

03063



UNIVERSIDAD NACIONAL  
AVENIDA DE  
MEXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“ESTRATEGIA DE IMPLEMENTACIÓN DEL  
PROCESO UNIFICADO DE RACIONAL (RUP)  
PARA DAR MANTENIMIENTO AL SISTEMA  
INTEGRAL DE INFORMACIÓN DEL  
REGISTRO FEDERAL DE ELECTORES”**

**T E S I S**

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA  
(COMPUTACIÓN)**

**P R E S E N T A:**

**SERGIO MARTÍNEZ GONZÁLEZ**

DIRECTORA DE LA TESIS: Dra. Hanna Oktaba.

MÉXICO, D.F.

Noviembre 2005.

**M: 350675**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

***Dedicado a:  
Julio César***

*Un agradecimiento muy especial  
a la Dra. Hanna Oktaba, por ser  
una gran maestra y una gran  
persona.*

*Un agradecimiento muy especial  
al Dr. Alberto Alonso y Coria, por  
brindarme la oportunidad y la  
confianza de ser parte de su  
equipo de trabajo.*

# CONTENIDO

INTRODUCCIÓN.....	3
OBJETIVOS.....	5
<b>CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>7</b>
1.1 EL PROGRAMA DE MODERNIZACIÓN TECNOLÓGICA Y OPERATIVA.....	9
1.1.1 Fase de definición e inicio.....	11
1.1.2 Fase de integración.....	13
1.1.3 Fase de consolidación.....	15
1.1.4 Fase de evolución y transición.....	16
1.2 EL SISTEMA INTEGRAL DE INFORMACIÓN DEL REGISTRO FEDERAL DE ELECTORES.....	17
1.3 LA PROBLEMÁTICA EN TORNO AL DESARROLLO DEL SIIRFE.....	25
1.3.1 El alcance de la primera versión del SIIRFE.....	26
1.3.2 El alcance de la segunda versión del SIIRFE.....	29
1.3.3 La necesidad de los trabajos de mantenimiento del SIIRFE.....	30
1.3.4 La necesidad de implementar un proceso de ingeniería.....	32
<b>CAPÍTULO 2. FUNDAMENTOS DEL PROCESO UNIFICADO DE RATIONAL.....</b>	<b>35</b>
2.1 EL PROCESO UNIFICADO DE RATIONAL COMO UN PROCESO DE INGENIERÍA DE SOFTWARE.....	37
2.2 LAS MEJORES PRÁCTICAS EN EL DESARROLLO DE SOFTWARE.....	39
2.2.1 Desarrollo iterativo e incremental.....	39
2.2.2 Administración de requerimientos y proceso dirigido por casos de uso.....	41
2.2.3 Proceso centrado en la arquitectura.....	44
2.2.4 Modelado Visual.....	45
2.2.5 Verificación continua de la calidad.....	46
2.2.6 Administración de los cambios.....	47
2.3 DESCRIPCIÓN DE LA ESTRUCTURA DINÁMICA DEL PROCESO.....	48
2.3.1 Fase de Inicio.....	49
2.3.2 Fase de Elaboración.....	50
2.3.3 Fase de Construcción.....	52
2.3.4 Fase de Transición.....	53
2.4 DESCRIPCIÓN DE LA ESTRUCTURA ESTÁTICA DEL PROCESO.....	54
2.5 DISCIPLINAS DEL PROCESO.....	57
2.5.1 Administración del proyecto.....	57
2.5.2 Modelado del negocio.....	59
2.5.3 Requerimientos.....	59
2.5.4 Análisis y diseño.....	61
2.5.5 Implementación.....	62
2.5.6 Pruebas.....	62
2.5.7 Administración de la configuración y cambios.....	63
2.5.8 Ambiente.....	65
2.5.9 Despliegue.....	66
<b>CAPÍTULO 3. ESTRATEGIA DE IMPLEMENTACIÓN DEL PROCESO.....</b>	<b>67</b>
3.1 CONTINUIDAD AL DESARROLLO Y MANTENIMIENTO DEL SIIRFE.....	69
3.2 PLANEACIÓN DE UN DESARROLLO ITERATIVO E INCREMENTAL.....	72
3.3 ETAPA DE ESTABILIZACIÓN.....	77
3.4 ETAPA DE CONSOLIDACIÓN.....	94
3.5 ETAPA DE TERMINACIÓN DE LA PRIMERA FASE DE DESARROLLO.....	105
3.6 JUSTIFICACIÓN DE LA ESTRATEGIA.....	117

<b>CAPÍTULO 4. IMPACTOS EN LA ORGANIZACIÓN .....</b>	<b>121</b>
4.1 ADECUACIÓN DE FUNCIONES EN LA ORGANIZACIÓN .....	123
4.2 CONSIDERACIONES EN LA IMPLEMENTACIÓN DE UN PROCESO DE INGENIERÍA DE SOFTWARE.....	126
<b>CAPÍTULO 5. CONCLUSIONES Y TRABAJOS A FUTURO .....</b>	<b>129</b>
<b>APÉNDICE A. ROLES DEFINIDOS EN RUP .....</b>	<b>133</b>
<b>APÉNDICE B. ARTEFACTOS DEFINIDOS EN RUP .....</b>	<b>139</b>
<b>APÉNDICE C. DESCRIPCIÓN DE UNA ARQUITECTURA BASADA EN J2EE .....</b>	<b>145</b>
<b>APÉNDICE D. GLOSARIO.....</b>	<b>153</b>
<b>BIBLIOGRAFÍA.....</b>	<b>161</b>

## INTRODUCCIÓN

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. En las pasadas cuatro décadas, el software ha pasado de ser una resolución de problemas especializada y una herramienta de análisis de información, a ser una industria por si misma. Pero la temprana cultura e historia de la “programación” ha creado un conjunto de problemas que persisten todavía hoy.

El software se ha convertido en un factor que limita la evolución de los sistemas informáticos. El software se compone de programas, datos y documentos. Cada uno de estos elementos compone una configuración que se crea como parte del proceso de la ingeniería del software. El intento de la ingeniería del software es proporcionar un marco de trabajo para construir software con mayor calidad.

El problema del tiempo en el desarrollo de sistemas informáticos es importante. En varios estudios realizados, se indica que alrededor de dos tercios de todos los proyectos superan ampliamente sus estimaciones [12]. Los grandes proyectos se retrasan en su fecha de entrega entre un 25 y un 50 por ciento de su duración [6], y el retraso medio se incrementa con el tamaño del proyecto.

Los proyectos encomendados a la Dirección de Desarrollo de Sistemas en el Registro Federal de Electores [3] no son la excepción, y por lo regular se desfasan en tiempo. La estructura organizacional de dicha área se basa en cuatro subdirecciones con nueve jefaturas de departamento, dando un plantilla de aproximadamente cincuenta personas, donde los proyectos se reparten por subdirección.

La problemática radica en la falta de un adecuado control y seguimiento a los proyectos; el personal realiza de todo tipo de actividades, desde programar, probar y mantener su propio código, hasta capacitar, investigar y realizar todas las actividades de gestión involucradas.

Actualmente, el Registro Federal de Electores lleva a cabo un Programa de Modernización Tecnológica y Operativa que hace viable la adecuación de las funciones de cada una de las áreas que lo conforman. Así pues, para la dirección de sistemas, se pretende mejorar el proceso de desarrollo del software, redefiniendo las funciones de cada puesto, especializando al personal y haciendo uso de las herramientas que permitan lograrlo de una manera más eficiente.

A su vez, como parte del Programa de Modernización Tecnológica y Operativa, se concibió el Sistema Integral de Información del Registro Federal de Electores como el software que daría cabida a la sistematización de todos los procesos que se llevan a cabo en el Registro Federal de Electores, alineados a los cambios previstos en ese mismo programa.

Actualmente, se tiene en producción una primera versión de dicho sistema, y es necesario comenzar con los trabajos de mantenimiento y de desarrollo de funcionalidades adicionales. El objetivo del presente trabajo es presentar una estrategia de tipo incremental para dar

continuidad al desarrollo y mantenimiento del sistema, con lo que se pretende, entre otras cosas, reducir los riesgos que conlleva el mantenimiento de un sistema de tal magnitud.

En particular, se propone una forma incremental o por etapas para implementar un proceso de ingeniería de software basado en el Proceso Unificado de Rational [10]. En cada una de las etapas propuestas se pretende alcanzar un cierto grado de madurez tanto en los trabajos de mantenimiento del sistema, como en el seguimiento y ejecución del proceso.

El presente trabajo se desarrolla en cinco capítulos. En el primero de ellos se describen los antecedentes en torno al Programa de Modernización Tecnológica y Operativa, así como la problemática que se pretende atacar en cuanto a la situación que guarda el Sistema Integral de Información del Registro Federal de Electores. Se destina el capítulo dos para explicar detalladamente en qué consiste el Proceso Unificado de Rational. En el capítulo tres y cuatro se presenta la propuesta de solución así como su justificación. Finalmente en el capítulo cinco se documentan las conclusiones del presente trabajo.



## OBJETIVOS

El objetivo general es establecer una estrategia de implementación del Proceso Unificado de Rational, que considere las necesidades muy particulares que se presentan en torno al mantenimiento del Sistema Integral de Información del Registro Federal de Electores. El fin de la estrategia es lograr que el desarrollo de sistemas sea una actividad controlada, eficaz y eficiente. Dicha estrategia definirá la forma en la que evolucionará el proceso de ingeniería de software, que se pretende implementar en la Dirección de Desarrollo de Sistemas del Registro Federal de Electores.

Los objetivos específicos son:

- a) Definir un alcance gradual y personalizado de la implementación del Proceso Unificado de Rational, en donde se observe, a través de distintas etapas, la forma en la que evolucionará el proceso de desarrollo.
- b) Definir una estrategia de desarrollo para iniciar con los trabajo de mantenimiento del Sistema Integral de Información del Registro Federal de Electores, que mitigue los riesgos que conlleva el tomar la responsabilidad y el control de un sistema tan grande y complejo.
- c) Proponer una estructura organizacional en el área de desarrollo, que considere tanto disciplinas de trabajo definidas en el Proceso Unificado de Rational, como las necesidades en cuanto al desarrollo, mantenimiento y soporte de los sistemas que son operados y requeridos por el Registro Federal de Electores.

# Planteamiento del problema

En este capítulo se plantea la problemática que se presenta en el desarrollo del Sistema Integral de Información del Registro Federal de Electores (SIIRFE). Así mismo, se detalla la visión y la implementación de dicho sistema en torno al Programa de Modernización Tecnológica y Operativa que lleva a cabo la Dirección Ejecutiva del Registro Federal de Electores.

## **1.1 El Programa de Modernización Tecnológica y Operativa**

El Programa de Modernización Tecnológica y Operativa es la conjunción de una serie de iniciativas y esfuerzos de la Dirección Ejecutiva del Registro Federal de Electores con el fin de hacer mas eficientes, en términos de recursos humanos, materiales y financieros, los procesos operativos que sustentan la generación, actualización y mantenimiento de los instrumentos y servicios electorales, que se derivan del cumplimiento de las atribuciones establecidas en el Código Federal de Instituciones y Procedimientos Electorales (COFIPE) para esa Dirección Ejecutiva [3].

El Programa de Modernización Tecnológica y Operativa fue concebido a principios del año 2001 en su primera versión y es en el mes de agosto de ese mismo año cuando el Consejo General del Instituto Federal Electoral lo aprobó como una de las nueve políticas prioritarias que se definieron como objetivos estratégicos de largo alcance.

Durante el mes de septiembre de 2001, la Dirección Ejecutiva del Registro Federal de Electores definió una visión estratégica para fortalecer y consolidar este programa. Algunos de los elementos que conforman la visión de ese programa son:

- Enfoque a procesos.
- Definición de una arquitectura de información.
- Diseño e instrumentación de una arquitectura de sistemas e infraestructura tecnológica modular y abierta.
- Orientar las plataformas tecnológicas a una cultura de servicio.
- Contar con elementos para desarrollar una planeación a corto, mediano y largo plazos, en el marco del ciclo electoral a tres años.

Adicionalmente, la estrategia definida en el último trimestre del 2001, considera que el Programa de Modernización Tecnológica y Operativa sería estructurado en fases, que a su vez serían conformadas por proyectos específicos con diversos objetivos y alcances.

La visión de este Programa es proporcionar las bases tecnológicas y operativas para iniciar, integrar y consolidar el ciclo electoral 2003 – 2006, desde la perspectiva de las funciones y responsabilidades del Registro Federal de Electores.

Los objetivos del Programa podrían resumirse en los siguientes:

- a) Hacer más eficientes los procesos operativos del Registro Federal de Electores.
  - Mejorar el nivel de servicio en los Módulos de Atención Ciudadana.
  - Disminuir el tiempo de producción y entrega de credenciales para votar con fotografía.
  - Abatir los costos de producción de la credencial para votar con fotografía.
  - Abatir los costos de procesamiento de datos e información que sustenta la entrega de productos y servicios electorales.

- b) Optimizar de manera permanente la calidad del Padrón Electoral, el cual es el instrumento principal de la misión del Registro Federal de Electores.
- Consolidar los programas preventivos y correctivos de verificación y depuración de la base de datos.
  - Incrementar los niveles de actualización, cobertura y vigencia del Padrón Electoral.
  - Establecer mecanismos y controles de seguridad que garanticen la confiabilidad y confidencialidad de la base de datos.
- c) Desarrollar y consolidar una cultura de calidad y servicio en la estructura organizacional del Registro Federal de Electores.
- Definir y establecer un programa de mejora continua de los procesos institucionales.
  - Certificar de manera organizada los procesos operativos conforme la norma ISO-9000:2000.
  - Desarrollar un programa de capacitación integral para el personal.

Con este programa se pretende alcanzar los siguientes aspectos:

- a) En lo funcional y operativo.
- Formar las bases y consolidar una reingeniería de procesos, sistemas de información y estructura organizacional del Registro Federal de Electores.
  - Establecer un Sistema Integral de Información del Registro Federal de Electores que automatice paulatinamente las tareas que sustentan los procesos operativos.
  - Centralizar la información del Padrón Electoral e instrumentos electorales asociados.
  - Incluir la interacción entre los diversos centros operativos del Registro Federal de Electores (Módulos de Atención Ciudadana, Vocalías del Registro Federal de Electores, organismos centrales, etc.).
- b) En lo geográfico.
- Se extiende a los niveles nacional, estatal y distrital. El territorio nacional se encuentra dividido en 300 distritos electorales de acuerdo a lo estipulado en el artículo 53 de la Constitución Política de los Estados Unidos Mexicanos.
- c) En lo tecnológico.
- Centralizar la información, integrando las bases de datos en una sola.
  - La creación de una nueva arquitectura de sistemas que forme la columna vertebral del intercambio de datos e información entre las diversas áreas operativas.
  - Privilegiar el uso de tecnologías abiertas y de licenciamiento gratuito.
  - Utilizar la Red Nacional de Informática del Instituto Federal Electoral (RedIFE), como medio de transmisión de datos del nuevo sistema.

El programa se desarrolla en cuatro fases, las cuales son descritas a continuación.

### 1.1.1 Fase de definición e inicio

La fase 1 del programa de modernización, denominada como de “Definición e inicio” tuvo lugar de Julio del 2001 a Octubre de 2002. Hasta entonces el esquema de operación del Registro Federal de Electores incluía la participación de 17 Centros Regionales de Cómputo (CRC), en los cuales se captura y se procesaban las solicitudes de expedición de credencial de elector para votar con fotografía. Las solicitudes eran levantadas en papel en los Módulos de Atención Ciudadana (MAC) y enviadas entonces al CRC correspondiente. En la figura 1.1 se muestra el diagrama de la red informática que soportaba ese esquema de operación.

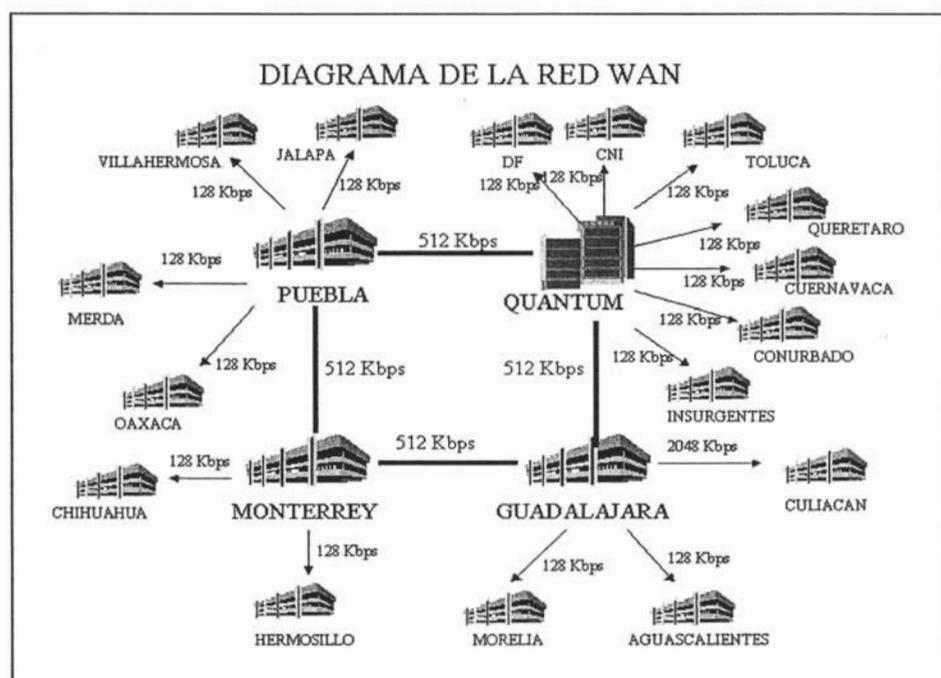


Figura 1.1 Red informática con la que operaba el Registro Federal de Electores hasta 2003

Lo trascendente en esta fase consistió en dotar a los MAC de infraestructura tecnológica para que en éstos centros de trabajo se capturara de forma electrónica los “Formatos Únicos de Actualización”. El Centro Regional de Cómputo recibía la información del MAC, verificaba y validaba los trámites de movimientos al padrón electoral, luego entonces enviaba archivos electrónicos al Centro Nacional de Cómputo (CNC) para su procesamiento y consolidación en una base de datos a nivel nacional. El CNC procesaba, verificaba y validaba por segunda ocasión y enviaba los archivos de producción al Centro

de Producción de Credenciales (CPC), el cual producía los formatos de credencial para su posterior distribución en la República Mexicana.

En el diagrama de la figura 1.2 se esquematiza la manera en la que se comunicaban los 17 Centros Regionales de Cómputo con los Módulos de Atención Ciudadana durante el periodo de la fase 1.

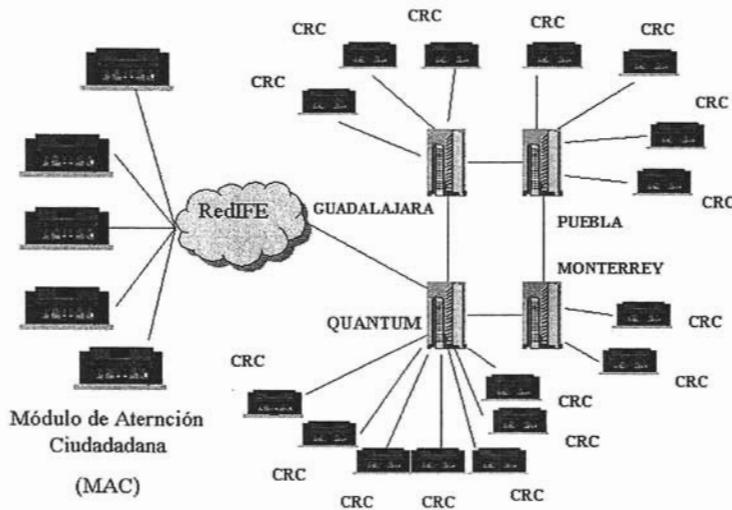


Figura 1.2 Comunicación de los MAC con los CRC

En la fase 1 se desarrollaron los siguientes proyectos:

- Integración tecnológica y de servicios en los Módulos de Atención Ciudadana (MAC) del Instituto Federal Electoral. Se dotó por primera vez de equipo tecnológico a los MAC y fueron integrados a la RedIFE. Se desarrolló una aplicación informática denominada "Sistema de Captura de Información Electoral (SCIE)" como la herramienta de software para registrar los trámites de los ciudadanos. Lo que se buscó con estos trabajos fue obtener mayor eficiencia y eficacia en el levantamiento de información de campo.
- El desarrollo de un nuevo proceso de la producción de credenciales, con la creación de un nuevo centro operativo que opera a través de la contratación de servicios externos, y en el que también se realizaron trabajos de innovación a varios elementos de la credencial para votar con fotografía. Lo que se buscó con estas actividades fue, entre otras cosas, la reducción en costos y contar con más elementos de seguridad en el formato de credencial para votar.

### 1.1.2 Fase de integración

La fase 2 del programa de modernización, denominada como de “Integración” tuvo lugar de Octubre de 2002 a Diciembre del 2003. En ese periodo se desarrolló y dio inicio la operación del Sistema Integral de Información del Registro Federal de Electores (SIIRFE), con el cual se logró la centralización de la información mediante una nueva definición de una arquitectura de información y de aplicaciones.

Con este nuevo sistema, los MAC capturan electrónicamente las solicitudes de expedición de formatos de credencial para votar, y las envían al centro de cómputo principal para su procesamiento. Es en esta fase donde se desincorporan los 17 Centros Regionales de Cómputo. El centro de cómputo principal procesa, verifica y valida a nivel nacional la información enviada por los MAC. Luego se envían los archivos de producción al CPC para la generación y posterior distribución de los formatos de credencial.

En la figura 1.3 se esquematiza la nueva operación del Registro Federal de Electores, en donde es posible observar que los MAC se comunican a través de la RedIFE con los centros de cómputo principal y alterno. También se observa que los Órganos desconcentrados del Instituto Federal Electoral, en sus ámbitos estatales y distritales son enlazados también a través de la RedIFE, la cual es administrada por la Unidad Técnica de Servicios de Informática (UNICOM) del mismo Instituto.

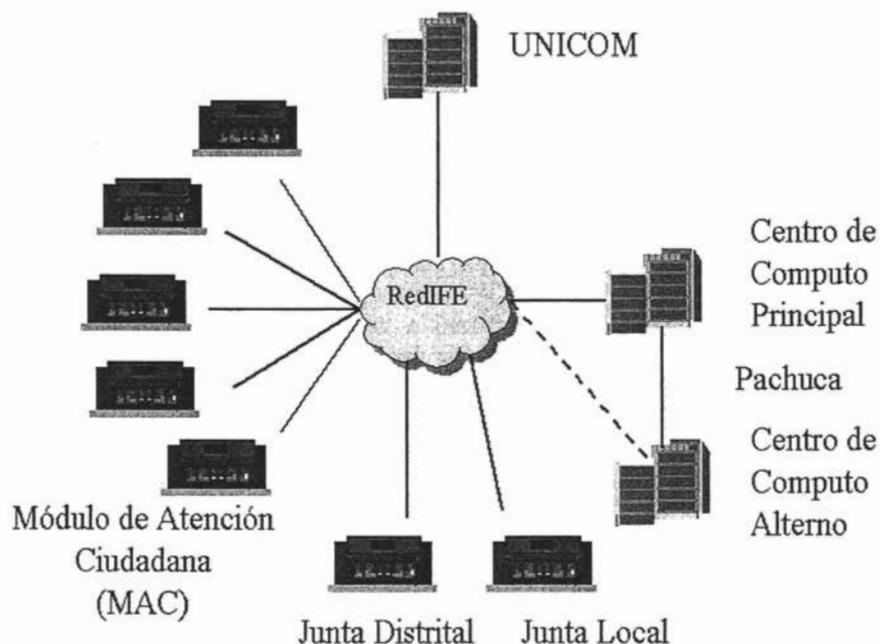


Figura 1.3 Comunicación de los MAC con los nuevos centros de cómputo

Para la fase 2 se contempló el desarrollo de los siguientes proyectos:

- Desarrollo de la primera fase del Sistema Integral de Información del Registro Federal de Electores, que consiste en cuatro de los ocho subsistemas que lo constituyen. Además se realizó la adquisición e instalación de la infraestructura tecnológica para soportar la operación de dicho sistema. Con ello se busca la reducción de los tiempos de atención de trámites del ciudadano e integración del Padrón Electoral en una sola base de datos. Este proyecto será descrito mas adelante en este mismo capítulo.
- Consolidación de los 17 Centros Regionales de Cómputo. Establecer los mecanismos y procedimientos que se deberán seguir para realizar la entrega, reubicación y/o desincorporación de los recursos humanos, materiales, financieros, bases de datos e imágenes que se tiene bajo resguardo de los Centros Regionales de Cómputo, para llevar a cabo la consolidación de los mismos. Este proyecto se desarrolló en seis etapas:
  - ❖ En la primera etapa se realizó el diagnóstico de la situación mediante la recopilación de la información. Se solicitó de información de los recursos con los que contaban los Centros Regionales de Cómputo.
  - ❖ En la segunda etapa se llevó a cabo la integración y el análisis de la información recopilada en la primera etapa, con el fin de determinar el estado de los recursos con los que contaba cada centro operativo.
  - ❖ En la tercera etapa se definió el procedimiento para la desincorporación y/o reubicación del personal y cada uno de los bienes. Se estableció con el área administrativa la logística del traslado de bienes.
  - ❖ En la cuarta etapa se realizó la preparación de los bienes y evaluación de requerimientos en el CECYRD y en oficinas centrales.
  - ❖ En la quinta etapa se llevó a cabo el proceso de desincorporación, reubicación y el traslado de los bienes, así como la liquidación y/o reubicación de personal en el mismo Registro Federal de Electores así como en otras áreas del Instituto Federal Electoral.
  - ❖ En la última etapa se llevó a cabo el proceso de entrega de inmuebles y cancelación de servicios contratados en cada uno de los 17 centros regionales de cómputo.
- La construcción de un nuevo edificio en el Estado de Hidalgo, nombrado “Centro de Cómputo y Resguardo Documental (CECYRD)” que consta de un conjunto de edificios para dar cabida a tres áreas operativas denominadas “Centro de Computo”, “Resguardo Documental” y “Credencialización e Impresión”. Se pretende, entre



otras cosas, proporcionar un espacio que permita realizar la actualización tecnológica que requiere el Registro Federal de Electores, contemplando un centro de cómputo que tenga la infraestructura eléctrica, comunicaciones, seguridad, protección y área necesaria para la operación. El lugar designado para la construcción del CECYRD es un predio ubicado en el Estado de Hidalgo relativamente cerca de la ciudad capital Pachuca, dentro del municipio llamado “San Agustín Tlaxcala”. El proyecto completo consta de cuatro etapas:

- ❖ La primera etapa contempla la definición total del Predio, la cual consiste en el sembrado de edificios, servicios, vialidades, subestación, caseta de vigilancia, barda en el perímetro y en general la constitución de lo que será el conjunto de edificios.
- ❖ La segunda etapa, que se encuentra en desarrollo se divide en 2 fases , una que consiste en construir el primer edificio del conjunto denominado “Centro de Cómputo”, así como las obras de limitación del predio, cuarto de maquinas, cisterna, caseta y torre de vigilancia. La segunda contemplará el suministro e instalación de aire acondicionado de precisión, cableado estructurado con equipo activo, sistemas de seguridad (Controles de Acceso, circuitos cerrados de televisión, extinción y supresión de incendios) y mobiliario.
- ❖ La tercera etapa esta en revisión y contempla un área de digitalización y un área de almacenamiento o resguardo documental de aproximadamente 3600 m2. Adicionalmente se construirá un espacio más, correspondiente al centro metropolitano IFETEL.
- ❖ La cuarta etapa corresponde al edificio de Credencialización e Impresión. Estas áreas compartirán un mismo edificio pero mantendrán cierto grado de independencia.

### 1.1.3 Fase de consolidación

La fase 3 del programa de modernización, denominada como de “Consolidación” tiene lugar en el año 2004 y parte del 2005, y contempla el desarrollo de los siguientes proyectos:

- El desarrollo de un DataWareHouse entorno a la información del Padrón y la Cartografía Electoral.
- El desarrollo de una estrategia para auditar y en su caso mejorar la calidad de la información. Al respecto se realizan actualmente programas de auditoria donde se corrobora, mediante muestras de información, que la información almacenada en la base de datos del padrón electoral se encuentre debidamente respaldada con la documentación de las solicitudes realizadas por los ciudadanos.

- La digitalización, el resguardo y consulta documental. Poner en operación los sistemas de digitalización de expedientes de los ciudadanos para atender los movimientos generados por el SIIRFE y los documentos históricos, además de contar con un sistema de consulta de documentos en pantalla, y su localización física en el almacén de resguardo documental, para que puedan ser extraídos físicamente en caso necesario. Para desarrollar este proyecto es necesario dividirlo en dos etapas, la digitalización de los expedientes generados por el SIIRFE, y la digitalización de los expedientes actualmente resguardados en los Centros Regionales de Cómputo (histórico):
  - ❖ Para la primera etapa, los documentos generados por el SIIRFE serán los “Formatos Únicos de Actualización y Recibo (FUAR)”, teniendo como antecedente que estos estarán impresos y mantendrán un estándar; se propuso como solución la digitalización con lectura de código de barras y reconocimiento de caracteres para su indexación en la base de datos. Se estima una demanda promedio de 40,000 documentos diarios. Estos documentos serán endosados y empacados para su resguardo físico en el CECYRD, para esto también fue necesario contar con un software que permitiera, además de la consulta en pantalla, la ubicación física del documento en el almacén.
  - ❖ La segunda etapa del proyecto está considerada para trabajarse de dos a tres años, esta documentación requiere de realizar la indexación manual, debido a la diversidad de formatos a lo largo del tiempo y una indexación semiautomática para el caso de los recibos de credencial, ya que éstos cuentan con la clave de elector en código de barras. Esta parte del proyecto deberá desarrollarse con equipo de cómputo adicional con el que opera el SIIRFE, y de igual manera los documentos serán endosados y resguardados en el CECYRD. El sistema de consulta deberá ser el mismo que se implementó para la primera etapa del proyecto.

#### **1.1.4 Fase de evolución y transición**

La fase 4 del programa de modernización, denominada como de “Evolución y transición”, se iniciará en el año 2005 y se pretende concluir en el 2006. Las actividades previstas para esta fase son las siguientes:

- Implementar un mecanismo de depuración preventiva de registros duplicados en el Padrón Electoral mediante el uso de biométricos en base a imágenes faciales y huellas digitales.
- Implementar un plan de seguridad integral entorno al Sistema Integral de Información del Registro Federal de Electores y a la información de las bases de datos.

- Mejora continúa de los procesos y sistemas de información. Como parte del control que se requiere se contempla el desarrollo de reportes estadísticos y de indicadores estratégicos. También se tiene considerado la instrumentación de un programa de calidad de acuerdo a la norma ISO9000:2000
- Reestructura de la Dirección Ejecutiva del Registro Federal de Electores, donde se reorganicen las funciones de las distintas áreas operativas de esta organización.

## 1.2 El Sistema Integral de Información del Registro Federal de Electores

Como parte del Programa de Modernización Tecnológica y Operativa, se definió el SIIRFE como la columna vertebral para el soporte y operación de los servicios propios del Registro Federal de Electores.

El modelo mostrado en la figura 1.4 representa el nuevo concepto de arquitectura de sistemas que utiliza la DERFE como soporte a sus operaciones.

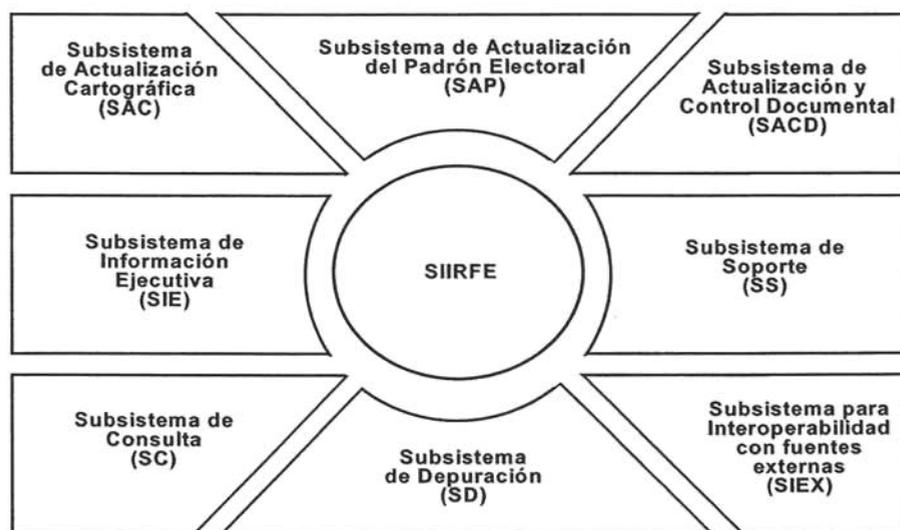


Figura 1.4 Los ocho subsistemas del SIIRFE

El SIIRFE está definido mediante ocho subsistemas principales, los cuales deben trabajar de manera conjunta para automatizar los siguientes procesos institucionales:

1. Conformación, integración y actualización del Padrón Electoral

2. Credencialización.
3. Verificación y diagnóstico del Padrón Electoral.
4. Depuración del Padrón Electoral.
5. Emisión de Listados Nominales Electorales.
6. Conformación, integración y actualización de la Cartografía Electoral.
7. Apoyo normativo en materia jurídica.
8. Apoyo a Procesos Electorales Locales.
9. Proporcionar acceso a la información Electoral.

La arquitectura de los sistemas, la estructura organizacional y la infraestructura tecnológica en la que el Registro Federal de Electores basaba sus operaciones y funciones principales no contaba con las características técnicas y operativas para responder de manera eficiente a las necesidades del Instituto. Entre otros, el impacto se podría resumir en los siguientes aspectos:

- Se requería de un gran esfuerzo de las diferentes áreas de la DERFE para cubrir en tiempo y forma con las demandas de servicios e información. El proceso de integración de la información para la toma de decisiones oportuna no era muy eficiente; entre otras cosas, por que se obligaba a generar múltiples extractos de información en los diferentes centros operativos (CRC, MAC, CNC, Vocalías) para cumplir con esas demandas de información.
- Dado que los sistemas se operaban de forma aislada en las diferentes ubicaciones (MAC, Juntas Locales, CRC's, CNC), era necesario implementar controles y puntos de verificación adicionales a fin de garantizar la confiabilidad, integridad y consistencia del proceso e información, los cuales deterioraban los tiempos de respuesta.
- No se podían rastrear ágilmente los movimientos y trámites específicos en cualquier momento. Tampoco se conocía a detalle el tiempo que tarda un trámite en cada punto del proceso.

Las áreas usuarias afectadas por las situaciones previamente descritas eran tanto las áreas internas de la Dirección Ejecutiva del Registro Federal de Electores (DERFE) como los demás órganos del Instituto Federal Electoral y partidos políticos, las cuáles requieren contar con información actualizada de todos los movimientos al Padrón Electoral, listas nominales, así como de indicadores de niveles de servicio de los procesos del negocio para toma de decisiones. También habría que considerar a los ciudadanos, quienes requieren conocer la situación de sus trámites relacionados con la inscripción o actualización de información en el Padrón Electoral y la expedición de credenciales para votar.

Con la instrumentación del SIIRFE se buscaba un sistema que proporcionara y mantuviera datos consistentes e íntegros y que garantizara la seguridad de la información, a la vez que permitiera a las áreas usuarias de la DERFE contar con elementos tecnológicos y de servicio que contribuyeran a aumentar la eficiencia y eficacia de sus labores. Entre otras cosas, dicho sistema debía permitir:

- Integrar la información referente al padrón y a la cartografía electoral en una base de datos centralizada, consolidando de igual manera la información que se generara durante la operación del Registro Federal de Electores.
- Actualizar la base de datos e imágenes del Padrón Electoral en línea, en función de la infraestructura tecnológica que lo soportara.
- Brindar la posibilidad a los ciudadanos para realizar sus trámites en cualquier módulo de la Entidad Federativa a la que pertenecieran.
- Dar seguimiento a los trámites en cualquier estatus en el que se encontraran, manteniendo un esquema de control estricto de todos los movimientos que afectan al padrón electoral, así como la integración de mecanismos de auditoría.
- Reducir los tiempos de algunos procesos relevantes para el Instituto, como los de la generación y entrega de credenciales y los de la generación de listados nominales.
- Proveer métricas de control de operaciones, que permitieran notificar desviaciones en cada parte del proceso de empadronamiento y credencialización.
- Integrar mecanismos preventivos que permitieran precisar el tipo de trámites de los ciudadanos y redujeran la posibilidad de ingreso de registros duplicados.
- Agilizar los procesos de depuración correctiva del padrón electoral.
- Contar con mecanismos versátiles para la extracción de información y generación de reportes.
- Implementar un esquema de seguridad que garantizara la integridad y confiabilidad de la información.

El Instituto Federal Electoral decidió llevar a cabo el desarrollo de la primera fase del Sistema Integral de Información del Registro Federal de Electores “SIIRFE” mediante la contratación de servicios externos. Fue durante los meses de agosto y septiembre de 2002 cuando se llevó a cabo la licitación pública nacional para desarrollar el proyecto denominado como “*Análisis, diseño, construcción, pruebas, implantación y mantenimiento del Sistema Integral de Información del Registro Federal de Electores (SIIRFE)*”.

Entre otras especificaciones, se solicitó utilizar el Proceso Unificado de Rational (RUP) como marco metodológico de desarrollo. En su primera fase, el software de operación del proyecto SIIRFE fue integrado por los siguientes cuatro subsistemas:

a) Subsistema de actualización al padrón electoral (SAP)

Este subsistema tiene como principales funciones realizar la actualización al padrón electoral y lista nominal, capturar los datos del ciudadano, determinar, en su caso, el tipo de trámite y la identificación geoelectoral del domicilio del ciudadano para actualizar la base de datos del padrón electoral e históricos de movimientos, así como generar la credencial para votar, además, permite llevar el control de impresión y entrega de la misma a sus titulares, acción que conlleva la actualización de la lista nominal.

b) Subsistema de actualización cartográfica (SAC)

Este subsistema tiene como función principal llevar un seguimiento de la actualización de la cartografía electoral y estar en posibilidades de proporcionar la información necesaria con respecto a todos los usuarios de la misma: para tal efecto, cuenta con una función que le permite llevar a cabo la captura y procesamiento de datos relativos a la cartografía electoral. La información que genera este subsistema en conjunto con la base cartográfica digital, es utilizada en los MAC para ubicar geográficamente el domicilio del ciudadano que acude a realizar algún trámite, así como para efectuar análisis estadísticos de la información geoelectoral.

c) Subsistema de depuración (SD)

Este subsistema tiene como función principal llevar a cabo de manera automatizada las actividades que forman parte de la depuración del padrón electoral, aplicar bajas de los registros que pierden vigencia producto de defunciones, suspensión de derechos o pérdida de la ciudadanía, así como considerar la incorporación de funciones que tienen como finalidad realizar procesos de validación preventiva y depuración correctiva del padrón electoral.

d) Subsistema de consulta (SC)

Este subsistema tiene como objetivo principal proporcionar acceso a la información electoral que se genera en la DERFE, tanto la información generada durante el proceso de producción de la credencial como la relativa al padrón electoral ya conformado. El subsistema explota fundamentalmente el banco de información electoral nominativa y estadística ya consolidada en el nivel central. La naturaleza de las consultas de este subsistema es de carácter general, permitiendo la búsqueda y acceso a expedientes y movimientos de ciudadanos, explotación de cifras consolidadas tanto del padrón electoral como de la operación para conformarlo. Todo esto dentro de un esquema de seguridad donde se definen diferentes niveles de acceso por tipo de usuario. Adicionalmente, en este subsistema se generan diversos reportes en medio impreso o magnético como: listados nominales de ciudadanos en el padrón electoral, listados de ciudadanos en el padrón electoral con fotografía y listados nominales de exhibición.

