términos habitualmente utilizados en los ciclos de vida en cascada o en V. El flujo de trabajo de una disciplina es la secuencia parcialmente ordenada de actividades realizadas para atender un objetivo común. La naturaleza semiordenada de los flujos de

trabajo de la disciplina se presenta por que no se puede representar los matices intelectuales en la vida cotidiana de los proyectos. Sin embargo, es indudable, el mérito que tiene de hacer comprender el proceso dividiéndolo en pequeñas zonas de interés.

Cada zona de interés o disciplina está asociada a uno o varios modelos, que están compuestos de productos de trabajo asociados. Los productos de trabajo más importantes son los modelos que cada disciplina produce: modelo de casos de uso, modelo de concepción, modelo de implementación y modelo de pruebas.

El RUP presenta 9 disciplinas siendo estas las siguientes:

- Modelamiento del Negocio.
- Análisis y Diseño.
- Pruebas Funcionales.
- Administración de Configuración y Cambios.
- Gestión del Proyecto.
- Entorno.
- Requerimientos.
- Desarrollo.
- Implementación.

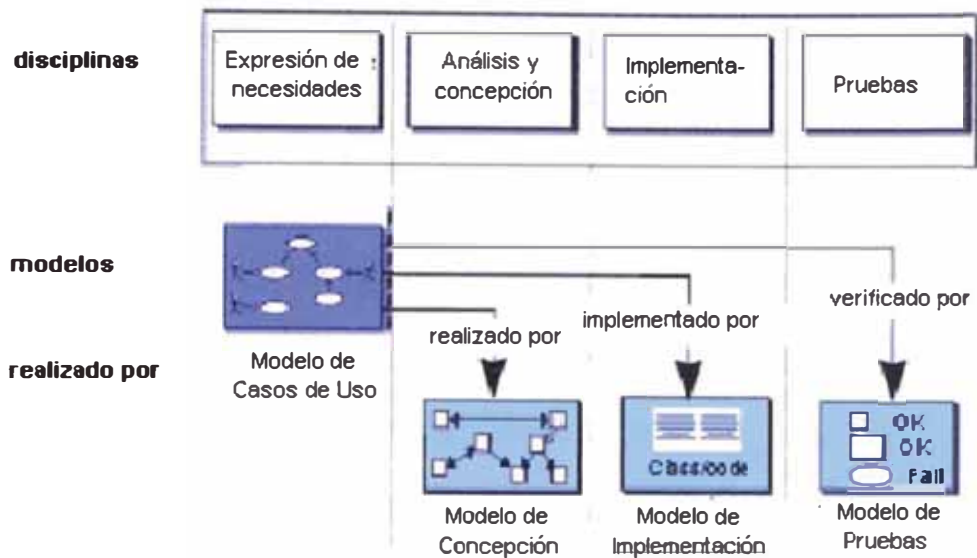


Figura 27 Cada disciplina está asociada a un conjunto particular de modelos

Para cada disciplina, una vista general de actividades es igualmente presentada. La vista general de las actividades muestra todas las actividades de la disciplina así como el rol que realiza estas actividades. La vista general de los productos de trabajo muestra todos los productos de trabajo y los roles implicados en la disciplina.

Es necesario recalcar que las disciplinas no son totalmente independientes las unas de las otras. Existen algunos productos de trabajo que son utilizados por varias disciplinas (por ejemplo la Visión entre la expresión de necesidades y en la gestión del proyecto).

2.2.4.11 GRUPO DE ACTIVIDADES

El flujo de trabajo de una disciplina no muestra directamente las actividades, en cambio los grupos de actividades muestran los reagrupamientos de actividades que son frecuentemente realizados juntos.

Cada grupo de trabajo es descrito en una tabla que muestra los roles implicados, los productos de trabajo de entrada y salida, y las actividades realizadas. Las razones por las que existen los grupos de actividades son las siguientes:

- Las actividades del grupo no se realizan ni en secuencia ni de manera simultánea. Típicamente el trabajo se hace en paralelo sobre varias actividades, con los productos de trabajo de entrada.
- Es demasiado complejo mostrar los documentos de entrada y salida para todas las actividades de una disciplina en un solo diagrama. El grupo de actividades permite mostrar a la vez las actividades y los productos de trabajo, para una parte de la disciplina en un momento determinado.

2.2.4.12 GUÍA DE UTILITARIOS

Las actividades, las etapas y las guías asociadas proveen de indicaciones a los usuarios del proceso. Por tal motivo, las guías de utilitarios permiten guiar a los usuarios en la manera de un utilitario lógico especificado.

Las guías de utilitarios permiten dejar el resto del proceso independiente de los utilitarios. Sólo la guía de utilitario encapsula las dependencias del proceso sobre los utilitarios.

CAPÍTULO III : META-PROCESO PROPUESTO

3.1 DEFINICIÓN Y REALIZACIÓN DEL META-PROCESO RUPK

Las actividades realizadas y los productos elaborados durante el proyecto son reagrupados por disciplinas, de manera análoga al RUP. Es necesario recordar que el desarrollo del RUPK es iterativo y que las disciplinas presentadas no se desarrollan de manera secuencial.

3.1.1 ENTORNO

Esta disciplina comprende la formación de un dominio, así como el aprendizaje de distintos utilitarios. Se toma como punto de evaluación a 2.25 hombre / mes, con una gran parte dada a la asimilación de términos y al aprendizaje en el manejo de utilitarios. Para un nuevo desarrollo del proceso, este tiempo será considerablemente disminuido.

3.1.2 DEFINICIÓN DEL META-PROCESO

La evaluación del proceso actual ha permitido determinar sus características, puntos débiles para identificar los aspectos sobre los cuales se debe poner mayor atención dentro del meta-proceso a realizar. (adoptar un ciclo iterativo, mejorar la administración de riesgos, profundizar en la expresión de exigencias, poner mayor atención en la arquitectura) y las reglas del negocio que deberán ser tomadas en consideración.

La visión presenta al meta-proceso a realizar con sus principales características y ventajas. Como el meta-proceso es derivado del RUP, la visión pone en evidencia las elecciones realizadas para la simplificación de los elementos esenciales del meta-proceso: disciplinas, roles y productos de trabajo. La visión toma también en consideración las particularidades de la organización. Así podemos mencionar a las alternativas consideradas y presentadas a continuación, que tienen una gran influencia sobre el meta-proceso final.

3.1.3 EVALUACIÓN DEL PROCESO ACTUAL

Para poder decidir sobre la implementación de un proceso iterativo se debe estudiar la situación actual en la que se encuentra la organización sobre la cual se va a hacer el estudio. Como anteriormente se indicó, el RENIEC es una entidad gubernamental encargada de la identificación y registro civil del ciudadano peruano. Los proyectos son responsabilidad de la Sub Gerencia de Planificación y Desarrollo de Sistemas Informáticos.

Un proyecto es recogido de los distintos requerimientos que hacen las gerencias del RENIEC (incluyéndose a la Gerencia de Informática). El Sub Gerente de Planificación y Desarrollo de Sistemas Informáticos se encarga de definir los proyectos y designar los jefes de proyecto, considerando un tiempo aproximado para la realización de los proyectos. A partir de la disponibilidad de los analistas y programadores se conforman los grupos de trabajo. El jefe de proyecto y el grupo de trabajo definen las etapas del proyecto y plazos para cada una de ellas. Las etapas se definen por sistema.

Los analistas funcionales empiezan con el análisis detallado de los requerimientos de manera conjunta con los usuarios líderes para luego proceder a definir los principales flujos de datos. Con los principales flujos de datos definidos se puede pasar al diseño de la base de datos, la cual será revisada por los analistas de base de datos. Teniendo diseñada la base de datos, se termina por definir los flujos de datos restantes y el refinamiento de la base de datos. El programador procede a generar código fuente y a diseñar las interfaces siguiendo la lógica que presentan los flujos de datos. Los documentos de análisis y diseño se van almacenando para posteriores revisiones, así como el código fuente del sistema a realizar.

Los analistas funcionales intervienen en las pruebas, terminando por desarrollar el manual de usuario y en conjunto con los desarrolladores hacen el manual del sistema. Luego de haberse dado la conformidad a las pruebas funcionales se realiza el pase de los aplicativos que conforman el sistema del entorno de desarrollo al entorno de producción. Estando en producción, son los propios usuarios finales los que realizan las pruebas para dar sus últimas observaciones. Al término de este pequeño ciclo, se procede a la completa documentación y a la puesta en marcha en producción con los posteriores trabajos de monitoreo y mantenimiento.

La arquitectura en ciertos aspectos ya se encuentra bien definida. Las entidades gubernamentales para adquirir software y hardware deben incluir sus compras en sus presupuestos anuales que son auditados por el Congreso de la República. Toda adquisición de acuerdo a la cantidad de dinero que signifique la transacción, debe ser sometida a adjudicación

directa, licitación o concurso público. Es muy difícil pensar en un cambio de plataforma inmediato y el estudio de ella requiere un profundo estudio para abarcar el mediano y largo plazo. Actualmente se está trabajando con Oracle como motor de la Base de Datos, servidores AIX de múltiples procesadores y arquitectura RISC RS6000 7017 S80. El sistema operativo cliente es el MS Windows 2000. Se han establecido tres lenguajes de programación para el desarrollo de aplicativos: Java, Developer 2000 y Visual Basic. El primero y el segundo son para el desarrollo de nuevos sistemas y el tercero es para dar mantenimiento a sistemas existentes realizados en ese lenguaje.

Para cada etapa el Jefe de Proyecto emite informes sobre los avances que se vienen dando al Sub Gerente de Planificación y Desarrollo de Sistemas Informáticos.

El desarrollo se basa en la utilización de patrones por lo que el código fuente puede ser entendido por cualquiera de los programadores que intervienen en el proyecto dejando de lado la idea de personas imprescindibles.

Como se puede observar el desarrollo es en cascada pues depende cada parte del término de la parte que le antecede. Para los proyectos esto genera muchos tiempos muertos, lo cual se agrava al considerar proyectos de mediana envergadura.

Todos los miembros de un equipo de trabajo pertenecen a la vez a otros grupos de trabajo para minimizar los tiempos muertos de desarrollo y optimizar los tiempos en función a los recursos disponibles. Cada miembro del equipo conoce perfectamente el rol que desempeña en el desarrollo de

un proyecto. La comunicación entre los miembros de un grupo de trabajo es bastante fluida facilitando las labores de administración y supervisión.

El principal problema del actual proceso viene dado con los retrasos para cumplir con los plazos establecidos y el considerable tiempo adicional que toman los proyectos. En algunos casos los proyectos se enfrascan en grandes indefiniciones de análisis por no haber una debida comunicación con los usuarios finales. Otro problema que en ciertos casos se presenta es la creación de un sistema que no responde a las necesidades reales de los usuarios finales, ya sea de manera parcial o total. La situación puede ser peor si el usuario realmente no sabe lo que necesita para mejorar la productividad en su trabajo no sabiendo por ende qué es lo que necesita de un sistema.

Cada sistema genera una gran cantidad de documentos que terminan por ser obsoletos a efectos del mantenimiento constante que se le hacen a los aplicativos. Es sumamente tedioso leer toda la documentación generada, aún cuando los documentos mantienen una misma estructura para facilitar su lectura.

En suma, se pueden encontrar las siguientes disciplinas:

- Gestión del Proyecto.
- Requerimientos.
- Análisis y Diseño.
- Desarrollo e implementación.

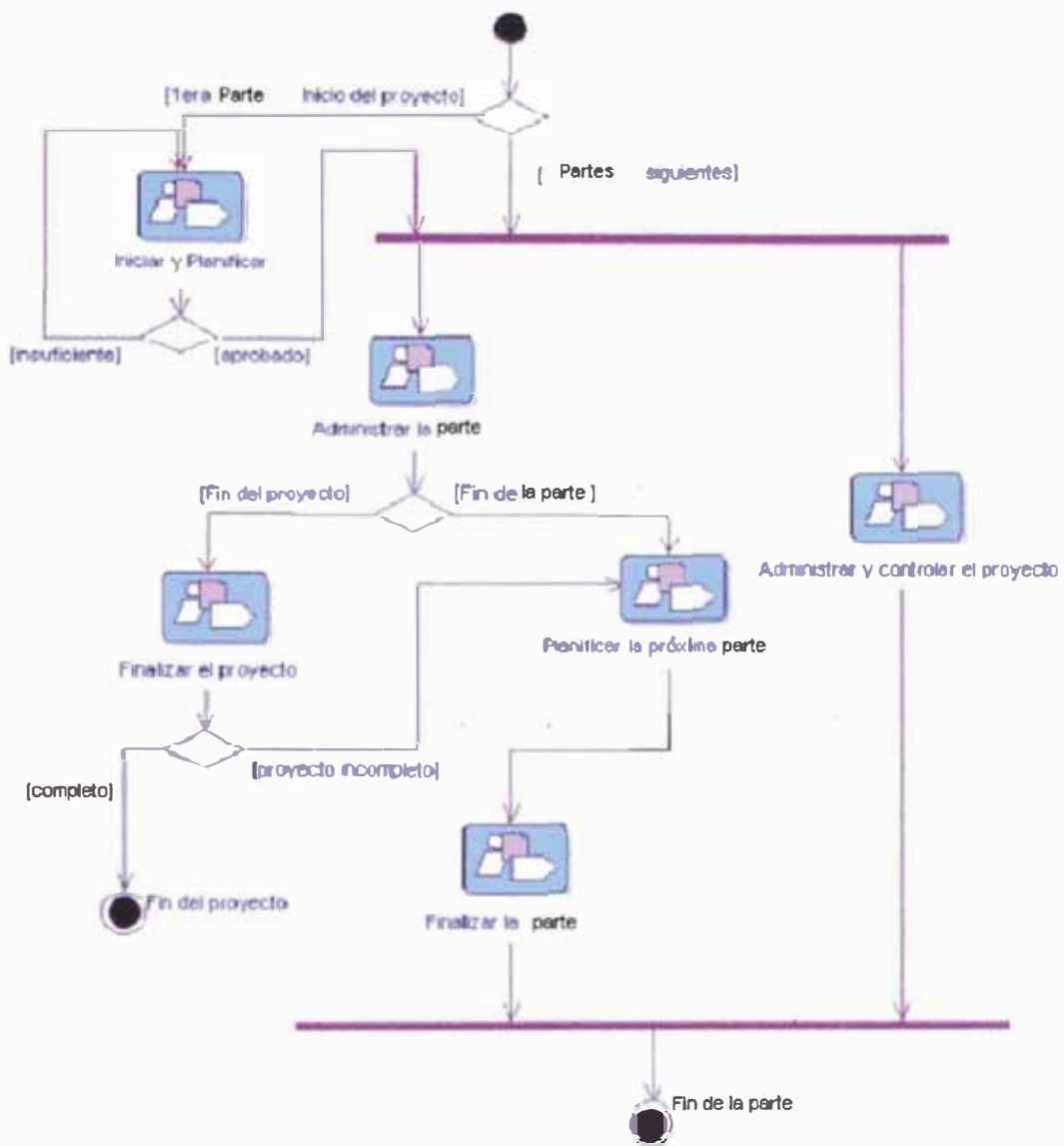


Figura 28 Diagrama de Actividad de la Disciplina Gestión del Proyecto del proceso actual.

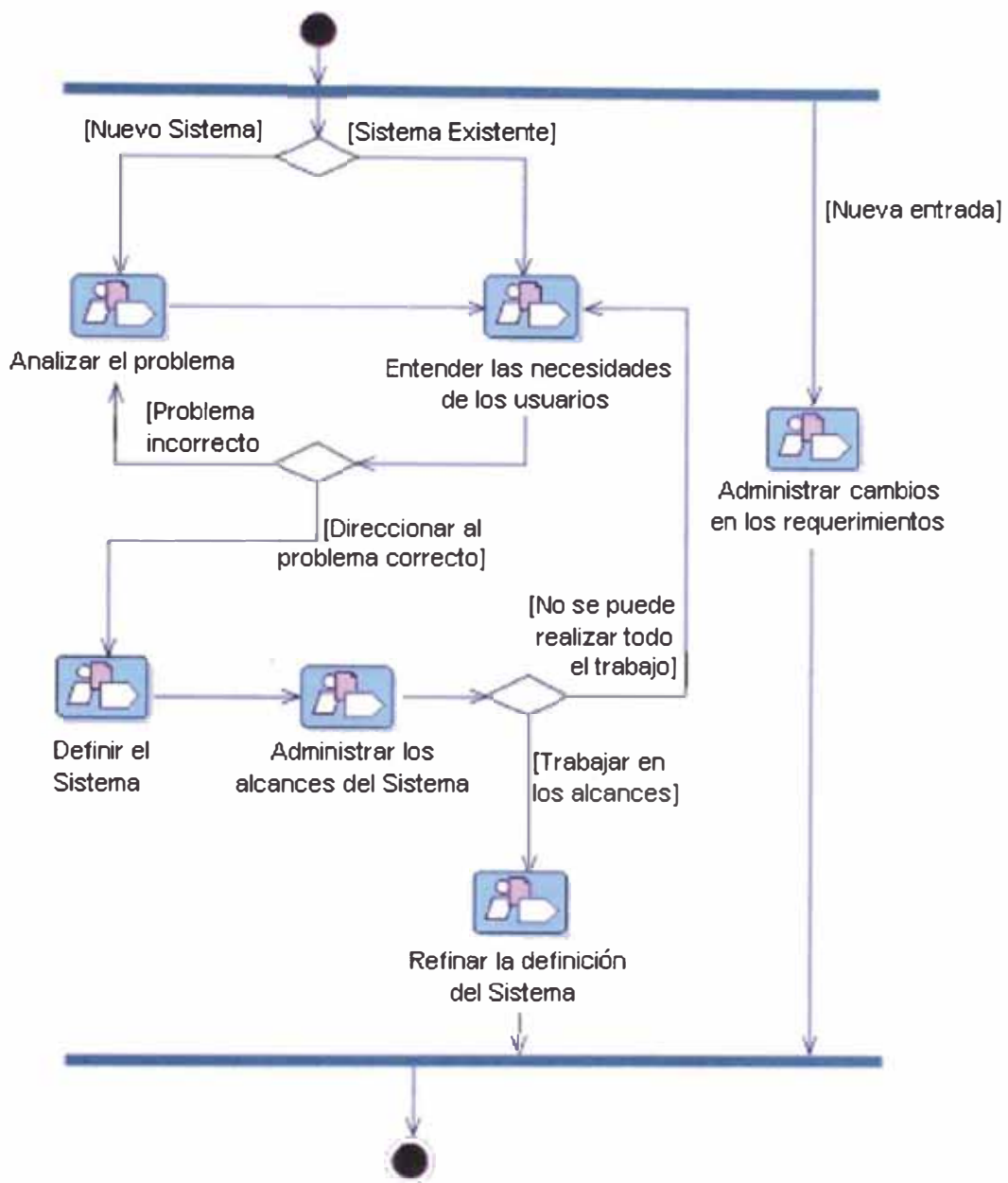


Figura 29 Diagrama de Actividad de la Disciplina Requerimientos del proceso actual.

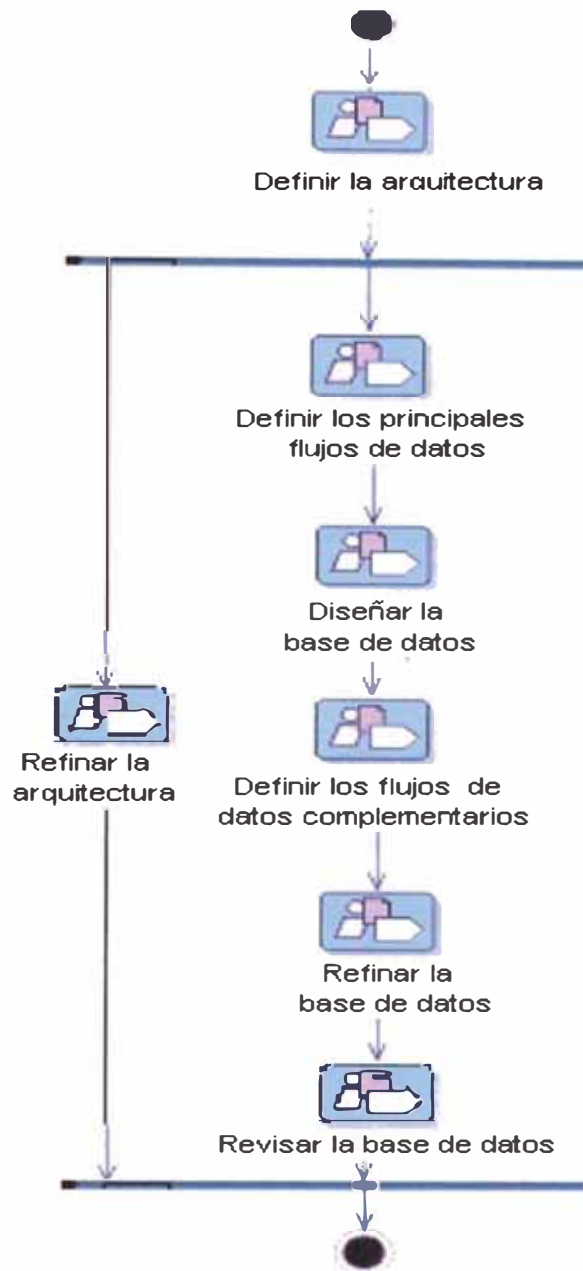


Figura 30 Diagrama de Actividad de la Disciplina Análisis y Diseño del proceso actual.

3.2 META-PROCESO PROPUESTO

Las actividades realizadas pueden ser ubicadas dentro del trabajo de modelamiento.

3.2.1 MODELO

El modelamiento del meta-proceso ha tomado menos de un hombre/mes (trabajando 40 horas a la semana que consta de 5 días útiles). El modelo del RUPK define el nuevo meta-proceso reutilizando en lo posible el modelo del RUP. Los productos del RUPK son un subconjunto de aquellos del RUP; es por ello posible establecer una dependencia entre los productos que nos interesan sin crearlos en el modelo. Se han suprimido una gran cantidad de roles y actividades, fusionándose algunas otras.

La herencia múltiple no está permitida en los utilitarios del Rational Rose, lo cual implica que este trabajo sólo procederá con la herencia simple en uno de los roles.

