



UNIVERSIDAD AUSTRAL DE CHILE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN INFORMÁTICA**

DISEÑO E IMPLEMENTACIÓN DE UN ASP (APPLICATION SERVICE PROVIDER) DE SISTEMAS ACADÉMICOS

**TESIS DE GRADO PARA OPTAR AL
TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA**

PATROCINANTE:

**RODRIGO GARBARINO FLORES
INGENIERO CIVIL EN INFORMÁTICA**

CO-PATROCINANTE:

**MARÍA ELIANA DE LA MAZA WERNER
INGENIERO CIVIL EN INFORMÁTICA**

PABLO ANTONIO ALARCÓN RODRÍGUEZ

VALDIVIA – CHILE

2004

Santiago, 17 de Agosto del 2004

DE: RODRIGO GARBARINO FLORES
GERENTE GENERAL TGA S.A.

A: MIGUELINA VEGA VEGA
DIRECTORA ESCUELA INGENIERÍA CIVIL EN INFORMÁTICA

MOTIVO:

Informar y revisión y calificación del Proyecto de Título: "Diseño e Implementación de un ASP (Application Service Provider) de Sistemas Académicos", realizado por Pablo Antonio Alarcón Rodríguez

FUNDAMENTO:

El proyecto de tesis ha cumplido con los objetivos planteados, satisfaciendo las necesidades de la empresa respecto del rediseño de procesos y de sistemas.

Durante la realización del proyecto se comprobó el adecuado planteamiento y desarrollo de la metodología científica.

En si el esfuerzo llevado a cabo contribuyó con los fines establecido como primordiales para la empresa, destacándose el adecuado enfoque e interpretación de las conclusiones que permiten la aplicación de gran parte del trabajo realizado.

Es necesario destacar que por otro lado se comprueba una adecuada precisión en el lenguaje técnico, composición, redacción e ilustración de este proyecto.

En conclusión este trabajo de tesis es evaluado con la nota máxima.

NOMBRE TRABAJO DE TITULACIÓN:

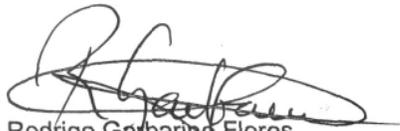
DISEÑO E IMPLEMENTACIÓN DE UN ASP (APPLICATION SERVICE PROVIDER) DE SISTEMAS ACADÉMICOS

NOMBRE DEL ALUMNO:

PABLO ANTONIO ALARCÓN RODRÍGUEZ

NOTA : 7,0
(en números)

Siete coma cero
(en letras)


Rodrigo Garbarino Flores
Gerente General
Tecnología y Gestión Académica

Valdivia, 16 de agosto de 2004

DE : Prof. María Eliana de la Maza W.
Instituto de Informática

A : Sra. Miguelina Vega R.
Directora Escuela de Ingeniería Civil en Informática

MOTIVO : Informar revisión y calificación del Proyecto de Título "Diseño e implementación de un ASP (Application Service Provider) de sistemas académicos", presentado por el alumno Pablo Antonio Alarcón Rodríguez, que refleja lo siguiente:

Se logró el objetivo planteado de estandarizar y parametrizar un sistema académico de manera de llevarlo desde un esquema de producto a un esquema de servicio. Lo anterior debería producir un impacto en la gestión de la empresa TGA S.A., al mejorar, entre otras cosas, el manejo del control de versiones.

Se presenta, en el documento, el modelo ASP e información relevante sobre Gestión de la Configuración.

Al leer el documento se aprecia la aplicación de criterios adecuados de análisis y diseño, además de una precisión en la redacción y en el lenguaje técnico utilizado.

Por lo anteriormente expuesto, y además en mi calidad de co-patrocinante, por conocer el trabajo realizado por el alumno, califico la tesis presentada con nota seis coma ocho (6,8).

Con este particular, saluda atte. a Ud.,



María Eliana de la Maza W.
Profesora Instituto de Informática



Universidad Austral de Chile

Instituto de Informática

Valdivia, 20 de agosto de 2004

Sra.

Miguelina Vega R.

Directora Escuela Ingeniería Civil en Informática

De mi consideración:

Mediante la presente, hago llegar a Ud. mi evaluación como profesor informante, del trabajo de Tesis de Grado de Ingeniero Civil en Informática del Sr. Pablo Antonio Alarcón Rodríguez, titulado "DISEÑO E IMPLEMENTACIÓN DE UN ASP (APPLICATION SERVICE PROVIDER) DE SISTEMAS ACADÉMICOS".

Estimo que el trabajo de titulación del Sr. Pablo Alarcón plantea de manera interesante el problema de transformación de una aplicación desarrollada en distintas versiones (producto) en un servicio de aplicación de software. La fundamentación teórica asociada a esta transformación es pertinente y precisa.

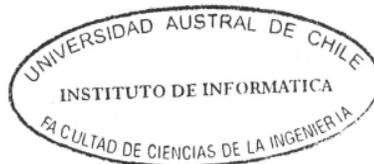
Respecto a la solución desarrollada, su enfoque plantea un claro énfasis en aspectos de interfaz (pantallas, informes) y en la herramienta de software utilizada para la implementación del esquema ASP, en desmedro de la re-definición de los procesos, que es lo que se indica en los objetivos del trabajo. No obstante, el énfasis dado a la solución propuesta tiene importantes méritos asociados a la viabilidad y documentación de la migración a un esquema de administración de aplicaciones de software cada vez más utilizado, como el ASP.

Por todo lo anteriormente expuesto, califico el trabajo de titulación del Sr. Pablo Alarcón Rodríguez, con nota 6.2 (seis coma dos).

Sin otro particular, se despide atte.

Mauricio Ruiz-Tagle Molina

Instituto de Informática



AGRADECIMIENTOS

Mis agradecimientos a la empresa TGA S.A. por la oportunidad de haber realizado esta tesis en sus dependencias. A Rodrigo Garbarino por su apoyo y orientación en el desarrollo de este trabajo de titulación. A mi querida profesora María Eliana de la Maza por su buena voluntad, disposición y sus recomendaciones.

A ti Vicky por todo el amor que me has entregado pero sobre todo, por tu apoyo incondicional sin el cual no hubiese podido lograr esta meta tan importante en mi vida.

A mi nueva familia, mi cuñada Patty y en especial a mi queridísima suegri, de quien siempre recibí un gran apoyo y a quien le debo mucho.

Quiero dedicar este logro a mis Padres, quienes hicieron un enorme esfuerzo para educarme y a quienes les debo todo lo que soy. Gracias por todo el cariño y amor que me han entregado siempre.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS.....	2
ÍNDICE DE CONTENIDOS.....	3
ÍNDICE DE FIGURAS.....	7
RESUMEN.....	8
SUMMARY.....	9
1INTRODUCCIÓN.....	10
1.1Antecedentes Generales.....	10
1.2ASP.....	11
1.3Sistema Académico.....	12
1.4Objetivos Generales.....	13
1.5Objetivos Específicos.....	13
2SERVICIOS ASP TGA.....	14
2.1Introducción.....	14
2.2Sistema Académico.....	14
2.2.1Servidor.....	15
2.2.2Cliente.....	15
2.3Problemas.....	15
2.3.1Administración de Programas.....	16
2.3.2Especialización de los Desarrolladores.....	16
2.3.3Código Inservible.....	17
2.3.4Versiones.....	17
2.3.5Modificaciones.....	17
2.3.6Omisiones.....	17
2.4Desarrollo.....	18
2.5Conclusión.....	18
3METODOLOGÍA.....	19
3.1Introducción.....	19
3.2Proceso de Software.....	20
3.2.1Modelos del Proceso de Software.....	22
3.2.1.1El Modelo de Cascada.....	23

3.3	Reingeniería de Procesos de Negocio (RPN).....	25
3.3.1	Procesos de Negocio.....	25
3.3.2	Modelo RPN.....	27
3.4	Application Service Provider (ASP).....	29
3.4.1	Modelo ASP.....	29
3.4.2	Ventajas de un ASP.....	30
3.4.3	Infraestructura Tecnológica.....	32
3.5	Gestión de la Configuración (GC).....	33
3.5.1	Planificación de la Gestión de la Configuración.....	35
3.5.1.1	Identificación del Elemento de la Configuración.....	36
3.5.1.2	Base de Datos de la Configuración.....	37
3.5.2	Administración del Cambio.....	38
3.5.3	Administración de Versiones y Liberaciones.....	39
3.5.3.1	Identificación de Versiones.....	40
3.5.3.2	Administración de las Liberaciones.....	41
3.5.4	Herramientas CASE para la Gestión de la Configuración.....	42
3.5.4.1	Apoyo a la Administración de Cambios.....	42
3.5.4.2	Apoyo a la Administración de Versiones.....	43
4	SITUACIÓN ACTUAL.....	45
4.1	Introducción.....	45
4.2	Desarrollo y Mantenimiento del Software.....	46
4.3	Procesos de Negocio del Sistema.....	47
4.3.1	Módulo de Admisión.....	47
4.3.1.1	Principales Funcionalidades.....	48
4.3.1.2	Preinscripción.....	49
4.3.1.3	Convocatoria.....	49
4.3.1.4	Inscripción.....	49
4.3.1.5	Aranceles.....	50
4.3.1.6	Entrevista.....	50
4.3.1.7	Selección.....	50
4.3.1.8	Informes.....	51
4.3.2	Módulo de Caja.....	51
4.3.2.1	Principales Funcionalidades.....	52
4.3.2.2	Caja Contrato.....	53
4.3.2.3	Beneficios.....	53
4.3.2.4	Modificación de Contrato.....	53

4.3.2.5Caja.....	54
4.3.2.6Cierre de Caja.....	54
4.3.2.7Informes.....	54
4.3.3Módulo Curricular.....	55
4.3.3.1Principales Funcionalidades.....	55
4.3.3.2Usuarios.....	57
4.3.3.3Carreras.....	57
4.3.3.4Asignaturas.....	57
4.3.3.5Currículum de estudio.....	57
4.3.3.6Toma de asignaturas.....	58
4.3.3.7Ponderaciones.....	58
4.3.3.8Notas.....	58
4.3.3.9Solicitudes.....	58
4.3.3.10 Carga.....	59
4.3.3.11 Cierres.....	59
4.3.3.12 Profesores.....	59
4.3.3.13 Histórico.....	59
4.3.3.14 Informes.....	59
5SOLUCIÓN.....	61
5.1Introducción.....	61
5.2Modificaciones al Sistema.....	62
5.2.1Módulo Admisión.....	63
5.2.1.1Convocatoria.....	64
5.2.1.2Inscripción.....	64
5.2.1.3Tutor Financiero.....	65
5.2.1.4Asignación de entrevistas.....	65
5.2.1.5Entrevistas.....	65
5.2.1.6Ponderaciones.....	66
5.2.1.7Reportes.....	66
5.2.2Módulo Caja.....	67
5.2.2.1Alumnos Nuevos.....	67
5.2.2.2Alumnos Antiguos.....	67
5.2.2.3Tutor Financiero.....	67
5.2.2.4Reimpresión de documentos.....	68
5.2.2.5Activación de Ficha.....	68
5.2.2.6Envío de letras Banco BCI.....	68

5.2.2.7Envío de letras Banco Santiago.....	68
5.2.2.8Reportes.....	68
5.2.3Módulo Curricular.....	70
5.3Modificaciones al proceso de desarrollo.....	72
5.3.1Oracle SCM.....	72
5.3.1.1Almacenamiento de datos.....	73
5.3.1.2Contenedores.....	74
5.3.1.3Administración de datos.....	75
5.3.1.3.1Áreas de trabajo.....	75
5.3.1.3.2Configuraciones.....	76
5.3.1.4Acceso al repositorio.....	77
5.3.1.5Implementación de Oracle SCM.....	78
5.3.1.5.1Áreas de Trabajo.....	79
5.3.1.5.2Contenedores.....	79
5.3.1.5.3Archivos.....	80
5.3.1.5.4Control de Versiones.....	81
5.3.1.5.5Comparación de Versiones.....	82
5.3.1.5.6Combinación.....	83
5.3.1.5.7Configuraciones.....	84
5.3.1.6Análisis de impacto.....	84
5.3.1.7Resumen.....	85
5.3.2Pasos a Producción.....	85
5.3.3Mesa de Ayuda.....	86
6CONCLUSIONES.....	87
7BIBLIOGRAFÍA.....	90

ÍNDICE DE FIGURAS

Figura 1-1. Modelo de interacción de los sistemas académicos.....	13
Figura 3-1. El ciclo de vida del software.....	23
Figura 3-2. El negocio: una jerarquía de sistemas.....	26
Figura 3-3. Un modelo de reingeniería de procesos de negocio.....	28
Figura 3-4. Evolución del software.....	29
Figura 3-5. Modelo ASP.....	30
Figura 3-6. Familias de sistemas.....	34
Figura 3-7. Jerarquía de la configuración.....	36
Figura 3-8. Estructura de derivaciones de las versiones.....	41
Figura 3-9. Descripción delta de las versiones.....	44
Figura 4-1. DFD del módulo de admisión.....	47
Figura 4-2. DFD del módulo de caja.....	52
Figura 4-3. DFD del módulo curricular.....	55
Figura 5-1. Modelo de proceso operativo de la toma de ramos.....	71
Figura 5-2. Repository Object Navigator mostrando contenedores.....	75
Figura 5-3. Vista de un área de trabajo del repositorio.....	76
Figura 5-4. Configuración a partir de versiones de objetos.....	77
Figura 5-5. Área de trabajo.....	79
Figura 5-6. Contenedor.....	80
Figura 5-7. Archivo.....	81
Figura 5-8. Control de versiones.....	82
Figura 5-9. Comparación de versiones.....	82
Figura 5-10. Combinación de versiones.....	83
Figura 5-11. Configuración.....	84

RESUMEN

Este proyecto tiene como objetivo transformar el actual Sistema TGA en un proceso de software, con el fin de llevar este sistema de un esquema de producto a uno de servicio. Para lograr este objetivo se hará uso de técnicas de reingeniería junto con modelos de procesos de Ingeniería de Software. Con la realización de este proyecto se logrará posicionar a la empresa TGA S.A. en el mercado junto con proporcionarle la arquitectura necesaria para convertirse en un proveedor de servicios de aplicación.

El sistema a modificar es una aplicación desarrollada por TGA S.A. para proveer servicios informáticos a las instituciones de educación. Básicamente este producto cuenta con tres módulos (admisión, caja y curricular) que son los encargados de gestionar todos los procesos requeridos en este ámbito.

Este proyecto contribuirá a que TGA S.A. logre una optimización como ASP; es decir, ser una empresa dedicada a la implementación de software y hardware para que así las instituciones se dediquen sólo a su negocio educacional. Además de proporcionarle la arquitectura necesaria para que sea un proveedor de servicios de aplicación.

La motivación para desarrollar este proyecto nace de la necesidad que tiene la empresa de potenciar su producto así como de la posibilidad de aplicar innovación tecnológica para transformar un producto de software en un sofisticado y completo servicio que tendrá una aplicación real y con resultados comprobables.

SUMMARY

The objective of this project is transform the current System TGA into a software process, in order to take this system of a scheme of product to one of service. In order to obtain this objective one will use reengineering techniques and software engineering process models. With the accomplishment of this project it will be achieved to position the company TGA S.A. in the market along with to provide the necessary architecture to become a application services provider.

The system to modify is an application developed by TGA S.A. to provide computer services to the education institutions. Basically this product have three modules (admission, financial and curricular) that managing all the needed processes in this scope.

This project will contribute to that TGA S.A. obtain an optimization as ASP, is to say, to be a company dedicated to the hardware and software implementation in order that this way the institutions devote themselves only to their educational business. Beside providing the necessary architecture in order that it is a application services provider.

The motivation to develop this project is born of the need that has the company to promote their product as well as the possibility of applying technological innovation to transform a software product into a sophisticated and complete service that will have a real application and with verifiable results.

1 INTRODUCCIÓN

1.1 Antecedentes Generales

Tecnología y Gestión Académica S.A. (TGA S.A.) es una empresa dedicada a crear soluciones informáticas en el ámbito académico, tiene su génesis en el año 1998 bajo el alero de EFT Banca, empresa dedicada al procesamiento de transferencias electrónicas de fondos. A fines del año 2000, nace como empresa autónoma Tecnología y Gestión Académica S. A., manteniendo a EFT Banca como su principal accionista.

Esta segmentación se produce a partir de una necesidad detectada en el mercado por contar con un grupo de profesionales dedicados completamente a los temas académicos, que permitieran entregar una atención más personalizada, especializada y adecuada a las necesidades crecientes y cambiantes de cada institución que se encuentra compitiendo en este mercado globalizado.

De esta forma, Tecnología y Gestión Académica se posiciona en el mercado como una empresa dedicada a proveer sistemas académicos con una amplia oferta de implantaciones, que van desde el outsourcing total del área académica, hasta instalaciones administradas por los clientes con actualización de versiones por parte de TGA.

1.2 ASP

En el ámbito de la informática la diferenciación entre lo que es un producto y lo que es un servicio es muy difícil, dado que la mayoría de las veces ambos son intangibles y con características en común [URL 4].

En la tabla 1-1 se observan las principales diferencias entre un servicio y un producto.

Producto	Servicio
La puesta en marcha empieza desde cero	Preparametrización avanzada
La administración y el mantenimiento no son incluidos	Administración y mantenimiento incluidos en el servicio
Recursos asociados a una tecnología	Recursos liberados para tareas de generación de valor
Peligro de obsolescencia	Continua vanguardia tecnológica

Tabla 1-1. Comparación de las principales diferencias entre producto y servicio

Un Application Service Provider (ASP) es una empresa prestadora de servicios de hospedaje de aplicaciones, la cual renta al usuario parte o toda la aplicación dependiendo de las necesidades del cliente, con el fin de evitarle parte de la implantación, administración y soporte a la infraestructura, así como también le permite reducir sus costos. A continuación se detallan las principales ventajas de un ASP:

- Menor costo de implantación / operación
- Permite que la empresa se dedique a su negocio central
- Mejor inversión en recursos
- Mejoramiento de los niveles de servicio
- Reducción de riesgos
- Esquema totalmente flexible y escalable

- Rápida implantación de software

1.3 Sistema Académico

El conjunto de sistemas que permiten abordar la solución completa de un sistema académico está construido sobre una base de datos Oracle con la suite de herramientas Oracle Developer 2000. Este sistema académico permite llevar un control centralizado de la información, proporcionando la interacción entre los módulos de Admisión, Caja y Curricular. De esa manera se obtiene una solución integrada y con todas las posibilidades que se requieren para una gestión moderna de la institución.

El sistema desarrollado por TGA para resolver la problemática de gestionar adecuadamente los datos del alumno durante su permanencia en la institución académica, parte desde el momento en el cual el alumno en su calidad de postulante se acerca a preguntar por alguna carrera de su preferencia y finaliza cuando éste termina el proceso de titulación en la institución, sin perjuicio que su historial quede guardado en la base de datos para realizar cualquier tipo de consulta en el futuro.