Adicionalmente a lo antes mencionado, en el proyecto de referencia se solicitaron servicios complementarios como fueron los de: capacitación, migración de datos y el desarrollo de

documentos estratégicos como los de certificación de calidad, administración del cambio, plan de seguridad, continuidad de operaciones, limpieza de datos, entre otros.

El esquema mostrado en la figura 1.5 representa el modelo conceptual del SIIRFE con la integración del “Data Ware House”. En ese modelo se pueden observar los siguientes aspectos:

- Aplicaciones tipo OLTP (“On Line Transaction Processing”) para actualización de la base de datos del padrón electoral.
- Consolidación de bases de datos e integración de texto e imágenes.
- Esquema de almacén de datos (“Data Ware House” y “Data Marts”).
- Uso de herramientas OLAP (“On Line Analytical Processing”) para explotación de información.

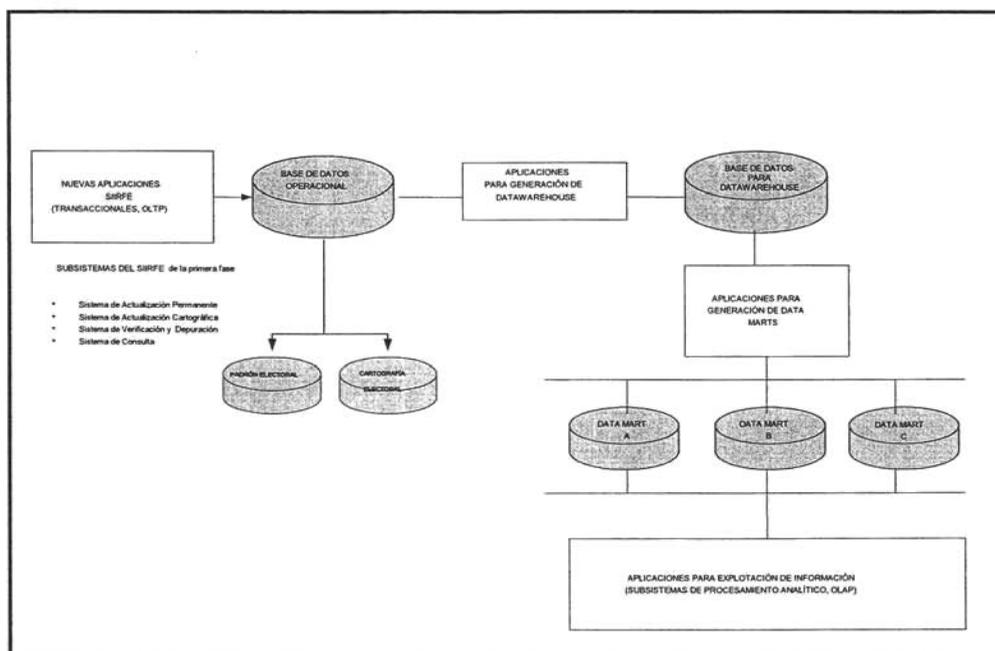


Figura 1.5 Modelo conceptual de las bases de datos del SIIRFE

El SIIRFE fue diseñado para ser operado en dos tipos de centros operativos, por un lado se requirió de una parte del sistema en los módulos de atención ciudadana, donde se capta la información de los ciudadanos, y otra en los centros de cómputo centrales, donde se procesa la información para actualizar las bases de datos nacionales y se solicita a un tercero la producción de los formatos de credencial de elector.

Para los MAC la infraestructura tecnológica que soporta al SIIRFE puede describirse con los siguientes componentes:

- Aplicación desarrollada en lenguaje de programación JAVA 1.3.1
- Sistema operativo: LINUX Red Hat 6.2
- Manejador de base de datos: Postgresql 7.2.1
- Servidor de aplicaciones: Tomcat 4.0.4
- Servidor de WEB: Apache 1.3
- Equipo personal con procesador pentium 4 y memoria RAM de 512MB.
- Dispositivos periféricos: captor de firma, lector de huellas digitales, lector de códigos de barra, cámara fotográfica e impresora.

La figura 1.6 muestra la infraestructura tecnológica en los MAC.

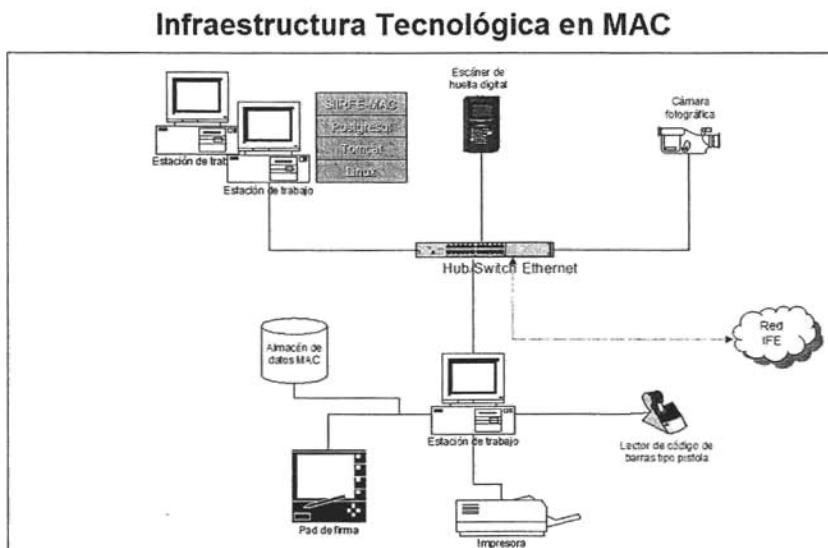


Figura 1.6 Infraestructura tecnológica en MAC

Para los centros de cómputo centrales, la infraestructura tecnológica para soportar la operación del SIIRFE puede describirse con los siguientes componentes:

- Aplicación desarrollada en lenguaje de programación: JAVA 1.3.1
- Sistema operativo: HP-UX
- Manejador de base de datos: Oracle Real Application Cluster 9.0.4
- Servidor de aplicaciones: Web Logic Integration versión 7
- Servidor de WEB: Apache 1.3



- Equipo HP SuperDome y RP8400
- Arreglos de discos XP128 y bibliotecas de cintas para almacenamiento.
- Red de acceso con seguridad perimetral (FireWalls) y balanceadores de carga.
- Red de administración con un servidor de administración de redes y sistemas, consola de administración, un servidor de respaldos y movimientos de datos, un servidor de identificación, autenticación y autorización de usuarios, un servidor de administración de seguridad y un servidor de control de procesos.

En la figura 1.7 se muestra la infraestructura tecnológica en los centros de cómputo primario y secundario.

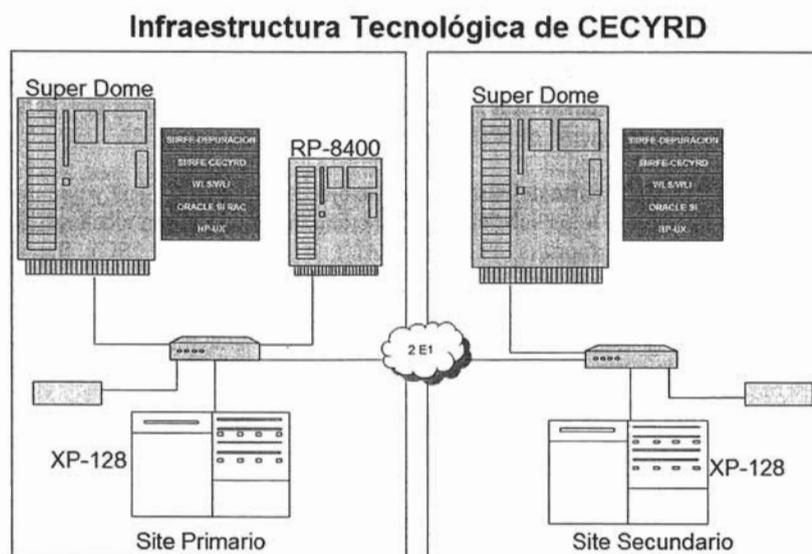


Figura 1.7 Infraestructura tecnológica de CECYRD

Durante su desarrollo, el SIIRFE fue implementado en cuatro grandes módulos:

1. SIIRFE-MAC. Es la parte del sistema que opera en los módulos de atención ciudadana. Entre otras cosas, cuenta con la funcionalidad para capturar los trámites de actualización al padrón electoral, para validar la información con respecto a la base de datos del padrón electoral estatal y nacional, para integrar la información con la cartografía electoral estatal, para llevar a cabo la lectura y entrega de credenciales, para la administración de usuarios, generación de reportes, etc. El SIIRFE-MAC permite el envío de la información, definidas como transacciones, de los MAC al centro de cómputo primario. Se cuenta con dos tipos de MAC, los que están conectados a la

RedIFE, denominado como fijos por estar ubicados en un espacio físico fijo, y los que no cuentan con una comunicación permanente a la RedIFE. Para los primeros, el envío de las transacciones se realiza mediante un proceso automático conforme estas se van generando en el MAC. Para los MAC fuera de línea, se generan archivos por lote que contienen las transacciones generadas en un periodo de tiempo, típicamente de un día. Estos archivos, denominados como archivos de transacciones son quemados en un CD, para que se integran al sistema eventualmente, en cuanto estos se hagan llegar a una oficina del IFE con enlace de comunicaciones.

2. SIIRFE-Intercambio. Es la parte del sistema que opera parte en el CECYRD y parte en los MAC y se utiliza para el envío de información por lotes. Esto es requerido para aquellos MAC que no cuenten de manera permanente con un enlace de comunicaciones a la RedIFE, y que son denominados como “móviles” por que se mueven entre poblados, principalmente rurales. Esta funcionalidad es ejecutada en el SIIRFE-MAC en aquellos módulos denominados como “fijos” que cuentan con conexión a la RedIFE, desde donde se transmiten los archivos de transacciones generados y remitidos por los MAC “móviles”. Dicha transmisión se realiza invocando un servicio en el CECYRD, que hace las veces de recepción de archivos.
3. SIIRFE-CECYRD. Es la parte del sistema que se ejecuta en el CECYRD y que incluye la funcionalidad referente a los subsistemas de actualización al padrón electoral (SAP), de actualización a la cartografía electoral (SAC) y de consultas (SC). Son estos subsistemas los que llevan a cabo la actualización a las bases de datos transaccionales del padrón y la cartografía electoral a nivel nacional. Es a través de estos subsistemas mediante los cuales se proporcionan los insumos para la producción de los formatos de credencial de elector y para la generación de los listados nominales con fotografía para votar.
4. SIIRFE-Depuración. Es la parte del sistema que se ejecuta en el CECYRD y que incluye la funcionalidad referente a la depuración correctiva y preventiva de registros duplicados en el padrón electoral, así como a la verificación al padrón electoral en cuanto a niveles de actualización y calidad del mismo. Tanto la depuración correctiva como la preventiva se basan en la comparación de información tanto de datos como de imágenes con ciertos criterios preestablecidos, en donde un operador del sistema determina o no la duplicidad del registro. Este subsistema opera con su propia base de datos de trabajo y se tiene previsto que en el mediano plazo que interactúe con un sistema de comparación de imágenes de las fotografías tomadas a los ciudadanos, así como con un sistema de comparación de huellas dactilares (AFIS).

El SIIRFE se implementó en tres capas: la capa de presentación, la capa de lógica de negocio y la capa de persistencia. La capa de presentación se basó en el paradigma de cliente delgado y el patrón de diseño Modelo-Vista-Controlador. Específicamente, se usó como cliente un navegador de Internet, y las páginas se diseñaron conforme al estándar XHTML 4.0.1 de la W3C. La capa de la lógica de negocio se implementó mediante clases Java estándar y mediante Enterprise Java Beans (EJB), esta capa es independiente de la implementación de la capa de presentación y de la capa de persistencia. Para el caso de MAC, la capa de persistencia se implementó mediante clases Java estándar y con

Hibernate, y para el caso de CECYRD, se llevó a cabo mediante la funcionalidad ya proporcionada por el servidor de aplicaciones Web Logic.

La figura 1.8 muestra esquemáticamente la arquitectura del SIIRFE.

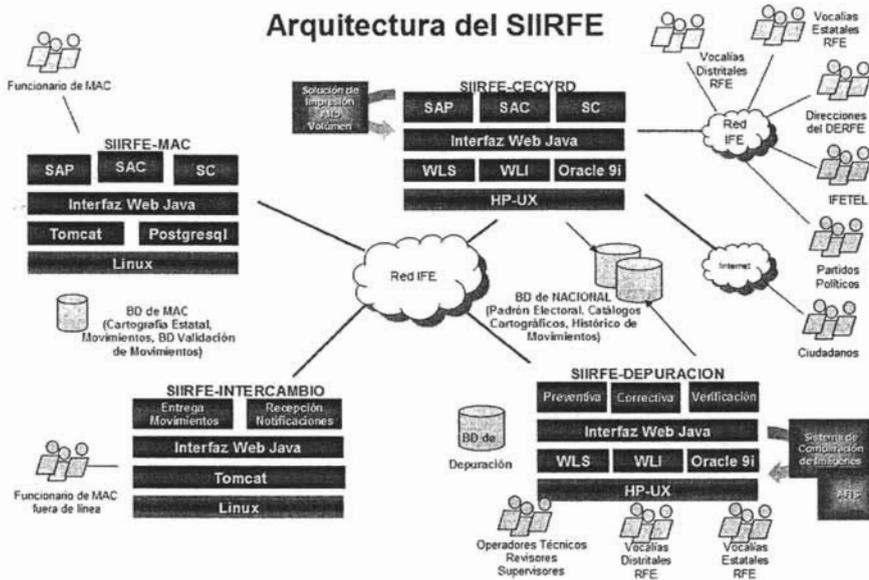


Figura 1.8 Arquitectura del SIIRFE

### 1.3 La problemática en torno al desarrollo del SIIRFE

El desarrollo del proyecto SIIRFE ha implicado el manejo de una nueva y compleja plataforma tecnológica que es la base para soportar la nueva funcionalidad de manera modular y es suficientemente flexible para adecuarse a los posibles cambios que se pudieran dar en el ámbito de las actividades del Registro Federal de Electores.

Durante el levantamiento de requerimientos del desarrollo del proyecto, se establecieron reuniones de trabajo con los usuarios de las diversas áreas operativas del Registro Federal de Electores con la finalidad de que la empresa recopilara los requerimientos que se convertirían en funcionalidades a desempeñar por el Sistema a desarrollar; el conjunto de requerimientos dio origen a 245 casos de uso, que constituyen los objetivos a desarrollar en la primera fase del SIIRFE. De esos, 13 casos de uso quedaron cancelados por considerarse fuera del alcance del proyecto tal y como se había licitado.

Dado que el alcance del desarrollo del sistema es muy amplio, basado en los requerimientos y alcance del contrato, se planteó implementarlo en dos versiones. La primera versión entró en producción en el mes de Septiembre de 2003 para la parte del SIIRFE-MAC, en tanto que para el SIIRFE-CECYRD y SIIRFE-Depuración entró en producción en Octubre de ese mismo año. No obstante, la estabilización del sistema se logró hasta el mes de febrero del 2004, siendo necesario desarrollar y liberar a producción varias versiones después del inicio de operaciones. La descripción y el alcance de la primera y segunda versión se explican a continuación.

### **1.3.1 El alcance de la primera versión del SIIRFE**

La primera versión del SIIRFE incluye el ciclo operativo para la atención de trámites de actualización al Padrón Electoral, el cual considera las diferentes actividades que se llevan a cabo en los Módulos de Atención Ciudadana y en oficinas centrales.

Las funcionalidades desarrolladas en el SIIRFE-MAC corresponden básicamente a satisfacer los requerimientos necesarios para procesar la información proporcionada por el ciudadano así como las validaciones establecidas en las reglas de actualización del Padrón y Lista Nominal de Electores. Este ciclo inicia con la recepción de información en los módulos de atención ciudadana, validando los datos generales del ciudadano en búsqueda inicial en la base de datos local y en el caso de módulos conectados en línea, en la base de datos nacional, con la finalidad de asignar correctamente el movimiento solicitado. Continúa con el envío de información al centro de cómputo para el respectivo proceso de información central de los cuales, de acuerdo a las validaciones y procesos realizados, activa servicios de depuración preventiva, validaciones cartográficas y de movimiento posterior para concluir en movimiento exitoso que constituye el suministro para la generación del archivo de órdenes y lotes de producción que será entregado al impresor para la generación de la credencial para votar.

En el subsistema de actualización al padrón electoral en SIIRFE-CECYRD, las funcionalidades desarrolladas corresponden básicamente a satisfacer los requerimientos necesarios para recibir y procesar la información enviada por los MAC para la correspondiente validación de los movimientos, afectación al Padrón Electoral, generación de órdenes de producción de credenciales, así como el envío de información a MAC. Este ciclo inicia con la recepción de información de los módulos de atención ciudadana, activa servicios de depuración preventiva, validaciones cartográficas y de movimiento posterior para concluir en movimiento exitoso que constituye el suministro para la generación del archivo de órdenes y lotes de producción que son entregados al impresor para la generación de la credencial para votar.

En el subsistema de actualización a la cartografía electoral en SIIRFE-CECYRD, las funcionalidades que se encuentran operando como parte de éste subsistema corresponden a la identificación de la GEO referencia electoral con base a los catálogos cartográficos y a la información proporcionada por los ciudadanos, así como la recuperación en campo de los registros que no se pueden referenciar.

En el subsistema de consultas en SIIRFE-CECYRD, las funcionalidades implementadas comprenden la consulta de ciudadanos en Padrón, así como la consulta del estatus de los trámites de acuerdo a los criterios de búsqueda preestablecidos.

La funcionalidad desarrollada en el SIIRFE-Depuración, contempla los servicios de depuración preventiva en el MAC, para la correcta asignación del movimiento desde este centro de captura. Si el registro no es identificado, activa los servicios de depuración preventiva en CECYRD, cuya función es detectar posibles candidatos en el Padrón Electoral a fin de evitar duplicados en el mismo. La funcionalidad incluye también la emisión de cédulas de visita a campo para poder determinar en última instancia la situación del registro del ciudadano.

El desarrollo de las funcionalidades del Sistema Integral de Información del Registro Federal de Electores se ha llevado a cabo en torno a los ciclos importantes identificados en el proceso general de los requerimientos solicitados, con lo cual se asegura el correcto cumplimiento de ciclos completos en las actividades de procesamiento de la información para la generación de la Credencial para Votar incluyendo las validaciones de depuración preventiva. Se definieron tres grandes ciclos operativos:

#### a) Ciclo de producción

La funcionalidad del ciclo de producción está orientada a satisfacer los requerimientos necesarios para procesar la información proporcionada por el ciudadano, así como las validaciones establecidas en las reglas de actualización del Padrón y Lista Nominal de Electores.

Este ciclo inicia con la recepción de información en los módulos de atención ciudadana, validando los datos generales del ciudadano en búsqueda inicial en la base de datos local y en el caso de módulos conectados en línea, en la base de datos nacional, con la finalidad de asignar correctamente el movimiento solicitado. Continúa con el envío de información al centro de cómputo para el respectivo proceso de información central de los cuales, de acuerdo a las validaciones y procesos realizados, activa servicios de depuración preventiva, validaciones cartográficas y de movimiento posterior para concluir en movimiento exitoso que constituye el suministro para la generación del archivo de órdenes y lotes de producción que será entregado al impresor para la generación de la Credencial para Votar.

Para asegurar el cumplimiento de este ciclo se incluye una funcionalidad general que se ejecuta en los siguientes componentes:

En SIIRFE-MAC:

- Registro del trámite en MAC.
- Validaciones en MAC para prevención de registros duplicados.
- Envío y recepción de información de módulos en línea y fuera de línea.

- Referencias cartográficas.
- Administración de transferencia de la información.
- Seguridad de información en MAC.

En SIIRFE-CECYRD y SIIRFE-Depuración:

- Recepción de información proveniente de los módulos de atención ciudadana.
- Procesamiento en el subsistema de actualización al Padrón.
- Servicios de depuración preventiva y comparación de registros candidatos.
- Servicios de actualización cartográfica y recuperación de rechazos cartográficos.
- Envío de notificaciones al MAC y a los subsistemas correspondientes.
- Generación de archivos de órdenes y lotes de producción para impresión de la credencial para votar.

b) Ciclo de calidad

Las funcionalidades desarrolladas para asegurar el ciclo de calidad inician con la lectura de las credenciales generadas, en un proceso manual de verificación de estándares de calidad establecidos para este fin. El proceso asegura también la reimpresión de las credenciales que por algún motivo no cumplan con la calidad requerida.

En SIIRFE-CECYRD:

- Lectura de credenciales impresas.
- Validación de los estándares de calidad en las credenciales impresas.
- Consultas relativas a los lotes disponibles.

c) Ciclo de distribución

Las funcionalidades incluidas en este ciclo constituyen el registro y seguimiento de las credenciales producidas, mismas que son entregadas como producto del ciclo de calidad y que pasan por los diversos puntos de distribución necesarios para ponerlos a disposición del ciudadano en el módulo de atención ciudadana en el cual se realizó la solicitud del trámite.

En SIIRFE-CECYRD:

- Registro de los lotes de credenciales en los diferentes puntos de distribución de la credencial para votar.

En SIIRFE-MAC:

- Entrega de la credencial al ciudadano

En la figura 1.9 se muestra un diagrama de flujo para la generación de la credencial donde se resaltan los tres ciclos descritos anteriormente.

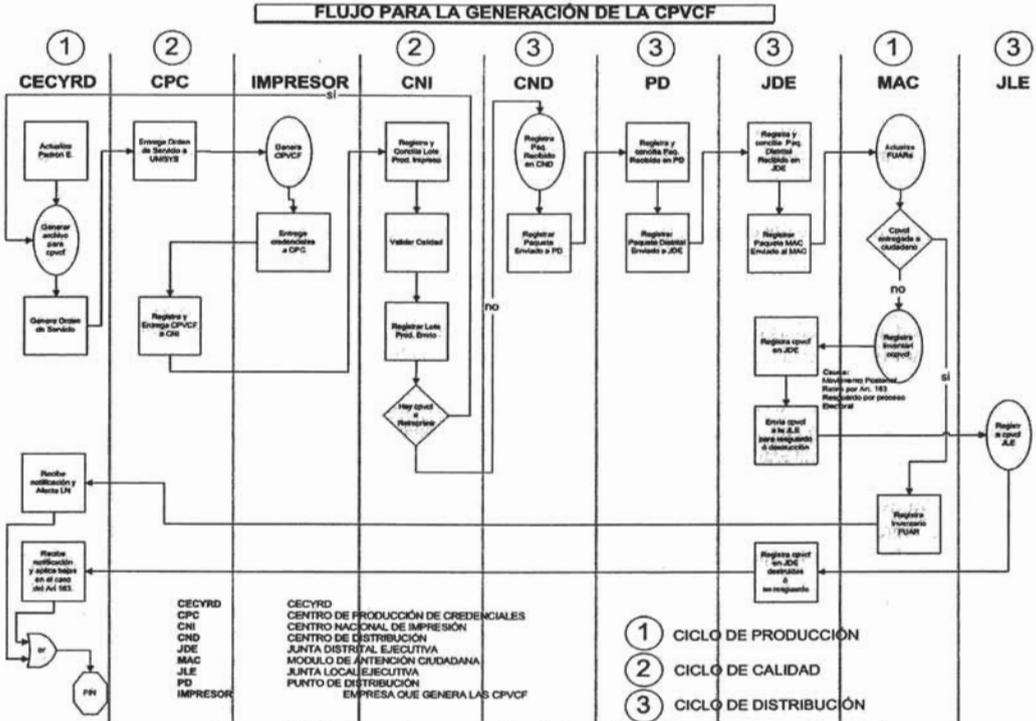


Figura 1.9 Flujo para la generación de la credencial para votar con fotografía

### 1.3.2 El alcance de la segunda versión del SIIRFE

La funcionalidad correspondiente a la segunda versión de los subsistemas del SIIRFE incluye requerimientos de tipo complementario a los ciclos descritos anteriormente, la cual permitirá llevar a cabo tareas que se realizan de manera programada y optimizar las funcionalidades de la primera versión.

El desarrollo pendiente en el SIIRFE-MAC es el que se refiere a la administración de la funcionalidad básica ya desarrollada, como la depuración, exportación e importación de la información contenida en la base de datos del MAC. También falta por desarrollar las funcionalidades correspondientes a la generación de actas administrativas de movimientos y la generación de listados de Credenciales a entregar.

Las funcionalidades de exportación e importación en el SIIRFE-MAC se utilizan cuando deja de operar un MAC y se requiere que los trámites que manejaba sean transferidos a otro generalmente del mismo distrito. El número de MAC que se encuentran operando es variable y depende del tipo de campaña de actualización que se este realizando, de acuerdo a lo indicado por el COFIPE o de acuerdo con la realización de procesos electorales locales.

En el subsistema de actualización al padrón electoral del SIIRFE-CECYRD, las funcionalidades programadas son relativas a la administración de directorios de módulos, administración de calendarios electorales, generación e impresión de reportes de lotes de producción y conciliación de los Formatos Únicos de Actualización y Recibos en CECYRD para su resguardo documental.

En el subsistema de actualización cartográfica del SIIRFE-CECYRD, las funcionalidades programadas para la segunda versión son las relacionadas con el proceso de registro y consultas de casillas extraordinarias, así como la generación de archivos de impresión para las notificaciones de cambios en la cartografía electoral hacia los ciudadanos.

En el subsistema de consultas del SIIRFE-CECYRD, las funcionalidades programadas para segunda versión son las referentes a los procesos de insaculación, mediante los cuales son sorteados los ciudadanos para elegir a los que fungirán como los funcionarios de las casillas electorales el día de votaciones, la generación de listados nominales de exhibición y definitivos para los procesos electorales locales a nivel entidad y federales, la impresión de los libros negros en donde se relacionan los ciudadanos que no pueden votar, y las consultas de los formatos de credencial para votar que han sido reportados como robados.

En el SIIRFE-Depuración, las funcionalidades pendientes por desarrollar corresponden al proceso de generación de universos y aplicación de bajas en su fase correctiva, los procesos relativos a la verificación nacional muestral, las consultas estadísticas de registros dados de baja, los respaldos de información de los registros dados de baja, la aplicación de bajas por pérdida de vigencia de acuerdo con el artículo 163 del COFIPE, y los monitoreos de procesos de bajas y análisis de registros a dar de baja.

### **1.3.3 La necesidad de los trabajos de mantenimiento del SIIRFE**

La empresa encargada de realizar los trabajos de desarrollo del SIIRFE en su primera fase, sólo logró concluir una parte de las funcionalidades enmarcadas en la primera versión de dicho sistema, quedando sin realizar parte de esta primera versión y toda la segunda versión. Se lograron terminar aproximadamente poco menos de la mitad de los casos de uso que, sin embargo, no se encontraron libres de incidencias. Para el desarrollo de la segunda versión, la empresa comunicó un problema de liquidez para la asignación de recursos, quedando sin entregar dicha versión.

La decisión de haber licitado el desarrollo del SIIRFE se fundamentó principalmente en dos sentidos, uno fue que la carga de trabajo de las áreas internas de sistemas no permitían destinar los recursos para el nuevo proyecto, y otro fue que se deseaba la utilización de



tecnologías abiertas y de punta que no necesariamente eran dominadas por el personal de dichas áreas.

No obstante que los trabajos relativos al proceso federal electoral del 2003 demandaban una fuerte atención por parte de los recursos con los que contaba el área de sistemas, fue posible incorporar a cierto personal de dicha área en las reuniones de levantamiento de requerimientos, seguimiento al proyecto, ejecución del plan de pruebas e inclusive en los grupos de desarrollo de la empresa encargada del proyecto. Esta última iniciativa con el objeto de comenzar con la transferencia de conocimientos en las tecnologías utilizadas. También, durante el desarrollo del proyecto, se llevaron a cabo planes de capacitación en los lenguajes y herramientas con las que fue desarrollado el SIIRFE.

Adicionalmente, y una vez que dieron inicio las operaciones con el SIIRFE, se solicitó a la empresa el servicio de transferencia tecnológica como estrategia para dar continuidad al mantenimiento del sistema en el mediano plazo. El cual consistió en que a través de su personal se dirigieran grupos de trabajo conformados por el personal de la Dirección de Desarrollo de Sistemas. El fin de estos grupos de trabajo fue el de continuar con algunos de los trabajos de desarrollo y mantenimiento del SIIRFE-MAC, logrando con ello asimilar más rápido los conocimientos requeridos conforme a la tecnología utilizada.

De igual manera, se creó un segundo nivel de centro de atención a usuarios para atender todo lo relativo al sistema SIIRFE-MAC. Esta mesa de ayuda lo integra personal de la Dirección de Desarrollo de Sistemas y de la Dirección de Soporte Técnico del Registro Federal de Electores. El IFE cuenta con un centro de atención a usuarios operado por la UNICOM y que es el conocido como el primer nivel de atención.

En principio ese segundo nivel de atención lo conformó personal que estuvo en la ejecución de las pruebas de ese sistema y que por su trayectoria laboral, tenía el conocimiento del negocio en particular de los centros de trabajo de los MAC. Posteriormente se ha ido incorporando personal con un perfil más adecuado para la atención a usuarios y el personal con un perfil de desarrollador ha sido reincorporado a los grupos de desarrollo. Ese centro de atención a usuarios fue creado con el inicio de operaciones del SIIRFE y continúa laborando actualmente.

Una vez que se logró contar con una base de conocimientos y procedimientos de solución a las problemáticas mas comunes que ocurrían entorno a la operación del SIIRFE-MAC, se llevo a cabo la capacitación al personal del primer nivel del centro de atención, la cual consistió en explicaciones teórico-prácticas del funcionamiento del sistema, así como su participación directa en la solución a problemas en el segundo nivel de atención.

Actualmente la situación del SIIRFE, en cuanto a funcionalidad no desarrollada y a una serie de incidencias que se presentan bajo ciertos escenarios, obliga a destinar recursos para la realización de trabajos de índole técnico que soporten la operación del sistema, los cuales son realizados por la Dirección de Operaciones, la Dirección de Desarrollo de Sistemas y la Dirección de Soporte Técnico. Personal de la Dirección de Operaciones del CECYRD cuenta con los elementos para administrar adecuadamente la base de datos del SIIRFE, así

como para proponer y realizar alternativas de corrección a incidencias detectadas en la operación.

Entre otras, las actividades que se realizan son: generación de alternativas para el tratamiento de incidencias en la base de datos, monitoreo y diagnóstico de posibles problemas que pudieran impactar en la producción de credenciales, puesta a punto y afinación de las aplicaciones y la configuración de la plataforma tecnológica que permita realizar los ajustes para mantener o mejorar el desempeño conforme se incorporan usuarios al SIIRFE.

Con la salida de la empresa en el proyecto, se requiere iniciar los trabajos de mantenimiento y soporte al sistema, comenzando por una etapa de estabilización y afinación de las funcionalidades que se encuentran operando. En esta etapa se plantean, de forma secuencial, las siguientes actividades de mantenimiento:

1. Continuar con la corrección definitiva de incidencias de la versión que se encuentra en producción, en especial del subsistema SIIRFE-CECYRD. Dichas correcciones permitirán liberar recursos de las actividades operativas que se implementaron de manera temporal para soportar, mediante procedimientos adicionales, las consecuencias de las principales incidencias del sistema.
2. Continuar con el desarrollo de funcionalidad que ya está liberada para el SIIRFE-MAC, pero que aún no está completamente desarrollada en SIIRFE-CECYRD. Se requiere cerrar los ciclos operativos de funcionalidad.

En un segundo momento se podrá comenzar con los trabajos de desarrollo de la totalidad de la funcionalidad definida para los cuatro subsistemas de la primera fase de desarrollo de *software* del SIIRFE, con lo que se podrá automatizar los procesos principales que se llevan a cabo en la DERFE. Para un tercer momento se planearán las actividades de desarrollo de los subsistemas que falten por desarrollar.

### **1.3.4 La necesidad de implementar un proceso de ingeniería**

La estructura de la Dirección de Desarrollo de Sistemas del Registro Federal de Electores cuenta actualmente con cuatro subdirecciones y con nueve Jefaturas de Departamento. El organigrama se muestra en la figura 1.10.

Las principales actividades de dicha Dirección antes del inicio de operaciones del SIIRFE, eran de mantenimiento a los diversos sistemas que se encontraban en producción. Se contaba con sistemas para la actualización cartográfica a nivel distrital, sistemas de apoyo para la actualización al padrón electoral y generación de listados nominales que se realizaba en los CRC, sistemas de registro de ciudadanos candidatos a baja por defunción, pérdida de derechos y duplicados a nivel estatal, el sistema de captura de información electoral en MAC, entre otros.

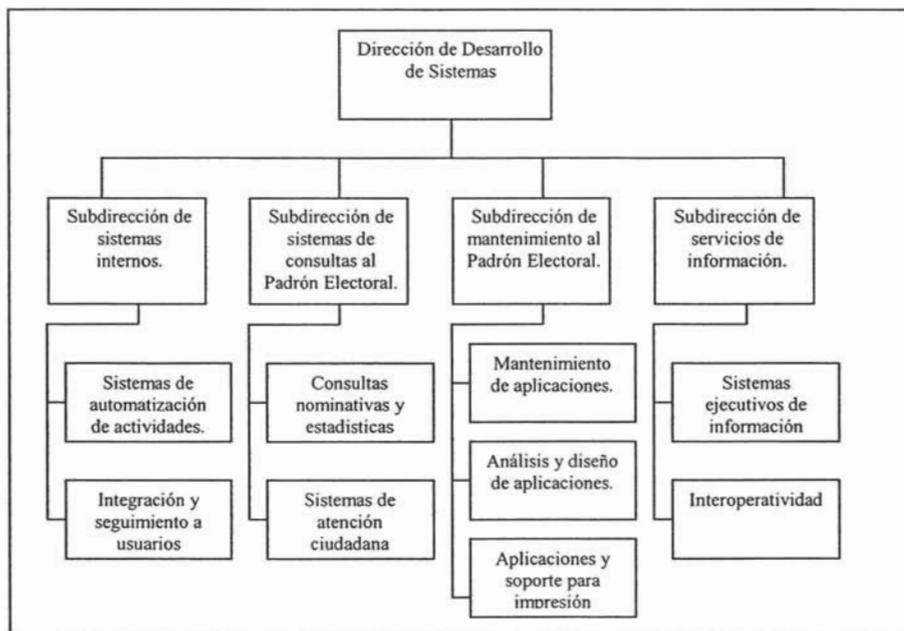


Figura 1.10 Estructura organizacional de la Dirección de Desarrollo de Sistemas.

Conforme a lo comentado ya respecto a la concepción e instrumentación del SIIRFE, sistema en el que se pretende incorporar todas las aplicaciones informáticas que soportan la operación de la DERFE, las funciones que en algún momento se habían definido para el área de sistemas quedaron obsoletas. Así pues, los nombres de los puestos no corresponden a las actividades que actualmente viene realizando el personal que los ocupa, ya que la visión y las necesidades que se tienen hoy en día son diferentes a las que en su momento, cuando se definió la estructura de esa manera, se tenían.

Además, se han observado oportunidades de mejora en la forma de trabajo, que permitan no sólo construir software de mayor calidad, sino también poder mitigar riesgos potenciales que se pueden presentar en mayor o menor medida durante los trabajos de mantenimiento al SIIRFE, toda vez que éste es un sistema que ya se encuentra en producción. A continuación se enlistan las necesidades más relevantes que se tienen en torno a la actual forma de trabajar:

- Se requiere contar con un método de trabajo que permita mantener la relación entre los procesos de negocio y los requerimientos del sistema (trazabilidad), así como la relación entre los requerimientos y los componentes de software generados.
- Se requiere homogenizar los diferentes niveles de detalle y profundidad en la especificación de requerimientos.
- Se requiere llevar una adecuada administración de los requerimientos.
- Se requiere definir un proceso de pruebas.

- Se requiere automatizar las pruebas.
- Se requiere tener un plan de administración de cambios y configuración.
- Se requiere un proceso homogéneo de administración del proyecto que involucre los diferentes subsistemas del SIIRFE.
- Se necesita la preparación y el soporte del ambiente del proyecto.
- Se requiere establecer políticas y procedimientos en la DERFE, que guíen las actividades del proyecto SIIRFE relativas a la planeación y seguimiento integral, aseguramiento de la calidad del software, estrategias de comunicación del equipo del proyecto y administración de riesgos.
- Se requiere contar con métricas específicas para conocer el estatus del proyecto en cualquier momento del tiempo y de esta manera poder asegurar la calidad del producto de software.

La problemática descrita en el presente capítulo puede entonces ser resumida en lo siguiente: a) se tiene una gran necesidad de comenzar con los trabajos de mantenimiento del SIIRFE que permitan no solamente mejorar el funcionamiento del sistema, sino también el poderlo adaptar de acuerdo a los cambios que se presentan en la organización, y b) se tiene también una necesidad de reestructurar el área de sistemas, redefiniendo puestos y funciones que permitan desarrollar aplicaciones de una manera mas controlada y con mayor calidad.

Atendiendo a estas necesidades, es que se presenta una propuesta concreta para comenzar con los trabajos de desarrollo del SIIRFE basados en un marco de trabajo: por una parte, se establecerán alcances en el desarrollo del sistema en forma incremental, y por otra parte, se definirá una forma de ir implementando un proceso de ingeniería de software basado en el Proceso Unificado de Rational, también de forma incremental.

# Fundamentos del Proceso Unificado de Rational

En este capítulo se exponen los conceptos teóricos de un proceso de ingeniería de software, y en particular del Proceso Unificado de Rational (RUP). Se describen detalladamente las mejores prácticas de la industria en el desarrollo de software, así como los fundamentos de RUP en cuanto a su estructura dinámica (tiempo) y su estructura estática (disciplinas y flujos de trabajo).

## 2.1 El Proceso Unificado de Rational como un proceso de ingeniería de software

Un proceso 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 proceso de ingeniería de software es el proceso mediante el cual se desarrolla un sistema a partir de ciertos requerimientos, o que son nuevos, para un ciclo de desarrollo inicial, o que son de cambio, para un ciclo de desarrollo evolutivo.

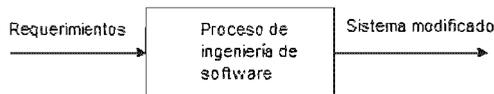


Figura 2.1 Un proceso de ingeniería de software

Un proceso de ingeniería de software debería de evolucionar a través del tiempo. Durante esta evolución debería limitar su alcance a las realidades actuales que permitan las tecnologías, herramientas, personas y formas de trabajo de la organización donde se pretende implementar.

- **Tecnologías.** El proceso debe construirse sobre las tecnologías (lenguajes de programación, sistemas operativos, computadoras, telecomunicaciones, etc.) disponibles en el momento en que se va a emplear el proceso.
- **Herramientas.** Los procesos y las herramientas deben desarrollarse en paralelo. Un proceso ampliamente utilizado puede soportar la inversión necesaria para crear las herramientas que lo soporten.
- **Personas.** Se debe limitar el conjunto de habilidades necesarias para trabajar en el proceso a las habilidades que los desarrolladores actuales poseen, o apuntar aquellas que los desarrolladores puedan obtener rápidamente.
- **Formas de trabajo de la organización.** El creador del proceso debe adaptar el proceso a las realidades del momento de la organización.

