

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



***TECNOLOGÍA DE INFORMACIÓN ORIENTADA A OBJETOS
APLICADA A LA GESTIÓN DE PROYECTOS EN UNA
PLATAFORMA WEB***

TESIS

Para optar el Título Profesional de:
INGENIERO DE SISTEMAS

Victor José Manuel Reyna Vargas
Erick Teodoro Phocco García

Lima Perú
2004

INDICE

CAPITULO I. RESUMEN	12
1.1 La Empresa.....	12
1.2 Spring.....	12
1.3 El Mercado de la Empresa.....	13
1.4 Organización de la Empresa.....	13
CAPITULO II. INTRODUCCIÓN	15
2.1 Definición y Planteamiento del problema.....	15
2.1.1 Definición del Problema.....	15
2.1.2 Planteamiento del Problema.....	15
2.2 Importancia del Tema.....	17
2.3 Objetivo del estudio.....	18
2.3.1 Objetivo Principal	18
2.3.2 Objetivos Secundarios.....	18
CAPITULO III. DIAGNÓSTICO ACTUAL	19
3.1 Definiciones de Reingeniería	19
3.2 Metodología rápida de Reingeniería	20
3.3 Enunciado del caso.....	21
3.4 Etapa 1: Preparación.....	21
3.4.1 Marco teórico.....	21
3.4.2 Implementación en Royal Systems.....	22
3.5 Etapa 2: Identificación	24
3.5.1 Marco teórico.....	24
3.5.2 Implementacion en Royal Systems.....	25
3.6 Etapa 3: Visión	27
3.6.1 Marco Teórico	27
3.6.2 Implementación en Royal Systems.....	28
3.7 Etapa 4A: Solución – Diseño Técnico	34
3.7.1 Marco Teórico	35
3.7.2 Implementación en Royal Systems.....	35

3.8 Etapa 4b: Solución – Diseño Social	37
3.8.1 Marco Teórico	37
3.8.2 Implementación en Royal Systems.....	37
3.9 Etapa 5: Transformación	39
3.9.1 Marco Teórico	39
3.9.2 Implementación en Royal Systems.....	39
CAPITULO IV. SISTEMA PROPUESTO.....	40
4.1 Elaboración de propuesta de solución.....	43
4.2 Captura de Requisitos	46
4.2.1 Modelo de casos de uso.....	47
4.2.2 Actor.....	47
4.2.3 Caso de uso	47
4.3 Análisis	50
4.3.1 Modelo de análisis.....	51
4.3.2 Clase del análisis	51
4.3.2.1 Clases de interfaz.....	51
4.3.2.2 Clases de entidad.....	52
4.3.2.3 Clases de control.....	52
4.3.3 Realización de caso de uso – análisis	53
4.3.3.1 Diagramas de clases	54
4.3.3.2 Diagramas de colaboración	54
4.3.3.3 Flujo de sucesos – análisis	54
4.4 Diseño	55
4.4.1 Modelo de diseño	55
4.4.2 Clase del diseño.....	56
4.4.3 Realización de caso de uso – diseño	56
4.4.3.1 Diagramas de clases	57
4.4.3.2 Diagramas de secuencia	57
4.4.3.3 Flujo de sucesos – diseño	57
4.4.4 Modelo de despliegue	58
4.4.5 Diseño de los casos de uso.....	60
4.4.5.1 Identificación de clases del diseño participantes.....	60
4.4.5.2 Descripción de las interacciones entre objetos del diseño	60

4.4.6	Diseño de una clase	61
4.4.6.1	Esbozar la clase del diseño	61
4.4.6.2	Identificar operaciones.....	62
4.4.6.3	Identificar atributos	62
4.4.6.4	Identificar asociaciones	62
4.4.7	Modelo de datos relacional.....	62
4.5	Implementación	65
4.5.1	Modelo de implementación.....	66
4.5.2	Componente.....	66
4.5.3	Plan de Integración de Construcciones	66
4.5.4	Construcción de la Base de Datos.....	68
4.5.5	Construcción de los componentes base del sistema	70
4.5.6	Construcción del Sistema de Control de Proyectos	72
4.5.6.1	Módulo de Ingreso, Bienvenida y Salida del SCPWeb.....	74
4.5.6.2	Módulo del Administrador del Sistema.....	75
4.5.6.3	Módulo del Jefe de Área.....	84
4.5.6.4	Módulo del Miembro de Area.....	89
4.6	Prueba.....	91
4.6.1	Modelo de pruebas.....	92
4.6.2	Caso de prueba.....	92
4.6.3	Procedimiento de prueba.....	94
4.6.4	Planificar prueba	94
4.6.5	Diseñar prueba.....	95
4.6.5.1	Diseño de los casos de prueba de integración.....	95
4.6.5.2	Diseño de los casos de prueba de sistema.....	95
4.6.5.3	Identificación y estructuración de los procedimientos de prueba	95
4.6.6	Realizar pruebas de integración	96
4.6.7	Realizar prueba de sistema	96
4.6.8	Evaluar prueba	96
CAPITULO V. ANÁLISIS ECONÓMICO		99
5.1	Estructura de la Inversión	99
5.1.1	Fija	99
5.1.1.1	Activos Fijos	99

5.1.1.2 Intangibles	99
5.1.2 Capital de Trabajo	99
5.2 Estructura de Costos	100
5.3 Alternativas de Financiamiento.....	101
5.3.1 Activos Fijos	101
5.3.2 Capital de Trabajo	102
5.3.3 Ratios Financieros.....	103
5.4 Evaluación Económica	103
5.4.1 Valor Actual Neto	103
5.4.2 Tasa Interna de Retorno.....	103
5.4.3 Beneficio / Costo	103
5.4.4 Tiempo de Recuperación de Capital.....	103
CAPITULO VI. CONCLUSIONES Y RECOMENDACIONES.....	104
GLOSARIO DE TÉRMINOS	107
BIBLIOGRAFIA	115
ANEXOS	116
ANEXO A. Modelo de Casos de Uso del Sistema de Control de Proyectos.....	117
ANEXO B. Detalle de los Casos de Uso del Sistema de Control de Proyectos	120
ANEXO C. Realización de Casos de Uso – Análisis del Sistema de Control de Proyectos	136
ANEXO D. Realización de Casos de Uso – Diseño del Sistema de Control de Proyectos	151
ANEXO E. Diccionario de Datos del Sistema de Control de Proyectos	176

FIGURAS

Figura 1.1 Organigrama de Royal Systems SAC	14
Figura 4.1 Un proceso de desarrollo de software	41
Figura 4.2 Los cinco flujos de trabajo fundamentales.....	43
Figura 4.3 Representación UML de un actor	47
Figura 4.4 Representación UML de un caso de uso.....	48
Figura 4.5 Modelo de Casos de Uso para el Sistema de Control de Proyectos.....	48
Figura 4.6 Estereotipos de clases estándar que podemos utilizar en el análisis.....	52
Figura 4.7 Clases del análisis para el Sistema de Control de Proyectos	53
Figura 4.8 Diagrama de despliegue para el Sistema de Control de Proyectos	59
Figura 4.9 Modelo de datos relacional para el Sistema de Control de Proyectos ...	64
Figura 4.10 Implementación de tablas en el MS SQL 2000.....	69
Figura 4.11 Relación entre tablas en MS SQL 2000.....	69
Figura 4.12 Proyecto SCPLib	70
Figura 4.13 Clase de control Gestor Usuario.....	71
Figura 4.14 Código del componente GUsuario.....	71
Figura 4.15 Proyecto SCPWeb	72
Figura 4.16 Página de Ingreso	74
Figura 4.17 Página de Bienvenida con las acciones que puede realizar el usuario	75
Figura 4.18 Página de Salida	75
Figura 4.19 Página de acciones sobre las Aplicaciones del Sistema.....	76
Figura 4.20 Crear Aplicación.....	76
Figura 4.21 Página de ingreso de criterios de búsqueda - Aplicación.....	77
Figura 4.22 Resultados de la búsqueda - Aplicación	77
Figura 4.23 Consultar Aplicación.....	78
Figura 4.24 Modificar Aplicación	78
Figura 4.25 Página de acciones sobre las Empresas del Sistema	78
Figura 4.26 Crear Empresa	79
Figura 4.27 Página de ingreso de criterios de búsqueda - Empresa	79
Figura 4.28 Resultados de la búsqueda - Empresa.....	80
Figura 4.29 Consultar Empresa.....	80
Figura 4.30 Modificar Empresa	81
Figura 4.31 Página de acciones sobre los Usuarios del Sistema	81
Figura 4.32 Crear Usuario.....	82

Figura 4.33	Página de ingreso de criterios de búsqueda - Usuario	82
Figura 4.34	Resultados de la búsqueda - Usuario.....	83
Figura 4.35	Consultar Usuario.....	83
Figura 4.36	Modificar Usuario	84
Figura 4.37	Página de acciones sobre los Proyectos del Sistema.....	84
Figura 4.38	Crear Proyecto	85
Figura 4.39	Página de ingreso de criterios de búsqueda - Proyecto.....	85
Figura 4.40	Resultados de la búsqueda - Proyecto	86
Figura 4.41	Consultar Proyecto.....	86
Figura 4.42	Modificar Proyecto.....	87
Figura 4.43	Página de Reportes del Sistema	87
Figura 4.44	Control Planeado	88
Figura 4.45	Control Planeado / Ejecutado	88
Figura 4.46	Uso de Recursos.....	88
Figura 4.47	Página de acciones sobre Informes Semanales.....	89
Figura 4.48	Consultar Proyecto.....	89
Figura 4.49	Resultados de la búsqueda – Informe Semanal	90
Figura 4.50	Ingreso del Informe Semanal	91

CUADROS

Cuadro 4.1 Hardware usado en la construcción del sistema	45
Cuadro 4.2 Software usado en la construcción del sistema	46
Cuadro 4.3 Descripción de los actores del sistema	49
Cuadro 4.4 Plantilla para describir un caso de uso.....	49
Cuadro 4.5 Relación entre clases de entidad y tablas de Base de Datos	63
Cuadro 4.6 Relación entre la clase de entidad Usuario y tablas de Base de Datos.....	63
Cuadro 4.7 Tablas Estado y Situación	64
Cuadro 4.8 Relación entre clases de diseño y componentes construidos	71
Cuadro 4.9 Relación entre clases de interfaz y componentes construidos	73
Cuadro 6.1 Inversión Inicial Total	100
Cuadro 6.2 Estructura de Costos	101

DESCRIPTORES TEMÁTICOS

- Control de Proyectos
- Flujo de Trabajo Fundamental
- Lenguaje Unificado de Modelado
- Proceso Unificado de Racional
- Reingeniería
- Royal Systems S.A.C.
- Visual Studio .NET

CAPITULO I

RESUMEN

1.1 LA EMPRESA

Royal Systems SAC, inició sus operaciones en 1997, centrando sus esfuerzos en el desarrollo de un sistema integrado al cual denominó **Spring**.

Debido a la tecnología imperante en ese momento se eligió como herramienta de desarrollo al Power Builder, en ambiente Cliente - Servidor y como manejador de Base de Datos a Microsoft SQL Server.

1.2 SPRING

Spring fue desarrollado siguiendo las siguientes fases:

- Fase I - Parte Central del Sistema (1996-1998): Incluye los sistemas de Cuentas por Pagar, Contabilidad, Logística, Importaciones, Activo Fijo, Facturación, Cuentas por Cobrar y Planillas.
- Fase II - Desarrollo de Sistemas Adyacentes (1999 – 2001): Incluye los sistemas de Ajustes por Inflación, Conciliación Bancaria, Mantenimiento de Equipos, Ingeniería, Producción, Costos de Producción, Presupuestos, Control Presupuestal, Flujo de Caja, Recursos Humanos, Módulos OLAP y WorkFlow.

Se trabaja siempre sobre un solo sistema **Core**. Dependiendo de las necesidades de la empresa, se realiza una etapa de *personalización* (desarrollo adicional).

1.3 EL MERCADO DE LA EMPRESA

El producto está dirigido a la mediana empresa: Empresas Comerciales, Productivas y de Servicios, que no necesariamente cuente con un área de sistemas.

Desde el año 1997 hasta la fecha se llevó en forma paralela el desarrollo y la implementación. Se implementaba lo que ya se tenía desarrollado mientras se iba desarrollando los nuevos módulos.

En el mercado elegido hay poca competencia importante. Las empresas medianas normalmente no desarrollan sus propios sistemas debido al alto costo de ello, y tampoco están en la capacidad económica de comprar software extranjero como SAP, BAAN, JDEdwards, etc. Sin embargo también son reacias a tomar cualquier software si es que este no tiene las recomendaciones de otras empresas.

1.4 ORGANIZACION DE LA EMPRESA

Royal Systems es una empresa de tipo familiar. El Gerente General y la Gerente Administrativo son esposos y dueños de la empresa. El Gerente General se encarga asimismo de las ventas.

Bajo ellos se encuentran 2 Gerentes de Desarrollo, cada uno de los cuales lleva el control de un cierto número de aplicaciones. Cada Gerente de Desarrollo controla en forma independiente a sus personal de Desarrollo e Implementación.

Se complementa la empresa con el siguiente personal:

- Gerentes de Proyecto
- Implementadores
- Analistas Programadores
- Personal Administrativo

De acuerdo a lo anteriormente mencionado, el organigrama de Royal Systems SAC esta representado por la Figura 2.1.

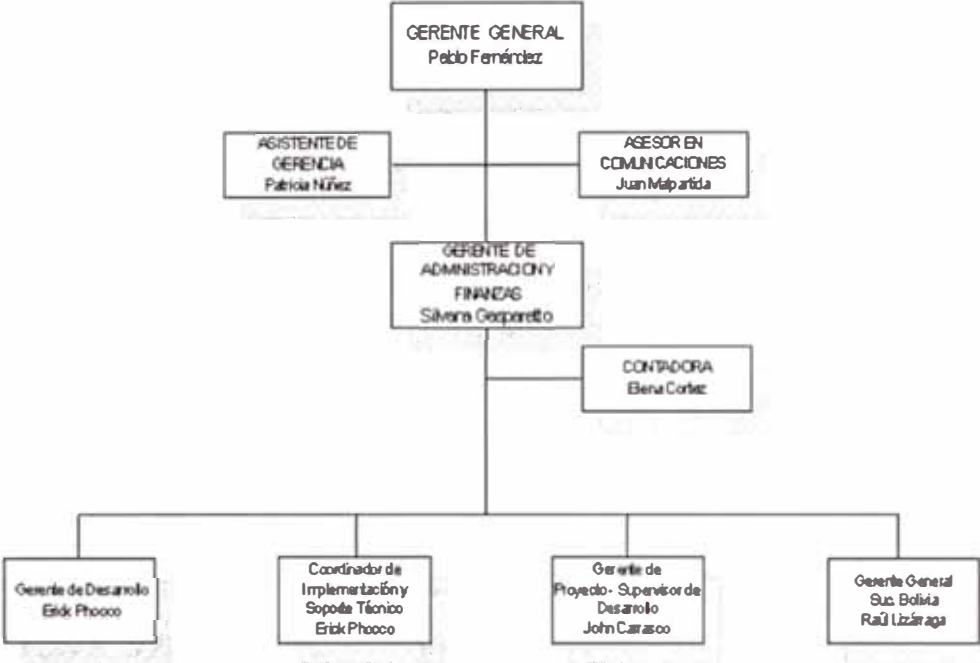


Figura 1.1 Organigrama de Royal Systems SAC

CAPITULO II

INTRODUCCION

2.1 DEFINICION Y PLANTEAMIENTO DEL PROBLEMA

2.1.1 DEFINICION DEL PROBLEMA

Royal Systems SAC es una empresa que inicia sus actividades en el 1996 y su función principal es desarrollar software. Su crecimiento fue desordenado, ya que dependía en gran parte de los proyectos que tenía. Llegó a tener un poco más de 44 empleados, entre personal de Desarrollo, Implementación y Administrativo. Sin embargo, no existía un adecuado control de las actividades del personal debido a la informalidad del trabajo “por proyectos”, lo que trajo como consecuencia grandes costos a la empresa, que al final pusieron en riesgo su viabilidad financiera.

Debido a la naturaleza del trabajo, el personal asignado a los proyectos se manejaba en forma casi independiente, bajo la dirección de un Jefe de Proyecto. El control de este personal era manual.

2.1.2 PLANTEAMIENTO DEL PROBLEMA

El principal desencadenante de los problemas al interior de Royal Systems fue el crecimiento desordenado de la cantidad de empleados en la empresa, producto de la buena cantidad de proyectos que se tuvo en los primeros años de funcionamiento. Esta gran cantidad de proyectos, en cierta forma, justificaba el descontrol. A mediados del 2001 se comenzaron a

presentar problemas en el Flujo de Caja de la empresa, como consecuencia de la recesión económica en la cual entro el país.

Se decidió entonces hacer por primera vez, un estudio de Costos de los Proyectos de la empresa, (que hasta ese momento nunca había habido necesidad de hacer). Entre las conclusiones más importantes se tuvieron:

- Algunos proyectos que arrojaban pérdidas importantes se llevaban toda la utilidad que dejaban el resto de los proyectos. Esta pérdida se debía a que se había asignado mucha más gente y por mucho más tiempo del que se había considerado al momento de hacer la venta. Por otro lado no había un control adecuado del proyecto.
- Algunas aplicaciones dejaban mucho mayor utilidad que otras, debido en parte a la forma de trabajo del personal asignado y a que requerían menor cantidad de trabajo durante cada personalización.
- Faltaba un adecuado control de gastos administrativos.

De Noviembre del 2001 a Abril del 2002 se comenzaron a aplicar la mayor parte de los correctivos indicados en el Informe de Costos.

- Hubo una reducción de personal de casi 30% en total, diferenciando los porcentajes de acuerdo a la aplicación y los proyectos en los cuales se encontraba el personal.
- Hubieron áreas de negocio que se cerraron debido a la baja demanda. El personal paso a ser contratado bajo demanda.
- Hubo una reducción de gastos administrativos de aproximadamente el 25%.
- Con estos correctivos la empresa se adecuó a su nuevo tamaño y equilibró su flujo de caja.

Adicionalmente como parte de las recomendaciones del Informe de Costos se tenía:

- El producto principal se encontraba maduro y estable, pero estaba empezando a entrar a la etapa de obsolescencia como consecuencia de la finalización del ciclo de vida de la herramienta de desarrollo. Se recomendaba migrar en forma paulatina todo el sistema a una plataforma que se adecuara sobre Internet.
- Se debía desarrollar un sistema de control de los proyectos y del personal para evitar volver a caer en los mismos problemas. De preferencia este sistema debía realizarse en la nueva plataforma Internet, para facilitar el acceso del personal a la herramienta, ya que estos la mayor parte del tiempo se encontraban fuera de la empresa.

2.2 IMPORTANCIA DEL TEMA

La principal importancia de este tema de tesis es mostrar con hechos, la forma como una reingeniería al control de proyectos en Royal Systems SAC puede controlar de manera más exacta el principal recurso de la empresa : Horas-Hombre de su personal, valiéndose para esto de un adecuado sistema informático que provea de información oportuna del estado actual de cada proyecto.

Además, otro aspecto importante es el uso del Proceso Unificado de Rational como metodología de trabajo para el correcto desarrollo de sistemas informáticos, el cual está dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Además, este proceso utiliza el Lenguaje Unificado de Modelado, un lenguaje que produce dibujos los cuales nos sirven para comunicarnos claramente entre cliente – analista – desarrollador.

Finalmente, la tesis tiene como objetivo secundario el servir como guía para todas aquellas personas interesadas en desarrollar sistemas informáticos teniendo como punto de partida el uso del Proceso Unificado de Rational.

2.3 OBJETIVO DEL ESTUDIO

2.3.1 OBJETIVO PRINCIPAL

Realizar una reingeniería al proceso de control de proyectos dentro de Royal Systems SAC.

2.3.2 OBJETIVOS SECUNDARIOS

- Usar el Proceso Unificado de Rational como metodología de desarrollo de software.
- Usar el Lenguaje Unificado de Modelado como herramienta de modelado de sistemas.
- Hacer uso del Visual Studio .Net como herramienta de desarrollo del Sistema de Control de Proyectos.

CAPITULO III

DIAGNOSTICO ACTUAL

El objetivo del presente capítulo es presentar el marco teórico dentro del cual se inicia un proceso de cambio en Royal Systems. Este proceso de cambio nos sirve como marco referencial para determinar donde nace la necesidad del análisis, diseño, implementación de un sistema de software, el cuál se explica con mayor detalle en el Capítulo IV.