Los roles conservados en el RUPK son todos heredados del RUP y redefinidos en el modelo RUPK. Las actividades son modeladas como operaciones sobre los roles, siendo posible:

- Heredarlos si convienen las entradas y salidas del RUP.
- Redefinirlos si deben cambiar las entradas y salidas en el RUPK.
- Eliminarlos si la actividad no es necesaria en el RUPK.

Para el caso del RENIEC, dentro del personal dedicado a los distintos proyectos de informática se cuenta con personas totalmente dedicadas a las labores de análisis y documentación, siendo además el número total de

trabajadores reducido por proyecto (entre 6 y 8 personas). Los proyectos no son de gran envergadura tomándose plazos medianos en tiempo de realización. Las cantidades reducidas de roles y productos de trabajo se consideran suficientes para poder llevar a cabo la realización de los distintos proyectos informáticos.

La cabeza del equipo es el Jefe de Proyecto, tal como actualmente se viene trabajando en el RENIEC para los diferentes proyectos y que recoge las mismas características heredadas del RUP. Es preferible recoger las similitudes del actual proceso del RENIEC con el meta-proceso RUP para que el cambio en la manera de trabajar no sea tan brusco.

Al contarse con un área totalmente dedicada a la administración de la Base de Datos, se puede tomar a los Analistas de Base de Datos para que sean los Diseñadores de la Base de Datos para los proyectos tal como se indica en el RUP dejando de ser supervisores del trabajo de diseño para formar parte activa en la creación de la Base de Datos..

La nueva forma de trabajo que se propone se basa principalmente en el trabajo con componentes, como agentes que llevan las reglas del negocio. Para este trabajo creemos que es necesario crear un nuevo rol el cual va a ser responsable de crear y desarrollar los distintos componentes, su integración e interrelación, las jerarquías y dependencias entre ellos, los permisos para accederlos y ejecutarlos y el repositorio de componentes. El nuevo rol encargado de esta importante labor es el Diseñador de Componentes.

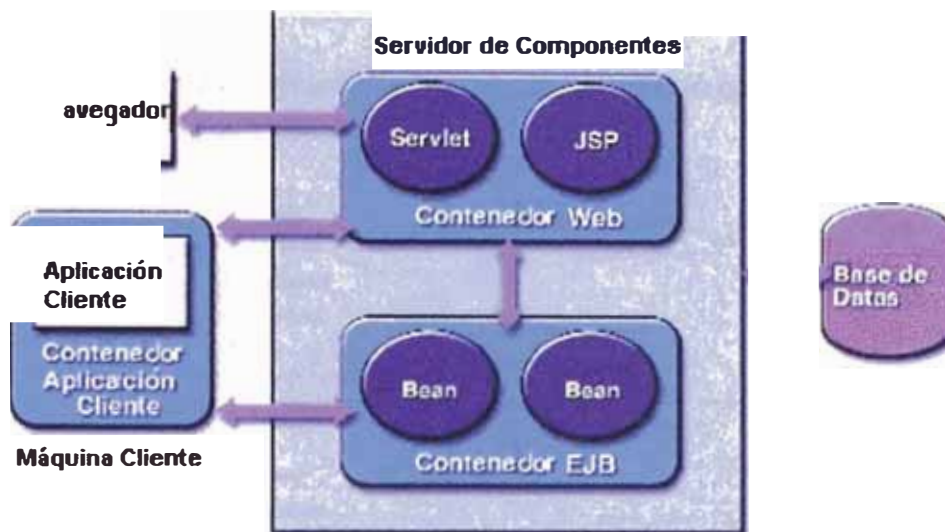


Figura 31 Ejemplo de trabajo basado en componentes.

Pero los componentes al igual que las tablas en una Base de Datos, no son necesariamente de uso para un solo aplicativo y por ello se requiere de todo un trabajo de diseño de componentes para usos genéricos y específicos. Es preferible contar con un área de diseñadores de componentes aunque ello para el caso del RENIEC sea poco factible por la cantidad de restricciones que se encuentran al intentar crear una nueva área. El desarrollo de componentes es una tarea relativamente compleja por lo que necesita de personal altamente calificado o de mayor experiencia entre los desarrolladores. El arquitecto se debe encargar de encaminar el trabajo de los desarrolladores de componentes haciendo conversar al análisis y diseño con la implementación. Los componentes para los desarrolladores de interfaces deben simplemente especificar los parámetros de entrada y de salida.

Debido a que los casos de uso son utilizados para recoger los requerimientos de los usuarios y que son además el punto de partida del

análisis, se ha definido al analista de sistemas del RUPK como la fusión del analista de sistemas y el especificador de casos de uso del RUP.

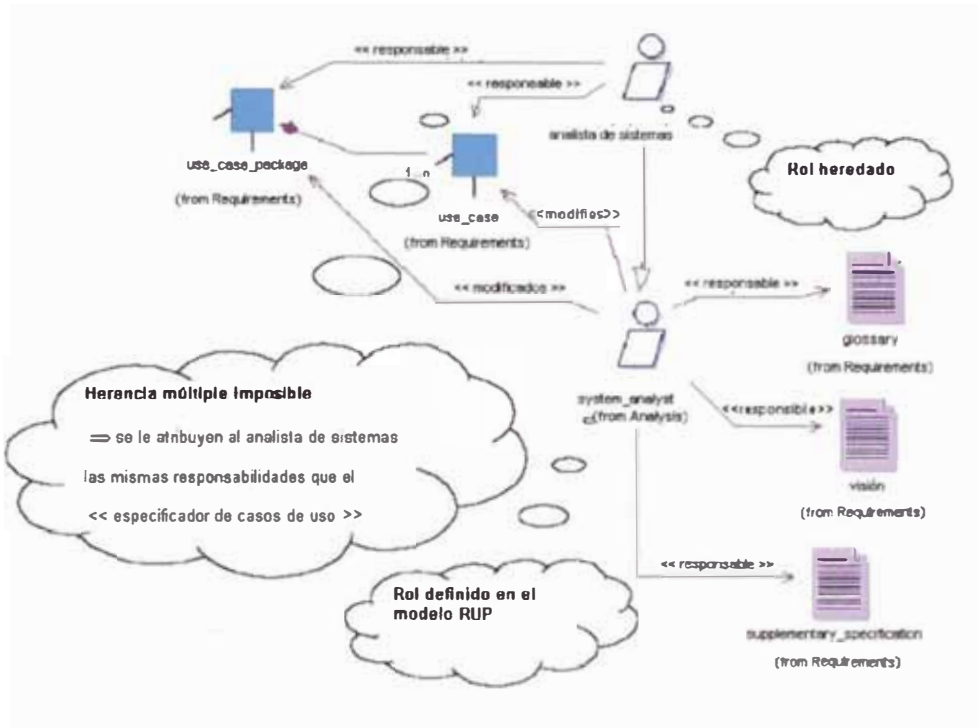


Figura 32 Ilustración de la herencia y de la fusión de roles

El analista de sistemas del RUPK hereda del “analista de sistemas del RUP”: dándole como atribuciones al primero las responsabilidades del segundo, que deseamos heredar: como el “especificador de casos de uso”, nosotros hemos hecho responsable al analista de sistemas de los productos de trabajo “casos de uso” y “paquetes de casos de uso”.

El supervisor RUPK hereda del “supervisor de proyecto RUP” las actividades de revisión de la aceptación del proyecto y los objetivos del ciclo de vida del proyecto, conteniendo además las actividades de revisión y control pertenecientes a distintos roles en el RUP. El supervisor de los proyectos vendría a ser el Sub Gerente de Planificación y de Desarrollo de Sistemas Informáticos. Como anteriormente se explicó, esto es lo que

actualmente realiza el Su Gerente de esta área siendo al final la persona que entrega resultados al Gerente de Informática por cada proyecto.



Figura 33 Ilustración de la herencia del rol Supervisor del RUPK

El Supervisor puede optimizar su trabajo si cuenta con la ayuda de un Ingeniero de Métodos para clarificar y disgregar el trabajo de la institución en procesos. El perfil puede ser cubierto por los Analistas Funcionales que cubrirían el rol de Ingeniero de Procesos.

El área de Innovación tecnológica dedicada abiertamente a la investigación de nuevas tecnologías y herramientas puede apoyar al trabajo con expertos en el uso de los diversos utilitarios para su uso en un proyecto orientando a los desarrolladores.

Podemos distinguir en el nuevo meta-proceso tres grupos definidos para los roles:

- **Analistas.**
 - a. Analista de Sistemas.
 - b. Analista Funcional.
- **Desarrolladores.**
 - c. Arquitecto.
 - d. Diseñador de componentes.

- e. Diseñador de la Base de Datos.
- **Administradores.**
 - f. Jefe del Proyecto.
 - g. Supervisor del Proyecto.
 - h. Ingeniero de Procesos.
 - i. Especialista en Utilitarios.

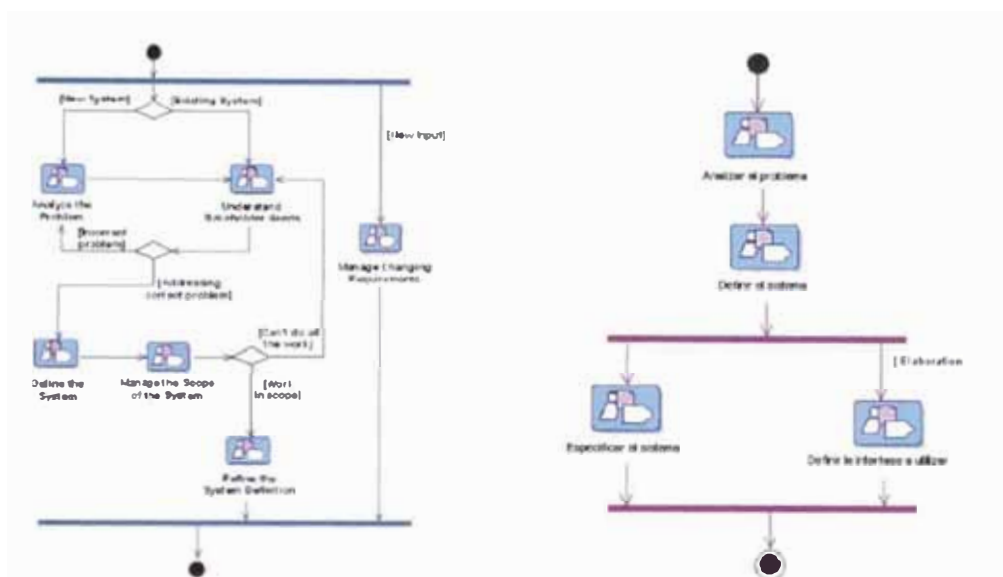


Figura 34 A la izquierda se muestra un flujo de trabajo del RUP, y a la derecha se muestra su equivalente en el RUPK (Requerimientos)

Con respecto al RUP, los elementos modificados radicalmente son los flujos de trabajo de las disciplinas. En efecto, los flujos de trabajo que describen la especificación del RUPK, son una adaptación a la organización y son menos cargados con respecto al RUP.

Una consecuencia de la supresión de algunas disciplinas del modelamiento y la gestión de configuración y de sus diferentes flujos de trabajo es el gran trabajo a realizar en las actividades creadas o redefinidas para que los productos de trabajo de entrada y de salida sean coherentes.

La relación de productos de trabajo para el RUPK es la siguiente:

- **Para Requerimientos**
 - Guía de Casos de Uso
 - Actor
 - Caso de Uso
 - Modelo de Casos de Uso
 - Paquete de Casos de Uso
 - Glosario
 - Modelo del Negocio
 - Especificaciones suplementarias
 - Prototipo de las Interfaces de Usuario
 - Visión

- **Para el Análisis y Diseño**
 - Clases del Análisis
 - Modelo de Análisis
 - Modelo de Despliegue
 - Clases de Concepción
 - Paquete de Concepción
 - Subsistema de Concepción
 - Interfaz
 - Realización de Casos de Uso
 - Modelo de Datos

- Arquitectura del Sistema

- **Para la Gestión del Proyecto**
 - Lista de problemas
 - Evaluación de la iteración
 - Plan de iteración
 - Lista de riesgos
 - Plan de desarrollo de Sistemas

- **Para el Entorno**
 - Guía de programación
 - Planes tipo Específicos del proyecto
 - Utilitarios
 - Plan del Ciclo de Vida

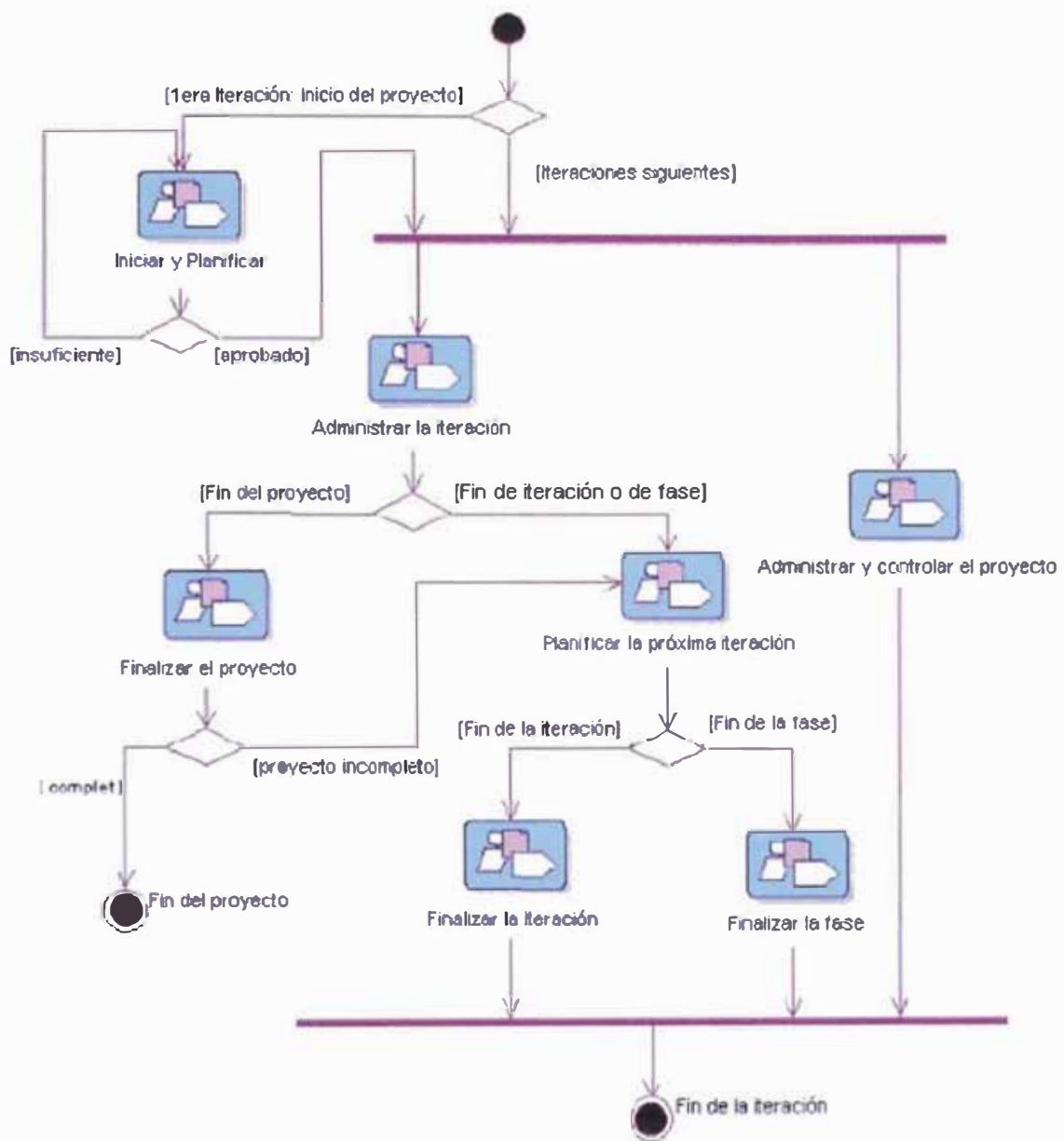


Figura 35 Diagrama de Actividad de la Disciplina Gestión del Proyecto del RUPK

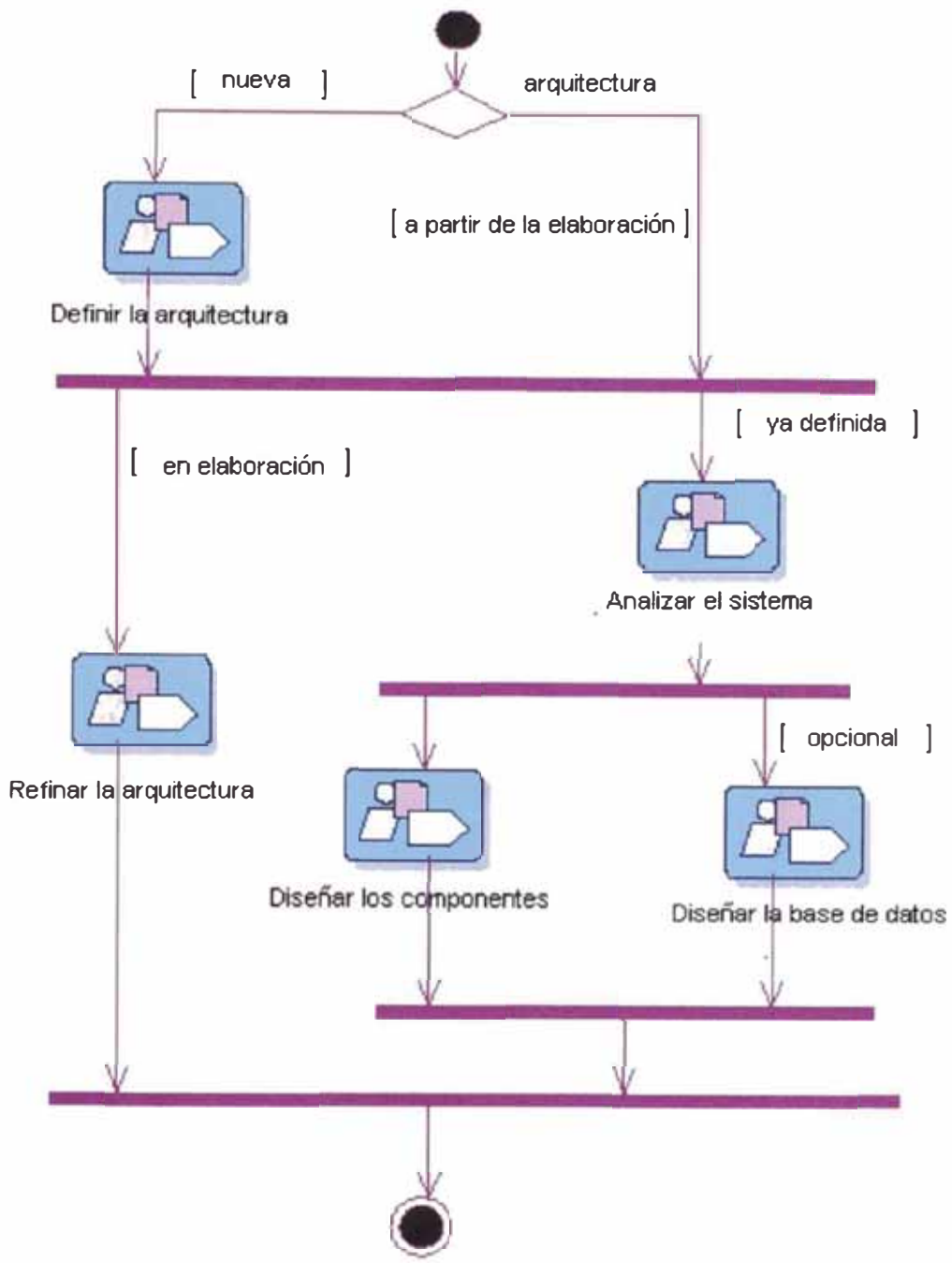
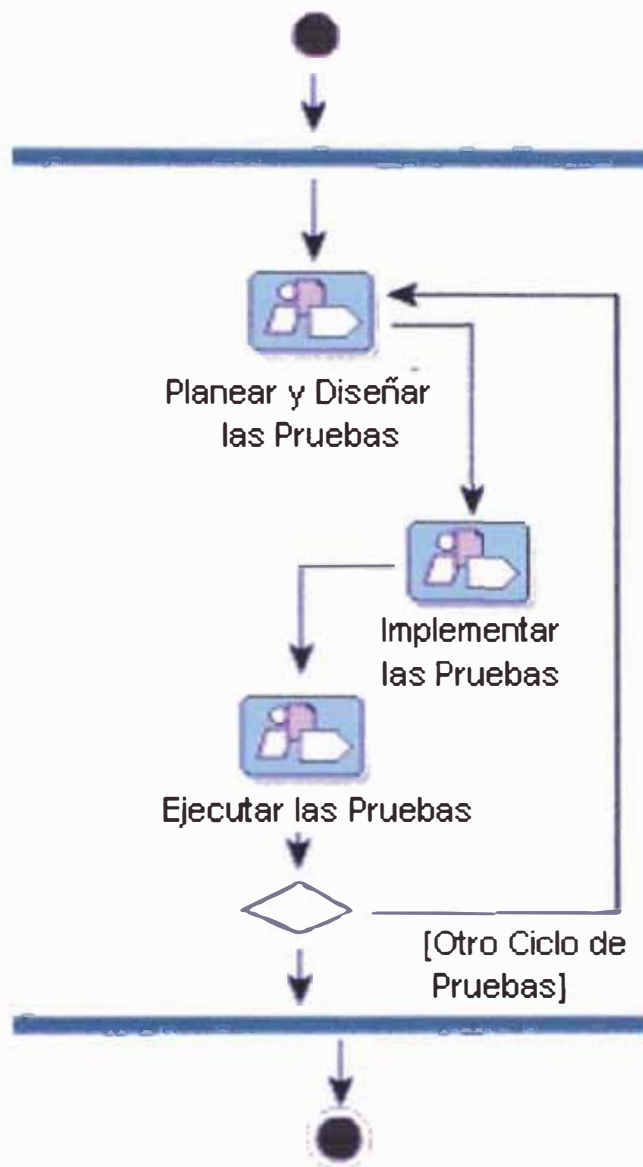


Figura 37 Diagrama de Actividad de la Disciplina Análisis y Diseño del RUPK



**Figura 38 Diagrama de Actividad de la Disciplina
Pruebas Funcionales del RUPK**

CAPÍTULO IV : APLICACIÓN DEL META-PROCESO



SISTEMA DE REGISTROS CIVILES



4.1 INTRODUCCION

El registro de los Hechos Civiles es actualmente un ámbito de la competencia de RENIEC que aún no ha sido abordado con las herramientas y tecnología adecuadas para su óptimo funcionamiento e integración en servicio de todos los ciudadanos peruanos.