Para definir un proceso en particular, se deben equilibrar estos cuatro elementos. Se debe diseñar el proceso de forma que pueda evolucionar, deberá madurar durante varios años antes de alcanzar el nivel de estabilidad y madurez que le permitirá resistir a los rigores del desarrollo de otros productos, manteniendo a la vez un nivel razonable de riesgo en su utilización. El desarrollo de un producto nuevo es bastante arriesgado en sí mismo como para añadirle el riesgo de un proceso que esté poco validado por la experiencia de su uso. Sin este equilibrio de tecnologías, herramientas, personas y organización, el uso del proceso sería bastante arriesgado.

El proceso Unificado de Rational (RUP) es un proceso de ingeniería de software que presenta una propuesta basada en disciplinas de trabajo para asignar tareas y responsabilidades dentro de una organización de desarrollo de sistemas. Su objetivo es asegurar la producción de software de calidad que cumpla con las necesidades de sus usuarios finales dentro de una planeación y presupuesto predecible.

El proceso puede ser descrito en términos de dos dimensiones: tiempo y contenido. La figura 2.2 muestra una representación gráfica de estas dimensiones. El eje horizontal representa el tiempo, y muestra el ciclo de vida o el aspecto dinámico del proceso. Esta dimensión es descrita en términos de fases e iteraciones y se detalla más adelante en el apartado 2.3. El eje vertical representa las disciplinas, en los cuales se agrupan las actividades definidas en el proceso. Esta dimensión es descrita en términos de los componentes del proceso como actividades, flujos de trabajo, artefactos y roles, y se detalla mas adelante en los apartados 2.4 y 2.5.

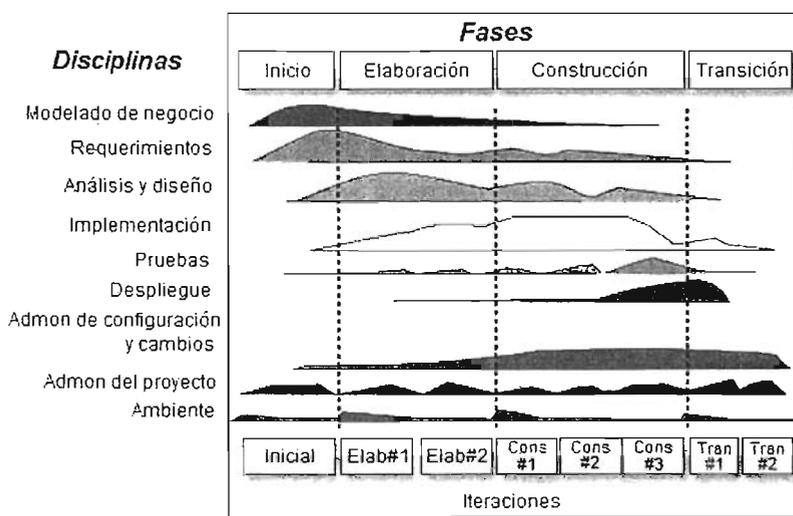


Figura 2.2 Fases y disciplinas de RUP, sus dos dimensiones

El Proceso Unificado de Rational describe una familia de procesos de ingeniería de software orientado a objetos compartiendo una misma estructura, y una arquitectura de proceso. Se trata de un proceso configurable, que puede personalizarse tanto para grupos pequeños como para organizaciones grandes, dedicadas al desarrollo de sistemas. Integra muchas de las mejores prácticas en el desarrollo de sistemas moderno de tal forma que puede adaptarse a una buena cantidad de tipos de proyectos y organizaciones.

El Proceso Unificado de Rational es como tal también un producto, desarrollado y mantenido por Rational® Software, que está en continuo mejoramiento tratando de asimilar

las mejores prácticas y las experiencias más recientes. Como producto provee de una base de conocimientos basado en guías de trabajo y plantillas para cada una de las actividades del desarrollo de sistemas de software. Al tener a todos los miembros del equipo de desarrollo utilizando la misma base de conocimientos, no importa si se está trabajando en la parte de requerimientos, o en el diseño, o en las pruebas, o en la administración del proyecto, estaremos seguros que todos los miembros del equipo comparten un lenguaje común, un proceso y una forma de cómo desarrollar sistemas.

El Proceso Unificado de Rational está soportado por un conjunto de herramientas con las cuales se pueden crear y mantener una serie de artefactos del proceso de ingeniería de software como modelado visual, programación, pruebas, etc.

## ***2.2 Las mejores prácticas en el desarrollo de software***

El Proceso Unificado de Rational integra ciertas maneras de llevar a cabo los trabajos de desarrollo de sistemas que han sido utilizadas en la industria del software. Estas son conocidas como “las mejores prácticas” por que en estudios realizados a varias empresas con proyectos de desarrollo de sistemas exitosos, se ha observado que han sido utilizadas regularmente. A continuación se describen las seis mejores prácticas que se integran en dicho proceso.

### **2.2.1 Desarrollo iterativo e incremental**

Dada la complejidad de hoy en día del software, no es posible en un solo ciclo definir el problema entero, diseñar completamente la solución, construir el software y probarlo. Es necesario implementar una metodología iterativa que permita ir entendiendo el problema poco a poco, así como también ir obteniendo una solución eficiente de manera incremental a través de múltiples iteraciones.

El Proceso Unificado de Rational soporta el desarrollo iterativo enfocando los elementos de mayor riesgo al inicio de cada etapa en el ciclo de vida del desarrollo, tratando de reducir significativamente los riesgos del proyecto en su totalidad. Hacer el desarrollo de manera iterativa permite atacar los riesgos a través de entregas parciales, frecuentemente versiones ejecutables que permiten involucrar al usuario final y con ello una continua retroalimentación.

Los usuarios cambian sus pensamientos a través del tiempo conforme el contexto cambia también. Forzar a que los usuarios acepten el sistema tal y como ellos lo imaginaron originalmente, es un error. Con el desarrollo iterativo, los usuarios pueden comprender más fácilmente un sistema en funcionamiento, aunque aún no opere completamente, que un sistema que sólo existe en forma documental. El tener un sistema con un funcionamiento parcial en una fase inicial permite que los usuarios hagan sugerencias sobre él o señalen requisitos que se hayan podido omitir.



Como cada iteración termina con una versión ejecutable, el equipo de desarrollo permanece enfocado en producir resultados, y la revisión constante del estado del proyecto permite identificar rápidamente cualquier desviación en la planeación del mismo. En la figura 2.3 se ilustra el desarrollo iterativo, en donde cada ciclo inicia con un plan y concluye con el despliegue y evaluación de una nueva versión.

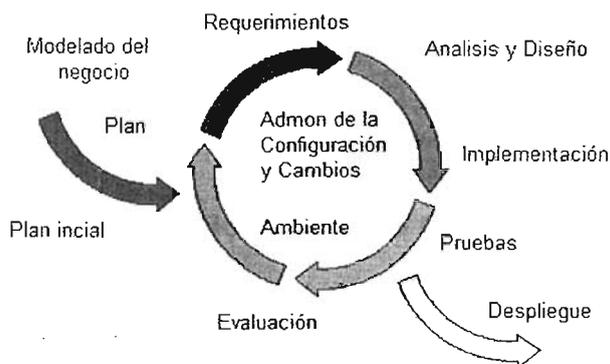


Figura 2.3 Desarrollo iterativo por ciclos

En otras palabras, se pretende dividir el proyecto en un número de mini proyectos, siendo cada uno de ellos una iteración. Cada iteración tiene todo lo que tiene un proyecto de desarrollo de software: planeación, levantamiento de requerimientos, análisis, diseño, implementación, pruebas y entrega. Cada uno de estos mini proyectos se parece al ciclo de vida en cascada debido a que se desarrolla a través de actividades en cascada.

Con el desarrollo iterativo e incremental, los desarrolladores pueden resolver los problemas y los aspectos no cubiertos por las primeras construcciones e incluir cambios para corregirlos casi a la vez. Mediante esta técnica, los problemas se van descubriendo con un ritmo de goteo constante que los desarrolladores pueden tratar fácilmente. Los ingenieros de prueba, los escritores de manuales, los encargados de las herramientas, el personal de gestión de configuración y los encargados del aseguramiento de la calidad, pueden adaptar sus propios planes al calendario en evolución del proyecto. Tienen conocimiento de la existencia de retrasos serios en las primeras fases del proyecto, cuando los desarrolladores encuentran por primera vez los problemas que los originan. Tienen tiempo para adaptar sus propios calendarios.

Con el método iterativo no se planifica el proyecto entero en detalle durante la fase de inicio, solo da los primeros pasos. El equipo de proyecto no intenta planificar las fases de construcción y transición hasta que se ha establecido una base de hechos durante la fase de elaboración. Naturalmente, hay un plan de trabajo durante las dos primeras fases, pero no es muy detallado.

Al término de la fase de elaboración, por tanto, existe una base para planear el resto del proyecto y para poner en marcha un plan detallado para cada iteración de la fase de construcción. El plan para la primera iteración estará muy claro. Las posteriores aparecerán

en el plan menos detalladas, y estarán sujetas a modificación, de acuerdo con los resultados y con el conocimiento que se adquiriera en las iteraciones previas. De igual forma, debería haber un plan para la fase de transición, pero puede que tenga que ser modificado a la luz de lo que el equipo aprenda de las iteraciones de la fase de construcción. Este tipo de planificación nos permite un desarrollo iterativo controlado.

Se deberán ordenar las iteraciones para conseguir una vía directa en la cual las primeras iteraciones proporcionen la base de conocimiento para las siguientes. Las primeras iteraciones del proyecto consiguen incrementar la comprensión de los requisitos, el problema, los riesgos y el dominio de la solución, mientras que las restantes añaden incrementos que conformarán la versión final. Lo que se debe buscar es una serie de iteraciones que siempre avancen; es decir, que nunca haya que volver hacia atrás para corregir el modelo debido a algo que hemos aprendido en la última iteración.

Las iteraciones pueden solaparse en el sentido de que una iteración está a punto de terminar cuando otra está comenzando. La planificación y el trabajo inicial de la siguiente iteración deben comenzar a medida que terminamos y preparamos la entrega de la iteración anterior. El orden en el cual se lleve a cabo la planeación depende, en un grado considerable, de factores técnicos. Sin embargo, el objetivo más importante es ordenar el trabajo en secuencia de modo que puedan desarrollarse antes las decisiones más importantes, aquellas que implican tecnologías, requerimientos y arquitecturas nuevas.

## **2.2.2 Administración de requerimientos y proceso dirigido por casos de uso**

La administración de los requerimientos es necesaria, por que regularmente en los proyectos se presentan las siguientes situaciones:

- Los requerimientos no siempre son obvios, y pueden ser definidos por muchos usuarios.
- Los requerimientos no siempre se pueden expresar en palabras de forma fácil y clara.
- Hay muchos tipos diferentes de requerimientos con diferentes niveles de detalle.
- El número de requerimientos puede llegar a ser inmanejable si no son controlados.
- Los requerimientos tienen diferentes niveles de importancia.
- Al existir varias áreas interesadas, se deben administrar los requerimientos por funcionalidad, en cada una de las cuales posiblemente se involucre a más de un área usuaria.
- Los requerimientos cambian.

El Proceso Unificado de Rational describe cómo encontrar, organizar y documentar la funcionalidad y las reglas del negocio requeridas; define la forma de dar seguimiento, así como la de documentar, entorno a las negociaciones y decisiones que se realizan durante el levantamiento de requerimientos; permite registrar y comunicar fácilmente los requerimientos del negocio.

Existen muchas formas de modelar el problema y representar los requerimientos y reglas de negocio que deben ser desarrolladas en un sistema. Una vez que este modelo es construido, ahora se deberá formular un modelo para la solución. Si ambos modelos son mantenidos por separado, se requerirá de un gran esfuerzo para traducir la expresión del problema de una forma entendible para el usuario final, a una forma entendible para los diseñadores y programadores. Esto podría originar dificultades para interpretar de la forma más adecuada posible las necesidades del usuario final. Un método efectivo para entender y modelar el problema es el modelado por casos de uso. Los casos de uso expresen los requerimientos del sistema de tal forma que pueden ser entendidos por los usuarios, el personal técnico y el personal directivo del proyecto.

Normalmente, un sistema tiene muchos tipos de usuarios. Cada tipo de usuario se representa por un actor. Los actores utilizan el sistema interactuando con los casos de uso. Un caso de uso es una secuencia de acciones que el sistema lleva a cabo para ofrecer algún resultado de valor para un actor. El modelo de casos de uso está compuesto por todos los actores y todos los casos de uso de un sistema.

La noción de casos de uso y escenarios descritos en el proceso permiten capturar los requerimientos funcionales y asegurar que estos sean utilizados durante el análisis, el diseño, la programación y las pruebas del software, logrando que el sistema final cumpla completamente con las necesidades del usuario final.

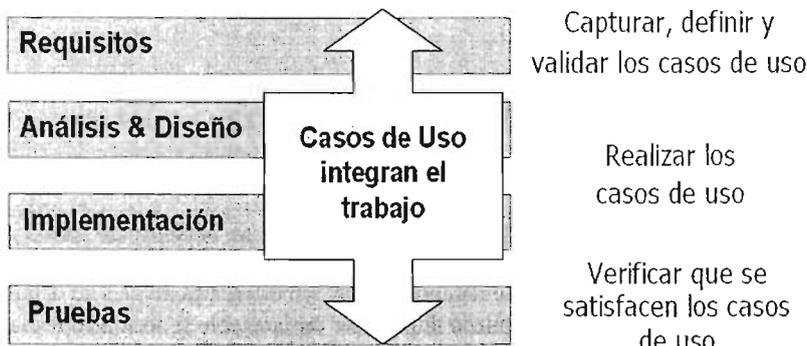


Figura 2.4 Un proceso dirigido por casos de uso

Se dice que el proceso está dirigido por casos de uso, por que el proceso de desarrollo avanza a través de una serie de flujos de trabajo que parte de los casos de uso. Los casos de uso describen los requerimientos del sistema en términos de importancia para el usuario y son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba. Los casos de uso son utilizados en varias disciplinas:

- El concepto de casos de uso puede ser usado para representar los procesos de negocio en la disciplina de modelado de negocio.
- El modelo de casos de uso es uno de los principales artefactos en la disciplina de requerimientos. Los casos de uso identifican lo que el sistema debe de hacer desde el punto de vista de los usuarios. Los casos de uso constituyen un concepto fundamental que deberá de ser aceptado tanto por los usuarios, como por los programadores y el personal de pruebas.
- Durante el análisis y diseño, los casos de uso son realizados en un modelo de diseño. La realización de los casos de uso describe, en términos de objetos, las diferentes partes del sistema que se deberán construir, y la forma en la que esas partes deberán interactuar para soportar la especificación de los casos de uso.
- Durante la implementación, los casos de uso son implementados en términos de modelos de colaboración entre clases.
- Durante las pruebas, los casos de uso proveen de la información necesaria para construir los escenarios de pruebas. La funcionalidad del sistema es verificada al evaluar cada escenario de prueba.
- En la disciplina de administración del proyecto, los casos de uso son usados como base para la planeación del desarrollo iterativo.
- En la disciplina de despliegue, los casos de uso proporcionan la información necesaria para realizar los manuales de usuario.

Los casos de uso nos ayudan a llevar a cabo el desarrollo iterativo. Cada iteración, excepto quizá la primera de todas en un proyecto, se dirige por los casos de uso a través de todos los flujos de trabajo, de los requisitos al diseño y a las pruebas, obteniendo un incremento. Cada incremento del desarrollo es por tanto una realización funcional de un conjunto de casos de uso. En otras palabras, en cada iteración se identifican e implementan unos cuantos casos de uso. En el cuadro siguiente se muestra el avance típico en cuanto al modelado de negocio y a los casos de uso, que se debería de obtener en cada una de las cuatro fases del proceso.

	Modelo de Negocio Terminado	Casos de uso identificados	Casos de uso descritos	Casos de uso analizados	Casos de uso diseñados, implementados y probados
Inicio	50% - 70%	50%	10%	5%	Solo algo relativo a un prototipo para probar conceptos
Elaboración	Casi el 100%	80% o más	40% - 80%	20% - 40%	Menos del 10%
Construcción	100%	100%	100%	100%	100%
Transición					

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos, el desarrollo continúa con la siguiente iteración, sino es así, se deberá analizar la situación y en su caso se deberán ajustar los objetivos planteados para cada iteración.

## 2.2.3 Proceso centrado en la arquitectura

Como se mencionó anteriormente, los casos de uso dirigen el proceso a través de todo el ciclo de vida, pero los trabajos de diseño se realizan entorno a la arquitectura del sistema. Aunque es cierto que los casos de uso guían el proceso, no se desarrollan aisladamente, sino que se desarrollan a la vez que la arquitectura del sistema. Es decir, los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso. A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura y esto lleva a su vez, a la maduración de más casos de uso.

Los arquitectos del software moldean el sistema para darle una forma, la cual debe diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo del tiempo. Para encontrar esa forma, los arquitectos deben trabajar sobre la comprensión general de las funciones clave, es decir, sobre los casos de uso claves del sistema. Estos casos de uso pueden suponer solamente entre el cinco y el diez por ciento de todos los casos de uso, pero son los significativos, los que constituyen las funciones fundamentales del sistema.

El principal objetivo de las primeras iteraciones del proceso, en la fase de elaboración, es producir y validar la arquitectura del sistema, la cual en el ciclo inicial de desarrollo, toma la forma de un prototipo ejecutable o prueba de concepto, que gradualmente llega a ser el sistema final en iteraciones posteriores. La prueba de concepto se refiere a una implementación parcial de los requerimientos no funcionales. Su propósito es mitigar los riesgos relativos al rendimiento, capacidad, robustez, etc., que permitan que la funcionalidad del sistema sea implementada en la fase de construcción sobre una arquitectura sólida.

El Proceso Unificado de Rational proporciona un método sistemático para diseñar, desarrollar y validar una arquitectura. Cuenta con plantillas con las que se pueden describir múltiples vistas de la arquitectura del sistema. En la disciplina de análisis y diseño se especifican actividades en las que se deben identificar los elementos más significativos de la arquitectura. En la planeación de las primeras iteraciones se especifican ciertos objetivos entorno al diseño de una arquitectura y a la resolución de los riesgos técnicos potenciales.

La definición de una arquitectura es relevante por las siguientes razones:

- Permite mantener un control intelectual sobre el proyecto, administrar la complejidad y mantener la integridad del sistema.
- Al definir claramente los componentes y las interfases críticas entre ellos, una arquitectura permite identificar las partes comunes que podrían reutilizarse.
- Permite una planeación organizada por partes funcionales del sistema. El equipo de desarrollo puede ser dividido en pequeños equipos de trabajo, cada uno responsables por una o varias partes del sistema.
- Permite construir un sistema fácil de modificar, esto es, se podrán implementar nuevas funcionalidades en el sistema sin tener que pensar en un impacto dramático en el diseño e implementación existentes.

Además, el Proceso Unificado de Rational soporta el desarrollo basado en componentes, con lo que se tienden a reducir el tamaño y la complejidad de la solución, logrando una arquitectura más robusta y más flexible. Los componentes son módulos, paquetes o subsistemas que realizan una determinada función de forma completa, tienen una interfaz bien definida y pueden ser integrados en más de un módulo o sistema. Son la implementación física de una abstracción en el diseño.

Entre las principales ventajas de utilizar una arquitectura basada en componentes en el proceso, se tienen:

- Con el desarrollo iterativo, los programadores pueden progresivamente identificar componentes y decidir cuales desarrollar, cuales reutilizar y cuales comprar.
- La arquitectura basada en componentes permite definir los elementos estructurales que compondrán el sistema y sus interfaces.
- Conceptos tales como paquetes, subsistemas y capas son usados durante el análisis y diseño para organizar componentes y especificar interfaces.
- Las pruebas son organizadas primero por componentes solos, y después gradualmente se pueden ir probando conjuntos integrados por más componentes.

## 2.2.4 Modelado Visual

Gran parte del proceso trata acerca del desarrollo y mantenimiento de modelos del sistema que se esta construyendo. Los modelos ayudan a entender y dar forma tanto al problema como a la solución. Un modelo es una simplificación de la realidad que permite comprender un sistema grande y complejo, muestra los elementos esenciales del sistema desde una perspectiva particular y oculta los detalles no esenciales.

El Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software. El UML proporciona una rica semántica, notaciones gráficas y de texto que permiten definir el diseño del sistema. De esta manera, mejora la comunicación en el equipo de desarrollo, permitiendo analizar la consistencia entre los trabajos de diseño y de implementación. UML representa la convergencia de las mejores prácticas en le modelado del software dentro de las tecnologías orientas a objetos.

UML es un lenguaje común para expresar varios modelos, pero no indica la manera de cómo programar el software. Proporciona el vocabulario, pero no indica como escribir el libro. RUP es una guía para usar de manera efectiva el UML, describiendo los modelos que se necesitan realizar, las razones del por qué son necesarios, y cómo construirlos.

Modelar el software visualmente permite encontrar solución a las causas de raíz de muchos de los problemas del desarrollo:

- Una especificación no ambigua de casos de uso y escenarios.

- Se diseña el sistema mediante modelos no ambiguos.
- Las arquitecturas no modulares y poco flexibles son expuestas.
- El detalle puede ser oculto cuando no es necesario.
- Los diseños no ambiguos revelan inconsistencias más rápidamente.
- La calidad de la aplicación inicia con un buen diseño.

## 2.2.5 Verificación continua de la calidad

Es importante que la calidad de todos los productos elaborados se evalúen en varios puntos durante el ciclo de vida del proyecto, generalmente en la conclusión de cada iteración. Conforme el software ejecutable es producido, debería ser sujeto a demostraciones y pruebas de los escenarios más importantes de cada iteración, con lo que se lograría obtener eventualmente un software de mejor calidad.

En la figura 2.5 se puede observar una proporción de 100 a 1000 veces más caro encontrar y reparar problemas encontrados después del despliegue que antes. Por esta razón es importante verificar continuamente la calidad del sistema, en cuanto a los requerimientos de robustez, funcionalidad y rendimiento, por ser estos los aspectos críticos en la aceptación de un sistema.

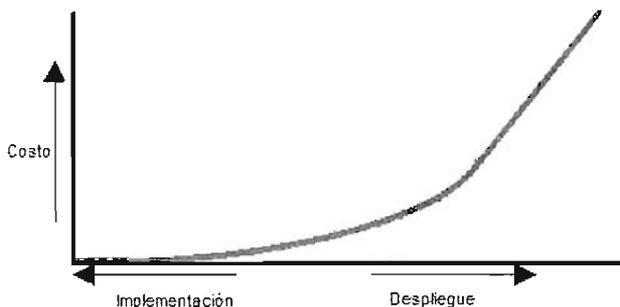


Figura 2.5 El costo de reparar problemas con respecto al tiempo

Verificar la calidad del software continuamente brinda soluciones a diversos problemas en el desarrollo del software:

- La evaluación del avance del proyecto es objetiva y no subjetiva, por que se evalúan resultados y no documentos.
- Esta evaluación objetiva expone inconsistencias encontradas en los requerimientos, diseños e implementaciones.
- Probar y evaluar áreas de alto riesgo, incrementa la calidad y efectividad de esas áreas.
- Los defectos son identificados tempranamente, reduciendo significativamente el costo de arreglarlos.

- Actualmente existen herramientas que permiten probar la funcionalidad, robustez y rendimiento de los sistemas de software.

En el Proceso Unificado de Rational, el aseguramiento de calidad es construido durante el proceso, en todas las disciplinas, flujos de trabajo, fases e iteraciones, involucrando a todos los participantes, usando criterios objetivos, y no es tratado como una actividad separada que es ejecutada por un grupo de trabajo en particular. El aseguramiento de la calidad a través de todo el ciclo de vida significa que se implementa, se mide y se evalúa tanto la calidad del proceso como la del producto.

## 2.2.6 Administración de los cambios

Un aspecto bastante delicado en el desarrollo de sistemas de software es que a menudo se debe trabajar con múltiples programadores, organizados en diferentes equipos, posiblemente en diferentes sitios, trabajando juntos en múltiples iteraciones, versiones, productos y plataformas. La ausencia de una disciplina de control, hará que el proceso de desarrollo degenera rápidamente en un caos. En el Proceso Unificado de Rational, la disciplina de administración de la configuración y cambios describe como se pueden controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.

Coordinar las actividades y los productos generados por los equipos de desarrollo conlleva el establecer procedimientos para la administración de los cambios. Esta coordinación permite una mejor asignación de los recursos basada en las prioridades y riesgos del proyecto, así como una mejor administración de los cambios a través de las iteraciones. Con el desarrollo iterativo, esta práctica permite monitorear los cambios continuamente y con ello activamente descubrir y reaccionar ante diversas problemáticas.

Coordinar las iteraciones y las versiones involucra el establecer y liberar un procedimiento para completar cada iteración. Mantener un control de versiones es esencial para evaluar y administrar el impacto de los cambios.

Controlar los cambios permite dar solución a diversos problemas en el desarrollo de sistemas:

- El flujo de trabajo de levantamiento de requerimientos es definido y repetible.
- Las peticiones de cambio se realizan mediante un procedimiento preestablecido.
- Controlar los cambios permite obtener buenas métricas para evaluar objetivamente el estatus del proyecto.
- Los diferentes espacios de trabajo contienen todos los artefactos realizados en el proyecto, lo cual facilita la consistencia y habilita a su vez, el desarrollo de actividades en forma paralela por diferentes personas o equipos de trabajo.
- La propagación del cambio es evaluada y controlada.
- Los cambios pueden ser mantenidos en un robusto y configurable sistema.



### 2.3 Descripción de la estructura dinámica del proceso

La organización dinámica del proceso se define en la dimensión del tiempo. El ciclo de vida del software es dividido en ciclos, en cada ciclo se trabaja para obtener una nueva generación del producto. El Proceso Unificado de Rational divide el ciclo de desarrollo en cuatro fases consecutivas, cada una de ellas concluye con un hito principal, un punto en el tiempo en el cual ciertas decisiones críticas debieron haber sido tomadas, y ciertas metas clave debieron haber sido alcanzadas. Cada fase tiene un propósito específico:

- Inicio: objetivos y visión del proyecto.
- Elaboración: arquitectura del sistema.
- Construcción: funcionalidad operativa inicial.
- Transición: versión entregable del producto.

Las fases no son idénticas en términos de tiempo y esfuerzo. Aunque esto puede variar considerablemente de acuerdo al proyecto, un típico inicio del ciclo de desarrollo para un proyecto mediano debería considerar la siguiente distribución de tiempo y de esfuerzo por fase:

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5%	20%	65%	10%
Tiempo	10%	30%	50%	10%

En la gráfica de la figura 2.6 se muestra la proporción del tiempo y de los recursos que demanda en promedio cada fase.

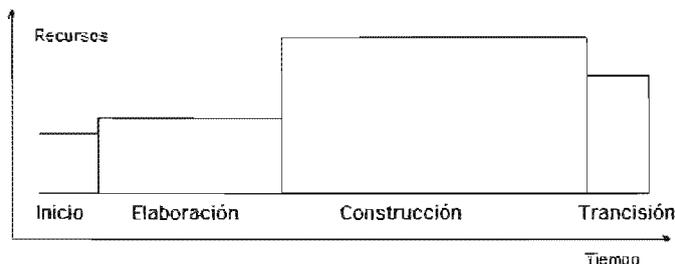


Figura 2.6 Recursos y tiempo destinados a cada fase de RUP

Un paso a través de las cuatro fases es un ciclo de desarrollo, en el cual se produce una generación del software. En las siguientes generaciones del software, se deberán repetir nuevamente la misma secuencia de las cuatro fases, pero esta vez con diferentes esfuerzos en cada una de ellas. Estos ciclos subsecuentes son denominados como ciclos de evolución. Para un ciclo de evolución, las fases de inicio y de elaboración deberían ser considerablemente más pequeñas, ya que la definición básica del producto y la arquitectura han sido determinadas previamente.

Con el desarrollo iterativo e incremental, las iteraciones permiten ir concluyendo poco a poco los modelos definidos en el proceso. Algunos de estos modelos, como el de casos de uso, reciben más atención en las primeras fases, mientras que otros, como el de implementación, la reciben durante la fase de construcción. En la figura 2.7 se ilustra la forma en la que se van concluyendo los distintos artefactos que se deben realizar en cada disciplina.

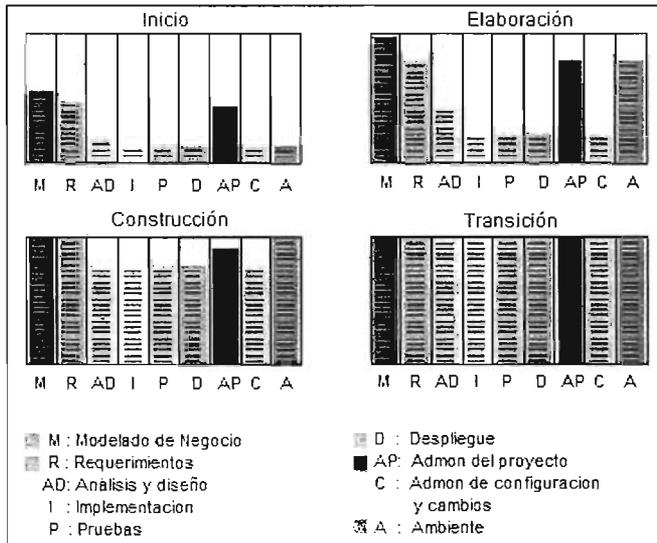


Figura 2.7 Avance por fase en la conclusión de los artefactos de cada disciplina

### 2.3.1 Fase de Inicio

Los objetivos fundamentales de la fase de inicio son el establecimiento del ámbito de lo que debería hacer el producto, la reducción de los riesgos más potenciales, y la preparación del análisis del negocio, que indique la viabilidad del proyecto desde la perspectiva del negocio.

Durante la fase de inicio, se establece el caso de negocio para el sistema y se delimita el alcance del proyecto. Para ello, se deberá identificar todas las entidades externas con las cuales el sistema interactúa y definir la naturaleza de esa interacción a alto nivel. Esto involucra identificar todos los casos de uso y describir algunos de los más significativos. El caso de negocio incluye criterios de éxito, evaluación de riesgos, estimación de recursos, y un plan que muestre las fechas de los hitos principales.

El resultado de la fase de inicio es:

- Un documento de visión: una visión general de los requerimientos esenciales del proyecto, características claves y principales reglas de negocio.
- Un modelo inicial de casos de uso (entre un 10% y un 20% completo).
- Un glosario inicial del proyecto.
- Un caso de negocio inicial, el cual incluye el contexto del negocio, criterios de éxito y presupuesto.
- Una evaluación de riesgos inicial.
- Un plan del proyecto que muestre las fases y las iteraciones.
- Un modelo del negocio, si es necesario.
- Uno o algunos prototipos.

Al final de la fase de inicio, el principal hito del proyecto es el de la definición de los objetivos del proyecto. Los criterios de evaluación para la fase de inicio son los siguientes:

- Las partes interesadas acuerdan el alcance del proyecto, así como una estimación en costo y tiempo.
- Se comprenden los requerimientos establecidos en los principales casos de uso.
- Se logra credibilidad en la estimación de costo y tiempo, prioridades, riesgos y en el proceso de desarrollo.
- Existe un planteamiento de una arquitectura prototipo.
- Se aceptan los gastos reales respecto a los gastos planeados.

El proyecto puede ser cancelado o considerablemente replanteado si no se logran estas metas.

### **2.3.2 Fase de Elaboración**

Los objetivos fundamentales de la fase de elaboración son obtener la base de la arquitectura, capturar la mayoría de los requisitos, y reducir los siguientes riesgos potenciales. Al final de esta fase, se podrán estimar los costos y planeación de la fase de construcción con cierto detalle.

Las decisiones relativas a la arquitectura tendrán que ser tomadas con la consideración de todos los elementos del sistema: su alcance, funcionalidad principal y requerimientos no funcionales como rendimiento y capacidad.

Es fácil argumentar que la fase de elaboración es la más crítica de las cuatro fases del sistema. Al final de esta fase, la parte fuerte de ingeniería se ha completado y se ha tomado la decisión sobre llevar a cabo o no las fases de construcción y transición. Mientras que el proceso deberá siempre adaptarse a los cambios, las actividades de la fase de elaboración aseguran que la arquitectura, los requerimientos y los planes son lo suficientemente estables, y los riesgos son lo suficientemente mitigados, como para poder calcular los costos y el tiempo para terminar el desarrollo del sistema. Conceptualmente, se alcanza el nivel necesario para que la organización lleve a cabo la fase de construcción a un costo determinado.

En la fase de elaboración, un prototipo ejecutable de la arquitectura es construido en una o más iteraciones, dependiendo del alcance, el tamaño, el riesgo y la novedad del proyecto. Este esfuerzo debería enfocarse sobre los casos de uso más críticos que se han identificado en la fase de inicio, con los cuales se exponen típicamente los principales riesgos técnicos del proyecto. Aunque la meta en la evolución del prototipo sea llegar a la definición de una arquitectura estable, esto no excluye la posibilidad de desarrollar uno o más prototipos exploratorios para mitigar algunos riesgos específicos, tales como algunas condiciones que se presenten en el diseño o en los requerimientos, el estudio de viabilidad de algunos componentes, o demostraciones para los inversionistas, clientes y usuarios finales.

El resultado de la fase de elaboración es:

- Un modelo de casos de uso con al menos el 80% completos. Todos los casos de uso y actores han sido definidos y la mayoría de las descripciones de los casos de uso han sido elaboradas.
- Se obtienen los requerimientos no funcionales y algunos otros que no están asociados específicamente con los casos de uso.
- Una descripción de la arquitectura de software.
- Un prototipo ejecutable de la arquitectura.
- Una lista de riesgos y un caso de negocio revisados.
- Un plan de desarrollo detallado que muestre las iteraciones y los criterios de evaluación de cada una de ellas.
- Un caso de desarrollo actualizado, donde se especifique el proceso a ser usado. Se denomina como caso de desarrollo al resultado de configurar o personalizar el Proceso Unificado de Rational. Alternativamente un caso de desarrollo podría ser un sitio Web con ligas a las secciones más relevantes del proceso.
- De manera opcional, se podrá elaborar de forma preliminar un manual de usuario.

Al final de la fase de elaboración el hito que se debe alcanzar es el de la definición de la arquitectura. En este punto, se deberán examinar a detalle los objetivos y el alcance del sistema, la elección de la arquitectura, y la resolución a los principales riesgos.

Los principales criterios de evaluación para la fase de elaboración son los siguientes:

- La visión del producto, los requerimientos y la arquitectura son estables.
- Los elementos de mayor riesgo han sido atacados y resueltos de forma creíble.
- Todos los aspectos significativos de la arquitectura del sistema, así como cierta funcionalidad representativa, han sido evaluadas mediante un prototipo ejecutable.
- El plan para la fase de construcción es lo suficientemente detallado y preciso.
- Se aceptan los gastos reales respecto a los gastos planeados.

Nuevamente el proyecto puede ser abortado o considerablemente replanteado si no se logran estas metas.

### 2.3.3 Fase de Construcción

Los objetivos fundamentales de la fase de construcción son el desarrollo del sistema entero y la garantía de que el producto puede comenzar su transición a los clientes o usuarios finales, es decir, que se ha alcanzado una funcionalidad operativa inicial.

Durante la fase de construcción, el resto de los componentes y demás características de la aplicación son desarrollados e integrados dentro del producto, el sistema es probado en su totalidad. La fase de la construcción es, en un solo sentido, un proceso de manufactura donde el énfasis es puesto en la administración de los recursos para optimizar costos, tiempos y calidad. En este sentido, se lleva a cabo una transición entre el desarrollo de trabajos intelectuales durante las fases de concepción y elaboración, al desarrollo de productos terminados durante las fases de construcción y transición.

Muchos proyectos son tan grandes que será necesario establecer equipos de trabajo en paralelo para llevar a cabo las actividades de la construcción. Estas actividades en paralelo pueden acelerar significativamente la disponibilidad y entrega de versiones; pero también se verá incrementada la complejidad en la administración de recursos y sincronización de flujos de trabajo. Una arquitectura robusta y una planeación razonable serán factores determinantes para el éxito en esta fase. Esta es una razón del por que, durante la fase de elaboración se establecen como criterios de evaluación, el desarrollo de la arquitectura estable y la elaboración de una correcta planeación.

El resultado de la fase de construcción es un producto listo para entregar a los usuarios finales. Consiste, al menos, de lo siguiente:

- El producto de software completo y probado en la plataforma en la que operará.
- Los manuales de usuario.
- Una descripción de la última versión.

Al final de la fase de construcción se debe lograr el hito correspondiente a la capacidad operacional inicial. Esto es, se decide si el software, la infraestructura tecnológica y los usuarios se encuentran listos para iniciar operaciones, sin exponer el proyecto a altos riesgos. La versión del producto que se obtiene en la fase de construcción es a menudo llamada versión "beta".

Los criterios de evaluación para la fase de construcción son los siguientes:

- La versión del producto es lo suficientemente estable y madura para ser desplegada a la comunidad de usuarios. Esto implica que toda la funcionalidad requerida ha sido incorporada en el sistema y que éste ha sido probado exitosamente en el ambiente de desarrollo.
- Los involucrados en el proyecto se encuentran listos para la transición dentro de la comunidad de usuarios.
- Se aceptan los gastos reales respecto a los gastos planeados.

Si el proyecto falla en alcanzar estos criterios de evaluación, será necesario producir más versiones del producto antes de iniciar con la fase de transición.

### 2.3.4 Fase de Transición

El objetivo fundamental de la fase de transición es garantizar que tenemos un producto preparado para su entrega a la comunidad de usuarios. Durante esta fase se enseñan a los usuarios a utilizar el sistema. Una vez que el producto de software ha sido entregado a los usuarios finales, regularmente sucede que se requiere desarrollar nuevas versiones para corregir algunos problemas o terminar funcionalidades que han sido postpuestas.

La fase de transición inicia cuando se cuenta con una versión del producto lo suficientemente madura para ser entregada a los usuarios finales. Para que la transición al usuario puede dar resultados positivos, se requiere que algunas funcionalidades del sistema ya hayan sido terminadas cumpliendo con un nivel aceptable de calidad, así como que la documentación del usuario esta disponible. Esto incluye:

- Una versión beta para validar el nuevo sistema respecto a las expectativas del usuario.
- Una operación paralela con el sistema anterior.
- Conversión de bases de datos operacionales.
- Capacitación a usuarios y personal técnico.

La fase de transición se enfoca en las actividades que son requeridas para poner el software en las manos de los usuarios. Típicamente, esta fase incluye algunas iteraciones, en las que se evalúan, se corrigen y se producen nuevas versiones de tipo beta. El esfuerzo principal se lleva a cabo en el desarrollo de la documentación orientada al usuario, en el entrenamiento a los usuarios, en el soporte técnico en la utilización inicial del producto, y en reaccionar ante la retroalimentación del usuario. En este punto del ciclo de vida, sin embargo, la retroalimentación del usuario debería tratarse principalmente en la afinación, configuración, instalación y uso del producto.

Los principales objetivos de la fase de transición incluyen:

- Entrenamiento a los usuarios sobre el uso del producto.
- Lograr la aceptación, por parte de las diferentes partes involucradas en el proyecto, de que el producto cumple completa y consistentemente con los criterios de evaluación de la visión.
- Iniciar las operaciones con el nuevo sistema.