Es importante reconocer que el proceso de cambio realizado en Royal Systems fue un proceso empírico, en el cuál no se aplicó de manera directa alguna metodología y por tanto sería incorrecto llamarlo Reingeniería propiamente dicha, ya que como veremos a lo largo de este capítulo, no se cumplen de manera total los criterios de una metodología de Reingeniería de Procesos.

3.1 DEFINICIONES DE REINGENIERIA

“Reingeniería es el rediseño rápido y radical de los procesos estratégicos de valor agregado – y de los sistemas, las políticas y las estructuras organizacionales que los sustentan – para optimizar los flujos y la productividad de una organización.”

“Un proceso es una serie de actividades relacionadas entre sí que convierten insumos en productos. Los procesos se componen de tres tipos principales de actividades:

- Las que agregan valor. Actividades importantes para los clientes
- De traspaso. Que mueven el flujo de trabajo a través de fronteras que son principalmente funcionales, departamentales u organizacionales
- De Control. Para controlar los traspasos a través de las fronteras mencionadas”

3.2 METODOLOGÍA RÁPIDA DE REINGENIERIA

Esta metodología tiene cinco etapas y permite a las organizaciones obtener resultados rápidos y sustantivos efectuando cambios radicales en los procesos estratégicos de valor agregado. Utiliza técnicas administrativas como modelación de procesos, medida del desempeño, análisis de flujo de trabajo, etc. La metodología esta diseñada para que la utilicen equipos de reingeniería internos a la empresa, sin tener que valerse mucho de expertos de fuera.

La metodología consta de 5 etapas:

- Preparación
- Identificación
- Visión
- Solución. Diseño Técnico y Diseño Social
- Transformación

Las dos primeras etapas se realizan una sola vez y capacitan a la empresa para resolver que procesos rediseñar y en *qué orden*. Las tres últimas etapas se repiten una vez para cada uno de aquellos procesos. Al final de la 2da etapa definiremos el proceso sobre el cual realizaremos las 3 últimas etapas. Es importante especificar que las aplicaciones de Royal Systems se clasifican en 2 grandes áreas; nosotros nos limitaremos a

analizar las 3 últimas etapas sobre el área de Aplicaciones Financieras. La otra área es de manejo similar.

3.3 ENUNCIADO DEL CASO

Royal Systems es una empresa que desarrolla e implementa sistemas informáticos. La empresa inicia sus operaciones en 1997 y entra con su producto al mercado en 1998 definiendo como su segmento del mercado el formado por las medianas empresas, que prefieren comprar un sistema informático externo a fin de reducir costos.

Entre 1998 y 2000, vive años de bonanza, consigue contratos con varias empresas en las cuales instala el software de manera satisfactoria, formándose una buena cartera de clientes.

Pero es a partir del 2001 que su cartera de nuevos clientes disminuye, en parte por la situación económica del país, pero también debido a la ausencia de un control efectivo de los recursos asignados a cada proyecto; el control de proyectos era ineficiente. La empresa termina el 2001 con pérdidas. A inicios del 2002 y la empresa se cuestiona que debe hacer para salir de la situación en la que se encuentra.

3.4 ETAPA 1: PREPARACION

3.4.1 MARCO TEORICO

En esta etapa, la empresa debe reconocer que existe la necesidad de rediseñar, generalmente como resultado de un cambio. Esta necesidad casi siempre está motivada por el dolor, el temor y la ambición. También se debe tener un consenso dentro de la administración ejecutiva a favor de la reingeniería.

Se debe seleccionar, constituir y capacitar un equipo de reingeniería y este equipo debe producir un plan del proyecto y los comienzos de una Plan de Gestión del Cambio para hacer frente a las dificultades sociales que la Reingeniería de Procesos crea inevitablemente.

3.4.2 IMPLEMENTACION EN ROYAL SYSTEMS

El punto determinante para que la empresa se de cuenta que hay algo que se tiene que hacer es el cambio en la situación financiera. Se esbozan varias posibles razones para esta situación: mala situación del mercado, bajas ventas, sobrecostos en la implementación, falta de control en los proyectos, aparición de nuevos competidores, etc.

Los motivos principales dentro del marco teórico serían el *Dolor* por la mala situación financiera de la empresa y el *Temor*, ya que también aparecen soluciones de empresas grandes tipo SAP, que sacan productos para las empresas medianas a un costo similar al del sistema de Royal Systems. Royal Systems es una empresa familiar, por lo tanto el patrocinio necesario para realizar la reingeniería nace de las dos personas dueñas y Gerentes de la empresa.

Los objetivos de este proceso son los siguientes:

- Recuperar el nivel de utilidades que había tenido en los años precedentes en un período de 2 años
- Reducir los costos de operación de sus procesos a fin de poder aprovechar el crecimiento del mercado una vez que la situación económica del mercado mejore.

Aquí podemos identificar algunas características de la metodología que no se llevaron a cabo en Royal Systems:

- El Tiempo. Según la metodología, el proceso debería durar entre 6 y 18 meses. En Royal Systems, se decidió que esto sería un proceso

progresivo cuya primera etapa debería tomar un máximo de 6 meses, debido a la urgencia de ver resultados de este proceso. Actualmente, a fines del 2003 este proceso aun continúa.

- El Costo. Este proceso no tiene personal asignado en forma exclusiva o algo parecido, sino que es una actividad que forma parte de las funciones normales del personal. Se puede considerar que el costo es mínimo y hundido.
- El Riesgo. ¿Qué pasa si fracasa el proyecto de Reingeniería de Procesos? En este punto, la metodología considera que Reingeniería son *Avances Decisivos*, no cambios incrementales. Y dichos avances requieren con frecuencia que uno se desprenda de lo conocido y seguro. En Royal Systems se hizo lo contrario, el cambio era un proceso incremental y progresivo.

Según la metodología, en esta etapa se debe seleccionar y capacitar un equipo el cuál debería representar todas las áreas funcionales interesadas y se les debe dar el tiempo suficiente para la dedicación al proyecto. En Royal Systems, este proceso de cambio se inicia por la parte donde mayor información se tenía: la parte contable y nace como un Estudio de Reducción de Costos, por lo que se encarga la dirección a un Gerente y cuyo equipo lo conformaba una asistente contable. Como vemos, no estaban todas las áreas interesadas, ni tampoco tenían todo el tiempo suficiente. La capacitación del Equipo no existió, salvo los conocimientos que tenía el Gerente por haber recibido cursos afines anteriormente.

Finalmente, debería haber un Plan de Gestión del Cambio, ya que se sabe que habrá una resistencia al cambio que introduzca la Reingeniería de Procesos, pero que este cambio debe ser administrado para que el proyecto salga adelante. Es importante aquí la comunicación, ya que una vez que se inicia la reingeniería, todo el mundo se entera y si la comunicación no es administrada, abundarán los rumores y los temores de la gente, la productividad del trabajo decaerá y crecerá la resistencia al cambio.

En Royal Systems, hubo una reunión del Gerente General con los empleados, en la cual se mencionó la difícil situación en la que se encontraba la empresa y la necesidad de una reorganización para poder continuar en el mercado. La evaluación de los puestos sería en base a la productividad de los proyectos en que el empleado haya participado. Sin embargo, esta comunicación se realizó una sola vez, porque la siguiente información que supieron los empleados es que algunos puestos se estaban eliminando. Esto trajo problemas como se menciona en los párrafos precedentes, pero fue una decisión de la Gerencia. La idea era que mientras más rápido acabara esta primera fase, sería mucho mejor para todos.

3.5 ETAPA 2: IDENTIFICACION

3.5.1 MARCO TEORICO

En esta etapa, se debe desarrollar un modelo del negocio, con procesos orientados al cliente; identificar los procesos estratégicos de valor agregado y recomendar los procesos específicos como objetivos de mayor impacto para la reingeniería.

Además, debemos definir que es lo que nuestro cliente espera y tratar de obtener métricas de los actuales niveles de rendimiento o las variaciones del rendimientos. Se deben definir las “entidades” que son abstracciones que se manifiestan en uno o más casos específicos. Por ejemplo: Entidad “empleado”, casos “Juan”, “Pedro”, etc. Asimismo, se deben definir los atributos y estado de estas entidades.

Luego, se deben definir los procesos, los cuales son actividades que se realizan sobre las entidades. Finalmente se deben priorizar estos procesos en base a:

- Impacto. La contribución actual y potencial de cada proceso a las metas de la empresa
- Magnitud. Los recursos que consume o utilice cada proceso
- Alcance. El tiempo, el costo, el riesgo y el cambio social implícito en la reingeniería de cada proceso

3.5.2 IMPLEMENTACION EN ROYAL SYSTEMS

El cliente de Royal Systems busca tener un software estable que cubra las necesidades de información de su empresa y que lo haga a un costo razonable, entendiéndose por costo a la inversión inicial y el costo de mantenimiento posterior.

Al tratar de conseguir las métricas de satisfacción del cliente, podemos mencionar:

- La mayoría de las empresas están satisfechas con el desempeño técnico y económico del sistema, ya que cubre sus necesidades y además han reducido costos con su implementación.
- Existen situaciones de demora en la atención de requerimientos para las nuevas implementaciones, producto de la falta de capacitación del personal y de picos generados por el desarrollo de nuevos requerimientos para las aplicaciones.

Las entidades identificadas son las siguientes:

- Aplicaciones. Son el producto de la empresa, son subsistemas independientes que tienen características propias, las cuales son desarrolladas e implementadas por áreas distintas dentro de la empresa
- Desarrolladores. Personal que desarrolla las Aplicaciones
- Implementadores. Personal que implementa las Aplicaciones
- Cliente
- Proyecto. Por medio del cual se une la aplicación, el implementador y el cliente

- Requerimiento. Generadas durante la implementación del Proyecto.

Los procesos identificados son los siguientes:

- Desarrollar las Aplicaciones y mejorarlas continuamente
- Vender las Aplicaciones a los Clientes, lo cual da origen a un Proyecto
- Administrar los Proyectos
- Administrar los Requerimientos de los Clientes, que nacen como consecuencia de los Proyectos, de mantenimiento o de cotizaciones a los Clientes
- Vender los contratos de mantenimiento de la Aplicación
- Atender las consultas de los usuarios, posteriores al período de implementación de las aplicaciones (Help Desk)
- Dar un servicio Post – Venta a los Clientes
- Administrar las funciones administrativas, contables, financieras y de personal de la empresa

Estos procesos se pueden agrupar en 3 grandes áreas, en orden de importancia:

- Control de Proyectos. Que comprende:
 - Desarrollo y mejora continua de las aplicaciones
 - Control de los requerimientos de desarrollo o de implementación, originados por el proyecto por cotizaciones adicionales, por el mantenimiento de las aplicaciones ya implementadas.
 - Atención de consultas de los usuarios (Help Desk)
- Actividades de Ventas
 - Venta Inicial, de Contratos de Mantenimiento y de Cotizaciones adicionales
 - Servicios Post – Venta
- Administrativas
 - Administración del Personal
 - Control de Gastos Administrativos y Operativos de la empresa

Para las 3 últimas etapas, hemos elegido el proceso de “Administrar los Requerimientos de los Clientes”, por ser el más importante y representativo de la empresa. La automatización de este proceso es el que da origen al software, tema de esta tesis.

3.6 ETAPA 3: VISION

3.6.1 MARCO TEORICO

En esta etapa, se debe desarrollar una visión del proceso, es decir, hasta dónde queremos llegar con el proceso “re - diseñado”. Comprende las siguientes actividades:

- Entender la estructura y flujo del proceso. Producir la información suficiente acerca del proceso actual para determinar para qué sirve y los defectos que tiene. No se busca analizar el proceso ya que tenemos la intención de reemplazarlo con el proceso rediseñado.
- Identificar actividades de valor agregado. Para reforzar las que agregan valor y minimizar las que no lo agregan
- Benchmarking del proceso. Tratar de comparar el rendimiento del proceso actual y determinar los factores que producen tal rendimiento
- Visualizar los ideales: Interno y Externo. Estos ideales son normalmente a un mediano o largo plazo
- Definir Subvisiones. Definir metas intermedias coherentes con nuestra Visión Ideal del Proceso

La estructura del proceso se define en función de actividades, pasos, insumos, productos y estímulos.

- Las Actividades son las principales subdivisiones de un proceso
- Cada Actividad produce un resultado: su Producto

- Cada Actividad utiliza el resultado material o informativo de otras actividades: sus Insumos
- Si un insumo proviene de fuera del proceso y también inicia una actividad, lo denominamos un Estímulo
- Los Pasos son subdivisiones de las Actividades

El tiempo que se necesita para completar una unidad de trabajo en una actividad se puede descomponer en los siguientes componentes:

- Tiempo en Tránsito. Tiempo que se necesita para que los insumos de una actividad o paso lleguen hasta él.
- Tiempo en Cola. Tiempo que media desde el momento en que se reciben los insumos hasta que se actúa sobre ellos.
- Tiempo en Proceso. Tiempo que transcurre desde que se empieza a trabajar en los insumos hasta que se termina el trabajo en los productos.
- Tiempo Efectivo del Proceso. Tiempo real en que se trabaja para convertir los insumos en productos.
- Tiempo Fuera de Cola. Tiempo que transcurre desde el momento en que se terminan los productos hasta que se empiezan a pasar a la nueva actividad o paso.

Las actividades se pueden clasificar en:

- De Valor Agregado. Aquellas que tienen un impacto sobre las medidas de rendimiento del proceso. Son aquellas que hacen algo que el cliente quiere.
- De Control
- Otros. Manejo de oficinas y locales, administración, comunicaciones y actividades de coordinación.

3.6.2 IMPLEMENTACION EN ROYAL SYSTEMS

En el proceso que hemos elegido, “Administración de los Requerimientos”, hay 3 fuentes por donde nacen los requerimientos:

- Los que nacen como consecuencia de una nueva implementación
 - Actividades normales de implementación
 - Modificaciones menores, formatos, nuevos reportes
 - Implementación de nuevas funciones que el sistema no tiene
- Lo que nacen como consecuencia del mantenimiento del sistema
 - Atribuibles al usuario, debido a un mal manejo del sistema
 - Atribuibles al sistema, debido a un error del sistema
 - Nuevos requerimientos por desarrollar (Cotizaciones)
- Los que nacen como parte del Plan de Mejoras del Sistema
 - Nuevas funciones o reportes del sistema

En el proceso se pueden identificar 3 áreas o actores

- Administración (1 persona). Sus productos son:
 - Cotizaciones para el Cliente (Usuario)
 - Control de los Mantenimientos realizados por Contrato
 - Orden de Facturación por Cotizaciones
- Implementación (9 personas). Sus productos son:
 - Atención al Usuario
 - Requerimientos Documentados para Desarrollo
 - Control de Calidad sobre los programas
 - Documento Sustentatorio de Cotizaciones
- Desarrollo (3 personas). Sus productos son:
 - Programas informáticos
 - Evaluaciones técnicas y económicas

Las actividades se pueden clasificar de la siguiente manera:

- Actividades de Implementación, que el implementador puede resolver
- Problemas o Consultas reportados, que el implementador puede resolver
- Problemas o Consultas reportados, que el implementador no puede resolver y lo transfiere a Desarrollo
- Nuevos Desarrollos

- Actividades de Control de Calidad

Para el proceso elegido, se han definido las siguientes actividades:

	Actividades / Pasos	Tipo de Actividad	De Valor Agregado	De Control	Otras
1	Actividades de Implementación				
1.1	Hacer el Plan de Proyecto.			X	
1.2	Generar las actividades de acuerdo al Plan,				X
1.3	Realizar las Actividades.		X		
2	Problemas o Consultas Reportados				
2.1	Recepcionar el Requerimientos : Vía Mail, Telefono o por Visita Personal.			X	
2.2	Evaluar el Requerimiento en la Oficina.			X	
2.3	Reproducir el problema o consulta.			X	
2.4	Responder al usuario, si se encuentra la solución.		X		
2.5	Transferir el Problema a Desarrollo si no se encuentra la solución			X	
2.6	Desarrollo evalúa el requerimiento			X	
2.7	Desarrollo resuelve el requerimiento si es algo inmediato.		X		
2.8	Desarrollo registra el requerimiento en la cola de Desarrollo si va a tomar más tiempo.				X
2.9	Implementación realiza el Control de Calidad del cambio			X	
2.1	Implementación responde al usuario.		X		
0					

	Actividades / Pasos	Tipo de Actividad	De Valor Agregado	De Control	Otras
3	Nuevos Desarrollos : Cotizados o Mantenimiento				
3.1	Administración Recepciona el Requerimiento			X	
3.2	Administración transfiere el Requerimiento a Desarrollo.			X	
3.3	Desarrollo hace la Evaluación Técnica.			X	
3.4	Desarrollo hace la Evaluación Económica.			X	
3.5	Implementación hace el Documento Sustentatorio de la Cotización			X	
3.6	Administración hace la Cotización Económica			X	
3.7	Administración envía la Cotización Económica y el Documento Sustentatorio.		X		
3.8	Si el Cliente acepta la Cotización, Administración informa a Desarrollo.			X	
3.9	Desarrollo lo registra en la Cola de Pendientes				X
3.10	Desarrollo resuelve el Requerimiento		X		
3.11	Implementación hace el Control de Calidad			X	
3.12	Implementación envía los cambios al Usuario		X		
3.13	Implementación informa a Administración que el requerimiento ha sido resuelto.			X	
3.14	Administración inicial el trámite de facturación y cobranza			X	

	Actividades / Pasos	Tipo de Actividad	De Valor Agregado	De Control	Otras
4	Nuevos Desarrollos : Mejoras o Por Proyecto				
4.1	Implementación documenta el Requerimiento,		X		
4.2	Desarrollo lo registra en la Cola de Pendientes				X
4.3	Desarrollo resuelve el Requerimiento		X		
4.4	Implementación realiza el Control de Calidad			X	
4.5	Implementación entrega el Requerimiento al Usuario		X		

En Royal Systems se identificaron varios problemas en el proceso:

- Había requerimientos que se “perdían” o quedaban traspapelados en Desarrollo ya que el Control se llevaba por listas independientes por cada implementador.
- Había demoras en algunos requerimientos debido a que cuando Desarrollo resolvía el requerimiento, el implementador no se encontraba y a veces no venía hasta el día siguiente, produciéndose una demora en el Control de Calidad.
- Muchas veces, los requerimientos que eran sencillos entraban a la cola de Pendientes y esperaban largo tiempo a ser resueltos.
- Cuando no había implementadores en la oficina, Desarrollo tenía que contestar las llamadas de los usuarios, produciendo una demora en sus funciones normales.
- El Control de Calidad era apresurado por falta de tiempo, produciéndose errores en los Clientes.

- La Cola de Desarrollo era normalmente larga, con un tiempo de espera promedio de 10 días.
- Las implementaciones a veces tomaban más tiempo de lo necesario, debido a la falta de metodología, falta de herramientas para agilizar la implementación, etc.
- A veces ocurrían problemas con los usuarios debido a que durante el proceso de venta se había ofrecido funciones que el sistema no tenía.
- Había intervalos de tiempo en los que la carga de trabajo de implementación no estaba bien distribuida debido a que los implementadores tenían asignadas empresas específicas para cada uno.

Finalmente, la Visión del proceso fue la siguiente:

- Actividades de Implementación.
 - Se crearía un puesto de Supervisor de Implementación, que desarrollaría la Metodología de Implementación y velaría por su cumplimiento, además coordinaría el trabajo de los implementadores. El puesto estaría a cargo de uno de los implementadores actuales de manera exclusiva, manteniendo sus funciones de implementación.
 - Se desarrollarían programas que faciliten las actividades de carga de datos y verificación de datos que realizaban los implementadores.
- Problemas o Consultas Reportados
 - Se crearía el puesto de Help Desk, encargado de atender los requerimientos telefónicos y vía email de los usuarios, a parte de coordinar los Requerimientos de todos los implementadores. Uno de los implementadores se haría cargo de esta función en forma exclusiva. Esta persona se quedaría todo el día en la oficina contestando las llamadas de los usuarios, coordinando los requerimientos con los demás implementadores y asegurando que

la documentación de los Requerimientos este disponible para cuando Desarrollo la requiera.