Los sistemas académicos de TGA operan sobre una base de datos centralizada, lo que implica mayores niveles de eficiencia pues no existe redundancia ni ausencia de información a la hora de ejecutar los procesos. Por otra parte, el uso de este tipo de base de datos posibilita que los cambios realizados en los datos de los alumnos, profesores, asignaturas, sedes, aulas, etc., se reflejen inmediata y automáticamente en el resto de los sistemas. De esta forma, se genera una interacción entre ellos que enriquece la gestión de la información de la institución.

En la figura 1-1 se observa un esquema de la interacción de los sistemas a partir del uso común de la base de datos única.

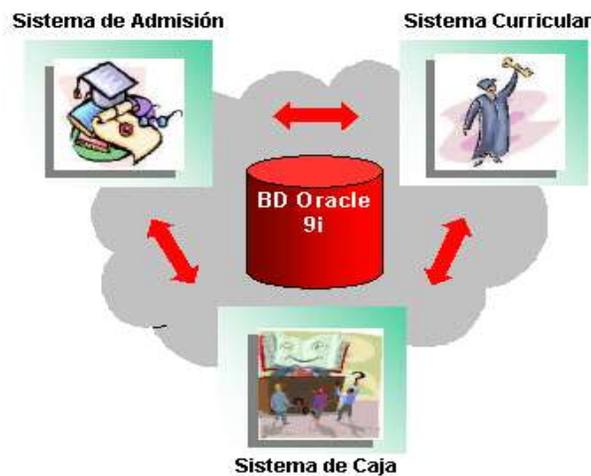


Figura 1-1. Modelo de interacción de los sistemas académicos

1.4 Objetivos Generales

Este proyecto tiene como objetivo estandarizar y parametrizar el Sistema TGA para transformarlo en un proceso de software; es decir, llevar este sistema de un esquema de producto a uno de servicio. Además, se pretende proveer la arquitectura necesaria para que la empresa TGA sea un proveedor de servicios de aplicación.

1.5 Objetivos Específicos

- Definir los procesos de negocio asociados al producto
- Diseñar el nuevo flujo de procesos del sistema asociado a la realidad de un proceso de software evolutivo y paramétrico
- Diseñar la arquitectura de procesos para soportar un proveedor de servicios de aplicación
- Desarrollar el diseño propuesto
- Evaluar el producto desarrollado

2 SERVICIOS ASP TGA

2.1 Introducción

Básicamente los problemas detectados se pueden agrupar en dos ámbitos, los que corresponden a los procesos de negocio del sistema y los que corresponden a la metodología de desarrollo y mantenimiento del software.

A continuación se explicará la naturaleza del problema a solucionar en este proyecto. Para una mejor comprensión de la problemática se explicará en qué consiste el sistema académico elaborado por TGA y cuales son los inconvenientes que este presenta.

2.2 Sistema Académico

En rasgos generales, el sistema académico de TGA está desarrollado con la herramienta Oracle Developer 2000 (básicamente con Oracle Forms y Oracle Reports), se sustenta sobre una base de datos Oracle y se compone de tres módulos: admisión, caja y curricular.

El sistema funciona sobre una plataforma cliente-servidor que se detalla a continuación.

2.2.1 Servidor

Corresponde al servidor donde se encuentra la base de datos del sistema. Dependiendo del tipo de servicio contratado esta base de datos se puede encontrar ubicada en la institución educacional o en las dependencias de TGA S.A. En este último caso, debe existir un enlace que permita la conexión entre TGA y la institución de educación o su sede central, en caso de ser multisede.

2.2.2 Cliente

Corresponde a la institución en donde está instalado el sistema. Esta debe contar con un servidor principal de aplicaciones, el que mantiene los programas ejecutables de cada módulo del sistema.

Para tener un mayor control de los archivos en el servidor, se crea una estructura de directorio que permita separar los archivos según el módulo al que pertenezcan y así poder mantener la última versión vigente del sistema.

Para el caso de las instituciones multisede, se tiene un servidor en cada sede con la misma estructura del que se encuentra en la sede central. Además, se cuenta con un procedimiento encargado de actualizar diariamente los programas, con lo que se asegura que siempre se cuente con la última versión del sistema en cada una de las sedes.

Para que una estación de trabajo pueda ejecutar los módulos del sistema y conectarse a la base de datos, se instala el sql*net y los componentes runtime del Developer 2000.

2.3 Problemas

El principal problema que presenta el sistema es la existencia de más de una versión de la aplicación, esto se debe a que cada cliente posee su propia instalación personalizada.

Naturalmente, existen muchas diferencias entre las distintas instituciones que componen el mercado de la educación superior, ya sea por el tipo de organización (centro de formación técnica, instituto profesional, universidad), su estructura corporativa (multisedes), tipo de postulaciones, etc. Lógicamente

estas diferencias involucran que un sistema académico deba satisfacer necesidades muy distintas e incluso opuestas.

Las diferencias entre las instituciones se pueden apreciar en muchas funcionalidades, un ejemplo claro es el proceso de postulación. En una institución este proceso puede depender de puntajes obtenidos en la PSU y notas de enseñanza media; en otra en cambio, puede depender de una entrevista adicional, incluso en algunas sólo puede ser necesario el matricularse. En este mismo proceso se pueden observar diferencias no sólo administrativas sino económicas; es decir, pueden haber instituciones en que este proceso no tenga costo y en otras si lo tenga.

Por tal razón, el sistema académico desarrollado por TGA se ha tenido que adaptar a cada cliente en particular, lo que ha producido tantas versiones del sistema como clientes en donde está instalada la aplicación. A continuación se describirán los principales problemas que se originan por esta situación.

2.3.1 Administración de Programas

Tener que administrar y mantener programas por separado dependiendo de cada cliente; es decir, separar los archivos en distintas carpetas según al cliente que correspondan. A su vez, esto produce una gran cantidad de archivos duplicados ya que la mayoría de los programas son los mismos y no poseen diferencias

2.3.2 Especialización de los Desarrolladores

La inevitable especialización de los desarrolladores con una sola versión, ya que las personas pasan a ser los “encargados” de un cliente en particular y la consecuente dependencia que se produce en el equipo de desarrollo.

Al existir grandes diferencias en las funcionalidades del sistema entre distintas versiones, es natural que el desarrollador (o grupo de desarrolladores) a cargo de programar componentes específicos de una versión, sea el más capacitado en ese componente, de esa versión en particular. De esta manera esta práctica se va haciendo repetitiva en el tiempo, lo que provoca la especialización y dependencia antes mencionada.

2.3.3 Código Inservible

La gran cantidad de código inservible en los programas ya que muchas veces se prefiere “comentar” líneas antes que borrarlas al momento de reutilizar código de otras versiones. De esta misma manera, existe gran cantidad de programas con funcionalidades o módulos que no se aplican al ser traspasados de una versión a otra, pero que se prefiere desactivar o esconder en vez de eliminar por la falta de seguridad al trabajar con distintas versiones o clientes.

2.3.4 Versiones

Poseer instaladas tantas versiones como clientes se tengan para poder darles soporte; es decir, cada persona dedicada a dar soporte al sistema deberá tener instaladas en su computador todas las versiones existentes, ya que de esta manera se asegura la entrega de un correcto soporte a todos los clientes. Por esta misma razón, cada vez que se realiza una modificación de alguna versión en particular, se debe comunicar a todo el equipo de desarrollo y soporte

2.3.5 Modificaciones

Evaluar cada modificación solicitada por un cliente para determinar la posibilidad de aplicarla a todas las versiones restantes, esto siempre y cuando sea factible hacerlo y se considere una mejora al sistema. Dicha evaluación inevitablemente demorará la entrega por el hecho de tener que revisar las excepciones de cada versión en particular.

2.3.6 Omisiones

La eventual omisión de cambiar los textos de pantalla y/o reportes que hagan alusión a un cliente en particular cuando se emplean para otra versión. Esto resta seriedad de la imagen que tiene el cliente hacia el equipo de desarrollo y la empresa en general, complicando las relaciones.

2.4 Desarrollo

Sin lugar a dudas el tener un sistema con múltiples versiones acarrea una gran cantidad de problemas; sin embargo, existe otro problema que no es menor y que tiene relación con la metodología de desarrollo y mantención del software.

Actualmente el equipo de desarrollo ve al sistema académico como un producto; es decir, una aplicación que se desarrolló bajo un modelo clásico. Lo que implica que esta aplicación fue diseñada, luego fue desarrollada para finalmente ser entregada al cliente, con lo cual sólo resta su mantención.

Contrariamente a este pensamiento se encuentra el esquema de servicio, en el cual la aplicación debe ir mejorando y evolucionando continuamente para así adaptarse a las nuevas necesidades que surgen en el mercado.

Por tal razón, más que modificar el software propiamente tal se hace indispensable cambiar la mentalidad del equipo de desarrollo, así como la metodología de desarrollo empleada hasta ahora.

2.5 Conclusión

Todos los problemas anteriormente descritos se descubrieron luego de un levantamiento realizado tanto al sistema propiamente tal como a la metodología de desarrollo y mantención del software, los que a su vez dan origen y son el motivo del cambio que se desea realizar llevando esta práctica a un esquema de ASP, ya que tal como se menciona en el capítulo anterior, en la práctica están presentes todos los problemas que se solucionan bajo este esquema.

3 METODOLOGÍA

3.1 Introducción

Una de las técnicas más conocidas para mejorar y/o reestructurar un producto de software es la reingeniería de procesos, metodología apropiada para revisar y rediseñar procesos, así como para implementarlos, enfocándose en agregar valor a cada uno de los pasos de un proceso y eliminar aquellos que no den o no puedan dar ningún valor agregado, siendo muy apropiada para generar organizaciones horizontales y organizaciones por procesos, así como para reducir costos, tiempos de procesos, mejorar el servicio y los productos, como también para mejorar la motivación y la participación del personal [URL 5].

Según la definición del IEEE (Institute of Electrical and Electronics Engineers), software es la suma total de programas de computadora, procedimientos, reglas, documentación asociada y los datos que pertenecen a un sistema de cómputo y un producto de software es un producto diseñado para un usuario. En este contexto, la ingeniería de software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, en conclusión, la ingeniería de software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones económicas a los problemas de desarrollo de software; es decir, permite elaborar consistentemente productos correctos, utilizables y eficaces en costo [URL 6].

El proceso de ingeniería de software se define como un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad. El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código; el código es probado, documentado y certificado para su uso operativo. Concretamente se define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.

Cuando una aplicación computacional o software alcanza su “madurez”; es decir, llega a tal punto en su desarrollo que no le es posible seguir creciendo se dice que comienza a “envejecer”. Las implicaciones que conlleva esto son muchas y muy variadas, van desde la falta de actualización y estancamiento del software (obsolescencia del producto) hasta la tan temida pérdida de clientes y la posterior desaparición del producto del mercado [Pre98].

Reparar lo que esté mal y darle una imagen más moderna a una aplicación muchas veces no es suficiente. Está comprobado que a medida que se hacen cambios en el software (mantenimiento), es bastante probable que se introduzcan nuevos defectos haciendo que el nivel de fallos de la aplicación crezca, por lo que el software se va deteriorando. Lo que se requiere en estos casos es una reingeniería o reestructuración importante de todo el software, de lo contrario el producto irá perdiendo cada vez más competitividad. Desgraciadamente muchas empresas no están dispuestas a comprometer recursos en esta reestructuración.

En resumen, para potenciar un producto software y obtener óptimos resultados se hace indispensable una reestructuración de la aplicación, acompañada por supuesto, de un proceso de ingeniería de software y tecnologías de información apropiadas.

3.2 Proceso de Software

Un proceso de software es un conjunto de actividades y resultados asociados que conducen a la creación de un producto de software. Esto puede

consistir en el desarrollo de software desde cero, aunque cada vez más se desarrolla nuevo software ampliando y modificando los sistemas existentes [Som02].

Los procesos del software son complejos y, como en todo proceso intelectual, se basan en el juicio humano. Debido a la necesidad de juzgar y crear, los intentos para automatizar estos procesos han tenido éxito limitado. Las herramientas CASE (Ingeniería de Software Asistida por Computadora) pueden ayudar en algunas actividades del proceso, pero no existe posibilidad alguna de una automatización mayor en el diseño creativo del software.

Una razón por la cual existe un enfoque limitado en el proceso de automatización es la inmensa diversidad de procesos del software. No existe un proceso ideal y diferentes organizaciones han desarrollado enfoques completamente diferentes para desarrollar software. Los procesos han evolucionado para explotar las capacidades de la gente de una organización, así como las características específicas de los sistemas que se están desarrollando. Por lo tanto, aun dentro de la misma compañía pueden existir muchos procesos diferentes para desarrollar software.

Aunque existen muchos procesos diferentes de software, tienen actividades fundamentales que son comunes para todos ellos. Éstas son:

1. *Especificación del software.* Se debe definir la funcionalidad del software y las restricciones en sus operaciones
2. *Diseño e implementación del software.* Se debe producir software que cumpla su especificación
3. *Validación del software.* Se debe validar el software para asegurar que hace lo que el cliente desea
4. *Evolución del software.* El software debe evolucionar para cumplir los cambios en las necesidades del usuario

3.2.1 Modelos del Proceso de Software

Un modelo del proceso de software es una representación abstracta de un proceso de software. Cada modelo representa un proceso desde una perspectiva particular por lo que sólo provee información parcial acerca de él [Som02].

Estos modelos generales no son descripciones definitivas de los procesos de software. Más bien, son abstracciones útiles que se pueden utilizar para explicar diferentes enfoques de desarrollo de software. Por supuesto, en sistemas muy grandes, no se usa un proceso de software sencillo. Para desarrollar diversas partes del sistema, se utilizan diferentes procesos.

Los principales modelos de procesos son:

1. *El modelo de cascada.* Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etc.
2. *Desarrollo evolutivo.* Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un primer sistema se desarrolla rápidamente a partir de especificaciones abstractas. Entonces, se refina con la ayuda del cliente para producir un sistema que satisfaga sus necesidades
3. *Desarrollo formal de sistemas.* Este enfoque se basa en la producción de una especificación matemática formal del sistema y en la transformación de esta especificación, utilizando métodos matemáticos, para construir un programa. La verificación de los componentes del sistema se lleva a cabo haciendo argumentos matemáticos acordes a su especificación
4. *Desarrollo basado en la reutilización.* Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero

Los procesos que se basan en el modelo de cascada y en el de desarrollo evolutivo se utilizan ampliamente en el desarrollo de sistemas prácticos. El desarrollo formal de sistemas se ha utilizado exitosamente en varios proyectos, pero pocas organizaciones utilizan los procesos del modelo. La reutilización informal es común en muchos procesos, pero la mayoría de las organizaciones no orientan el desarrollo de software alrededor de ésta.

Dado que para este proyecto se tienen claramente definidos sus alcances, a continuación sólo se describe el modelo utilizado: el modelo de cascada.

3.2.1.1 El Modelo de Cascada

El primer modelo de procesos de desarrollo de software que se publicó se derivó de otros procesos de ingeniería. Éste se muestra en la figura 3-1. Debido a la cascada de una fase a otra, este modelo se conoce como “modelo de cascada” o como ciclo de vida del software [Som02].

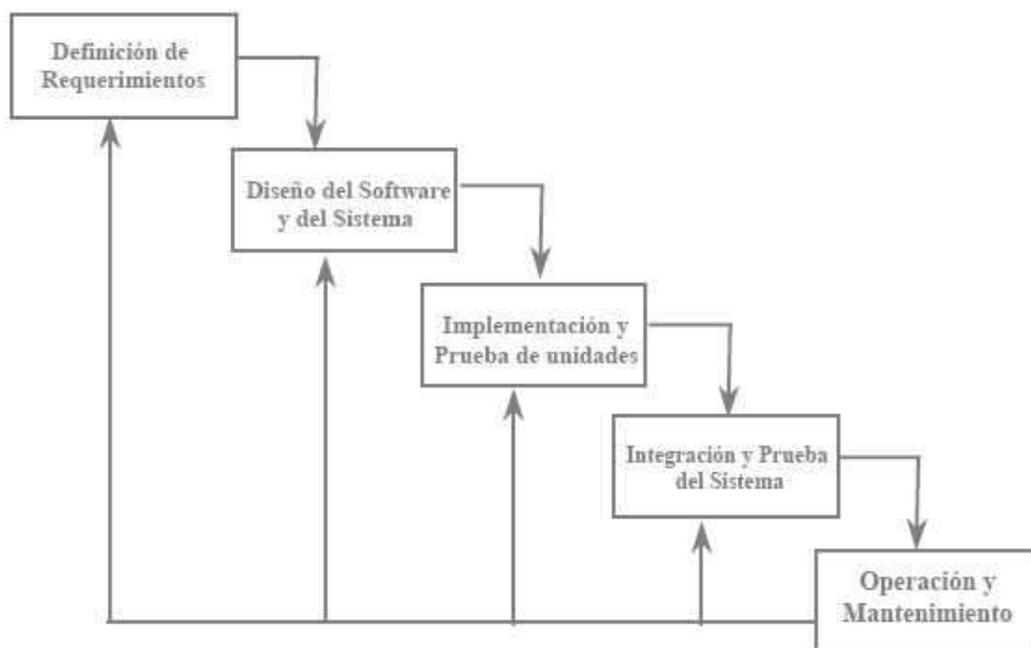


Figura 3-1. El ciclo de vida del software

Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo.

1. *Análisis y definición de requerimientos.* Los servicios, restricciones y metas del sistema se definen a partir de consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema
2. *Diseño de sistemas y de software.* El proceso de diseño de sistemas divide los requerimientos en sistemas de hardware o de software. Establece una arquitectura completa del sistema. El diseño de software identifica y describe las abstracciones fundamentales del sistema de software y sus relaciones
3. *Implementación y prueba de unidades.* Durante esta etapa, el diseño de software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación
4. *Integración y prueba del sistema.* Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema de software se entrega al usuario
5. *Operación y mantenimiento.* Por lo general, aunque no necesariamente, ésta es la fase más grande del ciclo de vida. El sistema se instala y pone en uso práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos

En principio, el resultado de cada fase es uno o más documentos aprobados. La siguiente fase empieza hasta que las otras hayan finalizado. En la práctica, estas etapas se traslapan y proporcionan información a las otras. Durante el diseño se identifican los problemas con los requerimientos, durante el diseño del código se hace la misma identificación, y así sucesivamente. El

proceso de software no es un modelo lineal simple, sino que implica una serie de iteraciones de las actividades de desarrollo.

Debido a los costos de producción y aprobación de documentos, las iteraciones son costosas e implican rehacer el trabajo. Por lo tanto, después de un número reducido de iteraciones, es normal congelar partes del desarrollo, como la especificación y continuar con las siguientes etapas de desarrollo. Los problemas se posponen para su resolución, se ignoran o se programan. Este congelamiento prematuro de requerimientos podría implicar que el sistema no haga lo que los usuarios desean. También conduce a sistemas mal estructurados debido a que los problemas se resuelven mediante trucos de implementación.