El hito que se logra al terminar la fase de transición es el de la versión entregable del producto. En este punto, se decide si los objetivos del proyecto fueron logrados, y si será necesario iniciar otro ciclo desarrollo. En algunos casos, este hito puede coincidir con el final de la fase de inicio del próximo ciclo.

Los criterios de evaluación para la fase de transición son los siguientes:

- Satisfacción del usuario con el nuevo sistema.
- Se aceptan los gastos reales respecto a los gastos planeados.

### 2.4 Descripción de la estructura estática del proceso

Anteriormente ya se mencionó que un proceso debería especificar quién hace qué, cómo y cuándo lo hace. RUP lo describe en términos de los siguientes conceptos:

- Artefactos: El qué.
- Roles: El quién.
- Actividades: El cómo.
- Las disciplinas y los flujos de trabajo detallados: El cuándo.

En la figura 2.8 se ilustran los conceptos claves en el Proceso Unificado de Rational y su interrelación. El proceso se desarrolla en cuatro fases, y estas a su vez se dividen en varias iteraciones. En cada iteración se consideran varias disciplinas y estas son descritas en términos de flujos de trabajo detallados. Éstos definen actividades, que son realizadas por algún rol en particular y se producen ciertos artefactos.

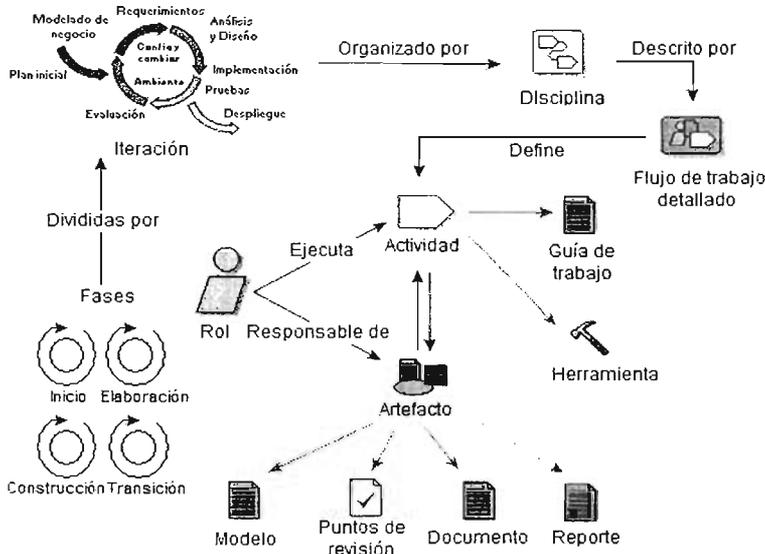


Figura 2.8 Conceptos claves de RUP

Un artefacto es una pieza de información que es producida y/o usada durante la ejecución del proceso. Los artefactos son productos tangibles. Los productos que se entregan a los clientes o usuarios finales son solo algunos de los artefactos que son elaborados durante todo el proceso de desarrollo.

Los artefactos pueden tomar diferentes formas:

- Un modelo, tal como un modelo de casos de uso, un modelo de diseño, etc.
- Un elemento de modelo, tal como un caso de uso, una clase del diseño, los cuales forman parte de un modelo.
- Un documento, tal como el documento de la arquitectura del software o el caso de negocio.
- Código fuente.
- Programas ejecutables.

Los artefactos en RUP han sido organizados dentro de cinco categorías de información como se ejemplifica en la siguiente tabla:

Administración	Requerimientos	Diseño	Implementación	Despliegue
<ul style="list-style-type: none"><li>• Plan de desarrollo del Software</li><li>• Caso de negocio</li><li>• Caso de desarrollo</li><li>• Descripción de la versión</li><li>• Evaluación del estado del proyecto</li></ul>	<ul style="list-style-type: none"><li>• Documento de visión</li><li>• Modelo de casos de uso</li></ul>	<ul style="list-style-type: none"><li>• Modelo de diseño</li><li>• Descripción de la arquitectura</li><li>• Modelo de prueba</li></ul>	<ul style="list-style-type: none"><li>• Código fuente y ejecutable</li><li>• Archivos de configuración y de datos</li></ul>	<ul style="list-style-type: none"><li>• Material de instalación</li><li>• Documentación del usuario</li><li>• Material de capacitación</li></ul>

Una lista de todos los artefactos definidos en RUP se muestra en el anexo C. Los artefactos son responsabilidad de un solo rol. Los roles usan artefactos como insumo para realizar ciertas actividades, y los roles producen o modifican artefactos en el desarrollo de sus actividades.

Los artefactos pueden a su vez ser compuestos por otros artefactos, se pueden generar varias veces a través de múltiples iteraciones y deben ser administrados a través de un control de versiones. También se sugiere que estos no sean impresos en papel, sino que sean mantenidos, y en su caso entregados, en forma digital con el formato de la herramienta con la que fueron creados. Cuando sea necesario, se podrán generar los documentos impresos utilizando la herramienta adecuada. Esto permitirá que la información siempre esté actualizada en base al estado actual del proyecto, y no será necesario realizar algún esfuerzo adicional para producirla.

Un rol define el comportamiento y las responsabilidades de un individuo, o conjunto de individuos trabajando en conjunto como un equipo, dentro del contexto de una organización dedicada al desarrollo de software. Un rol es responsable de uno o más artefactos y ejecuta una serie de actividades.



Es importante enfatizar que los roles no son individuos. Los individuos pueden jugar múltiples roles y múltiples individuos pueden jugar un solo rol. El administrador del proyecto lleva a cabo la asignación de individuos con roles durante la planeación y arranque del proyecto. Una descripción de todos los roles definidos en RUP se puede observar en el anexo B.

Una actividad es una unidad de trabajo compuesta por uno o más pasos, que proporciona un resultado significativo en el contexto del proyecto. Tiene un propósito claro, el cuál regularmente implica la creación o modificación de artefactos. Cada actividad es asignada a un rol específico. Las actividades pueden, al igual que los artefactos, ser repetidas varias veces, especialmente cuando son realizadas en diferentes iteraciones.

El tiempo de ejecución de una actividad es en términos de horas o de unos cuantos días. Una actividad debería ser un elemento de planeación y evaluación del progreso. Si es una actividad muy pequeña, no será significativa, pero si es muy grande, podrá llegar a ser dividida en partes para medir el progreso de los trabajos.

Las actividades son realizadas por pasos, los cuales pueden clasificarse en las siguientes categorías:

- Preparación. El rol entiende la naturaleza de la tarea, obtiene y revisa los artefactos de insumo, y formula los resultados.
- Ejecución. El rol crea o modifica los artefactos requeridos por la actividad.
- Revisión. El rol inspecciona los resultados en base a criterios predeterminados.

Una mera lista de los roles, actividades y artefactos no son suficientes para constituir un proceso. Se necesitan describir secuencias de actividades que produzcan algunos resultados concretos y que muestren la interacción entre roles. Un flujo de trabajo describe esa secuencia de actividades.

No siempre será posible o práctico representar todas las dependencias entre actividades. A menudo, dos actividades llegan a ser más entrelazadas de lo que se muestra, especialmente cuando ellas involucran al mismo rol o al mismo individuo. La gente no son máquinas, y el flujo de trabajo no puede ser interpretado literalmente como un programa que la gente sigue mecánicamente y de forma muy precisa.

En RUP, el proceso describe dos niveles de flujos de trabajo: el nivel de disciplina y el nivel de flujo de trabajo detallado. Las disciplinas son conjuntos de actividades agrupadas lógicamente por área de interés. RUP está organizado mediante nueve disciplinas como se muestra en la figura 2.2, las cuales serán tratadas a detalle más adelante en el apartado 2.5. Estas disciplinas pueden clasificarse en dos categorías:

1. Flujos de trabajo de ingeniería.
  - Modelado de negocio
  - Requerimientos

- Análisis y diseño
  - Implementación
  - Pruebas
  - Despliegue
2. Flujos de trabajo de soporte.
- Administración del proyecto
  - Administración de la configuración y cambios
  - Ambiente

Un flujo de trabajo detallado describe grupos específicos de actividades muy relacionadas dentro de una disciplina. Por ejemplo, las actividades son ejecutadas juntas o por un grupo de personas trabajando juntas en un espacio de trabajo, o bien producen algún resultado intermedio significativo.

Los flujos de trabajo, tanto a nivel de disciplina como a nivel de flujo de trabajo detallado, son descritos usando los diagramas de actividades del Lenguaje Unificado de Modelado. Los diagramas de disciplinas contienen los flujos de trabajo detallados de la disciplina. Dentro de cada flujo de trabajo detallado, las actividades pueden ser ejecutadas en paralelo y cada actividad puede requerir o producir uno o más artefactos. En estos diagramas se muestran los artefactos claves, las actividades y los roles que definen a cada flujo de trabajo.

## ***2.5 Disciplinas del proceso***

### **2.5.1 Administración del proyecto**

La administración de un proyecto de software deberá permitir alcanzar los objetivos del proyecto, administrando los riesgos y superando los obstáculos que se pudieran presentar, con el fin de liberar un producto que satisfaga las necesidades de los clientes y los usuarios finales. El hecho de que pocos proyectos son completamente exitosos es un indicador de lo difícil de esta tarea.

El propósito de esta disciplina es:

- Proveer de una guía para la administración de los proyectos de sistemas de software.
- Proveer de guías prácticas para la planeación, ejecución y monitoreo de los proyectos.
- Proveer de una guía para la administración de riesgos.

Sin embargo, esta disciplina no intenta cubrir todos los aspectos de la administración del proyecto. Esto es, no se cubren aspectos como administración de personal, del presupuesto y de contratos. Esta disciplina se enfoca principalmente en los aspectos más importantes de un proceso de desarrollo iterativo e incremental:

- Administración de riesgos.

- Planeación de un proyecto iterativo, a través del ciclo de vida que se basa en fases y en iteraciones.
- Monitoreo del progreso en el proyecto través de métricas.

Cuando se inicia un proyecto, la disciplina de administración del proyecto empieza en la iteración inicial con concebir el nuevo proyecto, durante el cual se crean las primeras versiones de los artefactos de visión, caso de negocio y lista de riesgos. El objetivo es obtener los suficientes fundamentos para evaluar el alcance y los riesgos del proyecto. Luego entonces se crea un primer plan de desarrollo del software.

Las actividades descritas podrán realizarse en más de una iteración hasta contar con la suficiente información acerca de los riesgos y de la viabilidad del proyecto, como para poder tomar una decisión adecuada entorno a continuar con el resto de la fase de inicio o cancelar el proyecto. En el caso de continuar con el proyecto, se lleva a cabo el plan para la siguiente iteración, en la cual el administrador del proyecto y el arquitecto de software deciden cuales requerimientos se deberán revisar o realizar. En las primeras iteraciones se deberá enfatizar en describir y refinar los requerimientos; en iteraciones posteriores se trabajará en la construcción de esos requerimientos.

En este punto, la disciplina de administración del proyecto sigue una secuencia común para todas las iteraciones subsecuentes. La iteración es ejecutada, revisada y evaluada para determinar si los objetivos preestablecidos para esa iteración han sido logrados. En esta revisión se deberá decidir si el proyecto continúa, o se cancela en el caso de que no se hayan alcanzado los objetivos principales y se determine que éstos ya no podrán alcanzarse durante iteraciones subsecuentes. Además, en paralelo a las actividades descritas, deberá llevarse a cabo el monitoreo y control del proyecto, para conocer el grado de avance del mismo e ir atacando las problemáticas que se vayan presentando.

Antes de planear la siguiente iteración, la visión, la lista de riesgos y el caso de negocio son revisados con la idea de que puedan ser replanteadas algunas expectativas en base a la experiencia lograda en la iteración previa.

Cuando la iteración final de una fase se completa, una revisión de un hito mayor es realizada como parte del cierre de esa fase y se realiza la planeación para la siguiente fase. A la conclusión del proyecto, una revisión de aceptación del mismo es llevada a cabo para terminarlo, a menos que se determine que el producto entregado no es aceptable, en cuyo caso se deberá planear otra iteración.

Para los casos de continuar con otra iteración, se deberá realizar la planeación detallada de la misma. También se deberá actualizar el plan de desarrollo de software así como el plan general del proyecto para el resto de las iteraciones, tomando en cuenta la experiencia adquirida previamente.

## 2.5.2 Modelado del negocio

El propósito del modelado del negocio es:

- Entender la estructura y el dinamismo de la organización para la cual se construye el sistema.
- Entender los problemas actuales en la organización e identificar mejoras potenciales.
- Asegurar que los clientes, usuarios finales y desarrolladores tienen una visión común respecto a la organización.
- Derivar los requerimientos del sistema necesarios para soportar la organización.

Al alcanzar estas metas, la disciplina del modelado de negocio describe como desarrollar una visión de la organización para la cual se construye el sistema, y basada en esta visión define los procesos, los roles y responsabilidades de esa organización en un modelo de casos de uso de negocio y un modelo de objetos de negocio. Adicionalmente, se desarrolla una especificación complementaria de negocio y un glosario de términos.

En la primera iteración se deberá evaluar el estatus de la organización para la cual se construye el sistema. Basado en esta evaluación, se podrá ser capaz de tomar una decisión entorno a continuar con esa iteración, así como determinar la forma en la se trabajarán las subsecuentes iteraciones.

Si se determina que no son necesarios los modelos de negocio completos, sino únicamente un modelo de dominio, se deberá realizar éste como un subconjunto del modelo de objetos de negocio, en donde se abarquen las entidades de negocio.

Si se determina que no habrá cambios significativos en los procesos de negocio, todo lo que se necesita hacer es diagramar los procesos y derivar los requerimientos del sistema. No es necesario mantener algún conjunto determinado de modelos de la organización.

Si se realiza el modelo de negocio con la intención de mejorar o llevar a cabo una reingeniería de los procesos de negocio, se deberá desarrollar tanto el modelo de negocio actual como el modelo al que se pretende llegar.

Si lo que se desea es desarrollar nuevos casos de negocio, se deberán construir modelos para estos nuevos casos de negocio, sin ser necesario describir el modelo actual de negocio.

## 2.5.3 Requerimientos

El propósito de la disciplina de requerimientos es:

- Establecer y mantener un acuerdo con los clientes y demás áreas interesadas en el proyecto.

- Proveer a los desarrolladores de un mejor entendimiento de los requerimientos del sistema.
- Definir los alcances del sistema
- Proveer de una base para planear el contenido técnico de cada iteración.
- Proveer de una base para la estimación de costos y tiempos para el desarrollo del sistema
- Definir una interfaz de usuario para el sistema, tomando en cuenta las necesidades y expectativas de los usuarios.

Para alcanzar estas metas, es importante, primero entender la definición y alcance del problema que se está tratando de resolver con el sistema. Las reglas de negocio, el modelo de casos de uso de negocio y el modelo de objetos de negocio desarrollados en la disciplina de modelado de negocio servirán como un insumo significativo. Las áreas afectadas son identificadas y las peticiones de éstas son obtenidas, registradas y analizadas.

Un documento de visión, un modelo de casos de uso, los casos de uso y una especificación complementaria son desarrollados para describir completamente el sistema, de tal forma que las áreas afectadas, incluyendo clientes y usuarios potenciales, puedan ver estos documentos como una fuente importante de información.

El documento de visión brinda una visión completa para el sistema que se construirá y soporta el contrato entre la organización que requiere el sistema y la que lo desarrollará. Este documento es escrito desde la perspectiva del cliente, se enfoca en las características esenciales del sistema y en los niveles de calidad esperados. Deberá también especificar las capacidades operacionales (volumen, tiempos de respuesta, concurrencia), usuarios del sistema, interfases con otros sistemas.

El modelo de casos de uso deberá servir como un medio de comunicación y como un contrato entre el cliente, los usuarios y los desarrolladores, entorno a la funcionalidad del sistema, lo cual permite que:

- Los clientes y usuarios validen que el sistema llegará a ser como ellos lo esperaban.
- Los desarrolladores construyan el sistema como es esperado.

El modelo de casos de uso consiste de casos de uso y actores. Cada caso de uso en el modelo es descrito a detalle, mostrando paso a paso como el sistema interactúa con los actores, y lo que el sistema hace en el caso de uso. Como ya se mencionó en el apartado 2.2.2, el proceso está dirigido por los casos de uso en el sentido de que éstos son utilizados para el análisis, el diseño, la implementación y las pruebas.

Una completa definición de los requerimientos del sistema descritos en los casos de uso y en las especificaciones suplementarias pueden ser empaquetadas juntas para definir una especificación de requerimientos del software.

Un plan para la administración de requerimientos especifica la información y los mecanismos de control que serán utilizados para medir, reportar y controlar los cambios en los requerimientos del producto.

Adicionalmente a los artefactos mencionados, podrán ser también desarrollados un glosario de términos usados consistentemente a través del proyecto y un prototipo de interfaz de usuario como un mecanismo de retroalimentación para descubrir requerimientos no claros o no definidos.

Los trabajos que se refieren a analizar el problema y a entender las necesidades de las áreas interesadas, son de mayor peso durante la fase de inicio del proyecto, en tanto que los que se refieren a definir el sistema son de mayor peso durante la fase de elaboración. Las tareas de administración del alcance del sistema y de administración de cambios en los requerimientos, son realizadas constantemente a través de todas las fases.

## 2.5.4 Análisis y diseño

Los propósitos del análisis y diseño son:

- Transformar los requerimientos establecidos en el dominio del problema a modelos de diseño en el dominio de la solución.
- Definir una arquitectura robusta.
- Adaptar el diseño a restricciones de rendimiento y de ambiente.

En la fase de inicio, el análisis y diseño se centra en establecer la viabilidad técnica del proyecto. Si los riesgos son pocos, como por ejemplo que la plataforma tecnológica o el sistema no son nuevos, se pueden omitir esos trabajos en esa fase.

Al comienzo de la fase de elaboración, los trabajos de análisis y diseño se enfocan en definir una arquitectura candidata para el sistema, con el fin de contar con un punto de partida para los principales trabajos del análisis. Si la arquitectura ya existe, como por ejemplo por que fue producida en iteraciones previas o en algún proyecto anterior, el enfoque de estos trabajos cambia hacia afinar la arquitectura y hacia analizar el comportamiento del sistema en base a los casos de uso para ir obteniendo los elementos de diseño.

Después de que los elementos iniciales son identificados, se podrán diseñar componentes que definan una funcionalidad bien delimitada y encaminada a proporcionar los comportamientos apropiados para satisfacer los requerimientos del sistema. En paralelo con estas actividades, se realiza el diseño de la base de datos.

El resultado del análisis y diseño es un conjunto inicial de componentes, los cuales serán afinados más adelante en la implementación. Como artefactos se obtiene un modelo de diseño y opcionalmente un modelo de análisis. El primero sirve como una abstracción del código fuente y consta del diseño de clases estructuradas, del diseño de paquetes y del

diseño de subsistemas con interfases bien definidas, en los que se representan los componentes que serán implementados. También contiene la descripción, de la manera en la que los objetos de ese diseño de clases colaboran para realizar los casos de uso.

## 2.5.5 Implementación

El propósito de la implementación es:

- Definir la organización del código, en términos de subsistemas organizados en capas.
- Implementar clases y objetos en términos de componentes.
- Probar los componentes desarrollados como unidades.
- Integrar los resultados producidos por individuos o equipos de desarrollo, dentro de un sistema ejecutable.

En cuanto a pruebas, en la disciplina de implementación sólo se definen las pruebas unitarias de clases individuales. Las pruebas de sistema y de integración son descritas en la disciplina de pruebas.

En los comienzos de la fase de elaboración se lleva a cabo la estructura del modelo de implementación. Para cada iteración, desde la fase de elaboración, se deberá planear la integración, implementar los componentes, integrar cada subsistema y finalmente integrar todo el sistema. Los dos últimos trabajos serán evaluados mediante las pruebas de integración.

## 2.5.6 Pruebas

La disciplina de pruebas representa un área de servicio y de retroalimentación para otras disciplinas. Esta disciplina se enfoca en evaluar la calidad del producto a través de varias acciones:

- Encontrar y documentar los defectos en el software.
- Validar que los requerimientos hayan sido implementados apropiadamente.
- Validar que el producto funciona conforme fue diseñado.
- Verificar la interacción entre objetos, así como la integración de todos los componentes del software.

Con el desarrollo iterativo, se pueden encontrar los defectos del sistema lo más rápido posible, lo cual lleva radicalmente a una reducción de costos en la reparación del sistema. Las pruebas se deberán realizar en tres aspectos: robustez, funcionalidad y rendimiento.

Una diferencia interesante entre la disciplina de pruebas y las otras disciplinas es que la de pruebas es una tarea que se avoca a encontrar y exponer las debilidades del producto. Se deberán realizar demostraciones concretas del producto y evaluaciones imparciales del mismo, cuidando ni de proteger tanto al proyecto ni de llegarlo a criticar de forma muy

negativa o destructiva. Adoptar una actitud negativa no permitirá nunca alcanzar algún nivel de aceptación del mismo.

De acuerdo con algunas mediciones, el costo de las pruebas se estima entre un 30 y un 50% del total del costo del desarrollo del software. Esto tal vez no sea creíble por la gente que piensa que el software no estuvo bien probado antes de su liberación. Esto tiene sus razones. En primera, las pruebas del software son bastante complejas: las diferentes opciones y combinaciones que un programa informático puede tener simplemente no se pueden cuantificar. Segundo, las pruebas típicamente se realizan sin alguna metodología clara, por lo que los resultados varían de proyecto en proyecto y de organización en organización. Tercero, no se utilizan herramientas de automatización de pruebas que permitan hacer manejable los laboriosos aspectos de las pruebas.

En tanto que la flexibilidad y la complejidad del software hacen imposible llevar a cabo las pruebas de todos los escenarios posibles, una metodología bien concebida y el uso de herramientas para automatizarlas, pueden ayudar a mejorar la productividad y efectividad de las pruebas del software.

Para sistemas de carácter crítico, donde una falla puede llegar a dañar a personas, un software de alta calidad es esencial para el éxito del sistema. Sin embargo, en esta era de la información en donde se observa una demanda creciente en servicios de información a través de Internet, muchos sistemas de éste tipo llegan a considerarse de misión crítica, debido a que las fallas en ellos pueden llegar a representar valiosas pérdidas económicas.

Una verificación continua de la calidad, iniciada tempranamente en el ciclo de vida, puede significar la disminución de costos en la terminación y mantenimiento del software. Esto también reduce los riesgos asociados con la liberación de un producto de baja calidad.

### **2.5.7 Administración de la configuración y cambios**

El objetivo de la administración de la configuración y cambios, es controlar los cambios y mantener la integridad de los artefactos del proyecto. Implica lo siguiente:

- Identificar los elementos de configuración.
- Restringir los cambios a estos elementos.
- Auditar los cambios realizados a estos elementos.
- Definir y administrar la configuración de estos elementos.

Los métodos, procesos y herramientas utilizadas por una organización, para administrar los cambios pueden ser integrados en un sistema para ese fin. Mediante este sistema se podrá manejar la información clave acerca del desarrollo del producto, acerca de los procesos de despliegue y mantenimiento, y acerca de los artefactos potencialmente reutilizables. Un sistema de este tipo es esencial y representa una parte integral para todos los procesos del desarrollo.

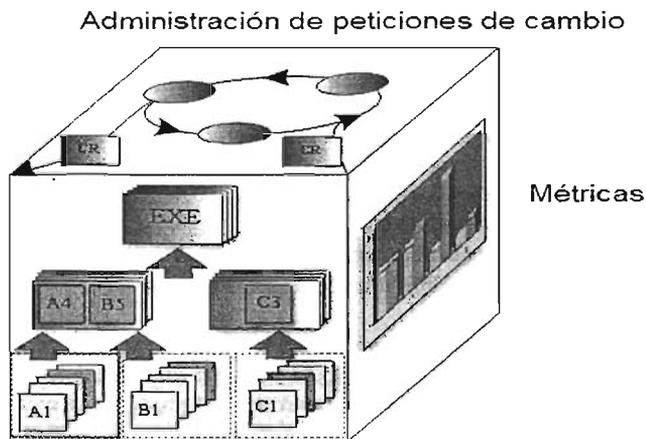


Lo que se pretende en esta disciplina es controlar los numerosos artefactos producidos por mucha gente que labora en un proyecto en común. Este control permitirá evitar confusiones costosas y asegurar que los artefactos resultantes no estén en conflicto debido a algunos de los siguientes problemas:

- Actualización simultánea. Cuando dos o más personas trabajan separadamente en el mismo artefacto, los últimos cambios destruyen el trabajo del otro compañero.
- Notificación limitada. Cuando un problema es corregido en artefactos compartidos por varios desarrolladores, y algunos de ellos no son notificados del cambio.
- Múltiples versiones. La mayoría de los programas grandes son desarrollados en varias versiones. Una versión podría estar en uso por los clientes, otra en pruebas y una tercera aún en desarrollo. Si los problemas son encontrados en algunas de esas versiones, la necesidad de corregir puede ser propagada entre ellas. La confusión puede llevar a costosas correcciones y re trabajos, a menos que los cambios sean cuidadosamente controlados y administrados.

Un sistema de administración de cambios proporciona los siguientes beneficios:

- Soporta los métodos de desarrollo.
- Mantiene la integridad del producto.
- Provee de un ambiente estable, dentro del cual se desarrolla el producto.
- Restringe los cambios a los artefactos, de acuerdo con las políticas del proyecto.
- Provee de un mecanismo para registrar y consultar quién, cuándo y por qué modifico algún artefacto.
- Controla el desarrollo del proceso en si mismo: registra quién creo una versión determinada, cuándo y por qué, y mantiene los fuentes tal y como se encontraban en la construcción de esa versión.



**Administración de la configuración**

Figura 2.9 Los tres aspectos de la administración de la configuración y cambios

La administración de la configuración y cambios cubre tres aspectos interdependientes. Estos pueden ser ilustrados usando el cubo de la figura 2.9. El cubo tiene tres caras, y cada cara examina un diferente aspecto del problema:

- La administración de la configuración se refiere a la estructura del producto: identificación de artefactos, versiones, dependencias entre artefactos, identificación de configuraciones, etc. También se encarga del mantenimiento a los espacios de trabajo para individuos y para equipos para que puedan ellos desarrollar sus trabajos sin entorpecerse unos con otros.
- La administración de las peticiones de cambio se refiere a la estructura del proceso: registra y administra las solicitudes de cambio. Junto con el análisis se determina el impacto en los cambios y se realiza el seguimiento de lo que sucede con éstos hasta que son integrados al sistema.
- Las métricas se refieren a la estructura del control del proyecto: muchas de las peticiones de cambio son registradas en una base de datos, y representan gran parte del trabajo a ser realizado durante una iteración, especialmente en las fases de construcción y transición. De esta base de datos es posible extraer información del comportamiento del proyecto:
  - Progreso en general del proyecto en lo que se refiere a los cambios.
  - Número de cambios realizados y su estatus.
  - Tiempo en el que los cambios permanecen en un mismo estatus.

### 2.5.8 Ambiente

La disciplina de ambiente se enfoca en las actividades necesarias para configurar el proceso para un proyecto. Describe las actividades requeridas para desarrollar las guías de trabajo que soportaran un proyecto. El propósito de esta disciplina es proveer de un ambiente de desarrollo de software, tanto del proceso como de las herramientas, al equipo de desarrollo para soportar su trabajo. Este soporte incluye lo siguiente:

- Selección y adquisición de herramientas.
- Instalación y configuración de herramientas para el equipo de desarrollo.
- Configuración del proceso.
- Mejoramiento del proceso.
- Servicios técnicos para soportar el proceso: infraestructura tecnológica, administración de usuarios, respaldos, etc.

El proceso real usado en el proyecto es descrito en un caso de desarrollo, el cual describe la forma en la que el proyecto aplicará el Proceso Unificado de Rational.

En las primeras iteraciones del proyecto, se inicia el flujo de trabajo realizando la actividad de preparar el ambiente para el proyecto, el cual tiene como insumo principal una evaluación de la organización de desarrollo. Luego entonces, para cada iteración, se ejecutan las actividades de preparar el ambiente para una iteración y de prepara las guías de trabajo para una iteración.

## 2.5.9 Despliegue

El propósito de la disciplina de despliegue es producir exitosamente versiones del producto, para ponerlo a disposición de los usuarios finales. Esto implica varias actividades, entre las más importantes se encuentran:

- Producir versiones del software para un ambiente de producción.
- Empaquetamiento del software
- Distribución del software
- Instalación del software
- Capacitar y asistir a los usuarios finales
- Planear y conducir pruebas beta
- Migración de datos o de software existentes
- Aceptación formal del producto

Aunque las actividades de despliegue son la mayoría de las veces centradas en la fase de transición, muchas de las actividades se deberán incluir en fases anteriores para preparar el despliegue al final de la fase de construcción.

La forma en la que son llevadas a cabo las actividades en esta disciplina depende del tamaño del proyecto, el modo de liberación y el contexto de negocio. Existen varias formas de ejecutar el despliegue de un proyecto y dependerá principalmente del grado de participación de la organización de desarrollo en el empaquetamiento y distribución del producto de software, así como de la forma en la que el usuario final aprenderá a utilizarlo.

# Estrategia de Implementación del Proceso

En este capítulo se describe la personalización del proceso de ingeniería para los trabajos de desarrollo y mantenimiento del SIIRFE. Cabe mencionar que la estrategia propuesta es de tipo incremental en tres etapas, es decir, en cada una de las etapas se integran nuevos flujos de trabajo, artefactos y roles. Así mismo, se presenta un plan general con fases e iteraciones para cada etapa del proyecto.

### **3.1 Continuidad al desarrollo y mantenimiento del SIIRFE**

Como se mencionó en el apartado 1.3.3, actualmente se tiene la necesidad de continuar con los trabajos de desarrollo y mantenimiento del SIIRFE a partir de las últimas versiones liberadas al ambiente de producción. Con estos trabajos de mantenimiento se podrán iniciar las actividades de soporte a los procesos operativos centrales de la Dirección Ejecutiva del Registro Federal de Electores. Se debe mencionar además que las versiones en referencia no están libres de errores y los ciclos de operación no se encuentran totalmente automatizados.

Con esos trabajos se pretende, primero, lograr la estabilización de la funcionalidad que opera actualmente, y segundo, comenzar con los trabajos referentes a los casos de uso que no fueron construidos correspondientes a la primera fase de desarrollo de SIIRFE. En el presente capítulo se trata de bosquejar de forma muy concreta una estrategia para llevar a cabo los trabajos de desarrollo y mantenimiento del SIIRFE.

Desde una perspectiva global se plantea dar seguimiento y continuidad al proyecto SIIRFE a través de la inclusión de cuatro áreas de competencia, las cuales contribuirán, desde su ámbito de responsabilidad, al buen desenvolvimiento del proyecto. Estas áreas de competencia se describen a continuación:

1. **Gerencia del proyecto.** Se propone formar un equipo de trabajo que se encargue de llevar a cabo las actividades de gestión que permitan facilitar los trabajos propios del proyecto. Entre otras cosas, dicho equipo de trabajo deberá llevar a cabo la administración de riesgos, el seguimiento a los planes de trabajo que se definan, la coordinación de reuniones entre distintas áreas, la identificación oportuna de necesidades, la gestión ante las áreas administrativas en lo relativo a la contratación de personal, adquisición de equipo de cómputo, adecuación de espacios, etc. Este grupo será el primer responsable de mantener informado del avance del proyecto a las instancias superiores del Instituto o en su caso, al comité directivo que se defina para tal efecto.
2. **Desarrollo del sistema.** El equipo de desarrollo estará integrado dentro de la estructura de la Dirección de Desarrollo de Sistemas, apoyado por trabajos de consultoría especializadas en la plataforma tecnológica utilizada en el SIIRFE. Los trabajos propiamente de desarrollo de software estarán enmarcados dentro del Proceso Unificado de Rational.
3. **Supervisión del proyecto.** Se formará un grupo de trabajo integrado por personal de la UNICOM, cuyas labores serán de seguimiento y supervisión a lo largo del desarrollo del proyecto. Entre sus principales funciones, éste grupo deberá de vigilar que se cumpla con la normatividad vigente que en materia de informática se ha establecido en el Instituto, deberá vigilar que los trabajos de desarrollo de sistemas se realicen de la mejor manera posible y siguiendo las políticas establecidas, y deberá de informar oportunamente de los riesgos que se presenten durante el desarrollo del proyecto, sus impactos y posibles alternativas de solución.

4. Apoyo en la infraestructura tecnológica. Se contará con apoyo de la Dirección de Soporte Técnico y Dirección de Operaciones de CECYRD, quienes se encargarán de proveer de la infraestructura tecnológica y soporte requerido a lo largo del desarrollo del proyecto.

En la figura 3.1 se podrá observar la manera en la que esos cuatro grupos de trabajo se integran en la estructura organizacional del Instituto.

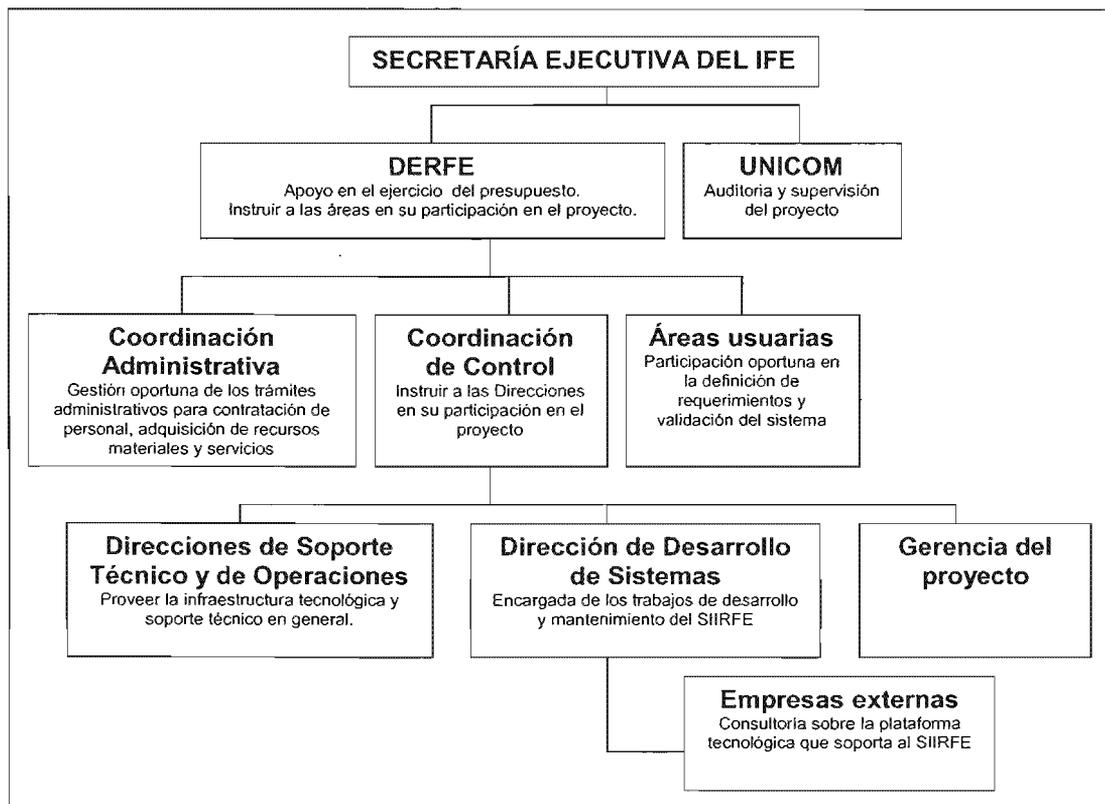


Figura 3.1 Estructura jerárquica de las distintas áreas de competencia involucradas en el proyecto SIIRFE.

Entre las principales ventajas respecto al hecho de que sea la Dirección de Desarrollo de Sistemas quien continúe con los trabajos de desarrollo del SIIRFE, se encuentran las siguientes:

- Se continúa adquiriendo el conocimiento técnico del sistema, que permitirá una completa autonomía a la DERFE en su operación. Esto es, los trabajos de

mantenimiento al sistema podrán llevarse a cabo por el mismo Instituto sin depender de un tercero.

- Hasta ahora se han podido llevar a cabo ciertos trabajos de mantenimiento y soporte al subsistema que opera en los MAC, por lo que se podrán complementar los esfuerzos de desarrollo de forma más eficiente.
- Es el personal técnico de la DERFE quien tiene el conocimiento de las actividades y procedimientos llevados a cabo respecto del proyecto, así como la experiencia adquirida durante la instrumentación del Programa de Modernización Tecnológica y Operativa, y en particular del SIIRFE.
- Se podrá acelerar la implementación de un proceso de ingeniería de software en la Dirección de Desarrollo de Sistemas, que permitirá realizar los trabajos de desarrollo con base a las mejores prácticas utilizadas en la industria, para construir sistemas de forma más eficiente y de mayor calidad.

Sin embargo, no se podrán ignorar aspectos relevantes que se han suscitado con el nuevo modelo operativo y con la consolidación de los Centros Regionales de Cómputo, como lo son las cargas de trabajo adicionales en las áreas técnicas del Registro Federal de Electores, sin que se haya podido concretar un alineamiento organizacional que permita hacer frente a las nuevas necesidades.

Por una parte será necesario hacer ciertos ajustes al interior de la Dirección de Desarrollo de Sistemas, de tal forma que, al destinar recursos a la continuidad en el desarrollo del SIIRFE, no se descuide el soporte a la operación diaria, ni a ciertos proyectos relevantes que lleva a cabo la DERFE. En específico, se requiere realizar la reasignación de varias actividades de índole operativo que viene realizando la Dirección de Desarrollo de Sistemas, entre las que destacan el centro de atención a usuarios como segundo nivel, la elaboración de formatos y archivos de impresión para listados nominales, generación de insumos para programas de detección de duplicados, etc.

Asimismo, y aunque actualmente ya se han llevado a cabo ciertos trabajos de mantenimiento al subsistema SIIRFE-MAC, se debe considerar que es poco el personal de la Dirección de Desarrollo de Sistemas que ha estado participando en los trabajos de desarrollo del SIIRFE, y que se ha tenido poca interacción práctica con las herramientas y el lenguaje de desarrollo. Esto debido a que la mayor parte del personal tuvo que estar dedicado a actividades del Proceso Electoral Federal 2002-2003 durante el desarrollo del SIIRFE, por lo que dicho personal tendrá que transitar por una curva de aprendizaje que nos permita obtener resultados más rápido y más eficientemente.

Por lo expuesto anteriormente será necesaria la contratación adicional de personal que cuente con un perfil técnico acorde con las tecnologías utilizadas en el SIIRFE, así como la contratación de empresas especializadas en la plataforma tecnológica que soporta al SIIRFE.

### **3.2 Planeación de un desarrollo iterativo e incremental**

En la estrategia aquí propuesta para dar continuidad al desarrollo y mantenimiento del SIIRFE, se considera que el desarrollo deberá llevarse a cabo de forma iterativa e incremental, con lo que se pretende, entre otras cosas, reducir los riesgos que conlleva el mantenimiento de un sistema de tal magnitud.