En la actualidad, la Ley Orgánica del Registro Nacional de Identificación y Estado Civil. (Ley Nro. 26497) establece una gran cantidad de hechos civiles inscribibles (alrededor de 20) de los cuales sólo se inscriben efectivamente los Nacimientos, Matrimonios y Defunciones.

El registro de un hecho civil se lleva a cabo por la oficina registral de cada Municipalidad del país. Cada una de ellas ha definido sus propios procesos de registro y archivo de la información obtenida, ocasionando heterogeneidad y escasa integración, incluso algunas municipalidades han optado por automatizar algunas partes del proceso de registro empleando soluciones informáticas externas al RENIEC. A partir del año 1997 se logró uniformizar el formato sobre el cual se registran los Nacimientos y Matrimonios, sin embargo este esfuerzo es aun insuficiente para lograr el verdadero objetivo.

El Sistema de Registros Civiles, plantea la estandarización y automatización de los procedimientos de registro de los principales Hechos Civiles con el fin de obtener una Base de Datos única y centralizada, Integridad y seguridad en la información y finalmente eficiencia y uniformidad en la captura de la información.

El Registro de los Hechos Civiles consta de varios procesos como son:

- **Nacimientos:**
 - La Inscripción del Nacimiento.
 - El reconocimiento de Hijos.
 - La paternidad o maternidad declarada por resolución Judicial firme.
 - Las Adopciones.
 - La renuncia a la Adopción.
 - Las Rectificaciones.

- **Matrimonios**
 - La inscripción del Matrimonio
 - La declaración de nulidad
 - El divorcio
 - La Separación de cuerpos
 - La Reconciliación
 - Los acuerdos de separación de patrimonios.

- **Defunciones**
 - La inscripción de la defunción
 - La muerte presunta declarada por Resolución Judicial firme.
 - El reconocimiento de existencia declarada por Resolución Judicial firme.

4.2 REQUERIMIENTOS

4.2.1 VISIÓN

4.2.1.1 VISIÓN DEL PROYECTO

La visión se orienta a ubicar al RENIEC como única entidad líder que administre el registro de los Hechos Civiles, siendo por ende, la institución que contenga toda la información de los ciudadanos peruanos.

4.2.1.2 PROBLEMAS

En la actualidad la información de los hechos registrales es administrada localmente por las municipalidades que a su vez reportan la información para actualizar la información en el RENIEC. La información no se encuentra centralizada y es responsabilidad de cada municipalidad. La información por ser distribuida no puede ser administrable y difícilmente auditable. Los costos en el traslado de la información son altos corriéndose el riesgo de perder información.

4.2.1.3 MISIÓN

La misión del proyecto es desarrollar un Sistema Integrado de Registro y Almacenamiento de Información de los principales Actos Registrales realizados por los ciudadanos peruanos.

4.2.1.4 OBJETIVOS

Los objetivos del proyecto son los siguientes:

- Base de datos única y centralizada de Actos Registrales.
- Emisión de Certificaciones desde cualquier punto del país
- Economía en el manejo de información

- Congruencia y seguridad en la información obtenida.

4.2.2 ESPECIFICACIONES SUPLEMENTARIAS

4.2.2.1 BASE LEGAL

1. Ley Nro. 26497 - Ley Orgánica del Registro Nacional de Identificación y Estado Civil
2. Resolución Jefatural Nro. 023 - 95 - JEF - Reglamento de Organización y Funciones del Registro Nacional de Identificación y Estado Civil y sus modificatorias.
3. Directiva N° 015-94-INEI/SJI, Normas Técnicas para el Almacenamiento y Respaldo de la Información que procesa en las Entidades del Estado.
4. Directiva N° 007-95 – INEI/SJI, Recomendaciones Técnicas para la Seguridad e Integridad de la Información que se procesa en la Administración Pública.

4.2.2.2 LISTA DE RIESGOS

Podemos encontrar los siguientes tipos de riesgo:

Riesgos de requerimientos

Es el riesgo en el que se incurre al desarrollar un sistema erróneo que no hace lo que el cliente desea que haga. Para ello se han de identificarse bien los requerimientos y sus prioridades relativas.

Riesgos tecnológicos

En este punto sale a relucir la experiencia necesaria en el diseño orientado a objetos. Por otro lado, a partir de la opción tecnológica

seleccionada y las observaciones que se hagan sobre su funcionamiento, la funcionalidad que proporciona. El sistema ha sido concebido para ser desarrollado en Java bajo un esquema de tres capas utilizando para ello la solución de Enterprise Java Beans (EJB) y Java Server Pages dentro de un entorno web.

Riesgos de habilidades

Dada en términos de asesoría y expertos necesarios. En el equipo existe una persona con amplia experiencia en las soluciones adoptadas, teniendo a su cargo a dos desarrolladores que si bien conocen al Java como lenguaje de programación, no están familiarizados con las soluciones anteriormente especificadas

Riesgos políticos

Identificación de fuerzas políticas que se puedan interponer en el camino y afectar seriamente el proyecto. Para el caso de los Registros Civiles se encontrará una respuesta negativa por parte de las municipalidades que perderán una gran fuente de ingresos.

4.2.2.3 POLÍTICAS DE MANEJO DE RIESGOS

Manejo de los riesgos de requerimientos

Los requerimientos son importantes, y es donde las técnicas del UML son especialmente provechosas. El punto de partida son los casos de uso. Éstos, por lo tanto, son los motores de todo el proceso de desarrollo.

Manejo de los riesgos tecnológicos

Lo más importante que se debe hacer al abordar los riesgos tecnológicos es construir prototipos que prueben las partes tecnológicas con

las que se piensa trabajar. Los riesgos tecnológicos mayores son inherentes a la manera como se integran los componentes de un diseño, en lugar de hallarse en los componentes mismos. La integración no es sencilla, y por ello es importante obtener todos los componentes con los que se pretende trabajar e integrarlos en esta etapa. También en esta etapa deberá ocuparse de cualquier decisión de diseño arquitectónico. Estas decisiones por lo general toman la forma de ideas acerca de lo que son los componentes y sobre la manera como se construirán sobretodo en sistemas distribuidos.

Manejo de los riesgos de habilidades

El entrenamiento es una forma de evitar errores, Otra forma de aumentar conocimientos es mediante la lectura de libros técnicos. También se puede manejar la tutoría para adquirir habilidades en el diseño orientado a objetos, dado que a partir de la tutoría un desarrollador experimentado puede colaborar con los demás en el proyecto durante un largo período de tiempo.

Manejo de los riesgos políticos

Es el trabajo más difícil, y debe contar para encararlos de un hábil político corporativo. Dentro de la realidad de las empresas estatales este factor se hace mayor, y su naturaleza es totalmente externa respondiendo más a factores coyunturales.

4.2.2.4 IDENTIFICACIÓN DEL ENTORNO TECNOLÓGICO

El sistema de registros civiles es una solución orientada a Internet, por lo tanto su uso se realizara en ambiente Web y el acceso al mismo será a través de la Intranet y de Internet.

Actualmente el RENIEC cuenta con equipos de ultima generaci3n, asimismo utiliza la BD Oracle catalogada como una de las mejores BD del mercado, cuenta con las herramientas de desarrollo de Oracle como Developer, Jdeveloper, y el servidor de aplicaciones 9i.

Teniendo en cuenta todas estas fortalezas tecnolog3as, creemos que las expectativas en cuanto al requerimiento tecnolog3ico est3n cubiertas.

En cuanto a la arquitectura tecnolog3ica ha emplear en el desarrollo del proyecto ser3 la Arquitectura de tres capas, utilizando el patr3n de dise1o MVC.

Estas son las especificaciones para el presente proyecto:

- **De software**

- **Servidor**

Sistema Operativo:	Windows 2000
Servidor Web:	Apache 1.3.27
Compilador Java	j2sdk1.4.0
Servidor de Aplicaciones:	Tomcat 4.1.29
Conector JDBC para Base de Datos Oracle:	ORC 40

- **Cliente**

Sistema Operativo:	Windows 9x/2k/Me/XP
Navegador de Internet:	MS Internet Explorer 4.0 o superior
JRE	1.4.2

- **De Hardware**

- **Servidor**

- IBM Netfinity 8100

- Pentium III 800 MHz

- 1GB de RAM

- 36 GB de Disco Duro

- **Cliente**

- Pentium II 233 MHz

- 256Kb de RAM

- 5 GB de Disco Duro

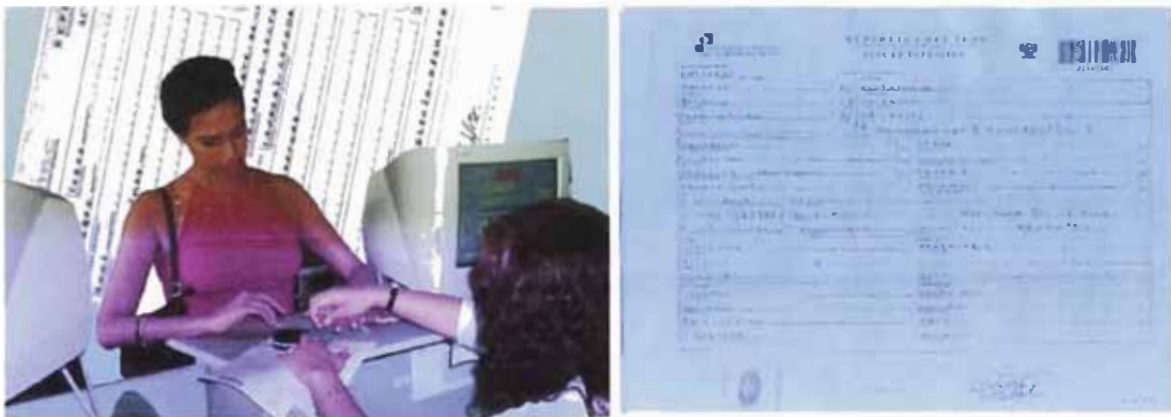
4.2.3 GLOSARIO

4.2.3.1 CONCEPTOS

- **Acta.-** Es un documento público que tiene validez legal para sustentar un registro civil. Existen tres tipos de actas :
 - Acta de Nacimiento.
 - Acta de Matrimonio.
 - Acta de Defunción.
- **Usuario.-** Persona con acceso al aplicativo, a la cual se le asigna un perfil ,estos son
 - Registrador Civil
 - Agencia Reniec

- Administrador
 - Operador de Línea de Producción
 - Despacho Logístico
- **Ciudadano.-** Persona registrada en la Base de Datos del RENIEC. Para tal efecto la persona debe ser mayor de edad y de nacionalidad peruana.
 - **Certificación.-** Son copias certificadas del Acta Original, son documentos públicos y prueban fehacientemente los hechos a que se refieren, salvo que se declare judicialmente la invalidez de dichos documentos, o se rectifique o cancele la información inscrita.
 - **Escaneo de Actas.-** Digitalización de las Actas manuales registradas con un número único de identificación.
 - **Rectificación .-** Se realiza cuando existen Actas con datos errados o Actas con omisión de datos relativos a los intervinientes del acto registral.
 - **Línea de Producción.-** Comprende los Módulos de Digitación masiva, escaneo masivo y Control de calidad de imágenes de las Actas ,que se realizará en la Sede Central RENIEC.

- **Despacho Logístico.-** Área del Reniec encargada entre otras de la distribución del papel que se utilizará en las certificaciones.
- **Oficina Registral.-** Establecimiento donde se pueden realizar los registros civiles tales como la generación de actas, la emisión de certificaciones, etc- - Cada registrador civil pertenece a una oficina registral que a su vez corresponde a una ubicación geográfica específica.



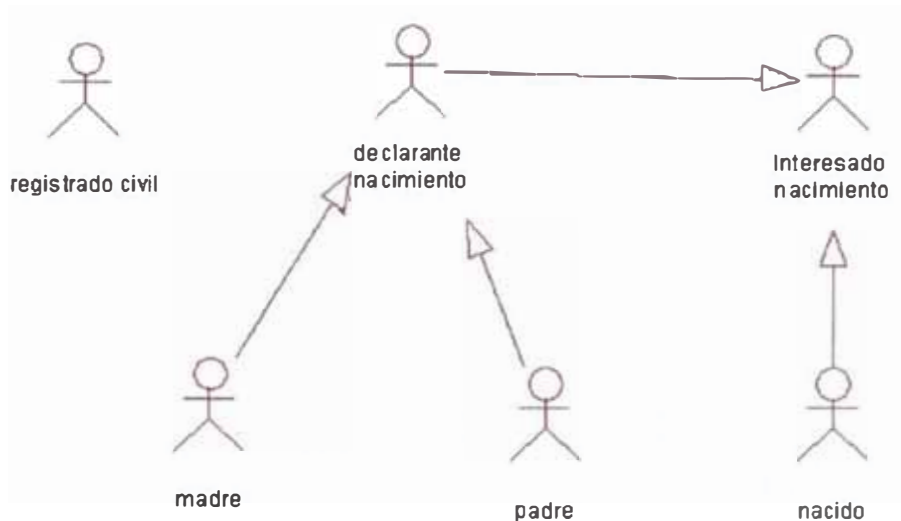
**Figura 39 Recepción de documentos para emisión de certificación de
acta**

4.3 MODELO DEL NEGOCIO

4.3.1 MODELO DE CASOS DE USO

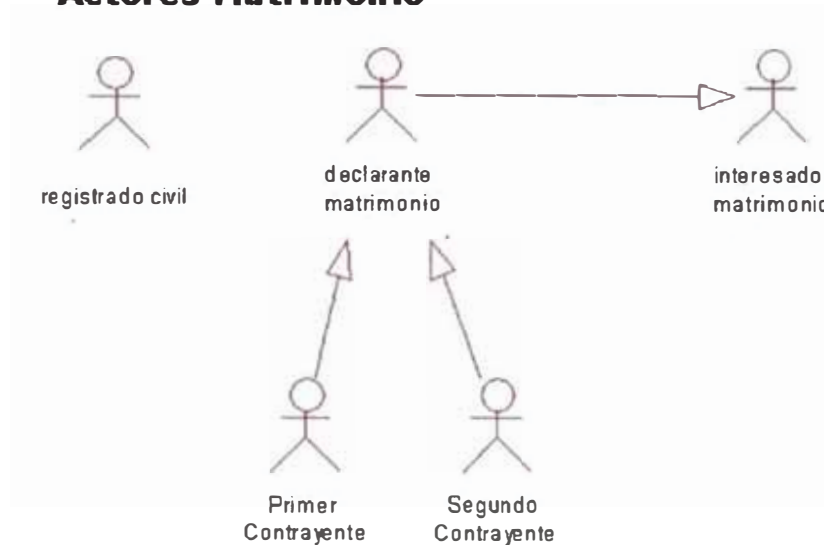
4.3.1.1 Actores

Actores Nacimiento



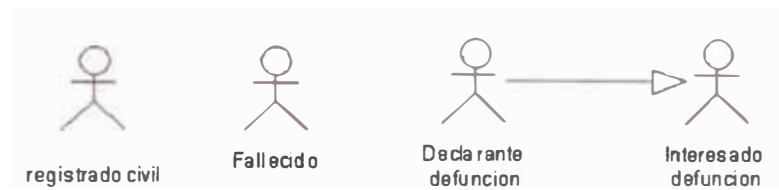
- **Declarante nacimiento:** Es el que inicia las operaciones en nacimientos, ya que él solicita una acción registra. Puede ser el padre, madre ó ambos.
- **Registrador Civil:** Es el actor que se encarga de recibir e ingresar los datos que trae el declarante.
- **Interesado nacimiento** Es un actor que en algún momento necesita una certificación de un acta de nacimientos. Así mismo puede realizar una rectificación (Administrativa, Notarial, Judicial).

Actores Matrimonio



- **Declarantes matrimonio:** Es el que inicia las operaciones en matrimonios, ya que él solicita una acción registra.
- **Registrador Civil:** Es el actor que se encarga de recibir e ingresar los datos que trae el declarante.
- **Primer contrayente:** Es una de las personas que conforman la pareja a casarse.
- **Segundo contrayente:** Es una de las personas que conforman la pareja a casarse.
- **Interesado matrimonio:** Es un actor que en algún momento necesita una certificación de un acta de matrimonios. Así mismo puede realizar una rectificación (Administrativa, Notarial, Judicial).

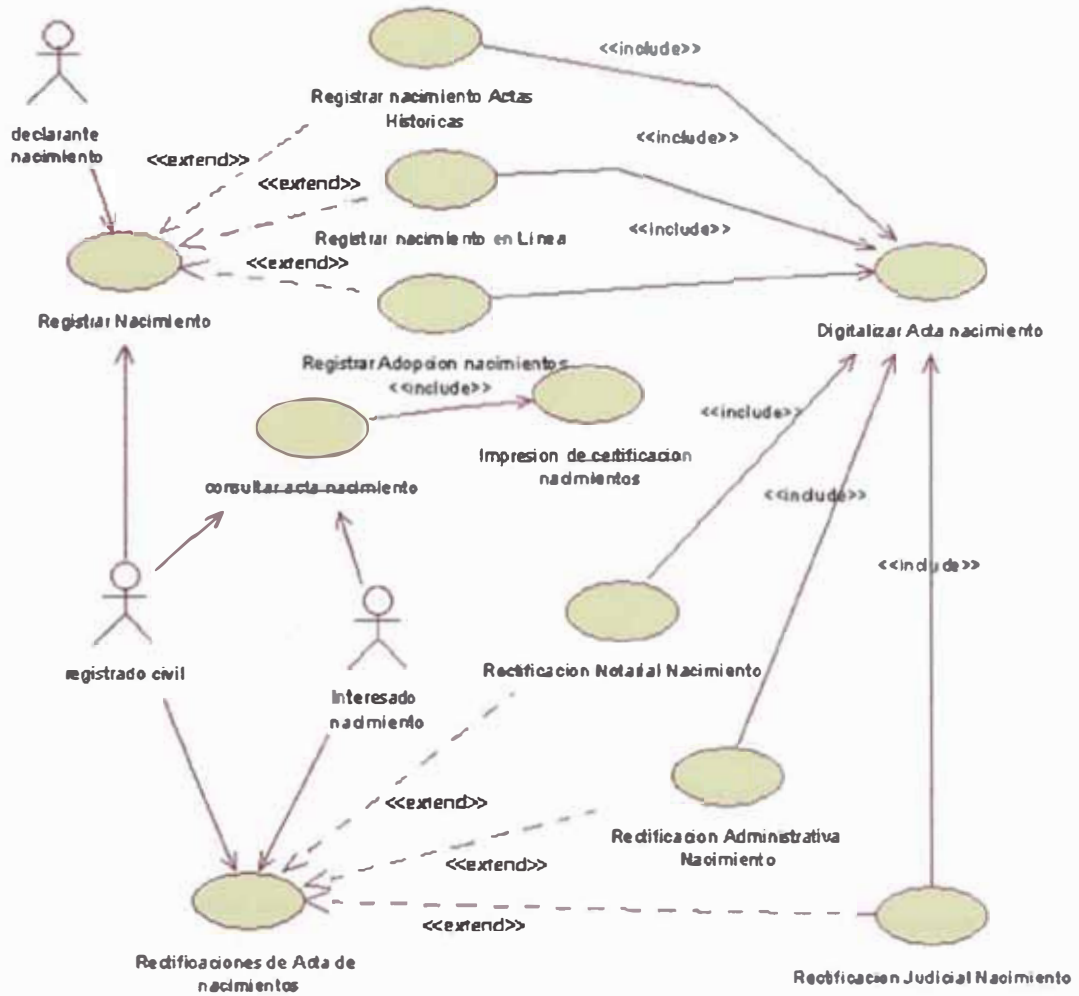
Actores Defuncion



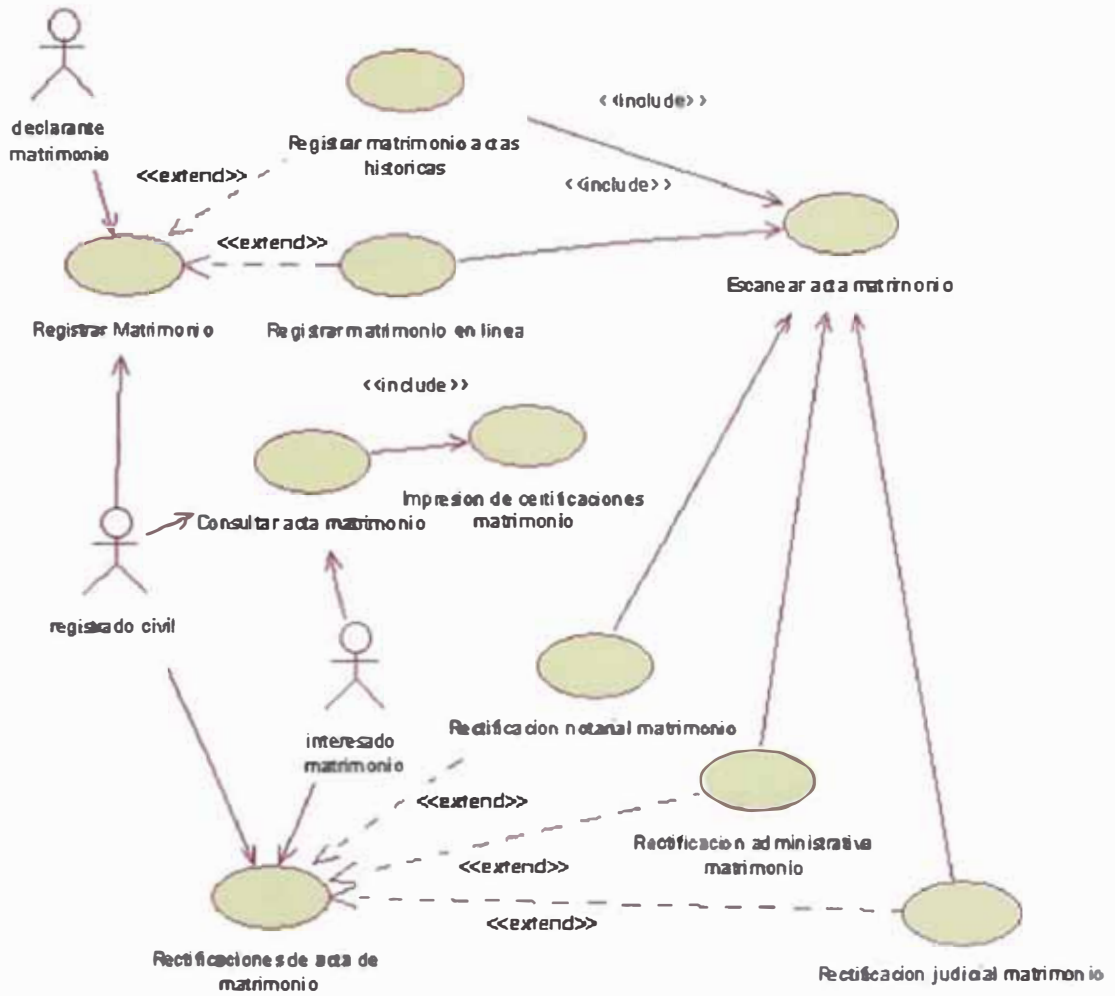
- **Declarante defunción:** Es el que inicia las operaciones en defunciones, ya que él solicita una acción registra.
- **Registrador Civil:** Es el actor que se encarga de recibir e ingresar los datos que trae el declarante.
- **Fallecido:** Viene a ser la persona que por una acción determinado pierde la vida. La ocurrencia de esta acción es la genera un registro de defunción.
- **Interesado matrimonio:** Es un actor que en algún momento necesita una certificación de un acta de defunciones. Asimismo puede realizar una rectificación (Administrativa, Notarial, Judicial).