- Se realizarían 2 niveles de Control de Calidad. El 1er nivel lo realizaría la persona de Help Desk y para el caso de los requerimientos sencillos estos sería suficiente. Para los requerimientos más complejos, el implementador a cargo realizaría un 2do Control de Calidad.
- Se desarrollaría un aplicativo informático que permitiría el registro de los requerimientos, fuesen estos de Desarrollo o de Implementación y que permitiría realizar una seguimiento efectivo a los mismos. Este sistema debería estar en Internet para que los implementadores puedan acceder a él desde donde se encuentren. Cada implementador registraría sus propios requerimientos y a partir de allí, Help Desk gestionaría con Desarrollo su resolución y enviaría un email al implementador cuando el requerimiento estuviese resuelto.
- Nuevos Desarrollos
 - Los nuevos requerimientos serían enviados directamente a Help Desk, que coordinaría con Desarrollo la evaluación.
 - Si se trata de Mantenimiento, la evaluación sería enviada directamente al cliente y si este la aprueba, recién se informaría a Administración.

Como se puede apreciar, esta visión es totalmente posible en el corto plazo ya que lo que se está implementando no son cambios drásticos, sino por el contrario, son cambios en los controles de los subprocesos para reducir algunos pasos. La herramienta que ayudará en este cambio es el nuevo aplicativo informático de Control de Proyectos.

3.7 ETAPA 4A: SOLUCION – DISEÑO TÉCNICO

3.7.1 MARCO TEÓRICO

El propósito de esta etapa es producir un diseño del proceso capaz de realizar la Visión. Debemos describir la tecnología que se usará, las normas, procedimientos, sistemas y controles empleados por el proceso rediseñado. También se deben producir planes preliminares para el desarrollo de sistemas y procedimientos, requerimientos de hardware necesario, mejoras de las instalaciones, pruebas, conversión e implementación.

Existen 3 factores que hacen posible la reingeniería de procesos:

- Tecnología
- Información
- Potencial Humano

3.7.2 IMPLEMENTACION EN ROYAL SYSTEMS

Se decidió que el sistema de “Control de Proyectos” se realizaría en 3 fases, siendo la primera de ellas el motivo del capítulo IV. Los requerimientos comunes se reducían a ser fácil de usar y ser desarrollado en tecnología web, de tal manera que pueda ser accedido desde Internet, facilitando de esta manera el acceso a los implementadores que normalmente se encuentran fuera de la oficina.

- 1ra Fase.- Controlar las Actividades de los Implementadores y Desarrolladores.
 - Planear las Actividades del Personal, relacionándolas a las empresas y proyectos a los cuales corresponde.
 - Facilitar al personal el reporte de avance de sus actividades.
 - Información básica: Empleado, Empresa, Proyecto, Fecha de Inicio, Término Esperado y Término Real, etc.
 - Implementación de reportes especializados (Control Planeado, Control Planeado / Ejecutado y Uso de Recursos)

- 2da Fase.- Controlar los Requerimientos de los Proyectos.
 - Registrar el requerimiento y toda la documentación posible del mismo: archivos word, excel, pantallas en formato jpg, etc.
 - Registrar las especificaciones técnicas del desarrollo: nuevas tablas o campos por crearse, nuevos parámetros, nuevos reportes o procesos creados, SQL necesarios para instalar el cambio, etc.
 - Registrar los comentarios de la persona que realizó el Control de Calidad.
 - Anexar el Documento de Implementación del cambio.
 - Indicar la conformidad del Usuario.
 - Información básica del requerimiento. -Empresa Cliente que solicita el Requerimiento, incluyendo el nombre del Usuario que presenta el requerimiento, Aplicación involucrada, Descripción detallada, Documentos anexos, Historial del avance, Fecha de Ingreso, Término Esperado y Término Real, Situación en la que se encuentra en cualquier momento, Origen del requerimiento: Problema, Mantenimiento, Plan de Mejoras, Proyecto, Personas involucradas: Implementadores y Desarrolladores, Nivel de Complejidad
 - Debe permitir sacar información estadística: Número de Requerimientos atendidos, Tiempo Promedio de atención según su nivel de complejidad, según la aplicación, etc.

- 3ra Fase.- Base de Conocimiento.
 - Formar una Base de Conocimiento sobre las preguntas que normalmente hacen los usuarios e implementadores a fin de ser usado en futuras consultas por los clientes de la empresa y por el mismo personal.
 - Publicar Documentación del Sistema como manuales, casos estudio, Mejoras del Sistema, etc.

Otros requerimientos complementarios dentro de esta Etapa de Solución Técnica son:

- Servidor de Internet Information Server para el alojamiento de la aplicación web
- Designación de la persona que realizará las funciones de Help Desk
- Selección de la persona encargada de la Gerencia de Operaciones

3.8 ETAPA 4B: SOLUCIÓN – DISEÑO SOCIAL

3.8.1 MARCO TEÓRICO

El diseño social debe realizarse al mismo tiempo que el diseño técnico. El propósito de esta etapa es especificar las dimensiones sociales del proceso. Esta etapa produce descripciones de la organización, cargos e incentivos que se emplean en el proceso rediseñado. Finalmente, produce planes preliminares de contratación, educación, capacitación, reorganización y nueva ubicación de personal.

En esta etapa también se debe prever quiénes se pueden oponer a los cambios que se proponen y cómo se les puede motivar para que los acepten. Además, se debe definir al “Dueño del Proceso”, gerente de más bajo nivel responsable de todo el proceso.

3.8.2 IMPLEMENTACIÓN EN ROYAL SYSTEMS

En primer lugar, se tiene que facultar al personal que tiene contacto con el cliente. En este sentido, Royal Systems planificó una serie de charlas para mejorar el trato de los implementadores con los clientes y otras de motivación para el personal. En forma paralela, se desarrollará la metodología que los implementadores deberán seguir para optimizar el proceso de implantación del sistema. Este se complementaría con el desarrollo de programas que ayuden a cargar y verificar la carga de la información inicial de los clientes en el sistema.

Dentro del proceso se identifican 3 puestos y las destrezas y conocimientos necesarios.

- Supervisor de Implementación. Relaciones interpersonales, capacidad de liderazgo, habilidades de negociación, conocimientos de las aplicaciones, conocimiento de metodologías.
- Implementador. Relaciones interpersonales, habilidades de negociación, conocimiento profundo de las aplicaciones asignadas a su cargo.
- Help Desk. Relaciones interpersonales, habilidades de negociación, conocimiento mediano de todas las aplicaciones del área.

Los 3 puestos deben dominar las herramientas de computación básicas como Microsoft Word, Excel, Project, Power Point y tener conocimientos de SQL.

No es posible en este caso utilizar el modelo de “Trabajador de Caso”, el cuál pregona que una sola persona pueda manejar el proceso de principio a fin, con las consecuentes mejoras en la productividad y la calidad de atención. En nuestro caso, si bien es cierto, se nombra a un implementador como Gerente de Proyecto para que se haga cargo de todas las coordinaciones con el Cliente, hay un equipo detrás que esta compuesto por otros implementadores, los cuales lo apoyan en actividades específicas de acuerdo a la complejidad del proyecto y por personal de Desarrollo. Se nombró como “Dueño del Proceso” al Supervisor de Implementación.

En cuanto al Programa de Gestión de Cambio, este no sería necesario desarrollar debido a que los cambios en las funciones de las personas no son significativos, por lo que no se prevé un rechazo al cambio. Sobre el cambio tecnológico, tampoco es importante ya que el personal esta familiarizado con este tipo de herramientas.

3.9 ETAPA 5: TRANSFORMACION

3.9.1 MARCO TEÓRICO

El objetivo de esta etapa es realizar la Visión del proceso implementando el diseño producido en la etapa 4. Se incluye realizar pruebas con una versión piloto del proceso, realizar el pase a producción y establecer mecanismos de cambio continuo durante su vida en producción. La mayor parte de esta etapa se explicará con mayor detalle en el Capítulo IV, cuando se especifique la construcción del aplicativo informático.

3.9.2 IMPLEMENTACIÓN EN ROYAL SYSTEMS

Se decidió realizar el desarrollo del sistema en la propia empresa, debido sobretodo a razones técnicas internas: El sistema principal se encuentra desarrollado en una arquitectura Cliente – Servidor y la empresa quiere empezar a hacer desarrollos en entorno Web, entonces esta era una buena oportunidad para dar el primer paso.

Los cambios en las funciones del personal se realizaron de forma rápida, quedando dos puntos pendientes:

- Definición de la Metodología de Implementación
- Construcción del Sistema

Se establecieron los procedimientos necesarios, se designó a las personas para las funciones de Supervisor de Implementación y Help Desk y temporalmente se diseño un procedimiento alternativo al sistema utilizando Excel para centralizar todos los requerimientos.

CAPITULO IV

SISTEMA PROPUESTO

El objetivo de este capítulo es plasmar la solución sistémica e informática al problema de Royal Systems relacionado al control de proyectos. En el capítulo anterior se explicó de manera pormenorizada la situación actual de la empresa y hemos visto que no existe una forma como controlar la información relacionada a los proyectos que se llevan a cabo tales como **costo, tiempo, nivel de eficiencia, etc.**

Debido a lo anterior, la falta de un sistema informático de control de proyectos era imprescindible para la obtención de dicha información de manera rápida y oportuna, logrando de esta manera una correcta decisión al momento de analizar los proyectos. Por otro lado, dicho sistema debía lograr una total integración con los ya existentes y con los que se desarrollen a futuro manteniendo de esta manera un adecuado tratamiento a la información que por ellos circule.

Para el desarrollo de dicho software, se hizo uso del **Proceso Unificado de Rational, RUP** (por sus siglas en inglés), el cual nos sirvió como proceso de desarrollo debido a que es adecuado para transformar los requerimientos de un usuario en un sistema de software (ver Figura 4.1). Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de

organizaciones, diferentes niveles de aptitud o diferentes tamaños de proyectos.



Figura 4.1. Un proceso de desarrollo de software

El RUP está basado en componentes, lo cual quiere decir que el sistema de software está formado por componentes software interconectados a través de interfaces bien definidas. Esto es una ventaja debido a que al momento de plasmar todo nuestro análisis en código fuente, la implementación de los respectivos componentes es totalmente independiente de la lógica de funcionamiento lográndose de esta manera una total integridad al momento de la implementación del sistema.

Por otro lado, el RUP utiliza el **Lenguaje Unificado de Modelado, UML** como lenguaje de modelado para bosquejar todos los esquemas de un sistema de software. Otra ventaja importante es la aceptación mundial del UML, lo que hace muy fácil la posterior migración del desarrollo del proyectos de un equipo de trabajo a otro. Además, facilita la creación de una adecuada documentación de la forma cómo fue desarrollado el sistema, logrando de esta manera una transferencia de información muy transparente hacia otras partes.

Lo más importante al momento de construir un sistema de software de éxito, es saber lo que sus futuros usuarios necesitan y desean. Cuando hablamos de *usuario*, no sólo hacemos referencia a usuarios humanos sino también a otros sistemas. Entonces, el término *usuario* representa alguien o algo que interactúa con el sistema que estamos desarrollando. Una interacción de este tipo es un **caso de uso**. Un caso de uso no es más que un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso no sólo nos representan los requisitos funcionales de un sistema, sino guían su diseño, implementación y

prueba; es decir, **guían el proceso de desarrollo**, siendo este el motivo fundamental por el cuál se escogió al RUP como proceso de desarrollo de software.

Hoy en día existen empresas desarrolladoras de software que lo único que les interesa es el ver código, nada más. Se cree que el avance de un proyecto está relacionado a la cantidad de nuevo código que se tenga desarrollado, dejando al margen la correcta funcionalidad del sistema. Es por esto que cuando un sistema comienza a fallar, se recurre a los universalmente conocidos **parches** que no son otra cosa que una corrección a una parte del código que en el mejor de los casos no altera el funcionamiento del sistema (algo poco común debido a que los parches siempre repercuten en dicho funcionamiento).

No debemos dejar de lado lo concerniente a la arquitectura (comprendido también dentro del RUP). El concepto de arquitectura incluye tanto los aspectos estáticos como dinámicos más significativos del sistema. ¿Cómo se relacionan los casos de uso y la arquitectura? De la siguiente manera: la función corresponde a los casos de uso y la forma a la arquitectura. Debe existir una interacción entre los casos de uso y la arquitectura. Y lo más importante, la arquitectura debe diseñarse para permitir que el sistema evolucione, no sólo en su desarrollo inicial, sino también a lo largo de las futuras generaciones.

Debido a la complejidad de un sistema de software, su desarrollo puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una **iteración** que da como resultado un incremento sobre el desarrollo. En cada iteración, los desarrolladores identifican y especifican los casos de uso más 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. De esta manera

vemos que cada iteración nos permite cumplir objetivos específicos en el desarrollo del sistema, lo que conlleva a un incremento del mismo.

Podemos plasmar al RUP de la siguiente manera (ver Figura 4.2):

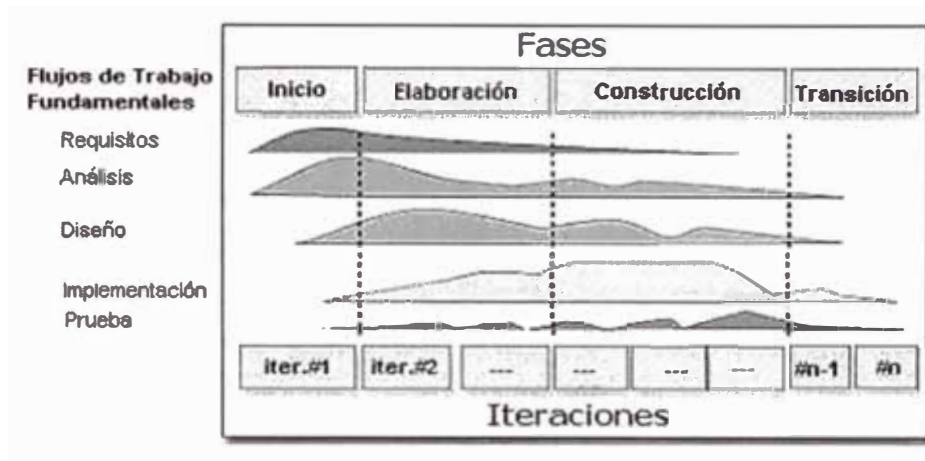


Figura 4.2. Los cinco flujos de trabajo fundamentales

Como podemos apreciar, el RUP tiene dos dimensiones:

- El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida del proceso (fases) así como su desarrollo
- El eje vertical representa las disciplinas a seguir, las cuales agrupan las actividades (flujos de trabajo) de manera lógica

Es por lo anteriormente mencionado que en Royal Systems se decidió hacer uso del RUP para llevar a cabo el desarrollo del sistema de control de proyecto, desarrollo que se verá plasmado en las siguientes páginas de este capítulo. Cabe mencionar que el desarrollo de este tema de tesis sólo tendrá en consideración el desarrollo de los flujos de trabajo en una sola iteración, esto debido a que hacerlo en varias iteraciones implicaría realizar varias veces los flujos de trabajo, escenario que sólo sería lógico realizar en un ambiente real.

4.1 ELABORACIÓN DE PROPUESTA DE SOLUCIÓN

La propuesta de solución comprende dos puntos importantes:

- Primero, se realizará un cambio total a la forma en la que se llevará a cabo el control de proyectos en Royal Systems. Esto implicará una reingeniería a los procedimientos actuales, descrito en el capítulo anterior. Dicha reingeniería quedará plasmada en los flujos de trabajo fundamentales del RUP. De esta forma, daremos las pautas de trabajo para los futuros usuarios del sistema así como también a los administradores del mismo.
- Segundo, y como se mencionó al comienzo de esta capítulo, se desarrollará (haciendo uso del RUP) un sistema informático que servirá como herramienta de toma de decisiones al momento de analizar los proyectos existentes. Este sistema informático será del tipo web, el cuál será usado dentro de la intranet de Royal Systems. Lo que se busca con esto es simular el comportamiento del mismo cuando se implemente definitivamente sobre internet.

Considerando los dos puntos anteriores, la tecnología a usarse será:

- En lo correspondiente a hardware:

Servidor	HP Proliant ML350 G3 Procesador: Intel Xeon 2.2GHz. Bus del Sistema: 400MHz. Memoria Caché: 512Kb. Nivel 2 Memoria RAM : PC133 1024 MB PC133 Controladora SCSI: Wide Ultra3 SCSI 2 canales Internos Arreglo de Disco: Opcional Smart Array 532 32MB cache Disco Duro 1: 18.2GB 10Krpm Hot Plug Disco Duro 2: 18.2GB 10Krpm Hot Plug Monitor: Compaq 15" Disquetera: 3.5 1.44MB Lectora CDROM: CD 48x Tarjeta de Red: 10/100/1000Mbps NC7760 Auto Switching Tarjeta de Red Redundante: 10/100Mbps WOL NC3123
----------	--

	<p>Chasis: Torre 5U</p> <p>Ventiladores: No Hot plug, No Redundantes</p> <p>Fuente de Poder: 500W. Hot Plug.</p> <p>Fuente de Poder Redundante: H.Plug Redund. Power Supply</p> <p>Tape Backup: Opcional DAT 12/24GB Interno</p> <p>UPS: Opcional UPS Smart King, LCD, 1.0 KVA</p> <p>Sistema Operativo: Opcional Windows 2000 Server OEM 5 Lic.</p>
Cliente	<p>Compaq EVO D310V</p> <p>Procesador: Intel Pentium 4 1.8GHz</p> <p>Memoria: 256MB DIMM DDR</p> <p>Disco Duro: 40GB IDE</p> <p>Monitor: Compaq 15 SVGA</p> <p>Video: Intel Extreme Graphics</p> <p>Sonido: Intel integrado Sin parlantes</p> <p>Lectora 1: Lectora de CD Lectora</p> <p>Red: 10/100Mbps. WOL Intel</p> <p>Teclado: Compaq</p> <p>Mouse: Compaq</p> <p>Estabilizador: Relay 1000W. 4 Salidas 220V</p>

Cuadro 4.1 Hardware usado en la construcción del sistema

- En lo correspondiente a software:

Servidor	Presentación	HTML, JavaScript
	Lógica del Negocio	ASP.NET C#
	Sistema Operativo	Microsoft Windows 2000 Server SP2 Microsoft .NET Framework
	Servidor Web	Internet Information Server 5.0
	Base de Datos	Microsoft SQL Server 7.0
	Cliente	Browser

	Sistema Operativo	Microsoft Windows XP Profesional
--	-------------------	----------------------------------

Cuadro 4.2 Software usado en la construcción del sistema

4.2 CAPTURA DE REQUISITOS

La *captura de requisitos* es el primer *Flujo de Trabajo Fundamental* dentro del RUP. Básicamente la captura de requisitos nos permite averiguar lo que se necesita, lo que se debe construir. En este caso, un sistema de software para el Control de Proyectos.

Este flujo de trabajo es el más complicado de todos los que existen dentro del RUP. ¿Por qué complicado? Un sistema debería proporcionar valor al negocio que lo utiliza y a sus clientes. Pero muchas veces, es difícil identificar o comprender qué es este valor. Esto es lo que hace de la captura de requisitos un flujo de trabajo complicado pero a la vez clave y fundamental para sacar adelante un proyecto de desarrollo de software.

Entonces, el propósito fundamental del flujo de trabajo de captura de requisitos es guiar el desarrollo hacia el sistema correcto. ¿De qué manera? Mediante una descripción de los requisitos que debe cumplir nuestro sistema. Esta descripción de los requisitos debe ser lo suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre **qué debe** y **qué no debe** hacer el sistema.

Esta captura de requisitos debe llevarse a cabo como casos de uso. Como lo mencionamos anteriormente, un caso de uso no es más que un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Por otro lado, proporcionan un medio intuitivo y sistemático para capturar los requisitos funcionales con un énfasis especial en el valor añadido para cada usuario individual o para cada sistema externo. Además, mediante el uso de los caso de uso, los analistas de

sistemas se ven obligados a pensar en términos de quiénes son los usuarios del sistema y qué necesidades u objetivos de la empresa pueden cumplir.

4.2.1 MODELO DE CASOS DE USO

De acuerdo al RUP, el *modelo de casos de uso* nos permite una clara identificación de los requisitos que debe cumplir un sistema de software. Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y las relaciones entre estos. El UML nos permite presentar el modelo en diagramas que muestran los actores y los casos de uso desde diferentes puntos de vista y con diferentes propósitos.