En principio se han definido cuatro grandes etapas, o momentos que marcarán el nivel de madurez en los trabajos de mantenimiento del SIIRFE:

1. Etapa de “estabilización”. Se pretende llevar a cabo la corrección de las incidencias más significativas que se tienen actualmente. El tiempo estimado para llevar a cabo estos trabajos es de tres meses.
2. Etapa de “consolidación”. Se pretende concluir con las principales funcionalidades pendientes entorno a la operación. El tiempo estimado para llevar a cabo estos trabajos es de cuatro meses.
3. Etapa de “terminación de la primera fase de desarrollo”. Se pretende desarrollar los casos de uso que no fueron construidos correspondientes a la primera fase del desarrollo del SIIRFE. El tiempo estimado para llevar a cabo estos trabajos es de alrededor de doce meses.
4. Etapa “complementaria o segunda fase de desarrollo”. Se pretende desarrollar los tres subsistemas del SIIRFE que no fueron objeto de la primera fase del desarrollo: subsistema de actualización y control documental, subsistema de soporte y subsistema para interoperabilidad con fuentes externas. Dependiendo de los alcances que se definan para cada uno de esos subsistemas, es que se podrán estimar los tiempos de desarrollo.

Se debe entender además que, en cada una de las etapas a partir de la denominada como “consolidación”, se estarán llevando cabo labores de mantenimiento, como corrección a incidencias, controles de cambio y nuevos requerimientos, que se deberán atender en su momento acorde con las necesidades propias de la operación de la DERFE.

En paralelo y conforme a las mismas etapas descritas, se pretende ir implementando de forma incremental un proceso de ingeniería de software basado en el Proceso Unificado de Rational, con el fin de lograr que el desarrollo de sistemas sea una actividad controlada, eficaz y eficiente.

El Proceso Unificado de Rational define una serie de actividades debidamente detalladas, artefactos, plantillas, guías de trabajo, puntos de revisión y ejemplos. Sin embargo, todo el proceso tal cual es descrito no deberá ser aplicado a cualquier proceso de desarrollo de sistemas. Esto es, RUP fue diseñado como un proceso en general que deberá ser configurado y personalizado a las necesidades particulares que se puedan presentar para alguna área dedicada al desarrollo de software.



De hecho, RUP adicionalmente cuenta con una guía para la personalización del proceso, esto es, RUP contiene información detallada de cómo adaptar el proceso a un proyecto en específico, tipo de solución, u organización. La guía de personalización del proceso está contemplada en la disciplina de “ambiente”, y será en base a los trabajos enmarcados en dicha disciplina mediante los cuales se irá configurando e implementando el proceso de ingeniería. Se debe mencionar además que RUP cuenta con herramientas que permiten precisamente configurar el proceso y publicarlo en un servidor para que pueda ser consultado por todo el equipo de trabajo.

Por ahora se considera que el nivel de implementación del proceso de ingeniería alcanzado en la tercera etapa denominada como de “terminación de la primera fase de desarrollo”, será el que se ocupe para llevar a cabo la última etapa denominada como “segunda fase de desarrollo”. De esta manera, la propuesta concreta consiste en implementar una metodología de trabajo basada en RUP, configurándola de forma incremental para adaptar el proceso a las características planteadas en cada una de las primeras tres etapas descritas anteriormente, y haciendo uso del conjunto de las herramientas que existen actualmente.

En el proyecto aquí planteado, se podrán observar tres tipos de planes:

- Plan de ciclos de desarrollo, en el que se muestren las cantidades de versiones que se planean liberar al ambiente de producción.
- Plan de fases, en el que se muestren las cuatro fases que se definen en RUP para cada ciclo de desarrollo.
- Plan de iteraciones, en los que se define, por cada iteración, el detalle de las actividades que se realizarán.

El plan de ciclos de desarrollo, en nuestro caso también llamados ciclos de evolución por tratarse de un sistema que ya se encuentra en operación, quedarán definidos, por ahora, solo para las tres primeras etapas, ya que aún no se tiene delimitado un alcance para el resto de los subsistemas que conforman al SIIRFE. El plan de ciclos de desarrollo se muestra en la figura 3.2

Etapa	Ciclo de desarrollo	MESES																		
		E	F	M	A	M	J	J	A	S	O	N	D	E	F	M	A	M	J	
Estabilización	1	■	■	■	■	■	■													
Consolidación	1																			
Terminación	1																			
	2																			

Figura 3.2 Plan de ciclos de desarrollo para las tres primeras etapas.

El plan de fases es un plan en lo general, que define los tiempos para cada una de las fases de RUP: inicio, elaboración, construcción y transición. En el proyecto aquí presentado, deberá existir un plan de fases para cada ciclo de desarrollo.

En un plan de fases se deberán observar las siguientes metas por alcanzar:

- Los objetivos del desarrollo. Al final de la fase de inicio, el proyecto deberá estar bien fundamentado y bien delimitado.
- La arquitectura del desarrollo. Al final de la fase de elaboración, la arquitectura deberá estar completa.
- La capacidad inicial de operación. Al final de la fase de construcción, se deberá contar con una versión beta del producto.
- Versión a liberar del producto. Al final de la fase de transición y, por ende, del ciclo de desarrollo, se deberá contar con una versión del sistema en el ambiente productivo.
- Los objetivos de cada iteración. Una iteración comienza con la planeación y levantamiento de requerimientos y termina con una versión del producto, ya sea interna o externa.

El plan de fases es producido al principio del proyecto, durante la fase de inicio, y deberá ser actualizado conforme sea necesario. En la figura 3.3 se muestra, como una propuesta inicial, un resumen del plan de fases para cada ciclo de desarrollo.

<b>Etapa de Estabilización.</b>			
<b>Versión única.</b>			
Fase	Número de iteraciones	Periodo de ejecución	
		Inicio	Fin
Inicio	0		
Elaboración	1	Semana 1	Semana 2
Construcción	1	Semana 3	Semana 10
Transición	1	Semana 11	Semana 13

<b>Etapa de Consolidación.</b>			
<b>Versión única.</b>			
Fase	Número de iteraciones	Periodo de ejecución	
		Inicio	Fin
Inicio	0		
Elaboración	1	Semana 1	Semana 2
Construcción	2	Semana 3	Semana 14
Transición	1	Semana 15	Semana 17

<b>Etapa de Terminación.</b>			
<b>Versión uno.</b>			
Fase	Número de iteraciones	Periodo de ejecución	
		Inicio	Fin
Inicio	1	Semana 1	Semana 2
Elaboración	2	Semana 3	Semana 7
Construcción	2	Semana 8	Semana 23
Transición	1	Semana 24	Semana 26

Etapa de Terminación. Versión dos.			
Fase	Número de iteraciones	Periodo de ejecución	
		Inicio	Fin
Inicio	1	Semana 1	Semana 2
Elaboración	2	Semana 3	Semana 7
Construcción	2	Semana 8	Semana 23
Transición	1	Semana 24	Semana 26

Figura 3.3 Un extracto del plan de fases para cada ciclo de desarrollo. El plan de fases es parte del artefacto “Plan de desarrollo de software”.

Idealmente, una iteración debería llevarse a cabo en un periodo de dos a seis semanas, pero esto varía dependiendo del proyecto. En principio dependerá del tamaño de la organización, por ejemplo:

- Cinco personas pueden hacer un plan el lunes por la mañana, tener un desayuno juntos cada día para monitorear el progreso, reasignar tareas, integrar clases y paquetes el jueves, y completar la iteración el viernes.
- Con veinte personas, el escenario es un poco más difícil. Tomará más tiempo distribuir el trabajo, sincronizar subgrupos, e integrar. Una iteración podría tomar de tres a cuatro semanas.
- Con cuarenta personas, tomará una semana en bajar la información, se tienen niveles intermedios de administración, y el entendimiento de los objetivos requerirá de documentación formal. Tres meses es un tiempo razonable para una iteración.

Otros factores son importantes como: la experiencia que se tenga en la organización del desarrollo iterativo, la estabilidad y madurez de la organización, el nivel de automatización para administrar código, distribuir información y ejecutar pruebas. También se debe considerar que una iteración demandará un cierto tiempo para actividades de planeación, sincronización y análisis de resultados. Convencidos del tremendo beneficio del desarrollo iterativo, se deberá buscar la forma de iterar tomando en cuenta las limitantes de la organización y de las personas.

A menudo, en la fase de inicio, no hay una iteración real; no hay un software producido, y solo hay actividades de planeación. En algunos casos, sin embargo, una iteración en esta fase permite: construir un prototipo para mitigar riesgos fuertes en cuestión de tecnología, o para convencimiento del proyecto, o bien se puede acelerar el uso de herramientas y adopción del proceso de ingeniería de software en la organización de desarrollo. En la fase de inicio no se pretende generar código, por lo que se puede tener cero o una iteración.

En la fase de elaboración, se deberá planear al menos una iteración. Si no se cuenta con una arquitectura de partida y se presenta una gran cantidad de factores adicionales, como nuevo personal, nuevas herramientas, nueva plataforma tecnológica o lenguaje de programación, entonces deberá ser necesario planear dos o hasta tres iteraciones. En esta fase será necesario mostrar un prototipo a los clientes o usuarios finales para ayudarlos a definir de

mejor forma sus requerimientos, así como también se deberán mitigar los riesgos en cuanto a la arquitectura. Para esta fase se podrán definir de una a tres iteraciones.

En la fase de construcción, se deberá planear al menos una iteración. Dos es más razonable si se desea explotar los beneficios del desarrollo iterativo y lograr un mejor trabajo de integración y de pruebas. Para proyectos más complejos, tres o más iteraciones son mejor si la organización de desarrollo lo puede soportar y si hay un suficiente nivel de automatización y madurez del proceso. Para esta fase se recomienda de una a tres iteraciones.

En la fase de transición, se deberá planear al menos una iteración. A menudo, la baja calidad de la versión inicial del producto forzarán a realizar más de una iteración. Para esta fase se recomienda de una a dos iteraciones.

Se puede decir, conforme a los tres niveles mostrados en la figura 3.4, que una cantidad normal de iteraciones estará en el rango de  $6 \pm 3$  iteraciones [10].

Nivel	Cantidad de iteraciones por fase				Total de iteraciones
	Inicio	Elaboración	Construcción	Transición	
Bajo	0	1	1	1	3
Medio	1	2	2	1	6
Alto	1	3	3	2	9

Figura 3.4 Cantidad estimada de iteraciones por fase, de acuerdo con el nivel de complejidad del proyecto.

El plan de iteración es un plan detallado de cada iteración. Un proyecto generalmente tiene dos planes de iteración “activos” al mismo tiempo:

- El plan de la iteración actual.
- El plan de la iteración siguiente, el cual deberá ser construido durante la segunda mitad de la iteración actual, y estar listo al final de esa misma iteración.

El plan de iteración es construido usando las técnicas y herramientas tradicionales de planeación, y en él se definen las tareas y su debida asignación a personas y equipos de trabajo. El plan contiene importantes fechas, tales como la generación de las principales versiones, la entrega de componentes de otros fabricantes, etc.

Debido a que el proceso iterativo es dinámico y esto significa que se pueden realizar cambios en las metas y tácticas, no se deberá gastar una cantidad de tiempo mayor a lo normal para producir planes con mayor detalle. Tales planes son difíciles de mantener, rápidamente quedan obsoletos, y típicamente son ignorados en la operación de la organización. El plan de iteración requiere de una inversión de tiempo aceptable, y mantiene un nivel de detalle lo suficientemente adecuado para hacer un buen trabajo de planeación y seguimiento.

### 3.3 Etapa de estabilización

El objetivo de esta etapa es liberar una versión al ambiente productivo que permita eliminar las incidencias más significativas en un plazo relativamente corto. Para la ejecución de estos trabajos, será de gran importancia el contar con el apoyo de consultorías especializadas en la arquitectura del sistema. Entre otras cosas, se pretende lograr el establecimiento del entorno de trabajo y procedimientos básicos para controlar el desarrollo de software.

En cuanto al alcance en la parte de implementación, a continuación se enlistan, muy brevemente y por subsistema, una serie de mejoras o de correcciones a desarrollar.

- a) SIIRFE-MAC.
  - Al capturar un trámite y este es identificado plenamente se debe recalcular la clave de elector.
  - Al capturar un trámite y este es identificado plenamente se debe obtener su código postal.
  - Antes de mandar un Formato Único de Actualización con Recibo (FUAR) a imprimir, el sistema debe de realizar las validaciones de tipo de movimiento.
  - Incorporar la funcionalidad necesaria para que al capturar algún apellido no se acepte más de dos "X".
  - En el apartado de "escolaridad y ocupación" se debe mostrar la clave correspondiente a cada descripción.
  - Revisar que las incidencias relacionadas a la falta de información ya estén corregidas
  - Optimizar el uso de memoria para los objetos de tipo "credencial" y con ello evitar las caídas del sistema por esta razón.
  - Permitir realizar una copia de algún respaldo parcial.
  - Cargar correctamente los archivos de notificación en MAC para clarificar FUAR.
  - Ordenar la generación de archivos de notificación con base a la jerarquía de los estatus.
  - Actualizar el proceso de lectura de archivos de notificación con base a la nueva nomenclatura.
  
- b) SIIRFE-CECYRD, Subsistema de Actualización al Padrón Electoral.
  - Regeneración de los archivos de notificaciones, construyendo archivos por alguna secuencia en particular, agregando la secuencia en el nombre del archivo.
  - Control de carga de los archivos de transacciones, registrando en archivos de bitácora los tipos y cantidades de notificaciones, lo cual permitirá la verificación de la carga total de los archivos de transacciones.
  - Adecuar la programación en el desencolado de trámites para que se permitan cargar las notificaciones no importando la secuencia.

- Garantizar el envío de las notificaciones para la clarificación de la información en los MAC.
  - Análisis y afinación del procedimiento para la generación de lotes de producción y ordenes de servicio, con el fin de mejorar el desempeño de la aplicación.
- c) SIIRFE-CECYRD, Subsistema de Actualización Cartográfica.
- Incorporación de remesas extraordinarias con fecha del día en curso.
  - Incorporación de remesas ordinarias.
  - Corregir archivo de respuesta de ciudadanos mal referenciados, ya que no coinciden los totales reportados.
  - Incluir, en el archivo para la Vocalía Estatal, la clave de la entidad.
- d) SIIRFE-CECYRD, Subsistema de consultas
- Agregar la función que permita el cambio de contraseña de usuarios.
  - Corregir la impresión oficial del ciudadano.
  - Corregir el ordenamiento en la impresión de la lista de registros localizados.
  - Corregir el ordenamiento en la impresión de la lista de registros no localizados.
- e) SIIRFE-Depuración.
- Corrección de la generación de las cédulas de confronta para la verificación en campo.
  - Corrección del listado de registros a verificar en campo por entidad.
  - Integración del sistema de comparación de imágenes.
  - Análisis y afinación del procedimiento de búsqueda inicial en la funcionalidad de depuración preventiva.

Para los flujos de trabajo de ingeniería definidos en el Proceso Unificado de Rational, se consideran dos grupos de trabajo: uno dedicado al SIIRFE-MAC, el cual prácticamente es el mismo grupo que viene trabajando actualmente con este módulo, y otro dedicado al SIIRFE que opera en CECYRD, un grupo conformado exclusivamente para iniciar con los trabajos de mantenimiento tanto de SIIRFE-CECYRD como de SIIRFE-Depuración. No se pretenden llevar a cabo actividades relacionadas con “Modelado de negocio”, “Requerimientos” y “Análisis y diseño”, ya que sólo se corregirán incidencias existentes.

Para los flujos de trabajo de soporte, se propone que se consideren las actividades como apoyo a todos los subsistemas y módulos de manera horizontal, para ello se proponen inicialmente dos grupos de trabajo: uno dedicado a la administración del proyecto, y otro dedicado a soportar el ambiente de desarrollo y llevar a cabo la administración de la configuración y cambios. En la figura 3.5 se muestra la estructura de trabajo para esta etapa.

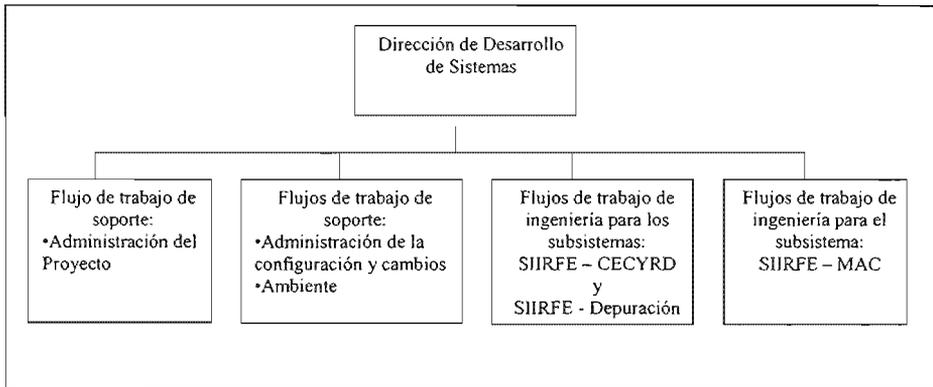


Figura 3.5 Estructura organizacional para atender los flujos de trabajo definidos en RUP para la etapa de “estabilización”.

A continuación se detalla, en cada una de las nueve disciplinas, el alcance en la implementación del proceso de ingeniería basado en RUP para la etapa de estabilización.

#### 1. Modelado de Negocio.

No se implementa en esta etapa, ya que sólo se corregirán incidencias existentes y por lo tanto no se requiere llevar a cabo trabajos de modelado de negocio.

#### 2. Requerimientos.

No se implementa en esta etapa, ya que sólo se corregirán incidencias existentes y por lo tanto no se requiere llevar a cabo trabajos de levantamiento de requerimientos. La retroalimentación que se deba tener con los usuarios se hará conforme el tratamiento de incidencias que se defina en la disciplina de pruebas.

#### 3. Análisis y Diseño.

No se implementa en esta etapa, ya que sólo se corregirán incidencias existentes y por lo tanto no se requiere llevar a cabo trabajos de análisis y diseño de nuevas funcionalidades.

#### 4. Implementación.

Esta disciplina se refiere a los trabajos de programación del sistema, por lo que deberá seguirse desde la etapa de estabilización. Además de las actividades enmarcadas en los flujos de trabajo, se pretende que el equipo de desarrollo se apegue a ciertas políticas de control de versiones, así como otras que permitan mantener un adecuado control en los cambios de las estructuras de datos.

**ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA**

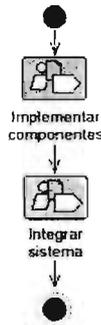


Figura 3.6 Flujo de trabajo de la disciplina de implementación.

En el flujo de trabajo “Implementar componentes” es donde se llevan a cabo los trabajos de programación, generando las clases y los paquetes a fin de que estos puedan ser integrados en un solo aplicativo. Para ello también será necesario llevar a cabo las pruebas unitarias que garanticen que los componentes producidos, al menos de forma aislada, funcionan adecuadamente.

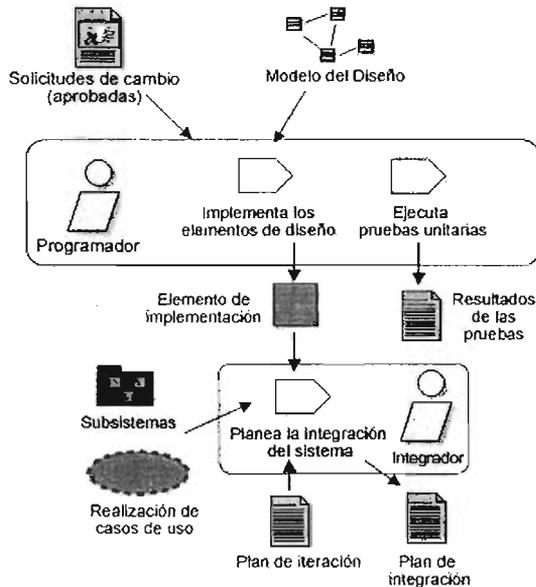


Figura 3.7 Flujo de trabajo “Implementar componentes”.

El flujo de trabajo “Integrar sistema” tiene como objetivo integrar los trabajos de desarrollo de los múltiples programadores para crear una nueva y consistente versión del sistema. Para ello también será necesario definir y ejecutar pruebas integrales del sistema.



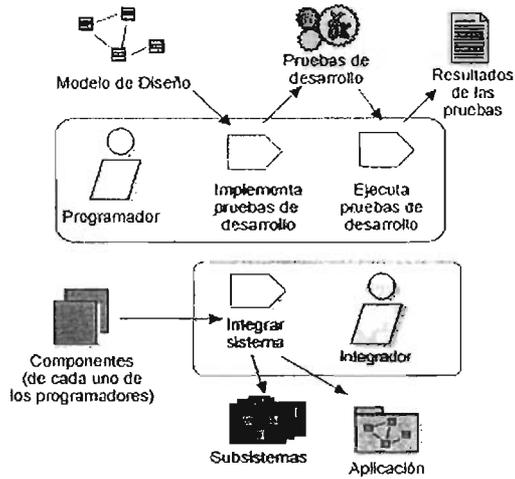


Figura 3.8 Flujo de trabajo "Integrar sistema".

## 5. Pruebas.

Para esta etapa, se pretende establecer los flujos de trabajo básicos para el desarrollo de las pruebas. Como herramienta para registrar y mantener un control de las incidencias reportadas se propone hacer uso del "Rational Clear Quest".

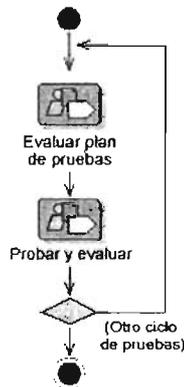


Figura 3.9 Flujo de trabajo de la disciplina pruebas.

El propósito del flujo de trabajo “Evaluar plan de pruebas” es desarrollar y en su caso, evaluar y mejorar el plan de pruebas. Esto es importante por que regularmente se reutilizan los insumos, casos de pruebas y demás definiciones del plan de pruebas, en subsecuentes ciclos de pruebas.

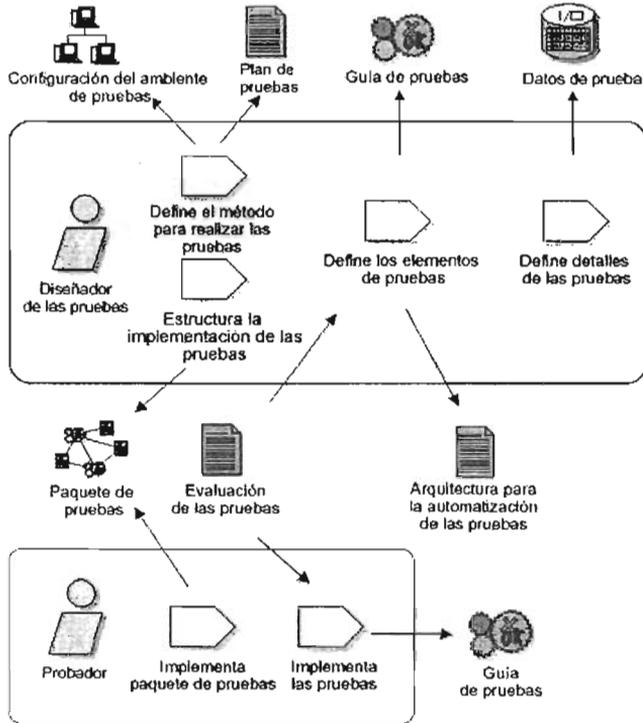


Figura 3.10 Flujo de trabajo “Evaluar plan de pruebas”.

En el flujo de trabajo “Probar y evaluar” es donde se realizan las actividades de implementación, ejecución y evaluación de las pruebas del sistema. Como productos intermedios de estas actividades se tienen que definen las pruebas: una descripción de los casos de pruebas, una guía y un paquete de pruebas. Los artefactos finales de este flujo de trabajo son: los resultados de pruebas donde se registren las incidencias encontradas, las peticiones de cambio generadas, y una evaluación de las pruebas que permita al equipo de desarrollo adecuar sus esfuerzos hacia los objetivos de la iteración actual.

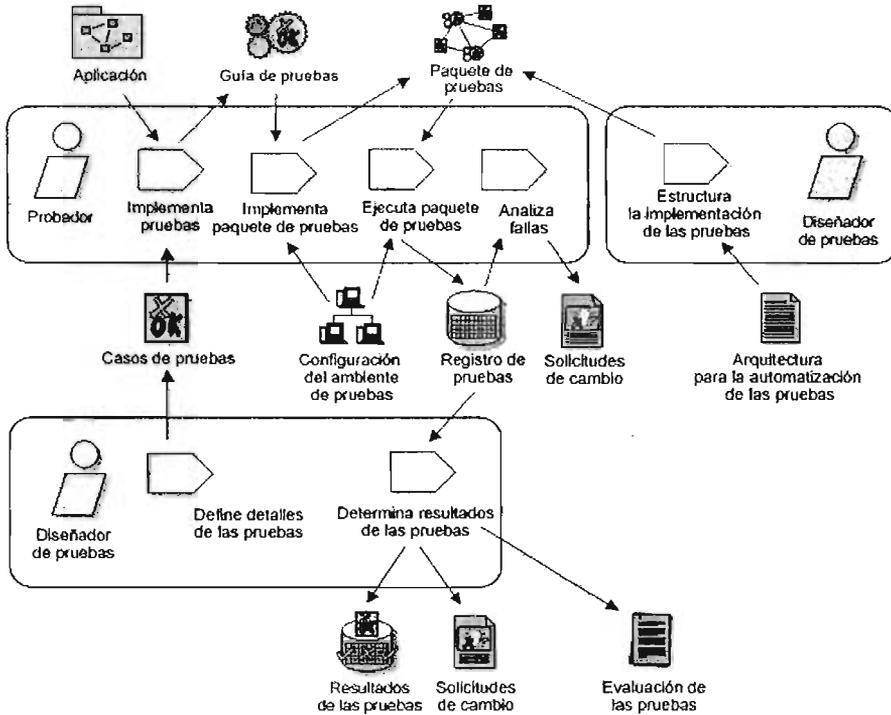


Figura 3.11 Flujo de trabajo "Probar y evaluar".

## 6. Despliegue.

Actualmente se han instrumentado mecanismos de evaluación para la liberación de nuevas versiones del subsistema SIIRFE-MAC. Al tener cientos de centros operativos a nivel nacional, se vuelve necesaria la liberación en forma escalonada, que permita mitigar un posible impacto negativo a nivel nacional respecto a alguna falla en el sistema. Para ello, primero se llevan a cabo pilotos en determinados Módulos de Atención Ciudadana del Distrito Federal, y posteriormente se lleva a cabo un piloto a nivel entidad.

Los flujos de trabajo que se implementarán se refieren a actividades que de alguna forma ya se vienen realizando. El compromiso será el de enmarcarlas dentro del proceso de ingeniería de software.

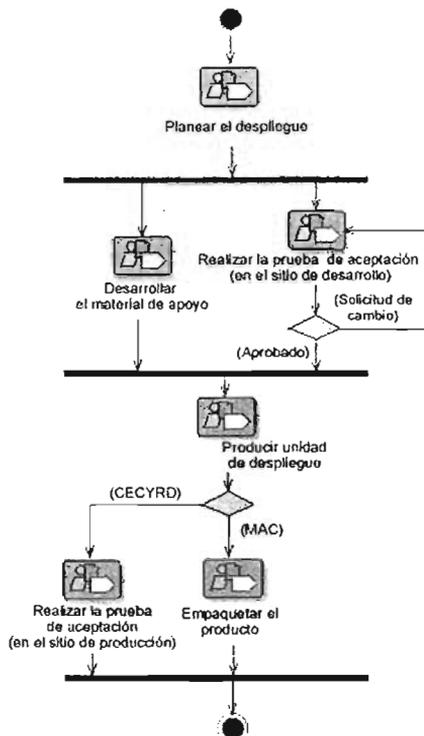


Figura 3.12 Flujo de trabajo de la disciplina de despliegue.

El propósito del flujo de trabajo “Planear el despliegue” es planear las actividades de capacitación, así como la puesta en operación del producto. Esto es, cómo y cuándo el producto estará disponible para el usuario final. Los principales artefactos de este flujo de trabajo son el plan de despliegue y la lista de materiales que permitan a los usuarios finales un adecuado uso del sistema.

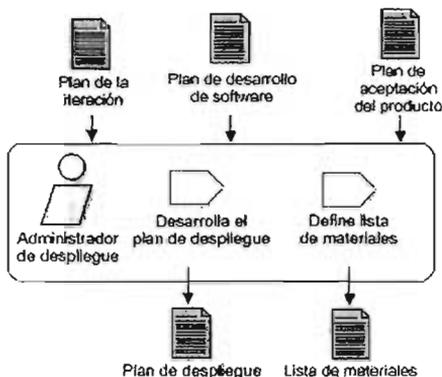


Figura 3.13 Flujo de trabajo “Planear el despliegue”.

En el flujo de trabajo “Desarrollar el material de apoyo” es donde se produce el material necesario para que el usuario final pueda operar adecuadamente el sistema. Se deberán producir tanto materiales de capacitación como manuales de usuario.

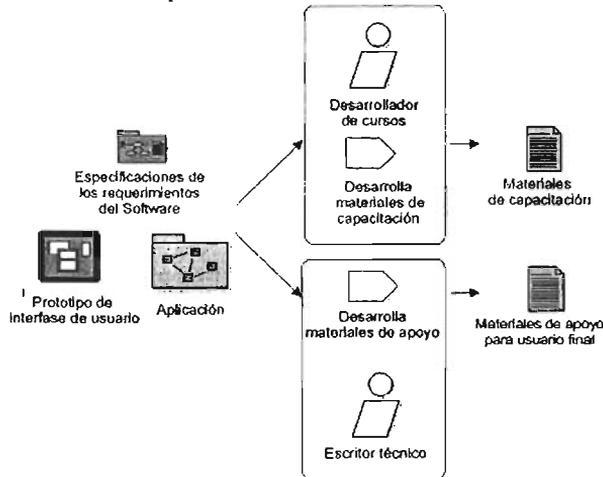


Figura 3.14 Flujo de trabajo “Desarrollar el material de apoyo”.

El flujo de trabajo “Realizar la prueba de aceptación en el sitio de desarrollo” tiene como objetivo asegurar que el producto es aceptado por las áreas usuarias. Para ello se deberá llevar a cabo una prueba de aceptación en un ambiente destinado para tal fin. En ese sentido es que se requerirá el apoyo de la figura “Administrador del sistema” quien será el encargado de tener el control de la configuración de los diferentes ambientes del proyecto: el de desarrollo, el de pruebas y el de producción.

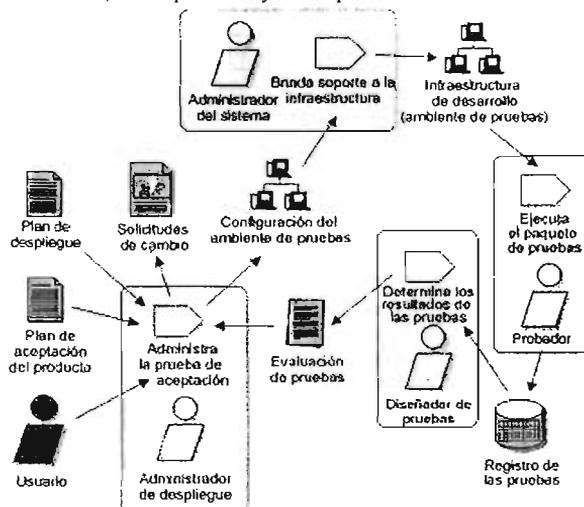


Figura 3.15 Flujo de trabajo “Realizar prueba de aceptación en el sitio de desarrollo”.

El propósito del flujo de trabajo “Producir unidad de despliegue” es crear un paquete en el que se incluya tanto el software como el material de soporte, que permitan que se instale y se use adecuadamente el producto.

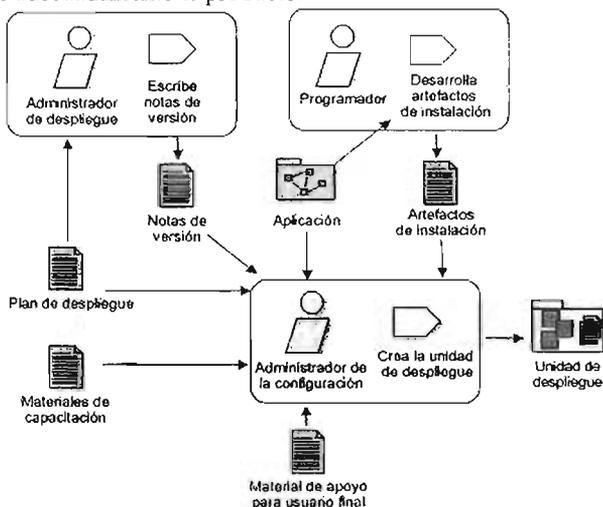


Figura 3.16 Flujo de trabajo “Producir unidad de despliegue”.

El flujo de trabajo “Realizar la prueba de aceptación en el sitio de producción” tiene como objetivo asegurar que el sistema funciona correctamente en el ambiente de producción. Para ello, recién instalada la nueva versión, se deberán verificar todas las funcionalidades críticas del sistema. Este flujo de trabajo se realizará para los subsistemas del SIIRFE que se ubican en el CECYRD, por ahora SIIRFE-CECYRD y SIIRFE-Depuración.

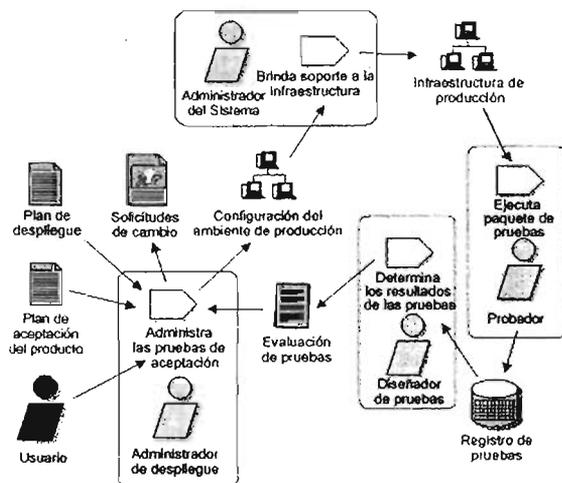


Figura 3.17 Flujo de trabajo “Realizar prueba de aceptación en el sitio de producción”.

El flujo de trabajo “Empaquetar el producto” tiene como propósito describir las actividades necesarias para crear un paquete que contenga el software, las guías de instalación y los manuales de usuario, para que pueda ser distribuido a todos los usuarios finales. En nuestro caso, nos referimos a la creación y la verificación del paquete de instalación del subsistema SIIRFE-MAC, el cual deberá ser distribuido a todos los MAC para su instalación.



Figura 3.18 Flujo de trabajo “Empaquetar el producto”.

## 7. Administración de la configuración y cambios

Para esta etapa se pretende llevar a cabo los flujos de trabajo básicos para la administración de la configuración. Además el equipo de desarrollo deberá utilizar una herramienta para el control de versiones, que permita llevar un control adecuado de las modificaciones de todos los programas y artefactos producidos durante el proyecto.



Figura 3.19 Flujo de trabajo de la disciplina de administración de la configuración.

El propósito del flujo de trabajo “Crear el ambiente para la administración de la configuración” es crear los repositorios y espacios de trabajo, donde el conjunto de artefactos del proyecto pueda ser producido, empaquetado y puesto a disposición de las áreas que así lo requieran.

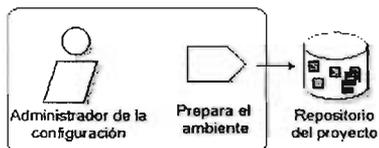


Figura 3.20 Flujo de trabajo “Crear el ambiente para la admón. de la configuración”.

En el flujo de trabajo “Realizar y registrar los cambios” es donde se llevan a cabo las actividades involucradas desde la creación inicial de los artefactos en un espacio de trabajo para tal fin, hasta la liberación y puesta a disposición de las últimas versiones de esos artefactos al equipo del proyecto. Se propone que en este flujo de trabajo también se pongan en práctica políticas de acceso a los repositorios del proyecto, procedimientos de respaldo de esos repositorios, identificación y organización de artefactos, así como procedimientos para la identificación y liberación de versiones.

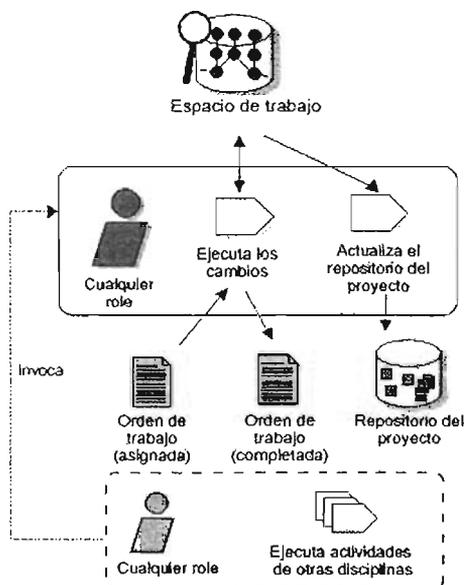


Figura 3.21 Flujo de trabajo “Realizar y registrar los cambios”.



## 8. Administración del proyecto.

Para la etapa de estabilización se proponen seguir los flujos de trabajo necesarios para la administración de las actividades por iteraciones y por fases. Esto es, manejar la estructura dinámica del proceso de ingeniería de software.

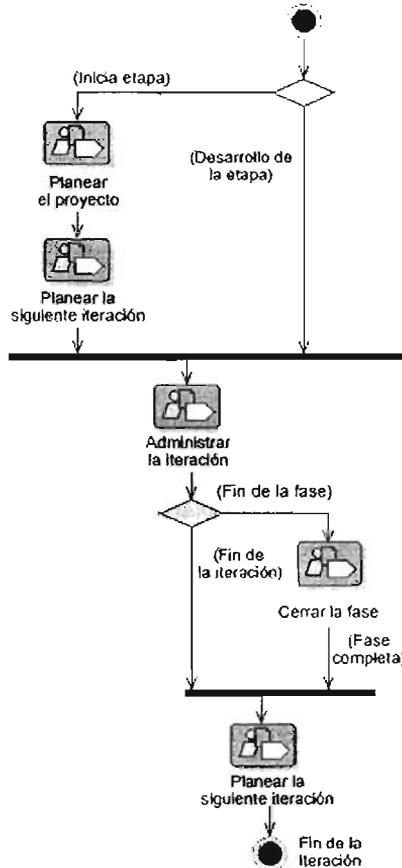


Figura 3.22 Flujo de trabajo de la disciplina administración del proyecto.

El propósito del flujo de trabajo "Planear el proyecto" es desarrollar el plan de desarrollo de software para cada una de las tres etapas. Es por ello que se propone que este documento se genere en la fase de inicio de cada ciclo de desarrollo. Este documento además representa la base para desarrollar una planeación de actividades más a detalle que se documenta en el plan de iteración. Otros de los principales artefactos que se producen con este flujo de trabajo son: el plan para la administración de riesgos, el plan de aceptación del producto y el plan de aseguramiento de la calidad.

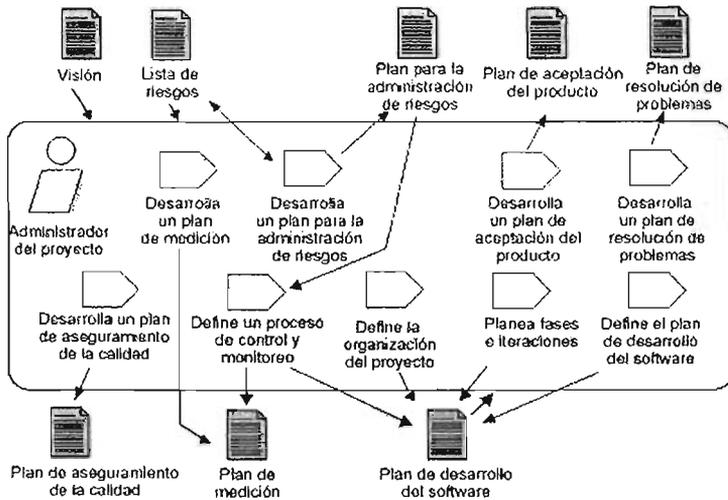


Figura 3.23 Flujo de trabajo "Planear el proyecto".