4.3.1.2 Diagramas de Casos de Uso

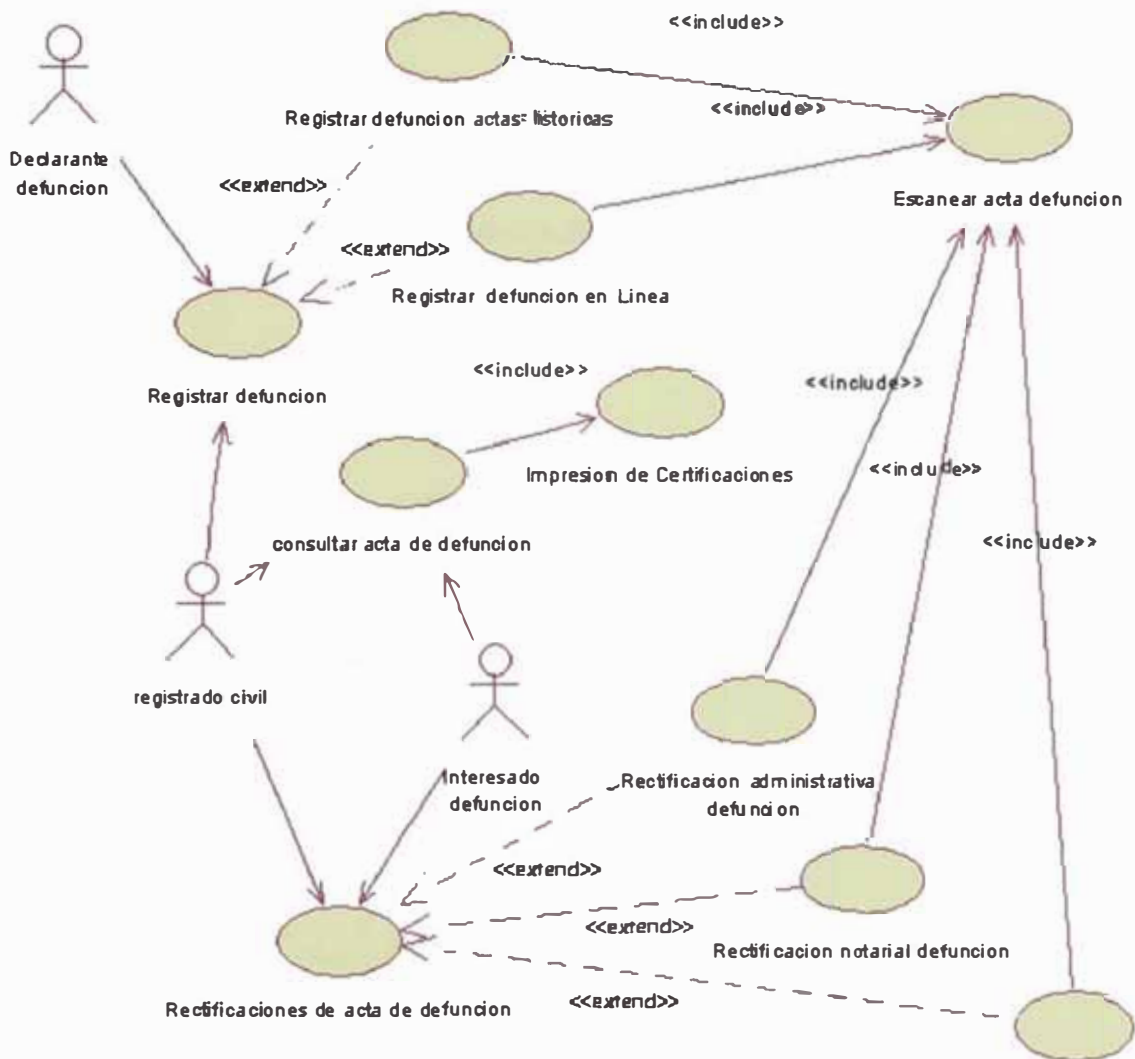
NACIMIENTO



MATRIMONIO



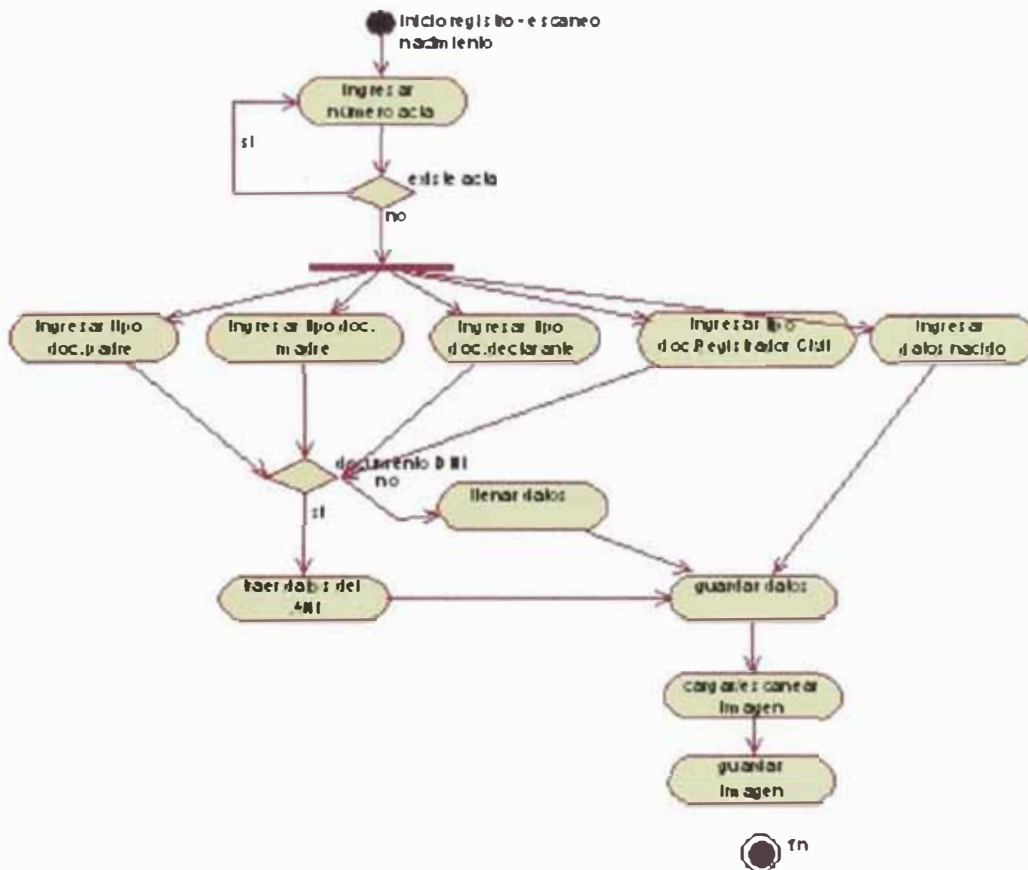
DEFUNCION



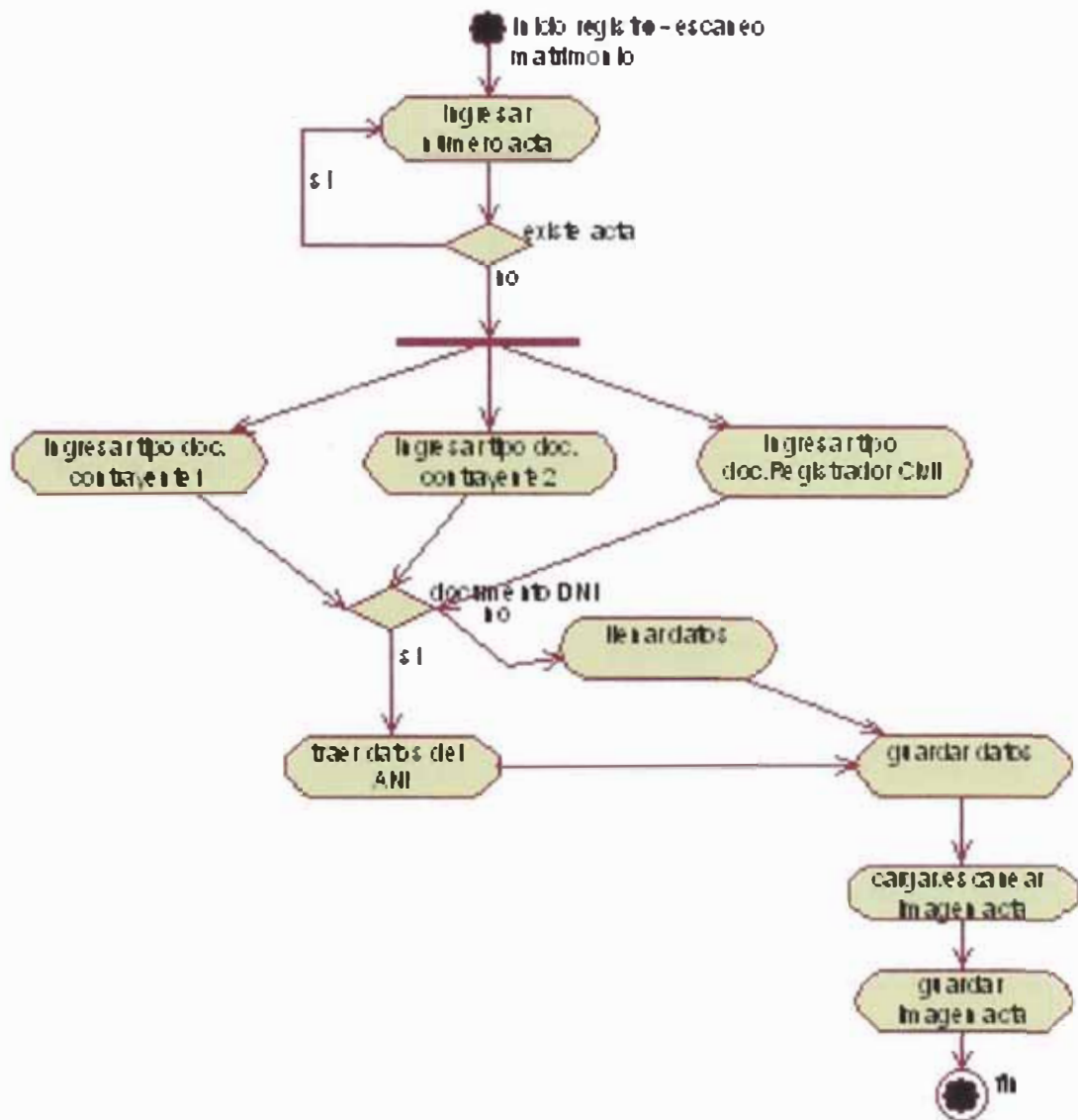
4.3.2 IDENTIFICACIÓN DE PROCESOS DEL SISTEMA