4.2.2 ACTOR

El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario. Cada uno de éstos se representa mediante uno o más actores. También se representa mediante uno o más actores cada sistema externo con el que interactúa el sistema. Por tanto, los actores representan terceros fuera del sistema que colaboran con el sistema. La Figura 4.3 es la representación de un actor en el UML.



Figura 4.3. Representación UML de un actor

4.2.3 CASO DE USO

Cada forma en que los actores usan el sistema de software se representa como un *caso de uso*. Los casos de uso son “fragmentos” de funcionalidad que el sistema nos ofrece, devolviendo un resultado de valor para los actores que lo utilizan. Es decir, un caso de uso nos especifica una sucesión de acciones que el sistema puede llevar a cabo interactuando con sus actores. La Figura 4.4 es la representación de un actor en el UML.



Caso de Uso

Figura 4.4. Representación UML de un caso de uso

Habiendo definido los conceptos de *actor* y de *caso de uso*, el modelo de casos de uso para el **Sistema de Control de Proyectos** vendría a estar representado por la figura 4.5. Este modelo solo contempla los casos de uso principales, los cuales van a dar forma a todo el sistema en sí. Existen otros actores y otros casos de uso, pero esto no es parte del grueso de este capítulo y por tanto, si se desea tener mayor detalle de los mismos, se pueden ubicar como parte del Anexo A.

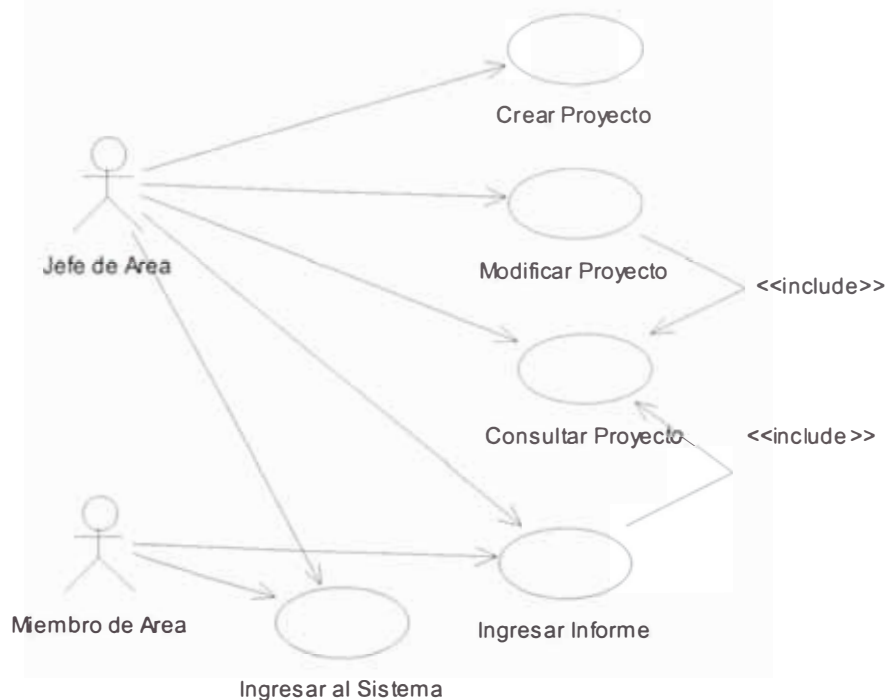


Figura 4.5. Modelo de Casos de Uso para el Sistema de Control de Proyectos

El siguiente cuadro describe brevemente lo que representa cada uno de los actores:

Jefe de Area El Jefe de Área representa a una persona que es responsable de determinada área de trabajo dentro de Royal Systems (Finanzas, Administración, RRHH, etc.).

	El Jefe de Área necesita el Sistema de Control de Proyectos para crear, modificar o consultar un proyecto o para ingresar el informe semanal de los proyectos que tiene a su cargo.
Miembro de Área	El Miembro de Área representa a una persona que pertenece a una determinada área de trabajo. El Miembro de Área necesita el Sistema de Control de Proyectos para ingresar el informe semanal de los proyectos que tiene a su cargo.

Cuadro 4.3 Descripción de los actores del sistema

A continuación, se detallará cada caso de uso. El objetivo principal de detallar un caso de uso es describir su flujo de sucesos en detalle, incluyendo cómo comienza, termina e interactúan con los actores. Para lograr esto, se usa una plantilla como la que se muestra a continuación:

Caso de uso	Nombre del caso de uso
Objetivo	Descripción informal de los objetivos del caso de uso
Actores	Actores que intervienen en el caso de uso: principales y secundarios
Precondiciones	Condiciones que deben cumplirse para que pueda realizarse el caso de uso
Pasos	Secuencia de pasos necesarios para que el caso de uso se desarrolle con éxito. Debemos mostrar las interacciones de los actores y las acciones del sistema
Variaciones	Variaciones en la secuencia de pasos
Postcondiciones	Condiciones que deben cumplirse después de realizarse el caso de uso

Cuadro 4.4 Plantilla para describir un caso de uso

Cabe mencionar que en lo correspondiente a **Pasos**, la secuencia de pasos debe ser el camino “normal”, esto es, el que el analista percibe como

que más habitualmente va a seguir un actor y aquél que proporciona el valor más obvio al actor, es decir, un resultado positivo. Entonces, según lo anterior, las descripciones de los casos de uso mostrados en la Figura 4.5 están dados por las descripciones que son parte del Anexo B.

Hasta aquí lo correspondiente a la *captura de requisitos*. En esta parte hemos identificado los requisitos funcionales principales que va a cumplir nuestro Sistema de Control de Proyectos, los cuales han sido representados como casos de uso y cuyo detalle se ha realizado valiéndonos de una plantilla fácil de comprender. Lo que viene a continuación es el análisis de lo obtenido mediante el modelo de casos de uso. Este análisis se verá en un mayor detalle en el siguiente subcapítulo.

4.3 ANALISIS

El *análisis* es el segundo *Flujo de Trabajo Fundamental* dentro del RUP. Durante el *análisis*, analizamos los requisitos que se han descrito en la *captura de requisitos*, refinándolos y estructurándolos. El objetivo de realizar este análisis es conseguir un entendimiento mucho más preciso de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero.

Analizar los requisitos en la forma de un modelo de análisis es importante porque:

- Ofrece una especificación más precisa de los requisitos que la que tenemos como resultado de la captura de requisitos, incluyendo al modelo de casos de uso.
- Describe utilizando el lenguaje de los desarrolladores, y puede de esta manera introducir un mayor nivel de formalismo y ser usado para razonar sobre el funcionamiento interno del sistema.
- Estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y en general su mantenimiento.

- Puede considerarse como una primera aproximación al modelo de diseño.

4.3.1 MODELO DE ANALISIS

El *modelo de análisis* se representa mediante un sistema de análisis que denota el paquete de más alto nivel del modelo. La utilización de otros paquetes de análisis es por tanto la mejor manera de organizar el modelo de análisis en partes más manejables que representan abstracciones de subsistemas y posiblemente capas completas del diseño del sistema.

4.3.2 CLASE DEL ANALISIS

Una *clase del análisis* representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Esta abstracción posee las siguientes características:

- Una clase de análisis se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales.
- Una clase de análisis raramente define u ofrece un interfaz en términos de operaciones y de sus signaturas.
- Una clase de análisis define atributos, aunque esos atributos también son de un nivel bastante alto.
- Una clase de análisis participa en relaciones, aunque esas relaciones son más conceptuales que sus contrapartidas de diseño e implementación.
- Las clases de análisis siempre encajan en uno de tres estereotipos básicos: de interfaz, de control o de entidad.

4.3.2.1 Clases de interfaz

Las *clases de interfaz* se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción a menudo implica recibir y presentar información y peticiones de y hacia los usuarios y los sistemas externos. Un claro ejemplo de interfaz de usuario es por ejemplo las abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, interfaces de

impresoras, sensores y terminales. Es suficiente que las clases de interfaz describan lo que se obtiene con la interacción.

4.3.2.2 Clases de entidad

Las *clases de entidad* se utilizan para modelar información que posee una vida larga y que es a menudo persistente. En consecuencia, las clases de entidad reflejan la información de un modo que beneficia a los desarrolladores al diseñar e implementar el sistema. En la mayoría de los casos, las clases de entidad suelen mostrar una estructura de datos lógica y contribuyen a comprender de que información depende el sistema.

4.3.2.3 Clases de control

Las *clases de control* representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. También se utilizan para representar derivaciones y cálculos complejos, como la lógica del negocio, que no pueden asociarse con ninguna información concreta.

La Figura 4.6 nos muestra la representación de los tipos de clase en el UML.



Figura 4.6. Estereotipos de clases estándar que podemos utilizar en el análisis

Para el Sistema de Control de Proyecto, las principales clases están representadas en la Figura 4.7. Como se puede observar, estas clases no contienen operaciones ni atributos, las cuales son características de las clases del análisis. En este flujo de trabajo fundamental, hemos decidido obviarlos por considerarlos de un nivel bastante alto. En el siguiente flujo de trabajo fundamental, *diseño*, se dará un énfasis mucho mayor tanto a las

operaciones como a los atributos por los motivos que se explicarán llegado el momento.

Clases de Interface



Clases de Control



Clases de Entidad



Figura 4.7. Clases del análisis para el Sistema de Control de Proyectos

4.3.3 REALIZACIÓN DE CASO DE USO – ANÁLISIS

Una *realización de caso de uso – análisis* es una colaboración dentro del modelo de análisis que describe como se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos del análisis en interacción.

Una realización de caso de uso posee una descripción textual del flujo de sucesos, diagramas de clases que muestran sus clases del análisis participantes y diagramas de interacción que muestran la realización de un

flujo o escenario particular del caso de uso en términos de interacciones de objetos del análisis.

4.3.3.1 Diagramas de clases

Una clase de análisis y sus objetos normalmente participan en varias realizaciones de casos de uso, por lo que es importante coordinar todos los requisitos sobre una clase y sus objetos en estas realizaciones. Para hacerlo, adjuntamos diagramas de clases a las realizaciones de casos de uso.

4.3.3.2 Diagramas de colaboración

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. En el análisis preferimos mostrar esto con diagramas de colaboración ya que nuestro objetivo es identificar *requisitos* y *responsabilidades* sobre los objetos. En estos diagramas, mostramos las interacciones entre objetos creando enlaces entre ellos y añadiendo mensajes a estos enlaces.

4.3.3.3 Flujo de sucesos – análisis

Los diagramas de una realización de caso de uso pueden ser difíciles de leer por sí mismos, de modo que es útil un texto adicional que los explique. Este texto debería escribirse en términos de objetos, particularmente objetos de control que interactúan para llevar a cabo el caso de uso.

Habiendo definido el marco teórico anterior y los principales artefactos que se utilizarán en el *análisis*, lo que quedaría por hacer es analizar cada uno de los casos de uso que hemos plasmado en el modelo de casos del Sistema de Control de Proyectos (Figura 4.5). Dada la extensión que posee la realización de los casos de uso, estos forman parte del Anexo C, junto con la descripción del flujo de sucesos – análisis.

Hasta aquí lo correspondiente al *análisis*. Se han identificado las principales clases del análisis, las mismas que se han enmarcado en los tres tipos de clases definidos en el RUP. Haciendo uso de las mismas, se ha procedido a la realización de los casos de uso - análisis del Sistema de Control de Proyectos. Entre ambos, las clases del análisis y la realización de los casos de uso – análisis, se conforma el Modelo del Análisis, el cuál servirá como punto de entrada para la realización del tercer flujo de trabajo fundamental, el *diseño*, motivo del siguiente subcapítulo.

4.4 DISEÑO

El *diseño* es el tercer *Flujo de Trabajo Fundamental* dentro del RUP. En el *diseño* modelamos el sistema y encontramos su forma para que soporte todos los requisitos que se han analizados en la *captura de requisitos*.

Los propósitos del diseño son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases.
- Descomponer los trabajos de implementación en partes más manejables que pueden ser llevados a cabo por diferentes equipos de desarrollo.

4.4.1 MODELO DE DISEÑO

El *modelo de diseño* es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos tienen impacto en el sistema a considerar, en este caso, el Sistema de

Control de Actividades. Además, el *modelo de diseño* sirve de abstracción de la implementación del sistema y es utilizada como una entrada fundamental de las actividades de implementación.

En el *modelo de diseño*, los casos de uso son realizados por las clases de diseño y sus objetos. Esto se representa por colaboraciones en el *modelo de diseño* y denota *realización de caso de uso – diseño*. Algo importante es notar la diferencia *entre realización de caso de uso – diseño* y *realización de caso de uso – análisis*. El primero nos describe cómo se realiza un caso de uso en términos de interacción entre objetos del diseño, mientras que el segundo describe cómo se realiza un caso de uso en términos de interacción entre objetos del análisis.

4.4.2 CLASE DEL DISEÑO

Una *clase del diseño* es una abstracción final de una clase o construcción similar en la implementación del sistema. Nos referimos a abstracción final en el siguiente sentido:

- El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación. En este caso, el C#.
- La visibilidad de los atributos y las operaciones de una clase de diseño se especifica con frecuencia.
- Las relaciones de las clases del diseño que se logren plasmar, a menudo tienen un significado directo cuando la clase es implementada.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.

4.4.3 REALIZACIÓN DE CASO DE USO – DISEÑO

Una *realización de caso de uso – diseño* es una colaboración en el modelo de diseño que describe cómo se realiza un caso de uso específico, y cómo se ejecuta, en términos de clases de diseño y sus objetos.

Una *realización de caso de uso – diseño* tiene una descripción de flujo de eventos textual, diagramas de clases que muestra sus clases de diseño participantes y diagramas de interacción que muestran la realización de un flujo de un caso de uso en términos de interacción entre objetos del diseño.

4.4.3.1 Diagramas de clases

Una clase de diseño y sus objetos a menudo participan en varias realizaciones de caso de uso. Además, puede darse el caso de algunas operaciones, atributos y asociaciones sobre una clase específica que son relevantes para sólo una realización de caso de uso. Para manejar todo esto, utilizamos diagramas de clases conectados a una realización de caso de uso, mostrando sus clases participantes, subsistemas y sus relaciones.

4.4.3.2 Diagramas de secuencia

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Dentro del sistema, existe un objeto de diseño que recibe el mensaje del actor. Después, el objeto de diseño llama a algún otro objeto y así sucesivamente. En el diseño, es preferible representar esto con diagramas de secuencia ya que nuestro centro de atención principal es el encontrar *secuencias de interacciones detalladas y ordenadas* en el tiempo. En los diagramas de secuencia, mostramos las interacciones entre los objetos mediante transferencias de mensajes entre objetos. El nombre del mensaje debería indicar una operación del objeto que recibe la invocación.

4.4.3.3 Flujo de sucesos – diseño

Los diagramas de secuencia de una realización de caso de uso son difíciles de interpretar por sí solos. Por esto, es bastante útil el *flujo de sucesos – diseño*, el cual es una descripción textual que explica y complementa a los diagramas. El texto debería redactarse en términos de objetos. Se recomienda que la descripción no haga mención de ninguno de los atributos, operaciones y asociaciones de los objetos.

4.4.4 MODELO DE DESPLIEGUE

El *modelo de despliegue* es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El *modelo de despliegue* se utiliza como entrada fundamental en las actividades de *diseño e implementación* debido a que la distribución del sistema tiene una influencia principal en su diseño.

Podemos observar lo siguiente sobre el modelo de despliegue:

- Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar.
- Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como *Internet, Intranet, bus* y similares.
- El *modelo de despliegue* puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación.
- La funcionalidad de un nodo se definen por los componentes que se distribuyen sobre ese nodo.
- El *modelo de despliegue* en sí mismo representa una correspondencia entre la arquitectura *software* y la arquitectura *del sistema* (el hardware).

La identificación de nodos y sus configuraciones de red juegan un papel importante al momento de plasmar el diseño de un sistema software. Las configuraciones físicas de red suelen tener una gran influencia sobre la arquitectura del software. El Sistema de Control de Proyectos se ejecutará sobre un nodo servidor y un cierto número de nodos clientes. En el nodo servidor se encontrarán alojados tanto el servidor web como el servidor de base de datos. Es decir, en el servidor se encontrará instalada la totalidad del sistema de software. Por otro lado, en el nodo cliente se presentará el resultado de la interacción entre un *actor* del sistema y el Sistema de Control de Proyecto. Esta interacción se verá reflejada en las respuestas que se den

desde el servidor web. Cabe recordar que el tipo de *actor* que interactuará con el sistema, se enmarca dentro de los actores: Administrador del Sistema, Jefe de Área y Miembro de Área. Según lo anterior, el *modelo de despliegue* para la arquitectura del sistema se verá reflejado por la Figura 4.8.

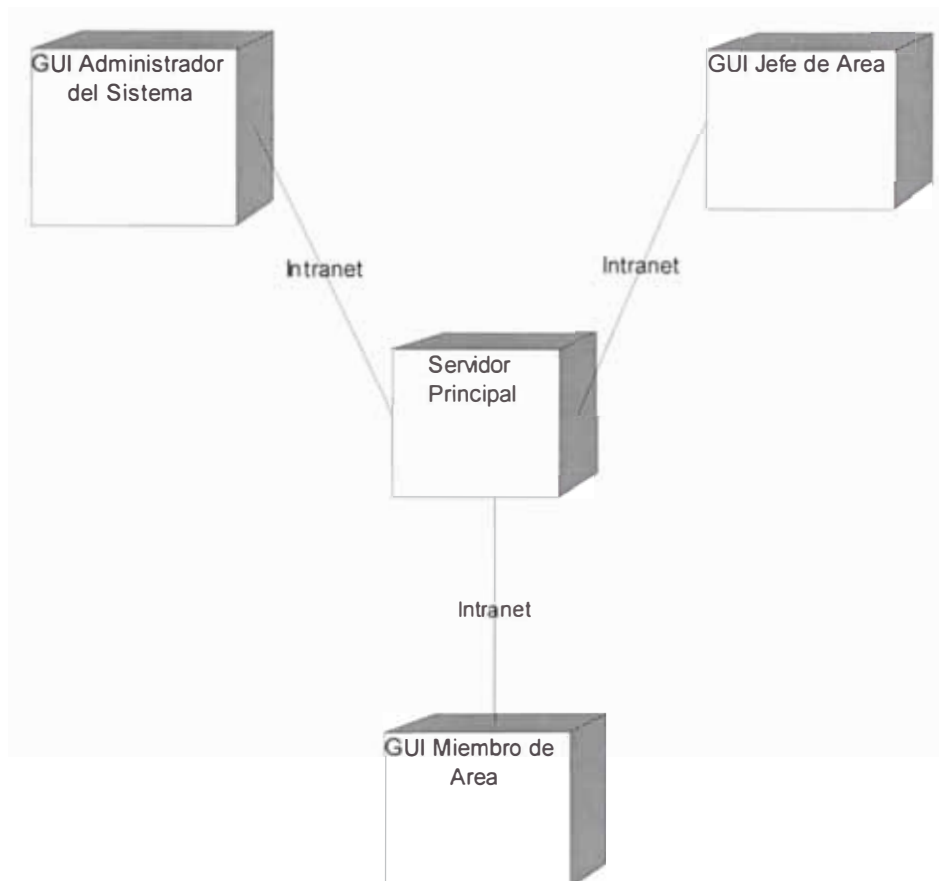


Figura 4.8. Diagrama de despliegue para el Sistema de Control de Proyectos

Del *diagrama de despliegue* podemos apreciar la distribución física de los equipos que serán parte de este Sistema de Control de Proyectos. Cabe mencionar que el hecho que el diagrama presente 3 equipos separados para la interacción de los actores con el sistema, esto no significa que sean exclusivos. Por tratarse de un sistema web, cada equipo cliente servirá para acceder al sistema como cualquier tipo de actor. Sólo bastará con presentar las credenciales adecuadas (identificador de usuario y contraseña), y el sistema presentará la interfase gráfica de usuario correspondiente, el cuál

permitirá al respectivo usuario realizar los casos de uso asignados según los visto en el *Modelo de Casos de Uso* mostrado en la Figura 4.5.

4.4.5 DISEÑO DE LOS CASOS DE USO

Los objetivos del diseño de un caso de uso son:

- Identificar las clases del diseño cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso.
- Distribuir el comportamiento del caso de uso entre los objetos del diseño que interactúan.
- Definir los requisitos sobre las operaciones de las clases del diseño.