Durante la fase final del ciclo de vida (operación y mantenimiento) el software se pone en funcionamiento. Se descubren errores y omisiones en los requerimientos originales del software. Los errores de programación y diseño emergen y se identifica la necesidad de una nueva funcionalidad. Por lo tanto, el sistema debe evolucionar para mantenerse útil. Hacer estos cambios (mantenimiento del software) implica repetir alguna o todas las etapas previas del proceso.

Este modelo sólo se debe utilizar cuando los requerimientos se comprendan del todo. Sin embargo, el modelo refleja la práctica de la ingeniería. Por consiguiente, los procesos de software que se basan en este enfoque se siguen utilizando para el desarrollo de software, particularmente cuando éste es parte de proyectos grandes de ingeniería de sistemas.

3.3 Reingeniería de Procesos de Negocio (RPN)

3.3.1 Procesos de Negocio

Un proceso de negocios es un conjunto de tareas lógicamente relacionadas que se efectúan con objeto de obtener un determinado resultado de negocios. Dentro del proceso de negocios, se combinan personas, los

equipos, los recursos materiales y los procedimientos de negocios con objeto de producir un resultado concreto [Pre98].

Cada proceso de negocios posee un cliente bien definido, una persona o grupo que recibe el resultado. Además, los procesos de negocios cruzan los límites organizativos; es decir, requieren que participen distintos grupos de la organización en las tareas lógicamente relacionadas que definen el proceso.

La figura 3-2 ilustra la jerarquía en la que funciona la RPN. El negocio global se segmenta en un cierto número de sistemas de negocios o funciones de negocios. Cada sistema de negocios está compuesto por uno o más procesos de negocio, y cada proceso de negocio está definido por un conjunto de subprocesos.

Se puede aplicar RPN a cualquier nivel de jerarquía, pero a medida que se asciende dentro de la jerarquía los riesgos asociados crecen de forma dramática. Por esta razón, la mayor parte de los esfuerzos se centran en procesos o subprocesos individuales [Som02].

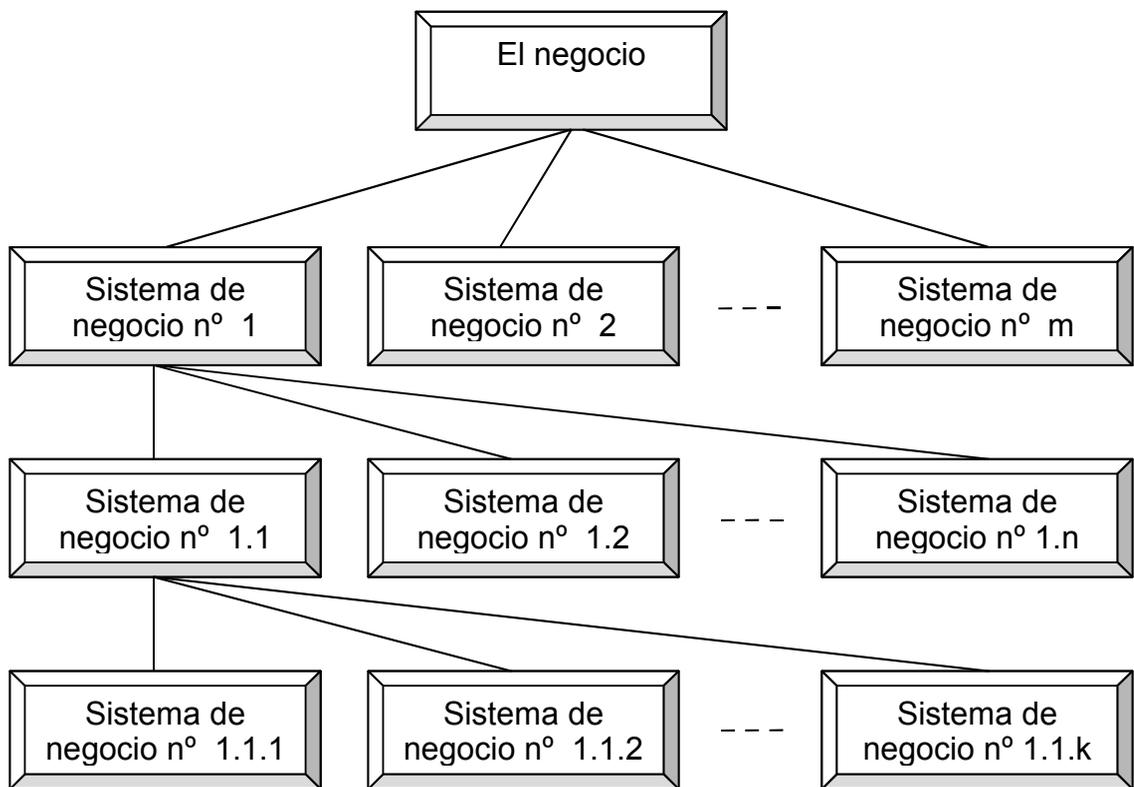


Figura 3-2. El negocio: una jerarquía de sistemas

3.3.2 Modelo RPN

Al igual que la mayoría de las actividades de ingeniería, la reingeniería de procesos de negocios es iterativa. Los objetivos de negocio, y los procesos que lo logran, deben de adaptarse a un entorno de negocios cambiante. Por esta razón, no existe un principio ni un fin, se trata de un proceso evolutivo. La figura 3-3 representa un modelo de reingeniería de procesos de negocios, en el que se definen seis actividades [Pre98]:

- *Definición del negocio.* Los objetivos del negocio se identifican en el contexto de cuatro controladores principales: reducción de costos, reducción de tiempos, mejora de calidad y desarrollo; y potenciación del personal. Los objetivos se pueden definir en el nivel de negocios o para un componente específico de ese negocio
- *Identificación de procesos.* Los procesos que sean críticos para alcanzar las metas estipuladas en la definición del negocio deben de ser identificados. A continuación se les pueden asignar prioridades por importancia, según su necesidad de cambio, o de cualquier otra forma que resulte adecuada para la actividad de reingeniería
- *Evaluación de procesos.* Los procesos existentes deben de analizarse y medirse exhaustivamente. Se identificarán las tareas de procesos; los costos y los tiempos consumidos por las tareas de proceso se anotarán cuidadosamente; se aislarán los problemas de calidad y rendimiento
- *Especificación y diseño de procesos.* Basándose en la información obtenida durante las tres primeras actividades de la RPN, se preparan casos prácticos para cada proceso que haya que rediseñar. En este contexto, los casos prácticos identifican un escenario que proporciona algún resultado a algún cliente. Mediante el uso de casos prácticos como especificación del proceso, se diseña para el proceso un nuevo conjunto de tareas

- *Prototipos.* Es preciso construir un prototipo del proceso de negocio rediseñado antes de integrarlo por completo en el negocio. Esta actividad comprueba el proceso para que sea posible efectuar refinamientos
- *Refinamiento y concreción.* Basándose en la realimentación procedente del prototipo, se refina el proceso de negocio y después se concreta en el seno de un sistema de negocio

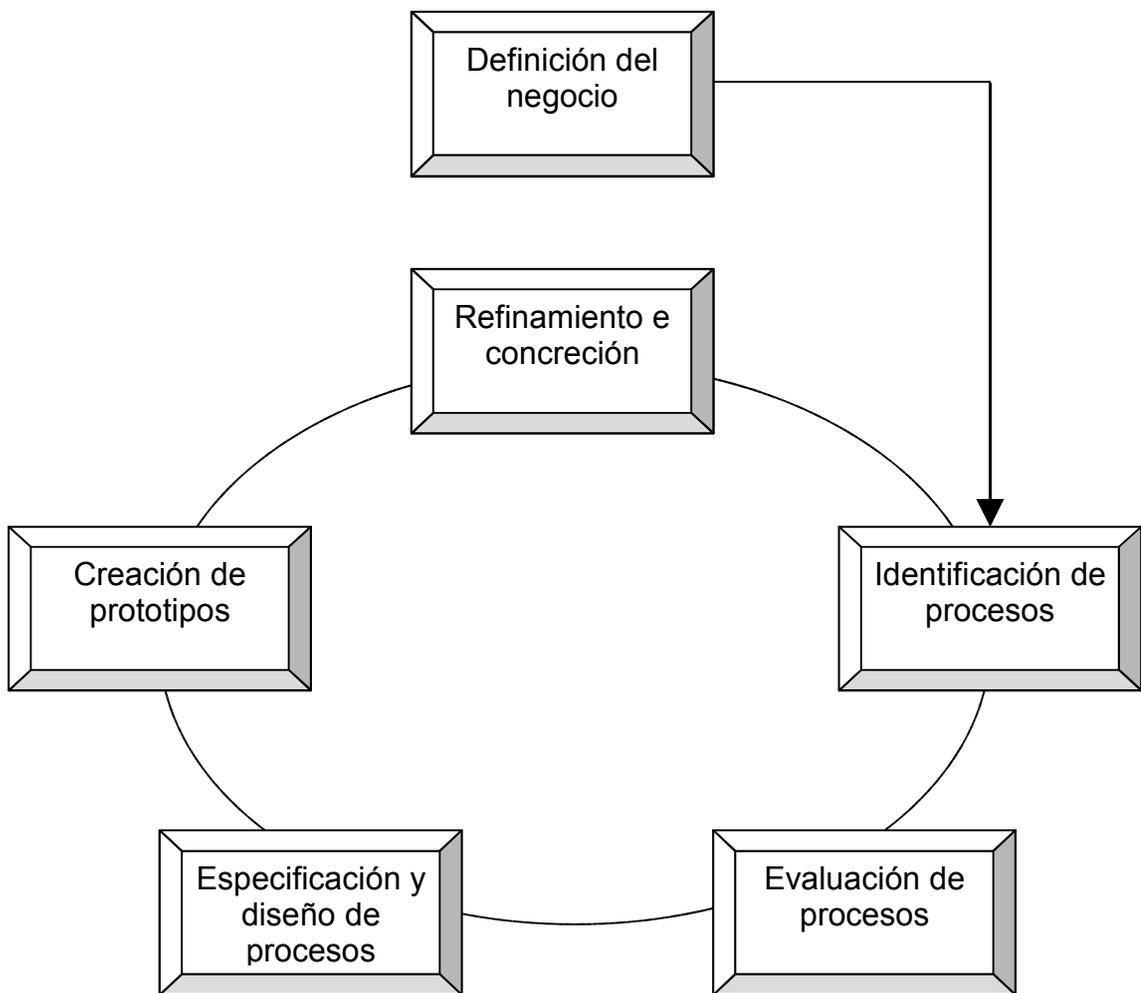


Figura 3-3. Un modelo de reingeniería de procesos de negocio

3.4 Application Service Provider (ASP)

3.4.1 Modelo ASP

ASP es una modalidad de proveer software en que la compra se reemplaza por la modalidad de arriendo del servicio de software y toda la infraestructura de hardware y servicios necesarios de configuración y mantención (ver figura 3-4) [URL 1].



Figura 3-4. Evolución del software

La tecnología ASP permite externalizar la gestión del software emplazándolo en los servidores de una empresa especializada. De este modo, el cliente no necesita comprar la licencia del software, sino que puede explotarlo pagando una cuota periódica previamente pactada, junto con el resto de condiciones del contrato. La ejecución del software arrendado se realiza a través de la red (ver figura 3-5) [URL 2].

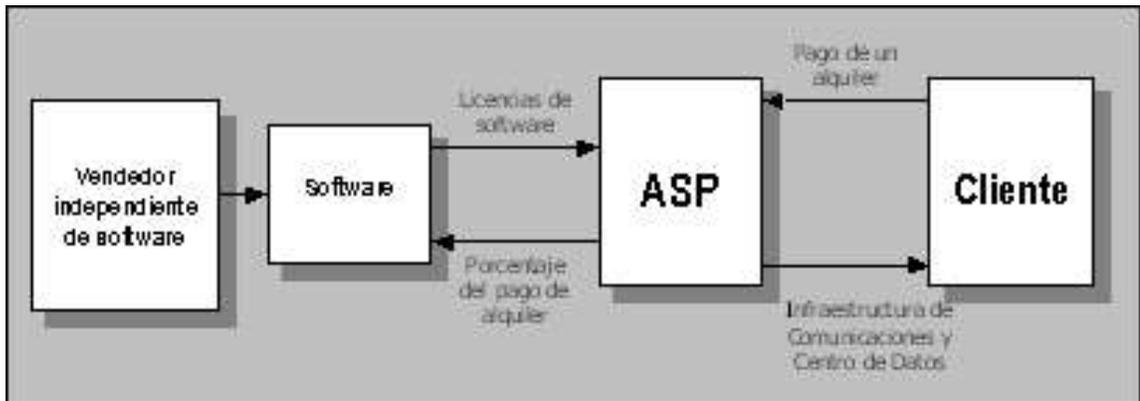


Figura 3-5. Modelo ASP

Debe hacerse especial énfasis en el término alquilado. El cliente final no es el dueño de la aplicación, ni debe aceptar las responsabilidades asociadas a su administración y mantenimiento. Usando un navegador de Internet (browser) u otro software liviano (generalmente basado en alguna tecnología cliente-servidor) los clientes acceden remotamente a los equipos que sirven las aplicaciones en cuestión.

3.4.2 Ventajas de un ASP

El modelo ASP cuenta con todas las ventajas derivadas de la externalización [URL 3]:

- *Rápida implementación.* No hace falta acomodar la infraestructura tecnológica de la empresa a los requerimientos del software en cuestión, puesto que éste se instala en los servidores del proveedor. El cliente se limita a ejecutarlo remotamente mediante una conexión a Internet. Técnicamente, la tecnología ASP permite ejecutar alguna de las aplicaciones de software más complejas del mercado mediante un simple computador personal (PC) y una conexión a Internet o una línea de comunicaciones dedicada. Por supuesto, que el acceso sea tecnológicamente sencillo no implica que se pueda implementar una aplicación ASP sin acometer previamente cambios

en la organización. Pero el ASP permite que la empresa se concentre en los cambios organizativos y de procedimiento; es decir, la preocupación de cómo financiar aiosamente la implementación de un software de última generación se sustituye por los cambios organizativos que se deben realizar para aprovechar al máximo el software ASP alquilado

- *Acceso inmediato a economías de escala.* La posibilidad de arrendar software, en lugar de comprarlo, permite trasladar todo el costo a la cuenta de explotación de forma inmediata e íntegra. Hasta el advenimiento de la tecnología ASP sólo las grandes organizaciones podían permitirse la compra de los paquetes de software de gestión más potentes, y por supuesto, más caros. Hacen falta enormes volúmenes de facturación para amortizar el costo fijo que representa la adquisición de la licencia de estas aplicaciones. Para las PYMES en cambio, la adquisición de una licencia resultaba a menudo una alternativa completamente inviable, dada la imposibilidad de llegar al punto muerto de facturación que permitiera rentabilizar la inversión. La tecnología ASP pone el software más potente del mercado al alcance de muchas PYMES, que gracias a la posibilidad de arrendar la aplicación, acceden directamente a costos marginales sin pasar por el tramo de costos fijos. En resumen, el ASP permite a las PYMES operar con costos similares a los de las grandes corporaciones
- *Eficiencia en el esfuerzo de la empresa.* Esta es precisamente la clave de que la externalización esté cobrando tanto auge en las economías más avanzadas. Externalizar permite que la empresa concentre sus esfuerzos en su núcleo de negocio; es decir, lo que de verdad dominan sus profesionales, aquello en lo que tiene ventajas competitivas. La contratación de software con un proveedor externo ASP evita que una empresa cuya actividad principal no sea la implementación y gestión de software, disperse sus esfuerzos de forma ineficiente. Gracias a este modelo tecnológico cada profesional se concentra en lo que mejor sabe hacer

- *Flexibilidad y escalabilidad.* El ASP permite reducir el desajuste entre el potencial de software y las verdaderas necesidades de la empresa. A menudo no se identifican las verdaderas necesidades de la empresa hasta que no se ha implementado la aplicación. La tradicional compra de aplicaciones obliga a las empresas a cargar con el desajuste: después de la compra de la licencia el software puede revelarse ineficiente o, por el contrario, excesivamente potente y por ende, demasiado caro. El arriendo de aplicaciones permite una dinámica constante y retroalimentada de ajuste entre los recursos por los que se paga y los que verdaderamente se usan
- *Ahorro y previsión en los costos de mantenimiento.* Cuando se adquiere una licencia de software por el método tradicional de compra, el gasto real de mantenimiento es incierto. La resolución de mayor incidencia puede acarrear gastos impredecibles. Sin embargo, todo contrato ASP incorpora el mantenimiento de las aplicaciones, las bases de datos y los equipos inherentes al servicio pactado. Este mantenimiento corre a cargo del proveedor del servicio. El cliente no ha de asumir gastos extraordinarios derivados de posibles contingencias. De este modo, el costo real del servicio no está sometido a incertidumbres
- *Eliminación del riesgo de obsolescencia.* Si no se actualiza regularmente, el software envejece y caduca con sorprendente rapidez. En un contexto de externalización ASP, el proveedor del servicio aconseja al cliente sobre mejora o actualización susceptible de ser incorporada y, de acuerdo con las necesidades del cliente, se encarga enteramente de su implementación

3.4.3 Infraestructura Tecnológica

Todo proveedor de ASP debe garantizar el uso de los mejores sistemas de seguridad para proteger los datos del cliente: copias de seguridad, cortafuegos, antivirus y sistemas de encriptación [URL 3].

Además de prevenir los ataques informáticos (hackers), un proveedor ASP debe garantizar la seguridad física de sus instalaciones, pues en ellas se alojará la aplicación del cliente junto a sus respectivas bases de datos. Esto incluye la vigilancia, los controles de acceso, los sistemas de climatización y antiincendios y, por supuesto, la existencia de grupos electrógenos que garanticen el funcionamiento ininterrumpido de los equipos.

3.5 Gestión de la Configuración (GC)

La gestión de la configuración (CM: Configuration Management) es la disciplina de la ingeniería de software que comprende las herramientas y técnicas (procesos o metodologías) que una organización utiliza para administrar las configuraciones de los componentes de software. Tiene como objetivo mantener la integridad de los componentes del producto de software, evaluar y controlar los cambios sobre ellos así como facilitar la visibilidad del producto a todo el equipo de proyecto [Feb02].

La definición más utilizada de GC es la de la IEEE que establece: “Gestión de configuración es la disciplina que abarca todo el ciclo de vida de la producción de software y productos asociados. Específicamente, requiere de la identificación de los componentes a controlar y la estructura del producto, controla todos los cambios sobre los elementos y garantiza mecanismos para auditar todas las acciones”.

La gestión de la configuración es el desarrollo y aplicación de estándares y procedimientos para administrar un producto evolutivo de sistemas. Es necesario administrar los sistemas evolutivos debido a que, cuando evolucionan, se crean varias versiones diferentes del software. Estas versiones contienen propuestas para el cambio, correcciones de fallas y adaptaciones para hardware y sistemas operativos diferentes. Pueden existir varias versiones bajo desarrollo y en uso al mismo tiempo. Por esto es necesario llevar un registro de los cambios implementados y de la manera en que éstos se han incluido en el software [Som02].