4.3.2.1 DIAGRAMAS DE ACTIVIDAD

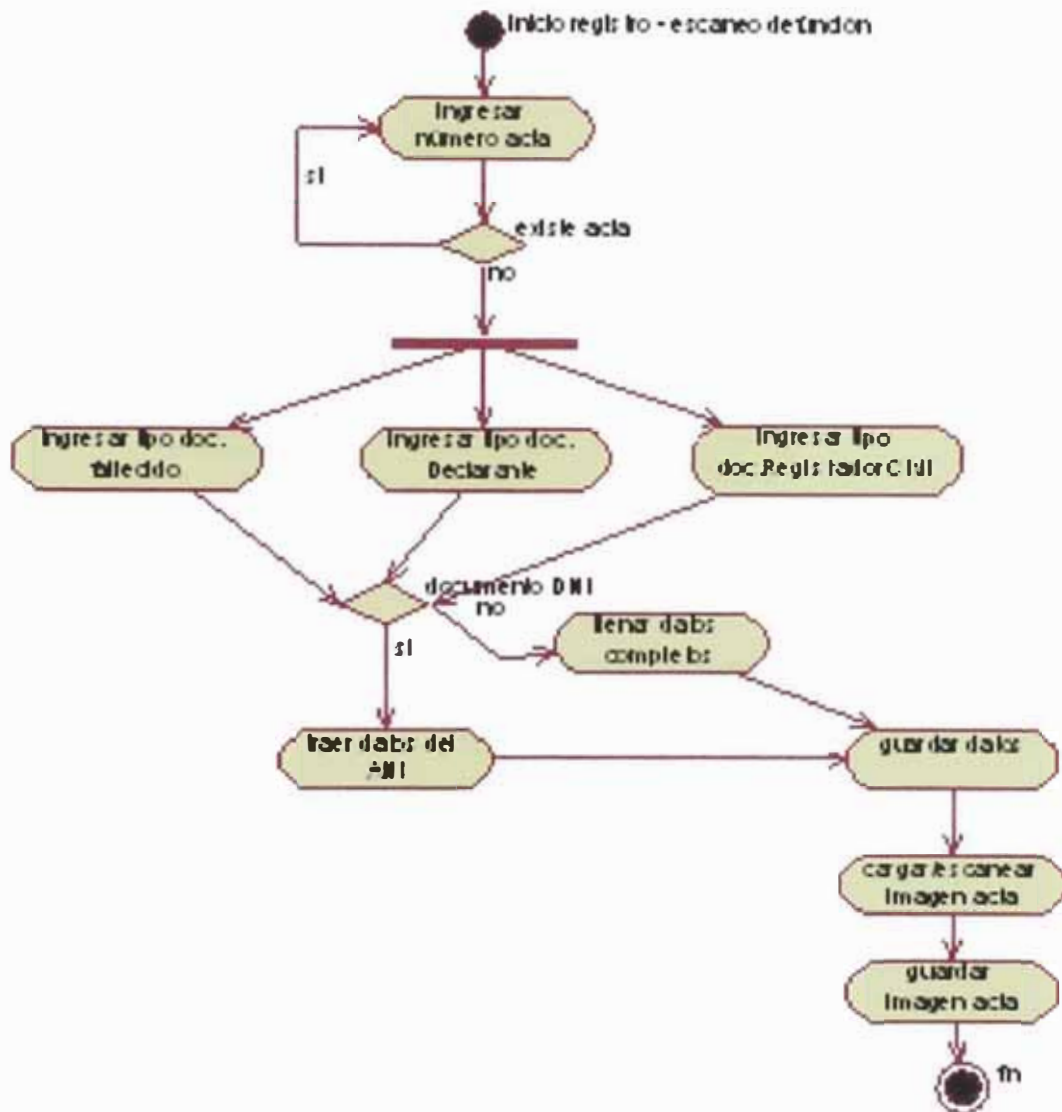
Registrar nacimientos en línea



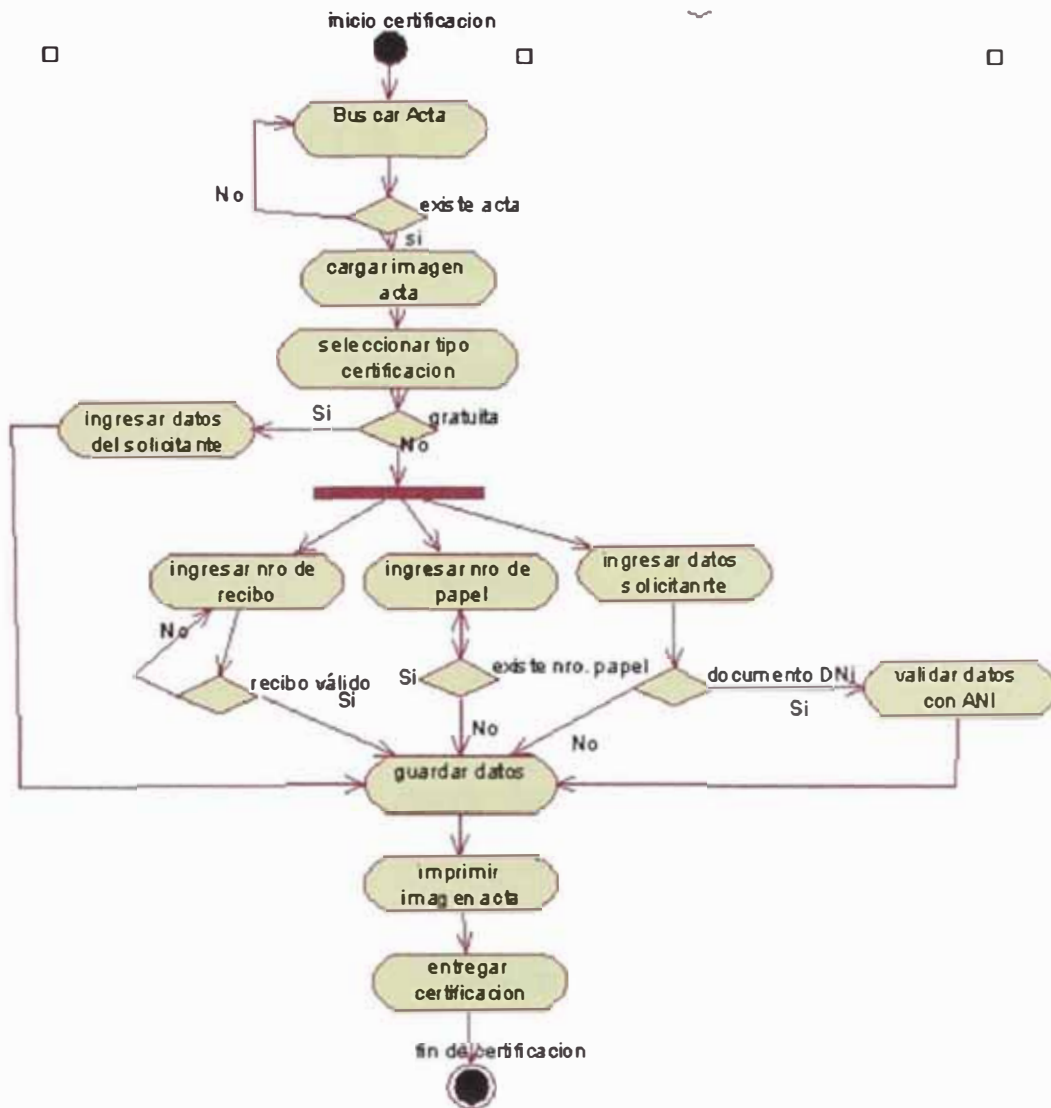
Registrar matrimonios en línea



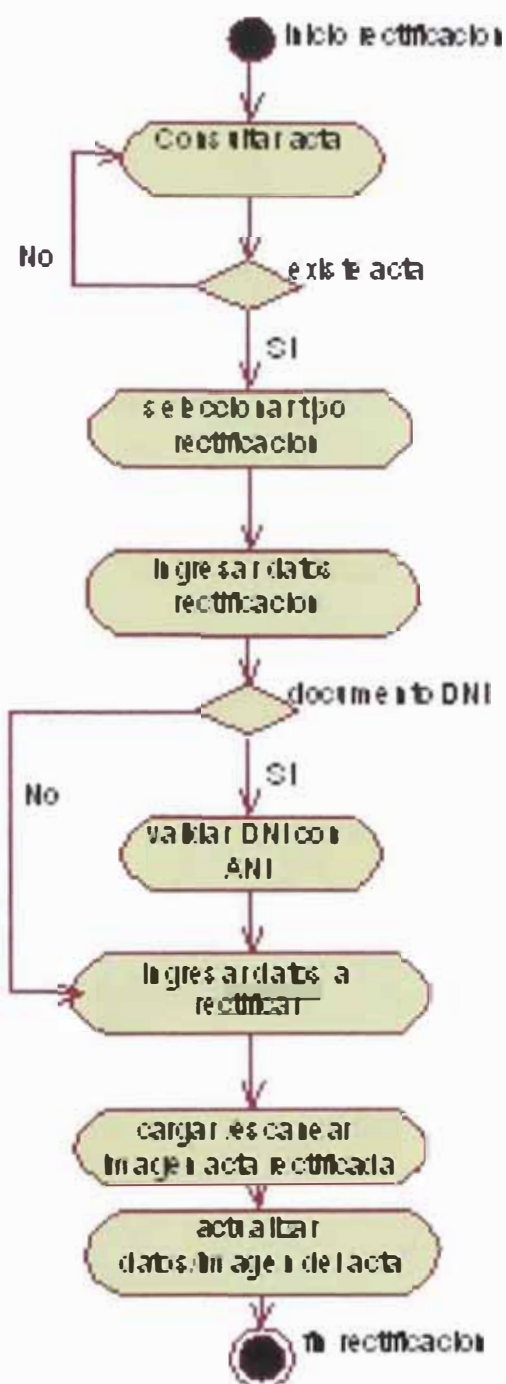
Registrar defunciones en línea



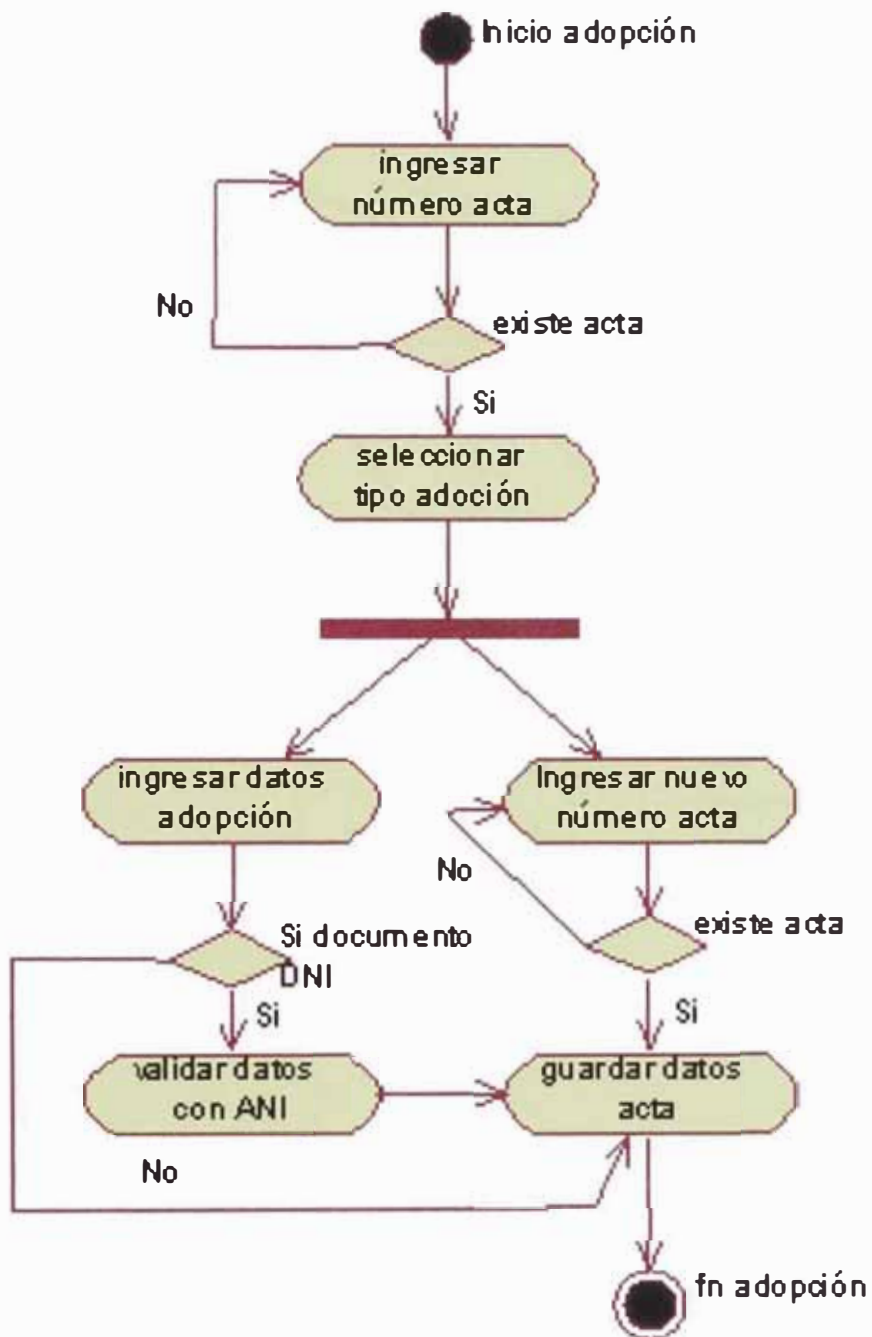
Imprimir Certificación de acta



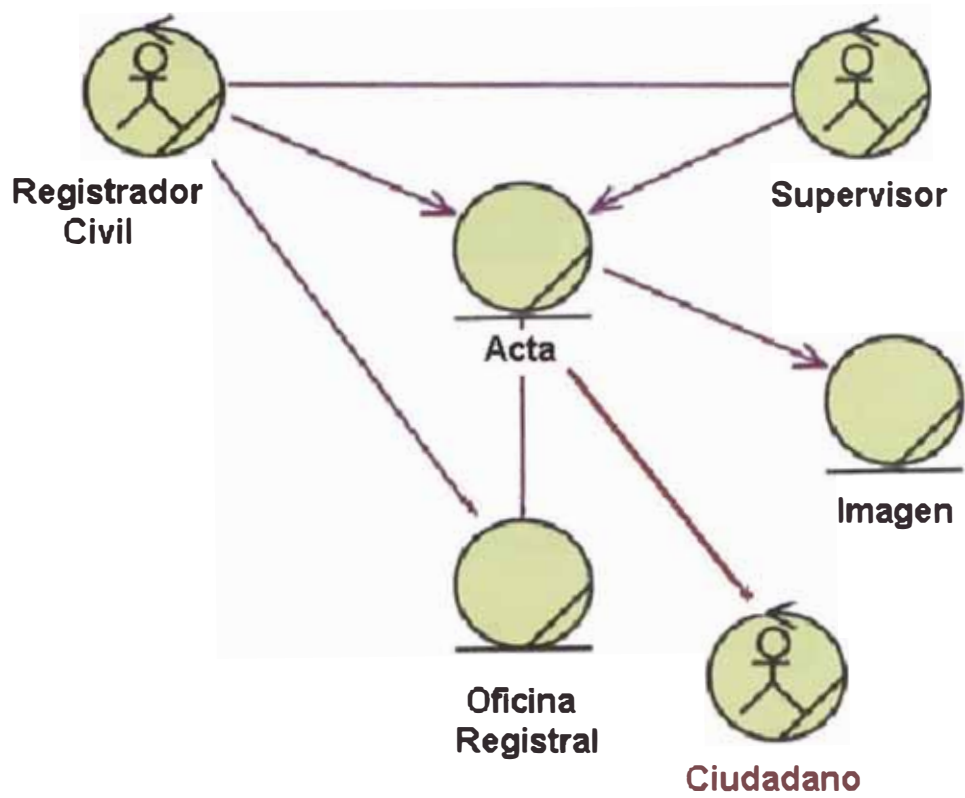
Registrar rectificación de acta



Registrar adopción nacimientos



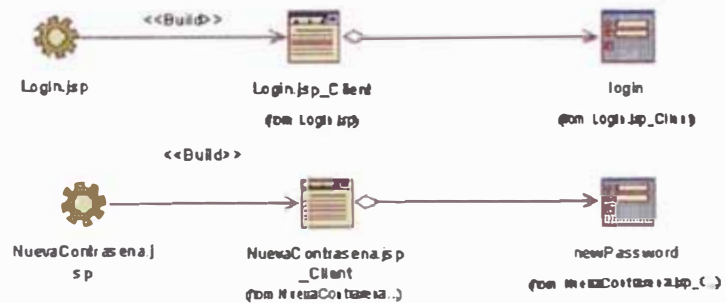
4.3.3 DIAGRAMA DE CLASES DE DOMINIO

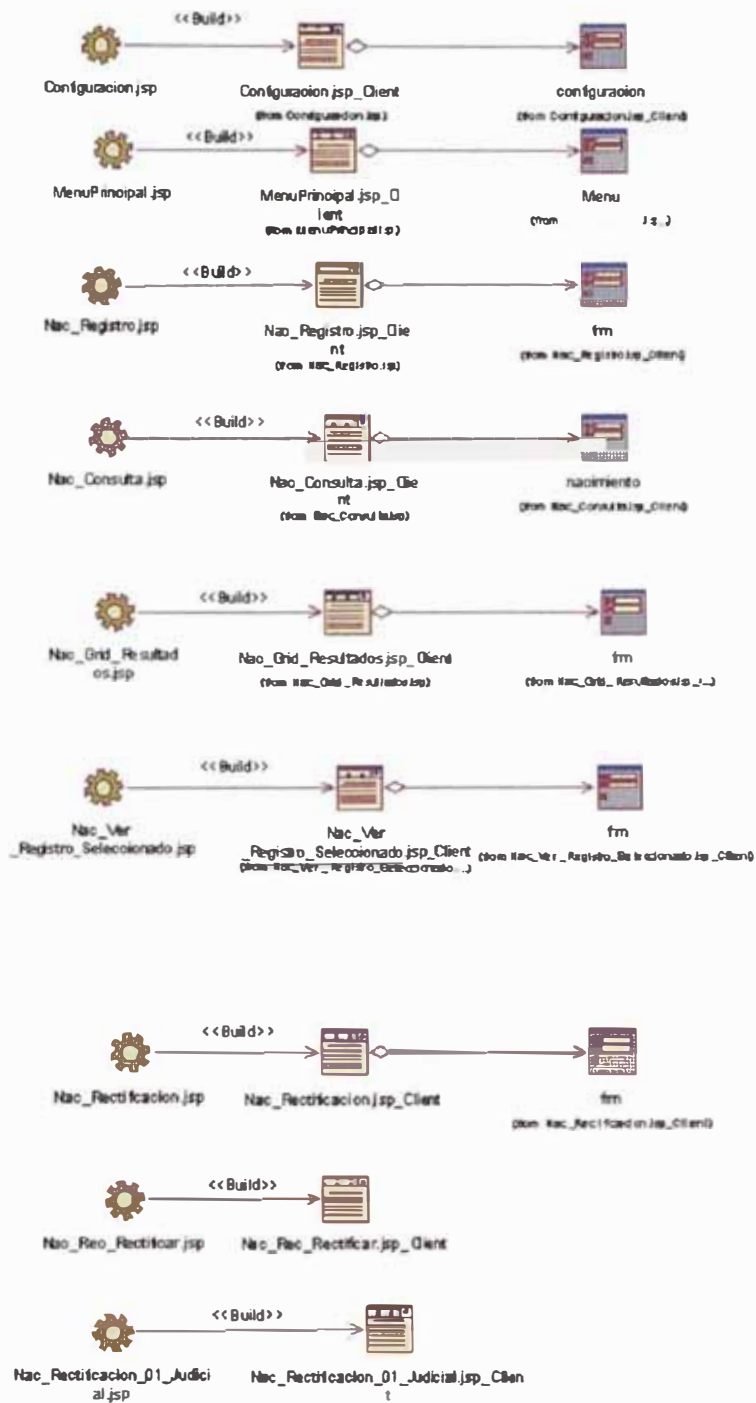


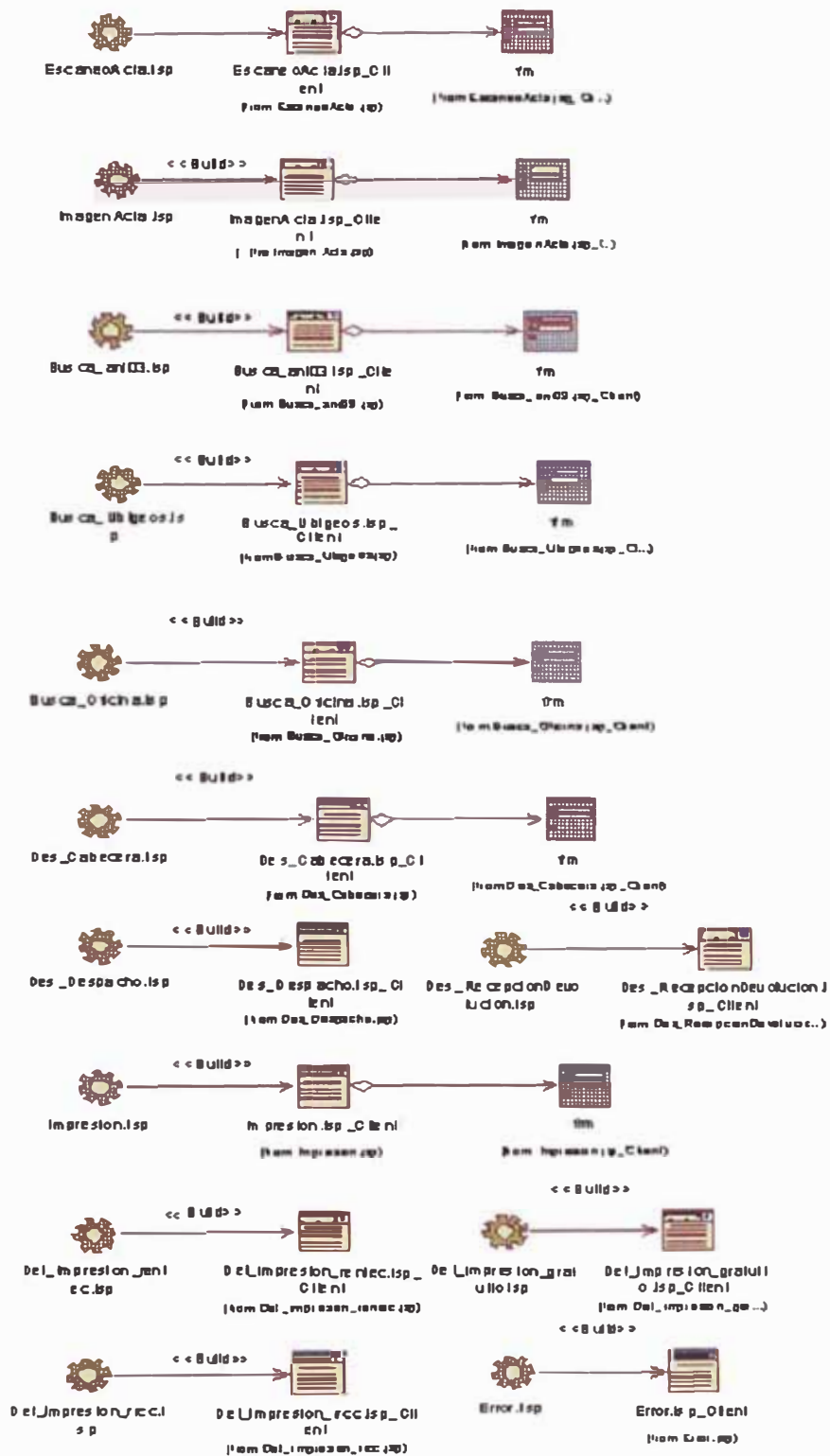
4.3.4 PROTOTIPEO DE LAS INTERFACES

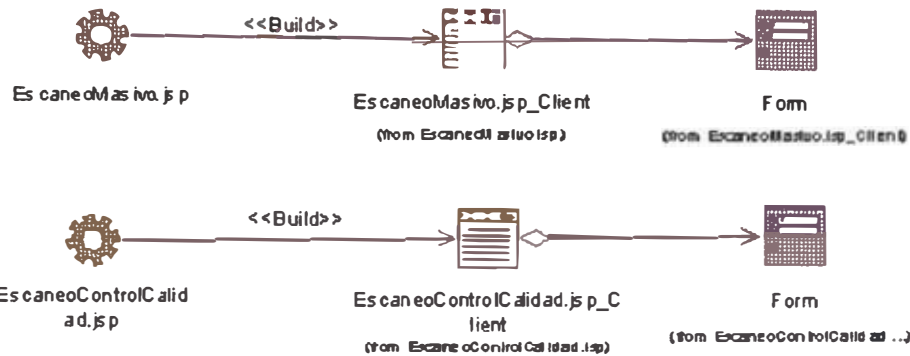


4.3.5 DIAGRAMAS DE NAVEGACIÓN









4.4 MODELO DE CONCEPCIÓN

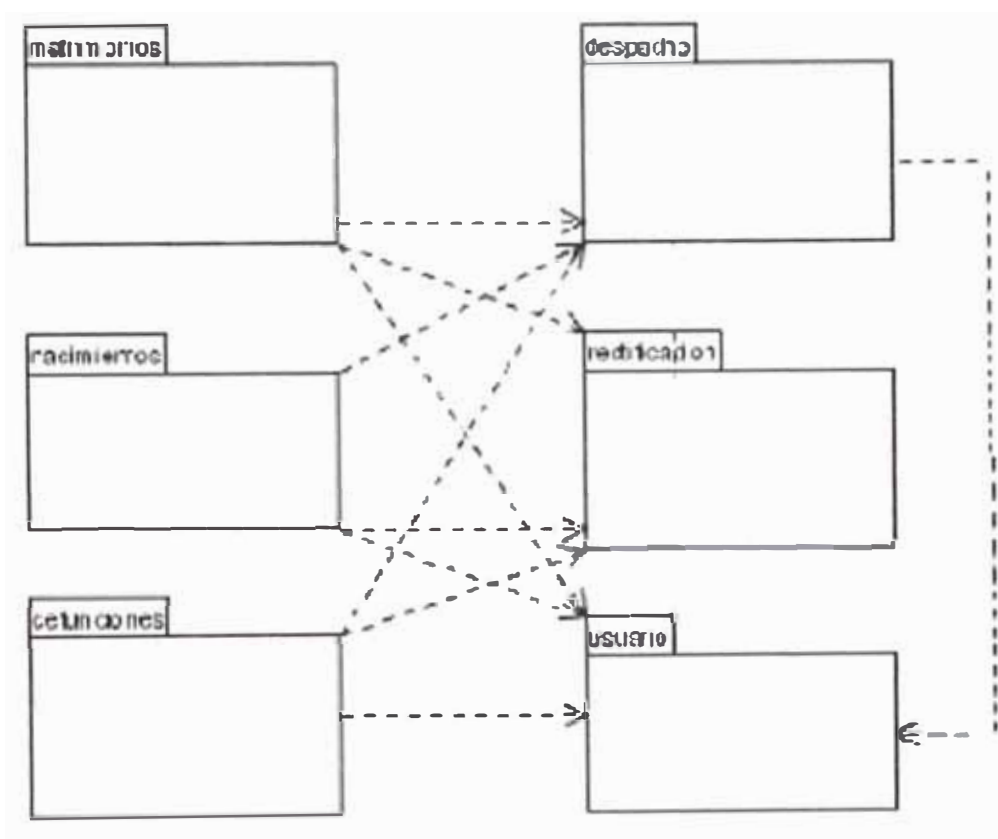
Este es otro punto importante dentro del modelado de análisis. Los siguientes diagramas representan la realización de los casos de uso elaborados en el análisis y sirven para separar interfaz y lógica interna del negocio. Estos diagramas muestran las operaciones que suceden en un caso de uso determinado, lo que nos permiten ver de manera dinámica la secuencia de sucesos que se dan al interior del caso de uso. Para cada caso de uso vamos a realizar dos tipos de diagramas de interacción de objetos, diagramas de secuencia y diagramas de colaboración. El diagrama de secuencia muestra al lado izquierdo en forma de descripción las acciones que se realizan por el caso de uso actual, y al lado derecho las operaciones que se realizarán con el sistema por cada acción nombrada. Tan igual como en las operaciones anteriores, se divide la presentación de los diagramas por área de negocios.

4.4.1 CLASES QUE REALIZAN LOS CASOS DE USO

En esta actividad procederemos a presentar todas las clases que hemos hallado sobre la base de las etapas anteriores, sus características y comportamiento. Las clases son el vocabulario y terminología de una área

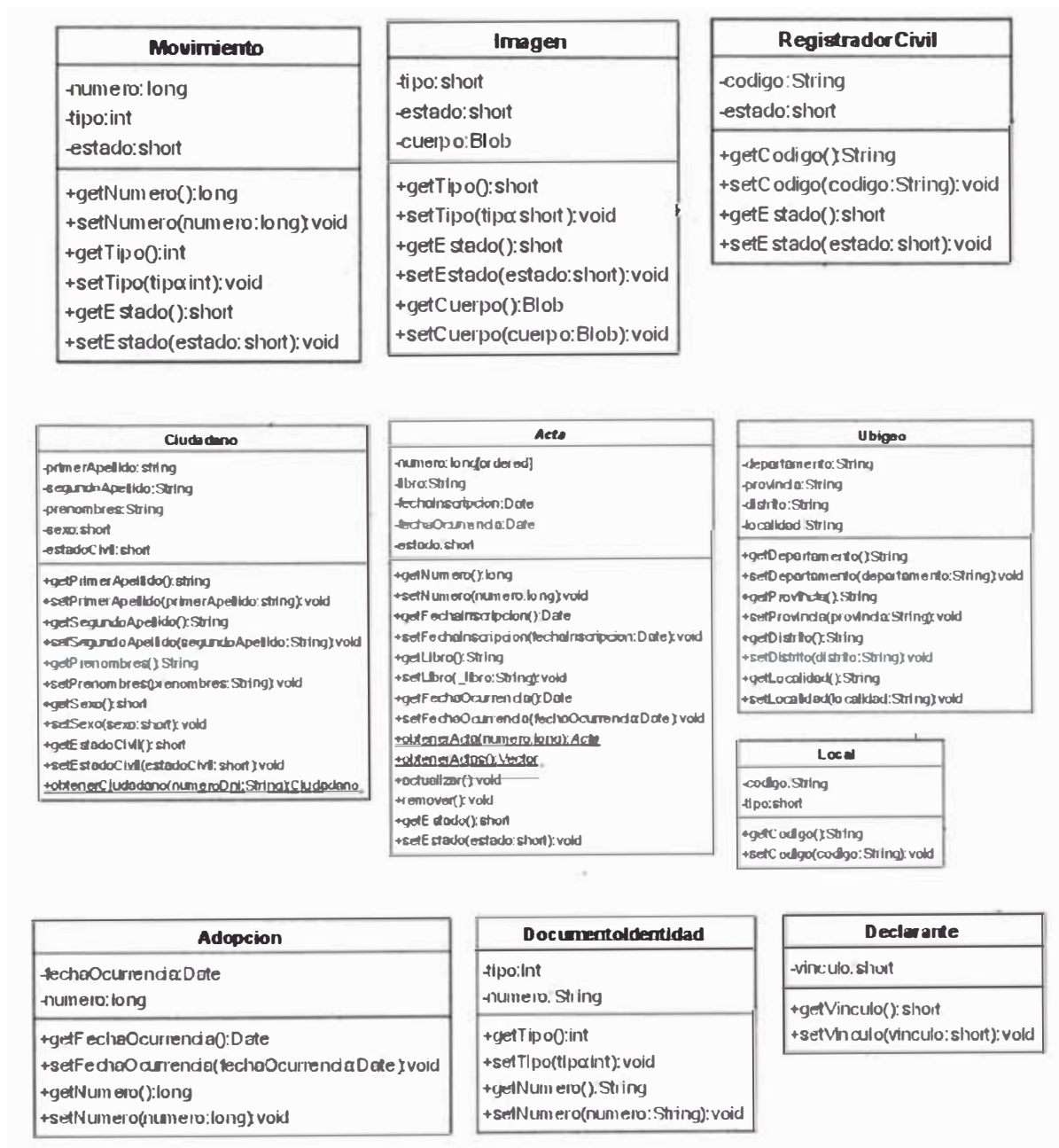
del conocimiento. Conforme hayamos analizado a registros civiles, habremos adquirido conocimientos respecto a la terminología de la misma.