4.4.5.1 Identificación de clases del diseño participantes

En este paso, la prioridad es el identificar las clases de diseño que se necesitan para realizar los casos de uso que forman parte del Sistema de Control de Proyectos. Para lograr este objetivo, debemos estudiar las clases del análisis que participan en la correspondiente realización de caso de uso – análisis. Debemos recoger las clases del diseño que participan en una realización de caso de uso en un diagrama de clases asociado con la realización.

4.4.5.2 Descripción de las interacciones entre objetos del diseño

El paso anterior nos proveerá de un diagrama de clases del diseño, las cuales son necesarias para realizar el caso de uso. En este paso, debemos describir cómo interactúan sus correspondientes objetos del diseño. Para logra esto, hacemos uso de diagramas de secuencia que contienen las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre éstos, que participan en el caso de uso. Para comenzar este paso, debemos estudiar la correspondiente realización de caso de uso – análisis. Para crear un diagrama de secuencia, debemos comenzar por el principio del flujo del caso de uso, y después seguir este flujo paso a paso, decidiendo qué objetos del diseño y qué interacciones de actores son necesarias para realizar cada paso.

Dada la extensión que posee el realizar los pasos anteriores, estos forman parte del Anexo E, junto con la descripción del flujo de sucesos – diseño.

4.4.6 DISEÑO DE UNA CLASE

El propósito de diseñar una clase es crear una clase del diseño que cumpla su papel en las realizaciones de los casos de uso.

4.4.6.1 Esbozar la clase del diseño

Como primer paso, necesitamos esbozar una o varias clases del diseño. Cuando se dan como entrada una o varias clases del análisis, los métodos utilizados dependen del estereotipo de la clase del análisis:

- Diseñar clases de interfaz es dependiente de la tecnología de interfaz específica que se utilice.
- Diseñar clases de entidad que representen información persistente a menudo implica el uso de tecnologías de bases de datos específicas. Se puede incluir la creación de clases de diseño para hacer la correspondencia con tablas en un modelo de datos relacional.
- Diseñar clases de control es una tarea delicada. Debido a que encapsulan secuencias, coordinación de otros objetos o algunas veces pura lógica del negocio, es necesario considerar los siguientes aspectos:
 - Aspectos de distribución: si la secuencia necesita ser distribuida y manejada por diferentes nodos de una red, se puede requerir separar las clases del diseño en diferentes nodos para realizar la clase de control.
 - Aspectos de rendimiento: puede que no sea justificable tener clases del diseño separadas para realizar la clase de control. En cambio, la clase de control podría realizarse por las mismas clases de diseño que están realizando algunas clases de interfaz o clases de entidad relacionadas.

- Aspectos de transacción: las clases de control a menudo encapsulan transacciones. Sus correspondientes diseños incorporan cualquier tecnología de manejo de transacción que se esté utilizando.

4.4.6.2 Identificar operaciones

En esta etapa identificamos las operaciones que las clases de diseño van a necesitar y describimos esas operaciones utilizando la sintaxis de los lenguajes de programación.

4.4.6.3 Identificar atributos

En esta etapa, identificamos los atributos requeridos por la clase de diseño y los describimos utilizando la sintaxis del lenguaje de programación.

4.4.6.4 Identificar asociaciones

Los objetos de diseño interactúan unos con otros en los diagramas de secuencia. Estas interacciones a menudo requieren asociaciones entre las clases correspondientes. El número de relaciones entre clases debe ser el mínimo.

Dada la extensión que posee el realizar los pasos anteriores, estos forman parte del Anexo D, junto con la descripción del flujo de sucesos – diseño.

4.4.7 MODELO DE DATOS RELACIONAL

Como hemos visto en el punto 4.4.6.1, una clase de entidad representa información persistente lo que implica el hecho de hacer uso de tecnologías de bases de datos. Se consideran las clases de entidad como parte importante al momento de plasmar en código lo obtenido hasta el *diseño*, pero es necesario el uso de una base de datos para poder almacenar la información que contienen las clases de entidad, lo que

permitirá una posterior inserción de nuevos datos, modificación de los datos ya existentes o la eliminación de los mismos.

Habiendo señalado la importancia del uso de un manejador de base de datos (DBMS), lo que quedaría por hacer es representarla por medio de un modelo de datos relacional. Este modelo de datos relacional contemplará los siguientes puntos:

1. Las clase de entidad encontradas en el *diseño* serán representadas por tablas de acuerdo a:

CLASE DE ENTIDAD DE DISEÑO	TABLA RELACIONAL EN BD
Aplicación	Aplicación
Empresa	Empresa
Informe	Informe
Proyecto	Proyecto

Cuadro 4.5 Relación entre clases de entidad y tablas de Base de Datos

2. Para el caso de la clase entidad Usuario, se representará por medio de la tabla Usuario, la cuál nos servirá para almacenar la información correspondiente a los tipos de usuario que harán uso del sistema, vale decir, Administrador del Sistema, Jefe de Área y Miembro de Área. Para lograr esto, se creará la tabla TipoUsuario con la cuál identificaremos el tipo de usuario que hace uso del sistema. Además, se hará uso de la tabla Área para almacenar e identificar las diferentes áreas que existen dentro de Royal Systems. Entonces, la representación de lo anteriormente explicado está dada por:

CLASE DE ENTIDAD DE DISEÑO	TABLA RELACIONAL EN BD
Usuario	Area
	TipoUsuario
	Usuario

Cuadro 4.6 Relación entre la clase de entidad Usuario y tablas de Base de Datos

3. Se hará uso de las tablas Estado y Situación para identificar el estado actual de una Aplicación, Empresa o Usuario, así como la situación de un Proyecto. Los valores que se almacenarán en dichas tablas están dadas por:

TABLA RELACIONAL EN BD	VALORES DISPONIBLES
Estado	ACTIVO, INACTIVO
Situación	INCOMPLETO, COMPLETO

Cuadro 4.7 Tablas Estado y Situación

Según lo expuesto en los tres puntos, el modelo de datos relacional está representado por el diagrama de la Figura 4.9.

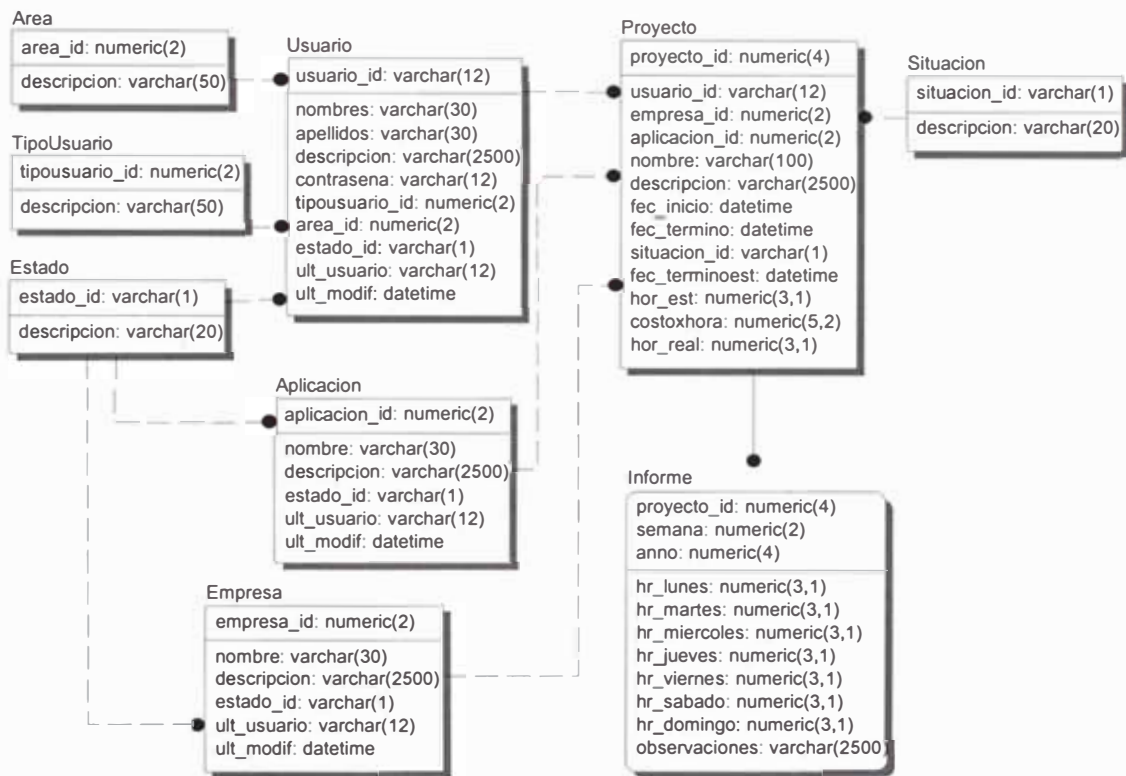


Figura 4.9. Modelo de datos relacional para el Sistema de Control de Proyectos

Del modelo de datos relacional podemos apreciar las tablas que serán usadas en el Sistema de Control de Proyectos, las cuáles formarán en su conjunto la base de datos que usaremos para almacenar de manera persistentes los valores que serán identificados por medio de las clases de entidad del diseño.

En el caso del Sistema de Control de Proyectos, se hará uso del Microsoft SQL Server 2000 por ser la base de datos de mayor uso en Royal Systems y por estar dentro de las especificaciones de la propuesta de solución que se vieron en el punto 4.1. Además, el uso de esta tecnología de base de datos se debe también a la facilidad en el manejo de la implementación y de su administración, lo que nos permitirá centrar nuestros esfuerzos en la implementación del Sistema de Control de Proyecto y no en la implementación de una base de datos en uno de los nodos de la red de Royal Systems. En el ANEXO E se puede encontrar el Diccionario de Datos que nos permitirá entender de manera más clara el objetivo de cada tabla y de sus respectivas columnas.

Hasta aquí lo correspondiente al *diseño*. Se han identificado las principales clases del diseño y los atributos y operaciones pertenecientes a estas clases, haciendo uso de diagramas de secuencia. Estos diagramas de secuencia nos ha servido también para las realizaciones de caso de uso – diseño para el Sistema de Control de Proyectos. El diagrama de clases y la realización de caso de uso forman el Modelo de Diseño junto con el Modelo de Despliegue nos servirán como punto de entrada para la realización del cuarto flujo de trabajo fundamental, la *implementación*, motivo del siguiente subcapítulo.

4.5 IMPLEMENTACIÓN

La *implementación* es el cuarto *Flujo de Trabajo Fundamental* dentro del RUP. En la *implementación* empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, archivos de código fuente, scripts, archivos de código binario, ejecutables y similares.

El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la implementación son:

- Planificar las integraciones del sistema necesarias en cada iteración.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables.

4.5.1 MODELO DE IMPLEMENTACIÓN

El *modelo de implementación* describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. Además, describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación utilizados, y cómo dependen los componentes unos de otros.

4.5.2 COMPONENTE

Un *componente* es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Los componentes tienen las siguientes características:

- Tienen relaciones directa con los elementos de modelo.
- Puede implementar varios elementos, por ejemplo, varias clases.
- Proporcionan las mismas interfaces que los elementos de modelo que implementan.
- Puede haber dependencias de compilación entre componentes.

4.5.3 PLAN DE INTEGRACION DE CONSTRUCCIONES

Es importante construir el software incrementalmente en pasos manejables, de forma que cada paso dé lugar a pequeños problemas de integración o prueba. El resultado de cada paso es llamado "construcción", que es una versión ejecutable del sistema, usualmente una parte específica del sistema. Cada prueba es sometida a pruebas de integración antes de

que se cree ninguna otra construcción. Para prepararse ante el fallo de una construcción se lleva un control de versiones de forma que se pueda volver atrás a una construcción anterior.

Los beneficios de este enfoque incremental son los siguientes:

- Se puede crear una versión ejecutable del sistema muy pronto. Las pruebas de integración comienzan pronto.
- Es más fácil localizar defectos durante las pruebas de integración.
- Las pruebas de integración tienden a ser más minuciosas que las pruebas del sistema completo porque se centran en partes más pequeñas y más manejables.

Un *plan de integración de construcciones* describe la secuencia de construcciones necesarias en una iteración. Más concretamente, un plan de este tipo describe lo siguiente para cada construcción:

- La funcionalidad que se espera que sea implementada en dicha construcción.
- Las partes del modelo de implementación que están afectadas por la construcción.

Luego de haber definido los conceptos anteriores, lo que sigue a continuación es realizar el *Plan de Integración de Construcciones*. Hemos visto desde el comienzo del Capítulo 4 que el Sistema de Control de Proyecto esta conformado por 3 grandes módulos, el módulo del Administrador del Sistema, el módulo del Jefe de Área y el módulo del miembro de Área. Entonces, un primer paso para la definición del *plan de integración de construcciones* es realizar la secuencia de construcciones a partir de la implementación de estos 3 grandes módulos. La construcción de cada módulo comprenderá tanto el *Modelo de Implementación* como también la construcción de los *Componentes* que se usarán dentro del módulo que se desea implementar. Además, a la construcción de estos 3

grandes módulos, se debe agregar la construcción de los componentes base sobre los cuales girará el desarrollo de los otros 3 módulos. Estos componentes base están formados mas que nada por las clases de control y de interfaz que hemos diseñado en el capítulo anterior. Para la implementación del *Sistema de Control de Proyectos*, se hizo uso del Microsoft Visual Studio .Net como entorno de desarrollo integrado (IDE) y del Microsoft C# como lenguaje de desarrollo. Como manejador de base de datos se uso el Microsoft SQL Server 2000.

Entonces, nuestro *Plan de Integración de Construcciones* estuvo dado por los siguientes pasos:

- Construcción de la Base de Datos
- Construcción de los componentes base del sistema (clases de diseño de análisis y control)
- Construcción del Sistema de Control de Proyectos (clases de diseño de interfaz)

Los proyectos que fueron creados en Visual Studio .NET fueron los siguientes:

- SCPLIB (Sistema de Control de Proyectos – Librería de Componentes): proyecto que consistió en la construcción de los componentes base del sistema.
- SCPWeb (Sistema de Control de Proyectos – Aplicativo Web): proyecto que consistió en la construcción de los módulos del Administrador del Sistema, Jefe del Área y Miembro de Área.

4.5.4 CONSTRUCCIÓN DE LA BASE DE DATOS

Cómo mencionamos en el *análisis*, las clases de entidad nos sirven para modelar información que posee una vida larga y que es persistente. Esta información, debe ser almacenada en una base de datos. Hemos plasmado un modelo relacional (figura 4.9) que nos sirvió mucho al momento

de realizar la construcción del sistema. Las tablas que aparecen en el modelo relacional interactúan de manera directa con el componente que implemente su respectiva clase de entidad. La implementación del modelo relacional se realizó en un MS SQL 2000, según se puede apreciar en la figura 4.10. Para lograr esta implementación, creamos una base de datos llamada SCP. Además, la implementación respeta el modelo relacional del cuál ha partido según lo muestra la figura 4.11.

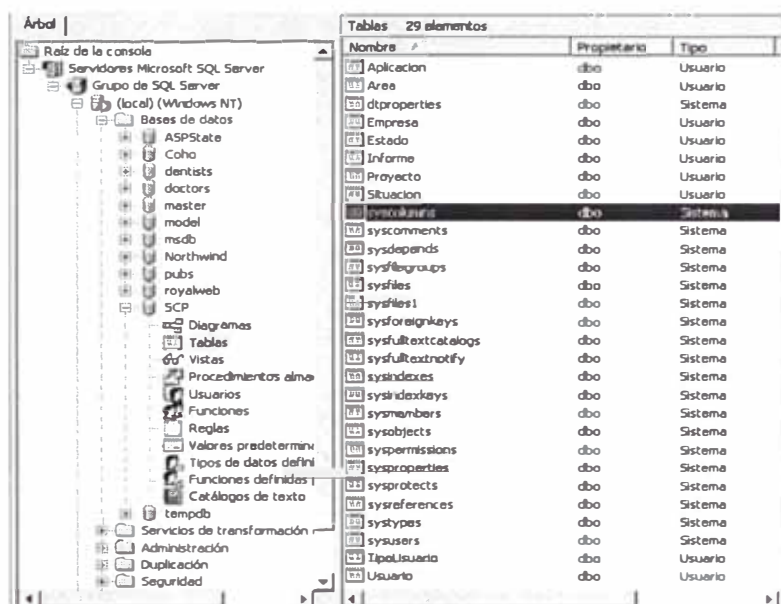


Figura 4.10. Implementación de tablas en el MS SQL 2000

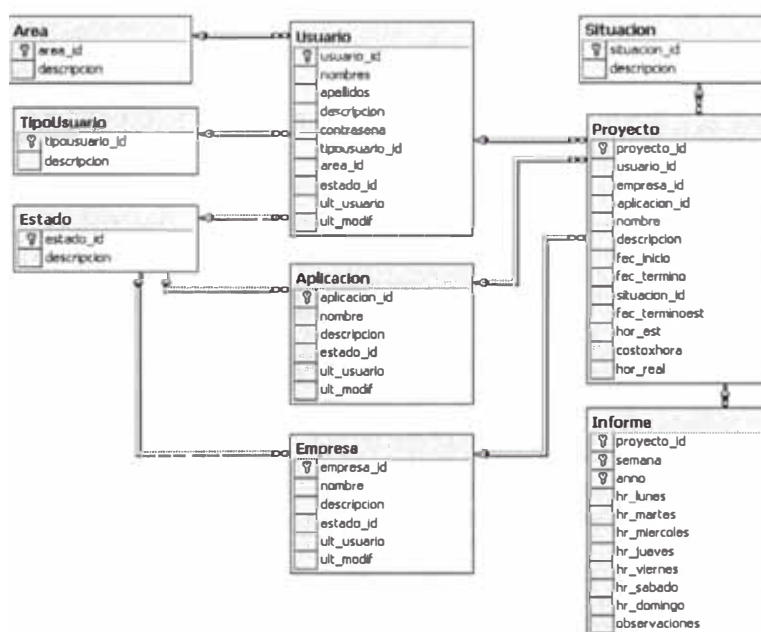


Figura 4.11. Relación entre tablas en MS SQL 2000

4.5.5 CONSTRUCCIÓN DE LOS COMPONENTES BASE DEL SISTEMA

Los componentes base del sistema están conformados por la implementación de las clases de entidad y de control que hemos identificado y definido en el *diseño*. Estos componentes fueron desarrollados en el proyecto *SCPLib* según lo muestra la figura 4.12.



Figura 4.12. Proyecto SCPLib

Los componentes que se construyeron en el proyecto plasman de manera exacta la funcionalidad que definimos para las clases en el *diseño*. El cuadro siguiente muestra la relación entre las clases de diseño y los componente construidos.

CLASES DE DISEÑO		COMPONENTE EN SCPLIB
Control	Gestor Aplicación	Gaplicacion
	Gestor Empresa	Gempresa
	Gestor Informe	Ginforme
	Gestor Proyecto	Gproyecto
	Gestor Usuario	Gusuario
Entidad	Aplicación	Aplicación
	Empresa	Empresa
	Informe	Informe
	Proyecto	Proyecto

CLASES DE DISEÑO	COMPONENTE EN SCPLIB
Usuario	Usuario

Cuadro 4.8 Relación entre clases de diseño y componentes construidos

Un ejemplo de lo que decimos es el que mostramos a continuación. La figura 4.13 nos muestra la clase de diseño **Gestor Usuario**, que es una clase de control. En la figura se aprecia las operaciones que forman parte de esta clase (recuérdese una clase de control esta formada principalmente por operaciones).

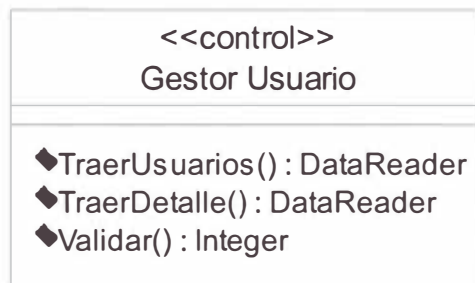


Figura 4.13. Clase de control Gestor Usuario

A continuación, en la figura 4.14, mostramos el código del componente **GUusuario**, el cuál implementa las operaciones que hemos mostrado en la figura 4.13.

```

Object Browser | Start Page | GUusuario.cs*
SCPLib.GUusuario
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace SCPLib
{
    public class GUusuario(
        private SqlConnection cn = new SqlConnection(ConfigurationSettings.AppSettings["cn"]);

        public int Validar(string Usuario, string Contraseña)...

        public DataView TraerUsuarios(string Usuario, string Nombres, string Apellidos,
            int TipoUsuario, int Area, int Estado)...

        public Usuario TraerDetalle(string Usuario)...
    }
}
  
```

Figura 4.14. Código del componente GUusuario

Como se puede ver, el componente construido contiene exactamente las mismas operaciones que plasmamos en el *diseño*. Obviamente, un componente implementado puede contener otros atributos y operaciones que no necesariamente están definidos en la clase *diseño*, pero que a la hora de implementarlo, nos van a facilitar el trabajo de sobremanera. Un mayor detalle del código que se ha escrito para el proyecto SCPLib se puede encontrar en el CD que acompaña la tesis. Para leerlo basta con abrir los archivos con extensión *cs* del proyecto con el *Bloque de Notas* de Windows.