Los procedimientos de gestión de la configuración definen como registrar y procesar los cambios propuestos al sistema, como relacionar éstos con los componentes del sistema y los métodos utilizados para identificar las diversas versiones del sistema. Las herramientas de gestión de la configuración se utilizan para almacenar las versiones de los componentes del sistema, construir sistemas a partir de estos componentes y llevar un registro de las liberaciones de las versiones del sistema.

Existen muchas razones por las que los sistemas existen en diversas configuraciones. Éstas se producen para diferentes computadoras, para diferentes sistemas operativos, para incorporar funciones específicas del cliente, etc. (ver figura 3-6).

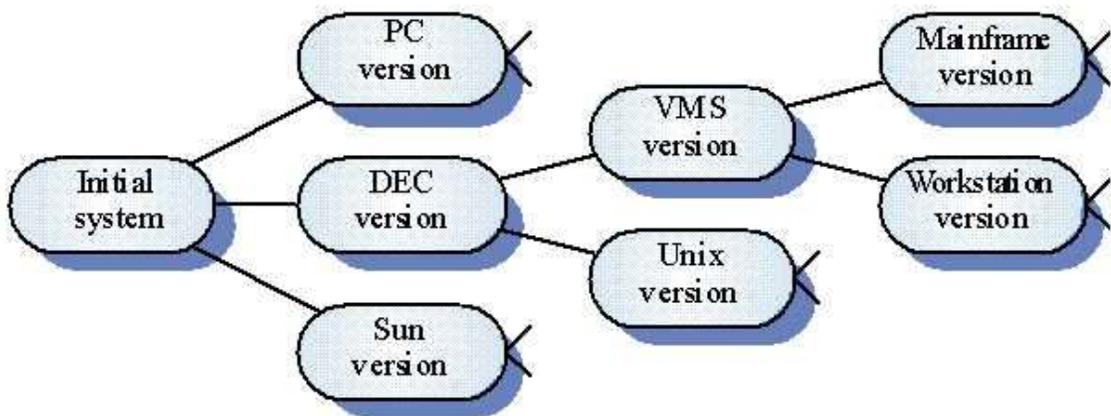


Figura 3-6. Familias de sistemas

El proceso de gestión de la configuración y la documentación asociada se basan en estándares. Los estándares externos se utilizan como base para estándares organizacionales detallados por lo que se ajustan a un entorno específico. No importa qué estándar se considere como punto de partida puesto que todos ellos incluyen procesos comparables. Tanto en los estándares de calidad ISO 9000 como en el modelo de madurez de la capacidad del SEI, las organizaciones deben definir y seguir los estándares formales de la GC para la certificación de calidad.

3.5.1 Planificación de la Gestión de la Configuración

Un plan de gestión de la configuración describe los estándares y procedimientos utilizados para la gestión de la configuración. El punto de inicio para desarrollar el plan es un conjunto de estándares generales de gestión de la configuración de toda compañía, adaptables a cada proyecto específico. El plan de la GC se organiza en varios capítulos que incluyen:

- La definición de las entidades a administrar y el esquema formal para identificar estas entidades
- Un enunciado de quien toma la responsabilidad de los procedimientos de gestión de la configuración y quien envía las entidades controladas al equipo de gestión de la configuración
- Las políticas de gestión de la configuración utilizadas para administrar el control de los cambios y las versiones
- Una descripción de los registros del proceso de gestión de la configuración a los que debe darse mantenimiento
- Una descripción de las herramientas a utilizar para la gestión de la configuración y el proceso a aplicar cuando se utilizan estas herramientas
- Una definición de la base de datos de la configuración que se utilizará para registrar la información de la configuración

Una parte importante del plan de la GC es la definición de responsabilidades. Define quien es el responsable de la entrega de cada documento o de cada componente de software para el aseguramiento de la calidad y la gestión de la configuración.

3.5.1.1 Identificación del Elemento de la Configuración

En el transcurso del desarrollo de sistemas de software se producen cientos de documentos, muchos de estos son documentos técnicos que están sujetos a cambios frecuentes y regulares. Otros documentos son memos, minutas, etc., los que son de interés para definir la historia del proyecto; sin embargo, no son necesarios para el futuro mantenimiento del sistema.

Durante el proceso de implantación de la gestión de la configuración, se decide exactamente qué elementos (o clases de elementos) se van a controlar. Los documentos o grupos de documentos relacionados del control de la configuración son documentos formales o elementos de la configuración. Normalmente, los planes del proyecto, las especificaciones, los diseños, los programas y los conjuntos de datos de prueba son elementos de la configuración; sin embargo, se controlan todos los documentos que son necesarios para el mantenimiento futuro del sistema.

El esquema de asignación de nombres a los documentos debe asignar un nombre único a todos los documentos de control de la configuración. Siempre existen relaciones entre estos documentos, las que conducen a un esquema de asignación de nombres jerárquicos (ver figura 3-7).

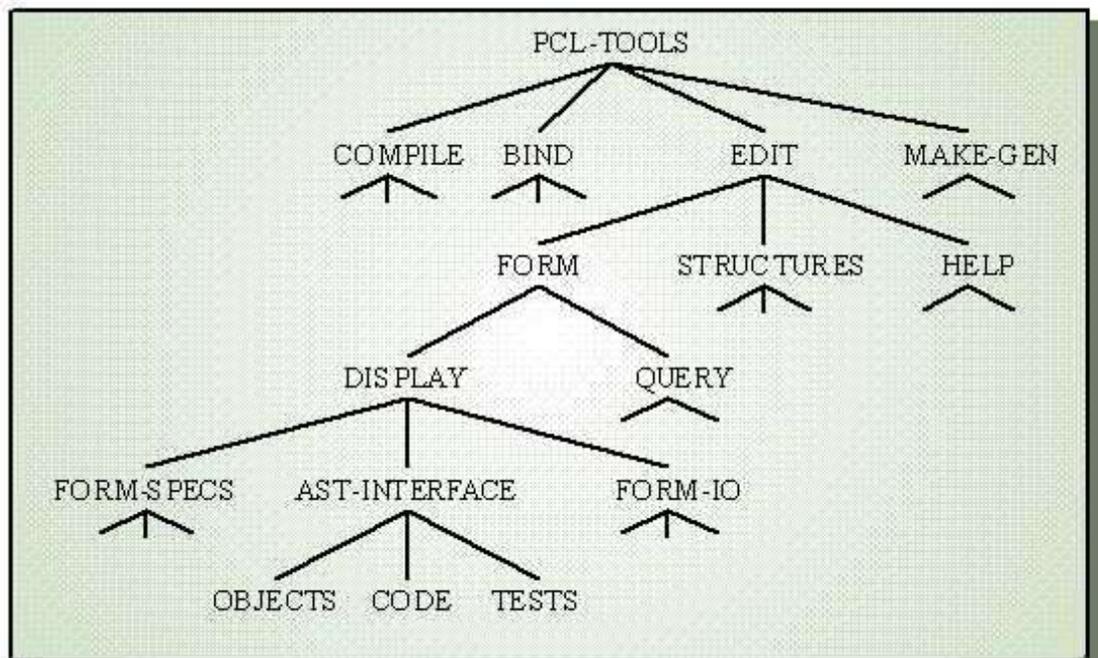


Figura 3-7. Jerarquía de la configuración

El problema con los esquemas de asignación de nombres de este tipo es que se basan en proyectos. Los identificadores asocian componentes a un proyecto particular lo que reduce las oportunidades de reutilización, las copias de componentes reutilizables normalmente se eliminan del esquema y se renombran de acuerdo a su dominio de aplicación. Podrían surgir otros problemas si el esquema de asignación de nombres a documentos se utiliza como una base directa para el diseño de una estructura de almacenamiento para los componentes administrados. También aparecen problemas al relacionar el esquema de asignación de nombres con el esquema de identificación utilizado por un sistema de administración de versiones.

3.5.1.2 Base de Datos de la Configuración

La base de datos de la configuración se utiliza para registrar toda la información relevante relacionada con las configuraciones. Sus funciones principales son ayudar a la evaluación del impacto de los cambios en el sistema y proveer información de la administración acerca del proceso de la GC. Además de definir el esquema de la base de datos de la configuración, como parte del proceso de implantación de la GC también se deben definir los procedimientos para registrar y recuperar la información del proyecto.

De forma ideal, la base de datos de la configuración se integra al sistema de administración de las versiones utilizado para almacenar y administrar los documentos formales del proyecto. Este enfoque, apoyado por algunas herramientas CASE integradas, hace posible vincular los cambios de forma directa con los documentos y componentes afectados por el cambio. Se da mantenimiento a los vínculos entre los documentos y el código del programa con el fin de que sea relativamente fácil encontrar todo lo que debe modificarse cuando se propone un cambio.

3.5.2 Administración del Cambio

El cambio es un hecho en la vida de los sistemas de software, las necesidades organizacionales y los requerimientos cambian durante el tiempo de vida de un sistema. Un proceso definido de la administración del cambio y las herramientas CASE asociadas asegura que estos cambios se registren y apliquen al sistema de forma costeable.

El proceso de administración del cambio se lleva a cabo cuando el software o la documentación asociada se ponen bajo el control del equipo de administración del cambio. Se inicia durante las pruebas del sistema o después de que el software se entrega a los clientes. Los procedimientos de administración del cambio se diseñan para asegurar que los costos y beneficios del cambio se realicen adecuadamente y que se hagan los cambios al sistema de una forma controlada.

La primera etapa del proceso de administración del cambio es completar un formulario de solicitud de cambio (CRF) en el cual el solicitante señala los cambios requeridos al sistema. Además de registrar los cambios requeridos, el CRF registra las recomendaciones pertinentes al cambio, los costos estimados del cambio y las fechas de cuando se solicita, prueba, implementa y valida el cambio. También incluye una sección donde el ingeniero de mantenimiento señala cómo implementar el cambio. Las solicitudes de cambio se registran en la base de datos de la configuración, por lo tanto, el equipo de la GC puede registrar el estado de las solicitudes de cambio y las solicitudes de cambio asociadas con los componentes de software específicos.

Una vez que se emitió el formulario, se analiza para comprobar que el cambio es válido. Algunas peticiones de cambio se deben a los malos entendimientos más que a las fallas del sistema; otras se refieren a fallas ya conocidas. Si el análisis descubre que el cambio solicitado es inválido, duplicado o ya ha sido considerado, el cambio se rechaza y esta razón se informa a la persona que emitió la solicitud del cambio.

Para cambios válidos, la siguiente etapa del proceso es evaluar y asignar costos al cambio. Esto involucra comprobar el impacto del cambio en el resto del sistema, realizar un análisis técnico de como implementar el cambio y

estimar el costo de realizar el cambio junto a la posibilidad de cambiar otros componentes del sistema para acomodar el cambio, esto último debe quedar registrado en el CRF. En este proceso de evaluación se utiliza la base de datos de la configuración donde se registran las interrelaciones de los componentes.

Cuando se aprueba un conjunto de cambios, el software se entrega al equipo de desarrollo o mantenimiento para su implementación. Una vez que esto se completa, el software revisado se vuelve a evaluar para comprobar que estos cambios se implementaron correctamente.

3.5.3 Administración de Versiones y Liberaciones

La administración de las versiones y liberaciones es el proceso de identificar y mantener los registros de las diversas versiones y liberaciones de un sistema. Los administradores de las versiones diseñan procedimientos para asegurar que las diversas versiones de un sistema se puedan recuperar cuando se requieran y que no se cambien de forma accidental. Además, trabajan en coordinación con los clientes para planear cuando deben distribuirse las nuevas versiones.

Una versión de un sistema es una instancia de un sistema que difiere, de alguna manera, de otras instancias. Las nuevas versiones de un sistema tienen diferente funcionalidad, desempeño o incorporan reparaciones de fallas del sistema. Algunas versiones son funcionalmente equivalentes pero diseñadas para diferentes configuraciones de hardware y/o software. Si sólo existen pequeñas diferencias entre las versiones, una de estas se denomina una variante de la otra.

Una liberación de un sistema es una versión que se distribuye a los clientes. Cada liberación incluye nueva funcionalidad o está concebida para diferentes plataformas de hardware. Siempre existen más versiones de un sistema que las liberaciones puesto que las versiones se crean dentro de una organización para el desarrollo interno o pruebas y nunca se entregan a los clientes.

En la actualidad, la administración de versiones se apoya en herramientas CASE, las que administran el almacenamiento de cada versión del sistema y controlan el acceso a los componentes del sistema. Se apoyan en el sistema para llevar a cabo las ediciones. Cuando los componentes se reintroducen en el sistema, se crea una nueva versión y el sistema de administración de versiones le asigna un nombre.

3.5.3.1 Identificación de Versiones

Dentro de un sistema de software existen cientos de componentes de software, cada uno de los cuales existe en muchas versiones diferentes. Los procedimientos para la administración de las versiones deben definir una forma no ambigua de identificar cada versión de los componentes. Las versiones específicas de los componentes después se pueden recuperar cuando se requieran para cambios posteriores.

Existen tres técnicas básicas utilizadas para la identificación de componentes:

1. *Numeración de las versiones.* Al componente se le asigna un número de versión explícito y único. Este es el esquema de identificación más usado (ver figura 3-8)
2. *Identificación basada en atributos.* Cada componente tiene un nombre (que no es único a lo largo de las versiones) y un conjunto asociado de atributos que difieren para cada versión del componente. Por lo tanto, los componentes se identifican por la combinación de un conjunto de nombres y atributos
3. *Identificación orientada al cambio.* Cada sistema se nombra con base en la identificación de atributos pero también se asocia a una o más solicitudes de cambio. La versión del sistema se identifica asociando el nombre con los cambios implementados en el componente

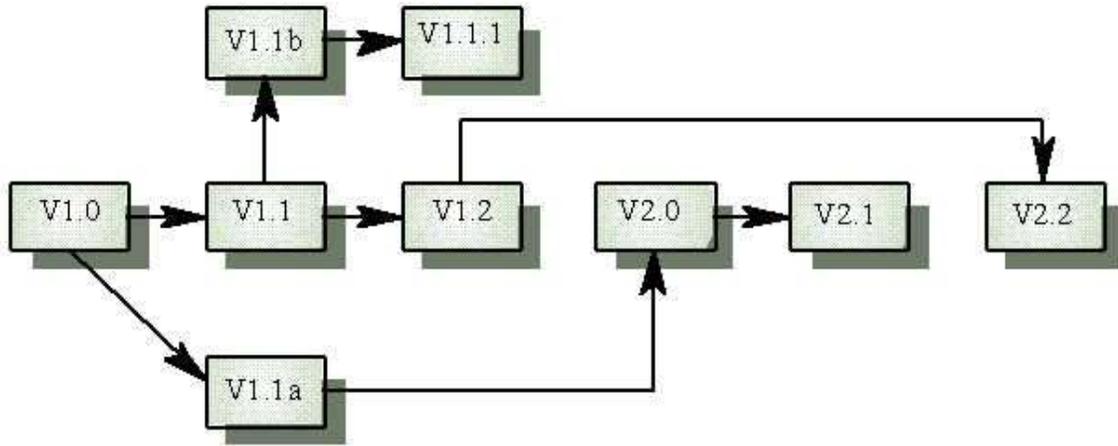


Figura 3-8. Estructura de derivaciones de las versiones

3.5.3.2 Administración de las Liberaciones

Una liberación del sistema es una versión del sistema que se distribuye a los clientes. Las versiones liberadas deben incorporar cambios que fueron forzados en el sistema por errores descubiertos por usuarios y/o cambios en el hardware, junto con una nueva funcionalidad del sistema.

Los administradores de las liberaciones no pueden asumir que los clientes siempre instalarán las nuevas versiones del sistema. Al cliente puede no agradaarle la nueva versión del sistema, pueden estar contentos con la versión actual del sistema ya que la nueva versión puede contener una funcionalidad no deseada. Por lo tanto, las nuevas versiones del sistema no pueden depender de la existencia de liberaciones previas.

Preparar y distribuir una liberación es un proceso costoso, si las liberaciones son muy frecuentes, los clientes pueden no actualizarse a las nuevas; por el contrario, si no son muy frecuentes, el mercado se puede perder puesto que los clientes consideran sistemas alternativos.

Cuando se produce la liberación de un sistema, debe estar documentada para asegurar que se puede reconstruir exactamente en el futuro.

Para documentar una liberación, se tienen que registrar las versiones específicas de los componentes del código fuente utilizados para crear el

código ejecutable. También se deben mantener copias del código fuente y ejecutable así como de todos los archivos de datos y de configuración. También se deben registrar las versiones del sistema operativo, las bibliotecas, los compiladores y otras herramientas utilizadas para construir el sistema. Estos pueden requerirse para construir exactamente el mismo sistema en alguna fecha posterior; en estos casos, las copias de las plataformas de software y las herramientas también se almacenan en un sistema de administración de versiones.

3.5.4 Herramientas CASE para la Gestión de la Configuración

Los procesos de gestión de la configuración por lo general están estandarizados e involucran la aplicación de procedimientos predefinidos, requieren de la administración cuidadosa de grandes cantidades de datos y la atención a los detalles es esencial. Cuando se construye un sistema a partir de las versiones de los componentes, un simple error en la gestión de la configuración puede implicar que el software no trabaje de forma adecuada; en consecuencia, las herramientas CASE de apoyo son esenciales para la gestión de la configuración.

3.5.4.1 Apoyo a la Administración de Cambios

Cada persona involucrada en el proceso de administración del cambio es responsable de alguna actividad. Una vez que completa esta actividad pasa los formularios y elementos de configuración asociados a otra persona. La naturaleza de procedimiento de este proceso significa que un cambio en el modelo del proceso se diseña e integra con un sistema de administración de versiones, entonces este modelo se interpreta para que los documentos correctos se pasen a la persona indicada en el momento justo.

Por lo tanto, las herramientas de administración del cambio proveen las siguientes características de apoyo al proceso:

1. Un editor de formularios que permite cambiar los formularios propuestos a crear y llenar
2. Un sistema de flujo de trabajo que permiten al equipo de la GC especificar las personas que deben procesar las solicitudes de petición de cambio y el orden del procesamiento. Este sistema también pasa de forma automática los formularios a las personas indicadas en el momento justo e informa a los miembros relevantes del equipo el progreso del cambio
3. Una base de datos de cambios que se utiliza para administrar todas las propuestas de cambio y que puede vincularse al sistema de administración de versiones. Por lo general, se proveen las características de consulta que permiten al equipo de la GC encontrar propuestas específicas de cambios

3.5.4.2 Apoyo a la Administración de Versiones

La administración de versiones involucra administrar grandes cantidades de información para asegurar que los cambios en el sistema se registren y controlen. Las herramientas de administración de versiones controlan un depósito de elementos de la configuración donde el contenido de ese depósito no cambia. Para trabajar sobre un elemento de la configuración, debe extraerse del depósito y colocarlo en un directorio de trabajo. Después que se completa el trabajo, se reintroduce en el depósito y automáticamente se crea una nueva versión.