Es en el flujo de trabajo "Planear la siguiente iteración" donde se crea el plan de la iteración, el cual es un plan a detalle de las actividades a seguir en la siguiente iteración. Este flujo de trabajo regularmente se lleva a cabo al final de la iteración actual y sirve como insumo para la siguiente iteración; la excepción es al inicio del ciclo de desarrollo, donde el plan de iteración se realiza al comienzo de la primera iteración.

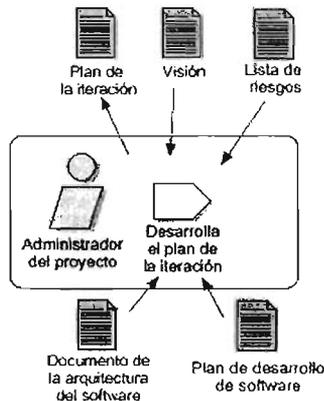


Figura 3.24 Flujo de trabajo "Planear la siguiente iteración".

El flujo de trabajo "Administrar la iteración" contiene las actividades que dan inicio y fin a la iteración actual. Se detona la adquisición de los recursos necesarios para la ejecución de las actividades enmarcadas en la iteración actual, se asigna el trabajo, y finalmente se realiza una evaluación de los resultados donde se determina si fueron o no alcanzados los objetivos de esa iteración.

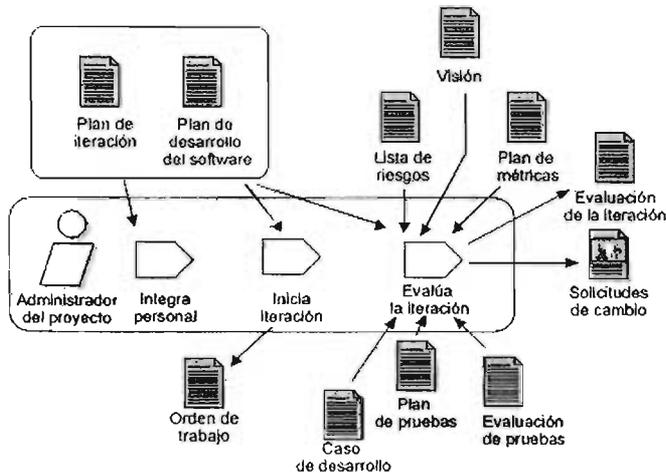


Figura 3.25 Flujo de trabajo “Administrar la iteración”.

El propósito del flujo de trabajo “Cerrar la fase” es concluir en lo formal una de las cuatro fases definidas en RUP, asegurándose que se han alcanzado los objetivos planteados para esa fase.



Figura 3.26 Flujo de trabajo “Cerrar la fase”.

## 9. Ambiente.

Todos los flujos de trabajo de la disciplina de ambiente, que define el Proceso Unificado de Rational, son necesarios de llevar a cabo desde la etapa de estabilización, ya que proporcionan el entorno y la infraestructura requeridos para soportar al resto de las actividades.

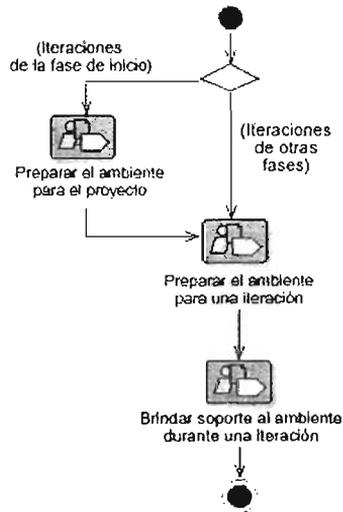


Figura 3.27 Flujo de trabajo de la disciplina de ambiente.

El propósito del flujo de trabajo “Preparar el ambiente para el proyecto” es establecer a detalle el proceso de desarrollo en cada una de las etapas definidas del proyecto de continuidad al desarrollo del SIIRFE. También se consideran actividades para proporcionar las herramientas de software requeridas por el proyecto, así como la preparación de plantillas de los diversos artefactos a producir. Como uno de los objetivos particulares de la etapa de estabilización, se tiene el de establecer un entorno homogéneo de desarrollo.

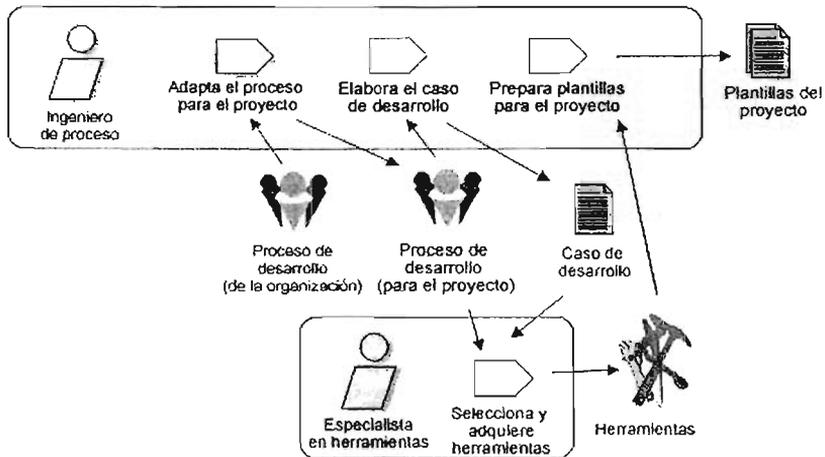


Figura 3.28 Flujo de trabajo “Preparar el ambiente para el proyecto”.

El flujo de trabajo “Preparar el ambiente para una iteración” tiene como objetivo asegurar que el ambiente del proyecto se encuentra listo para la iteración que inicia. Esto abarca tanto la instalación de las herramientas de software como lo referente al proceso de desarrollo. El principal artefacto producido en este flujo de trabajo es el caso de desarrollo en la que se basará la iteración presente.

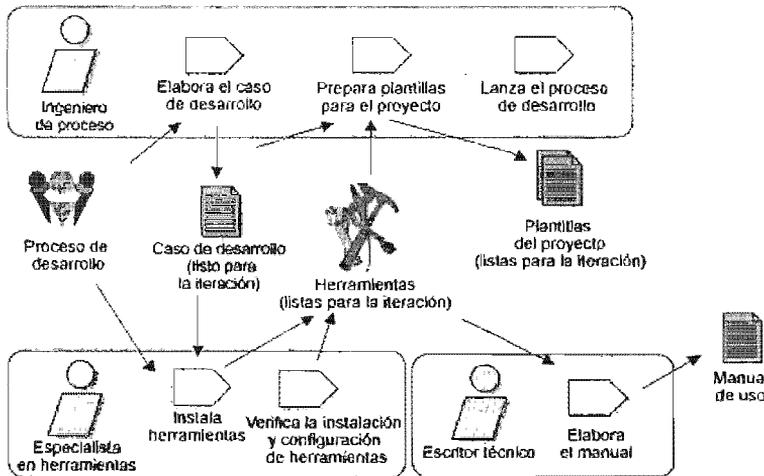


Figura 3.29 Flujo de trabajo “Preparar el ambiente para una iteración”.

El flujo de trabajo “Brindar soporte al ambiente durante una iteración” se refiere a las actividades permanentes que realiza el administrador del sistema, para brindar el soporte requerido por el equipo de desarrollo en cuanto a la instalación y uso de las herramientas, así como la instalación, configuración y operación de los ambientes de desarrollo y de pruebas. Esto permitirá que los miembros del proyecto realicen sus actividades eficientemente sin algún contratiempo originado por cuestiones de infraestructura.

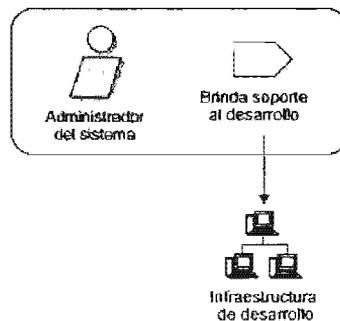


Figura 3.30 Flujo de trabajo “Brindar soporte al ambiente durante una iteración”.

### **3.4 Etapa de consolidación**

El objetivo de esta etapa es liberar una versión al ambiente productivo que permita poner en operación las principales funcionalidades que se encuentran implementadas en el sistema de forma parcial. Con ello lo que se busca es cerrar ciclos de operación. Será importante que, al iniciar con las actividades referentes a esta etapa, se lleve a cabo la incorporación de personal adicional al grupo de desarrollo, con la finalidad de contar con un equipo más robusto que permita hacer frente a los trabajos de implementación tanto de esta etapa como de la siguiente.

Entre otras cosas, se pretende lograr que el equipo de trabajo domine la arquitectura del sistema y pueda incorporar funcionalidades completas bajo un entorno administrado y controlado.

En cuanto al alcance en la parte de implementación, a continuación se enlistan, muy brevemente y por subsistema, una serie de mejoras y de funcionalidades por construir.

- a) SIIRFE-MAC.
  - Revisar y terminar el caso de uso “Mostrar alertas”.
  - Revisar y terminar los principales reportes de MAC.
  - Incorporar el código de identificación de la credencial en las funcionalidades siguientes: retirar credenciales, resguardar credenciales, recibir credenciales resguardadas, reportar robo o extravío parcial de credenciales.
  - Administración de módulos de atención ciudadana.
- b) SIIRFE-CECYRD, Subsistema de Actualización al Padrón Electoral.
  - Validación de Movimientos posteriores.
  - Afinación de funcionalidades para la implementación del proceso de calidad en las credenciales.
  - Afinación de funcionalidades para la implementación del proceso de distribución.
  - Administración de módulos de atención ciudadana.
- c) SIIRFE-CECYRD, Subsistema de Actualización a la Cartografía Electoral.
  - Actualización del calendario de remesas.
  - Corrección de reportes estadísticos.
  - Corrección a la actualización de los ciudadanos mal referenciados en cuanto a datos nulos.
  - Generar ciudadanos mal referenciados sin producir un reporte CIF-05 para trámites con cartografía inválida a nivel de manzana.
- d) SIIRFE-CECYRD, Subsistema de consultas
  - Agregar la función de impresión al detalle del trámite.
  - Incorporar un usuario específico para el tipo de consulta de IFETEL.
  - Corregir la información de la lista de registros localizados por mosaico.

- Corregir las impresiones de bajas.
  - Consultar operación de módulos a nivel nominativo.
- e) SIIRFE-Depuración.
- Resolver la problemática que se presenta para casos de movimiento posterior.
  - Registro de errores en archivos de Bitácoras.
  - Controles de Cambio de Depuración Preventiva como la incorporación de datos adicionales en la búsqueda inicial y exhaustiva.

Para los flujos de trabajo de ingeniería definidos en el Proceso Unificado de Rational, con respecto a la etapa anterior, se llevará a cabo la separación de los subsistemas SIIRFE-CECYRD y SIIRFE-Depuración. Esto es, considerando el SIIRFE-MAC, se tendrán ahora tres grupos de desarrollo.

Para los flujos de trabajo de soporte, en esta etapa se consolidan en uno solo. Así pues, las disciplinas de administración del proyecto, ambiente y administración de la configuración y cambios, serán llevados a cabo por un mismo grupo de trabajo. Nuevamente se considera que esas actividades apoyan a todos los subsistemas de forma horizontal. En la figura 3.31 se muestra la estructura de trabajo para esta etapa.

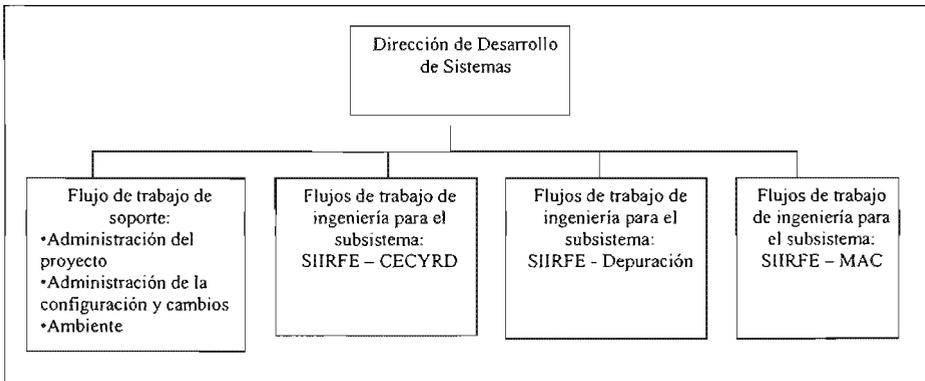


Figura 3.31 Estructura organizacional para atender los flujos de trabajo definidos en RUP para la etapa de “consolidación”

A continuación se detalla, en cada una de las nueve disciplinas, el alcance en la implementación del proceso de ingeniería basado en RUP para la etapa de consolidación.

#### 1. Modelado de Negocio.

Los trabajos a realizar en esta disciplina son con el objeto de encontrar áreas de mejoras en el uso del sistema, esto es, áreas que no se han incorporado en el flujo de trabajo a través de SIIRFE, así como la identificación de ciertos procesos que no fueron

considerados dentro del sistema y que son susceptibles de sistematizar. Para esta etapa, la meta será actualizar la documentación respecto a los procesos del Registro Federal de Electores.



Figura 3.32 Flujo de trabajo de la disciplina modelado de negocio.

El propósito de realizar el flujo de trabajo “Evaluar estado de negocio” será la evaluación de los procesos del Registro Federal de Electores, identificar áreas de mejoras, y actualizar la documentación que haya respecto a la arquitectura y reglas de negocio.

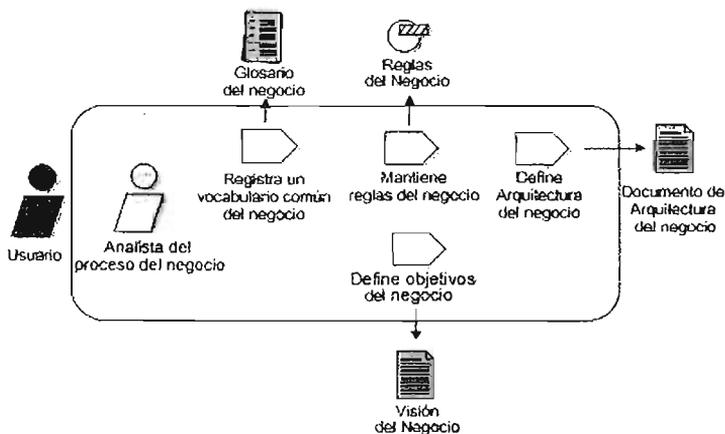


Figura 3.33 Flujo de trabajo “Evaluar estado de negocio”.

El propósito de realizar el flujo de trabajo “Identificar los procesos de negocio”, será actualizar la documentación que se tenga respecto a los procesos del Registro Federal



de Electores, e identificar aquellos que requieran descripciones a más detalle para su documentación en el flujo de trabajo “Refinar definiciones de los procesos de negocio”.

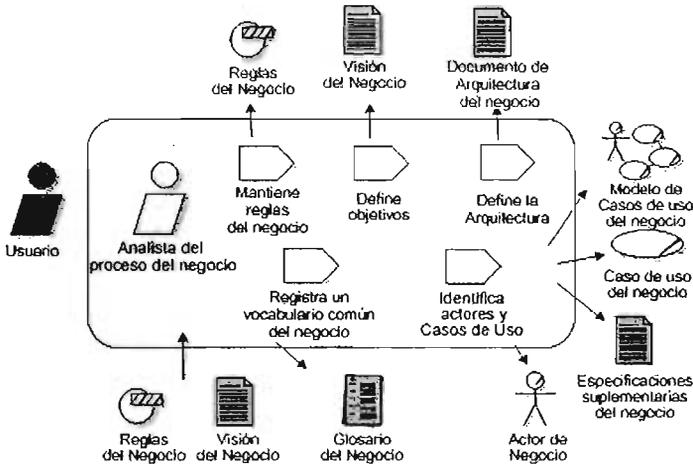


Figura 3.34 Flujo de trabajo “Identificar los procesos de negocio”.

El propósito de realizar el flujo de trabajo “Refinar definiciones de los procesos de negocio” será el de detallar la descripción de los procesos que así se determine en el flujo de trabajo “Identificar los procesos de negocio”, describiendo la forma en la que soportan la misión del Registro Federal de Electores.

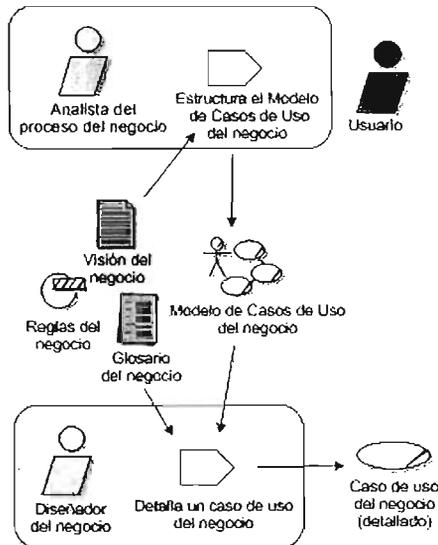


Figura 3.35 Flujo de trabajo “Refinar definiciones de los procesos de negocio”.

## 2. Requerimientos.

Los trabajos a realizar en esta etapa tienen como principal objetivo, sentar las bases para llevar una buena administración de requerimientos, a través de una serie de artefactos definidos para tal fin, así como el uso de una herramienta de software que permita llevar el registro y el control de cada requerimiento.

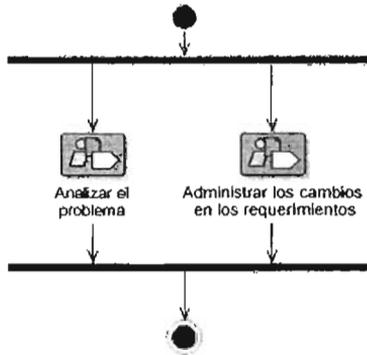


Figura 3.36 Flujo de trabajo de la disciplina de requerimientos.

El propósito de implementar el flujo de trabajo “Analizar el problema” es el de implementar una estrategia de trabajo para la administración de los requerimientos. Además se deberá delimitar el alcance del desarrollo e identificar las áreas usuarias y demás afectadas por el proyecto. Como resultados importantes se deberá construir un glosario de términos, se deberá definir un plan para la administración de requerimientos, y se deberá implantar el uso de la herramienta de software “Requisite Pro”.

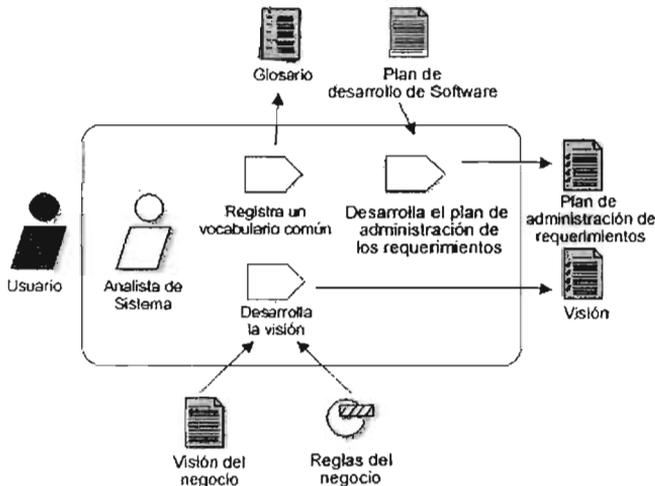


Figura 3.37 Flujo de trabajo “Analizar el problema”.

Respecto al flujo de trabajo “Administración de cambios en requerimientos”, se deberán realizar las siguientes definiciones por el analista de sistema, las cuales quedarán integradas en el artefacto “Plan para la administración de requerimientos”:

- Definición de un esquema de control de cambios,
- Definición de un formato para las solicitudes de cambio, y
- Definición de un mecanismo para identificar dependencias que permita medir el impacto en cascada de las peticiones de cambio.

### 3. Análisis y Diseño.

Las actividades que se deberán realizar en esta disciplina para la etapa de consolidación, permitirán documentar los principales modelos, tales como el modelo de análisis, el modelo de diseño, el modelo de interfase de usuario y el modelo de datos, producto de las actividades de análisis y diseño de las funcionalidades a desarrollar.

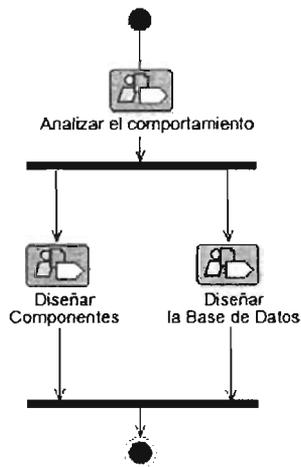


Figura 3.38 Flujo de trabajo de la disciplina de análisis y diseño.

El propósito del flujo de trabajo “Analizar el comportamiento” es transformar el comportamiento del sistema, definido por los requerimientos, en un conjunto de elementos que servirán de base para los trabajos de diseño. Los principales artefactos generados en este flujo de trabajo son el modelo de análisis, el prototipo de interfase de usuario y el modelo de diseño.

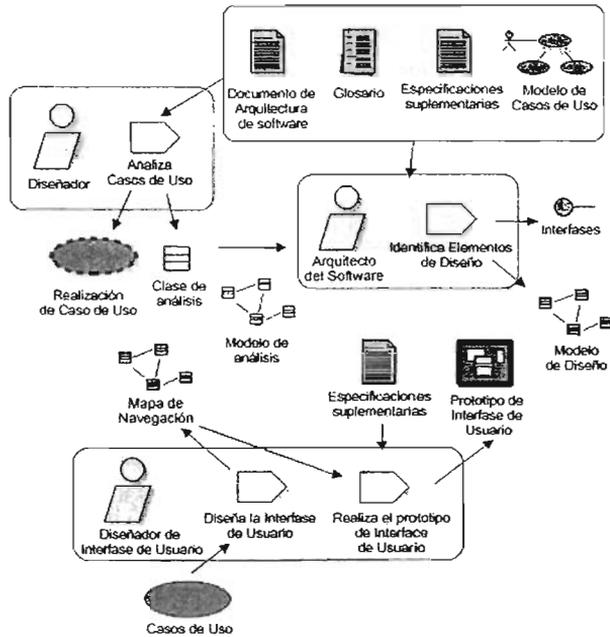


Figura 3.39 Flujo de trabajo “Analizar el comportamiento”.

El objetivo de implementar el flujo de trabajo “Diseñar componentes” es realizar las actividades de diseño del sistema, a partir del modelo de casos de uso y del modelo de diseño que se haya podido documentar en el flujo de trabajo “Analizar el comportamiento”. Los principales artefactos generados son el modelo y las clases de diseño.

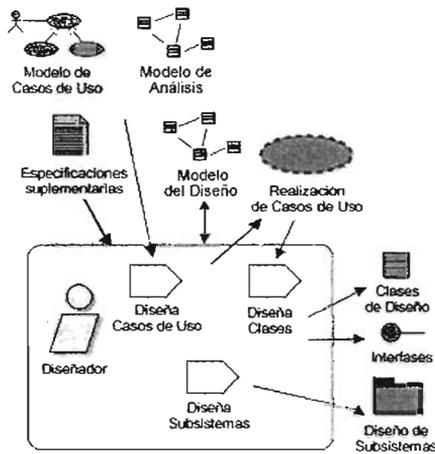


Figura 3.40 Flujo de trabajo “Diseñar componentes”.

El propósito de llevar a cabo el flujo de trabajo “Diseñar la base de datos” es identificar las clases de diseño que deberán ser persistentes en una base de datos, diseñando las estructuras de datos correspondientes.

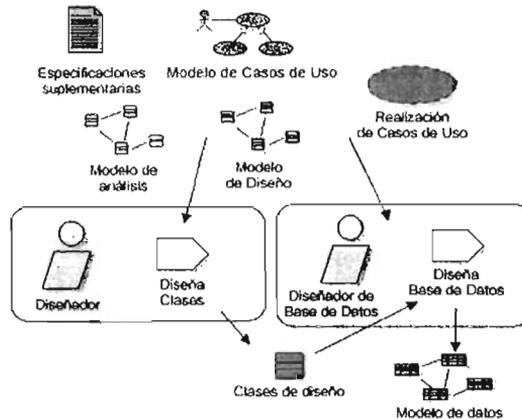


Figura 3.41 Flujo de trabajo “Diseñar la base de datos”.

#### 4. Implementación.

Con el mismo alcance que en la etapa anterior, se seguirán los flujos de trabajo “Implementar componentes” e “Integrar sistema”, con los cuales se llevarán a cabo los trabajos de implementación.

#### 5. Pruebas.

En esta etapa, al igual que en la anterior, se seguirán los flujos de trabajo “Probar y evaluar” y “Evaluar plan de pruebas”. Adicionalmente se pretende implementar el flujo de trabajo “Verificar el método de las pruebas”, a fin de destinar esfuerzos para robustecer el esquema de pruebas.

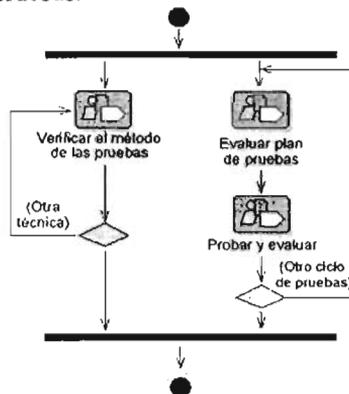


Figura 3.42 Flujo de trabajo de la disciplina de pruebas.

El propósito de realizar el flujo de trabajo “Verificar el método de las pruebas” es verificar que los mecanismos utilizados para llevar a cabo las pruebas nos permitan producir resultados certeros y apropiados de acuerdo con los recursos que se tienen disponibles. Se deberá realizar una revisión de todos los componentes de las pruebas, desde los casos y las guías de pruebas, hasta las configuraciones de los ambientes de pruebas y la automatización de las mismas.

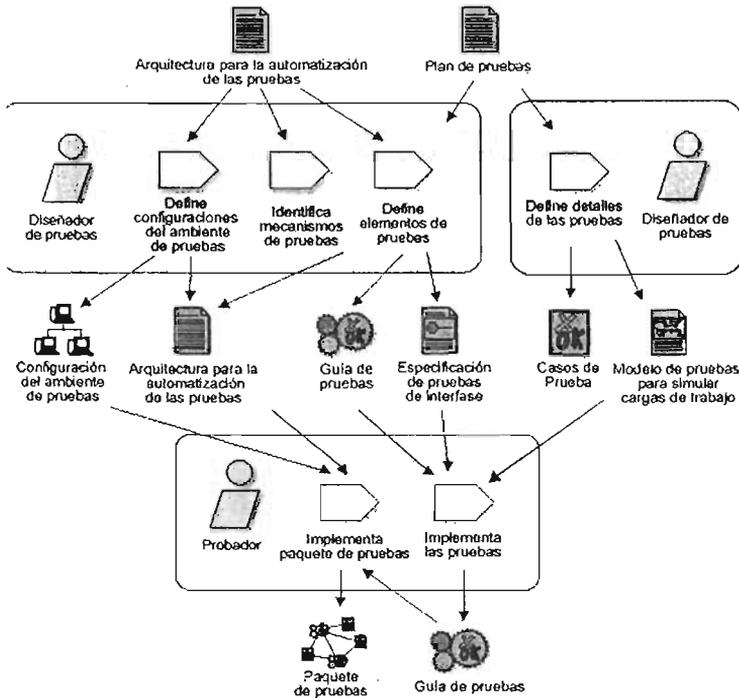


Figura 3.43 Flujo de trabajo “Verificar el método de las pruebas”.

## 6. Despliegue.

En esta etapa se seguirán llevando con el mismo alcance los seis flujos de trabajo que fueron implementados en la etapa de estabilización.

## 7. Administración de la configuración y cambios

Adicionalmente a los flujos de trabajo que se establecieron llevar a cabo en la etapa de estabilización, se propone, para esta etapa en la que comienzan los trabajos de desarrollo de funcionalidades, seguir el flujo de trabajo “Administrar los cambios” con la finalidad de mantener un control formal de los cambios.

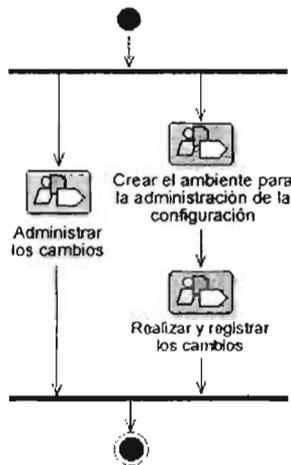


Figura 3.44 Flujo de trabajo de la disciplina de administración de la configuración.

El propósito del flujo de trabajo “Administrar los cambios” es registrar las peticiones de cambio informando de los posibles impactos en el desarrollo del proyecto, así como asegurar que los cambios aprobados son realizados de forma consistente con el resto de los artefactos del proyecto.

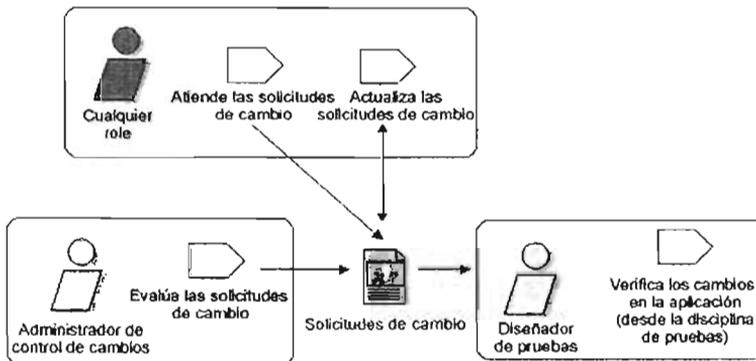


Figura 3.45 Flujo de trabajo “Administrar los cambios”.

## 8. Administración del proyecto.

Adicionalmente a los flujos de trabajo definidos en la etapa de estabilización, se propone implantar el flujo de trabajo “Monitorear y controlar el proyecto”. Esto con la finalidad de generar los artefactos propios del monitoreo y control del proyecto de manera sistemática y formal.

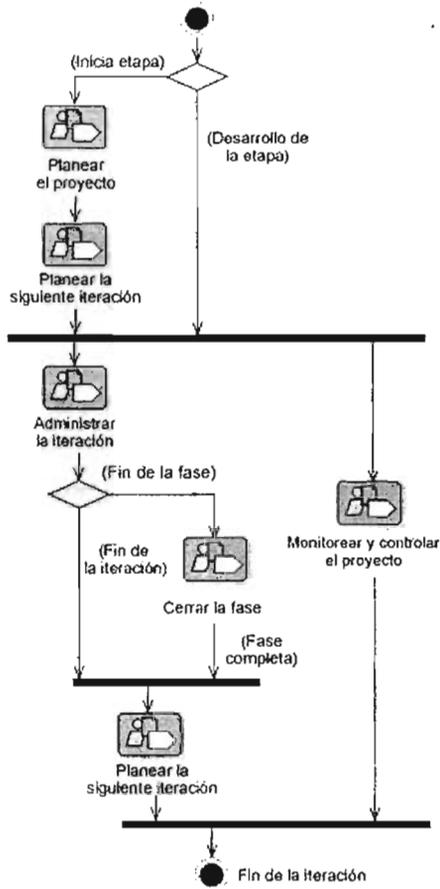


Figura 3.46 Flujo de trabajo de la disciplina de administración del proyecto.

El flujo de trabajo “Monitorear y controlar el proyecto” contiene las actividades que lleva a cabo de forma permanente el administrador del proyecto. Entre las principales se pueden mencionar: el monitoreo del estado del proyecto, la generación de informes de avance y la planificación del trabajo.



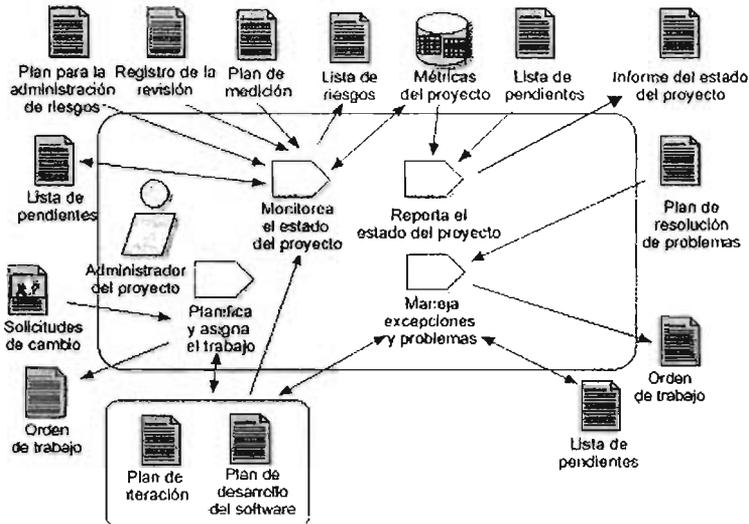


Figura 3.47 Flujo de trabajo “Monitorear y controlar el proyecto”.

## 9. Ambiente.

En esta etapa se seguirán llevando con el mismo alcance los tres flujos de trabajo que fueron implementados en la etapa de estabilización.

### 3.5 Etapa de terminación de la primera fase de desarrollo

El objetivo de esta etapa es liberar dos versiones de SIIRFE, en las cuales se vayan implementando los casos de uso que se encuentran pendientes de construir, y de los cuales se tiene solo la documentación del requerimiento levantado. En esta etapa se conforma el equipo de desarrollo en su totalidad y se lleva a cabo, de forma escalonada y por prioridad, la construcción del resto de las funcionalidades de los subsistemas correspondientes a la primera fase de desarrollo del SIIRFE.

En cuanto al alcance en la parte de implementación, a continuación se enlistan, muy brevemente y por subsistema, las funcionalidades por construir.

#### a) SIIRFE-MAC

- Concluir segunda etapa de depurar movimientos
- Concluir segunda etapa de exportar movimientos
- Concluir segunda etapa de importar movimientos
- Depurar movimientos exportados
- Impresión de varios formatos
- Transmisión de archivos de transacciones de módulos móviles a través de módulos en red

- b) SIIRFE-CECYRD, Subsistema de Actualización al Padrón Electoral
  - Desarrollo de los casos de uso referentes a instancias administrativas
  - Desarrollo de los casos de uso referentes al registro y tratamiento de los formatos de credencial retirables, robados y extraviados
  
- c) SIIRFE-CECYRD, Subsistema de Actualización Cartográfica
  - Aprobación de la conformación de casillas extraordinarias
  - Asignación de casillas extraordinarias a ciudadanos
  - Captura de casillas extraordinarias
  - Consultar casillas extraordinarias
  - Depurar el banco nacional de ciudadanos mal referenciados
  - Generación de los archivos de impresión de notificaciones
  - Incorporación de archivos de casillas extraordinarias
  - Registro de Casillas Extraordinarias
  
- d) SIIRFE-CECYRD, Subsistema de consultas
  - Consulta de formatos robados.
  - Consulta del estatus del trámite vía Internet.
  - Consultar a ciudadano en lista nominal vía Internet.
  - Consultas nominativa por tipo de baja al padrón electoral.
  - Extraer nominativos de movimientos por tipo.
  - Impresión de libros negros.
  - Mantenimiento a distritos locales.
  - Proceso de insaculación.
  - Proceso de listas nominales de exhibición.
  - Proceso de listas nominales definitivas.
  
- e) SIIRFE-Depuración.
  - Desarrollo de los casos de uso referentes a bajas por pérdida de vigencia.
  - Desarrollo de los casos de uso pendientes referentes a depuración preventiva.
  - Desarrollo de los casos de uso referentes a las bajas por defunción y suspensión de derechos.
  - Desarrollo de los casos de uso referentes a la depuración correctiva por detección de registros duplicados.

Al igual que en las etapas anteriores, para los flujos de trabajo de ingeniería definidos en el Proceso Unificado de Rational, se considera un grupo de trabajo por subsistema. Para los flujos de trabajo de soporte, se propone que se consideren como actividades que apoyan a todos los subsistemas, por lo que solo será necesario contar con un equipo de trabajo que atienda esas actividades de forma horizontal. En la figura 3.48 se muestra la estructura de trabajo para esta etapa.

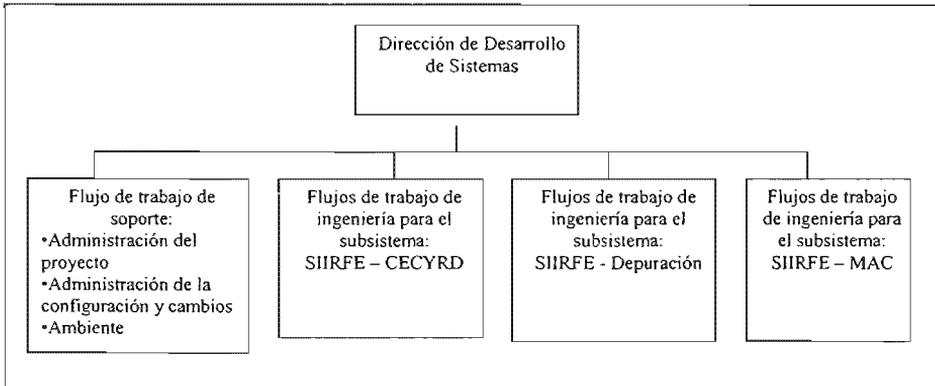


Figura 3.48 Estructura organizacional para atender los flujos de trabajo definidos en RUP para la etapa de “terminación de la primera fase de desarrollo del SIIRFE”

La diferencia con respecto a la etapa anterior, en cuanto a la estructura de trabajo, consiste en que los grupos de trabajo cuentan con un mayor número de gente, y en cuanto a la implementación de RUP, lo que se busca alcanzar es la implementación de un número mayor de flujos de trabajo y actividades.

A continuación se detalla, en cada una de las nueve disciplinas, el alcance en la implementación del proceso de ingeniería basado en RUP para esta etapa.

### 1. Modelado de Negocio.

En la etapa anterior, se adelantaron trabajos en esta disciplina en cuanto a la documentación de los procesos que soportan la operación del Registro Federal de Electores. En esta etapa se debe lograr la identificación de ciertos procesos que no fueron considerados dentro del SIIRFE y que son susceptibles de sistematizar, para la definición y construcción de funcionalidades a desarrollar durante esta etapa y la siguiente.

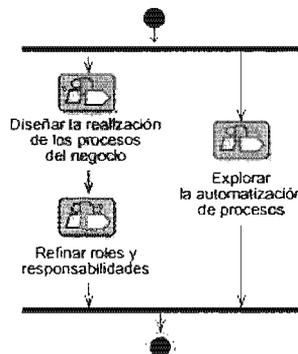


Figura 3.49 Flujo de trabajo de la disciplina de modelado de negocio.

El propósito de realizar el flujo de trabajo “Diseñar la realización de procesos de negocio” será identificar y documentar los roles, productos y eventos en los trabajos del Registro Federal de Electores, para describir la realización de los casos de uso de negocio que son ejecutados por las áreas que lo conforman.