4.5.6 CONSTRUCCION DEL SISTEMA DE CONTROL DE PROYECTOS

El proyecto *SCPWeb* esta formado completamente por los componente que implementan las clases de interfaz del *diseño*. Estos componentes fueron desarrollados según lo muestra la figura 4.15.

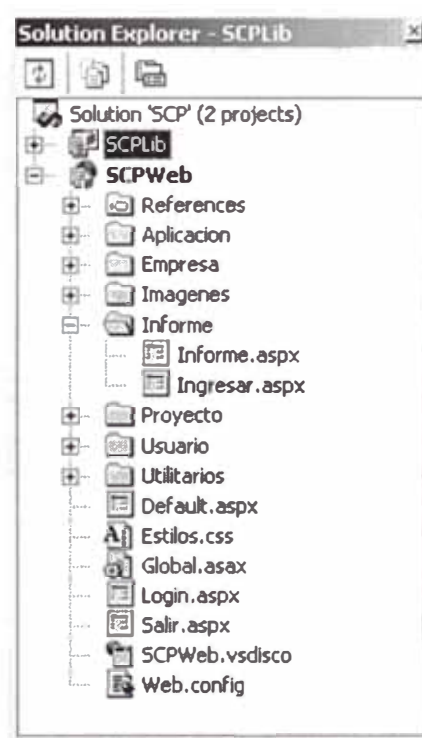


Figura 4.15. Proyecto SCPWeb

Como se puede apreciar, para la construcción de los componentes, hicimos uso de *fólderes* por dos motivos importantes:

- Ordenar y clasificar los componentes según el módulo de trabajo al cuál iban a pertenecer
- Trazar un identificador único por el cual identificamos a cada componente de manera única

El cuadro que viene a continuación muestra la relación entre las clases de interfaz y los componente construidos en *SCPWeb*.

CLASE DE INTERFAZ	COMPONENTE EN SCPWEB
Página Inicial	/Login.aspx
Página de Bienvenida	/Default.aspx
Crear Usuario	/Usuario/Crear.aspx
Consultar Usuario	/Usuario/Consultar.aspx
Modificar Usuario	/Usuario/Modificar.aspx
Mostrar Usuario	/Usuario/Mostrar.aspx
Crear Empresa	/Empresa/Crear.aspx
Consultar Empresa	/Empresa/Consultar.aspx
Modificar Empresa	/Empresa/Modificar.aspx
Mostrar Empresa	/Empresa/Mostrar.aspx
Crear Aplicación	/Aplicación/Crear.aspx
Consultar Aplicación	/Aplicación/Consultar.aspx
Modificar Aplicación	/Aplicación/Modificar.aspx
Mostrar Aplicación	/Aplicación/Mostrar.aspx
Crear Proyecto	/Proyecto/Crear.aspx
Consultar Proyecto	/Proyecto/Consultar.aspx
Modificar Proyecto	/Proyecto/Modificar.aspx
Mostrar Proyecto	/Proyecto/Mostrar.aspx
Ingresar Informe	/Informe/Ingresar.aspx

Cuadro 4.9 Relación entre clases de interfaz y componentes construidos

4.5.6.1 Módulo de Ingreso, Bienvenida y Salida del SCPWeb

Este módulo consiste en las páginas que se nos presentan para ingresar al sistema, cuando estamos dentro del sistema y cuando deseamos salir del sistema.

- **Página de Ingreso.** Esta página se presenta cada vez que cualquier usuario del sistema desea ingresar al mismo. Lo más importante de esta página es que se presenta cada vez que el usuario quiera acceder a una página del sistema sin haber presentado sus credenciales. La figura 4.16 nos presenta esta página.



Figura 4.16. Página de Ingreso

- **Página de Bienvenida.** Esta página se presenta siempre que el usuario ingresa al sistema (figura 4.17). La página nos da una pequeña explicación de cómo realizar una acción determinada. Dentro de esta pequeña explicación, el sistema reconoce que tipo de usuario ha ingresado (ADMINISTRADOR, JEFE DE AREA, MIEMBRO DE AREA). Además, en la parte superior de la página siempre se nos presenta una fila que consiste en el nombre de usuario que ha accedido al sistema, así como también una serie de enlaces referidos a las acciones que podemos realizar (casos de uso). Siempre se pinta con color naranja la acción que se está realizando en el momento. En el caso de la figura 4.17, las

acciones que se muestran están referidas a las de un Administrador del Sistema.

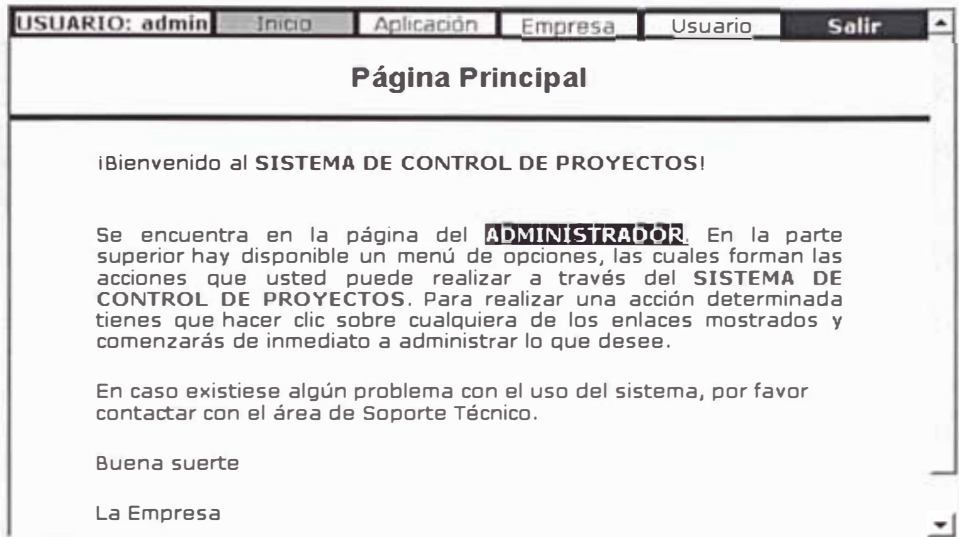


Figura 4.17. Página de Bienvenida con las acciones que puede realizar el usuario

- **Página de Salida.** Una vez que el usuario realiza las acciones que desea, procede a acceder clic sobre el enlace "Salir" que se encuentra en la parte superior de la pantalla. A continuación se le muestra la página de la figura 4.18. Para salir del sistema, basta con hacer clic sobre el botón "SALIR". Lo importante de esta página es que al momento de ser procesada por acción del usuario, se procede a liberar todos los recursos que se han destinado al usuario al momento de ingresar al sistema.

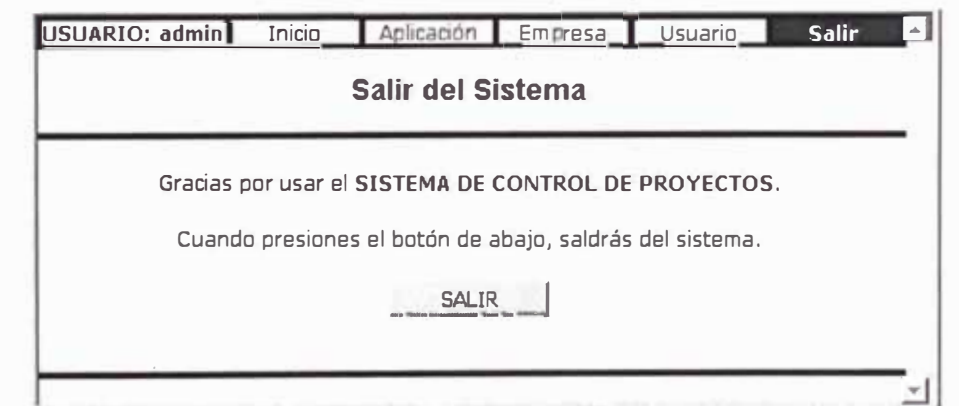


Figura 4.18. Página de Salida

4.5.6.2 Módulo del Administrador del Sistema

Este módulo consiste en las páginas que se presentan al usuario que ha accedido al sistema como Administrador. Las acciones que se pueden

realizar en este módulo están referidas a la creación, consulta y modificación de Aplicaciones, Empresas y Usuarios.

- **Aplicación.** Cuando se desee realizar acciones sobre las aplicaciones del sistema, solo basta con hacer clic sobre el enlace "Aplicación". Esto nos llevará a la página de la figura 4.19.

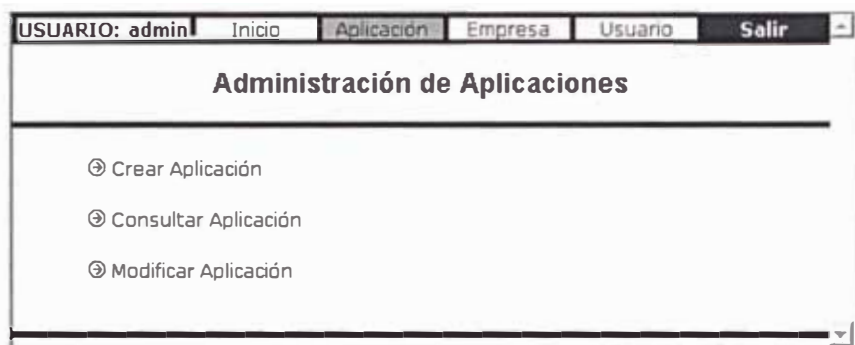


Figura 4.19. Página de acciones sobre las Aplicaciones del Sistema

Desde esta página se podrán realizar las acciones especificadas en la misma. Estas acciones están dadas por Crear Aplicación (figura 4.20), Consultar Aplicación y Modificar Aplicación.

Figura 4.20. Crear Aplicación

En el caso de Consultar y Modificar Aplicación, se mostrará siempre la figura 4.21. En esta página se ingresarán los criterios por los cuales se va a buscar una determinada Aplicación.

Figura 4.21. Página de ingreso de criterios de búsqueda - Aplicación

El resultado de la búsqueda según los criterios ingresados se mostrará en otra página según se puede apreciar en la figura 4.22.

Aplicación	Estado
ABC Costing	ACTIVO
Activo Fijo	INACTIVO
Administración	ACTIVO
Capacitación	INACTIVO
Central FILE / IDM	ACTIVO
Comercial	INACTIVO
Conciliación Bancaria	INACTIVO
Contabilidad	INACTIVO
Control de Asistencia	ACTIVO
Costo Naviero	ACTIVO

1 2 3 4
Otra búsqueda...

Figura 4.22. Resultados de la búsqueda - Aplicación

En la figura 4.22 podemos ver que el resultado de la búsqueda puede traer varias coincidencias. Por cuestiones de orden, hemos creído conveniente en mostrar sólo 10 coincidencias a la vez. Si se desea buscar una coincidencia en particular, bastará con hacer clic sobre el número de página a la cuál deseamos ir. Los números de página se muestran debajo de la décima coincidencia.

Si la acción que realizamos fue Consultar Aplicación, se mostrará la página de la figura 4.23 mientras que si fue Modificar Aplicación, se mostrará la página de la figura 4.24.

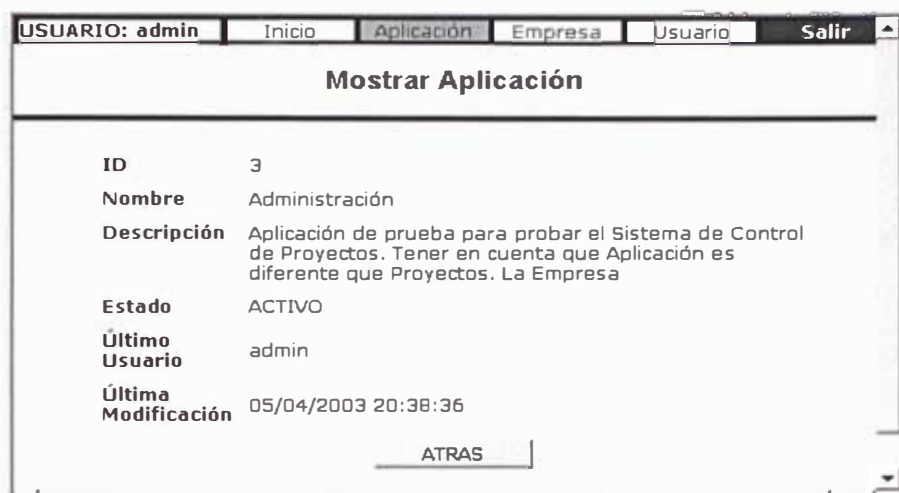


Figura 4.23. Consultar Aplicación

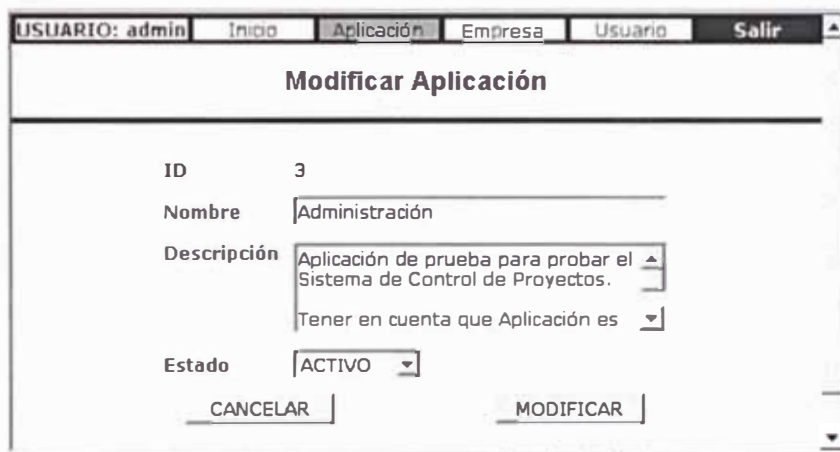


Figura 4.24. Modificar Aplicación

- **Empresa.** Cuando se desee realizar acciones sobre las empresas del sistema, solo basta con hacer clic sobre el enlace "Empresa". Esto nos llevará a la página de la figura 4.25.

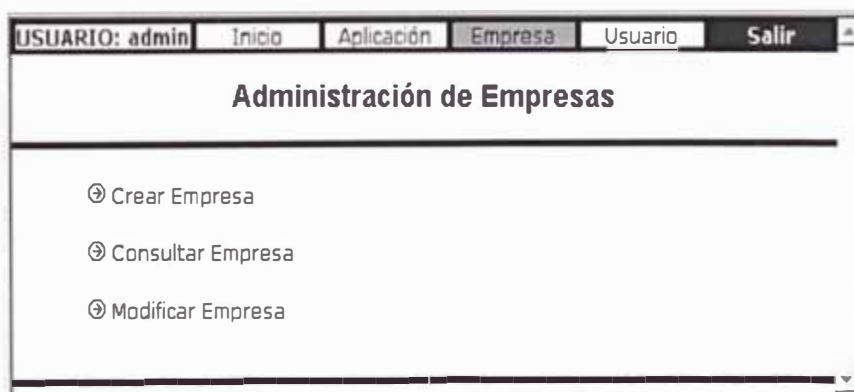


Figura 4.25. Página de acciones sobre las Empresas del Sistema

Desde esta página se podrán realizar las acciones especificadas en la misma. Esta acciones están dadas por Crear Empresa (figura 4.26), Consultar Empresa y Modificar Empresa.

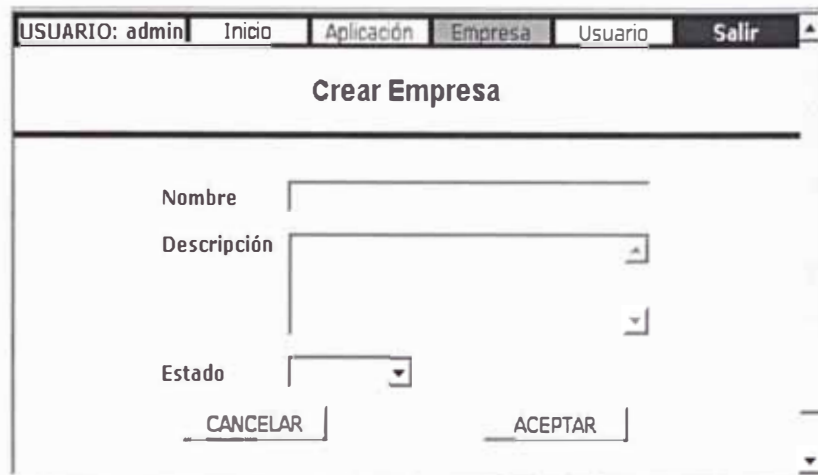


Figura 4.26. Crear Empresa

En el caso de Consultar y Modificar Empresa, se mostrará siempre la figura 4.27. En esta página se ingresarán los criterios por los cuales se va a buscar una determinada Empresa.

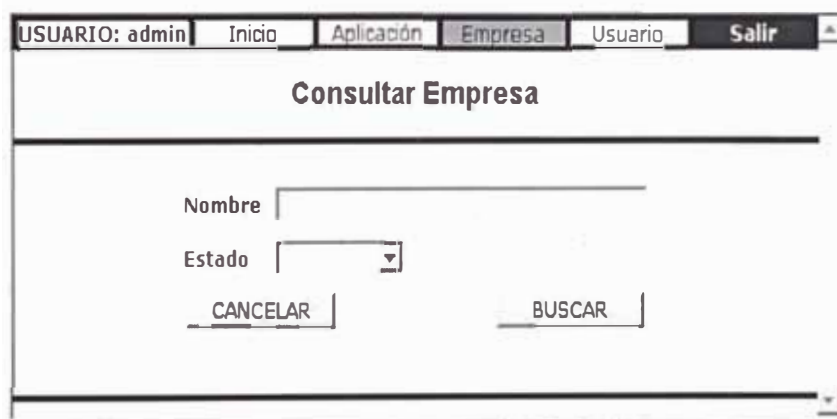


Figura 4.27. Página de ingreso de criterios de búsqueda - Empresa

El resultado de la búsqueda según los criterios ingresados se mostrará en otra página según se puede apreciar en la figura 4.28.

USUARIO: admin		Inicio	Aplicación	Empresa	Usuario	Salir
Consultar Empresa						
Empresa		Estado				
AFP Unión		INACTIVO				
Agencia Naviera MAYNAS S.A.		ACTIVO				
Ajinomoto del Perú		INACTIVO				
América Televisión		INACTIVO				
Asociación Benéfica Prisma		ACTIVO				
Banco Nuevo Mundo		INACTIVO				
Banco Wiese Sudameris		ACTIVO				
British Gas Bolivia		INACTIVO				
BSH Electrodomesticos		ACTIVO				
Cementos Lima S.A.		ACTIVO				
1 2 3						
Otra búsqueda...						

Figura 4.28. Resultados de la búsqueda - Empresa

Si la acción que realizamos fue Consultar Empresa, se mostrará la página de la figura 4.29 mientras que si fue Modificar Empresa, se mostrará la página de la figura 4.30.

USUARIO: admin		Inicio	Aplicación	Empresa	Usuario	Salir
Mostrar Empresa						
ID	3					
Nombre	Agencia Naviera MAYNAS S.A.					
Descripción	Empresa de prueba para realizar acciones sobre el sistema. Tener en cuenta el estado al momento de la consulta. Gracias					
Estado	ACTIVO					
Último Usuario	admin					
Última Modificación	04/04/2003 19:31:23					
<input type="button" value="ATRÁS"/>						

Figura 4.29. Consultar Empresa

USUARIO: admin Inicio Aplicación Empresa Usuario Salir

Modificar Empresa

ID 3

Nombre Agencia Naviera MAYNAS S.A.