Todos los sistemas de administración de versiones proveen un conjunto básico de capacidades comparables aunque algunos son más sofisticados que otros, ejemplos de estas capacidades son:

1. Identificación de la versión y la librería. A las versiones administradas se les asignan identificadores cuando se introducen en el sistema
2. Administración del almacenamiento. Para reducir el espacio de almacenamiento requerido por las diferentes versiones, los sistemas

de administración de versiones proveen las características de administración del almacenamiento donde las versiones se representan como un delta que encapsula las instrucciones requeridas para reconstruir la versión asociada del sistema. La figura 3-9 muestra cómo se aplican los deltas en reversa a la última versión de un sistema para reconstruir las versiones anteriores

3. Registro de la historia del cambio. Todos los cambios realizados al código de un sistema o componente se registran y listan. En algunos sistemas, estos cambios se utilizan para seleccionar una versión particular del sistema
4. Desarrollo independiente. Las diferentes versiones de un sistema se pueden desarrollar en paralelo y cada versión cambia de forma independiente. El sistema de administración de versiones mantiene un registro de los componentes que han sido extraídos para su edición y asegura que no interfieran los cambios que diferentes desarrolladores hayan hecho a los mismos componentes. Algunos sistemas sólo permiten que se extraiga una instancia de un componente para su edición, otros resuelven los conflictos potenciales cuando los componentes editados se reintroducen en el sistema

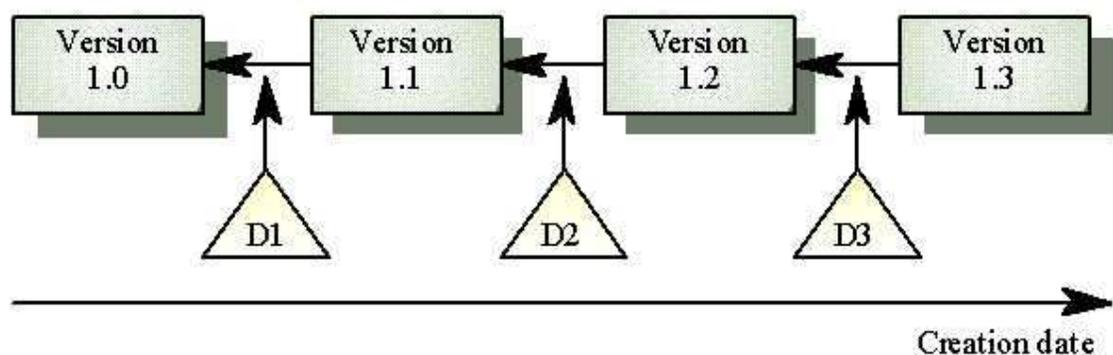


Figura 3-9. Descripción delta de las versiones

4 SITUACIÓN ACTUAL

4.1 Introducción

La solución del problema se aborda desde dos puntos distintos, uno guarda relación con las modificaciones realizadas al sistema propiamente tal y el otro con la modificación del proceso de desarrollo de software.

Las modificaciones al sistema se basan en la revisión de las funcionalidades del mismo, con el objetivo de eliminar y/o modificar aquellas que no son necesarias o que se pueden mejorar.

Las modificaciones al proceso de desarrollo se basan en la aplicación de gestión de la configuración, lo que permite llevar un control histórico y de versiones en el desarrollo de un producto software.

En este capítulo se describen tanto los procesos de desarrollo y mantenimiento del software como los actuales procesos de negocio del sistema. Para una mejor comprensión de estos procesos, se describen las principales funcionalidades de los módulos que componen este sistema así como sus flujos de datos respectivos.

4.2 Desarrollo y Mantenimiento del Software

En el proceso de desarrollo de software - si bien es cierto existen diversos estándares implementados, estos no permiten llevar un control exhaustivo del proceso. A continuación se describen las metodologías empleadas para desarrollar y mantener el software:

- Para todo desarrollo que se realiza sobre el sistema, ya sea una nueva funcionalidad o la modificación de una existente, se crea una ficha de atención. En ella se especifica claramente la persona solicitante así como las actividades que involucra el desarrollo. Una vez realizada la estimación de estas actividades, la ficha de atención es enviada al cliente para su aprobación
- Una vez aprobada la ficha de atención se procede a su realización en un ambiente de desarrollo. Recién cuando los programas son probados se pasan al ambiente de producción para finalmente enviárselos al cliente, el que los instala en su servidor de aplicaciones
- Los códigos fuentes se almacenan en un servidor de aplicaciones, en donde se definió una estructura de archivos que permite poseer tanto un ambiente de producción como uno de desarrollo
- Las versiones del sistema se separaron en distintos directorios dependiendo del cliente al que pertenecen, dentro de cada uno de éstos existe una estructura de archivos que permite mantener por separado el código fuente de cada módulo que compone el sistema

En resumen, el gran problema que se presenta es la falta de control tanto sobre los archivos fuentes de los programas como de los archivos ejecutables de los mismos. La carencia de este control permite por ejemplo, la pérdida involuntaria de archivos, la inexistencia de un control sobre las versiones lo que provoca programas con distintos nombres dependiendo de la versión, etc.

4.3 Procesos de Negocio del Sistema

A continuación se detalla el levantamiento realizado a los procesos de negocio del sistema, para determinar cómo mejorarlos y así optimizar el modelo. Esto conducirá a una modificación del sistema de acuerdo al modelo cascada y de esta manera lograr que estos se adecuen al esquema ASP.

4.3.1 Módulo de Admisión

Este módulo tiene como finalidad ingresar a los alumnos nuevos, ya sea por ingreso directo o por postulación a carrera (entrevista). Además, permite apoyar eficazmente la realización del proceso de selección de postulantes, desde que llegan a consultar por alguna carrera hasta que son seleccionados para ingresar a la institución.

En la figura 4-1 se observa un diagrama con el DFD (Diagrama de Flujo de Datos) del módulo de admisión.

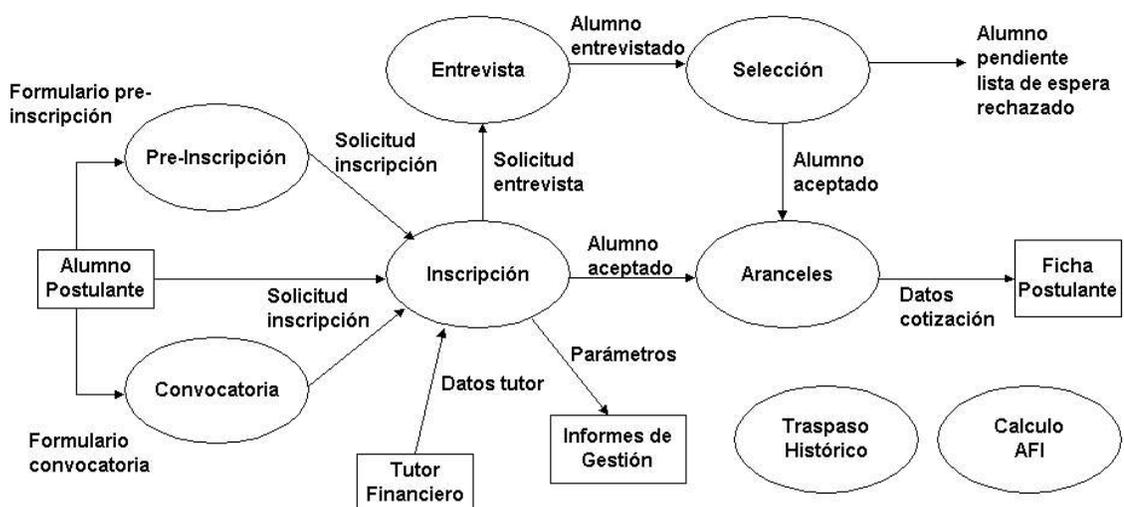


Figura 4-1. DFD del módulo de admisión.

4.3.1.1 Principales Funcionalidades

A continuación se detallan las principales funcionalidades del módulo de admisión.

- Ingreso de la información personal del postulante y de los antecedentes necesarios para postular
- Ingreso de las postulaciones a carreras
- Solicitud y autorización de beneficios (convenios y becas)
- Ingreso y gestión de horarios de entrevistas
- Asignación de entrevistas de acuerdo a horario y entrevistadores disponibles
- Entrevista evaluada en línea, donde el resultado se obtiene a partir de la ponderación asignada a cada pregunta
- Determinación del estado de admisión del postulante, en donde el entrevistador de acuerdo a la evaluación establecerá la categoría de aceptado o rechazado. Adicionalmente, el sistema considera el manejo de ingresos especiales y lista de espera
- Manejo de convocatoria para el próximo período académico, de tal forma de permitir la realización de una gestión de preventa
- Manejo de puntos de venta de donde se puede realizar admisión y matrícula
- Encuesta al postulante respecto de temas como la publicidad, referencias de la institución, etc.
- Control de pago de admisión
- Informes de gestión

4.3.1.2 Preinscripción

Esta función corresponde a una pantalla que tiene por objetivo inscribir a los alumnos que se enteraron del proceso de postulaciones por medio de un captador de alumnos.

4.3.1.3 Convocatoria

Esta función corresponde a una pantalla que tiene por objetivo inscribir a los alumnos que se enteraron del proceso de postulaciones por una de las siguientes alternativas:

- Visita a colegio
- Iniciativa propia
- Página Web
- Volantes
- Visita a empresa
- E-mail

4.3.1.4 Inscripción

Esta función corresponde a una pantalla que tiene por objetivo completar la información del alumno, su tutor financiero y su postulación. Además, existe la opción de inscribir directamente a un alumno por medio de esta pantalla, para finalmente generarle su ficha de postulación.

Por esta pantalla se inscribe tanto a los alumnos que postularon por preinscripción como por convocatoria.

4.3.1.5 Aranceles

Esta función corresponde a una pantalla que tiene por objetivo generar una cotización para la carrera en la que quedó aceptado el alumno.

Esta cotización, también llamada ficha de postulación, contiene los datos personales del alumno, de su tutor financiero y la información financiera del contrato (plan de pago pactado, cantidad de cuotas, valor de las cuotas, etc.). Esta misma ficha es posteriormente presentada en el módulo de caja para matricular al alumno.

4.3.1.6 Entrevista

Esta función corresponde a tres pantallas que tienen por objetivo realizar una entrevista vocacional al alumno. En la primera pantalla se definen los horarios de los entrevistadores, en la segunda se define la hora de la entrevista y la tercera corresponde a la entrevista propiamente tal.

4.3.1.7 Selección

Esta función corresponde a una pantalla que tiene por objetivo calcular el puntaje del alumno en función de su entrevista, notas de enseñanza media, puntaje P.S.U., puntaje prueba especial y de las ponderaciones ingresadas al sistema.

Dependiendo del puntaje final obtenido se puede aceptar, dejar pendiente, dejar en lista de espera o rechazar al alumno. Otra característica de esta pantalla es la posibilidad de permitir el ingreso especial de un alumno, ya sea por quintil, extranjero u otro ingreso especial.

4.3.1.8 Informes

A continuación se describen los principales reportes del módulo de admisión:

- **Resumen de admisión:** resumen comparativo del proceso de admisión con respecto a años anteriores, entrega la cantidad de alumnos inscritos y matriculados por carrera para los años comparados
- **Maestro de admisión:** resumen del proceso de admisión, entrega la cantidad de alumnos por carrera para los distintos estados de postulación
- **Convocatoria y cierre de negocios:** cantidad de visitas, informaciones y alumnos matriculados para todas las sedes. Además, entrega la razón entre las visitas y los matriculados (cierre de negocio)
- **Postulante estado:** listado de los alumnos diferenciados por carrera, jornada y estado

4.3.2 Módulo de Caja

Este módulo tiene dos grandes finalidades, la primera es contratar tanto a los alumnos nuevos como a los de continuidad y la segunda es realizar los pagos de conceptos. Además, soluciona toda la problemática relacionada al manejo de la situación financiera de los alumnos que cursan alguna carrera o poseen un contrato de prestación de servicios en la institución (curso, diplomados, etc.).

En la figura 4-2 se observa un diagrama con el DFD (Diagrama de Flujo de Datos) del módulo de caja.

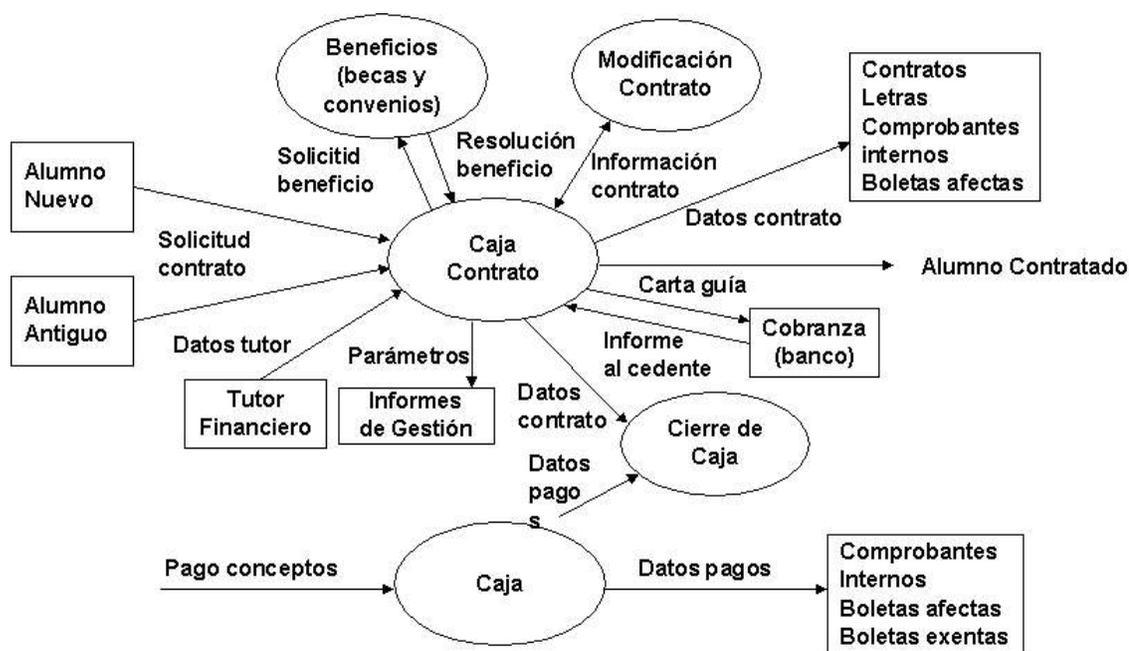


Figura 4-2. DFD del módulo de caja

4.3.2.1 Principales Funcionalidades

A continuación se detallan las principales funcionalidades del módulo de caja:

- Registro de contratos, pagos, matrículas, aranceles, boletas, comprobantes de ingreso, etc.
- Manejo de beneficios (becas y convenios)
- Solicitudes de beneficios y autorizaciones en niveles jerárquicos
- Concede múltiples formas de pago: efectivo, cheques, depósitos, vales vista, letras, cuponerías, etc.
- Maneja la cuenta corriente del alumno
- Ficha financiera del alumno
- Maneja la información de apoderados y aceptantes de la deuda

- Permite enviar y recibir documentos a instituciones financieras y de cobranza, en forma electrónica o por medio de formularios
- Registra y gestiona la cobranza de documentos, canje de documentos y repactación en letras y pagaré
- Informes de gestión

4.3.2.2 Caja Contrato

Esta función corresponde a tres pantallas que tienen por objetivo matricular un alumno. La primera pantalla es para matricular a los alumnos nuevos, la segunda es para matricular a los alumnos antiguos y la tercera es para definir la forma de pago del contrato.

4.3.2.3 Beneficios

Esta función corresponde a todo un submenú del módulo de caja y tiene por objetivo administrar todos los aspectos de un beneficio: empresas con convenios, convenios, becas, usuarios de autorización, solicitud de beneficios, autorización de beneficios y eliminación de beneficios.

4.3.2.4 Modificación de Contrato

Esta pantalla tiene como finalidad modificar un contrato vigente. Permite modificar tanto los documentos pendientes como los pagados. Además, posee la opción de anular un contrato por completo y generar uno nuevo inmediatamente así como la opción de aplicar beneficios.

4.3.2.5 Caja

Esta pantalla tiene como objetivo realizar los pagos de conceptos, los cuales pueden o no estar asociados a un contrato. Entre estos se destacan los siguientes: arancel, matrícula, repactación, multa, interés, pago de letras, certificados, venta estampillas, gastos de protesto, etc.

4.3.2.6 Cierre de Caja

Esta pantalla tiene como objetivo traspasar todos los movimientos de la caja a contabilidad. Una vez realizados el traspaso de los pagos estos no se pueden modificar por ninguna pantalla del sistema. Además, esta pantalla incluye la opción de anular los cierres de caja.

4.3.2.7 Informes

A continuación se describen los principales reportes del módulo de caja:

- Proyección de ingresos: resumen de los dineros documentados (letra, cheque y vale vista) por cada sede. La información se agrupa por mes y por la ubicación de los documentos (institución o banco)
- Resumen de morosos: comparación por carrera entre lo documentado y lo moroso. La información se agrupa en cantidad de documentos, cantidad de alumnos y dinero
- Estadística de alumnos nuevos matriculados: resumen de los alumnos nuevos matriculados por sede. Entrega información de las carreras ofrecidas, sus vacantes, cantidad de alumnos matriculados y la razón entre estos dos últimos (cumplimiento)

- Estadística de ingresos: resumen de la cuenta corriente (conceptos y pagos) por sede para alumnos nuevos y antiguos. Además, posee la opción de generar un detalle del mismo

4.3.3 Módulo Curricular

Este módulo administra todo el ciclo académico del alumno desde que se contrata por vez inicial hasta el momento de su titulación, controlando todo tipo de situaciones intermedias. Además, permite realizar el control y la mantención de las actividades del personal docente.

En la figura 4-3 se observa un diagrama con el DFD (Diagrama de Flujo de Datos) del módulo curricular.