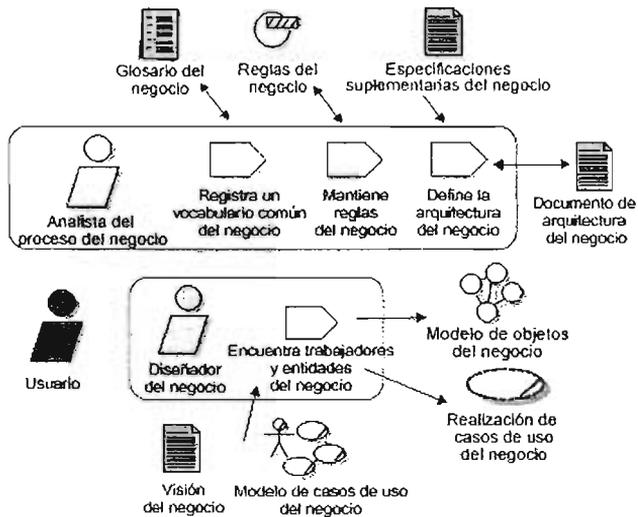


Figura 3.50 Flujo de trabajo “Diseñar la realización de procesos de negocio”.

El objeto de llevar a cabo el flujo de trabajo “Refinar roles y responsabilidades” es enlistar las entidades o áreas que conforman el Registro Federal de Electores, así como describir los acontecimientos que le demandan de ciertos productos y/o trabajos.



Figura 3.51 Flujo de trabajo “Refinar roles y responsabilidades”.

El propósito de realizar el flujo de trabajo “Explorar la automatización de procesos” es revisar las porciones de los procesos del Registro Federal de Electores que pueden y deben ser automatizadas para su óptimo desempeño.

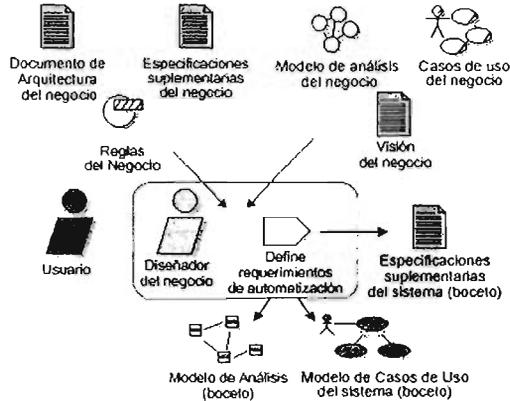


Figura 3.52 Flujo de trabajo “Explorar la automatización de procesos”.

## 2. Requerimientos.

En esta etapa se busca que la administración y el control de los requerimientos se lleven a cabo de forma sistemática a través del flujo de trabajo que se muestra en la figura 3.53.

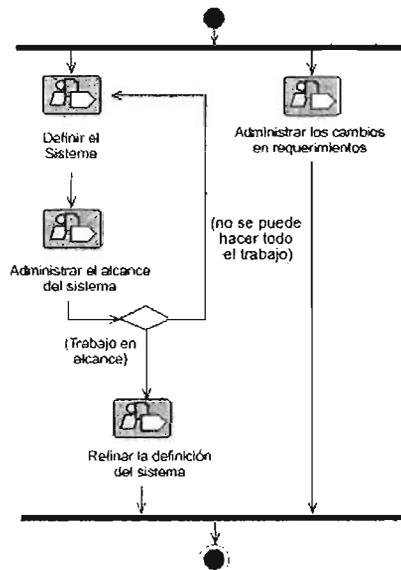


Figura 3.53 Flujo de trabajo de la disciplina de requerimientos.

El objetivo de implementar el flujo de trabajo “Administración de cambios en requerimientos” es el de poder evaluar el impacto de peticiones de cambio a los requerimientos. En la etapa anterior ya se habían establecido ciertas definiciones, que en esta etapa deberán de seguir vigentes. Las actividades a realizar se refieren, una a mantener actualizados los casos de uso, y otra a la evaluación de los impactos en cascada producto de las peticiones de cambio.

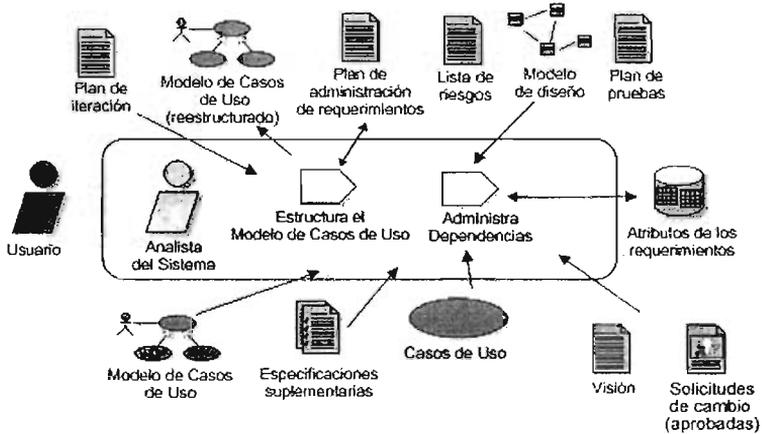


Figura 3.54 Flujo de trabajo “Administración de cambios en requerimientos”.

El propósito de llevar a cabo el flujo de trabajo “Definir el sistema” es obtener la descripción en lo general de los requerimientos. Como principales productos de este flujo de trabajo se encuentran los casos de uso y el modelo de casos de uso.

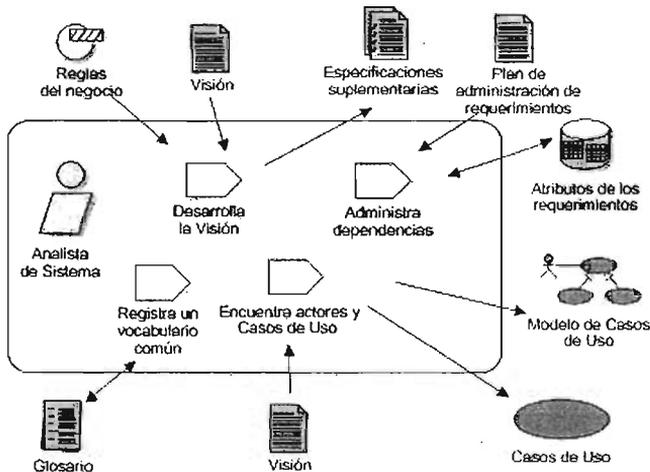


Figura 3.55 Flujo de trabajo “Definir el sistema”.

El propósito de ejecutar el flujo de trabajo “Administrar el alcance del sistema” es destinar esfuerzos para definir el alcance del sistema de la manera mas clara posible, revisando los recursos necesarios para lograr los objetivos de la iteración presente. Las actividades esenciales de este flujo de trabajo son las que se refieren a la definición de prioridades de los casos de uso y a la administración de dependencias entre requerimientos.

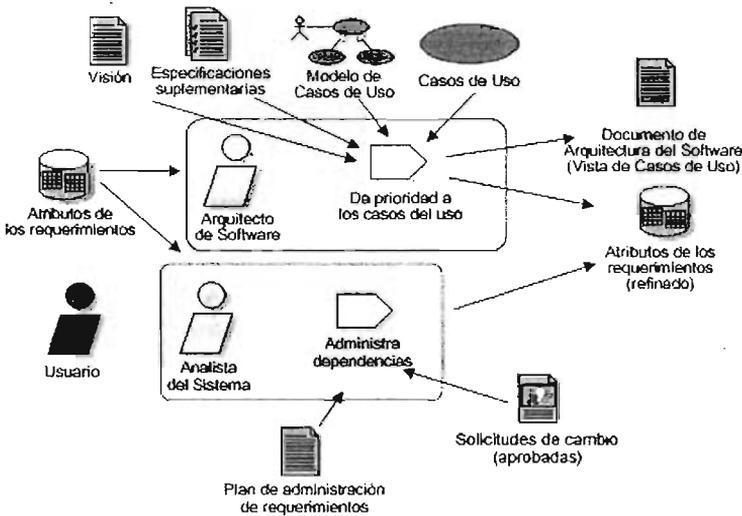


Figura 3.56 Flujo de trabajo “Administrar el alcance del sistema”.

El objetivo de implementar el flujo de trabajo “Refinar la definición del sistema” es obtener, tanto una descripción a detalle de los casos de uso que serán abordados en la presente iteración, como el detalle de las especificaciones de requerimientos suplementarios.

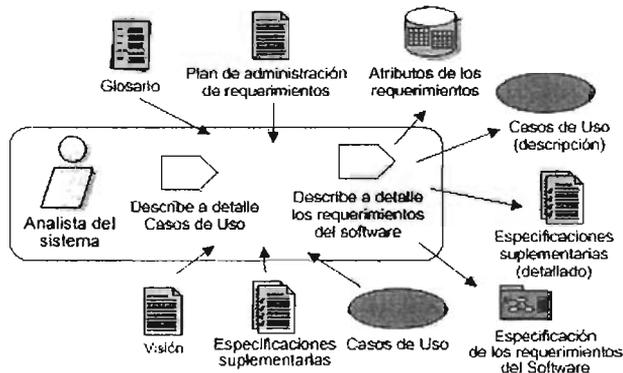


Figura 3.57 Flujo de trabajo “Refinar la definición del sistema”.

### 3. Análisis y Diseño.

Al igual que en la etapa anterior, se deberán ejecutar los flujos de trabajo “Analizar el comportamiento”, “Diseño de componentes” y “Diseño de la base de datos”, mediante los cuales se generarán los principales modelos del sistema, como los son el modelo de diseño, el modelo de datos y el modelo de interfase de usuario o mapa de navegación. Adicionalmente se propone llevar a cabo el flujo de trabajo “Refinar la arquitectura”, ya que es en esta etapa cuando se inician los desarrollos de los casos de uso no construidos, y que podrían representar algún impacto en la arquitectura.

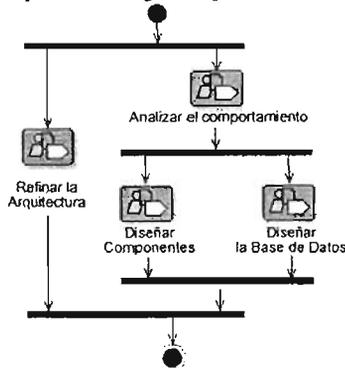


Figura 3.58 Flujo de trabajo de la disciplina de análisis y diseño.

El propósito de llevar a cabo flujo de trabajo “Refinar la arquitectura” es definir las cuestiones de arquitectura implicadas en el desarrollo de los casos de uso de la iteración presente. Se trabaja sobre los modelos de análisis, de diseño, de implementación y de despliegue.

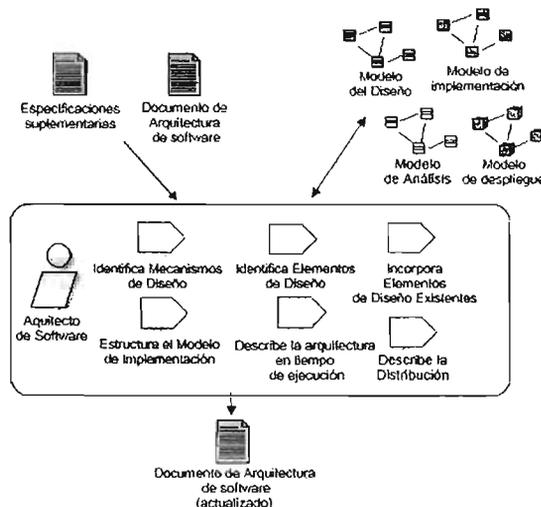


Figura 3.59 Flujo de trabajo “Refinar la arquitectura”.



#### 4. Implementación.

Además de los flujos de trabajo implantados en las etapas anteriores, se deberá llevar a cabo el flujo de trabajo “Estructurar el modelo de implementación”, con el fin de integrar de forma adecuada al SIIRFE los nuevos casos de uso a desarrollar. En esta etapa, se prevé la posibilidad de que se definan subsistemas adicionales a los ya implementados en el SIIRFE.

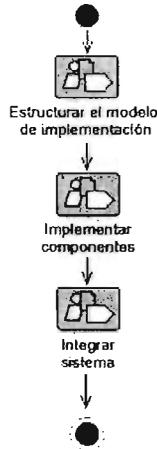


Figura 3.60 Flujo de trabajo de la disciplina de implementación.

El propósito del flujo de trabajo “Estructurar el modelo de implementación” es definir un modelo de implementación, que generalmente resulta en la definición de varios módulos o subsistemas, que pueden ser desarrollados de forma independiente. Un modelo bien organizado ayudará a prevenir problemas con la configuración, permitiendo integrar cada vez más módulos del sistema sin hacer el proceso de integración más complicado.

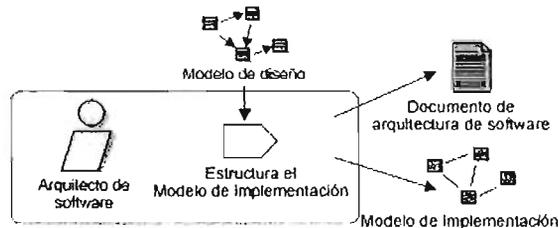


Figura 3.61 Flujo de trabajo “Estructurar el modelo de implementación”.

#### 5. Pruebas.

En esta etapa se seguirán madurando los flujos de trabajo que se han ido implementando en las etapas anteriores: “Probar y evaluar”, “Evaluar plan de pruebas”

y “Verificar el método de las pruebas”. Adicionalmente se pretende llevar a cabo el flujo de trabajo “Definir la misión de las pruebas”, con la finalidad de comenzar a documentar los objetivos de las pruebas, toda vez que es en esta etapa cuando se desarrollan funcionalidades nuevas.

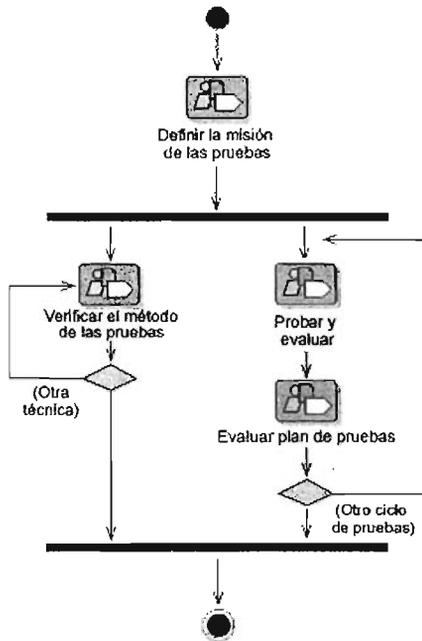


Figura 3.62 Flujo de trabajo de la disciplina de pruebas.

El propósito del flujo de trabajo “Definir la misión de las pruebas” es identificar el enfoque apropiado de los esfuerzos de pruebas para la iteración actual. Aspectos como los objetivos de las pruebas, la definición de la estrategia de pruebas, y la definición de las herramientas para la automatización y control de las pruebas, quedarán documentados en el plan de pruebas.

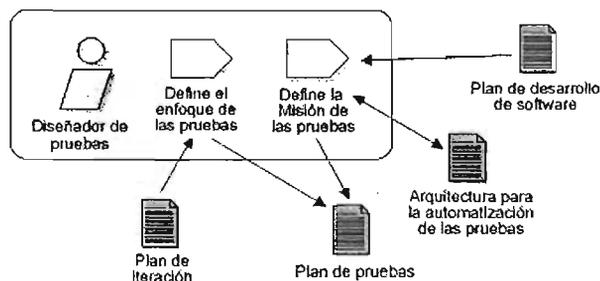


Figura 3.63 Flujo de trabajo “Definir la misión de las pruebas”.

6. Despliegue.

En esta etapa se seguirán madurando los seis flujos de trabajo que se definieron desde la etapa de estabilización y que se siguieron también en la etapa de consolidación.

7. Administración de la configuración y cambios

Adicionalmente a los flujos de trabajo establecidos en las etapas de estabilización y consolidación, se propone llevar a cabo el flujo de trabajo “Planear la administración de la configuración y cambios”. Esto con la finalidad de establecer en lo formal, los mecanismos que se deberán seguir para mantener un adecuado control en la generación de los diversos artefactos del proyecto.

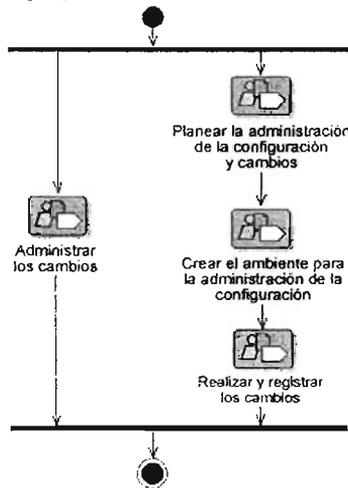


Figura 3.64 Flujo de trabajo de la disciplina de administración de la configuración.

El objetivo del flujo de trabajo “Planear la administración de la configuración y cambios” es establecer un plan que permita una adecuada administración y control de los cambios a los artefactos que son desarrollados como producto del proceso de desarrollo de software.

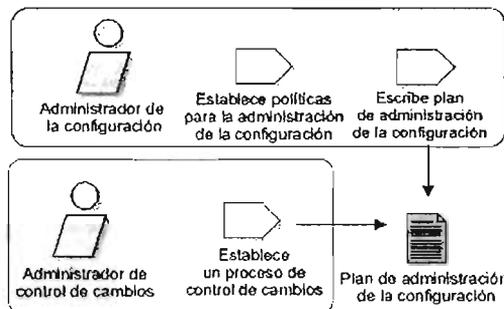


Figura 3.65 Flujo de trabajo “Planear la admón. de la configuración y cambios”.

### 8. Administración del proyecto.

Adicionalmente a los flujos de trabajo establecidos en las etapas de estabilización y consolidación, se incorporan dos flujos de trabajo: “Evaluar el alcance y los riesgos del proyecto” y “Cerrar el proyecto”. El primero de ellos obedece principalmente a que en esta etapa se visualiza el mayor esfuerzo de desarrollo, con la necesidad inmersa de poder delimitar el alcance en el desarrollo de nuevas funcionalidades, así como la evaluación y mitigación de nuevos riesgos. El segundo flujo de trabajo se vuelve necesario, toda vez que el alcance definido hasta el momento, para el proyecto de continuidad al desarrollo del SIIRFE, concluye con esta tercera etapa.

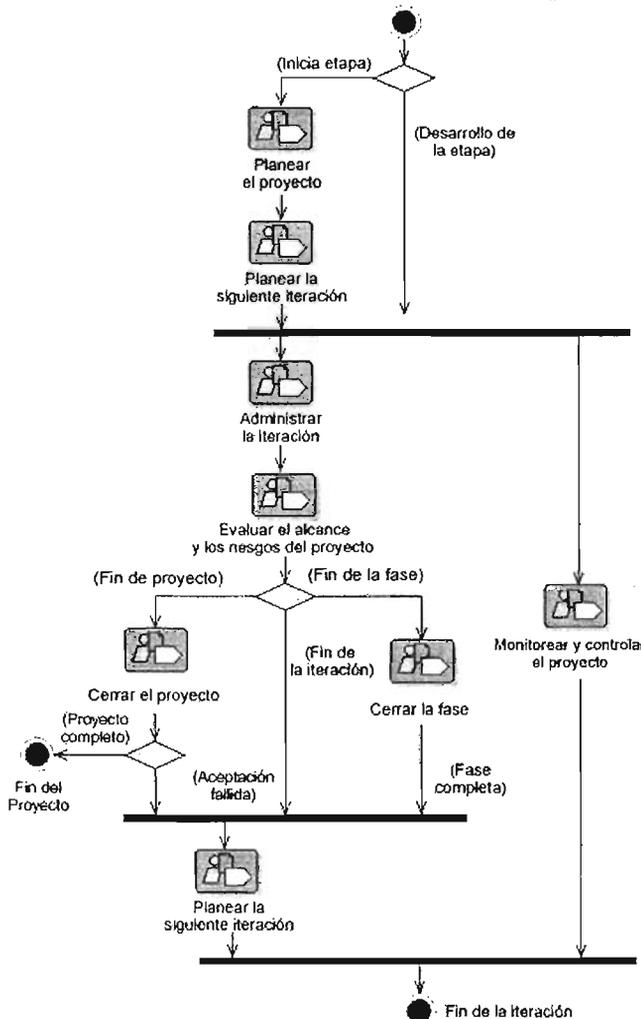


Figura 3.66 Flujo de trabajo de la disciplina de administración del proyecto.

El propósito del flujo de trabajo “Evaluar el alcance y los riesgos del proyecto” es el de identificar, evaluar y detonar las acciones encaminadas a mitigar los riesgos. Como parte de los riesgos, será necesario delimitar el alcance de los trabajos de desarrollo con la finalidad de alcanzar metas reales.



Figura 3.67 Flujo de trabajo “Evaluar el alcance y los riesgos del proyecto”.

En el flujo de trabajo “Cerrar el proyecto”, el administrador del proyecto realiza las actividades encaminadas a la conclusión formal del proyecto.

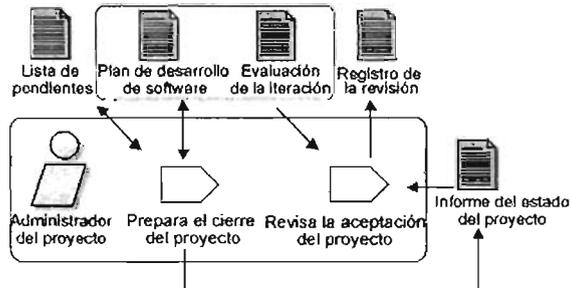


Figura 3.68 Flujo de trabajo “Cerrar el proyecto”.

## 9. Ambiente.

En esta etapa se seguirán madurando los tres flujos de trabajo que se definieron desde la etapa de estabilización y que se siguieron también en la etapa de consolidación.

### 3.6 Justificación de la estrategia

Para dar continuidad al desarrollo de los subsistemas restantes del SIIRFE, se prevé la dificultad que afrontará el equipo de desarrollo para coordinar los múltiples flujos de trabajo que implica ese proyecto, por lo que se requiere de la definición de un proceso que integre las múltiples facetas del desarrollo. Con la implementación de un proceso de

ingeniería de software, basada en la estrategia descrita en este capítulo, se busca lograr lo siguiente:

- Tener una guía ordenada de las actividades del equipo de trabajo
- Una guía de las tareas de cada desarrollador por separado y del equipo como un todo
- Una lista de los artefactos o entregables que deben desarrollarse
- Contar con criterios de control y de medición de los productos y actividades del proyecto
- Contar con un lenguaje común
- Compartir una misma visión
- Un método de trabajo basado en un conjunto de procesos y prácticas probadas y seleccionadas en una amplia comunidad del software
- Contar con un marco de referencia para dar prioridad a las acciones

Una de las mejores propuestas de solución es la de implantar un proceso de ingeniería de software basado en el Proceso Unificado de Rational, el cual es un proceso que no solo cubre la parte conceptual y teórica, sino que además se apoya en una serie de herramientas que en forma conjunta ofrecen un ambiente homogéneo que ayuda a aumentar la productividad del equipo de desarrollo.

La adopción y utilización del Proceso Unificado de Rational ofrece un proceso:

- Estándar
- Reconocido en la industria
- Práctico
- Adaptable a cada tipo de proyecto y equipo de trabajo
- Soportado por herramientas

Entre las principales ventajas de usar RUP en el proyecto SIIRFE, se encuentra que se puede adaptar el proceso para cada una de las etapas definidas en la estrategia de desarrollo, que se cuenta con las herramientas de software que soportan el proceso, que se cuenta con un cierto grado de capacitación inicial en el personal de la Dirección de Desarrollo de Sistemas, y que los trabajos de desarrollo de SIIRFE, en su primera versión, fueron llevados a cabo siguiendo dicha metodología.

Un proceso de ingeniería de software basado en RUP deberá estar sustentado en un esquema de desarrollo iterativo que permita atacar los riesgos más críticos desde el principio, minimizándolos en las primeras iteraciones del ciclo de desarrollo. Se deberá basar en iteraciones y liberaciones parciales a través de nuevas versiones del sistema, en la pronta retroalimentación de los usuarios, en la temprana y continua verificación de la calidad y en la adaptabilidad a recibir los frecuentes cambios de la organización, permitiendo hacer liberaciones más predecibles en tiempo y presupuesto, pero sobre todo, evitando y reduciendo sorpresas en la entrega final al usuario, es decir, garantizando la calidad del producto. Además se prevé que el seguimiento del proceso permitirá y fomentará la comunicación entre los miembros del equipo, eliminando la ambigüedad en la información que cada desarrollador o participante genere.

Cuando se haya logrado el nivel de madurez suficiente, el equipo de desarrollo en general contará con elementos muy poderosos, como lo son el proceso en sí mismo, las técnicas, la experiencia y las herramientas de software, para poder soportar y planear, eficaz y eficientemente, las actividades de mantenimiento del SIIRFE:

- Eficientemente, por que la implantación de esta metodología de desarrollo de sistemas y el aprovechamiento de las herramientas, elevarán el nivel de desempeño del equipo de trabajo, aportando un valor agregado en la búsqueda de una mejor calidad, estimación de costos, productividad y determinación de tiempos de entregas, permitiendo además administrar de una manera más eficiente los riesgos inherentes al proyecto.
- Eficazmente, por que al llevar a cabo las mejores prácticas en cuanto a desarrollo de software que propone el RUP, se podrán cumplir con los compromisos de entrega, se podrá incrementar la calidad de los diferentes productos elaborados en torno al proyecto, y se podrá encaminar el esfuerzo de todo el equipo de trabajo de forma correcta.

# Impactos en la Organización

En este capítulo se describe la manera en que el área de desarrollo evoluciona en forma paralela con la implementación del proceso de ingeniería. También se presentan ciertas consideraciones relevantes para la instrumentación de un proceso de ingeniería en una organización de desarrollo de sistemas.



## 4.1 Adecuación de funciones en la organización

Ante los cambios que se presentan en la DERFE, tanto en la misma operación del Registro Federal de Electores, como al interior de la Dirección de Desarrollo de Sistemas en cuanto a la implementación de un proceso de ingeniería de software, se observa necesaria una reestructuración de funciones en cada una de las áreas que conforman a la Dirección de Desarrollo de Sistemas. Esa Dirección encontrará su misión en el mantenimiento del SIIRFE, eventualmente los otros sistemas que son operados y mantenidos en el Registro Federal de Electores deberán integrarse a éste.

Así pues, un buen punto de partida para llevar a cabo esa reestructuración, sería la estructura organizacional propuesta para la tercera etapa definida en la estrategia para la continuidad al desarrollo y mantenimiento del SIIRFE, mostrada en la figura 4.1

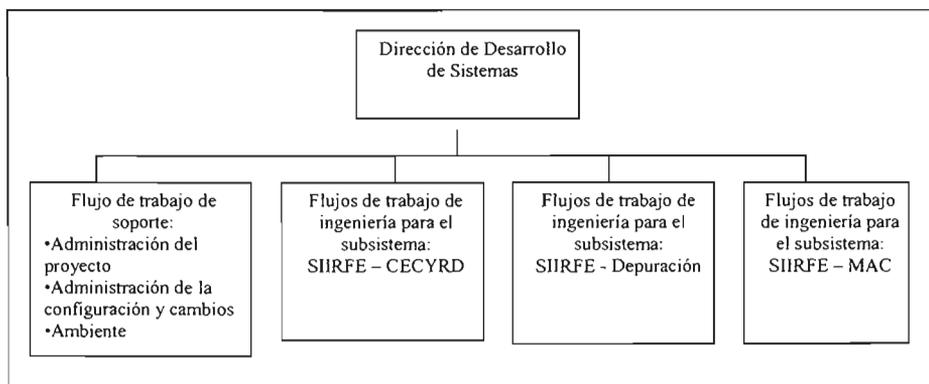


Figura 4.1 Estructura organizacional para atender los flujos de trabajo definidos en RUP para la etapa de “terminación de la primera fase de desarrollo del SIIRFE”

Será importante valorar la división del trabajo precisamente a través de las disciplinas que se definen en RUP, delegando a un área en particular los flujos de trabajo de soporte, y a otras áreas los flujos de trabajo de ingeniería. Una propuesta inicial para una estructura organizacional a mayor detalle, consistiría en subdividir a su vez, por grupos de flujos de trabajo, cada una de las áreas definidas en la figura 4.1, de la siguiente manera:

1. Para los flujos de trabajo de soporte, se podrán crear tres grupos de trabajo atendiendo en cada uno de ellos una disciplina del proceso:
  - Administración del proyecto.
  - Administración de la configuración y cambios.
  - Ambiente.

- Para los flujos de trabajo de ingeniería, se podrán crear dos grupos de trabajo, cada uno de los cuales atendería actividades de ciertas disciplinas conforme a lo siguiente:
  - Modelado de Negocio, Requerimientos, Pruebas y Despliegue.
  - Análisis, Diseño e Implementación.

La manera en la que se propone agrupar las disciplinas correspondientes a los trabajos de ingeniería, obedece a ciertos criterios como lo son: la afinidad en las actividades, un equilibrio en las cargas de trabajo y una cantidad controlable de grupos de trabajo. La estructura organizacional propuesta se puede observar en la figura 4.2

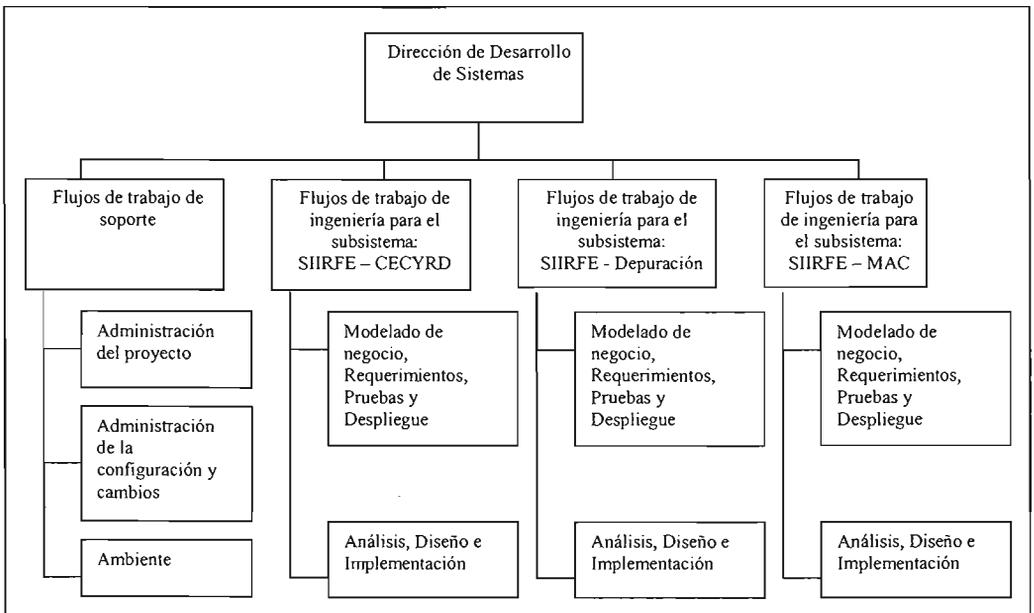


Figura 4.2 Propuesta inicial de una reestructura organizacional basada en las disciplinas de trabajo definidas en RUP.

No obstante, habría que recordar que el SIIRFE lo constituyen otros cuatro subsistemas, tres de los cuales aún no han sido desarrollados, así como poner en consideración la inclusión de otras actividades, como las de soporte a la operación cotidiana.

Considerando estos últimos aspectos, se proponen dos adecuaciones más a la estructura organizacional:

- Que se considere un nuevo grupo en las áreas dedicadas a llevar a cabo los flujos de trabajo de ingeniería, que realice tanto las actividades de despliegue como las de soporte a la operación.

2. Que se redistribuyan los subsistemas aún no desarrollados entre las áreas dedicadas a los flujos de trabajo de ingeniería, considerando que los otros subsistemas estarán sólo en una etapa de mantenimiento y mejora continua.

Las dos adecuaciones descritas tratan de cumplir con ciertos criterios como lo son: la afinidad en las actividades, un equilibrio en las cargas de trabajo y una cantidad controlable de grupos de trabajo. La estructura organizacional propuesta se puede observar en la figura 4.3

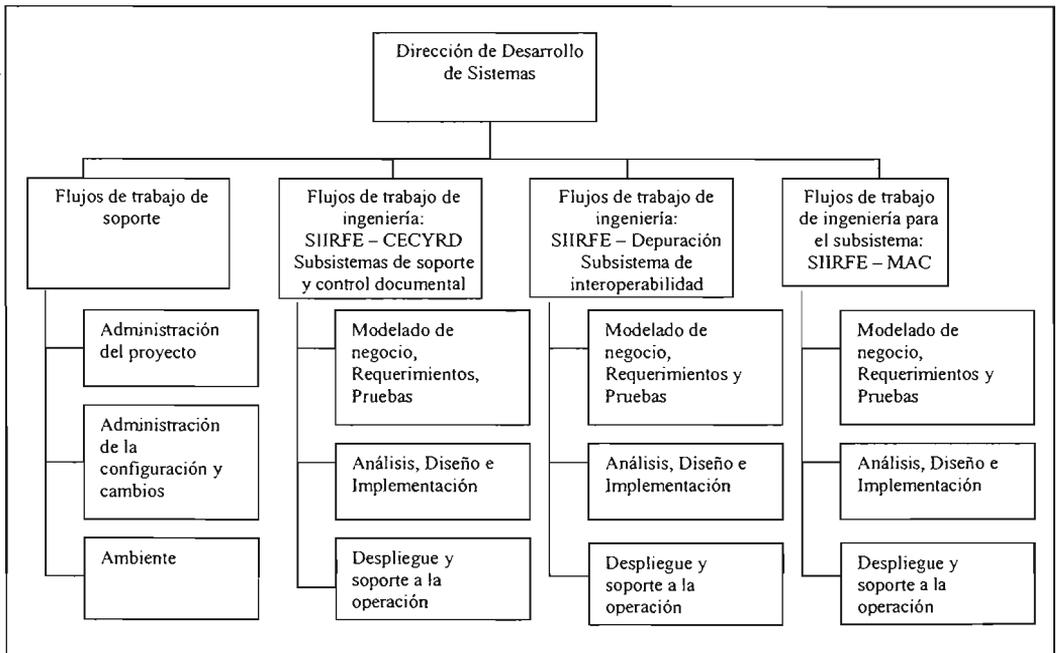


Figura 4.3 Propuesta de una reestructura organizacional considerando las actividades de soporte a la operación y los demás subsistemas del SIIRFE.

Es muy importante mencionar que en las estructuras organizacionales aquí propuestas, se considera que los distintos grupos de trabajo deberán interactuar de forma horizontal, que no podrán trabajar de forma autónoma e independiente en el cumplimiento de todas las actividades de las disciplinas del proceso que le fueron conferidas, es decir, que para el desarrollo de ciertas actividades de determinadas disciplinas de trabajo, será necesaria la colaboración de personal que participa de lleno en otras disciplinas de trabajo.

Los flujos de trabajo que definen actividades a realizar por personal cuyo rol principal se encuentra definido en otros flujos de trabajo son: pruebas, despliegue y ambiente. Entre los

flujos de trabajo con mayor interacción se encuentran: requerimientos, análisis y diseño, implementación, pruebas y despliegue.

## **4.2 Consideraciones en la implementación de un proceso de ingeniería de software**

Cambiar una metodología de trabajo es difícil, y puede llevar un largo tiempo antes de observar los verdaderos efectos. No se trata solo de adoptar nuevas herramientas, lo cual debería ser algo relativamente sencillo de realizar: se podrán instalar, se leerán los manuales de usuario, se verán ejemplos y se podrá capacitar al personal sobre el uso de ellas. La transición para utilizar nuevas herramientas puede tomar de unas cuantas horas a unas cuantas semanas. Pero cambiar el proceso de desarrollo de software a menudo significa que se afectan formas de pensar y valores en los individuos involucrados, así como también se cambia la forma en la que perciben su trabajo. Representa un cambio en la cultura y hasta un cambio en la estructura organizacional.

Un cambio en el proceso de trabajo afecta a los individuos y a la propia organización más profundamente que un cambio en la tecnología o en las herramientas. Deberá ser planeado y administrado cuidadosamente. Se deberá identificar la oportunidad y los beneficios e ir implementando el cambio gradualmente.

Cuando se implementa un proceso, se deberán observar los siguientes aspectos:

- El personal, su perfil, habilidades, motivación y actitud. Se deberá tratar de capacitar y motivar adecuadamente. Para que la implementación del proceso se realice de manera exitosa se deberá involucrar al personal en el esfuerzo que esto conlleva lo más rápido posible. Se debe hacer conciencia en el personal de la situación actual de la organización, dejando ver la forma en la que su trabajo puede mejorar.
- Las herramientas para soportar el nuevo proceso. Será necesario asegurar la correcta utilización de las herramientas que permitan al grupo de desarrollo cumplir con el proceso que se busca implementar.
- El proceso de desarrollo de software, esto es, el modelo del ciclo de vida, la estructura organizacional, las actividades a realizar, las prácticas a seguir, los artefactos a producir, y el alcance del mismo proceso.

La implementación de un nuevo proceso en un área dedicada al desarrollo de sistemas deberá considerar las siguientes actividades:

1. Evaluación del estado actual de la organización en términos de su gente, su proceso y herramientas que lo soportan. También se deberán identificar los problemas y áreas potenciales de mejora. Con ello se pretende primero no introducir el proceso entero y todas las herramientas de una sola vez, en vez de eso, se debe llevar a cabo

la implementación de forma incremental, iniciando con las áreas que tienen mayor necesidad y el mayor potencial para mejorar.

2. Fijar metas para el proceso, la gente y las herramientas, estableciendo a donde se quiere llegar cuando el la implementación del proceso este completa. Esto es importante por que nos brinda elementos para realizar una mejor planeación. Se deberán fijar metas concretas que puedan ser entendidas por los miembros del proyecto. Las metas nos darán una visión de la situación futura de la organización.
3. Se deberá realizar un análisis de riesgos para identificar los riesgos potenciales, tratando de medir el impacto para ver la forma en la que se podrán ir mitigando y en qué orden. Cambiar un proceso lineal a un proceso iterativo no esta libre de riesgos.
4. Se deberá desarrollar un plan para la implementación del proceso y las herramientas que lo soporten. No se deberá tratar de hacer todo a la vez, en vez de eso, la implementación se deberá realizar por etapas, en las que se pretenda implementar una parte a la vez del nuevo proceso. Se deberá poner mayor énfasis en las áreas que se cree tengan una mayor oportunidad de mejora. Evaluando los riesgos y las necesidades de la organización, se podrá, para algunos casos, iniciar rápidamente usando el proceso y las herramientas en un proyecto real, y en otros, se deberá ser más cuidadoso usando proyectos pilotos para verificar el proceso tratando de mitigar algunos riesgos.
5. La implementación propiamente del nuevo proceso de desarrollo deberá realizarse considerando las actividades descritas previamente. Consiste en la definición del nuevo modelo de desarrollo, la adaptación de herramientas para soportar y automatizar el proceso, el entrenamiento del equipo de trabajo y la aplicación finalmente del proceso y las herramientas en un proyecto.
6. Cuando se ha llevado a cabo la implementación del proceso en un proyecto de desarrollo de software, ya sea real o piloto, se deberán evaluar los resultados entorno a las metas esperadas, a la productividad del personal y en cuanto a la implementación del mismo proceso.

La implementación de un proceso de desarrollo de software es una tarea compleja que deberá ser controlada cuidadosamente. Deberá ser dividida dentro de varias etapas y en cada una de ellas será necesario ejecutar las actividades descritas anteriormente hasta que el proceso y las herramientas que lo soportan, han sido instalados y utilizados completamente en la organización.

**Capítulo**

**5**

Conclusiones y  
Trabajos a futuro

La intención final de una organización no es tener un software bueno, sino administrar sus procesos de negocio, de forma que le permita producir rápidamente bienes y servicios de alta calidad con costos razonables. El software es el arma estratégica con el cual las empresas o los gobiernos pueden conseguir enormes reducciones en costos y tiempos de producción tanto para bienes como servicios. En el entorno de una economía global que opera todo el tiempo, muchos de esos procesos no pueden funcionar sin el software. Un buen proceso de desarrollo de software es, por tanto, un elemento crítico para el éxito de cualquier organización.