4.4.2 DIAGRAMA DE PAQUETES

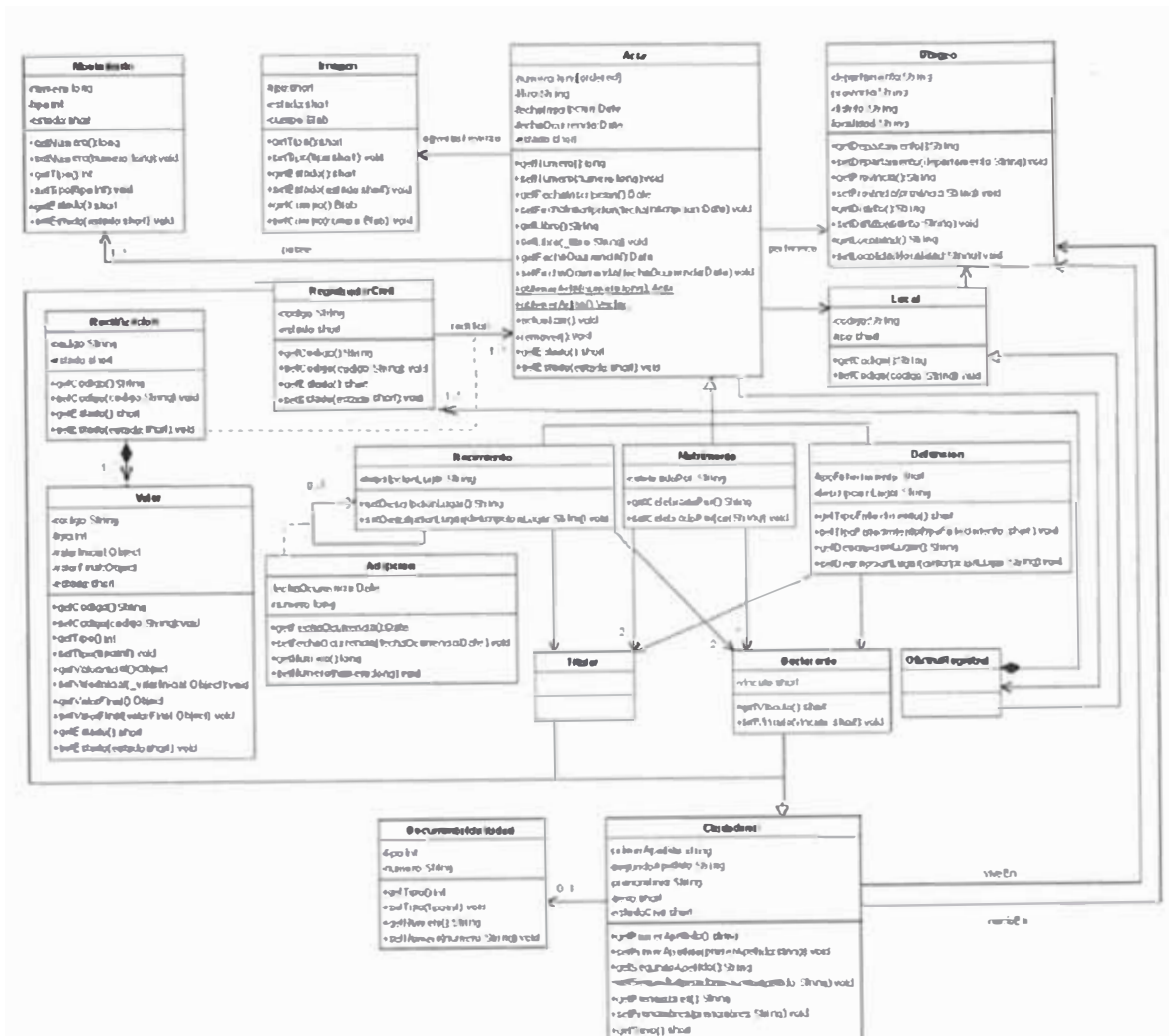


4.4.3 IDENTIFICACION DE OPERACIONES Y ATRIBUTOS

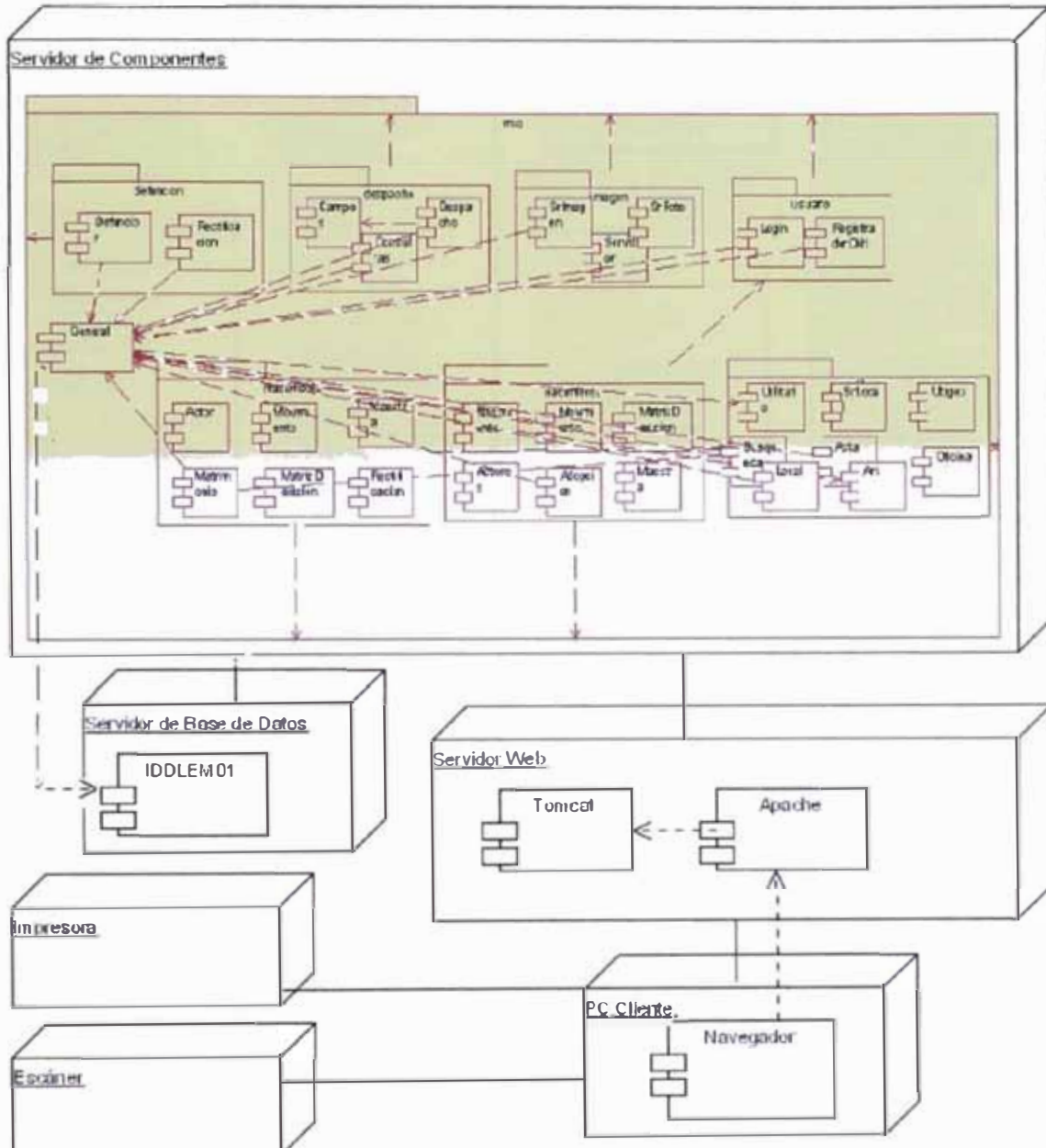
Cuando hablamos de características de una clase, entendemos que son los atributos o propiedades. Una clase puede contener varios o ningún atributo, a cada atributo le corresponde un tipo de valor específico. Al referirnos al comportamiento, estamos hablando de las operaciones que la clase puede realizar.



4.4.4 DIAGRAMA DE CLASES DEL SISTEMA

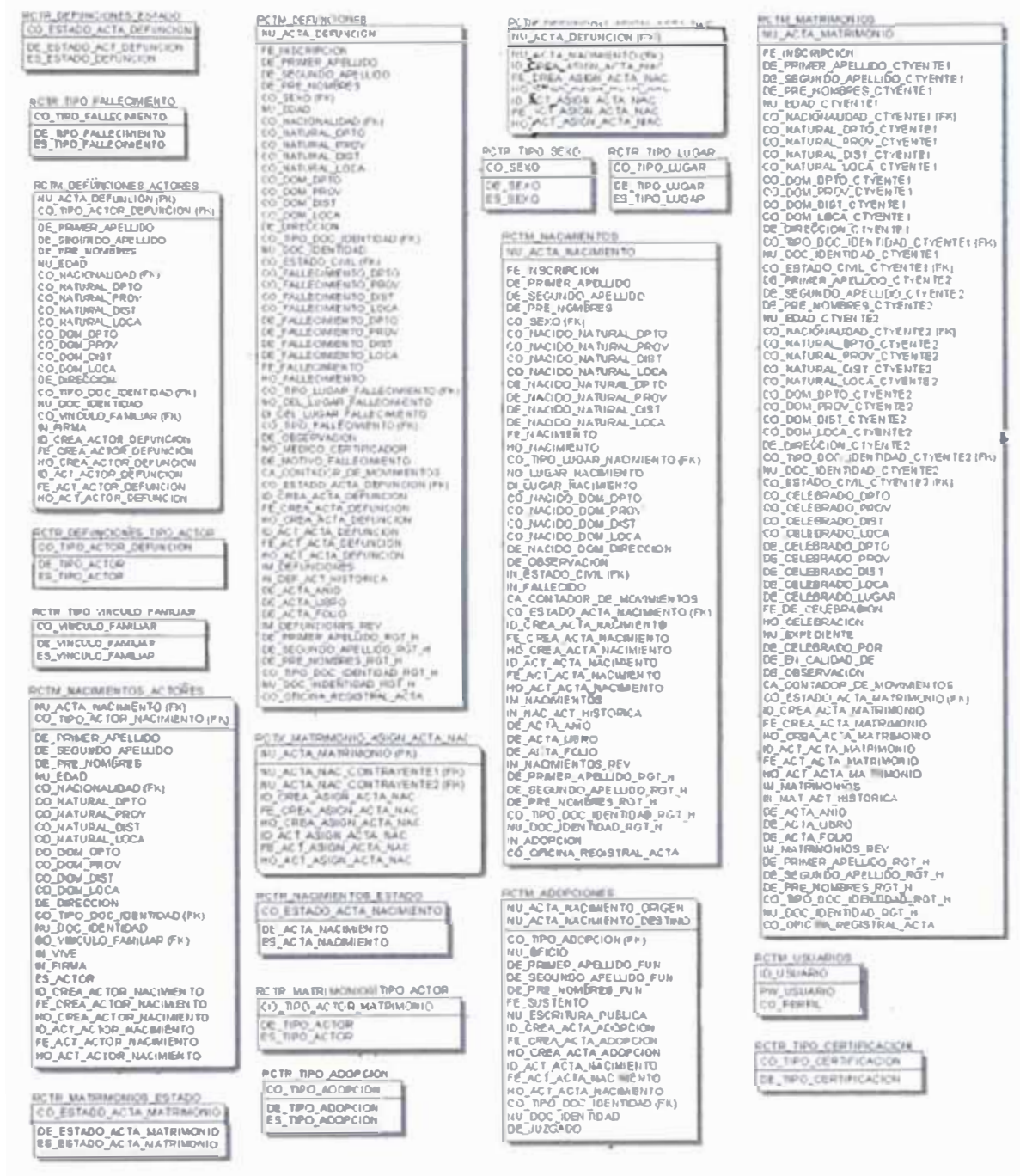


4.5.2 DIAGRAMA DE DESPLIEGUE



4.6 MODELO DE DATOS

4.6.1 RELACION DE ENTIDADES



RCTM MATRIMONIOS ACTORES

NU_ACTA_MATRIMONIO (FK)
CO_TIPO_ACTOR_MATRIMONIO (FK)
DE_PRIMER_APELLIDO
DE_SEGUNDO_APELLIDO
DE_PRE_NOMBRES
NU_EDAD
CO_NACIONALIDAD (FK)
CO_NATURAL_CPTO
CO_NATURAL_PROV
CO_NATURAL_OST
CO_NATURAL_LOCA
CO_DOM_CPTO
CO_DOM_PROV
CO_DOM_LOCA
DE_DIRECCION
CO_DOM_OST
CO_TIPO_DOC_IDENTIDAD (FK)
NU_DOC_IDENTIDAD
DE_VINCULO_FAMILIAR
IN FIRMA
ID_CREA_ACTOR_MATRIMONIO
FE_CREA_ACTOR_MATRIMONIO
HO_CREA_ACTOR_MATRIMONIO
ID_ACT_ACTOR_MATRIMONIO
FE_ACT_ACTOR_MATRIMONIO
HO_ACT_ACTOR_MATRIMONIO

OSLLOC
CODLOC

RCTR TIPO LOCAL
CO_TIPO_LOCAL
DE_TIPO_LOCAL

RCTR ESTADO USO PAPEL
CO_ESTADO_USO_PAPEL
DE_ESTADO_USO_PAPEL

RCTX SOLICITANTE

NU_PAPEL (FK)
CO_TIPO_DOC_IDENTIDAD (FK)
NU_DOC_IDENTIDAD
DE_PRIMER_APELLIDO
DE_SEGUNDO_APELLIDO
DE_PRE_NOMBRES

RCTV USO PAPEL

NU_PAPEL (FK)
CO_TIPO_ACTA (FK)
NU_ACTA
CO_TIPO_LOCAL (FK)
CO_LOCAL_REGISTRO_CIVIL (FK)
CO_LOCAL (FK)
CO_ESTADO_USO_PAPEL (FK)
ID_REGISTRO
FE_REGISTRO
HO_REGISTRO
ID_ACTUALIZA
FE_ACTUALIZA
HO_ACTUALIZA

RCTR TIPO OPERACION MATERIALES

CO_TIPO_OPERACION_MATERIALES
DE_TIPO_OPERACION_MATERIALES

RCTR TIPO DOC REMISION

CO_TIPO_DOC_REMISION
DE_TIPO_DOC_REMISION

RCTR LADO ACTA

CO_LADO_ACTA
DE_LADO_ACTA

RCTR ESTADO REMISION

CO_ESTADO_REMISION
DE_ESTADO_REMISION
ES_ESTADO_REMISION

RCTX SOLICITANTE CERTIF GRATUITA

NU_ACTA (FK)
CO_TIPO_ACTA (FK)
SE_CERTIFICACION_GRATUITA (FK)
CO_TIPO_DOC_IDENTIDAD (FK)
NU_DOC_IDENTIDAD
DE_PRIMER_APELLIDO
DE_SEGUNDO_APELLIDO
DE_PRE_NOMBRES

RCTM CERTIFICACION GRATUITA

NU_ACTA
CO_TIPO_ACTA (FK)
SE_CERTIFICACION_GRATUITA
CO_TIPO_LOCAL (FK)
CO_LOCAL_REGISTRO_CIVIL (FK)
CO_LOCAL (FK)
FECHA
HORA
CO_MOTIVO_ORA_NU_IDADO (FK)
DE_OBSERVACION

RCTR MOTIVO GRATUIDAD

CO_MOTIVO_GRATUIDAD
DE_MOTIVO_GRATUIDAD

RCTR TIPO RECIBO

CO_TIPO_RECIBO
DE_TIPO_RECIBO

RCTR TIPO ACTA

CO_TIPO_ACTA
DE_TIPO_ACTA

RCTV UB PAPEL

NU_PAPEL (FK)
SE_DESPACHO (FK)
CO_ESTADO_REMISION (FK)
ID_REGISTRO
FE_REGISTRO
HO_REGISTRO
ID_ACTUALIZA
FE_ACTUALIZA
HO_ACTUALIZA

RCTR ESTADO ESCANEADO

CO_ESTADO_ESCANEO
DE_ESTADO_ESCANEO
ES_ESCANEO

RCTV ACTA RECTIFICACION NA

NU_ACTA_NACIMIENTOS (FK)
NU_SECUENCIA_NACIMIENTOS (FK)
NU_CREC
NU_RESOLUCION
FE_RESOLUCION
FE_SENTENCIA
FE_ESCRITURA
FE_ESCRITURA
FE_EXPEDIENTE
FE_EXPEDIENTE
CO_TIPO_DOC_IDENTIDAD_FUN (FK)
NU_DOC_IDENTIDAD_FUN
DE_PRIMER_APELLIDO_FUN
DE_SEGUNDO_APELLIDO_FUN
DE_PRE_NOMBRES_FUN
DE_JUZGADO

RCTV ESCANEADO

SE_ESCANEO
NU_ACTA (FK)
CO_TIPO_ACTA (FK)
CO_LADO_ACTA (FK)
CO_ESTADO_ESCANEO (FK)
ID_REGISTRO
FE_REGISTRO
HO_REGISTRO
ID_ACTUALIZA
FE_ACTUALIZA
HO_ACTUALIZA

IDTM_OFICINA_REGISTRO_CIVIL
CO_OFICINA_RC

RCTR DEFUNICIONES CAMPOS

CO_CAMPO_DEFUNICION
DE_CAMPO_DEFUNICION
CO_TIPO_ACTOR_DEFUNICION (FK)
IN_CAMBO
DE_CAMPO_ACTUALIZA
ES_CAMPO_DEFUNICION

RCTV DEFUNICIONES VARIABLES

NU_ACTA_DEFUNICION (FK)
NU_SECUENCIA_DEFUNICION (FK)
CO_TIPO_ACTOR_DEFUNICION (FK)
DE_VALOR_ANTERIOR
DE_VALOR_NUEVO
DE_IN_VALOR_ANTERIOR
DE_IN_VALOR_NUEVO

RCTV DEFUNICIONESI

NU_ACTA_DEFUNICION (FK)
NU_SECUENCIA_DEFUNICION
CO_OFICINA_REGISTRAL (FK)
NU_DNI_REGISTRADOR (FK)
NU_SECUENCIA (FK)
FE_ACTA_DEFUNICION (FK)
CO_PROCESO_DEFUNICION (FK)
ID_CREA_MOVIMIENTO
FE_CREA_MOVIMIENTO
HO_CREA_MOVIMIENTO
ID_ACT_MOVIMIENTO
FE_ACT_MOVIMIENTO
HO_ACT_MOVIMIENTO

RCTV ACTA RECTIFICACION DEF

NU_ACTA_DEFUNICION (FK)
NU_SECUENCIA_DEFUNICION (FK)
NU_OFICIO
NU_RESOLUCION
FE_RESOLUCION
FE_SENTENCIA
FE_ESCRITURA
FE_ESCRITURA
FE_EXPEDIENTE
FE_EXPEDIENTE
CO_TIPO_DOC_IDENTIDAD_FUN (FK)
NU_DOC_IDENTIDAD_FUN
DE_PRIMER_APELLIDO_FUN
DE_SEGUNDO_APELLIDO_FUN
DE_PRE_NOMBRES_FUN
DE_JUZGADO

RCTR DEF COMBINACION DOCUMENTO

CO_COMBINACION
ES_COMBINACION

RCTV DEFUNICIONES DOCUMENTOS

NU_ACTA_DEFUNICION (FK)
NU_SECUENCIA_DEFUNICION (FK)
CO_TIPO_DOC_DEFUNICION (FK)
NU_DOC_DEFUNICION
FE_DOC_DEFUNICION

RCTR DEFUNICIONES PROCESOS

CO_PROCESO_DEFUNICION
DE_PROCESO_DEFUNICION
ES_PROCESO_DEFUNICION
CO_TIPO_TRAMITE_DEFUNICION (FK)

RCTR NACIMIENTOS TIPO ACTOR

CO_TIPO_ACTOR_NACIMIENTO
DE_TIPO_ACTOR
ES_TIPO_ACTOR

RCTR NACIMIENTOS CAMPOS

CO_CAMPO_NACIMIENTOS
DE_CAMPO_NACIMIENTOS
CO_TIPO_ACTOR_NACIMIENTO (FK)
DE_CAMPO_ACTUALIZA
IN_CAMBO
ES_CAMPO_NACIMIENTOS

RCTV NACIMIENTOS VARIABLES

NU_ACTA_NACIMIENTO (FK)
NU_SECUENCIA_NACIMIENTOS (FK)
CO_CAMPO_NACIMIENTOS (FK)
DE_VALOR_ANTERIOR
DE_VALOR_NUEVO
DE_IN_VALOR_ANTERIOR
DE_IN_VALOR_NUEVO

RCTV NACIMIENTOS

NU_ACTA_NACIMIENTO (FK)
NU_SECUENCIA_NACIMIENTOS
CO_OFICINA_REGISTRAL (FK)
NU_DNI_REGISTRADOR (FK)
NU_SECUENCIA (FK)
FE_ACTA_NACIMIENTO
CO_DECLARANTES (FK)
CO_PROCESO_NACIMIENTO (FK)
ID_CREA_MOVIMIENTO
FE_CREA_MOVIMIENTO
HO_CREA_MOVIMIENTO
ID_ACT_MOVIMIENTO
FE_ACT_MOVIMIENTO
HO_ACT_MOVIMIENTO