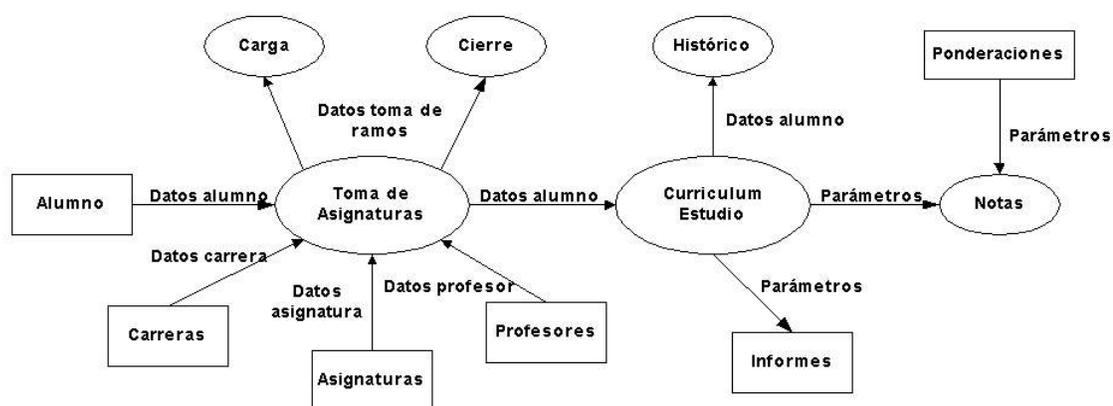


Figura 4-3. DFD del módulo curricular

4.3.3.1 Principales Funcionalidades

A continuación se detallan las principales funcionalidades del módulo curricular:

- Ingreso de alumnos nuevos, antiguos y reingresados con o sin carga académica asignada
- Creación de carga y precarga académica, la que puede ser asignada o definida por los alumnos
- Manejo de la historia de las carreras y asignaturas cursadas por cada alumno
- Generación de boletines de carga académica, notas y alumnos con situaciones especiales
- Ingreso de notas de evaluaciones tales como ejercicios, cátedras y pruebas finales
- Cálculo de notas de presentación, exámenes y exámenes de repetición considerando promedios automáticos y estado final de asignaturas
- Manejo de la aprobación de las asignaturas y carreras mediante las ponderaciones asociadas a las evaluaciones que se desarrollan en ellas
- Ingreso y gestión de la información de profesores referente a sus títulos, grados, postgrados, experiencia, etc.
- Registro de cursos dictados y asistencia a clases programadas por parte de los docentes de la institución
- Registro de solicitudes y resoluciones
- Manejo de homologaciones, cambios de carrera, convalidaciones y cambios de plan
- Gestión global de horarios, salas y docentes para las asignaturas
- Manejo de carreras modulares, las que incorporan los nuevos conceptos de educación por competencias
- Generación de informes de gestión tales como acta de examen, concentración de notas, nómina de alumnos por asignatura, carga académica, etc.

4.3.3.2 Usuarios

Es esta pantalla se administran los usuarios y sus contraseñas, tanto del sistema como de la base de datos (estos últimos se muestran encriptados); además, en esta pantalla se asocian los usuarios con los distintos módulos.

4.3.3.3 Carreras

En esta pantalla se administran las carreras, entre las características que se pueden manejar se encuentran las siguientes: código, nombre, jornada, régimen, estado, etc.

4.3.3.4 Asignaturas

En esta pantalla se administran las asignaturas según la carrera a la que pertenecen, entre las características que se pueden manejar están las siguientes: código, nombre, periodicidad, vigencia, créditos, horas, etc.

4.3.3.5 Currículum de estudio

En esta pantalla se administran los planes de las carreras; es decir, la duración en semestres, cantidad de niveles, asignaturas comunes, asignaturas electivas, prerrequisitos, etc.

4.3.3.6 Toma de asignaturas

En esta pantalla se administran los ramos del alumno para un período determinado, las asignaturas se muestran en distintos colores dependiendo de su estado (aprobada, reprobada, inscrita, pendiente, no cursada, etc.)

4.3.3.7 Ponderaciones

En esta pantalla se administran las ponderaciones por asignaturas, por ejemplo: ponderaciones de ejercicios, ponderaciones de cátedras, porcentaje de asistencia, nota de presentación, nota de eximición, etc.

4.3.3.8 Notas

En esta pantalla se administran las notas de los alumnos por asignaturas y sus asistencias; además se controlan los cierres de exámenes y sus actas respectivas.

4.3.3.9 Solicitudes

Esta funcionalidad cuenta con una gran cantidad de pantallas las que comprenden un menú propio. Posee pantallas para el ingreso de las solicitudes, las que pueden ser: cambios de sede, carrera, jornada o currículum, convalidación, reincorporación, etc.

Entre las pantallas que incorpora también se encuentran las de autorización de solicitudes así como las de eliminación de las mismas. Para estos dos casos se emplea un mantenedor de usuarios, para filtrar los que tengan permisos para realizar estas acciones.

4.3.3.10 Carga

Esta funcionalidad comprende todo un menú, en ella se encuentran las pantallas que permiten: imprimir los boletines de carga, precarga y carga masiva, realizar la carga, precarga y su simulación, administrar los paquetes fijos, el calendario académico, etc.

4.3.3.11 Cierres

Esta funcionalidad comprende las pantallas de cierre de asignaturas, de módulos, semestral y anual.

4.3.3.12 Profesores

Esta funcionalidad permite administrar al personal docente, sus datos personales, antecedentes laborales y académicos (grados, títulos y post títulos). Además, se pueden supervisar tanto sus actividades como sus asistencias.

4.3.3.13 Histórico

Esta pantalla permite administrar el historial de los alumnos, asignaturas, estados, notas, asistencias, fechas, cierres, etc.

4.3.3.14 Informes

A continuación se describen los principales reportes del módulo curricular:

- Certificados. Entre los reportes de esta sección se destacan los siguientes: certificado de alumno regular, concentración de notas, certificado de egreso y certificado de asignaturas
- Deserciones. Entre los reportes de este tipo se encuentran los reportes de deserciones académicas, deserciones no académicas y deserciones del periodo
- Alumnos. Entre estos reportes se encuentran los de alumnos egresados (anual e histórico), listado de alumnos por carrera y estadística de alumnos (regulares, con carga académica, etc.)

5 SOLUCIÓN

5.1 Introducción

Como se mencionó en el capítulo anterior, la solución planteada en este proyecto se divide en dos partes: modificaciones al sistema y modificaciones al proceso de desarrollo y mantenimiento del software.

Las modificaciones al sistema surgen de la necesidad de estandarizar y parametrizar el sistema académico de TGA. Para lograr este objetivo se realizó un levantamiento de los procesos de negocios de cada módulo, con la finalidad de aplicar una reingeniería a éstos. De esta manera se obtiene como resultado las funcionalidades del sistema que necesitan ser eliminadas o modificadas.

En el caso de las modificaciones al sistema, además de revisar las funcionalidades del sistema para eliminar y/o modificar aquellas que no son necesarias o que se pueden mejorar, se trabajó en la apariencia de las pantallas para dar una mayor uniformidad y de esta manera mejorar la imagen del sistema.

Las modificaciones al proceso de desarrollo y mantención del software surgen de la necesidad de llevar el sistema académico desde un esquema de producto a uno de servicio. En este ámbito se implementó una metodología de gestión de configuración del software con la ayuda de la herramienta Oracle9i SCM y se definieron procedimientos de paso a producción y mesa de ayuda. Todo esto con la finalidad de definir una arquitectura organizacional que permita a TGA convertirse en un ASP.

Para la realización de las modificaciones al proceso de desarrollo y mantención del software se aprovechó la condición de la empresa como partner de Oracle ya que se utilizó el software Oracle9i SCM, herramienta incluida en la distribución de Oracle Developer Suite 9i.

5.2 Modificaciones al Sistema

La solución pasa básicamente por aplicar la reingeniería de procesos al sistema, con esto se pretende optimizar cada módulo del sistema, eliminado y/o modificando las funcionalidades que no aporten ningún valor o lo hagan de una manera reducida.

Para llevar a cabo esta tarea se tomaron en cuenta las apreciaciones de los clientes. Esta retroalimentación permitió conocer las expectativas que tienen estos respecto al sistema, lo que dio como resultado una pauta de lo que debía hacer el sistema y como lo debía hacer. Este proceso junto a un completo estudio del sistema, dieron origen a las modificaciones realizadas a la aplicación.

Como se dijo anteriormente, la reingeniería de procesos de negocios involucra la realización de seis tareas principales, a continuación se describe lo realizado en estas tareas en el marco de la reingeniería al sistema académico.

- *Definición del negocio.* Los objetivos del negocio se identifican en el contexto de cuatro controladores principales: reducción de costos, reducción de tiempos, mejora de calidad y desarrollo; y potenciación del personal. Aunque de alguna manera el resultado final ayudó en la reducción de tiempo, el principal objetivo de esta reingeniería es la mejora de calidad y desarrollo del sistema académico
- *Identificación de procesos.* Para cada módulo se identificaron los procesos críticos, no existió la necesidad de asignarles prioridad por importancia ya que se trabajó en orden al flujo del sistema; es decir, se empezó por el módulo de admisión, se continuó con el de caja para finalizar con el curricular
- *Evaluación de procesos.* Para cada proceso analizado en la fase anterior se identificaron las tareas que estos involucran
- *Especificación y diseño de procesos.* Esta etapa comprende la especificación y diseño del conjunto de tareas resultantes para cada proceso, esto involucra tanto la eliminación y/o modificación de las antiguas tareas como la creación de las nuevas
- *Prototipos.* Esta etapa corresponde a la realización de las modificaciones en un ambiente de desarrollo, en donde se realizan todas las pruebas necesarias para eliminar los posibles errores
- *Refinamiento y concreción.* En esta última etapa se refinan todas las observaciones surgidas de la etapa anterior y finalmente, se concreta la implantación del proceso en el ambiente de producción

A continuación se presentarán las modificaciones ordenadas por cada módulo del sistema.

5.2.1 Módulo Admisión

Las modificaciones realizadas en este módulo afectaron a las siguientes funcionalidades y/o pantallas.

5.2.1.1 Convocatoria

Esta funcionalidad originalmente correspondía a una sola pantalla. Ahora corresponde a un submenú con tres pantallas, de las cuales dos pertenecen a pantallas de generación de informe y la restante al ingreso de alumnos por convocatoria propiamente tal.

- *Ingreso por convocatoria:* La pantalla es básicamente la misma que la original sólo que se le agregaron los campos pertenecientes al ítem de antecedentes educacionales; estos son: colegio de egreso (4º medio), año de egreso, modalidad de enseñanza media y régimen de estudio
- *Informe por institución:* Este reporte es nuevo y permite obtener un listado de los alumnos ingresados por convocatoria para el año académico deseado dentro de un rango de fechas seleccionado. La pantalla genera el listado en dos formatos distintos, el primero corresponde a un resumen, en donde sólo se indica la cantidad de alumnos por sede, agrupados por institución. El segundo pertenece a un detalle, en donde se genera un listado con los siguientes datos de los alumnos: RUT, teléfono, nombre, carrera y colegio. Para este último caso también se agrupan los resultados por sede e institución
- *Informe por sede:* Es idéntico al anterior con la condición que filtra los resultados obtenidos según la sede de conexión

5.2.1.2 Inscripción

Esta funcionalidad se mantuvo en una pantalla pero se le aplicaron varias modificaciones, tanto de presentación visual como de contenido. En el ámbito de la forma y organización de la pantalla, se eliminó el texto de encabezado que describía brevemente las instrucciones de utilización de la pantalla y se invirtió el orden de presentación entre los ítem Carreras a Postular y Datos Postulación, quedando el primero sobre el segundo en la pantalla.

En relación al contenido, se eliminaron los ítem de datos laborales y antecedentes educacionales, este último es el mismo que se traspasó a la pantalla de ingreso por convocatoria.

Otra modificación que se hizo en esta pantalla fue la de sacar el botón de tutor financiero, funcionalidad a la que ahora se accede directamente desde el menú.

5.2.1.3 Tutor Financiero

Es la misma pantalla que se accedía desde la pantalla de inscripción, pero ahora se colgó directamente desde el menú.

5.2.1.4 Asignación de entrevistas

Esta pantalla si bien es cierto no es nueva, se modificó para incorporarla en este módulo.

5.2.1.5 Entrevistas

Como todas las pantallas que hacen relación a la funcionalidad de entrevista, esta ya existía pero se modificó para su correcta integración al módulo. La modificación consistió en eliminar algunas columnas de la pantalla de selección para entrevistas.

5.2.1.6 Ponderaciones

En esta funcionalidad, además de eliminar algunas columnas de la pantalla selección de ponderaciones, se modificó la pantalla de su detalle para actualizar, entre otras cosas, el formato de la PAA por la PSU.

5.2.1.7 Reportes

Los reportes que se modificaron fueron los siguientes:

- *Encuesta de medios.* En este reporte se ampliaron la cantidad de ítem desplegados, así como también se incluyó la opción de generar un archivo en texto plano exportable a Excel. Específicamente se modificó el reporte y el procedimiento almacenado que lo genera para poder desplegar los ítem contenidos en el campo general (línea 800, contacto telefónico, volante, tríptico, vía pública, visita colegio, sitio web)
- *Encuesta de medios todas las sedes.* Al igual que el reporte anterior se ampliaron la cantidad de ítem desplegados así como la opción de generar un archivo de texto. Además se modificó el gráfico que genera este reporte
- *Gráfico matrículas diarias por sede de conexión.* Se modificó este gráfico para que despliegue la cantidad de alumnos matriculados por día
- *Gráfico matrículas diarias consolidado.* Al igual que el gráfico anterior, se modificó para que despliegue la cantidad de alumnos matriculados por día

5.2.2 Módulo Caja

Las modificaciones realizadas en este módulo afectaron a las siguientes funcionalidades y/o pantallas.

5.2.2.1 Alumnos Nuevos

Esta fue una de las pantallas que sufrió la mayor cantidad de modificaciones. Se separó el ítem compromisos de pagos en 2 secciones, un encabezado para los conceptos y un detalle para los documentos, esto permitió eliminar la funcionalidad de pago actual. Además, se agregaron las opciones de cambio de carrera, tutor financiero, impresión de boletas e impresión de letras.

5.2.2.2 Alumnos Antiguos

En esta pantalla se eliminaron los datos relacionados a la Ubicación Santiago. Además, se eliminó el botón de Matriculados.

5.2.2.3 Tutor Financiero

En esta pantalla se agregó el botón de Copiar Tutor y se eliminaron los botones de Matrícula Nueva, Alumnos, Ver Matriculados y Anula.

5.2.2.4 Reimpresión de documentos

Esta funcionalidad es nueva y permite la reimpresión de documentos tales como contratos y boletas.

5.2.2.5 Activación de Ficha

Esta pantalla permite deshacer un contrato sólo en el caso que se produzca un error en la impresión de sus letras o boletas.

5.2.2.6 Envío de letras Banco BCI

En esta pantalla se creó tanto el proceso para generar el archivo de texto con el formato requerido por el banco, como su reporte respectivo.

5.2.2.7 Envío de letras Banco Santiago

En esta pantalla se creó tanto el proceso para generar el archivo de texto con el formato requerido por el banco, como su reporte respectivo.

5.2.2.8 Reportes

La mayoría de las pantallas que invocan los reportes sufrieron modificaciones para estandarizarlas. Esto se refiere al orden en que se despliegan los datos solicitados, longitud y formato de estos campos; y sus validaciones correspondientes. Para no alargar esta sección, sólo se hará referencia a las modificaciones que afectan a los reportes propiamente tal y se omitirán aquellas que corresponden a las pantallas que invocan estos reportes.

Los reportes que se modificaron fueron los siguientes:

- *Estadísticas de documentos.* En este reporte se configuró la página como landscape por defecto y se solucionó el problema del encabezado, ya que este sólo aparecía en la primera hoja
- *Resumen de documentos por fecha de vencimiento.* En este reporte se configuró la página como landscape por defecto y se corrigieron los campos de fecha y año académico, que estaban corridos
- *Resumen de documentos por fecha de pago.* En este reporte se configuró la página como landscape por defecto y se corrigieron los campos de fecha y año académico, que estaban corridos
- *Resumen de letras por carrera y por fecha de vencimiento.* En este reporte se configuró la página como landscape por defecto
- *Resumen de letras por carrera y por fecha de pago.* En este reporte se configuró la página como landscape por defecto
- *Morosos para finanzas.* En este reporte se amplió el campo del tutor
- *Morosos para finanzas.* En este reporte se amplió el campo del tutor
- *Resumen de morosos.* En este reporte se reemplazaron con ceros los caracteres nulos en el campo de recargos
- *Informe de alumnos nuevos matriculados.* En este reporte se configuró la página como landscape por defecto y se modificó la consulta que lo genera, haciendo uso de funciones
- *Informe de alumnos antiguos matriculados.* En este reporte se configuró la página como landscape por defecto y se modificó la consulta que lo genera, haciendo uso de funciones
- *Libro de letras.* Se modificó tanto el reporte como el procedimiento almacenado que lo genera ya que la información que entregaba no era del todo correcta
- *Estadística de ingresos por institución.* En este reporte se reemplazó el carácter coma (,) por punto (.) como separador de miles

- *Estadística de ingresos por sede.* En este reporte se agruparon los resultados por tipo de alumnos (nuevos y antiguos)

Nota: landscape se refiere a la orientación de la hoja. En muchas aplicaciones se utiliza este termino para referirse a una página horizontal. Su traducción al español es apaisado, que según la RAE (Real Academia de la Lengua Española) tiene el siguiente significado: dicho de una figura o de un objeto de forma rectangular cuya base es mayor que su altura.

5.2.3 Módulo Curricular

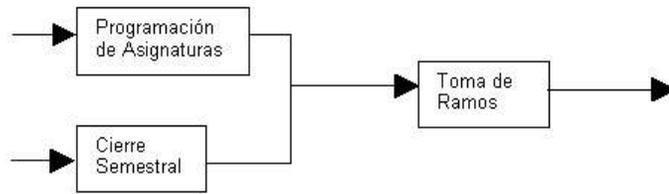
Este módulo fue el que menos modificaciones sufrió ya que en el transcurso del desarrollo de esta tesis, estaba siendo efecto de numerosas transformaciones por parte del equipo de desarrollo

Uno de los trabajos realizado sobre este módulo fue el de participar en el rediseño del modelo de proceso operativo de la toma de ramos, pero cuyo desarrollo no formó parte de este proyecto.

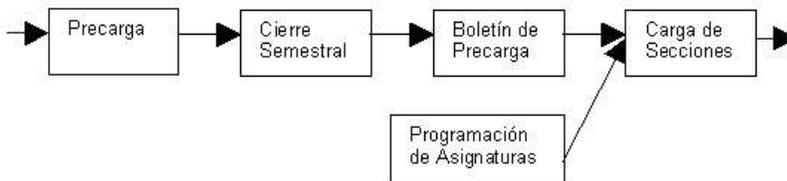
Las tres funcionalidades afectadas por el rediseño antes mencionado corresponden a la toma de ramos por parte de los alumnos, la precarga de asignaturas con cierre y la precarga de asignaturas sin cierre.

En la figura 5-1 se pueden apreciar el nuevo diseño del modelo de proceso operativo de la toma de ramos.