Para el caso del Registro Federal de Electores, se ha expuesto una propuesta de la forma en la que se puede implementar un proceso de desarrollo de software basado en el Proceso Unificado de Rational, con lo que se busca hacer mucho más eficientes y eficaces los trabajos encomendados al área de sistemas.

Al final del primer capítulo se describió una serie de oportunidades de mejora en dicha área que bien deberán ser atendidas mediante la instrumentación del proceso de ingeniería, ya que en éste, además de que se especifican a detalle todos los flujos de trabajo necesarios para el desarrollo de sistemas, se contemplan las mejores prácticas en la industria del software como son: el desarrollo iterativo e incremental a través de iteraciones y liberaciones parciales, el desarrollo basado en una administración de requerimientos descritos en casos de uso, una verificación continua de la calidad, y una administración de los cambios que permita medir impactos y mantener un control adecuado sobre los artefactos producidos.

Con el fin de mitigar posibles riesgos en la propia instrumentación del proceso, se ha definido una estrategia gradual en la implementación de cada uno de los flujos de trabajo definidos en RUP, en la que se busca atacar primero aquellas áreas que representan una necesidad más fuerte de atención.

La estrategia presentada en este trabajo será de gran importancia para el proyecto de mantenimiento del SIIRFE, ya que en ella se expone, de forma congruente, un esquema incremental tanto para la implementación del proceso de ingeniería, como para los alcances en el desarrollo del sistema.

También se ha presentado, acorde con la estrategia de tipo incremental, una manera de reestructurar las funciones en el área de sistemas en la que se contempla, por un lado, cada uno de los flujos de trabajo definidos en RUP, y por otro lado, los diferentes subsistemas que integran al SIIRFE.

Aún cuando en las etapas iniciales del proyecto, el tiempo y los recursos destinados a la implementación del proceso de ingeniería de software, no permitan observar una ganancia significativa en los costos asociados al mantenimiento de dicho sistema, la relación costo beneficio deberá retribuir en el mediano y largo plazo, toda vez que un proceso de ingeniería de software garantiza el control eficaz en el desarrollo de sistemas de alta complejidad, como lo es el SIIRFE.

Los trabajos a futuro serán aquellos que se refieran a la instrumentación de la propia estrategia, observando los avances y los resultados en el desarrollo y calidad de las futuras versiones del SIIRFE. Será de gran valor mantener las métricas adecuadas para conocer los indicadores de la productividad del área de desarrollo a través del tiempo.

La aportación tangible de la estrategia aquí presentada se llevará a cabo en el momento de iniciar con la instrumentación del proceso de ingeniería en los trabajos de desarrollo del SIIRFE y, sin embargo, dicha estrategia no deberá ser considerada de forma rígida, sino que se podrá ir adaptando tanto en los alcances del proceso como en los trabajos de desarrollo, en función de los cambios y necesidades que se presenten en la organización.



**Apéndice**

**A**

# Roles definidos en RUP

En este apéndice se resumen los roles involucrados en el Proceso Unificado de Rational. Se debe mencionar también que un role no es un individuo, sino que representa un trabajo que ciertos individuos o equipos realizan para producir determinados artefactos.

**Administrador de control de cambios.** El administrador de control de cambios dirige el proceso de control de cambios. Este role generalmente es llevado a cabo mediante un equipo de trabajo de control de cambios, el cual debería integrarse por representantes de todas las partes interesadas, incluyendo clientes, desarrolladores y usuarios. En un pequeño proyecto, una simple persona, tal como el administrador del proyecto o arquitecto de software, pueden jugar este role.

**Administrador de despliegue.** El administrador de despliegue es responsable de planear la transición del producto a la comunidad de usuarios. Estas tareas son documentadas en planes de despliegue.

**Administrador de la configuración.** Un administrador de la configuración es responsable de proveer totalmente de la infraestructura y del ambiente al equipo de desarrollo. Su función es soportar las actividades de desarrollo para que los programadores e integradores cuenten con los espacios de trabajo necesarios para construir y probar sus aplicaciones y que todos los artefactos estén disponibles conforme se requieran. El administrador de la configuración se deberá asegurar de llevar un adecuado control de versiones que facilite la revisión del producto, el registro y ejecución de los cambios. También es responsable de generar el plan de la administración de la configuración y de informar de los impactos en tiempo y costo de los cambios solicitados.

**Administrador del proyecto.** El administrador del proyecto designa recursos, indica prioridades, coordina la relación de usuarios y clientes, y generalmente trata de mantener al equipo de trabajo enfocado hacia la meta correcta. El administrador del proyecto establece un conjunto de prácticas para asegurar la integridad y la calidad de los artefactos del proyecto.

**Administrador del sistema.** El administrador del sistema mantiene el ambiente de desarrollo, tanto hardware como software, y es responsable de la administración del sistema, respaldos, etc.

**Analista del proceso de negocio.** El analista del proceso de negocio dirige y coordina el modelado de casos de uso de negocio, describiendo la organización que esta siendo modelada.

**Analista del sistema.** El analista del sistema dirige y coordina el levantamiento de requerimientos y modelado de casos de uso, delimitando la funcionalidad y el alcance del sistema.

**Arquitecto.** El arquitecto dirige y coordina las actividades y artefactos de índole técnico a través del proyecto. El arquitecto establece la estructura en general para cada vista

de la arquitectura: la descomposición de la vista, el agrupamiento de elementos, y sus interfaces.

**Desarrollador de cursos.** El desarrollador de cursos genera el material de capacitación para enseñar a los usuarios a usar el producto de software. Crea presentaciones, notas de estudiante, ejemplos y todo lo demás necesario para el entendimiento del producto.

**Diseñador.** El diseñador define las responsabilidades, operaciones, atributos y relaciones de algunas clases y determina como ellas deberán ser construidas en el ambiente de implementación. Adicionalmente, el diseñador puede definir el diseño de paquetes o de subsistemas y su interrelación.

**Diseñador de bases de datos.** El diseñador de bases de datos define las tablas, índices, vistas, reglas, desencadenantes, procedimientos almacenados, parámetros de almacenamiento, y otras especificaciones de bases de datos necesarias para almacenar, recuperar y eliminar objetos persistentes.

**Diseñador de capsulas.** Este role se enfoca en asegurar que el sistema es capaz de responder a eventos de forma oportuna; de acuerdo con los requerimientos de concurrencia que se hayan definido. Una capsula es un patrón de diseño específico el cual representa un hilo de control encapsulado en el sistema.

**Diseñador de interfaz de usuario.** El diseñador de interfase de usuario dirige y coordina el prototipo y diseño de la interfase de usuario capturando los requerimientos en cuanto al uso de controles gráficos. Construye prototipos de interfase de usuario, los cuales los somete a diversas pruebas con los usuarios finales, con el fin de obtener una apropiada retroalimentación para la obtención final de la interfase de usuario.

**Diseñador de negocio.** El diseñador de negocio detalla la especificación de una parte de la organización al describir el flujo de trabajo de uno o algunos casos de uso del negocio. Especifica los roles y las entidades del negocio necesarias para realizar un caso de uso de negocio. El diseñador de negocio define las responsabilidades, operaciones, atributos, y relaciones de uno o algunos roles de negocio y entidades de negocio.

**Diseñador de pruebas.** El diseñador de pruebas es responsable de la planeación, implementación y evaluación de las pruebas, incluyendo la generación del plan y el modelo de pruebas, la implementación de los procedimientos de pruebas y la evaluación del alcance, resultados y efectividad de las mismas.

**Escritor de casos de uso.** Este role deberá detallar la especificación de una parte de la funcionalidad del sistema, describiendo los requerimientos de uno o varios casos de uso. Puede también ser responsable por un paquete de casos de uso y por mantener la integridad de ese paquete.

**Escritor técnico.** El escritor técnico produce material de soporte para el usuario final, tal como guías de usuario, textos de ayuda, notas de versión, etc.

**Especialista en herramientas.** El especialista en herramientas es responsable de soportar las herramientas usadas en el proyecto. Esto incluye la evaluación de la necesidad, selección y adquisición de herramientas.

**Ingeniero del proceso.** El ingeniero del proceso es el responsable del proceso de desarrollo de software en sí mismo. Esto incluye tanto la configuración del proceso antes del arranque del proyecto, como la mejora continua del mismo durante el esfuerzo de desarrollo.

**Integrador del sistema.** Los programadores colocan componentes probados dentro de un área designada para tal fin, en donde los integradores del sistema los integran para producir una versión interna. Un integrador del sistema es también responsable de planear la integración del sistema.

**Involucrado.** Un involucrado es cualquier persona que es materialmente afectada por el resultado del proyecto.

**Probador.** El probador es responsable de ejecutar las pruebas; deberá evaluar los resultados de las mismas, así como también registrar en una bitácora las incidencias detectadas y la manera de reproducirlas.

**Programador.** Este role lleva a cabo la construcción y las pruebas de componentes, de acuerdo con los estándares adoptados en el proyecto, de tal forma que dichos componentes puedan ser integrados en subsistemas mas complejos.

**Revisor de la arquitectura.** El revisor de la arquitectura planea y conduce las revisiones formales de la arquitectura en general.

**Revisor de código.** Un revisor de código es responsable de asegurar la calidad del código fuente, así como de la planeación y ejecución de dichas revisiones.

**Revisor de requerimientos.** El revisor de requerimientos planea y conduce las revisiones formales del modelo de casos de uso.

**Revisor del diseño.** El revisor del diseño planea y conduce las revisiones formales del modelo de diseño.

**Revisor del modelo de negocio.** Este role participa en las revisiones formales del modelo de casos de uso de negocio y el modelo de objetos de negocio.

**Revisor del proyecto.** El revisor del proyecto es responsable de la evaluación de los artefactos de planeación y evaluación del proyecto, en los principales puntos de revisión del ciclo de vida del proyecto.

**Apéndice**

**B**

Artefactos  
definidos en RUP

En este apéndice se resumen los principales artefactos que son producidos o usados durante el proceso. Éstos son organizados en este apéndice de acuerdo con las nueve disciplinas de RUP y son relacionados con el role responsable de producirlos.

Artefactos de la disciplina del *modelado de negocio*.

Artefacto	Role responsable
Evaluación de la organización	Analista del proceso de negocio
Visión del negocio	Analista del proceso de negocio
Glosario de negocio	Analista del proceso de negocio
Reglas de negocio	Analista del proceso de negocio
Especificación suplementaria del negocio	Analista del proceso de negocio
Modelo de casos de uso de negocio <ul style="list-style-type: none"> <li>• Casos de uso de negocio</li> <li>• Actores de negocio</li> </ul>	Analista del proceso de negocio Diseñador de negocio Diseñador de negocio
Realización de casos de uso de negocio	Diseñador de negocio
Modelo de objetos de negocio <ul style="list-style-type: none"> <li>• Unidad de organización</li> <li>• Roles de negocio</li> <li>• Entidades de negocio</li> </ul>	Analista del proceso de negocio Diseñador de negocio Diseñador de negocio Diseñador de negocio
Documento de arquitectura de negocio	Analista del proceso de negocio

Artefactos de la disciplina de *requerimientos*.

Artefacto	Role responsable
Plan de administración de requerimientos	Analista del sistema
Peticiones de involucrados	Analista del sistema
Glosario	Analista del sistema
Visión	Analista del sistema
Atributos de requerimientos	Analista del sistema
Especificación suplementaria	Analista del sistema
Especificación de requerimientos de software	Escritor de casos de uso
Modelo de casos de uso <ul style="list-style-type: none"> <li>• Paquete de casos de uso</li> <li>• Casos de uso</li> <li>• Sistemas actores</li> <li>• Humanos actores</li> </ul>	Analista del sistema Escritor de casos de uso Escritor de casos de uso Analista del sistema Diseñador de interfase de usuarios
Limites de clases	Diseñador de interfase de usuarios
Relatos de casos de uso	Diseñador de interfase de usuarios
Prototipo de interfase de usuarios	Diseñador de interfase de usuarios

Artefactos de la disciplina de *análisis y diseño*.

<b>Artefacto</b>	<b>Role responsable</b>
Documento de arquitectura de software	Arquitecto
Realización de casos de uso	Diseñador
Modelo de análisis <ul style="list-style-type: none"><li>• Clases de análisis</li></ul>	Arquitecto Diseñador
Modelo de diseño <ul style="list-style-type: none"><li>• Diseño de subsistemas</li><li>• Diseño de paquetes</li><li>• Diseño de clases</li><li>• Interfase</li></ul>	Arquitecto Diseñador Diseñador Diseñador Diseñador
Modelo de datos	Diseñador de bases de datos

Artefactos de la disciplina de *implementación*.

<b>Artefacto</b>	<b>Role responsable</b>
Modelo de implementación <ul style="list-style-type: none"><li>• Implementación de subsistemas</li><li>• Componentes</li></ul>	Arquitecto Programador Programador
Plan de integración de la aplicación	Integrador del sistema

Artefactos de la disciplina de *pruebas*.

<b>Artefacto</b>	<b>Role responsable</b>
Plan de pruebas	Diseñador de pruebas
Modelo de pruebas <ul style="list-style-type: none"><li>• Procedimientos de pruebas</li><li>• Casos de pruebas</li></ul>	Diseñador de pruebas Diseñador de pruebas Diseñador de pruebas
Programas de pruebas	Diseñador de pruebas
Pruebas de cargas de trabajo	Diseñador de pruebas
Pruebas de paquetes <ul style="list-style-type: none"><li>• Clases de pruebas</li></ul>	Diseñador Diseñador
Pruebas de subsistemas <ul style="list-style-type: none"><li>• Pruebas de componentes</li></ul>	Programador Programador
Resultados de las pruebas	Probador
Resumen de la evaluación de pruebas	Diseñador de pruebas

Artefactos de la disciplina de *despliegue*.

Artefacto	Role responsable
Plan de despliegue	Administrador del despliegue
Listado de materiales	Administrador del despliegue
Notas de versión	Administrador del despliegue
Instalación de componentes	Programador
Material de soporte	Escritor técnico
Material de capacitación	Desarrollador de cursos
Presentación del producto	Diseñador gráfico

Artefactos de la disciplina de *administración de la configuración y cambios*.

Artefacto	Role responsable
Plan de administración de la configuración	Administrador de la configuración
Repositorio del proyecto	Administrador de la configuración
Solicitudes de cambio	Administrador de control de cambios

Artefactos de la disciplina de *administración del proyecto*.

Artefacto	Role responsable
Caso de negocio	Administrador del proyecto
Plan de desarrollo de software <ul style="list-style-type: none"> <li>• Plan de iteraciones</li> <li>• Plan de resolución de problemas</li> <li>• Plan de administración de riesgos</li> <li>• Plan de aceptación del producto</li> <li>• Plan de medición</li> </ul>	Administrador del proyecto Administrador del proyecto Administrador del proyecto Administrador del proyecto Administrador del proyecto Administrador del proyecto
Evaluación de las iteraciones	Administrador del proyecto
Evaluación de estatus	Administrador del proyecto
Listado de riesgos	Administrador del proyecto
Métricas del proyecto	Administrador del proyecto
Registro de revisiones	Administrador del proyecto



Artefactos de la disciplina de *ambiente*.

<b>Artefacto</b>	<b>Role responsable</b>
Plan de aseguramiento de la calidad	Ingeniero del proceso
Evaluación de la organización de desarrollo	Ingeniero del proceso
Plantillas requeridas en el proyecto	Ingeniero del proceso
Caso de desarrollo <ul style="list-style-type: none"><li>• Lineamientos de diseño</li><li>• Lineamientos de modelado de casos de uso</li><li>• Lineamientos de negocio</li><li>• Lineamientos de programación</li><li>• Lineamientos de pruebas</li></ul>	Ingeniero del proceso Arquitecto Analista del sistema Analista del proceso de negocio Arquitecto Diseñador de pruebas
Soporte del ambiente	Administrador del sistema
Evaluación de las herramientas de soporte	Especialista en herramientas
Herramientas	Especialista en herramientas

Descripción de  
una arquitectura  
basada en J2EE

La definición de una arquitectura de software involucra un conjunto de decisiones claves de diseño, reglas y patrones que definen un espacio de trabajo en el cual el diseño y la implementación del sistema toman lugar. Las decisiones a nivel de la arquitectura serán fundamentales en la construcción del sistema, y cambiarlas más adelante durante el desarrollo, puede repercutir significativamente sobre el sistema mismo y sobre la planeación y costos del proyecto.

En la elección de una arquitectura de software se definen elementos de la organización del sistema de software, tales como la estructura e interrelación de componentes, condiciones de funcionalidad, de uso, de confiabilidad, de rendimiento, de reutilización, así como algunas otras de índole económico y tecnológico.

Resulta imposible tratar de describir la arquitectura de un sistema desde una sola perspectiva. Para describir un sistema de software se deberán utilizar modelos. La figura C.1 ilustra los principales modelos utilizados en una arquitectura basada en J2EE y la relación que existe entre ellos.

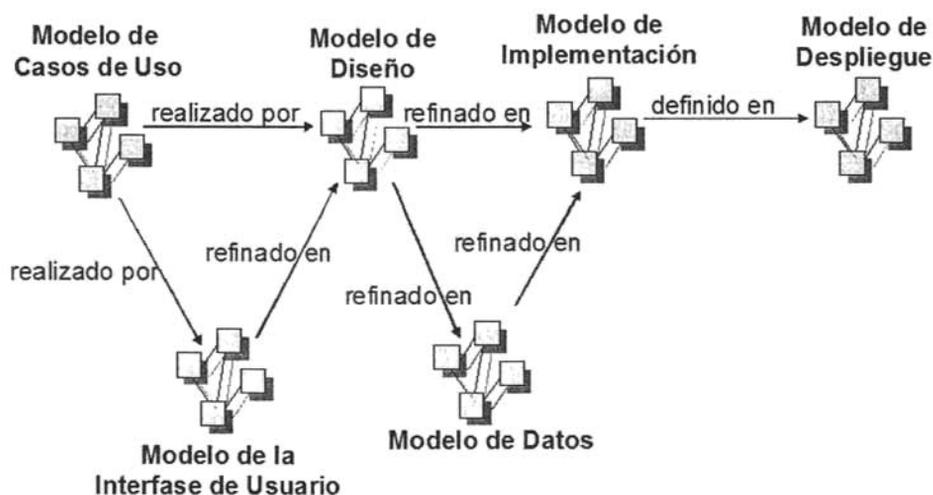


Figura C.1 Modelos principales utilizados bajo una arquitectura J2EE

*El modelo de casos de uso* describe el comportamiento del sistema visto a través de su interacción con su ambiente, y sirve como un contrato entre el cliente y el equipo de desarrollo. El modelo de casos de uso es el fundamento de los otros modelos, ya que contiene una especificación completa del comportamiento del sistema. Es usado como un insumo esencial para las actividades de análisis, diseño y pruebas.

*El modelo de interfaz de usuario* describe lo que el usuario verá cuando interactúe con el sistema. Describe las pantallas, el contenido dinámico de las mismas, y la forma en la que el usuario navegará a través de ellas ejecutando los casos de uso del sistema. Sirve como contrato entre las capas de presentación y la de negocio de un sistema, y describe cómo los resultados de la capa de negocio son presentados al usuario.

*El modelo de diseño* describe la realización de los casos de uso del sistema, y sirve como una abstracción del modelo de implementación. Este modelo abarca el análisis y diseño de clases, el diseño de componentes, paquetes y subsistemas, así como la realización de casos de uso. El modelo de diseño es un insumo esencial para las actividades de implementación.

*El modelo de datos* describe los datos persistentes del sistema, incluyendo los comportamientos definidos en la base de datos, tales como procedimientos almacenados, desencadenantes, restricciones, etc. El modelo de datos define además un conjunto de elementos, los cuales representan el almacenamiento físico de cada uno de los elementos persistentes del mismo modelo. En este modelo se define la relación entre las clases de diseño persistentes y las estructuras de datos persistentes. El modelo de datos es necesario cuando las estructuras de datos persistentes no pueden ser inferidas desde la estructura de clases persistentes en el modelo de diseño. En este caso una asociación, del modelo de diseño a los mecanismos de almacenamiento persistente, deberá ser definida explícitamente.

*El modelo de implementación* es un modelo UML de la implementación del modelo de diseño y del modelo de datos. Describe la implementación de directorios y archivos necesarios para construir y administrar el sistema en el ambiente de desarrollo. Estos archivos incluyen tanto los archivos operacionales, tales como los ejecutables y los archivos de configuración, como también los archivos de código fuente desde los cuales son generados los archivos ejecutables. Con las herramientas adecuadas, la consistencia entre la representación visual del modelo de implementación y los archivos físicos puede ser mantenida usando ingeniería inversa.

*El modelo de despliegue* muestra la configuración de los nodos con el sistema en tiempo de ejecución, la comunicación entre ellos, así como los elementos del modelo de implementación que son instalados en cada uno de ellos. Este modelo describe la distribución del comportamiento a través de los nodos.

Cada modelo captura un aspecto particular del sistema o es una proyección del sistema desde un punto de vista. Esos puntos de vista deberán describir lo siguiente:

- Uno o más modelos del sistema y vistas de esos modelos.
- Las partes involucradas interesadas en esas vistas.
- Los temas de interés que deberán ser tratados en esas vistas.

Las vistas que generalmente son necesarias para la descripción de una arquitectura son las siguientes:

- La vista de requerimientos. En ella se describen tanto los requerimientos funcionales como los no funcionales que impactan de forma significativa en la arquitectura del sistema. Los artefactos definidos en RUP más representativos para esta vista son el modelo de casos de uso y la especificación suplementaria.

- La vista lógica. En ella se describen los mecanismos clave de diseño, los elementos de diseño relevantes desde el punto de vista de la arquitectura, sus interdependencias y la organización de esos elementos dentro de subsistemas y capas. Los artefactos definidos en RUP más representativos para esta vista son el modelo de diseño, el modelo de interfase de usuario, el modelo de datos y las guías base de diseño.
- La vista de implementación. En ella se describen los elementos clave de implementación tales como código fuente, código ejecutable y módulos. El artefacto definido en RUP más representativo para esta vista es el modelo de implementación.
- La vista de proceso. En ella se describen procesos e hilos de ejecución, y sus correspondientes elementos de implementación. El artefacto definido en RUP más representativo para esta vista es el modelo de diseño.
- La vista de despliegue. Esta vista contiene la descripción de varios nodos del sistema, tales como computadoras, equipos de comunicaciones, así como la ubicación lógica de los elementos del sistema o procesos de ejecución en cada uno de esos nodos. Los artefactos definidos en RUP más representativos para esta vista son el modelo de despliegue y el de implementación.

Este conjunto de vistas es conocido como un modelo de arquitectura de 4+1 vistas y fue primeramente propuesto por Philippe Kruchten en 1995.

El documento de arquitectura de software es el principal artefacto donde la arquitectura del sistema es descrita, y contiene las referencias a todos los otros artefactos que, desde el punto de vista de la arquitectura, son significativos. En otras palabras, si alguien estuviera interesado en entender la arquitectura de un sistema, entonces deberá empezar por estudiar el documento de arquitectura.

A continuación se describe brevemente la estructura de un documento de arquitectura, con una pequeña descripción del contenido de cada sección.

1. Introducción. Esta sección describe el alcance de la arquitectura y lista las referencias de todos los documentos y artefactos de soporte. Entre los artefactos que generalmente son referidos se encuentran: el glosario, la especificación suplementaria, los modelos del sistema, y algunos estándares utilizados.
2. Representación de la arquitectura. Esta sección describe la forma en la que se representa la arquitectura. Generalmente será suficiente la representación de la arquitectura del sistema mediante las 4+1 vistas descritas anteriormente. No obstante, el arquitecto podrá incorporar vistas o definiciones adicionales, y estas también deberán ser descritas en esta sección. Para cada una de las vistas que se definan se deberá incorporar una sección en el documento.
3. Vista de requerimientos. Esta sección es generalmente dividida en dos partes. En la primera, se deberán discutir los casos de uso arquitectónicamente significativos.

Los casos de uso que, desde el punto de vista de la arquitectura, son significativos deberán ser desarrollados en las iteraciones de la fase de elaboración para:

- Llevar a cabo las pruebas de concepto de los principales elementos del sistema y su interrelación.
- Investigar áreas con cierto grado de incertidumbre, que permitan ayudar con la definición de riesgos, así como con la implementación de estrategias de mitigación y eliminación de los mismos.

En la segunda parte de esta sección se deberán discutir los objetivos de la arquitectura del sistema y sus restricciones, independientemente de la definición de cualquier caso de uso. Se refiere a la especificación suplementaria y describe los requerimientos del sistema que tienen un impacto significativo en la arquitectura del sistema. En este apartado también se deberán describir los requerimientos en cuanto a las propiedades del sistema tales como rendimiento, tamaño, escalabilidad, seguridad y privacidad. También se documentarán restricciones especiales tales como el uso del producto, integración con sistemas heredados, estrategia de reutilización, herramientas de desarrollo requeridas, así como la estructura del equipo de trabajo y la planeación.

4. Vista lógica. Esta sección describe las partes que, desde el punto de vista de la arquitectura, son significativas en el modelo de diseño y en el modelo de interfase de usuario. Deberá empezar con una discusión de los mecanismos de diseño significativos y patrones que muestren la estructura del sistema. Deberá describir la composición del sistema a través de capas y subsistemas, así como de cualquier otro elemento de diseño arquitectónicamente significativo. Las abstracciones claves del sistema son también consideradas como arquitectónicamente significativas y deberán ser descritas en esta sección. Las descripciones de los elementos de diseño arquitectónicamente significativos deberán hacer referencia a partes específicas del modelo de diseño, donde las responsabilidades de los elementos y sus relaciones son descritas.

En casos de una complicada o inusual interfase de usuario, la descripción de los elementos de diseño arquitectónicamente significativos deberá también hacer referencia a partes del modelo de interfase de usuario y deberá explicar las dependencias entre esos elementos de diseño y los elementos de la interfase de usuario, como pantallas, flujos de pantallas y formas de entrada.

Esta sección deberá ilustrar la forma en la que los elementos de diseño arquitectónicamente significativos operan juntos. La mejor forma de hacerlo es mostrar la realización de unos cuantos casos de uso debidamente seleccionados de los casos de uso arquitectónicamente significativos, y explicar la forma en la que los elementos de diseño contribuyen a su funcionalidad.

5. Vista de proceso. Esta sección es particularmente importante para entender la operación de procesos e hilos de ejecución de un sistema. La plataforma J2EE soporta la administración de recursos de cómputo, por lo que, en casos de sistemas

basados en esa tecnología, la vista de procesos es generalmente usada solo para describir la concurrencia de aplicaciones y/o procesos. En tales casos, la vista de proceso puede describir:

- Los servlets usados para crear y administrar múltiples hilos de control.
  - Cualquier aplicación java que se integra al sistema con su propia administración de hilos de control, así como los elementos de diseño con los cuales esas aplicaciones se comunican.
  - El uso de mensajes entre componentes para administrar la concurrencia en la aplicación.
6. Vista de implementación. Esta sección es una vista del modelo de implementación, con el énfasis en la estructura del modelo (paquetes y directorios). Esta vista es importante porque la estructura del modelo de implementación tiene un impacto relevante, por ejemplo, en el desarrollo y en las pruebas concurrentes.
  7. Vista de despliegue. Esta sección es una vista del modelo de despliegue y del modelo de implementación. Muestra una o más configuraciones de red sobre las cuales el sistema es instalado y operado. También muestra la ubicación de contenedores en nodos, así como la ubicación de módulos J2EE en esos contenedores. En algunos casos será relevante mostrar también la manera en la que los elementos de diseño son ubicados en los nodos.
  8. Propiedades del sistema. El documento de arquitectura concluye con una discusión de la forma en la que la arquitectura del software contribuye a lograr las metas y restricciones de la arquitectura descritas en la sección de requerimientos. Esta sección generalmente incluye también una discusión de las propiedades claves del sistema tales como rendimiento y portabilidad.

**Apéndice**

**D**

Glosario



**Actividad.** Una unidad de trabajo que produce un resultado significativo en el contexto del proyecto.

**Actor.** Un actor define un conjunto coherente de roles que los usuarios de un sistema pueden jugar cuando interactúan con él. La instancia de un actor puede ser realizada por un individuo, un sistema externo, o un dispositivo externo. Los actores están descritos en los modelos de casos de uso.

**Administración de cambios.** La actividad de controlar y registrar los cambios a los artefactos.

**Administración de la configuración.** En RUP, un flujo de trabajo o disciplina cuyo propósito es identificar, definir, controlar modificaciones y versiones de los elementos del sistema. Reporta y registra el estatus de dichos elementos, así como los requerimientos de cambios sobre ellos. Se asegura que los elementos del sistema estén completos, correctos y consistentes.

**Administración del proyecto.** En RUP, un flujo de trabajo o disciplina cuyo propósito es planear y administrar el desarrollo del proyecto.

**Ambiente.** En RUP, un flujo de trabajo o disciplina cuyo propósito es definir y administrar el ambiente en el cual el sistema es desarrollado. Se definen descriptores del proceso, administración de la configuración y herramientas de desarrollo.

**Arquitectura de software.** Se refiere a la estructura y organización del software, describiendo sus componentes y la relación entre ellos. También define aspectos de uso, funcionalidad, rendimiento, confiabilidad, reutilización, así como restricciones económicas y tecnológicas.

**Artefacto.** Una pieza de información que es producida, modificada o usada por un proceso. Un artefacto puede ser un modelo, un elemento de modelo o un documento, y deberá mantenerse bajo un control de versiones.

**Calidad.** Las características de un artefacto que satisfacen o exceden un conjunto de requerimientos predefinidos, y que son sujetas de evaluación a través de métricas y criterios preestablecidos.

**Capa.** Un específico grupo de paquetes de clases en un modelo con el mismo nivel de abstracción.

**Caso de desarrollo.** El proceso de ingeniería de software usado por una organización de desarrollo. Éste es usado como una configuración o personalización del Proceso Unificado de Rational y adaptado a las necesidades de un proyecto.

**Casos de uso.** Una secuencia de acciones que ejecuta un sistema, y que dan como resultado un determinado valor a un actor.

**CCPE.** Coordinación de Control del Padrón Electoral.

**CECYRD.** Centro de Cómputo y Resguardo Documental, lugar de trabajo donde residen los equipos de cómputo servidores que procesan y mantienen la base de datos del padrón electoral.

**Ciclo de desarrollo.** Una ejecución completa de las cuatro fases de RUP: inicio, elaboración, construcción y transición; el tiempo transcurrido entre el comienzo de la fase de inicio y el fin de la fase de transición.

**Clase.** Una descripción de un conjunto de objetos que comparten la misma responsabilidad, relaciones, operaciones, atributos y semántica.

**CNC.** Centro Nacional de Cómputo del Registro Federal de Electores.

**COFIPE.** Código Federal de Instituciones y Procedimientos Electorales.

**Componente.** Es una parte casi independiente, no trivial, y reemplazable, que lleva a cabo una funcionalidad bien delimitada en el contexto de una arquitectura bien definida.

**Configuración.** Un conjunto de artefactos de un proyecto que definen una versión particular de un sistema o parte de él.

**Construcción.** La tercera fase definida en RUP, en la cual el software es llevado de una definición de arquitectura al punto en el que se encuentra listo para ser transferido a la comunidad de usuarios.

**CRC.** Centro Regional de Cómputo, lugar de trabajo donde se procesaban los trámites de inscripción al padrón electoral antes del inicio de operaciones del SIIRFE.

**DDS.** Dirección de Desarrollo de Sistemas.

**Defecto.** Anomalía. Como ejemplos se tienen omisiones e imperfecciones encontradas durante las primeras fases del ciclo de vida de desarrollo, así como incidencias encontradas en un software lo suficientemente maduro para ser probado u operado.

**DERFE.** Dirección Ejecutiva del Registro Federal de Electores.

**Despliegue.** En RUP, un flujo de trabajo o disciplina cuyo propósito es asegurar el éxito en la transición del sistema desarrollado a la comunidad de usuarios. Los principales artefactos producidos son materiales de capacitación y procedimientos de instalación.

**Disciplina.** En RUP, un conjunto de actividades que se relacionan entorno a un tema de interés dentro de un proyecto de desarrollo de software.

**Diseño.** La parte del proceso de desarrollo de software cuyo propósito principal es decidir la forma en la que el sistema será implementado. Durante el diseño, las decisiones tácticas y estratégicas son tomadas para cubrir los requerimientos funcionales y de calidad de un sistema.

**Elaboración.** En RUP, la segunda fase del proceso, en la cual se detallan los alcances del producto y su arquitectura.

**Escenario.** Una instancia de un caso de uso.

**Evolución.** La vida del software después de su ciclo de desarrollo inicial; cualquier ciclo subsiguiente durante el cual el producto crece.

**Fase.** Es el periodo en el que se alcanzan ciertos objetivos, se desarrollan ciertos artefactos, y se toman decisiones para continuar o no a la siguiente fase. En RUP, se definen cuatro fases para un ciclo de desarrollo: inicio, elaboración, construcción y transición.

**Flujo de trabajo.** Secuencia de actividades ejecutadas que producen como resultado, un valor observable para un individuo o equipo de trabajo.

**Flujo de trabajo detallado.** Es un grupo de actividades que a menudo son ejecutadas de forma integral para producir un resultado específico. En RUP, cada disciplina o flujo de trabajo se describe por flujos de trabajo detallados.

**FUAR.** Formato Único de Actualización con Recibo. Se trata de la solicitud que se genera al momento en el que un ciudadano acude a tramitar su credencial para votar.

**IFE.** Instituto Federal Electoral.

**Implementación.** En RUP, un flujo de trabajo o disciplina cuyo propósito es implementar clases y llevar a cabo pruebas unitarias.

**Incremento.** La diferencia entre dos versiones.

**Inicio.** En RUP, la primera fase del proceso de ingeniería de software, en la cual se definen los objetivos del sistema, se conocen a todas las partes involucradas y se determina la factibilidad del proyecto.

**Integración.** La actividad en el desarrollo de software en la cual los componentes de software son combinados en un solo paquete ejecutable.

**Involucrado.** Cualquier persona o representante de una organización quien se ve afectado por el resultado de un proyecto o cuya opinión deberá ser tomada en cuenta. Puede tratarse de usuarios finales, clientes, administrador del proyecto, desarrolladores, etc.

**Iteración.** Una secuencia de actividades en base a un plan y criterios de evaluación, que dan como resultado una versión del producto.

**MAC.** Modulo de Atención Ciudadana, lugar de trabajo donde se registran los ciudadanos para tramitar su credencial para votar con fotografía y su inscripción al padrón electoral.

**Modelo.** Una descripción completa de un sistema desde una perspectiva particular.

**Modelo de diseño.** Un modelo que describe la realización de los casos de uso; sirve como una abstracción del modelo de implementación y su código fuente.

**Modelo de implementación.** Una colección de componentes y subsistemas que los contienen.

**Nodo.** Un objeto físico en tiempo de ejecución, que representa un recurso de cómputo, en el que se albergan objetos y componentes del sistema.

**Objeto.** Una entidad que define un estado y un comportamiento. El estado se representa por medio de atributos y relaciones, y el comportamiento se representa por métodos y operaciones. Un objeto es una instancia de una clase.

**OLTP.** Procesamiento de transacciones en línea.

**OLAP.** Procesamiento analítico en línea.

**Petición de cambio.** Un requerimiento para cambiar un artefacto. Se deberá documentar el origen y el impacto del problema actual, la solución propuesta y su costo.

**Pizarrón de control de cambios.** Su función es proveer de un mecanismo de control central para asegurar que cada petición de cambio es considerada, autorizada y coordinada apropiadamente.

**Pruebas.** En RUP, un flujo de trabajo o disciplina cuyo propósito es integrar y probar el sistema.

**Realización de casos de uso.** Una descripción de un camino en particular de un caso de uso, en el que se analizan el comportamiento de objetos y su interacción.

**RedIFE.** La red de comunicaciones del Instituto Federal Electoral.

**Requerimientos.** Una descripción de una condición o capacidad de un sistema. En RUP, un flujo de trabajo o disciplina cuyo propósito es definir lo que hará el sistema; la actividad más relevante es el desarrollo del modelo de casos de uso.

**Riesgo.** Una situación o impedimento que tiene una probabilidad significativa de afectar negativamente el éxito de los principales objetivos del proyecto.

- Role.** Una definición del comportamiento y responsabilidades de un individuo o conjunto de individuos trabajando en equipo, dentro del contexto de una organización de desarrollo de software. Un role es responsable de uno o más artefactos y realiza un conjunto de actividades.
- RUP.** Proceso Unificado de Rational, un proceso de ingeniería de software definido por Rational.
- SAC.** Subsistema de Actualización a la Cartografía Electoral, el cual forma parte del SIIRFE.
- SACD.** Subsistema de Actualización y Control Documental, el cual forma parte del SIIRFE.
- SAP.** Subsistema de Actualización al Padrón Electoral, el cual forma parte del SIIRFE.
- SC.** Subsistema de Consultas, el cual forma parte del SIIRFE.
- SD.** Subsistema de Depuración, el cual forma parte del SIIRFE.
- SIE.** Subsistema de Información Ejecutiva, el cual forma parte del SIIRFE.
- SIEX.** Subsistema para Interoperabilidad con Fuentes Externas, el cual forma parte del SIIRFE.
- SIIRFE.** Sistema Integral de Información del Registro Federal de Electores.
- SS.** Subsistema de Soporte, el cual forma parte del SIIRFE.
- Transición.** En RUP, la cuarta fase del proceso de desarrollo, en la cual el software producido es entregado para su operación a la comunidad de usuarios.
- UML.** Lenguaje de Modelado Unificado. Un Lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- Vista.** Una descripción simplificada de un modelo que es visto desde una determinada perspectiva, en la que se omiten entidades que no son relevantes para ésta.

## BIBLIOGRAFÍA

- [1] Bodof Stephanie Green Dale. "The J2EE Tutorial". Addison Wesley. (2002).
- [2] CONSTITUCIÓN POLÍTICA DE LOS ESTADOS UNIDOS MEXICANOS.
- [3] CÓDIGO FEDERAL DE INSTITUCIONES Y PROCEDIMIENTOS ELECTORALES Y OTROS ORDENAMIENTOS ELECTORALES.
- [4] Jacobson, Ivar., Booch, Grady., Rumbaugh, James. "El Proceso Unificado de Desarrollo de Software". Addison Wesley. (2000).
- [5] Jacobson, Ivar., Booch, Grady., Rumbaugh, James. "El lenguaje unificado de modelado". Addison Wesley. (2000).
- [6] Jones Capers. "Assessment and Control of Software Risks". Yourdon Press. (1994)
- [7] Khawar Zaman Ahmed, Cary E. Umrysh. "Developing Enterprise Java Applications with J2EE and UML". Addison Wesley. (2002).
- [8] Larman Craig. "UML y Patrones". Prentice Hall. (1999).
- [9] Peter Eeles, Kelli Houston, Wojtek Kozaczynski. "Building J2EE applications with the Rational Unified Process". Addison Wesley. (2003).
- [10] Philippe Kruchten. "The Rational Unified Process an introduction". Second Edition. Addison Wesley. (2000).
- [11] Qatrani Terry. "Visual Modeling with Rational Rose 2002 and UML". Addison Wesley. (2002).
- [12] The Standish Group. "Charting the Seas of Information Technology". The Standish Group. (1994).