RCTR DEFUNICIONES TIPOS DOC

CO_TIPO_DOC_DEFUNICION
DE_TIPO_DOC_DEFUNICION
IN_LLENAR_NUMERO
IN_LLENAR_FECHA
ES_TIP_DOC_DEF

RCTM ESCANEADO

NU_ACTA
CO_TIPO_ACTA (FK)
CO_LADO_ACTA (FK)
CO_ESTADO_ESCANEO (FK)
IN_ACTA
ID_REGISTRO
FE_REGISTRO
HO_REGISTRO
ID_ACTUALIZA
FE_ACTUALIZA
HO_ACTUALIZA

RCTR NACIMIENTOS PROCESOS
 CO_PROCESO_NACIMIENTO
 DE_PROCESO_NACIMIENTO
 ES_PROCESO_NACIMIENTO
 CO_TIPO_TRAMITE_NACIMIENTO (FK)

RCTV REGISTRADOR CIVIL
 NU_DNI_REGISTRADOR_CIVIL
 CO_OFICINA_DC
 NU_SECUENCIA

RCTO MATRIMONIOS CAMPOS
 CO_CAMPO_MATRIMONIO
 DE_CAMPO_MATRIMONIO
 CO_TIPO_ACTOR_MATRIMONIO (FK)
 IN_CAMBIO
 DE_CAMPO_ACTUALIZA
 ES_CAMPO_MATRIMONIO

RCTR NACIMIENTOS TIPOS DOC
 CO_TIPO_DOC_NACIMIENTO
 DE_TIPO_DOC_NACIMIENTO
 IN_LLENAR_NUMERO
 IN_LLENAR_FECHA
 ES_TIPO_DOC_NACIMIENTO

RCTR NAC COMBINACION ACEPTABLE
 CO_COMBINACION (FK)
 CO_TIPO_DOC_NACIMIENTO (FK)
 IN_PARTICIPACION
 ES_COMBINACION

RCTR NAC COMBINACION DOCUMENTO
 CO_COMBINACION
 ES_COMBINACION

RCTR NACIMIENTOS DECLARANTES
 CO_DECLARANTES
 DE_DECLARANTES
 ES_DECLARANTES

RCTV MATRIMONIOS VARIABLES
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIO (FK)
 CO_CAMPO_MATRIMONIO (FK)
 DE_VALOR_ANTERIOR
 DE_VALOR NUEVO
 DE_IM_VALOR_ANTERIOR
 DE_IM_VALOR NUEVO

RCTV MATRIMONIOS
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIOS
 CO_OFICINA_REGISTRAL (FK)
 NU_DNI_REGISTRADOR (FK)
 NU_SECUENCIA (FK)
 FE_ACTA_MATRIMONIO
 CO_PROCESO_MATRIMONIO (FK)
 ID_CREA_MOVIMIENTO
 FE_CREA_MOVIMIENTO
 HO_ACT_MOVIMIENTO
 FE_ACT_MOVIMIENTO
 HO_ACT_MOVIMIENTO

RCTM MATRIMONIOS ALIQUOTAS
 NU_ACTA_MATRIMONIO (FK)
 CO_TIPO_ACTOR_MATRIMONIO (FK)
 DE_PRIMER_APELLIDO
 DE_SEGUNDO_APELLIDO
 DE_PRE_NOMBRES
 CO_TIPO_DOC_IDENTIDAD
 NU_DOC_IDENTIDAD
 ID_CREA_ACTA_MATRIMONIO
 FE_CREA_ACTA_MATRIMONIO
 HO_CREA_ACTA_MATRIMONIO

RCTV MATRIMONIOS
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIOS
 CO_ESTADO_LINEA (FK)
 INSCRIPCIÓN 2
 FE_CREA_MOVIMIENTO
 HO_CREA_MOVIMIENTO
 ID_CREA_MOVIMIENTO

RCTV MATRIMONIOS DOCUMENTOS
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIO (FK)
 CO_TIPO_DOC_MATRIMONIO (FK)
 CO_TIPO_ACTOR_MATRIMONIO (FK)
 DE_NUMERO_DOCUMENTO
 FE_DOCUMENTO

RCTR MAT COMBINACION DOCUMENTO
 CO_COMBINACION
 ES_COMBINACION

RCTR MAT COMBINACION ACEPTABLE
 CO_COMBINACION (FK)
 CO_TIPO_DOC_MATRIMONIO (FK)
 IN_PARTICIPACION
 ES_COMBINACION

RCTV ACTA RECONOCIMIENTO MAT
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIOS (FK)
 NU_OFICIO
 NU_RESOLUCION
 FE_RESOLUCION
 FE_SENTENCIA
 NU_ESCRITURA
 FE_ESCRITURA
 NU_EXPEDIENTE
 FE_EXPEDIENTE
 CO_TIPO_DOC_IDENTIDAD_FUN (FK)
 NU_DOC_IDENTIDAD_FUN
 DE_PRIMER_APELLIDO_FUN
 DE_SEGUNDO_APELLIDO_FUN
 DE_PRE_NOMBRES_FUN
 BE_JUZGADO

RITX RECIBO DEL SECCO
 NU_RECIBO (FK)
 CO_TIPO_RECIBO (FK)
 CO_TIPO_DOC_IDENTIDAD (FK)
 NU_DOC_IDENTIDAD
 NU_RECIBO
 NU_DGTO_CHEQUEO
 FE_RECIBO
 HO_RECIBO
 CO_CAJERO
 CO_AGENCIA
 IN_RECIBO

RCTM DEFUNICIONES ACTORES
 NU_ACTA_DEFUNCION (FK)
 CO_TIPO_ACTOR_DEFUNCION (FK)
 CO_TIPO_DOC_IDENTIDAD
 NU_DOC_IDENTIDAD
 DE_PRIMER_APELLIDO
 DE_SEGUNDO_APELLIDO
 DE_PRE_NOMBRES
 ID_CREA_ACTOR_DEFUNCION
 FE_CREA_ACTOR_DEFUNCION
 HO_CREA_ACTOR_DEFUNCION

RCTM MATRIMONIOS
 NU_ACTA_MATRIMONIO
 FE_INSCRIPCION
 CO_NACIONALIDAD_CTYENTE2
 CO_NATURAL_DPTO_CTYENTE2
 CO_NATURAL_PROV_CTYENTE2
 CO_TIPO_DOC_IDENTIDAD_CTYENTE1
 NU_DOC_IDENTIDAD_CTYENTE1
 DE_PRIMER_APELLIDO_CTYENTE1
 DE_SEGUNDO_APELLIDO_CTYENTE1
 DE_PRE_NOMBRES_CTYENTE1
 CO_TIPO_DOC_IDENTIDAD_CTYENTE2
 NU_DOC_IDENTIDAD_CTYENTE2
 DE_PRIMER_APELLIDO_CTYENTE2
 DE_SEGUNDO_APELLIDO_CTYENTE2
 DE_PRE_NOMBRES_CTYENTE2
 CO_CELBRADO_DPTO
 FE_DE_CELBRACION
 HO_CELBRACION
 CO_ESTADO_ACTA_MATRIMONIO
 ID_CREA_ACTA_MATRIMONIO
 FE_CREA_ACTA_MATRIMONIO
 HO_CREA_ACTA_MATRIMONIO
 ID_ACTA_MATRIMONIO
 FE_ACTA_MATRIMONIO
 HO_ACTA_MATRIMONIO
 ES_ACTA_LIBRO
 DE_ACTA_LIBRO
 DE_ACTA_FOLIO
 CO_TIPO_DOC_IDENTIDAD_RGT_H
 NU_DOC_IDENTIDAD_RGT_H
 DE_PRIMER_APELLIDO_RGT_H
 DE_SEGUNDO_APELLIDO_RGT_H
 DE_PRE_NOMBRES_RGT_H

RCTM DEFUNICIONES DECISION
 CO_PROCESO_DEFUNCION (FK)
 CO_TIPO_FALLECIMIENTO (FK)
 CO_TIPO_DOC_IDENTIDAD (FK)
 CO_COMBINACION (FK)
 IN_DIFUNTO_PRIMER_APELLIDO
 IN_DIFUNTO_SEGUNDO_APELLIDO
 IN_DIFUNTO_PRE_NOMBRES
 IN_DIFUNTO_SEXO
 IN_DIFUNTO_EDAD
 IN_DIFUNTO_TIPO_DOC_IDENTIDAD
 IN_DIFUNTO_DOC_IDENTIDAD
 IN_DIFUNTO_NATURAL_DPTO
 IN_DIFUNTO_NATURAL_PROV
 IN_DIFUNTO_NATURAL_DIST
 IN_DIFUNTO_NATURAL_LOCA
 IN_DIFUNTO_ESTADO_CIVIL
 IN_CONYUGE_PRIMER_APELLIDO
 IN_CONYUGE_SEGUNDO_APELLIDO
 IN_CONYUGE_PRE_NOMBRES
 IN_CONYUGE_TIPO_DOC_IDENTIDAD
 IN_CONYUGE_DOC_IDENTIDAD
 IN_DIFUNTO_DCM_DPTO
 IN_DIFUNTO_DCM_DIST
 IN_DIFUNTO_DCM_LOCA
 IN_DIFUNTO_DIRECCION
 IN_CO_FALLECIMIENTO_DPTO
 IN_CO_FALLECIMIENTO_PROV
 IN_CO_FALLECIMIENTO_DIST
 IN_CO_FALLECIMIENTO_LOCA
 IN_DE_FALLECIMIENTO_HORA
 IN_DE_FALLECIMIENTO_DPTO
 IN_DE_FALLECIMIENTO_PROV
 IN_DE_FALLECIMIENTO_DIST
 IN_DE_FALLECIMIENTO_LOCA
 IN_FALLECIMIENTO_HORA
 IN_FALLECIMIENTO_FECHA
 IN_FALLECIMIENTO_TIPO_LUGAR
 IN_FALLECIMIENTO_NOMBRE_LUGAR
 IN_FALLECIMIENTO_DIREC_LUGAR
 IN_MADRE_PRIMER_APELLIDO
 IN_MADRE_SEGUNDO_APELLIDO
 IN_MADRE_PRE_NOMBRES
 IN_MADRE_EDAD
 IN_MADRE_NACIONALIDAD
 IN_MADRE_NATURAL_DPTO
 IN_MADRE_NATURAL_PROV
 IN_MADRE_NATURAL_DIST
 IN_MADRE_NATURAL_LOCA
 IN_MADRE_DOM_DPTO
 IN_MADRE_DOM_PROV
 IN_MADRE_DOM_DIST
 IN_MADRE_DOM_LOCA
 IN_MADRE_DIRECCION
 IN_MADRE_TIPO_DOC_IDENTIDAD
 IN_MADRE_DOC_IDENTIDAD
 IN_MADRE_VINCULO_FAMILIAR
 IN_PADRE_PRIMER_APELLIDO
 IN_PADRE_SEGUNDO_APELLIDO
 IN_PADRE_PRE_NOMBRES
 IN_PADRE_EDAD
 IN_PADRE_NACIONALIDAD
 IN_PADRE_NATURAL_DPTO
 IN_PADRE_NATURAL_PROV
 IN_PADRE_NATURAL_DIST
 IN_PADRE_NATURAL_LOCA
 IN_PADRE_DOM_DPTO
 IN_PADRE_DOM_PROV
 IN_PADRE_DOM_DIST
 IN_PADRE_DOM_LOCA
 IN_PADRE_DIRECCION
 IN_PADRE_TIPO_DOC_IDENTIDAD
 IN_PADRE_DOC_IDENTIDAD
 IN_PADRE_VINCULO_FAMILIAR
 IN_DECLARANTE_PRIMER_APELLIDO
 IN_DECLARANTE_SEGUNDO_APELLIDO
 IN_DECLARANTE_PRE_NOMBRES
 IN_DECLARANTE_EDAD
 IN_DECLARANTE_NACIONALIDAD
 IN_DECLARANTE_NATURAL_DPTO
 IN_DECLARANTE_NATURAL_PROV
 IN_DECLARANTE_NATURAL_DIST
 IN_DECLARANTE_NATURAL_LOCA
 IN_DECLARANTE_DOM_DPTO
 IN_DECLARANTE_DOM_PROV
 IN_DECLARANTE_DOM_DIST
 IN_DECLARANTE_DOM_LOCA
 IN_DECLARANTE_DIRECCION
 IN_DECLARANTE_TIPO_DOC_IDENTIDAD
 IN_DECLARANTE_DOC_IDENTIDAD
 IN_DECLARANTE_VINCULO_FAMILIAR
 IN_REGISTRADOR
 IN_DISEÑADOR
 IN_MEDICO_OJE_CERTIFICA_FALLE
 IN_MOTIVO_FALLECIMIENTO
 IN_ESTADO_DELACTA
 IN_AÑO
 IN_LIBRO
 IN_FOLIO
 IN_INSCRIPCION_FECHA
 IN_RGT_TIPO_DOC_IDENTIDAD
 IN_RGT_DOC_IDENTIDAD
 IN_RGT_PRIMER_APELLIDO
 IN_RGT_SEGUNDO_APELLIDO
 IN_RGT_PRE_NOMBRES

RCTV MATRIMONIOS VARIABLES
 CO_CAMPO_MATRIMONIO (FK)
 NU_ACTA_MATRIMONIO (FK)
 NU_SECUENCIA_MATRIMONIOS (FK)
 DE_VALOR_ANTERIOR
 DE_VALOR NUEVO
 DE_IM_VALOR_ANTERIOR
 DE_IM_VALOR NUEVO

RCTM DEFUNICIONES
 NU_ACTA_DEFUNCION
 FE_INSCRIPCION
 DE_PRIMER_APELLIDO
 DE_SEGUNDO_APELLIDO
 DE_PRE_NOMBRES
 CO_SEXO
 NU_EDAD
 CO_NACIONALIDAD
 CO_NATURAL_DPTO
 CO_NATURAL_PROV
 CO_TIPO_DOC_IDENTIDAD
 NU_DOC_IDENTIDAD
 FE_FALLECIMIENTO
 HO_FALLECIMIENTO
 DE_ACTA_AÑO
 DE_ACTA_LIBRO
 DE_ACTA_FOLIO
 CO_TIPO_DOC_IDENTIDAD_RGT_H
 NU_DOC_IDENTIDAD_RGT_H
 DE_PRIMER_APELLIDO_RGT_H
 DE_SEGUNDO_APELLIDO_RGT_H
 DE_PRE_NOMBRES_RGT_H
 CO_ESTADO_ACTA_DEFUNCION
 ID_CREA_ACTA_DEFUNCION
 FE_CREA_ACTA_DEFUNCION
 HO_CREA_ACTA_DEFUNCION
 ID_ACTA_DEFUNCION
 FE_ACTA_DEFUNCION
 HO_ACTA_DEFUNCION

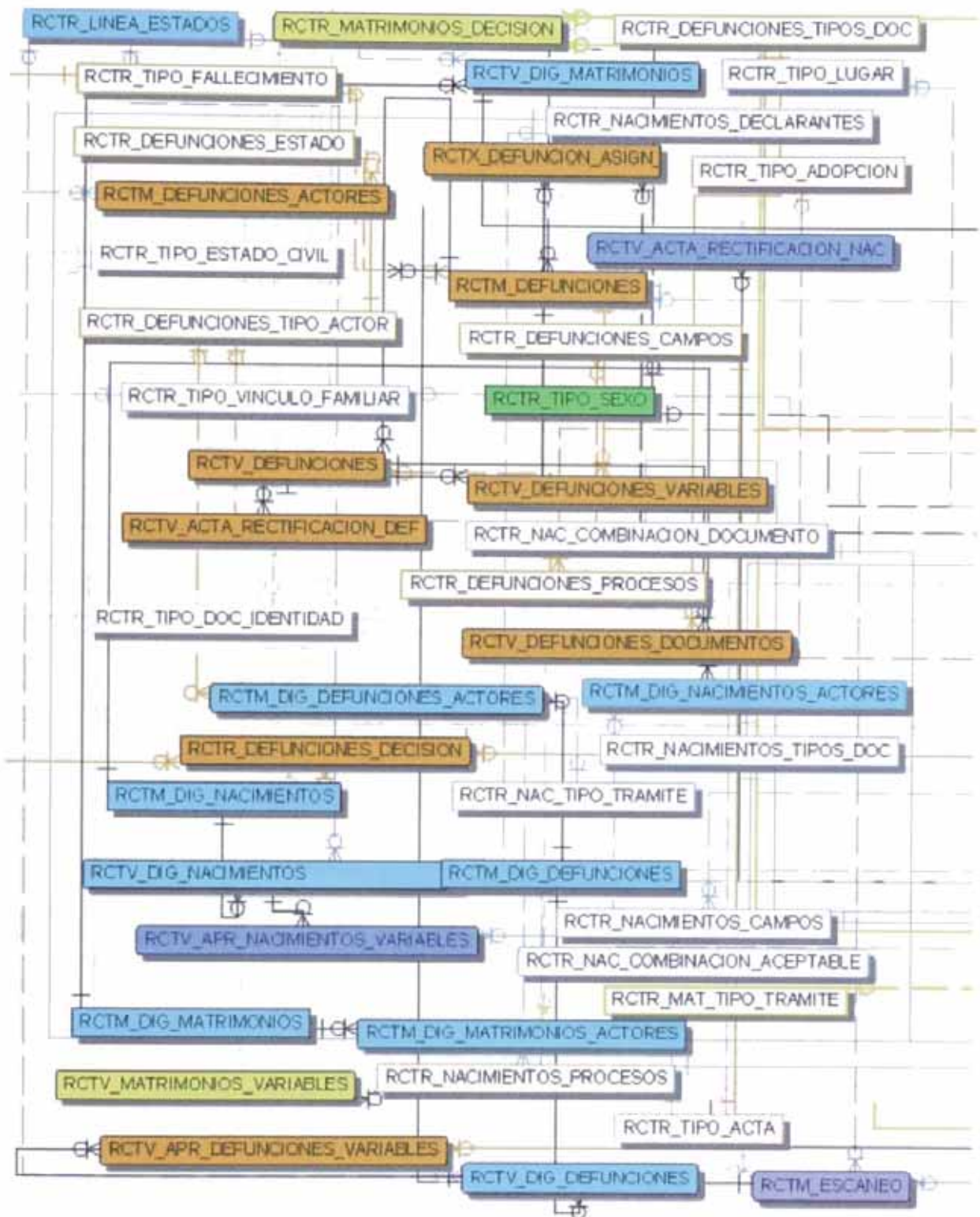
RCTM NACIMIENTOS ACTORES
 NU_ACTA_NACIMIENTO (FK)
 CO_TIPO_ACTOR_NACIMIENTO (FK)
 DE_PRIMER_APELLIDO
 DE_SEGUNDO_APELLIDO
 DE_PRE_NOMBRES
 CO_TIPO_DOC_IDENTIDAD
 NU_DOC_IDENTIDAD
 ES_ACTOR
 ID_CREA_ACTOR_NACIMIENTO
 FE_CREA_ACTOR_NACIMIENTO
 HO_CREA_ACTOR_NACIMIENTO

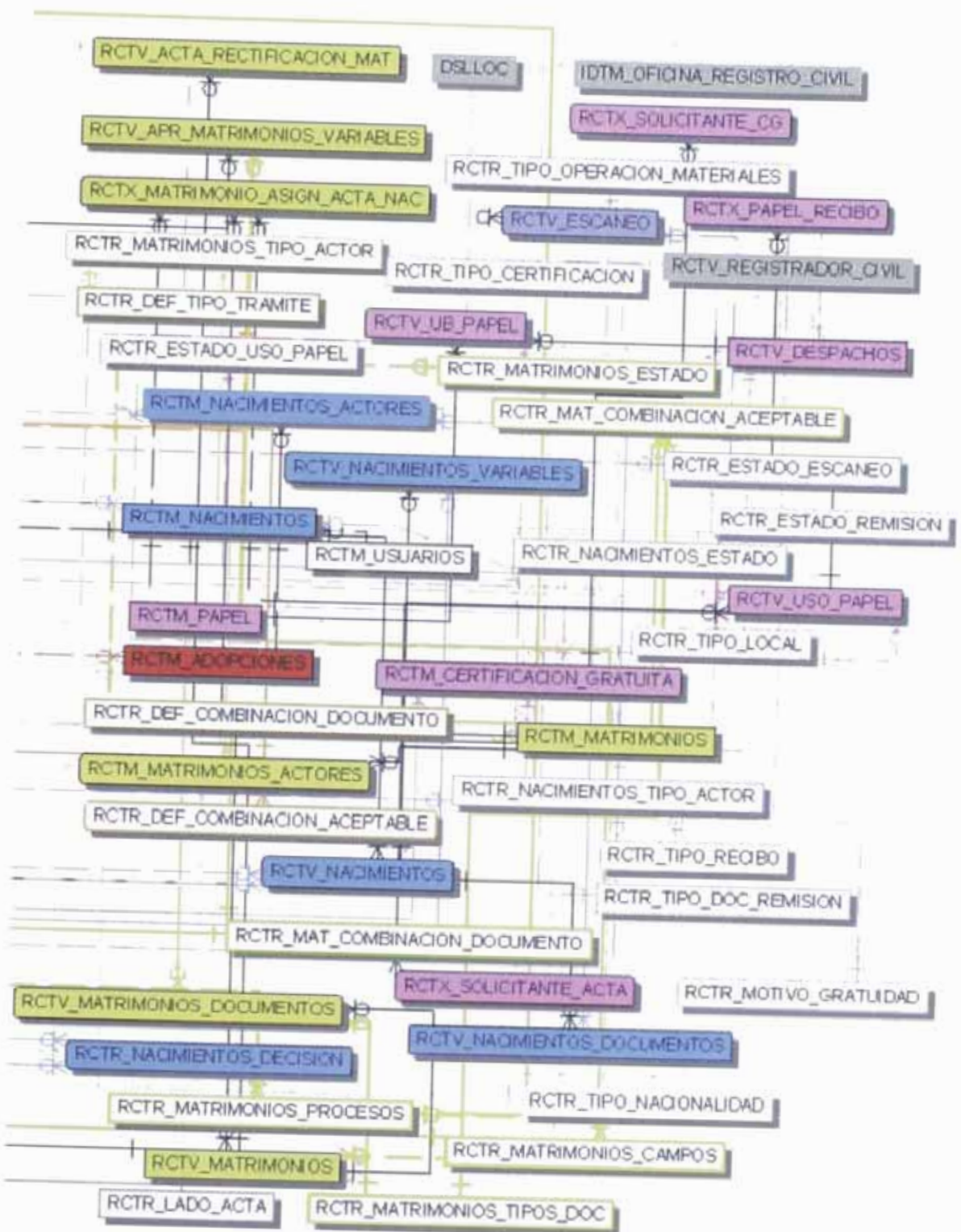
RCTV DEFUNICIONES
 NU_ACTA_DEFUNCION (FK)
 NU_SECUENCIA_DEFUNCION
 CO_ESTADO_LINEA (FK)
 INSCRIPCIÓN 2
 FE_CREA_MOVIMIENTO
 HO_CREA_MOVIMIENTO
 ID_CREA_MOVIMIENTO