Toma de ramos por alumnos:



Precarga con cierre:



Precarga sin cierre:

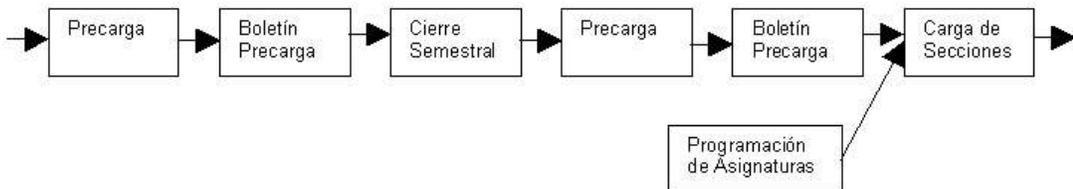


Figura 5-1. Modelo de proceso operativo de la toma de ramos

Otra de las tareas realizadas en este módulo correspondió a la adaptación del sistema para el soporte de las carreras modulares. Esta adaptación correspondió tanto a modificaciones en funcionalidades de este módulo como de tablas en la base de datos, para soportar este nuevo formato educacional denominado competencia laboral.

5.3 Modificaciones al proceso de desarrollo

A continuación se describen las modificaciones realizadas en el ámbito del proceso de desarrollo de software. Esto involucra la implementación de: Oracle SCM como herramienta de ayuda a la gestión de la configuración, pasos de producción y mesa de ayuda.

5.3.1 Oracle SCM

La Gestión de Configuración de Software (SCM) ayuda al desarrollo, controlando quién tiene acceso a las versiones de código y maneja las versiones particulares que van juntas para crear una liberación. La gestión de dependencia entre versiones y productos también puede mejorar la calidad y la fiabilidad de software desarrollado. Oracle SCM proporciona estas capacidades SCM como un componente integral de Oracle Internet Developer Suite [URL 7].

La educación es esencial para asegurar que los equipos de desarrollo tomen la ventaja máxima de SCM. Esto incluye un entendimiento claro de las herramientas y los beneficios de utilización de ellos.

Permitir que los desarrolladores de software trabajen juntos en un ambiente de desarrollo seguro y cuidadosamente manejado requiere:

- un repositorio para el almacenamiento de objetos de desarrollo
- mecanismos para manejar los objetos de versiones
- controlar el acceso al repositorio
- políticas que impongan restricciones a las operaciones sobre versiones

En resumen, Oracle SCM es un sistema de gestión de la configuración del software. Anteriormente esto estaba disponible sólo con el Oracle Designer, que es usado para modelar y generar datos y aplicaciones. La disponibilidad de Oracle SCM como un componente separado ayuda a los desarrolladores

quienes necesitan una capacidad SCM integrada para la administración del control de versiones y código fuera del ambiente de Oracle Designer. Como un componente integral de Oracle Developer Suite, Oracle SCM maneja datos estructurados y no estructurados a través del ciclo de vida de desarrollo. Esto apoya a proyectos de desarrollo de software de múltiples desarrolladores y de cualquier tamaño y complejidad.

Los componentes claves de Oracle SCM están articulados alrededor del repositorio de meta datos almacenados en una base de datos Oracle y varias herramientas que están disponibles para tener acceso sobre los meta datos como las siguientes:

- *Repository Object Navigator*. Proporciona una navegación gráfica de fácil utilización a través del almacenamiento del repositorio
- *Dependency Manager*. Mantiene las relaciones de dependencia entre los objetos, por ejemplo qué datos son usados por qué código
- *Matrix Diagrammer*. Para el análisis de impacto de los cambios sobre los objetos
- *Repository Administration Library*. Herramientas para mantener y administrar el Oracle SCM
- *Repository Reports*. Proporcionan una variedad de informes sobre el Oracle SCM y su uso.

5.3.1.1 Almacenamiento de datos

Oracle SCM actúa como un área de almacenamiento central para datos estructurados y no estructurados. Los datos estructurados son los meta datos usados como componentes básicos de aplicación. Los desarrolladores de aplicaciones usan herramientas de modelado como Oracle Designer para crear estas definiciones de objeto en el repositorio. Los datos no estructurados consisten en todos tipos de archivos y sistemas de archivos por ejemplo Oracle Form (archivos .FMB), documentos de texto, archivos XML, esquemas DTD's, hojas de estilos XSL, scripts SQL, programas e imágenes Java. Archivos

individuales o sistemas de archivos completos pueden ser copiados en el repositorio (subidos), y entonces pueden ser utilizados por otras herramientas (por ejemplo, editores de textos o procesadores de texto). Los archivos creados en el repositorio, o subidos a él, también pueden ser copiados en cualquier sistema de archivos (descargado).

En esta descripción, se usa el término genérico 'objeto' tanto para referirse a objetos estructurados como a objetos de sistema de archivo. Dentro del repositorio, los objetos son almacenados en contenedores.

Los objetos pueden ser manipulados vía API de Java o PL/SQL o vía las herramientas que utilizan estas APIs, tales como la Interfaz de Línea de Comando o el Navegante de Objeto del Repositorio.

5.3.1.2 Contenedores

Los contenedores son el medio de organizar datos de una manera lógica, y son similares a directorios en un sistema de archivos. Cada objeto pertenece a un contenedor y tiene un conjunto de propiedades de auditoría que registran detalles como la fecha cuando el objeto ha sido creado y quien lo creó (ver figura 5-2).

Hay dos tipos de contenedores: carpetas y sistemas de aplicación. Ambos pueden mantener instancias de cualquier tipo de objetos del repositorio. Las carpetas están disponibles si se instala sólo el repositorio principal (por ejemplo cuando se usa el repositorio como un sistema de control de código fuente). Los sistemas de aplicación están disponibles (así como las carpetas) si se instala el modelo Oracle Designer en el repositorio. Los contenedores también son usados para controlar el acceso a los objetos del repositorio. Pueden conceder a un usuario el acceso a un contenedor y así a los objetos del contenedor.

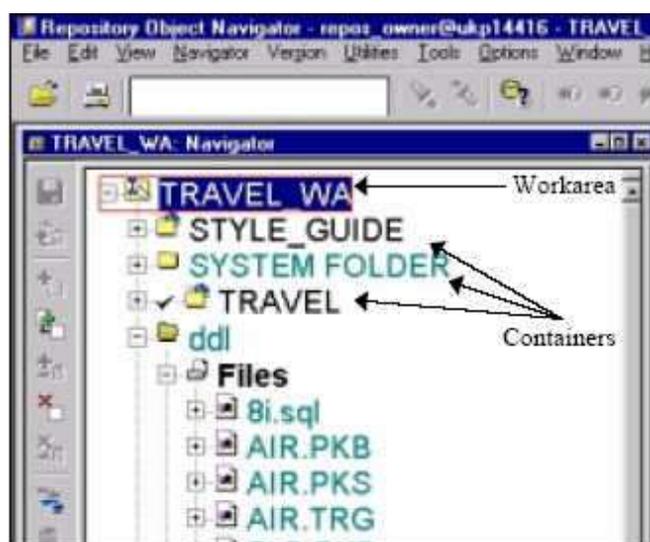


Figura 5-2. Repository Object Navigator mostrando contenedores

5.3.1.3 Administración de datos

Dentro del repositorio puede haber muchas versiones de objetos. Oracle SCM tiene dos mecanismos primarios para la administración de versiones de los objetos del repositorio: áreas de trabajo y configuraciones.

5.3.1.3.1 Áreas de trabajo

Las áreas de trabajo son vistas específicas del repositorio por el cual se puede trabajar sobre los propios objetos (ver figura 5-3). Un usuario individual puede no querer ver una lista de cada versión de cada objeto en el repositorio. Además, es necesario algún tipo de mecanismo para asegurar que un usuario no interfiera con el trabajo de otro usuario. Oracle SCM administra ambos requerimientos con áreas de trabajo.

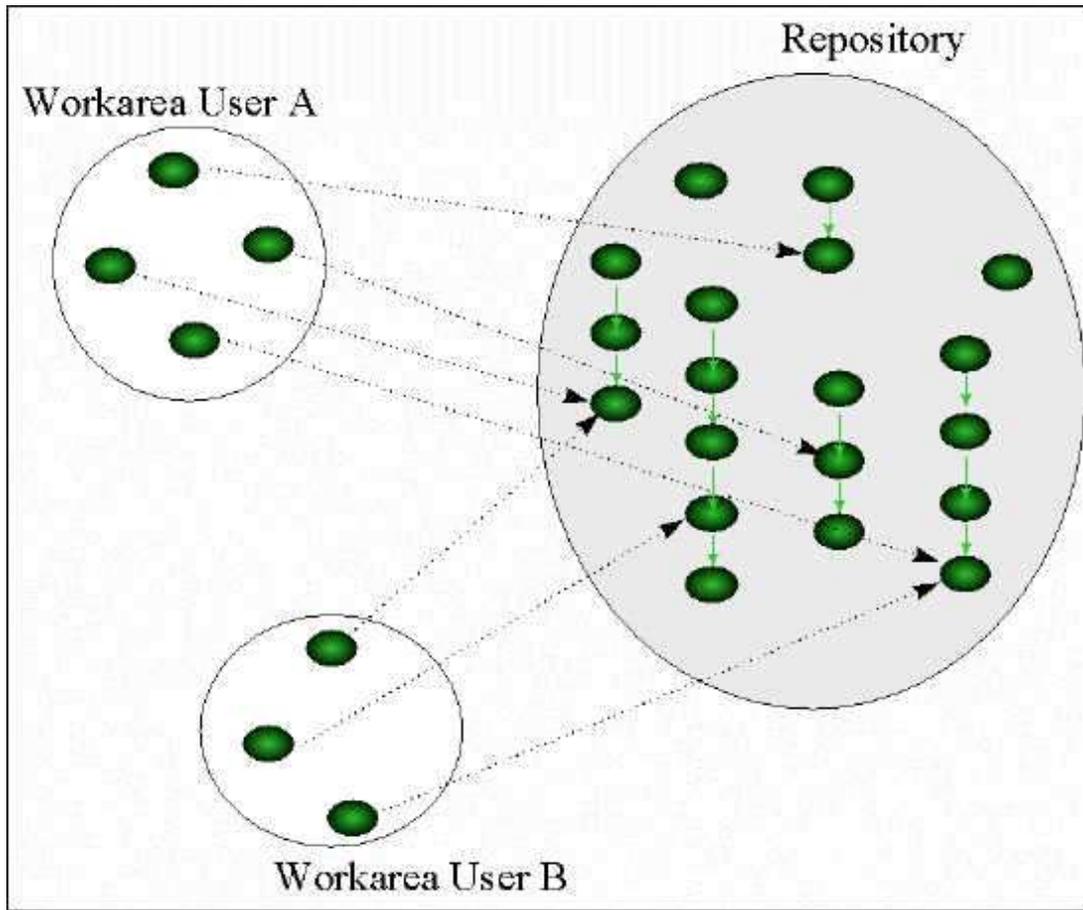


Figura 5-3. Vista de un área de trabajo del repositorio

5.3.1.3.2 Configuraciones

Las configuraciones identifican las versiones específicas de objetos y las agrupan, por ejemplo, construir una suite de prueba o liberar un producto (ver figura 5-4). Una configuración es una colección de versiones de objetos del repositorio que están relacionados de algún modo. Por lo general una configuración detalla todas las versiones de objetos que representan un punto de comprobación en el desarrollo o en el componente de una aplicación, por ejemplo, todas las versiones de objetos que componen una aplicación de pago de nómina.

Una versión de objeto específica que es incluida en una configuración, se dice que es un miembro de aquella configuración. Por ejemplo, cuando el desarrollo de objetos individuales alcanza la etapa donde se puede construir

una aplicación en particular, se tiene que especificar exactamente que versión de cada objeto debe ser usada para construirla. Lo mismo se aplica cuando se ensambla un conjunto de objetos para ser usados para una prueba o incluidos en una liberación de parche - la configuración define qué versiones de qué objetos deben ser usados.

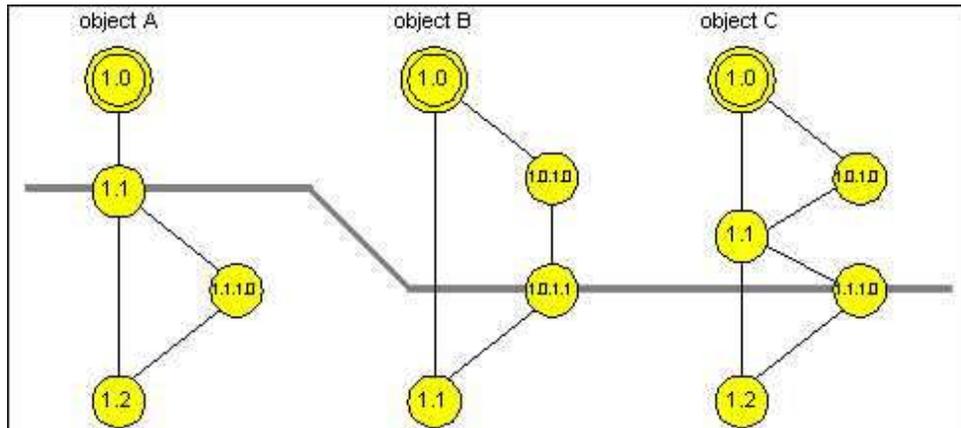


Figura 5-4. Configuración a partir de versiones de objetos

5.3.1.4 Acceso al repositorio

El acceso a todo el repositorio es controlado por la seguridad de la base de datos de Oracle. Quien desee usar el repositorio debe tener una cuenta en la base de datos donde este reside. Sólo un usuario con privilegios de administrador de base de datos (DBA) puede crear cuentas de usuario sobre la base de datos. Cuando el repositorio es instalado, sólo su dueño puede tener acceso al repositorio. Posteriormente, el dueño de depósito puede autorizar a otros usuarios existentes de la base de datos para tener acceso sobre el repositorio.

Una vez que se crea una cuenta, el acceso al repositorio y las tareas que pueden realizar los usuarios son determinados por los privilegios y derechos de acceso que tienen sobre el repositorio. Los privilegios permiten a un usuario ejecutar una herramienta particular del repositorio. Los derechos de acceso

sobre cualquier otro tipo de objeto del repositorio son determinados por el contenedor de objetos propio. Por ejemplo, si se tiene derechos de acceso de Select sobre un contenedor, también se tienen derechos de acceso de Select sobre todos los objetos en ese contenedor.

5.3.1.5 Implementación de Oracle SCM

Como se dijo anteriormente, Oracle SCM es una poderosa y completa herramienta para administrar la configuración del software. Su instalación se puede dividir en dos partes; la que corresponde a la parte servidor y la del cliente.

La parte servidor involucra la instalación de una base de datos en donde se alojará el repositorio, para este proyecto se utilizó una base de datos Oracle9i Enterprise Edition Release 9.2.0.1.0.

La parte cliente es la más compleja ya que involucra tanto la instalación del Oracle SCM como la creación del repositorio. Se instaló Oracle9i Developer Suite con la opción RAD (Rapid Application Development) que instala el Oracle9i SCM entre otras herramientas (Oracle9i JDeveloper, Oracle9i Reports Developer, Oracle9i Forms Developer y Oracle9i Designer). El repositorio se creó a partir de scripts proporcionados por el propio software y ejecutados desde una sesión SQL en el cliente (previo ajuste de algunos parámetros en la base de datos).

Oracle9i SCM se integra directamente con Oracle9i Forms Developer y Oracle9i Reports Developer, lo que permite realizar la gestión de la configuración del software directamente desde estas herramientas. A continuación se mostrará como Oracle9i SCM (por medio de su herramienta Repository Object Navigator) proporciona el control del código fuente, así como sus opciones de comparar objetos, administrar las versiones y liberaciones.

Como se mencionó anteriormente, el sistema académico de TGA está formado básicamente por dos tipos de programas: las pantallas que se realizan con Oracle Forms (archivos .fmb) y los reportes que se realizan con Oracle Reports (archivos .rdf). Para ejemplificar el uso de Oracle SCM se utilizará la

integración con Oracle9i Forms Developer ya que el caso de los reportes es muy similar.

5.3.1.5.1 Áreas de Trabajo

Las áreas de trabajo, al ser vistas particulares de los objetos del repositorio, permiten manejar los permisos de los usuarios sobre dichos objetos. Esto se debe a que el contenido de estas áreas de trabajo está ligado a los privilegios del usuario que las utiliza.

En la figura 5-5 se puede ver el área de trabajo de ejemplo que se utilizará para mostrar el funcionamiento del Oracle SCM.

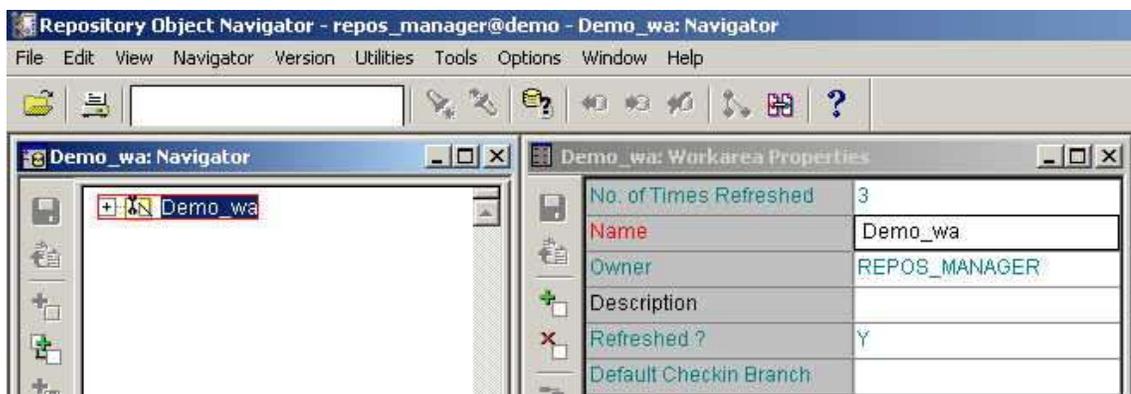


Figura 5-5. Área de trabajo

5.3.1.5.2 Contenedores

Los contenedores son la manera lógica de agrupar los objetos del repositorio y tienen su correspondencia con los directorios en un sistema de archivos; es decir, se mapea el contenedor a una carpeta ubicada en algún disco accesible del servidor de archivos. Esto es necesario ya que el Oracle

Forms Developer no va a editar los archivos contenidos en el repositorio sino que los archivos ubicados en el servidor.

En la figura 5-6 se puede ver el contenedor de ejemplo que se utilizará para mostrar el funcionamiento del Oracle SCM.