Descripción Empresa de prueba para realizar acciones sobre el sistema.
Tener en cuenta el estado al

Estado ACTIVO

CANCELAR MODIFICAR

Figura 4.30. Modificar Empresa

- **Usuario.** Cuando se desee realizar acciones sobre los usuarios del sistema, solo basta con hacer clic sobre el enlace "Usuario". Esto nos llevará a la página de la figura 4.31.

USUARIO: admin Inicio Aplicación Empresa Usuario Salir

Administración de Usuarios

➤ Crear Usuario

➤ Consultar Usuario

➤ Modificar Usuario

Figura 4.31. Página de acciones sobre los Usuarios del Sistema

Desde esta página se podrán realizar las acciones especificadas en la misma. Estas acciones están dadas por Crear Usuario (figura 4.32), Consultar Usuario y Modificar Usuario.

Figura 4.32. Crear Usuario

En el caso de Consultar y Modificar Usuario, se mostrará siempre la figura 4.33. En esta página se ingresarán los criterios por los cuales se va a buscar un determinado Usuario.

Figura 4.33. Página de ingreso de criterios de búsqueda - Usuario

El resultado de la búsqueda según los criterios ingresados se mostrará en otra página según se puede apreciar en la figura 4.34.

USUARIO: admin Inicio Aplicación Empresa Usuario Salir					
Consultar Usuario					
Usuario	Apellidos	Nombres	Tipo Usuario	Area	Estado
candrade	Andrade	Cesar	MIEMBRO	IMPLEMENTACION	ACTIVO
jandrade	Andrade	Jhaneth	MIEMBRO	ADMINISTRACION	ACTIVO
arbulum	Arbulu	Manuel	MIEMBRO	NUEVAS TECNOLOGIAS	ACTIVO
ebenitez	Benitez	Ericka	MIEMBRO	ADMINISTRACION	INACTIVO
lcaceres	Cáceres	Luis	MIEMBRO	NUEVAS TECNOLOGIAS	INACTIVO
mcaceres	Cáceres	Miguel	MIEMBRO	COSTOS ABC	ACTIVO
jcarrasco	Carrasco	John	JEFE	SISTEMAS TECNICOS	ACTIVO
changh	Chang	Hui Sang	MIEMBRO	COMERCIAL	INACTIVO
ocordova	Cordova	Omar	JEFE	COSTOS ABC	ACTIVO
ecortez	Cortez	Elena	MIEMBRO	ADMINISTRACION	ACTIVO
1 2 3					
Otra búsqueda...					

Figura 4.34. Resultados de la búsqueda - Usuario

Si la acción que realizamos fue Consultar Usuario, se mostrará la página de la figura 4.35 mientras que si fue Modificar Usuario, se mostrará la página de la figura 4.36.

USUARIO: admin Inicio Aplicación Empresa Usuario Salir	
Mostrar Usuario	
Usuario	arbulum
Nombres	Manuel
Apellidos	Arbulu
Descripción	usuario de prueba usuario de prueba usuario de prueba usuario de prueba usuario de prueba usuario de prueba usuario de prueba usuario de prueba
Contraseña	arbulum
Tipo de Usuario	MIEMBRO
Area	NUEVAS TECNOLOGIAS
Estado	ACTIVO
Último Usuario	admin
Última Modificación	06/04/2003 14:55:10
<input type="button" value="ATRAS"/>	

Figura 4.35. Consultar Usuario

Figura 4.36. Modificar Usuario

4.5.6.3 Módulo del Jefe de Área

Este módulo consiste en las páginas que se presentan al usuario que ha accedido al sistema como Jefe de Área. Las acciones que se pueden realizar en este módulo están referidas a la creación, consulta y modificación de Proyectos así como también el ingreso del Informe Semanal y Reportes especializados. En este punto solo vamos a explicar lo concerniente a la manipulación de Proyectos y Reportes. Lo referente al ingreso del Informe Semanal se verá con mayor detenimiento en la siguiente sección concerniente al *Módulo del Miembro de Área*.

- **Proyecto.** Cuando se desee realizar acciones sobre los proyectos del sistema, solo basta con hacer clic sobre el enlace "Proyecto". Esto nos llevará a la página de la figura 4.37

Figura 4.37. Página de acciones sobre los Proyectos del Sistema

Desde esta página se podrán realizar las acciones especificadas en la misma. Estas acciones están dadas por Crear Proyecto (figura 4.38), Consultar Proyecto y Modificar Proyecto.

USUARIO: vreyña Inicio Informe Semanal Proyecto Salir

Crear Proyecto

Nombre

Descripción

Usuario

Empresa

Aplicación

Inicio

Término Estimado

Horas Estimadas

Costo x Hora (NS/.)

CANCELAR ACEPTAR

Figura 4.38. Crear Proyecto

En el caso de Consultar y Modificar Proyecto, se mostrará siempre la figura 4.39. En esta página se ingresarán los criterios por los cuales se va a buscar un determinado Usuario.

USUARIO: vreyña Inicio Informe Semanal Proyecto Salir

Consultar Proyecto

Nombre

Usuario

Empresa

Aplicación

Inicio Entre el y el

Término Entre el y el

Situación

CANCELAR BUSCAR

Figura 4.39. Página de ingreso de criterios de búsqueda - Proyecto

El resultado de la búsqueda según los criterios ingresados se mostrará en otra página según se puede apreciar en la figura 4.40.

Proyecto	Situación
Actualización de Módulos	INCOMPLETO
Actualización Logística	INCOMPLETO
Actualización de Personal	INCOMPLETO
Envío online de datos	INCOMPLETO
Incremento de Inventarios	INCOMPLETO
Incremento de Inventarios	INCOMPLETO
Nuevas naves	INCOMPLETO
Planillas Version 2.2	INCOMPLETO
Primer Proyecto de Prueba	INCOMPLETO
Proyecto para probar valores	INCOMPLETO

1 2

Otra búsqueda...

Figura 4.40. Resultados de la búsqueda - Proyecto

Si la acción que realizamos fue Consultar Proyecto, se mostrará la página de la figura 4.41 mientras que si fue Modificar Proyecto, se mostrará la página de la figura 4.42.

Mostrar Proyecto

ID: 1
 Nombre: Primer proyecto de prueba
 Descripción: Proyecto en el que se manejan las horas que trabaja cada trabajador
 Usuario: Reyna Vargas, Víctor José Manuel Inicio: 23/07/2003
 Empresa: Agencia Naviera MAYNAS S.A. Estimado Real
 Aplicación: Central FILE / IDM Término: 28/08/2003
 Situación: INCOMPLETO Horas: 150.36 87.5
 Costo x Hora (NS/.): 12.56 Costo Total: 1888.52 1099

ATRÁS

Informes Ingresados

Semana	Lu	Ma	Mi	Ju	Vi	Sa	Do	Observaciones
21/07/2003 al 27/07/2003	9.8	8.7	7.6	6.5	5.4	4.3	3.2	ninguna observación importante

Figura 4.41. Consultar Proyecto

USUARIO: vrey na		Inicio	Informe Semanal	Proyecto	Salir
Modificar Proyecto					
ID	1				
Nombre	Primer proyecto de prueba				
Descripción	Proyecto en el que se manejan las horas que trabaja cada trabajador				
Usuario	Reyna Vargas, Víctor José Manuel				
Empresa	Agencia Naviera MAYNAS S.A.				
Aplicación	Central FILE / IDM				
Inicio	23/07/2003 ...				
Término Estimado	28/08/2003 ...				
Término Real	...				
Horas Estimadas	150.36				
Horas Reales	87.5				
Costo x Hora (NS/.)	12.56				
Situación	INCOMPLETO				
CANCELAR		MODIFICAR			

Figura 4.42. Modificar Proyecto

- **Reportes.** Cuando se desee realizar acciones sobre los reportes del sistema, solo basta con hacer clic sobre el enlace "Reportes". Esto nos llevará a la página de la figura 4.43.

USUARIO: vrey na		Inicio	Informe Semanal	Proyecto	Reportes	Salir
Administración de Proyectos						
<ul style="list-style-type: none"> ⊕ Control Planeado ⊕ Control Planeado/Ejecutado ⊕ Uso de Recursos 						

Figura 4.43. Página de Reportes del Sistema

Desde esta página se podrán consultar los reportes disponibles en el Sistema. Si hacemos clic sobre Control Planeado (Figura 4.44), tendremos una lista de actividades que un usuario de área desarrollará en una semana; Control Planeado / Ejecutado (Figura 4.45) nos muestra lo mismo que

Control Planeado además de un resumen de lo que realmente ocurrió en la semana de trabajo y finalmente, Uso de Recursos (Figura 4.46) nos muestra un reporte de lo hecho por un usuario en un período de tiempo.

Control Planeado												
Usuario	Empresa	Aplicación	H.Pl.	H.Ac.	% Av.	Lu	Ma	Mi	Ju	Vi	H.Sem.	% Av.Esp.
Omar Cordova	AFP Unión	ABC Costing	90	51	56.67	3	0	0	8	2	13	71.11
Cesar Andrade	América Televisión	ABC Costing	150	25	16.67	0	5	0	4	6	15	26.67
	AFP Unión	Capacitación	50	15	30	5	0	5	0	5	15	60
	Hospital Militar Central	Contabilidad	120	152	126.67	2	2	2	2	2	10	135
Elena Cortez	América Televisión	Facturación	70	46	65.71	5	5	0	8	8	26	108.57
Hector Montero	AFP Unión	Inventarios	60	14	23.33	3	3	0	4	0	10	40
	ENAPU S.A.	Capacitación	40	57	142.5	3	0	3	3	0	9	165

H. Pl.: Horas Totales Planeadas, H. Ac.: Horas Acumuladas, % Av.: Porcentaje Avance, H. Sem.: Horas Semanales, % Av. Esp.: Porcentaje de Avance Esperado

Figura 4.44. Control Planeado

Control Planeado/Ejecutado															
Usuario	Empresa	Aplicación	H.Pl.	H.Ac.	% Av.	Lu	Ma	Mi	Ju	Vi	H.Sem.	% Av.Esp.	H. Real	% Av. Real	Dif. %
Omar Cordova	AFP Unión	ABC Costing	90	51	56.67	3	0	0	8	2	13	71.11	10	67.78	-3.33
Cesar Andrade	América Televisión	ABC Costing	150	25	16.67	0	5	0	4	6	15	26.67	12	24.67	-2
	AFP Unión	Capacitación	50	15	30	5	0	5	0	5	15	60	12	54	-6
	Hospital Militar Central	Contabilidad	120	152	126.67	2	2	2	2	2	10	135	15	139.17	4.17
Elena Cortez	América Televisión	Facturación	70	46	65.71	5	5	0	8	8	26	108.57	20	94.29	14.28
Hector Montero	AFP Unión	Inventarios	60	14	23.33	3	3	0	4	0	10	40	15	48.33	8.33
	ENAPU S.A.	Capacitación	40	57	142.5	3	0	3	3	0	9	165	15	180	15

H. Pl.: Horas Totales Planeadas, H. Ac.: Horas Acumuladas, % Av.: Porcentaje Avance, H. Sem.: Horas Semanales, % Av. Esp.: Porcentaje de Avance Esperado

H. Real: Horas Reales Semanales, % Av. Real: Porcentaje de Avance Real, Dif %: Diferencia de % (Real - Acumulado)

Figura 4.45. Control Planeado / Ejecutado

Uso de Recursos						
Usuario	Empresa	Aplicación	Horas	Costo x Hora (NS/.)	Horas Totales	Costo Total (NS/.)
Omar Cordova	AFP Unión	ABC Costing	61	12.56	61	766.16
	América Televisión	ABC Costing	37	12.56		
Cesar Andrade	AFP Unión	Capacitación	27	15.00	231	3842.32
	Hospital Militar Central	Contabilidad	167	17.80		
Elena Cortez	América Televisión	Facturación	66	13.69	66	903.54
Hector Montero	AFP Unión	Inventarios	29	13.58	101	1473.82
	ENAPU S.A.	Capacitación	72	15.00		

Figura 4.46. Uso de Recursos

4.5.6.4 Módulo del Miembro de Área

Este módulo consiste en las páginas que se presentan al usuario que ha accedido al sistema como Miembro de Área. Las acciones que se pueden realizar en este módulo sólo se limitan al ingreso del Informe Semanal.

- **Informe Semanal.** Cuando se desee ingresar informes semanales asociados a los proyectos del sistema, solo basta con hacer clic sobre el enlace "Informe Semanal". Esto nos llevará a la página de la figura 4.47.

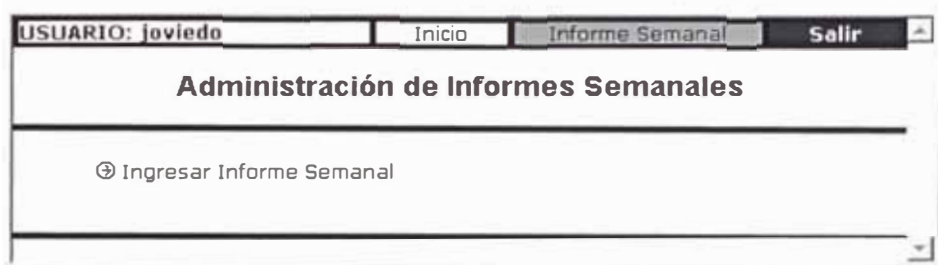


Figura 4.47. Página de acciones sobre Informes Semanales

Desde esta página se podrán realizar el ingreso de los informes semanales. Solo tendremos que hacer clic sobre el enlace "Ingresar Informe Semanal", mostrándonos la figura 4.48. Como se puede apreciar, antes de ingresar el informe debemos buscar el proyecto al cual va a estar asociado el informe. Es por esta razón que la página que se nos muestra es aquella que vimos en la sección anterior relacionada a la administración de proyecto. Además, debemos tener presente que un *Miembro de Área* sólo podrá ingresar informes semanales en aquellos proyectos que le estén asignados y que se encuentren en la situación de INCOMPLETO.

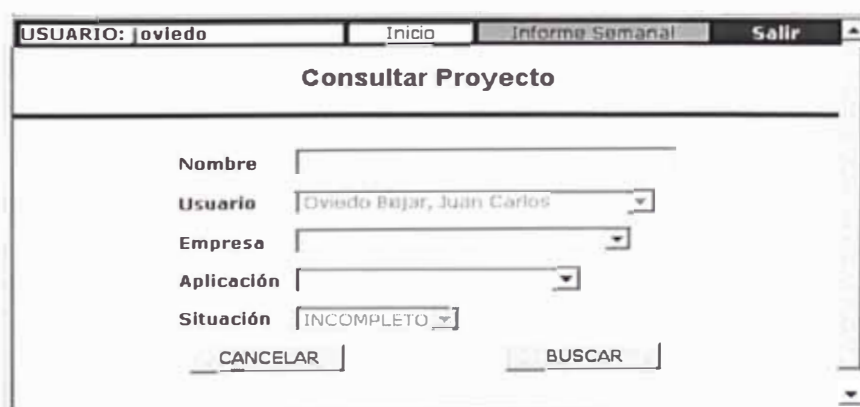
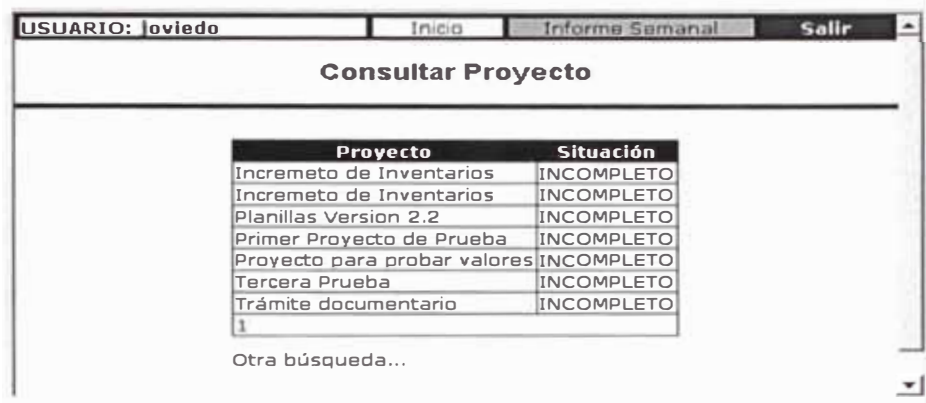


Figura 4.48. Consultar Proyecto

En esta página se ingresarán los criterios por los cuales se va a buscar un determinado Proyecto. Una vez hecho lo anterior, se mostrarán los resultados de la búsqueda como lo vemos en la figura 4.49.



Proyecto	Situación
Incremento de Inventarios	INCOMPLETO
Incremento de Inventarios	INCOMPLETO
Planillas Version 2.2	INCOMPLETO
Primer Proyecto de Prueba	INCOMPLETO
Proyecto para probar valores	INCOMPLETO
Tercera Prueba	INCOMPLETO
Trámite documentario	INCOMPLETO

Otra búsqueda...

Figura 4.49. Resultados de la búsqueda – Informe Semanal

Hacemos clic sobre el proyecto en el cual queremos ingresar un informe semanal, como vemos en la figura 4.50. Como podemos ver, en la página se muestran los datos generales asociados al proyecto. Por otro lado, se nos muestra un reporte de los informes semanales ingresados anteriormente, ordenados desde el mas antiguo hasta el último ingresado. Finalmente se muestra unos casilleros donde vamos a ingresar las horas que se han empleado en desarrollar el proyecto en la presente semana así como alguna observaciones que se deseen hacer.

USUARIO: loviedo Inicio Informe Semanal Salir

Ingresar Informe Semanal

Datos del Proyecto

Nombre: Planillas Version 2.2
 Descripción: Actualización de las planillas de la empresa.
 Usuario: Oviedo Béjar, Juan Carlos Inicio: 10/04/2003
 Empresa: Cementos Lima S.A. Término:
 Aplicación: Planillas Situación: INCOMPLETO

Informes Anteriores

Semana	Lu	Ma	Mi	Ju	Vi	Sa	Do	Observaciones
07/04/2003 al 13/04/2003	5	6	7	8	9	0	5	Primer Informe
14/04/2003 al 20/04/2003	9	8	7	6	3.6	5	0	Segundo Informe semanal Víctor Reyna Vargas
21/04/2003 al 27/04/2003	9	8	7	6	5	4	3	1

Informe Semana Actual

Ingresar en la parte inferior los valores correspondientes a las horas avanzadas en el proyecto actual.

Lu	Ma	Mi	Ju	Vi	Sa	Do	Observaciones

CANCELAR ACEPTAR

Figura 4.50. Ingreso del Informe Semanal

Hasta aquí lo correspondiente a la *implementación*. Se ha mostrado la construcción de los componentes, que han tenido como base para su desarrollo las clases que hemos identificados en el *diseño*. Hemos mostrado los proyectos que se han construido para la implementación de las clases de entidad, control y de interfaz. Por último, hemos mostrado mediante figuras la apariencia que tienen estas páginas que en conjunto forman el *Sistema de Control de Proyectos*. Lo relacionado a las pruebas de rigor serán el motivo del siguiente subcapítulo.

4.6 PRUEBA

La *prueba* es el quinto *Flujo de Trabajo Fundamental* dentro del RUP. En la *prueba* verificamos el resultado de la *implementación* probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros.

En concreto, los objetivos de la prueba son:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y las pruebas de sistema.

- Diseñar e implementar las pruebas creando los casos de prueba que especifican qué probar.
- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente.

Durante la fase de inicio, se puede hacer parte de la planificación inicial de las pruebas cuando se define al ámbito del sistema. Sin embargo, las pruebas se llevan a cabo sobre todo cuando una construcción (como un resultado de implementación) es sometida a pruebas de integración y de sistema.

Por otro lado, debido a la naturaleza iterativa del desarrollo, algunos de los casos de prueba pueden ser utilizados varias veces. Por lo tanto, es natural mantener el modelo de pruebas a lo largo del ciclo de vida del software completo, aunque el modelo de pruebas cambia constantemente debido a:

- La eliminación de casos de prueba obsoletos
- El refinamiento de algunos casos de prueba
- La creación de nuevos casos de uso para cada nueva construcción

4.6.1 MODELO DE PRUEBAS

El *modelo de pruebas* describe principalmente cómo se prueban los componentes ejecutables (construcciones) del modelo de implementación con pruebas de integración y de sistema. Este *modelo de pruebas* es a su vez, una colección de casos de prueba, procedimientos de prueba y componentes de prueba.