RCTV DEFUNICIONES VARIABLES
 NU_ACTA_DEFUNCION (FK)
 NU_SECUENCIA_DEFUNCION (FK)
 CO_CAMPO_DEFUNCION (FK)
 DE_VALOR_ANTERIOR
 DE_VALOR NUEVO
 DE_IM_VALOR_ANTERIOR
 DE_IM_VALOR NUEVO

RCTM NACIMIENTOS
 NU_ACTA_NACIMIENTO
 FE_INSCRIPCION
 DE_PRIMER_APELLIDO
 DE_SEGUNDO_APELLIDO
 DE_PRE_NOMBRES
 CO_SEXO
 CO_NACIDO_NATURAL_DIST
 CO_NACIDO_NATURAL_LOCA
 DE_NACIDO_NATURAL_DPTO
 FE_NACIMIENTO
 HO_NACIMIENTO
 DE_ACTA_AÑO
 DE_ACTA_LIBRO
 DE_ACTA_FOLIO
 DE_PRIMER_APELLIDO_RGT_H
 DE_SEGUNDO_APELLIDO_RGT_H
 DE_PRE_NOMBRES_RGT_H
 CO_TIPO_DOC_IDENTIDAD_RGT_H
 NU_DOC_IDENTIDAD_RGT_H
 CO_ESTADO_ACTA_NACIMIENTO
 ID_CREA_ACTA_NACIMIENTO
 FE_CREA_ACTA_NACIMIENTO
 HO_CREA_ACTA_NACIMIENTO
 ID_ACTA_NACIMIENTO
 FE_ACTA_NACIMIENTO
 HO_ACTA_NACIMIENTO

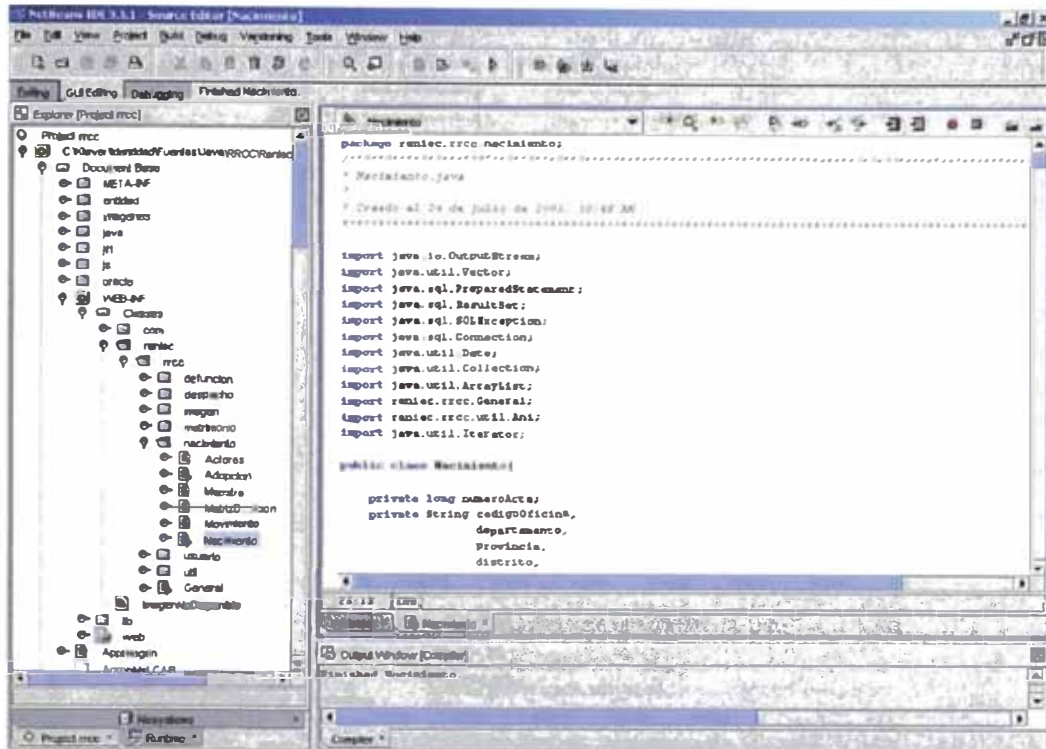
4.6.2 DIAGRAMA ENTIDAD RELACIÓN



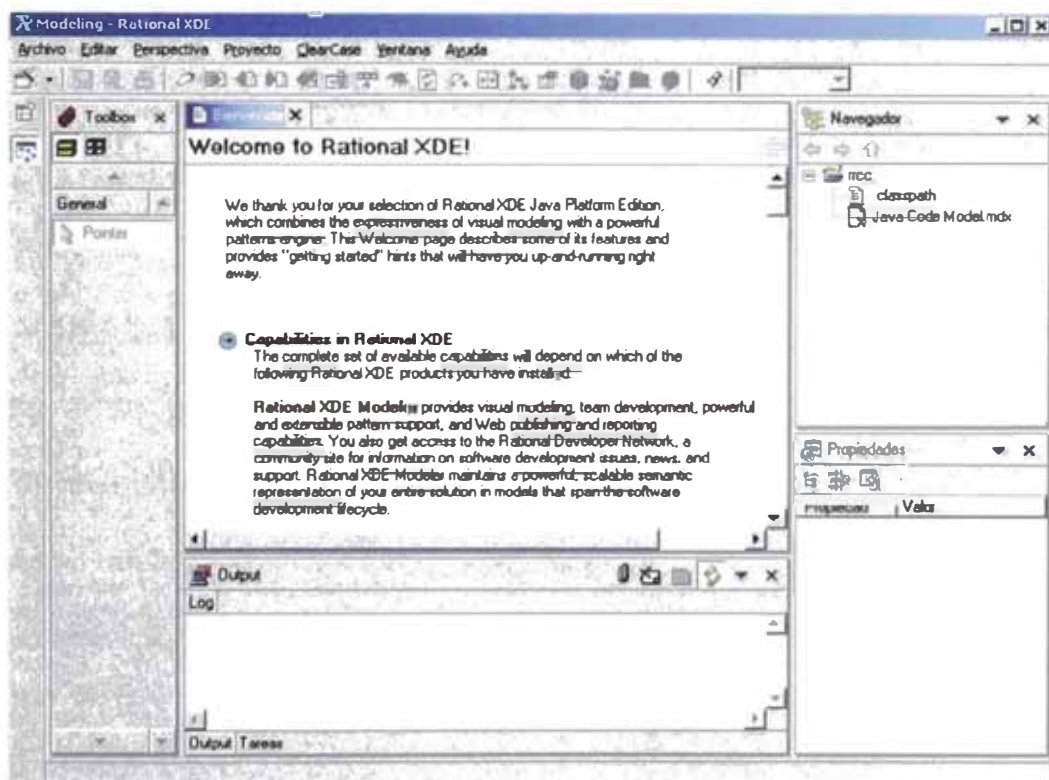


4.7 UTILITARIOS

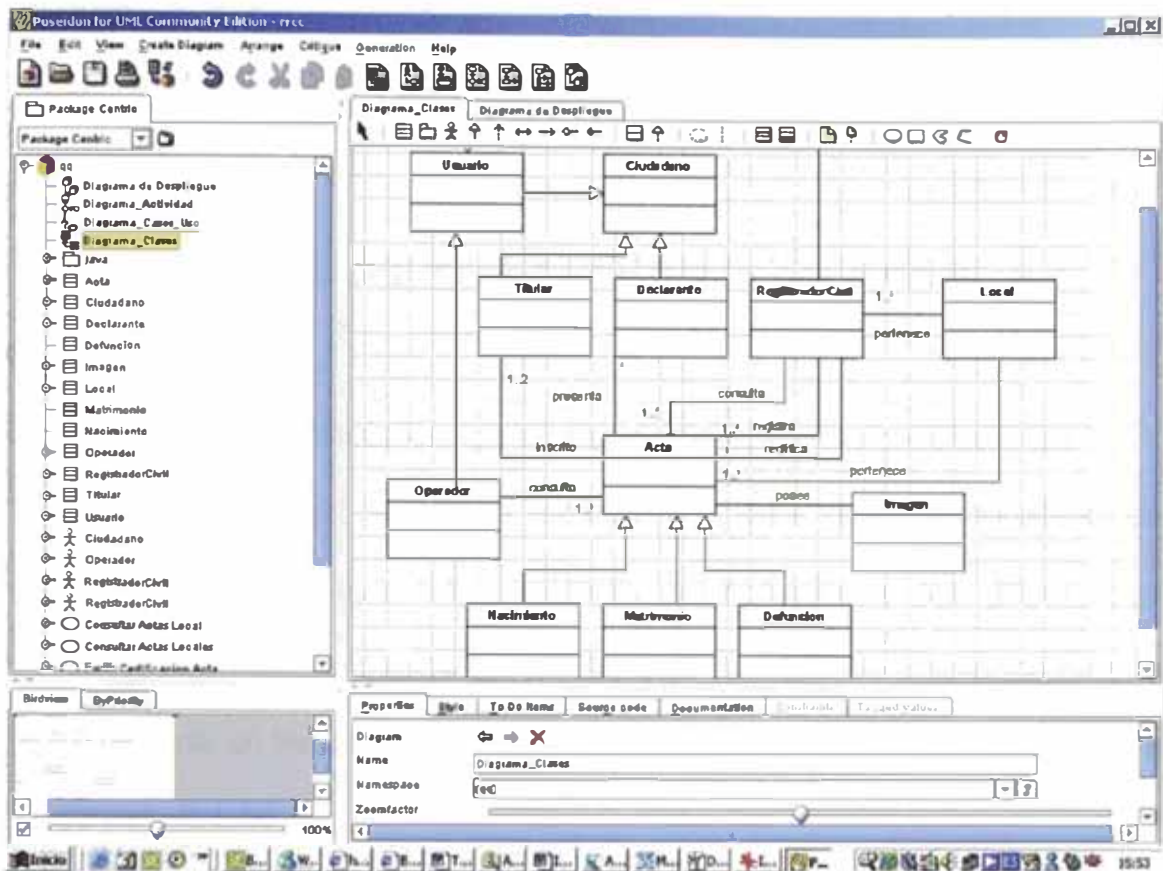
a.-) NETBEANS



b.-) RATIONAL XDE



c.-) POSEIDON



CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

De acuerdo a este trabajo hemos llegado a las siguientes conclusiones:

1. Con este proceso se puede mostrar una guía para el modelado de requisitos en la que los diagramas de actividad y las clases que representan los conceptos del dominio, se obtienen de una forma directa a partir de un modelado del negocio centrado en el empleo de los casos de uso de UML.
2. Con esta guía, el modelador dispone de una forma sistemática de identificar y organizar los casos de uso, e identificar y definir las clases que conforman el modelo conceptual.
3. Los procesos de negocio de la empresa son identificados a partir de los tipos de objetivos propuestos y se describen representando el flujo de actividades mediante un diagrama de casos de uso UML, de modo que las actividades del proceso se obtienen de los casos de uso, siendo estos últimos organizados mediante una jerarquía de niveles.
4. Las clases del modelo conceptual se obtienen a partir de las informaciones que fluyen entre actividades.

5. El modelado de casos de uso se realiza de forma paralela al modelado conceptual para “no obtener casos de uso inútiles”.
6. Pasar una aplicación de proceso de desarrollo de software basado en UML a una aplicación de gestión concreta es viable. Y no sólo es viable, sino que, además, nos ha enseñado que se pueden crear productos de calidad para aplicaciones de un tamaño considerable. Creemos, además, que el proceso presentado puede ir mejorándose con cada nuevo proyecto al que se aplique.
7. El mantenimiento del sistema instalado a partir de nuestra aplicación del meta-proceso es fácil, reduciendo la tasa de fallos, con correcta comprensión de los requerimientos del usuario final, una cómoda adaptación a los cambios frecuentes en el dominio del problema y se dispone de una documentación que trasciende al proyecto pues como se ha explicado, la documentación es generada en tiempo real a partir de los esquemas existentes.
8. El modelo de tres capas permite definir y distinguir las actividades tanto del lado del cliente, del lado de la Base de Datos y del lado de las reglas del negocio.
9. El manejo de patrones de diseño favorece grandemente al desarrollo rápido de sistemas como lo demuestra el uso del patrón Facade para acceder a las entidades de manera indirecta por medio de una sesión anteponiendo una interfaz sencilla para ingresar a subsistemas complejos.

RECOMENDACIONES

A partir de la experiencia que nos ha dejado el trabajo de desarrollar un nuevo meta-proceso aplicando una instanciación (proceso) para un proyecto específico, podemos dar las siguientes recomendaciones:

1. El ciclo de vida para el desarrollo de sistemas debe ser iterativo e incremental por las evidentes ventajas que tiene ante los otros ciclos de vida tradicionales de sistemas (manejo de cambios, manejo de riesgos, controles de avance, etc.).
2. Es necesario contar con un arquitecto de sistemas y además con un área para el desarrollo de componentes para manejar las reglas del negocio que sirvan como repositorio de componentes en los distintos proyectos de sistemas.
3. Se deben definir de manera clara los roles en los grupos de trabajo dentro del área de desarrollo de sistemas.
4. Tomar en consideración el nuevo meta-proceso presentado para mejorarlo y utilizarlo ya que ha podido ser de gran ayuda en el manejo de un proyecto.
5. Manejar patrones de diseño y estándares de programación para un mejor desarrollo en términos de duración y seguimiento.
6. Manejar de requerimientos de los usuarios a través de los casos de uso pues ello permite un entendimiento mayor tanto por parte de los usuarios como por parte de los desarrolladores de sistemas.
7. Revisar los conceptos de la orientación a objetos, notaciones, procesos y métodos de trabajo. Un mejor entendimiento de estos conceptos ayudará

a reconocer de manera más amplia las posibilidades y ventajas que se van abriendo camino.

8. Homogenizar el lenguaje de las personas que conforman el área de desarrollo de sistemas.
9. Contar con un apropiado manejo de versiones y establecer estrictas políticas de seguridad.
10. Crear una nueva área que se encargue únicamente a realizar pruebas funcionales, que eviten la presencia de errores posibles al pasar los sistemas a producción.
11. Evitar en lo posible el manejo del papel, dejando de lado informes o documentos de gran tamaño por vistas que puedan ser generados en cualquier instante a partir de los esquemas existentes, los cuales siempre se mantendrán actualizados con respecto al desarrollo.
12. Enfocar la atención en los aspectos importantes antes que en aquellos que son urgentes dentro del desarrollo de sistemas. No es posible que el desarrollo se maneje por lo inmediato, pues ello impide asimilar los verdaderos avances dentro del campo de sistemas, además de generar soluciones poco robustas que no responden a la problemática de un negocio.

GLOSARIO

Este glosario se apoya sobre aquel desarrollado en inglés para la versión de Abril del 2001 del SPEM, el cual es el estándar de representación de los procesos.

Introducción

Principio o preámbulo. Breve descripción de un tema específico

Concepto

Idea que se concibe para el desarrollo del meta-proceso.

Actividad

Definición de un trabajo describiendo aquello que es producido dentro del marco de un Rol. Las actividades son el elemento principal de un trabajo (SPEM).

Componente de proceso

Un Componente de Proceso es un reagrupamiento coherente de Elementos del modelo organizados para obtener una perspectiva diferente, tal como la Disciplina, por ejemplo la prueba, o la producción de ciertos Productos de Trabajo, por ejemplo la Gestión de Exigencias (SPEM).

Ciclo

Un pasaje completo por las fases del ciclo de vida (RUP).

Ciclo de Vida

Un ciclo de vida para un proceso es definido como una secuencia de fases para conseguir un objetivo preciso. El ciclo

de vida define el proceso que será aplicado sobre un proyecto dado (RUP).

Definición de Trabajo

Este es un Elemento de modelo de un proceso que describe la ejecución, las operaciones realizadas, y las transformaciones operadas sobre los Productos de Trabajo dentro del marco de un Rol. En medio de las definiciones de trabajo se encuentran las Actividades, las Iteraciones, las fases y el Ciclo de Vida (SPEM).

Dependencia

Una Dependencia es una relación particular, específica para el proceso, que sirve de unión entre los Elementos de Modelo (SPEM).

Disciplina

Una Disciplina es una unidad del proceso, organizada de acuerdo a una perspectiva característica de Ingeniería de Sistemas: Gestión de Configuración, Análisis y Concepción, Gestión del Proyecto, etc (SPEM).

Elemento de Modelo

Es un elemento que describe un aspecto de un proceso (SPEM).

Etapa

Una Etapa es una Definición de Trabajo atómico, para un pequeño objetivo, utilizada para la descomposición de Actividades. Las Actividades son un conjunto de etapas parcialmente ordenadas (SPEM).

Ejecutor de Proceso

Elemento del modelo que describe los roles, las responsabilidades y las competencias de un individuo suministrando Actividades dentro de Proyecto, responsable de ciertos Productos de Trabajo (SPEM).

Guía

La Guía es un Elemento de Modelo asociado a los elementos principales de definición de procesos, que contienen descripciones suplementarias tales como técnicas, métodos, perfiles UML, procedimientos, estándares, planes tipo, ejemplos, productos de trabajo, definiciones, etc (SPEM).

Guía de Producto

Es una Guía asociada a un producto, que provee de reglas y recomendaciones prácticas sobre la manera de crear y organizar el producto.

Guía de Trabajo

Es una Guía asociada a una Actividad, que provee de técnicas útiles para la realización de la Actividad (RUP).

Guía de Utilitario

Es una Guía que provee de una descripción detallada sobre la manera de realizar una actividad o las etapas de la actividad, con la ayuda de un utilitario.

Iteración

Una Iteración es una definición de trabajo, que representa un conjunto de Actividades, dirigiéndose al desarrollo de una porción del sistema que termina por el suministro (interno o externo) de un producto informático (SPEM).

Lista de Control

Es una Guía que contiene una lista de puntos a controlar para evaluar la calidad de un producto de trabajo (RUP).

Fase

Definición de alto nivel de un trabajo, que termina por una cadena (SPEM).

Paso

Se define como un Elemento de Modelo atómico y granular utilizado para descomponer Actividades. Las Actividades son parcialmente ordenadas como un conjunto de Pasos (SPEM).

Plan Tipo

Es una Guía que provee un documento genérico en un formato estándar para un Producto de Trabajo particular.

Proceso

Un Proceso consiste en la descripción completa de una metodología aplicada a la Ingeniería de Sistemas, en términos de Roles, de definición de Trabajo, de suministro de Productos de Trabajo (o Artefactos) y de Guías asociadas (SPEM).

Producto de Trabajo

Un Producto de Trabajo es un constituyente informativo o una entidad física (documento, modelo UML, código fuente, plan, etc.) producida o utilizada por una Actividad del proceso de Ingeniería de Sistemas (SPEM). Llamado “Artefacto” en el RUP.

Rol del Proceso

Un Rol del Proceso es un Elemento de modelo que describe al propietario de una Definición de Trabajo. Un Rol del Proceso es utilizado para las Definiciones de Trabajo que no pueden ser asociados con un Ejecutor de Proceso, como el Ciclo de Vida o la Fase (SPEM).

Página JSP

Archivo plano que a través de una traducción se convierte en un servicio (servlet) para finalmente ser convertido en un archivo plano HTML.

BIBLIOGRAFÍA

1. Began to go through the RUP workflows. Dr. Hawker, CS600 Spring 2002 Lecture Notes:
2. Rational Unified Process: Product Information
<http://www.rational.com/products/rup/prodinfo.jsp>
3. Software Process Engineering Metamodel Specification
4. CHAOS, The Standish Group International, Inc., Dennis, MA, 1994, 1997.
5. The Mythical Man-Month—Essays on Software Engineering 2nd ed. Frederick P. Brooks, Jr. 1995. Reading, MA, Addison Wesley Longman. Dorfman, M. and R. Thayer, *Software Engineering*, IEEE Computer Society Press, Los Alamitos, CA, 1997 pp. 79.
6. Dorfman, M. and R. Thayer, *Software Engineering*, IEEE Computer Society Press. Los Alamitos, CA, 1997 pp. 80.
7. *Traceability Strategies for Managing Requirements with Use Cases*. Spence, Ian and Leslee Probasco, White Paper, Rational Software Corporation, 1998.
8. UML User Guide. G. Booch, J. Rumbaugh, and I. Jacobson, 1998. Addison-Wesley Longman.
9. OMG Unified Modeling Language Specification, Version 1.4. Rational Software Corporation, 18880 Homestead Road, Cupertino, CA 95014, and Object Management Group, Inc., 492 Old Connecticut Path, Framingham, MA 01701

- 10.** Analysis and Design with Applications, 2nd edition. Grady Booch
1993. Object-Oriented Redwood City, CA: The Benjamin/Cummings
Publishing Company
- 11.** Rational Unified Process™, Rational Software Corporation, Cupertino,
CA, 1999.
- 12.** OMT Insights. James Rumbaugh 1996. New York: SIGS Books.
- 13.** Managing Software Requirements - A Unified Approach, Leffingwell,
Dean and Don Widrig, Addison- Wesley, 2000.
- 14.** Software Architecture in Practice. Len Bass, Paul Clements, and Rick
Kazman 1998. Addison Wesley Longman.
- 15.** Software Reuse -Architecture, Process and Organization for
Business Success. Ivar Jacobson, Martin Griss and Patrik Jonsson,
1997. Addison Wesley Longman.
- 16.** Adaptive Software Development: A Collaborative Approach to
Managing Complex Systems. James A. Highsmith 2000. Dorset
House.
- 17.** More Process Patterns: Delivering Large-Scale Systems Using Object
Technology. Scott W. Ambler 1999. New York, NY: SIGS
Books/Cambridge University Press.
- 18.** Design Considerations: From Client Server Applications to e-business
Applications. International Business Machines Corporation 1999.
<http://www.redbooks.ibm.com/abstracts/sg245503.html>