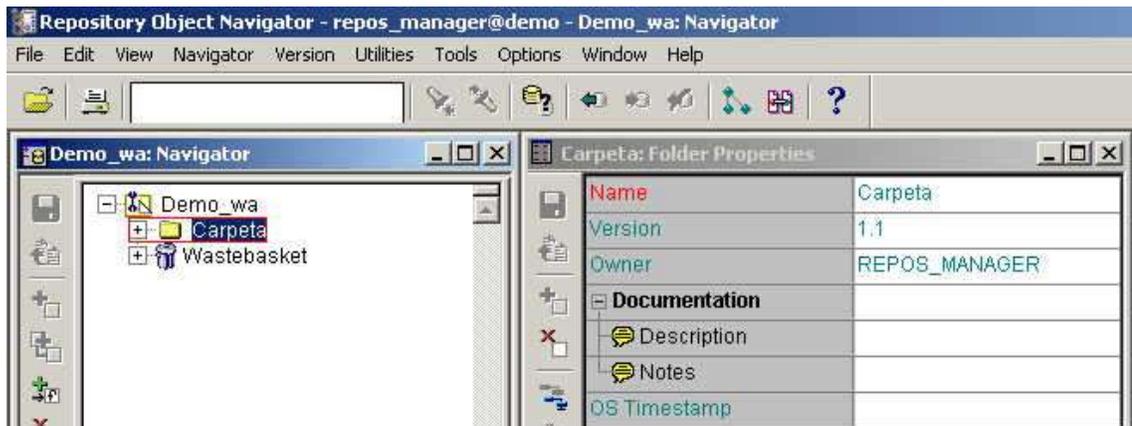


Figura 5-6. Contenedor

5.3.1.5.3 Archivos

Los archivos son los objetos almacenados en el repositorio propiamente tal, en este caso corresponden a archivos fuente de pantallas del sistema académico; es decir, archivos con extensión .fmb.

En la figura 5-7 se pueden ver el archivo de ejemplo que se utilizará para mostrar el funcionamiento del Oracle SCM.

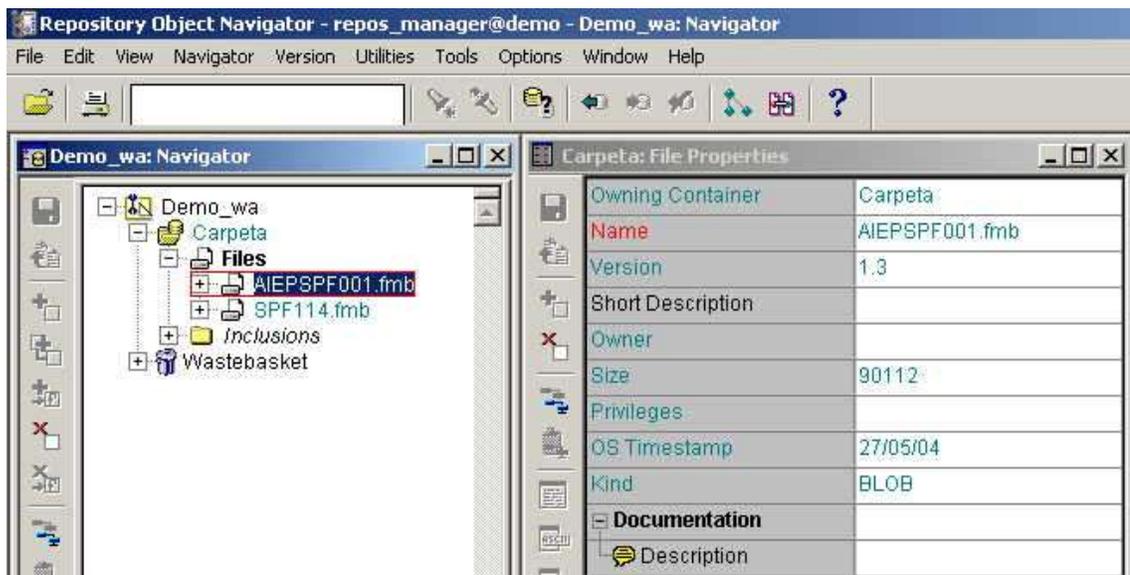


Figura 5-7. Archivo

5.3.1.5.4 Control de Versiones

Una vez que los archivos son cargados al repositorio (proceso denominado check in) estos quedan registrados con la versión 1.0. A su vez, estos archivos son descargados al directorio mapeado en el contenedor, quedando como archivos de modo de lectura solamente.

Para poder editar los archivos, éstos se abren desde el Oracle Forms Developer y se bloquean (proceso denominado check out). Este proceso involucra conectarse al repositorio y descargar una versión del archivo con permisos de escritura (lo que sobrescribe el archivo almacenado en el disco).

Después de hacer los cambios necesarios al archivo, grabar y compilar, este debe ser devuelto al repositorio. Esta acción de desbloquear el archivo (check in) crea su nueva versión.

La figura 5-8 muestra el visor del historial de versiones, herramienta que permite visualizar las distintas versiones de un archivo así como información referente a estas.

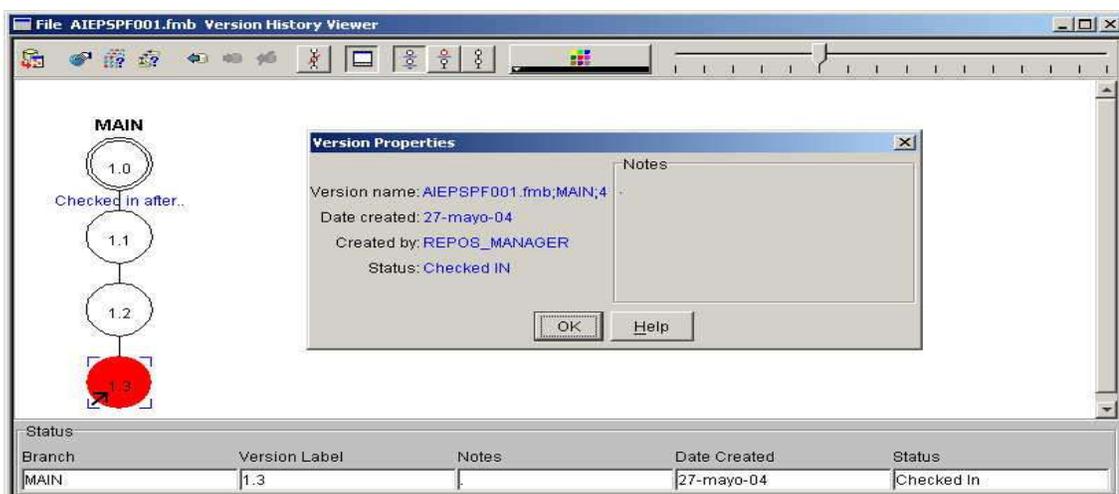


Figura 5-8. Control de versiones

5.3.1.5.5 Comparación de Versiones

Los archivos .fmb no pueden ser comparados usando herramientas de comparación de texto. Oracle SCM incluye la capacidad de comparar archivos .fmb, para que cualquier cambio entre dos versiones de una pantalla pueda ser identificado.

La figura 5-9 muestra la ventana de comparación de objetos, herramienta que permite visualizar cualquier cambio entre dos versiones de un objeto, estos pueden ser de tres tipos: agregado, borrado o actualización.



Figura 5-9. Comparación de versiones

5.3.1.5.6 Combinación

La combinación o fusión es la herramienta de Oracle SCM para administrar el cambio. Esta opción permite fusionar dos versiones de una pantalla; es decir, se fusiona una versión denominada fuente con una versión denominada objetivo y el resultado es la versión objetivo actualizada. La idea es aplicar todos los cambios de la versión objetivo sobre versión fuente; sin embargo, existe la posibilidad de elegir los valores para los campos modificados antes de obtener la versión resultante.

Esta opción es la encargada también de soportar el desarrollo en paralelo; por ejemplo, si un usuario bloquea una pantalla para modificarla pero antes de liberarla otro usuario hizo lo mismo (la bloqueó, modificó y liberó), al querer bloquearla se le avisará lo sucedido dándole la opción de realizar la combinación u omitirla.

En la figura 5-10 se muestra la ventana de combinación, herramienta que permite la fusión de dos versiones de una pantalla. En este ejemplo la versión fuente es la 1.4.1.2 y la versión objetivo es la 1.4.



Figura 5-10. Combinación de versiones

5.3.1.5.7 Configuraciones

Como se mencionó anteriormente, las configuraciones sirven para agrupar versiones específicas de los objetos; es decir, es la herramienta de Oracle SCM para administrar las liberaciones.

Una versión específica de un objeto incluido en una configuración es llamado miembro, y estos pueden ser agregados o eliminados de la configuración cuando se desee. Cada configuración es un repositorio en si mismo y puede por lo tanto ser puesto bajo control de versión.

En figura 5-11 se puede ver una configuración de ejemplo, en este caso corresponde a las últimas versiones de los objetos pertenecientes a la misma línea base.

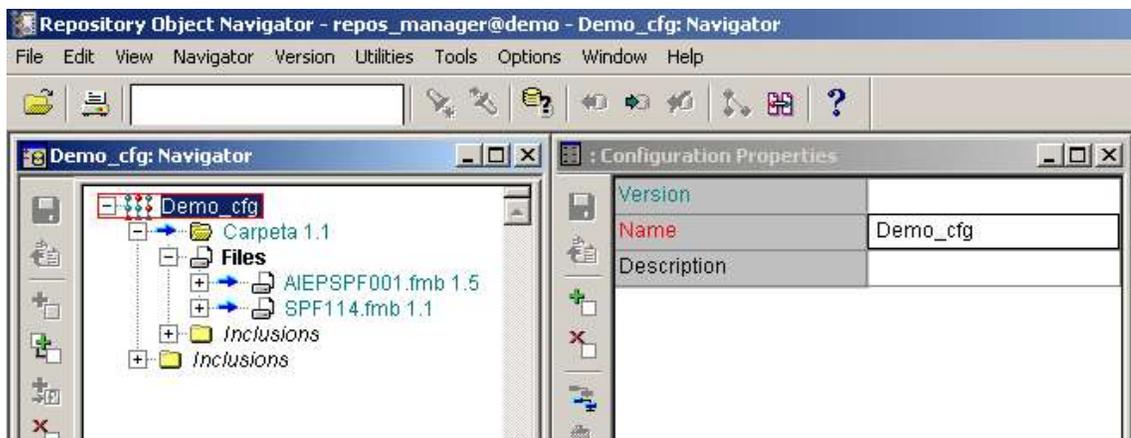


Figura 5-11. Configuración

5.3.1.6 Análisis de impacto

Para muchos usuarios del repositorio, es importante evaluar el impacto sobre el resto de una aplicación al cambiar un componente - proceso conocido como análisis de impacto. Por ejemplo, los desarrolladores podrían investigar cuantas aplicaciones usan una función de PL/SQL particular, y así descubrir si un cambio a la función tendría un impacto significativo en la empresa. Oracle SCM proporciona un conjunto de herramientas que son útiles para el análisis de impacto. Por ejemplo, el analizador de dependencia es una utilidad que analiza

las relaciones entre objetos en una aplicación y permite a los usuarios determinar los efectos de un cambio.

5.3.1.7 Resumen

Oracle SCM proporciona un ambiente de desarrollo multiusuario, administrado y seguro que permite la construcción de aplicaciones complejamente integrados. Oracle SCM almacena información tanto estructurada como no estructurada, asegurando que toda la documentación, definiciones y el código de los proyectos sea almacenada en un solo lugar. Sin la administración de todos los desarrolladores del proyecto se podría trabajar sobre el código incorrecto, gastando valiosos recursos. Como se requiere que los desarrolladores sean más productivos y reaccionen al cambio más rápidamente un ambiente SCM integrado se hace esencial.

5.3.2 Pasos a Producción

Los pasos a producción corresponden a documentos que tienen la finalidad de especificar y definir los pasos necesarios para la puesta en marcha de los desarrollos.

Cada nueva aplicación, ya sea una nueva funcionalidad o la modificación de una existente, se debe realizar en un ambiente de desarrollo. Una vez realizadas todas las pruebas de rigor y con la aprobación respectiva, se debe instalar la nueva aplicación en el ambiente de producción. Dicha instalación está a cargo del personal de operaciones, a quién se le debe instruir cómo se debe proceder para realizar la instalación deseada.

Para poder instruir debidamente al personal de operaciones a la hora de instalar las aplicaciones, se ha diseñado un documento que tiene por objetivo describir todas las acciones necesarias que se deben realizar para lograr una correcta instalación. El documento debe poseer los siguientes puntos:

- *Portada.* Título del documento y su fecha de realización

- *Tabla de contenidos.* Índice del documento
- *Información del documento.* Principales características del documento (autor, cliente, fecha, versión, equipo consultor, autorización, etc.)
- *Cuerpo del documento.* Contiene una breve descripción del documento y a continuación se deben explicar clara y secuencialmente, todos los pasos necesarios para instalar la aplicación en producción, por ejemplo: configuración de la base de datos (roles, dueños, tablas, objetos, sinónimos, usuarios de conexión, etc.), compilación de Developer (pantallas y reportes), etc.

Esta medida permitió eliminar la problemática que surgía cuando una persona realizaba las instalaciones de las aplicaciones en el ambiente de producción y no lo dejaba documentado.

5.3.3 Mesa de Ayuda

Para proporcionar un mejor servicio se implementó una mesa de ayuda, esta consiste en dar soporte a los clientes ya sea por medio del teléfono o vía correo electrónico. Para estos efectos se habilitó una cuenta de correo de soporte, a la cual llegan todos los correos de los clientes y la que todo el personal de soporte tiene configurada. De esta manera se centralizaron los requerimientos de los clientes, los que son encausados por un encargado de soporte, el que los evalúa, les asigna prioridad y finalmente, los distribuye al resto del personal.

Esta medida permitió eliminar la problemática que surgía cuando un cliente solicitaba un requerimiento a una persona en particular, como el requerimiento era enviado a su cuenta de correo nadie más se enteraba de la existencia de tal requerimiento.

6 CONCLUSIONES

Este capítulo presenta las conclusiones y/o posibles mejoras de efectuar al diseño e implementación de un ASP para sistemas académicos abordado en este proyecto de tesis.

De acuerdo a lo anterior, uno de los principales aportes que se estima fue obtenido de este trabajo es la utilización de una metodología llamada Gestión de la Configuración. La cual aporta un instrumento útil para el desarrollo y mantención de software.

Por otro lado, las modificaciones a los procesos de negocio de la aplicación contribuyeron a fortalecer el sistema ya sea mejorando sus funcionalidades o estandarizando las pantallas y reportes que lo componen.

Dentro de las ventajas o aportes obtenidos al utilizar una herramienta de gestión de la configuración, se pueden nombrar los siguientes:

- Integración directa con las herramientas de desarrollo. Esto simplificó la implementación de esta metodología minimizando el impacto en el equipo de desarrollo
- Se logró llevar un control de versiones tanto de los archivos fuente de la aplicación como de sus archivos ejecutables. Entre otras cosas, esto permite eliminar la necesidad de conservar código inservible por miedo a obtener resultados indeseados, ya que ahora se puede recuperar cualquier versión de un programa
- Se pueden comparar fácilmente distintas versiones de un mismo archivo. Gracias al comparador de objetos es posible detectar

cualquier modificación, eliminación o creación de código dentro de un archivo

- Se pueden fusionar fácilmente dos versiones de un mismo archivo. Gracias a la herramienta de fusión de objetos es posible elegir de entre las diferencias que poseen las dos versiones, cual de ellas se desea conservar. Esta cualidad es la que permite la realización de desarrollo en paralelo
- Proporciona tranquilidad al equipo de desarrollo ya que implementa un ambiente de desarrollo seguro y cuidadosamente manejado. Esto se debe a que se cuenta con un repositorio para el almacenamiento de objetos de desarrollo, mecanismos para manejar los objetos de versiones, control del acceso al repositorio, políticas que imponen restricciones a las operaciones sobre versiones, etc.
- Proporciona un mecanismo eficiente y seguro para administrar las liberaciones

Dentro de las ventajas o aportes obtenidos al modificar el sistema académico, se pueden nombrar los siguientes:

- Simplificación en el funcionamiento de las pantallas y reportes que se vieron afectados por la modificación de sus funcionalidades
- Homogenización y estandarización de pantallas y reportes. Se parametrizó toda la información que hace referencia a la institución que utiliza el sistema. Además, se homogenizaron todas las pantallas de parámetros que invocan a los reportes, de manera que los campos iguales efectivamente se llamasen igual, que aquellos del mismo tipo tuviesen el mismo formato, que los campos se desplieguen siempre en un mismo orden preestablecido, etc.
- Eliminación de todos aquellos reportes que se pudiesen obtener como resultado de otro reporte. En este mismo sentido, se logró la consolidación de reportes utilizando mecanismos de filtrado

- Eliminación de reportes que no se utilizaban, ya sea porque su principal información está contenida en otro reporte, es redundante o simplemente porque carece de valor

Dentro de las limitaciones y/o mejoras de este proyecto de tesis se pueden nombrar las siguientes:

- OracleSCM, la herramienta utilizada para gestionar la configuración del software sólo soporta objetos desarrollados con alguna aplicación perteneciente a Oracle Internet Developer Suite, como lo son Oracle JDeveloper, Oracle Forms, Oracle Reports y Oracle Designer.
- Para homogenizar aún más el sistema académico, sería bueno estandarizar todas las pantallas que lo componen, es decir, definir los tipos, tamaños y colores de las letras utilizadas; colores, tamaños y opciones (minimizar, redimensionar, etc.) de las pantallas; colores, tamaños y opciones de los objetos (botones, marcos, etc.); etc.

Como conclusión no puedo dejar de mencionar la experiencia obtenida durante todo el tiempo transcurrido en realizar este proyecto de tesis. Sin duda alguna, aplicar las nociones y conocimientos de ingeniería adquiridos en mi etapa de estudiante, para dar solución a una problemática real de una empresa desarrolladora de software, es muy gratificante.

7 BIBLIOGRAFÍA

- [Pre98] Pressman, R. S. (1998). *Ingeniería del Software: Un Enfoque Práctico*. España: McGraw-Hill/Interamericana de España S.A.U.
- [Som02] Sommerville, I. (2002). *Ingeniería de Software*. México: Pearson Educación de México S.A. de C.V.
- Referencia a publicaciones de páginas Web
 - [URL 1] Humano2 S. A. (2003). Modelo ASP
www.humano2.com/h2_presite/modelo.html
 - [URL 2] Lagash System S.A. (2001). Application Service Providers
www.lagash.com/papers/paper_asp.html
 - [URL 3] Plataforma-ASP.com. Qué es el ASP
www.plataforma-asp.com/asp.html
 - [URL 4] Presentación ONO (2001)
www.aslan.es/cumbre/ponencias/ono.ppt
 - [URL 5] Schuldt, J. E. (1998). Reingeniería de Procesos
www.geocities.com/WallStreet/Exchange/9158/reingen.htm
 - [URL 6] Zavala, J. (2002). Ingeniería de Software
www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html

[URL 7] Oracle Corporation (2004). Oracle 9i SCM and Oracle 9i Forms Integration

<http://otn.oracle.com/products/repository/htdocs/9iscmforms.html>

- Artículo en revista electrónica, dos autores

[Feb02] Febles, A. Pérez, I. Medir el proceso de control de configuración, ¿una utopía para la Industria Nacional de Software? *Revista electrónica del Departamento de Ingeniería Informática y Ciencias de la computación (DIICC) de la Universidad de Concepción*, 9 (ISSN: 0717-4195): Agosto 2003

<http://www.inf.udec.cl/revista/ediciones/edicion9/febles.pdf>