4.6.2 CASO DE PRUEBA

Un *caso de prueba* especifica una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las

condiciones bajo las que ha de probarse. Los siguientes, son casos de prueba comunes:

- Un caso de prueba que especifica cómo probar un caso de uso o un escenario específico de un caso de uso. Obsérvese que un caso de prueba basado en un caso de uso especifica típicamente una prueba del sistema como “caja negra”, es decir, una prueba del comportamiento observable externamente del sistema.
- Un caso de prueba que especifica cómo probar una realización de caso de uso-diseño o un escenario específico de la realización. Obsérvese que los casos de prueba basados en una realización de caso de uso típicamente especifican una prueba del sistema como “caja blanca”, es decir, una prueba de la interacción interna entre los componentes del sistema.

Muchas veces, los casos de prueba pueden ser parecidos y diferenciarse únicamente en un solo valor de entrada o resultado. Esto se da a veces para aquellos casos de prueba que verifican diferentes escenarios del mismo caso de uso. En estos casos, puede ser apropiado especificar los casos de uso en forma de una tabla, donde cada caso de prueba está representado por una fila y cada rango de valores de entrada y resultado se representa por una columna.

Se pueden especificar otros casos de prueba, los cuales nos servirán para probar el sistema como un todo. Por ejemplo:

- Las *pruebas de instalación* verifican que el sistema puede ser instalado en la plataforma del cliente y que el sistema funcionará correctamente cuando sea instalado.
- Las *pruebas de configuración* verifican que el sistema funciona correctamente en diferentes configuraciones.

- Las *pruebas negativas* intentan provocar que el sistema falle para poder así revelar sus debilidades. También se le conocen como “casos de abuso”.
- Las *pruebas de tensión o estrés* identifican problemas con el sistema cuando hay recursos insuficientes o cuando hay competencia por los recursos.

4.6.3 PROCEDIMIENTO DE PRUEBA

Un *procedimiento de prueba* especifica cómo realizar uno a varios casos de prueba o partes de éstos, pero es a menudo útil reutilizar un *procedimiento de prueba* para varios casos de prueba y reutilizar varios procedimientos de prueba para un caso de prueba. Es decir, la relación entre el procedimiento de prueba y el caso de prueba es del tipo muchos a muchos. Otra característica de un *procedimiento de prueba* es la similitud que posee con la *descripción de flujo de eventos* de un caso de uso.

4.6.4 PLANIFICAR PRUEBA

El propósito de la planificación de la prueba es planificar los esfuerzos de prueba en una iteración llevando a cabo las siguientes tareas:

- Describiendo una estrategia de pruebas.
- Estimando los requisitos para el esfuerzo de la prueba, por ejemplo, los recursos humanos y sistemas necesarios.
- Planificando el esfuerzo de la prueba.

Cuando se prepara el plan de pruebas, los ingenieros se mueven sobre un rango de valores de entrada. Ejemplo de estos valores de entrada viene a ser el modelo de casos de uso y los requisitos adicionales. Además, también desarrollan una estrategia de prueba para la iteración, es decir, qué tipo de pruebas ejecutar, cómo ejecutar dichas pruebas, cuándo ejecutarlas y cómo determinar si el esfuerzo de prueba tiene éxito.

4.6.5 DISEÑAR PRUEBA

Los propósitos de diseñar las pruebas son:

- Identificar y describir los casos de pruebas para cada construcción.
- Identificar y estructurar los procedimientos de prueba especificando cómo realizar los casos de prueba.

4.6.5.1 Diseño de los casos de prueba de integración

Los *casos de prueba de integración* se utilizan para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción. La mayoría de los *casos de prueba de integración* pueden ser derivados de las realizaciones de casos de uso – diseño, ya que en ellas se describen las interacciones de las clases y objetos, es decir, los componentes.

4.6.5.2 Diseño de los casos de prueba de sistema

Las *pruebas de sistema* se usan para probar que el sistema funciona correctamente como un todo. Cada *prueba de sistema* prueba principalmente combinaciones de casos de uso instanciados bajo condiciones diferentes. Estas condiciones incluyen diferentes configuraciones hardware, niveles de carga del sistema, número de actores y tamaño de base de datos.

4.6.5.3 Identificación y estructuración de los procedimientos de prueba

Se puede trabajar caso de prueba a caso de prueba y sugerir procedimientos de prueba para cada uno. Pero lo mejor es intentar reutilizar procedimientos de prueba existentes tanto como sea posible, lo que significa que podemos necesitar modificarlos de forma que puedan usarse para especificar cómo realizar un caso de prueba nuevo o cambiado.

4.6.6 REALIZAR PRUEBAS DE INTEGRACIÓN

En esta actividad se realizan las pruebas de integración necesarias para cada una de las construcciones creadas en una iteración y se recopilan los resultados de las pruebas.

Las pruebas de integración se llevan a cabo en los siguientes pasos:

1. Realizar las pruebas de integración relevantes a la construcción.
2. Comparar los resultados de las pruebas con los resultados esperados e investigar los resultados de las pruebas que no coinciden con los esperados.
3. Informar de los defectos a los desarrolladores responsables de los componentes que se cree que contienen los fallos.
4. Informar de los defectos a los encargados de las pruebas, quienes usarán los defectos para evaluar los resultados del esfuerzo de prueba.

4.6.7 REALIZAR PRUEBA DE SISTEMA

El propósito de la *prueba de sistema* es el de realizar las pruebas de sistema necesarias en cada iteración y el recopilar los resultados de las pruebas. La *prueba de sistema* puede empezar cuando las pruebas de integración indican que el sistema satisface los objetivos de calidad de integración fijados en el plan de prueba de la iteración actual.

4.6.8 EVALUAR PRUEBA

El propósito de la *evaluación de la prueba* es el de evaluar los esfuerzos de prueba en una iteración. Los encargados de la prueba evalúan los resultados de la prueba comparando los resultados obtenidos con los objetivos esbozados en el plan de prueba. Estos preparan métricas que les permiten determinar el nivel de calidad del software y qué cantidad de pruebas es necesario hacer.

En el caso del Sistema de Control de Proyectos, las pruebas que fueron realizadas contemplaron dos grandes grupos, las cuales fueron referidas anteriormente:

1. Pruebas de integración

Las pruebas de integración contemplaron los casos de uso que se desarrollaron en el diseño. Para eso, se dividieron las pruebas en 2 grupos:

- Pruebas de integración dentro del caso de uso. Se probó el correcto funcionamiento de los componentes que forman parte de un caso de uso. Es decir, se comprobó que cada componente realizaba la función para la cuál se construyó.
- Pruebas de integración entre casos de uso. Se probó el correcto funcionamiento de la interacción entre los casos de uso. Es decir, se comprobó que un cambio realizado a un caso de uso no interfería con el funcionamiento de los otros, logrando de esta manera la independencia funcional entre ellos.

Para ambos tipos de pruebas, se mantuvo el plan de prueba de los casos de prueba que se tomaron en cuenta para un caso de uso. Es decir, se uso una plantilla como plan de prueba. Se decidió realizar la prueba de esta manera ya que funcionalmente, todos los casos de uso seguían los mismos pasos:

- Invocación (por medio de un enlace) al caso de uso
- Introducción de datos
- Validación de los datos
- Introducción de los datos en la BD
- Resultado de la acción realizada

2. Pruebas de sistema

Las pruebas de sistema contemplaron el funcionamiento del sistema al interior de Royal Systems, es decir, la integración sobre el servidor de

producción sobre el cuál residiría el mismo, la comunicación entre las diferentes estaciones que usarían el sistema, pruebas de stress. El resultado de esta prueba fue el correcto funcionamiento del Sistema de Control de Proyectos en la Intranet de la empresa. Hasta este momento, el sistema no se ha colocado en internet por motivos que se explican en las observaciones.

Hasta aquí lo correspondiente a la *prueba*. Se ha explicado como planificar una prueba por medio de la construcción de un *Modelo de pruebas*, el cuál contempla *casos de prueba* y *procedimientos de prueba*. Por otro lado, hemos mencionado las tareas necesarias al momento de realizar dicha planificación. Finalmente, mencionamos las acciones realizadas al momento de probar el Sistema de Control de Proyectos, obteniendo resultados positivos y alentadores.

CAPITULO V

ANÁLISIS ECONÓMICO

5.1 ESTRUCTURA DE LA INVERSIÓN

5.1.1 FIJA

5.1.1.1 Activos Fijos

En primer lugar, aclaremos que estos costos son solamente de la 1ra Etapa de la Propuesta de Solución Técnica, según lo indicado en el Capítulo 3. En este proyecto de desarrollo de software, el activo fijo estaría representado por el hardware necesario para el desarrollo. Se ha considerado un monto de US\$ 100 mensuales por el equipo utilizado.

5.1.1.2 Intangibles

Las licencias de software se pueden considerar de costo hundido ya que la empresa ya cuenta con el software necesario debido a la naturaleza de sus proyectos. Se asume también que el personal que se empleará en el desarrollo del sistema ya conoce las herramientas de desarrollo por lo que no habría gastos de capacitación.

5.1.2 CAPITAL DE TRABAJO

El tiempo de desarrollo del proyecto se ha calculado en 4 meses, durante estos meses los costos estarán dados por:

- Sueldo mensual de un programador: $US\$ 700 \times 4 = US\$ 2800$
- Sueldo mensual de un Jefe de Proyecto: $US\$ 2000$, pero solo se consideraría una dedicación del 25% de su tiempo, lo que nos da un costo mensual de $US\$ 500 \times 4 = US\$ 2000$
- La empresa tiene un ratio de $US\$ 20$ que se costea por día-hombre trabajado en el proyecto. Este ratio incluye todos los gastos corrientes (luz, agua, teléfono, internet, movilidad, gastos administrativos, depreciación de muebles y equipos, alquileres)
- Asimismo, para el desarrollo de los proyectos, la empresa considera un promedio de 20 días útiles por mes, lo que nos da un costo de $US\$ 400$ por mes-hombre trabajado. En nuestro caso, tendríamos $4 \text{ meses} \times 1.25 \text{ hombres} \times US\$ 400 = US\$ 2000$ (gastos corrientes)

En resumen, nuestra inversión inicial se resumiría de la siguiente manera:

Concepto	Monto
Sueldos de Jefe de Proyecto	US\$ 2000
Sueldos de Programador	US\$ 2800
Gastos Corrientes del Proyecto	US\$ 2000
Alquiler de Equipo	US\$ 400
Total Inversión Inicial	US\$ 7200

Cuadro 6.1 Inversión Inicial Total

5.2 ESTRUCTURA DE COSTOS

- Los costos, luego del *Pase a Producción* del sistema se han calculado en $US\$ 250$, basados en un 25% de dedicación de una persona de desarrollo del staff de la empresa. El sueldo de esta persona es de $US\$ 1000$.
- Los *Beneficios Directos* del proyecto serían los ahorros en días-hombre utilizados para preparar los informes del avance del sistema. Esto se ha calculado en $US\$ 500$ mensual aproximadamente considerando el tiempo de Jefes de Proyecto, Implementadores y Programadores.

- Hay un *Beneficio No Tangible*, expresado por la Gerencia de la empresa y es que el hecho de contar con la información adecuada permitirá tomar los correctivos adecuados, lo cuál redundará en el cumplimiento de los proyectos y por ende en el mejoramiento de la empresa y la posibilidad de conseguir otros proyectos con los clientes satisfechos. Es muy difícil medir este beneficio en términos monetarios, pero para efectos de este proyecto se ha considerado un beneficio pesimista de US\$ 6000 anuales, lo cual nos da un beneficio mensual de US\$ 500.

En resumen, nuestra estructura de costos se resumiría de la siguiente manera:

Concepto	Monto (US\$)
Ingresos	
Ingreso Directo	500.00
Ingreso Indirecto	500.00
Total Ingresos Mensuales	1000.00
Total Ingresos Anuales	12000.00
Egresos	
Costo de Mantenimiento	(250.00)
Total Egresos Mensuales	(250.00)
Total Egresos Anuales	(3000.00)
Flujo Anual	9000.00

Cuadro 6.2 Estructura de Costos

5.3 ALTERNATIVAS DE FINANCIAMIENTO

5.3.1 ACTIVOS FIJOS

Una alternativa fué tomar un *Leasing*, pero fue descartada debido al monto pequeño de la inversión (US\$ 400).

5.3.2 CAPITAL DE TRABAJO

Otra alternativa era realizar el sistema a manera de *Outsourcing*. Se cotizó el documento de análisis del sistema a una consultora de sistemas con experiencia en el desarrollo de aplicaciones Web, obteniéndose la siguiente información:

- Tiempo de Desarrollo: 2 meses
- Costo de Desarrollo: US\$ 5000.00
- Tiempo del Jefe del Proyecto por parte de Royal Systems: 2 meses x 50% = 1 meses
- Costo del Jefe del Proyecto por parte de Royal Systems: 1mes x US\$2000.00 = US\$ 1000.00
- Ventajas:
 - Desarrollo más rápido
 - Menor Costo Total: US\$ 6000.00, Ahorro: US\$ 1200.00
- Desventajas:
 - Se perdería la oportunidad de iniciar a la empresa en el desarrollo de aplicaciones Web.
 - Se perdería la confidencialidad sobre los procesos internos de la empresa.
 - Limitaciones en el desarrollo posterior de mejoras al sistema; no hay que olvidar que esto es solo la 1ra etapa del proyecto.

5.3.3 RATIOS FINANCIEROS

Los ratios mostrados son los vigentes al momento del inicio del proyecto, no hay alternativas significativas en este rubro.

5.4 EVALUACIÓN ECONÓMICA

5.4.1 VALOR ACTUAL NETO

Considerando un *Costo de Oportunidad de Capital (C.O.K)* de 15% anual y un horizonte de evaluación de 5 años, debido a la naturaleza del proyecto se obtiene un Valor Actual de US\$ 51,714.09 y un Valor Actual Neto de US\$ 44,514.09.

5.4.2 TASA INTERNA DE RETORNO

Para un horizonte de 5 años, se obtiene una Tasa Interna de Retorno de 124% anual.

5.4.3 BENEFICIO / COSTO

Para los mismos parámetros, se obtiene un Índice de Rentabilidad Beneficio/Costo (Valor Actual / Inversión) de 7.182512.

5.4.4 TIEMPO DE RECUPERACIÓN DE CAPITAL

Para los mismos parámetros, el Tiempo mínimo de Recuperación de Capital es de 1 año, al final del cual se obtiene un Valor Actual Neto de positivo de US\$ 1688.89.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

Algunas conclusiones producto de la elaboración de estas tesis son:

1. La situación actual de la empresa, muestra que un inadecuado control de proyectos conlleva una mala administración de los recursos de la empresa y consecuencia mayores costos para la empresa.
2. La Solución al problema incluye medidas administrativas y de Control. Parte la Solución Técnica incluye el desarrollo de un sistema de software, que debía ser accedido desde cualquier sitio, característica principal de una solución web. Además, la información recolectada por este sistema debía ser fácilmente manipulada por medio a mecanismos programáticos informáticos tales como reportes y/o consulta SQL. Por otra parte, el sistema debía ser fácil de administrar, interactuar y que fuera escalable, es decir, que fuera flexible para futuros cambios que se necesitaran hacer a fin de mejorar el sistema actual.
3. La construcción del sistema informático que nos permitió solucionar en gran medida los problemas mencionados anteriormente tomó como punto de partida el uso del *Proceso Unificado de Rational (RUP)*, proceso de desarrollo de software que se caracteriza por identificar de manera clara y concisa los requerimientos de un usuario y transformarlos en un sistema de software. Además, el RUP es un proceso de desarrollo de software **orientado a objeto**, principal característica que tiene el sistema

construido. Hemos hecho una explicación bastante incisiva de los cinco flujos de trabajo fundamentales, para de esta manera explicar los fundamentos teóricos del RUP.

4. La construcción del Sistema de Control de Proyectos se realizó a partir de la teoría explicada del RUP. De esta manera, siguiendo la metodología de trabajo propuesta por el RUP, en cada uno de los flujos de trabajo fundamentales hemos obtenido productos de salida que a su vez servían como productos de entrada para el siguiente flujo de trabajo fundamental. De esta manera hemos apreciado que en todo momento se tenía como resultado de nuestro trabajo un producto capaz de representar la realidad que se pretendía resolver. Además, valiéndonos del uso de iteraciones, característica principal del RUP, hemos utilizado los productos finales de cada flujo de trabajo de cada iteración a manera de punto de partida de la siguiente iteración, logrando de esta manera una retroalimentación constante y fructífera.
5. Como parte del uso de RUP, se realizaron las pruebas tanto de integración como de sistema, logrando resultados mas que satisfactorios. Como consecuencia de esta pruebas, obtuvimos modelos de prueba, es decir, situaciones por las que todo sistema debe pasar para comprobar que cumplen los objetivos esperados. En el caso de Royal Systems, estos modelos de prueba serán tomados a partir de la fecha como un estándar para realizar las correspondientes pruebas de los sistemas desarrollados y a desarrollar.
6. De la Evaluación Económica se concluye que el proyecto es viable económicamente, ya que a partir del primer año de implementación se recupera el capital invertido. A partir de ese momento, todos los beneficios que se obtengan se consideran utilidad del proyecto.

Algunas recomendaciones producto de la elaboración de esta tesis son:

1. Habiendo demostrado mediante este trabajo de tesis el uso del RUP con muy buenos resultados, se debe considerar el hecho de tomarlo como un estándar al interior de Royal Systems para su posterior desarrollo de sistemas de software. Además, es necesario considerar que los sistemas desarrollados son en sí subsistemas que forman parte de una suerte de gran Sistema de Información Gerencial (SIG). En este sentido, es recomendable el fomentar una cultura sistémica de desarrollo de software no sólo al interior de Royal Systems sino la de toda empresa que quiera desarrollar sistemas en serio.
2. Dado que el Sistema de Control de Proyectos sólo se implementó en la INTRANET de la empresa, una segunda recomendación sería la implementación sobre INTERNET. Para esto, un primer paso sería el cambio del servidor web con el cual tienen acceso a INTERNET (actualmente hablamos de un servidor web Apache que corre sobre un Sistema Operativo Linux). Además, al momento de realizar esta migración e implementación, es indispensable considerar las medidas de seguridad a adoptarse a fin de que el funcionamiento del sistema no se vea mellado por la presencia de hacker, crackers, etc.
3. Siempre queda la alternativa de usar los resultados obtenidos por el RUP para implementar el mismo Sistema de Control de Proyectos sobre la plataforma Java. Como se mencionó al comienzo de la tesis, el objetivo principal de la misma fue el de explicar de manera teórica y práctica el uso del RUP como proceso de desarrollo de software a la par de realizar una reingeniería al control de proyectos actual de la empresa. De esta manera, queda de parte de los futuros lectores el tratar de implementar los modelos obtenidos en el lenguaje de desarrollo que mejor le acomode.

BIBLIOGRAFIA

ANDERSON, R.; FRANCIS, B.; HOMER, A.; HOWARD, R.; SUSSMAN, D.; WATSON, K. 2001. *Professional ASP.NET*. Wrox Press Ltd.

ERIKSSON, H.; PENKER, M. 1998. *UML Toolkit*. Wiley Computer Publishing.

FOWLER, M.; SCOTT, K. 1997. *UML gota a gota*. Addison Wesley Longman.

GUNDERLOY, M.; CHIPMAN, M. 2001. *Microsoft SQL Server 2000*. Anaya.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. 1999. *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.

JOSHI, B; DICKINSON, P; FERRACCHIATI, F. 2001. *Professional ADO.NET Programming*. Wrox Press Ltd.

LARMAN, C. 1999. *UML y Patrones*. Pearson.

LOGIC WORKS, 1998. *Modelamiento de Datos usando Erwin*. Logic Works, Inc.

MANGANELLI, Raymond ; KLEIN, Mark, 1994, *Cómo Hacer Reingeniería*, Editorial Norma.

MANGER, J. 1997. *Fundamentos de JavaScript*. McGraw Hill.

ORTIN IBÁÑEZ, M. 2001. *Un Proceso basado en UML para aplicaciones Web*. Universidad de Murcia, España.

PRESSMAN, R. 1998. *Ingeniería de Software: Un enfoque práctico*. McGraw Hill.

QUATRANI, T, 1998. *Visual Modeling with Rational Rose and UML*. Addison Wesley

SCHMULLER, J. 2000. *Aprendiendo UML en 24 horas*. Prentice Hall.